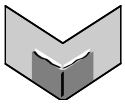
The Journal of Machine Learning Research
Print-Archive Edition

Volume 19
Issues 1–84



Microtome Publishing
Brookline, Massachusetts
www.mtome.com

The Journal of Machine Learning Research
Print-Archive Edition
Volume 19
Issues 1–84

The Journal of Machine Learning Research (JMLR) is an open access journal. All articles published in JMLR are freely available via electronic distribution. This Print-Archive Edition is published annually as a means of archiving the contents of the journal in perpetuity. The contents of this edition are articles published electronically in JMLR in 2018.

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

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

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

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

JMLR Editorial Board

Journal of Machine Learning Research

Volume 19, Issues 1–84

Journal of Machine Learning Research 19 (2018) 1-39 Submitted 4/16; Revised 4/18; Published 8/18

Numerical Analysis near Singularities in RBF Networks

Weili Guo WLGUO@SEU.EDU.CN
Haikun Wei HKWEI@SEU.EDU.CN
Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation
Southeast University
Nanjing, Jiangsu Province 210096, P.R. China

Yew-Soon Ong ASYSONG@NTU.EDU.SG
Jaime Rubio Hervas JHERVAS@NTU.EDU.SG
School of Computer Science and Engineering
Nanyang Technological University,
50 Nanyang Avenue 639798, Singapore

Junsheng Zhao ZHAOJUNSHSHAO@163.COM
School of Mathematics Science
Liaocheng University
Liaocheng, Shandong Province 252059, P.R. China

Hai Wang HWANG@SMU.CA
Sobey School of Business
Saint Mary's University
Halifax, Nova Scotia B3H 3C3, Canada

Kanjian Zhang KJZHANG@SEU.EDU.CN
Key Laboratory of Measurement and Control of CSE, Ministry of Education, School of Automation
Southeast University
Nanjing, Jiangsu Province 210096, P.R. China

Editor: Yoshua Bengio

Abstract

The existence of singularities often affects the learning dynamics in feedforward neural networks. In this paper, based on theoretical analysis results, we numerically analyze the learning dynamics of radial basis function (RBF) networks near singularities to understand to what extent singularities influence the learning dynamics. First, we show the explicit expression of the Fisher information matrix for RBF networks. Second, we demonstrate through numerical simulations that the singularities have a significant impact on the learning dynamics of RBF networks. Our results show that overlap singularities mainly have influence on the low dimensional RBF networks and elimination singularities have a more significant impact to the learning processes than overlap singularities in both low and high dimensional RBF networks, whereas the plateau phenomena are mainly caused by the elimination singularities. The results can also be the foundation to investigate the singular learning dynamics in deep feedforward neural networks.

Keywords: RBF networks, Singularity, Learning dynamics, Numerical analysis, Deep learning

1. Introduction

The results in (Watanabe, 2007) indicate that the parameter spaces of almost all types of learning machines have singular regions where the Fisher information matrices degenerate, including layered neural networks, normal mixtures, binomial mixtures, Bayes networks, hidden Markov models, Boltzmann machines, stochastic context-free grammars, and reduced rank regressions. For the widely used feedforward neural networks, researchers have found that the learning dynamics are affected by the existing singularities. Some strange behaviors occur in the learning process, such as learning dynamics often become very slow and the learning process is trapped in plateaus.

Researchers have realized that such plateau phenomena arise from the singular structure of the parameter space and the Fisher information matrix degenerates at singularities (Fukumizu, 1996; Fukumizu and Amari, 2000; Amari and Ozeki, 2001; Amari et al., 2009). The geometrical structure of such statistical models has been studied by information geometry (Amari and Nagaoka, 2000). The standard statistical paradigm of the Cramer-Rao theorem does not hold at singularities and the model selection criteria, such as Akaike information criterion (AIC), Bayes information criterion (BIC) and minimum description length (MDL), may fail due to the existence of singularities (Amari et al., 2006). The effect of singularity in Bayesian inference was studied in (Watanabe, 2001a,b, 2010; Aoyagi, 2010), and a widely applicable Bayesian information criterion (WBIC) was proposed which remains efficient for the singular model (Watanabe, 2013). Mononen (2015) applied the WBIC to the analytically solvable Gaussian process regression case.

The error function was used instead of traditional log-sigmoid function to investigate online learning dynamics of the multilayer perceptrons (MLPs)(Biehl and Schwarze, 1995; Saad and A.Solla, 1995; Park et al., 2003). Cousseau et al. (2008) used the error function to discuss the learning dynamics of a toy model of MLPs near singularities. Guo et al. (2014, 2015) obtained the analytical expression of averaged learning equations and took the theoretical analysis of learning dynamics near overlap singularities of MLPs. For the Gaussian mixtures, Park and Ozeki (2009) analyzed the dynamics of the EM algorithm around singularities. Radial basis function (RBF) networks are typical feedforward neural networks which have been applied in many fields. Wei et al. (2008) gave a general mathematical analysis of the learning dynamics near singularities in layered networks, and obtained universal trajectories of learning near the overlap singularities. By using the methods in (Wei et al., 2008), Wei and Amari (2008) obtained the averaged learning equations of RBF networks, analyzed the learning dynamics near overlap singularities, and revealed the mechanism of plateau phenomena near the singularities. Nitta (2013, 2015) discussed the singular learning dynamics of complex-valued neural networks. Due to the existence of singularities, the standard gradient method is not Fisher efficient and the gradient descent direction is no longer the steepest descent direction. In order to overcome this problem, natural gradient method was proposed to accelerate the learning dynamics (Rattray et al., 1998; Amari, 1998; Amari et al., 2000; Park et al., 2000; Heskes, 2000; Pascanu and Bengio, 2014; Zhao et al., 2015a).

In recent years, deep learning has become a very hot topic in the machine learning community. Deep neural networks are designed based on traditional neural networks; however, it is very difficult to train deep neural networks by using the Backpropagation (BP)

algorithm. The training is computationally expensive and often presents vanishing gradient problems (Bengio et al., 1994). Till Hinton et al. (2006) proposed deep belief networks to overcome the difficulties by constructing multilayer restricted Boltzmann machines and training them layer-by-layer in a greedy fashion, many types of deep neural networks, including deep Boltzmann machine, deep convolutional neural networks, deep recurrent neural networks etc, have been applied to various fields successively, such as computer vision, pattern classification, natural language processing, nonlinear system identification, etc (Schmidhuber, 2015; Goodfellow et al., 2016).

Due to the much larger number of hidden layers and architecture size, training deep neural networks also faces many challenges (van Hasselt et al., 2016; Gulcehre et al., 2017). On the other hand, the robustness of the training effect cannot be guaranteed, even with a pre-training process (Erhan et al., 2009). Researchers are very interested in what causes the difficulties in training the deep neural networks and various analytical tools are used to study this problem. Goodfellow et al. (2014) provided some empirical evidence that the learning processes did not seem to encounter significant obstacles on a straight path from initialization to solution (obtained via gradient descent method). However, they also puzzled why the training of large models remained slow despite the scarcity of obstacles. Dauphin et al. (2014) came to the conclusion that the training difficulties were originated from the proliferation of saddle points and local minima with high error are exponentially rare in high dimensions. The saddle points caused the long plateaus in the training process. Choromanska et al. (2015) obtained the results that the gradient descent converge to the band of low critical points, and that all critical points found there are local minima of high quality. Lipton (2016) thought that large flat basins in the parameter space were the barrier to training the networks.

From the point of view of singularities in the parameter space, the above results have a certain rationality. From the theoretical results in previous literature and simulation results in this paper, we can find that the points in the elimination singularity are saddles, the points in the overlap singularity are local minima (in the batch mode learning) and the generalization error surface near the overlap singularity is very flat. It would be much clearer if the analytical form of Fisher information matrix of such deep neural networks is obtained.

Besides, Saxe et al. (2014) investigated the deep linear neural networks and found that the error did not change under a scaling transformation. This would cause the training difficulty which was called scaling symmetries in (Dauphin et al., 2014). The scaling symmetries are very similar to elimination singularities which will be discussed in Section 3. These results can be applied to a more general case. For instance, deep belief nets are based on the restricted Boltzmann machine. However, the restricted Boltzmann machine is singular, which implies the learning dynamics of deep belief nets may be seriously affected by the singularities. The learning processes of deep convolutional neural networks and deep multilayer perceptrons also face this problem. The analytical results of learning dynamics near singularities in shallow neural networks can be generalized to the deep neural networks and improve the learning efficiency. Due to overfitting issues in deep learning and the singular structure of the learning machine, it is worthy to analyze the influence of singularities in the deep neural networks in the future.

Currently, the effects of singularities to the learning dynamics of neural networks are still unknown and, therefore, it is important to examine the learning dynamics near singularities. As there are only two types of singularities (i.e. overlap and elimination singularities) in the parameter space of RBF networks and Wei and Amari (2008) has obtained the analytical form of averaged learning equations, we choose the RBF networks as the research objective in this paper. Based on the theoretical analysis results, we numerically analyze the learning dynamics near singularities through a large number of simulation experiments. From the results in (Wei and Amari, 2008; Park and Ozeki, 2009; Guo et al., 2015), it can be seen that the learning dynamics near singularities are similar in RBF networks, multilayer perceptrons and Gaussian mixtures. Thus, though the analysis is taken based on RBF networks in this paper, the statistical results can also reflect other feedforward neural networks.

For typical RBF networks with k hidden units:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x}, \boldsymbol{J}_i), \tag{1}$$

where $\boldsymbol{x} \in \mathcal{R}^n$ denotes the input vector, $\boldsymbol{J}_i \in \mathcal{R}^n$ is the center vector for neuron i, and w_i is the weight of neuron i in the linear output neuron. $\phi(\cdot)$ denotes the Gaussian function and $\phi(\boldsymbol{x}, \boldsymbol{J}_i) = \exp(-\frac{\|\boldsymbol{x}-\boldsymbol{J}_i\|^2}{2\sigma^2})$, $\boldsymbol{\theta} = \{\boldsymbol{J}_1, \cdots, \boldsymbol{J}_k, w_1, \cdots, w_k\}$ represents all the parameters of the model.

Next we introduce two types of singularities. If two hidden units i and j overlap, i.e. $\boldsymbol{J}_i = \boldsymbol{J}_j$, then $w_i\phi(\boldsymbol{x}, \boldsymbol{J}_i) + w_j\phi(\boldsymbol{x}, \boldsymbol{J}_j) = (w_i + w_j)\phi(\boldsymbol{x}, \boldsymbol{J}_i)$ remains the same value when $w_i + w_j$ takes a fixed value, regardless of particular values of w_i and w_j. Therefore, we can identify their sum $w = w_i + w_j$, nevertheless, each of w_i and w_j remains unidentifiable. When one output weight $w_i = 0$, $w_i\phi(\boldsymbol{x}, \boldsymbol{J}_i) = 0$, whatever value $\boldsymbol{J}_i$ takes. These are the only two types of singularities existed in the parameter space of RBF networks (Fukumizu, 1996; Wei and Amari, 2008):
(1) Overlap singularity:

$$\mathcal{R}_1 = \{\boldsymbol{\theta} | \boldsymbol{J}_i = \boldsymbol{J}_j\},$$

(2) Elimination singularity:

$$\mathcal{R}_2 = \{\boldsymbol{\theta} | w_i = 0\}.$$

In this paper, we first derive the explicit expression of the Fisher information matrix for RBF networks. Secondly, we use the average learning equations (ALEs) to investigate the batch mode learning dynamics of RBF networks. A large number of numerical simulations are conducted. By judging whether the Fisher information matrix degenerates and tracing important variables of numerical simulations, we evaluate the learning processes which are seriously affected by the two types of singularities. We also examine the effects of the existence of singularities to RBF networks.

The rest of the paper is organized as follows. Section 2 shows the analytical expression of Fisher information matrix of RBF networks. Section 3 contains the numerical analysis near singularities for various specific cases. Finally, Section 4 presents our conclusions.

2. Analytical Expression of Fisher Information Matrix in RBF Networks

As the singularities are the regions where the Fisher information matrix of system parameters degenerates, the Fisher information matrix can be seen as an important indicator to judge whether the learning process has arrived to the singularities. We show the explicit expression of the Fisher information matrix in this section.

In the case of regression, we have a number of observed data $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_t, y_t)$, which are generated by an unknown teacher function:

$$y = f_0(\boldsymbol{x}) + \varepsilon, \tag{2}$$

where ε is an additive noise, usually subject to Gaussian distribution with zero mean.

We also assume that the training input is subject to a Gaussian distribution with zero mean and a covariance matrix $\boldsymbol{\Sigma}$:

$$q(\boldsymbol{x}) = (\sqrt{2\pi})^{-n} |\boldsymbol{\Sigma}|^{-\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{x}\right). \tag{3}$$

As the covariance matrix plays a constant term role in the numerical analysis process and does not essentially influence the analytical results, without loss of generality, we choose the covariance to be the identity matrix, namely $\boldsymbol{\Sigma} = \mathbf{I}$. $q(\boldsymbol{x})$ can be generalized as an uniform distribution (Wei and Amari, 2008).

For the RBF networks (1), the Fisher information matrix is defined as follows (Amari and Nagaoka, 2000):

$$F(\boldsymbol{\theta}) = \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right\rangle, \tag{4}$$

where $\langle \cdot \rangle$ denotes the expectation with respect to the teacher distribution. The teacher distribution is given by:

$$p_0(y, \boldsymbol{x}) = q(\boldsymbol{x}) \frac{1}{\sqrt{2\pi}\sigma_0} \exp\left(-\frac{(y - f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right). \tag{5}$$

Then by using the results obtained in (Wei and Amari, 2008) and taking further calculations, we can obtain the following theorem:

Theorem 1 *The explicit expression of Fisher information matrix for RBF networks is:*

$$\begin{aligned} F(\boldsymbol{\theta}) &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{\theta}^T} \right\rangle \\ &= \begin{bmatrix} F_{11} & \cdots & F_{1k} & F_{1(k+1)} & \cdots & F_{1(2k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ F_{k1} & \cdots & F_{kk} & F_{k(k+1)} & \cdots & F_{k(2k)} \\ F_{(k+1)1} & \cdots & F_{(k+1)k} & F_{(k+1)(k+1)} & \cdots & F_{(k+1)(2k)} \\ \vdots & & \vdots & \vdots & & \vdots \\ F_{(2k)1} & \cdots & F_{(2k)k} & F_{(2k)(k+1)} & \cdots & F_{(2k)(2k)} \end{bmatrix}, \end{aligned} \tag{6}$$

where:

$$C(\boldsymbol{J}_i, \boldsymbol{J}_j) = \left(\frac{\sigma^2}{\sigma^2 + 2}\right)^{\frac{N}{2}} \exp\left(-\frac{\sigma^2(\|\boldsymbol{J}_i\|^2 + \|\boldsymbol{J}_j\|^2) + \|\boldsymbol{J}_i - \boldsymbol{J}_j\|^2}{2\sigma^2(\sigma^2 + 2)}\right), \tag{7}$$

$$B(\boldsymbol{J}_i, \boldsymbol{J}_j) = -\frac{\sigma^2 \boldsymbol{J}_j - (\boldsymbol{J}_i - \boldsymbol{J}_j)}{\sigma^2(\sigma^2 + 2)}, \tag{8}$$

$$\left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_j)}{\partial \boldsymbol{J}_j^T} \right\rangle = \frac{C(\boldsymbol{J}_i, \boldsymbol{J}_j)}{\sigma^2(\sigma^2 + 2)} \left(\mathbf{I}_n + (\boldsymbol{J}_j - (\sigma^2 + 1)\boldsymbol{J}_i) B^T(\boldsymbol{J}_i, \boldsymbol{J}_j)\right), \tag{9}$$

$$\mathbf{I}_n \text{ is the compatable identity matrix.} \tag{10}$$

For $1 \le i \le k, \quad 1 \le j \le k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_i} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_j^T} \right\rangle = w_i w_j \left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_j)}{\partial \boldsymbol{J}_j^T} \right\rangle \\ &= w_i w_j \frac{C(\boldsymbol{J}_i, \boldsymbol{J}_j)}{\sigma^2(\sigma^2 + 2)} \left(I_n + (\boldsymbol{J}_j - (\sigma^2 + 1)\boldsymbol{J}_i) B^T(\boldsymbol{J}_i, \boldsymbol{J}_j)\right). \end{aligned} \tag{11}$$

For $1 \le i \le k, \quad k+1 \le j \le 2k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_i} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{j-k}} \right\rangle = w_i \left\langle \frac{\partial \phi(\boldsymbol{x}, \boldsymbol{J}_i)}{\partial \boldsymbol{J}_i} \phi(\boldsymbol{x}, \boldsymbol{J}_{j-k}) \right\rangle \\ &= w_i C(\boldsymbol{J}_i, \boldsymbol{J}_{j-k}) B(\boldsymbol{J}_i, \boldsymbol{J}_{j-k}). \end{aligned} \tag{12}$$

For $k+1 \le i \le 2k, \quad 1 \le j \le k,$

$$F_{ij} = \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{i-k}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial \boldsymbol{J}_j^T} \right\rangle = F_{ji}^T. \tag{13}$$

For $k+1 \le i \le 2k, \quad k+1 \le j \le 2k,$

$$\begin{aligned} F_{ij} &= \left\langle \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{i-k}} \frac{\partial f(\boldsymbol{x}, \boldsymbol{\theta})}{\partial w_{j-k}} \right\rangle = \langle \phi(\boldsymbol{x}, \boldsymbol{J}_{i-k}) \phi(\boldsymbol{x}, \boldsymbol{J}_{j-k}) \rangle \\ &= C(\boldsymbol{J}_{i-k}, \boldsymbol{J}_{j-k}). \end{aligned} \tag{14}$$

■

Remark 1: When the Fisher information matrix is near singular, the condition value of the matrix becomes very large, namely, the inverse of the condition value is near to 0. Thus the inverse of the condition value can be used to measure how close the system parameters are to the singularities. In the following numerical analysis, we record the inverse of condition value of the Fisher information matrix to show the influence of singularities on the learning process more clearly.

Remark 2: By adding the inverse of Fisher information matrix as an coefficient to the weights update in the standard gradient descent algorithm, researchers proposed the natural gradient descent method to overcome or decrease the serious influence of the singularities. Thus the Fisher information matrix plays a key role in natural gradient descent method. This means that besides being the fundamental in the following numerical analysis, obtaining the analytical form of Fisher information matrix can greatly help us in designing the modified natural gradient descent algorithms with better performance in the future.

3. Numerical Analysis near Singularities

After having obtained the analytical form of Fisher information matrix in Theorem 1, we numerically analyzed the learning dynamics of RBF networks by taking four experiments in this section, where the specific learning dynamics influenced by different types of singularities are shown and the experiment results are statistically analyzed. In Section 3.1 and Section 3.2, we conduct artificial experiments for low and medium dimensional cases, where the input distribution is known. For these cases, the Fisher information matrix can be obtained by using Theorem 1, and the relation between the stage where the singular learning dynamics occur and the stage where the Fisher information matrix degenerates can be clearly observed. In Section 3.3, the experiment for high dimensional case is carried out by a real data set to investigate the effects of the singularities.

3.1 Two-hidden-unit RBF Networks

The results obtained in (Wei and Amari, 2008) indicate that the batch mode learning dynamics are very similar to the averaged learning dynamics and we can use the averaged learning equations (ALEs) to investigate the dynamics in batch mode learning. Moreover, the ALEs do not depend on any specific sample data set which can overcome the disturbance caused by randomness of the model noise. Besides, as the ALEs are ordinary differential equations (ODEs), and for the given teacher parameters and initial values of student parameters, the learning processes of the student parameters can be obtained by solving ODEs. Thus, in this section, we use the ALEs to perform the experiments, where the analytical form of ALEs in RBF networks has been obtained in (Wei and Amari, 2008).

The student RBF network is defined in Eq.(1). We also assume that the teacher function is described by a RBF network with s hidden units:

$$f_0(\boldsymbol{x}) = f(\boldsymbol{x}, \boldsymbol{\theta_0}) + \varepsilon = \sum_{i=1}^{s} v_i \phi(\boldsymbol{x}, \boldsymbol{t}_i) + \varepsilon, \tag{15}$$

where ε denotes zero mean Gaussian additive noise that is uncorrelated with training input $\boldsymbol{x}$. When the true teacher function $f_0(\boldsymbol{x})$ cannot be represented by a RBF network, $f(\boldsymbol{x}, \boldsymbol{\theta}_0)$ is assumed to be its best approximation by the RBF network.

The analytical form of ALEs is as follows (Wei and Amari, 2008):

$$\dot{\boldsymbol{J}}_i = \eta w_i \left(\sum_{j=1}^{s} v_j C(\boldsymbol{t}_j, \boldsymbol{J}_i) B(\boldsymbol{t}_j, \boldsymbol{J}_i) - \sum_{j=1}^{k} w_j C(\boldsymbol{J}_j, \boldsymbol{J}_i) B(\boldsymbol{J}_j, \boldsymbol{J}_i) \right), \tag{16}$$

$$\dot{w}_i = \eta \left(\sum_{j=1}^{s} v_j C(\boldsymbol{t}_j, \boldsymbol{J}_i) - \sum_{j=1}^{k} w_j C(\boldsymbol{J}_j, \boldsymbol{J}_i) \right), \tag{17}$$

where $i = 1, 2, \cdots, k$. $C(\boldsymbol{t}, \boldsymbol{J})$ and $B(\boldsymbol{t}, \boldsymbol{J})$ have the same meanings with Eq.(7) and Eq.(8), respectively.

The generalization error is:

$$\begin{aligned} L(\boldsymbol{\theta}) &= \left\langle \frac{1}{2} \left(f(\boldsymbol{x}, \boldsymbol{\theta}_0) - f(\boldsymbol{x}, \boldsymbol{\theta}) \right)^2 \right\rangle \\ &= \frac{1}{2} \sum_{i,j} v_i v_j C(\boldsymbol{t}_i, \boldsymbol{t}_j) - \sum_{i,j} v_i w_j C(\boldsymbol{t}_i, \boldsymbol{J}_j) + \frac{1}{2} \sum_{i,j} w_i w_j C(\boldsymbol{J}_i, \boldsymbol{J}_j). \end{aligned} \tag{18}$$

Results in (Wei and Amari, 2008) indicate that investigating the model with two hidden units is enough to capture the essence of the learning dynamics near singularities. Therefore, we first perform the numerical analysis of the RBF networks with two hidden units, and we then analyze the RBF networks in a more general case in the following sections. The learning dynamics of RBF networks are all obtained by solving ALEs for the given teacher parameters and initial student parameters.

In this subsection we analyze the case where the teacher and student models both have two hidden units.

The student model has the following form:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = w_1 \phi(\boldsymbol{x}, \boldsymbol{J}_1) + w_2 \phi(\boldsymbol{x}, \boldsymbol{J}_2). \tag{19}$$

The teacher model is also described by a RBF network with two hidden units:

$$f(\boldsymbol{x}, \boldsymbol{\theta_0}) = v_1 \phi(\boldsymbol{x}, \boldsymbol{t}_1) + v_2 \phi(\boldsymbol{x}, \boldsymbol{t}_2) + \varepsilon. \tag{20}$$

We choose the spread constant $\sigma = 0.5$.

In order to investigate the influence of the singularities in the learning process of RBF networks more accurately, we mainly focus on input $\boldsymbol{x}$ with dimension 1. For this type of RBF networks, the global minimum is the point where the generalization error is 0, which makes easier to analyze the simulation results.

3.1.1 Toy Model of RBF Networks

In order to visually represent the learning trajectories of parameters in the loss error surface and given that a 3D figure can only show three parameters, we initially focus on a special case of RBF networks, where part of the student parameters will remain invariable in the training process.

In the case of overlap singularity, we choose the teacher model parameters v_1 and v_2 to be the initial state of w_1 and w_2, and only J_1 and J_2 will be modified in the learning process. In all the other cases, the weights J_2 and w_2 are fixed to be the same as the teacher parameters t_2 and v_2, and therefore, only J_1 and w_1 will be modified in the learning process. Thus, for all cases, there are only two variable parameters: J_1-J_2 or J_1-w_1. When the learning process has been completed, we can plot the learning trajectories of parameters through the generalization error surface in a 3D figure. Although the student model is a toy model, the simulation results can show the influence of singularities during the learning process in a direct and visual manner.

In what follows, the teacher model is chosen as: $t_1 = -1.95$, $t_2 = -0.90$, $v_1 = 1.35$, $v_2 = 1.72$, the width spread $\sigma = 0.5$. The main reason behind only choosing one teacher function is to illustrate that the learning process of a RBF network can be affected by all

the types of singularities under different initial states. For a given initial state of student parameters, the learning trajectories of J_i and w_i can be obtained by solving Eqs.(16) and (17). The generalization error trajectory and error surface can also be obtained from Eq.(18) after J_i and w_i have been calculated.

By analyzing the simulation results, we list all the cases of learning processes as follows. In the following figures, '$\circ$' and '$\times$' represent the initial state and final state, respectively.

Case 1 (**Fast convergence**) : The learning process converges to the global minimum fast.

In this case, the singularities do not affect the learning process and the learning dynamics quickly converge to the global minimum after the beginning of learning process. An example is provided in Figure 1, which represents the trajectories in a log scale of the inverse of the condition number, generalization error and learning trajectory in the generalization error surface, respectively. In the training process, J_2 and w_2 remain invariable. As shown in Figure 1, the parameters J_1 and w_1 directly converge to the global minimum (Figure 1(c)) and the Fisher information matrix remains regular (Figure 1(a)).

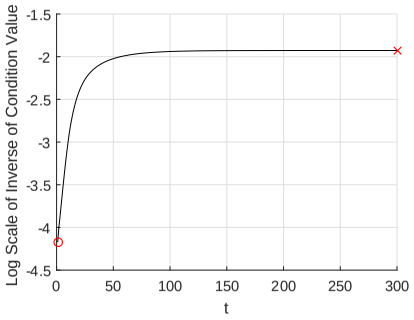
Case 2 (**Overlap singularity**) :The learning process is significantly affected by overlap singularity.

In this case, the learning trajectories of parameters J_1 and J_2 arrive at the overlap singularity, namely $J_1 = J_2$. An example is given in Figure 2, which shows the trajectories of log scale of inverse of condition number, generalization error, weights J_i and learning trajectory in the generalization error surface, respectively. In the training process, w_1 and w_2 remain invariable.

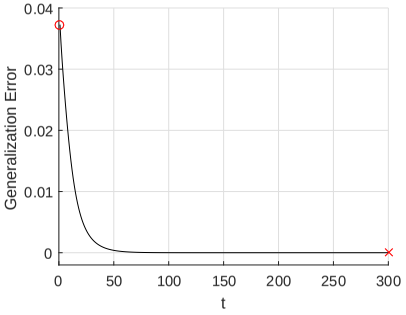
From Figure 2(a), we can see that the inverse of the condition number of Fisher information matrix gets closer and closer to 0 as the training process runs, and finally smaller than $10e-15$ which means that the Fisher information matrix nearly degenerates. Meanwhile, J_1 and J_2 nearly equal to each other (Figure 2(c)), namely the parameters arrive at the overlap singularity. The generalization error descents fast at the beginning of the learning process, and after J_1 and J_2 nearly overlap, the generalization error changes slightly. As shown in Figure 2(d), the generalization error surface is very flat near the final state of J_1 and J_2, which indicates that the parameters present difficulties escaping from the overlap singularity. In order to explore what causes the difficulties in training large-scale networks, (Lipton, 2016) revealed the high degree of nonlinearity in the learning path by analyzing the learning trajectories using the 2D PCA and thought that the large flat regions of the weight space hinder the learning process. From Figure 2(d), we can see that the error surface near the overlap singularity is very flat. Due to the so flat error surface around the overlap singularity, the learning may become very slow even if the two hidden units do not equal to each other exactly.

Remark 3: It can be seen that the log scale of the inverse of the condition number obviously fluctuates at the end of the learning process (Figure 2(a)). We think this is mainly because the value is too small (smaller than $10e-15$), and even a slight change of the parameters would cause the obvious fluctuation of the condition number of the Fisher information matrix due to the limit to the degree of accuracy of computer.

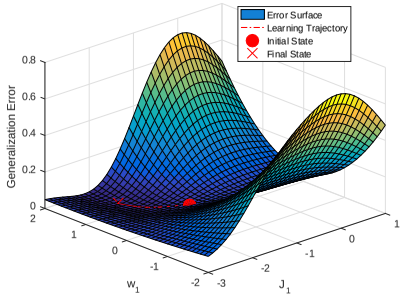
Case 3 (**Cross elimination singularity**): The learning process crosses the elimination and reaches the global optimum after training.



(a) Trajectory of log scale of inverse of condition value

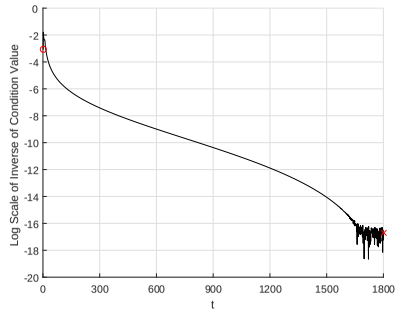


(b) Time evolution of generalization error

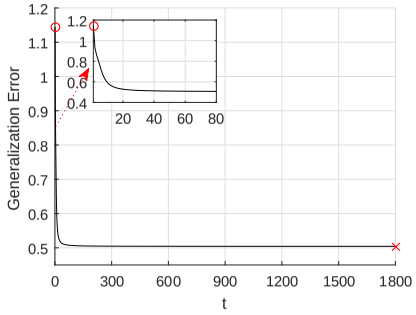


(c) Learning trajectory in generalization error surface

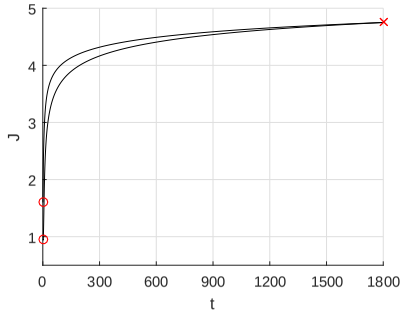
Figure 1: Case 1 (Fast convergence) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.30$, $w_1^{(0)} = 0.57$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.



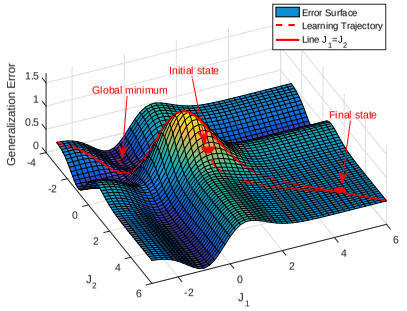
(a) Trajectory of log scale of inverse of condition value



(b) Time evolution of generalization error



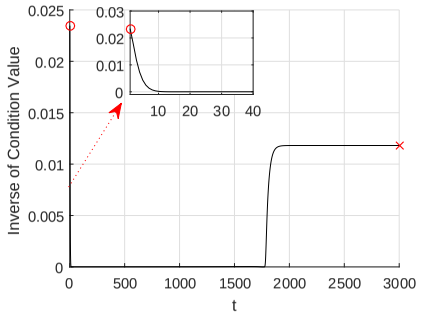
(c) Time evolution of J_1 and J_2



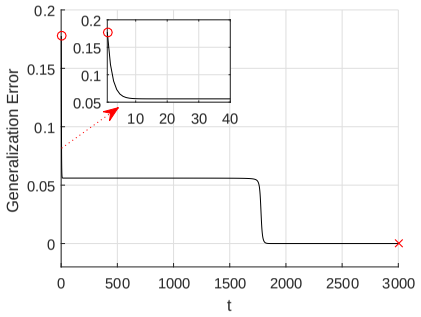
(d) Learning trajectory in generalization error surface

Figure 2: Case 2 (Overlap singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 1.60$, $J_2^{(0)} = 0.95$, $w_1^{(0)} = v_1$, $w_2^{(0)} = v_2$. In the training process w_1 and w_2 remain invariable. The final student parameters are $J_1 = 4.7504$ and $J_2 = 4.7504$.

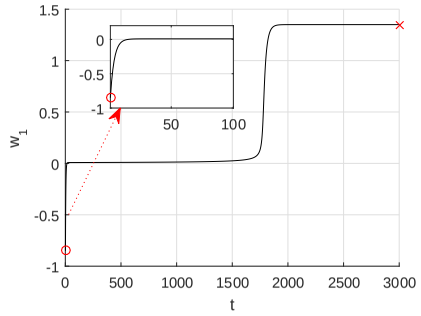
When the learning process arrives at the elimination singularity, e.g. $w_1 = 0$, the term $w_1\phi(x, J_1)$ vanishes. Hence J_1 does not affect the behavior of $f(x, \boldsymbol{\theta})$ and is not identifiable on the subspace $w_1 = 0$. An example is shown in Figure 3, where the learning process arrived at the elimination singularity and finally reached the global optimum after crossing it. Figure 3 shows the trajectories of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.



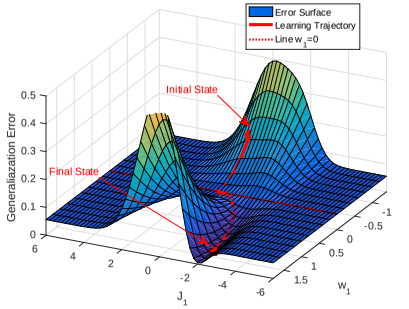
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w_1



(d) Learning trajectory in generalization error surface

Figure 3: Case 3 (Cross elimination singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.18$, $w_1^{(0)} = -0.85$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.

From the trajectory of the inverse of the condition value of Fisher information matrix in Figure 3(a), the Fisher information became nearly singular at the early stage of training and remain so for some time. w_1 nearly equaled 0 at this stage (Figure 3(c)) which means

that the learning process has arrived at the elimination singularity. It can be clearly seen from Figure 3(d) that the points on the line $w_1 = 0$ are all saddle points. Then the student parameters randomly walked around $w_1 = 0$ and finally the learning process skipped the elimination singularity and the student model exactly learned the teacher model. An obvious plateau phenomenon can be observed during the learning process as shown in Figure 3(b).

Case 4 (**Near elimination singularity**): When the student parameters are near the elimination singularity in the training, the learning process is significantly affected by the elimination singularity.

In our simulation experiments, we observed another case in which, when w_1 is close to 0 but not equal to 0, the learning process is also significantly affected by elimination singularity. Then the parameters do not skip the elimination singularity and reach the global optimal points. Figure 4 shows the trajectories of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.

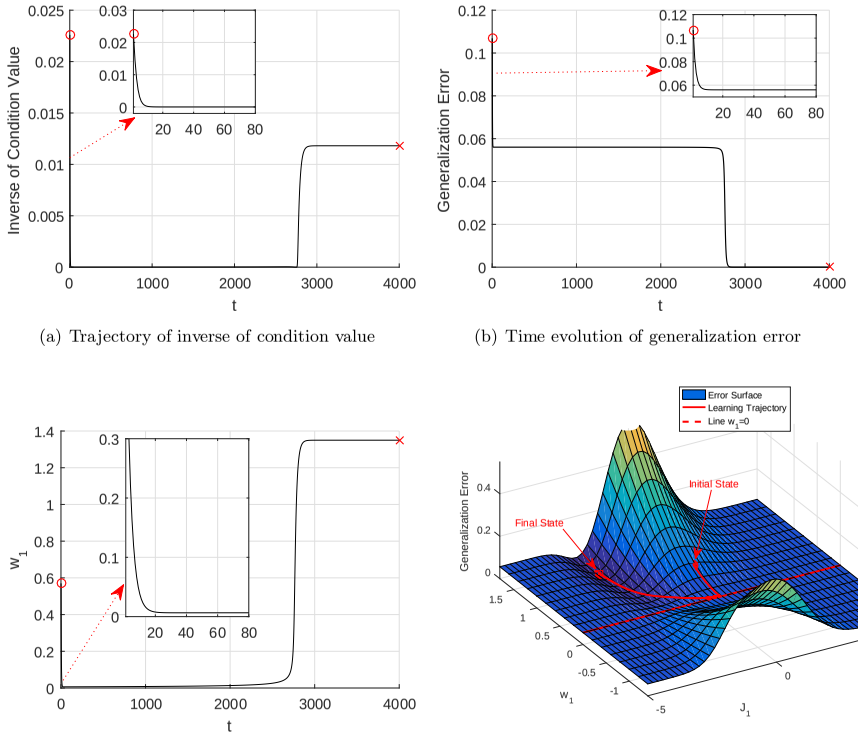
The two learning processes are similar to each other by comparing Figure 4(a) with Figure 3(a) and Figure 4(b) with Figure 3(b), respectively. However, the trajectory of w_1 in Figure 4(c) shows that w_1 is close to 0 during the training process but does not equal 0. During the stage where w_1 approaches 0 and departs from it, the learning process is significantly affected and a plateau phenomenon can clearly be observed. This means that the elimination singularity will significantly affect the learning process even if the parameters are only near to it.

By investigating the deep linear neural networks, (Saxe et al., 2014) obtained a case similar to the elimination singularity that slows down the learning process. The equation of error E is derived as $E(a,b) = \frac{1}{2\tau}(s-ab)^2$, where τ represents the inverse of the learning rate, s represents the input-output correlation information, a represents the weight from the input node to the hidden layer and b represents the weight from the hidden layer to the output node. Obviously $b = 0$ represents the elimination singularity. It can be seen that the error did not change under the scaling transformations $a \to \lambda a$, $b \to \frac{b}{\lambda}$. $a = 0$, $b = 0$ is also a fixed point. As shown in Figure 2 in (Saxe et al., 2014), we can see that certain directions of the learning point to $a = 0$, $b = 0$ which implies the parameters will converge to the point at first under an appropriate initial state. As the point $a = 0$, $b = 0$ is not stable, the parameters will escape from it and finally converge to the global minimum. During this process, long plateau can be observed. This is basically the same as with the learning trajectories in Figure 3(d) and Figure 4(d). The results illustrate the importance of investigating the singularities in deep neural networks.

Case 5 (**Output weight 0**) : After training, output weight w_1 becomes nearly equal to 0.

In the simulation experiments, we also observe that sometimes the output weight w_1 becomes nearly equal to 0 after training. Even if the training process lasts longer, the weight also remains nearly 0. We give an example of this case in Figure 5. Figure 5 shows the trajectories of log scale of the inverse of the condition number, generalization error, weight w_1 and learning trajectory in the generalization error surface, respectively.

From Figure 5(d), it can be seen that w_1 quickly drops to 0 at the beginning of the training, and does not escape from it till the end. Even if we continue the training process for a longer time, the student parameters remain almost unchanged. This is mainly because



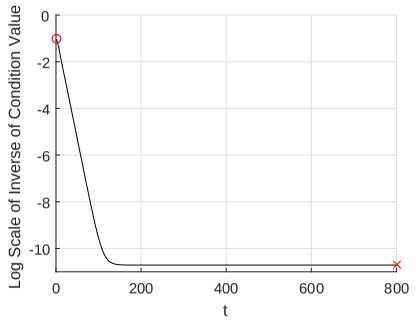
(a) Trajectory of inverse of condition value

(b) Time evolution of generalization error

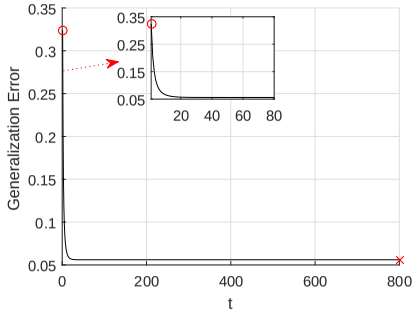
(c) Time evolution of w_1

(d) Learning trajectory in generalization error surface

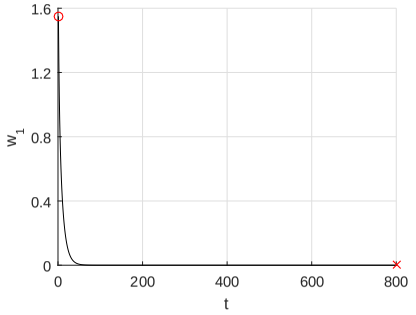
Figure 4: Case 4 (Near elimination singularity) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.30$, $w_1^{(0)} = 0.57$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = -1.95$ and $w_1 = 1.35$.



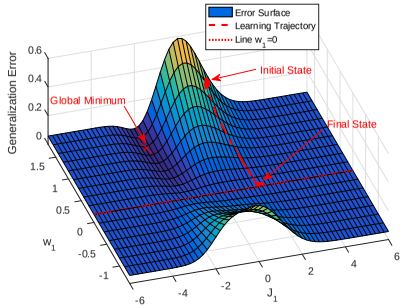
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w_1



(d) Learning trajectory in generalization error surface

Figure 5: Case 5 (Output weight 0) in toy model of RBF networks
The initial student parameters are $J_1^{(0)} = 0.95$, $w_1^{(0)} = 1.55$, $J_2^{(0)} = t_2$, $w_2^{(0)} = v_2$. In the training process J_2 and w_2 remain invariable. The final student parameters are $J_1 = 1.6192$ and $w_1 = 0$.

the radial basis function has little effect on a region that far from the center. When the centers of the teacher and the student are very far from each other, the student cannot exactly approximate the teacher and the output weight w_1 will become zero in order to avoid a bigger error. In this example, the initial student center J_1 is far away from the teacher center t_1, and w_1 is close to 0 after training. In this case, the student model is trapped in local optimum after the training process.

Hitherto, four cases of interesting learning processes in RBF networks have been visually introduced. In addition to the overlap singularity case, the other cases are actually one-hidden-unit RBF network to approximate one-hidden-unit RBF network. Even under only one hidden unit situation, the learning dynamics of RBF networks are still seriously affected by singularities.

In summary, 1) in the overlap singularity case, as the generalization error surface is very flat around the overlap singularity, the parameters cannot escape from it once they have been affected by overlap singularity. 2) For the elimination singularity case, it can be seen that the points in the elimination singularity are saddles, where part of the region is in a local minimum direction and another part of the region is in a local maximum direction. When the learning process arrives near the elimination singularity by local minimum direction, the parameters walk randomly on the singularity till they arrive at the local maximum direction, then the parameters converge to the global minimum. During the random walk stage, a plateau phenomenon can be obviously observed. If the parameters can not walk to the local maximum direction (mainly because the student center is far from the teacher center), the output weight finally nearly equals 0.

In the following subsection, we investigate the case of two-hidden-unit RBF networks approximated by normal two-hidden-unit RBF networks.

3.1.2 RBF Networks with Two Hidden Units

In this subsection, we consider three cases of v_1 and v_2: (1) v_1 and v_2 are both positive; (2) v_1 and v_2 are both negative; and (3) v_1 and v_2 have opposite sign, respectively. For each case of v_1 and v_2, we consider three cases of w_1 and w_2: (1) w_1 and w_2 are both positive; (2) w_1 and w_2 are both negative and (3) w_1 and w_2 have opposite sign. Therefore, there are 9 cases of the teacher parameters.

The procedure followed for the numerical analysis is given as:
Step 1: The teacher parameters are generated uniformly in the interval $[-2,\ 2]$. There are 9 cases. For each case, we generate 500 groups of teacher parameters.
Step 2: After each group of teacher parameters is generated, 20 groups of initial student parameters are generated uniformly in the interval $[-2,\ 2]$.
Step 3: For each group of teacher parameters and initial student parameters, we use the ALEs to accomplish the learning process. Some important variables, such as the generalization error, or the student parameters J_i and w_i, are traced and recorded.
Step 4: For each learning process, as the student parameters have been traced, the Fisher information matrix can be obtained. Then we record the inverse of the condition number of the Fisher information matrix.
Step 5: After the inverse of the condition value of the Fisher information value is recorded, a primary screening can be taken to judge whether the inverse of the condition number of

the Fisher information matrix of the learning processes has been close to 0.

Step 6: After this primary screening, we make a further analysis. If weight w_i was nearly equal to 0 in the process, then the process was affected by elimination singularity. If the two weights J_1 and J_2 nearly overlapped after training, the learning process was affected by overlap singularity. We count the numbers of the learning processes which were affected by elimination singularities and overlap singularities, respectively.

In this experiment, we totally accomplish the learning processes 90000 ($3\times3\times500\times20$) times. Given that the cases which are affected by the singularities in this subsection are exactly the same with those in Section 3.1.1, in order to keep the paper more concise, we do not show the learning trajectories belong to these cases in this subsection. Next, we count the number of learning processes which contain one of the cases above and focus on the ratio of learning processes influenced by different singularities. As the learning processes are both affected by elimination singularities in case 3 and case 4, we view case 3 and case 4 as one case in the counting process. The statistical results are shown in Table 1.

Number of total experiments	90000
Number of case 1 (Fast convergence)	61299
Number of case 2 (Overlap singularity)	6786
Number of case 3 and case 4 (Elimination singularity)	11288
Number of case 5 (Output weight 0)	10627

Table 1: Statistical results of two-hidden-unit RBF networks

From the 4 cases of observed behaviors and the statistical results shown in Table 1, we can obtain some conclusions as follows:

1) Many researchers have noticed the plateau phenomenon in the learning dynamics of feedforward neural networks (Amari et al., 2006; Saad and A.Solla, 1995; Biehl and Schwarze, 1995; Fukumizu and Amari, 2000). However, the reason why the plateau phenomenon occurs remains controversial. From the experimental results in Figure 3 and Figure 4, we found that the existence of singularities in the student parameter space results in the plateaus.

2) As shown in Table 1, nearly 68 percent of all the experiments did not get affected by the singularities and the learning dynamics converged to the global minimum fast. Almost 7.5 percent of experiments have been affected by overlap singularities and 12.5 percent of experiments have been affected by the elimination singularities. The data indicates that the singularities have a great impact on the learning processes of RBF networks. In light of the wide application of the RBF networks in practice, the influence of singularities ought to attract more attention of researchers. For the two-hidden-unit RBF networks, the initial center of the student model may be often relatively too far from the center of the teacher model, which causes the output weight of the student model to be nearly 0 after training. This case has been observed and mentioned in (Wei et al., 2007). From the results in Table 1, nearly 12 percent of experiments belong to this case.

3) From the statistical results shown in Table 1, it can be seen that the elimination singularities have much more influence in the learning dynamics than the overlap singularities. However, by now, few results in analyzing the elimination singularities have been obtained, which forms a sharp contrast to the overlap singularities. Due to the serious influence of

elimination singularities on the learning dynamics, it is worthy to take a theoretical analysis of elimination singularities.

3.2 RBF Networks in a General Case

In the previous subsection, we showed the results for the RBF networks with two hidden units. In this subsection, we generalize these results for the general RBF networks.

Without loss of generality, we introduce the student as a ten-hidden-unit model, namely:

$$f(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x},\boldsymbol{J}_i), \tag{21}$$

where $k = 10$.

We also assume that the teacher model is represented by a RBF network with 10 units, namely:

$$f(\boldsymbol{x},\boldsymbol{\theta_0}) = \sum_{i=1}^{s} v_i \phi(\boldsymbol{x},\boldsymbol{t}_i) + \varepsilon, \tag{22}$$

where $s = 10$.

We choose the spread constant $\sigma = 0.5$ and the input dimension $n = 2$.

When the number of hidden units in the student model is larger than that of the teacher, the redundant case exists. This implies that the teacher parameter might be on the singularity and the learning processes are basically affected by the singularity. In order to overcome this problem and avoid the overlap of the teacher units, we choose the minimal distance between the teacher units $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ to be bigger than $2\sigma^2$. The main reason behind this choice are based on the results obtained in (Wei and Amari, 2008). (Wei and Amari, 2008) obtained that the two teacher units are well separated when the distance between two hidden units is bigger than $2\sigma^2$.

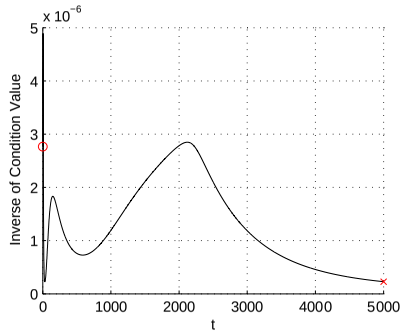
In our experiments, the teacher parameters $\boldsymbol{t}_i$, v_i, are uniformly generated in the interval $[-4,\ 4]$ and we generate 50 groups of teacher parameters. After each group of teacher parameters is generated, 20 groups of initial student parameters $\boldsymbol{J}_i^{(0)}$, $w_i^{(0)}$ are generated uniformly in the interval $[-4,\ 4]$. We use the ALEs to accomplish the learning processes. The experiment procedure is similar to that in Section 3.1.2.

By analyzing the simulation results, the cases where the learning processes present the undesirable behaviors are similar to those of RBF networks with two hidden units. To make the paper concise, the teacher parameters, the initial student parameters and the final student parameters of the following cases are listed in Appendix A. In the following figures, 'o' and '×' represent the initial state and final state, respectively.

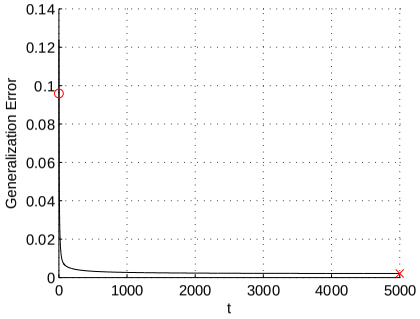
Case 1 (**Fast convergence**): The learning process quickly converges to the global minimum and the singularities do not affect the learning process.

We provide an example of this case. Figure 6 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively.

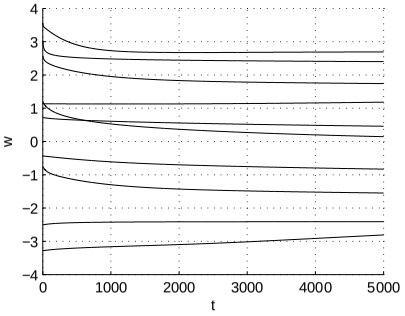
From Figure 6(a), the Fisher information matrix did not become singular during the learning process. Meanwhile, the generalization error dropped fast after the beginning of the learning process, and the singularity did not obviously affect the learning process. After the training process, the student model has converged to the global minimum.



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 6: Case 1 (Fast convergence) in RBF networks of general case

Case 2 (**Overlap singularity**): The learning process is affected by overlap singularity.

In this case, the learning processes are trapped in overlap singularities after training. An example belonging to this case is shown in Figure 7.

By analyzing the simulation results, it can be seen that, apart from the case where two student hidden units overlapped exactly after training, the phenomenon that two hidden units did not exactly overlap sometimes occurs. For the multidimensional parameters, we adopt the variable $h(i,j) = \frac{1}{2}\|\boldsymbol{J}_i - \boldsymbol{J}_j\|^2$ to indicate the distance between $\boldsymbol{J}_i$ and $\boldsymbol{J}_j$. When $\boldsymbol{J}_i$ and $\boldsymbol{J}_j$ nearly overlap, $h(i,j)$ nearly equals 0. Figure 7 shows the trajectories of the inverse of the condition number, generalization error, $h(4,8)$, weights w_i, respectively.

From Figure 7(a), the inverse of the condition value reduced to nearly 0 at the early stage of training process which implies that the Fisher information matrix nearly degenerated, and therefore this state remains till the end. Meanwhile, $h(4,8)$ (shown in Figure 7(c)) dropped to a very small value which implied $\boldsymbol{J}_4$ and $\boldsymbol{J}_8$ nearly overlapped, and the learning process is trapped in overlap singularities. From the final state of $\boldsymbol{J}$, it can be seen that $\boldsymbol{J}_4$ and $\boldsymbol{J}_8$ are close to each other, but do not exactly overlap. However, the gradient of the generalization error $L(\boldsymbol{\theta})$ respect to the final student parameters is nearly 0, which is too small to influence the learning process, and the student parameters will remain almost unchanged even though the learning process lasts longer. This is mainly because the error surface of RBF networks in a general case near overlap singularities is very flat. When the learning process arrives at the neighborhood of overlap singularities, although the student units have not overlapped completely, the student units will slightly change as the result of the relatively unchanged error in the remaining stage. The trajectories in Figure 7(a) and Figure 7(b) are similar to the corresponding trajectories in Figure 2(a) and Figure 2(b), respectively.

Case 3 (**Elimination singularity**): The learning process is affected by the elimination singularity and a plateau phenomenon can be observed.

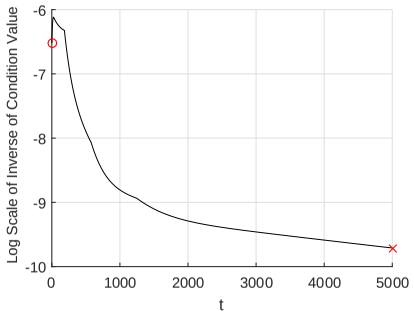
In this case, the learning process is significantly affected by the elimination singularity. This case is similar to cases 3 or 4 in Section 3.1.2. A plateau phenomenon can be observed during the learning process. We give an example of this case in Figure 8, which shows the trajectories of the inverse of the condition number, generalization error, and weights w_i.

As shown in Figure 8(a), the Fisher information matrix became nearly singular at an early stage of the learning process, and then became regular again. From the trajectories in Figure 8(b), it can be observed that w_5 (the wider line) skipped 0 when the Fisher information matrix became singular and then regular. This means that the learning process was affected by the elimination singularity. A plateau phenomenon can be observed in the trajectory of the generalization error as shown in Figure 8(b).

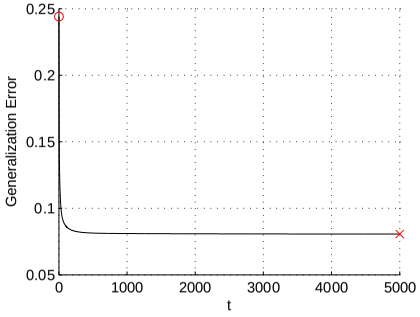
Case 4 (**Output weight 0**): After training, one of output weights w_i becomes nearly equal to 0.

This case is similar to case 5 in Section 3.1.2. When the initial student center is too far from the center of the teacher model, the output weight w_i of the student model usually becomes nearly 0 after the training process. An example is shown in Figure 9.

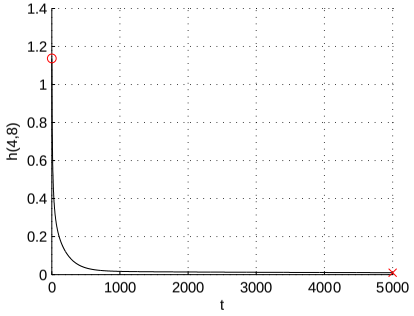
Figure 9 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively. From Figure 9(c), w_5 (the wider line) has become nearly 0 after training.



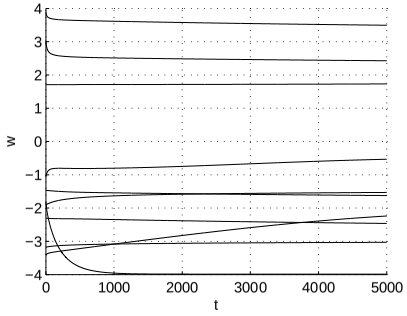
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error

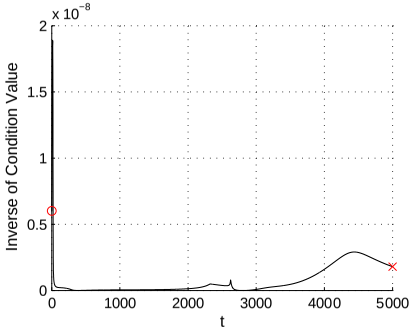


(c) Time evolution of $h(4, 8)$

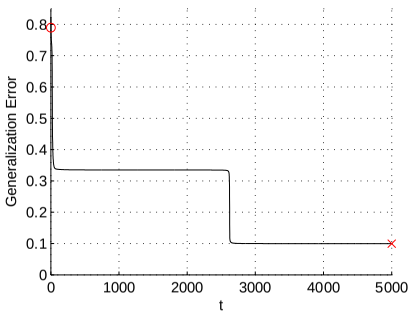


(d) Time evolution of w

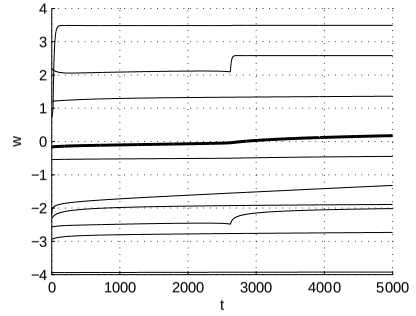
Figure 7: Case 2 (Overlap singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value

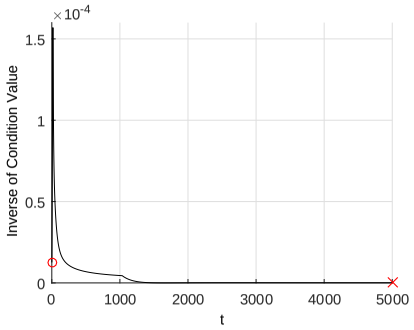


(b) Time evolution of generalization error

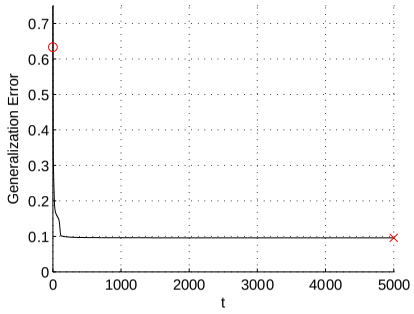


(c) Time evolution of w

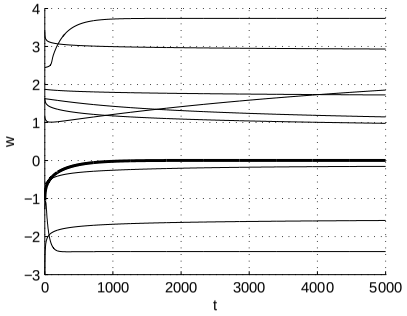
Figure 8: Case 3 (Elimination singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 9: Case 4 (Output weight 0) in RBF networks of general case

Case 5 (**Overlap and elimination singularity**): The learning process is affected by not only the overlap singularity but also the elimination singularity.

Different from the case of RBF with two hidden units, we find that sometimes the learning process in a more general case is simultaneously affected by the elimination singularity and the overlap singularities. We give an example of this case in Figure 10, which shows the trajectories of log scale of the inverse of the condition number, generalization error, $h(1,9)$, and weights w_i, respectively.

From Figure 10(a), the Fisher information matrix became singular at the early stage of the learning process, and as a result the learning process arrived in singularities. As shown in Figure 10(b), a plateau phenomenon can be obviously observed. From Figure 10(d), it can be seen that w_3 (the wider line) crossed 0 when the plateau phenomenon occurred, namely the learning process crossed the elimination singularity. From Figure 10(c), $h(1,9)$ became very small along the training process. After training, $\boldsymbol{J}_1 = [2.4494, \ -2.1973]^T$ and $\boldsymbol{J}_9 = [2.4020, \ -2.2118]^T$, i.e. $\boldsymbol{J}_1$ and $\boldsymbol{J}_9$ nearly equaled to each other, so the learning process was trapped in an overlap singularity.

Case 6 (**Elimination singularity and output weight 0**): The learning process is affected by elimination singularity and one of the output weights w_i becomes nearly 0 after the learning process.

In addition to the case above, we also find a case where the learning dynamics are affected by the elimination singularities during the learning process, one of the weights w_i becomes nearly 0 after training and the student parameters are trapped in an local minimum. We give an example that belongs to this case in Figure 11.

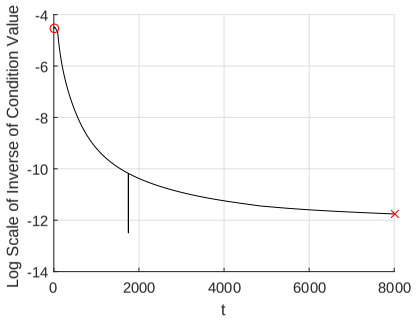
Figure 11 shows the trajectories of the inverse of the condition number, generalization error, and weights w_i, respectively. From Figure 11(b) and Figure 11(c), at the stage where w_3 crossed 0, a plateau phenomenon occurred and the learning process was affected by the elimination singularity. After training, $w_5 = -0.0004$, which is nearly equal to 0.

In comparison with the analysis results in Section 3.1.2, the RBF networks in a more general case have similar singular behaviors as those in RBF networks with two hidden units. The statistical results are summarized in Table 2.

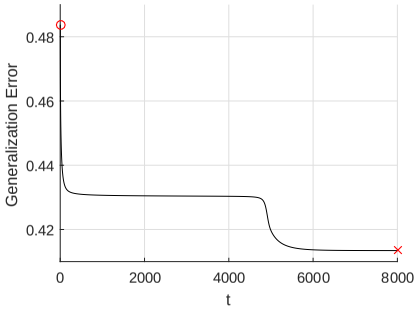
Number of total experiments	1000
Number of case 1 (Fast convergence)	675
Number of case 2 (Overlap singularity)	109
Number of case 3 (Elimination singularity)	56
Number of case 4 (Output weight 0)	123
Number of case 5 (Overlap and elimination singularity)	16
Number of case 6 (Elimination singularity and output weight 0)	21

Table 2: Statistical results of RBF networks in a general case

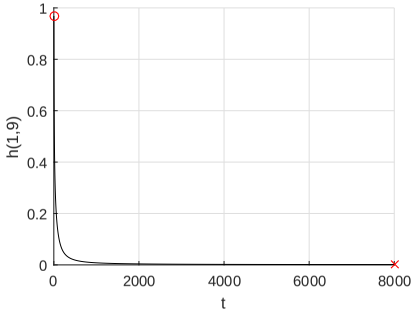
From the results shown in Table 2, 67.5 percent of experiments did not get affected by the singularities and the learning dynamics converged to the global minimum fast. On the other hand, 20.2 percent of the learning processes are affected by the singularities. This ratio is close to the one for RBF networks with two hidden units, which implies that the existence of singularities indeed significantly affects the learning process of RBF networks. 12.3 percent of the experiments belong to case 4, which implies that the case should attract



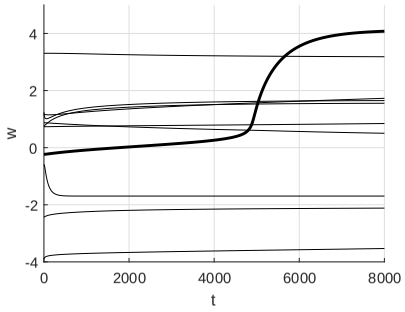
(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error

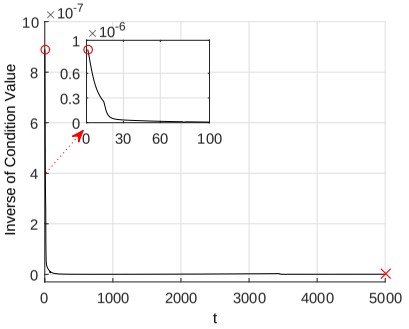


(c) Time evolution of $h(1, 9)$

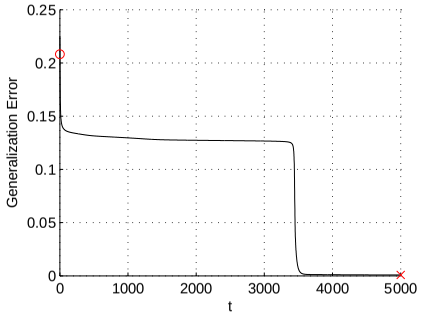


(d) Time evolution of w

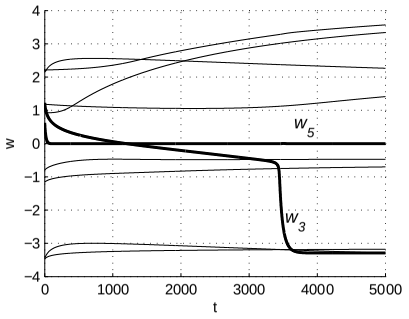
Figure 10: Case 5 (Overlap and elimination singularity) in RBF networks of general case



(a) Trajectory of inverse of condition value



(b) Time evolution of generalization error



(c) Time evolution of w

Figure 11: Case 6 (Elimination singularity and output weight 0) in RBF networks of general case

more attention. In case 5 and case 6, plateau phenomenons can be obviously observed where the learning dynamics are affected by the elimination singularities.

3.3 Extended Complex Scene Saliency Data set (ECSSD)

In the above experiments, we use artificial examples. We now perform an experiment by using a factual data set. Salient object detection plays a key role in many image analysis tasks that identifies important locations and structure in the visual field (Borji and Itti, 2013; Zhang et al., 2017). In recent years researchers utilize deep learning to improve the performance of saliency detection (Zhao et al., 2015b; Lee et al., 2016). As a benchmark data set in saliency detection community, extended complex scene saliency data set (ECSSD) has been widely used since its release in 2013 (Yan et al., 2013). In this experiment, we use the method proposed in (Zhang et al., 2014) to extract the features of the images in ECSSD data set as the input of the RBF networks. We get three conspicuity maps in both the rarity and the distinctiveness factors, and one conspicuity map in central bias factor. Thus the number of the nodes in the input layer is 7. The output of the training samples is '1' or '0', where '1' represents this part of the image is salient and '0' represents this part of the image is not salient.

As the distribution of input data is unknown in this experiment, we cannot obtain the analytical form of both ALEs of the training process and the Fisher information matrix. Thus we use batch mode learning to accomplish the experiment. By using a trial-and-error method, we choose the number of hidden unit in the student model to be $k = 90$ and the spread constant $\sigma = 0.5$, such that the student RBF network for the input $\boldsymbol{x}$ is given by:

$$f(\boldsymbol{x}, \boldsymbol{\theta}) = \sum_{i=1}^{90} w_i \phi(\boldsymbol{x}, \boldsymbol{J}_i). \tag{23}$$

We use 200 samples to train the RBF network. For the learning rate $\eta = 0.002$, the model is trained by the gradient algorithm for 15000 times and the sum squared training error $E = \frac{1}{2}\sum_{i=1}^{200}(y_i - \sum_{j=1}^{90} w_j \phi(\boldsymbol{x}_i, \boldsymbol{J}_j))^2$ is used to replace the generalization error. Then we clone it 200 times. Each clone is trained with different random initial weights. The initial student parameters $\boldsymbol{J}_i^{(0)}$ and $w_i^{(0)}$ are uniformly generated in the interval $[-2,\ 2]$.

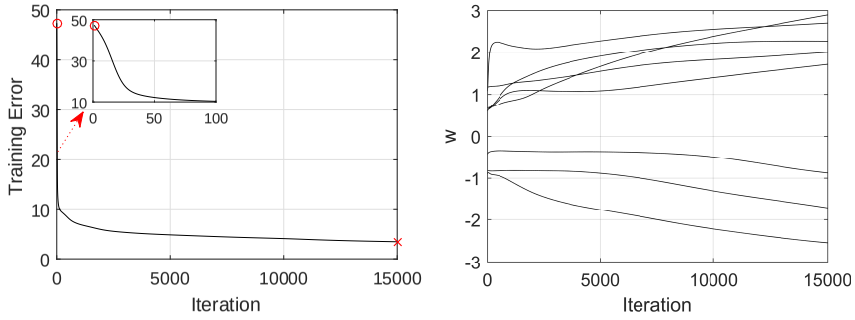
By analyzing the simulation results, the different types of learning processes are listed as follows. In the following figures, '$\circ$' and '$\times$' represent the initial state and final state, respectively.

Case 1 (**Fast convergence**): The learning process is not affected by singularities.

We give an example of this case in Figure 12, which shows the trajectories of the training error and part of output weights $\boldsymbol{w}$. From Figure 12(a), we can see that the training error comes down to a small number after the training starts and remains small till the learning stops. We do not observe that the singularities have affected the training process.

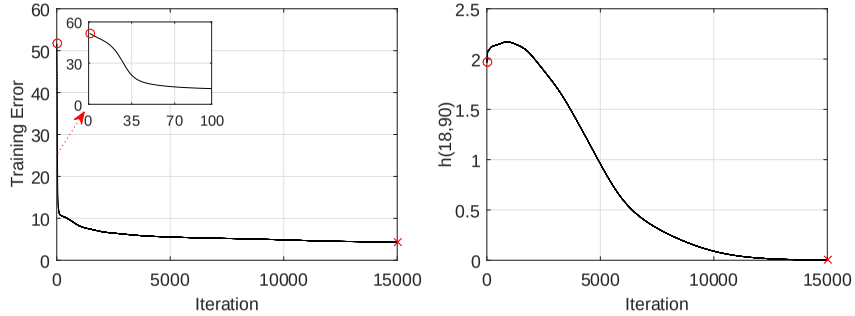
Case 2 (**Overlap singularity**): The learning process is affected by overlap singularity.

We give an example of this case in Figure 13, which shows the trajectories of the training error and $h(18, 90)$. From Figure 13(b), the Euclid distance between $\boldsymbol{J}_{18}$ and $\boldsymbol{J}_{90}$ became nearly 0, which means $\boldsymbol{J}_{18}$ and $\boldsymbol{J}_{90}$ nearly overlapped after learning.



(a) Trajectory of training error (b) Trajectory of w

Figure 12: Case 1 (Fast convergence) in approximating ECSSD data set



(a) Trajectory of training error (b) Trajectory of $h(18, 90)$

Figure 13: Case 2 (Overlap singularity) in approximating ECSSD data set

The initial state is:

$\boldsymbol{J}_{18}^{(0)} = [-0.4159,\ -1.0079,\ -0.3436,\ 0.1162,\ 0.6212,\ 0.3521,\ 0.9892]^T$,

$\boldsymbol{J}_{90}^{(0)} = [-0.1619,\ 0.0519,\ 0.1225,\ 0.6200,\ -0.8639,\ -0.0874,\ 0.8953]^T$.

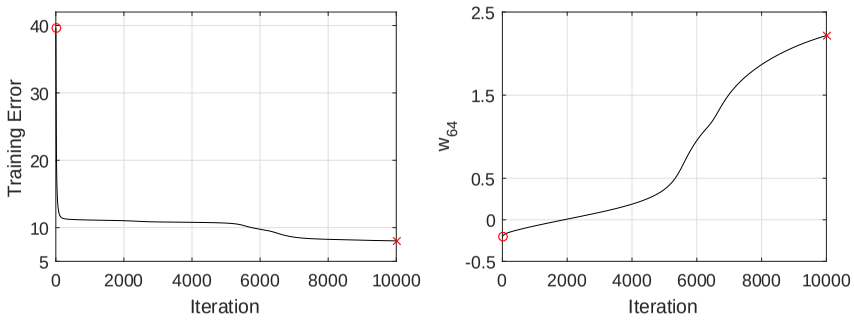
The final state is:

$\boldsymbol{J}_{18} = [-0.2480,\ 0.0515,\ -0.1948,\ 0.2474,\ 0.0130,\ 0.0531,\ 1.2120]^T$,

$\boldsymbol{J}_{90} = [-0.2353,\ 0.0871,\ -0.1995,\ 0.2585,\ -0.0267,\ 0.0538,\ 1.1964]^T$.

Case 3 (**Elimination singularity**): The learning process is affected by elimination singularity.

We give an example of this case. Figure 14 shows the trajectories of the training error and output weight w_{64}. We can see that during the stage where w_{64} crosses 0 (Figure 14(b)), a plateau phenomenon can be observed (Figure 14(a)). The learning process is, thus, significantly affected by the elimination singularity.



(a) Trajectory of training error (b) Trajectory of w_{64}

Figure 14: Case 3 (Elimination singularity) in approximating ECSSD data set

Case 4 (**Output weight 0**): After training, one of the output weights w_i nearly equals to 0.

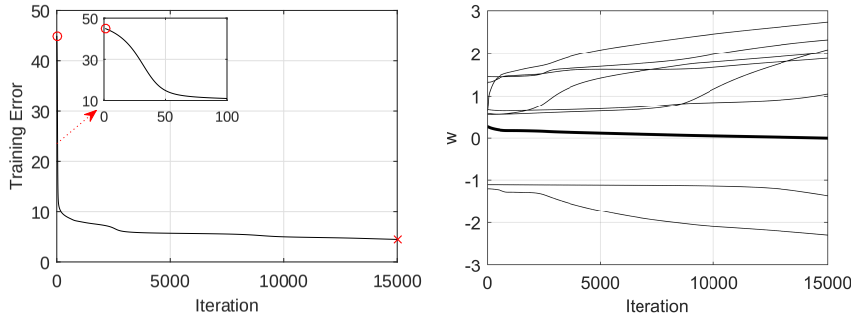
We give an example of this case. Figure 15 shows the trajectories of the training error and part of output weights $\boldsymbol{w}$. From the trajectory in Figure 15(b), it can be seen that w_{80} (the wider line) became nearly 0 after the training process.

Next, we count the learning processes which belong to each of the three cases and show them in Table 3.

Number of total experiments	200
Number of case 1 (Fast convergence)	153
Number of case 2 (Overlap singularity)	3
Number of case 3 (Elimination singularity)	42
Number of case 4 (Output weight 0)	2

Table 3: Statistical results of RBF networks in approximating ECSSD data set

It can be seen from the statistical results in Table 3 that as many as 22.5 percent of the experiments were seriously affected by the different types of singularity. There are only three experiments affected by the overlap singularity. On the other hand, we can see that 21 percent of the experiments were affected by elimination singularities. The results indicate that, in a high dimensional data scenario, the learning process is more likely affected by the elimination singularity. Therefore, it is worthy to pay more attention to investigating



(a) Trajectory of training error (b) Trajectory of w

Figure 15: Case 4 (Output weight 0) in approximating ECSSD data set

the elimination singularity. Moreover, different from our other earlier simulation results, only 1 percent of the experiments belong to case 4. The ratio is much less than those of the former experiments. The main reason is that a factual function can be represented by different suboptimal RBF networks which are equivalent to each other and the case where the initial center of the student model is far away from the center of teacher model becomes infrequent.

The statistical results confirm the previous results investigating the training difficulties in large networks from another view of point. (Dauphin et al., 2014) concluded that the local minima with high error were rare in high dimensions and the training difficulties were mainly caused by saddle points. From Table 3, we can see that the experiments affected by the overlap singularities (local minimum case) are much less than those in low dimensional networks. However, nearly all of the singular learning dynamics are affected by the elimination singularities (saddle point case). The results are in accordance with those obtained in (Dauphin et al., 2014).

4. Conclusion

Many previous works have demonstrated that the learning dynamics of feedforward neural networks are affected by the existence of singularities, but which type of singularity has more influence on the learning dynamics remains unclear. RBF networks are typical feedforward neural networks, and the learning dynamics near overlap singularities have been theoretically analyzed. Based on the obtained results, we have focused on the relationship between the existence of singularities and the learning dynamics of RBF networks in this paper. We have presented the analytical expression of the Fisher information matrix for RBF networks, as the singularities are the subspaces of the parameter space where the Fisher information matrix degenerates.

From the learning trajectories of the parameters in the generalization error surface, it can be clearly seen that the learning dynamics of RBF networks are affected by the singularities. Through a large number of numerical simulation experiments for RBF networks with two hidden units, we have identified 4 cases presenting strange learning behaviors. Nearly 7.5 percent and 12.5 percent of our experiments have shown significant effects of the overlap singularities and the elimination singularities, respectively. The points in the overlap singularity are local minima and the points in the elimination singularity are saddle points. The elimination singularities have a more significant impact to the learning processes than the overlap singularities. Our experimental results have also indicated that the plateau phenomena are mainly caused by the elimination singularities. Moreover, about 12 percent of our experiments have shown that one of the output weights of RBF networks could be close to zero after training and the student parameters are trapped into local minimum.

Through numerical simulation experiments for large scale RBF networks using a practice data set, we have found that the results are some different. Nearly all singular cases belong to the elimination singularity case and the overlap singularity case rarely occurred. This means that the large scale networks are more likely affected by the saddle points. The cases that converge to a local minimum with high error rarely appeared and the networks mainly converge to the global minimum or local minimum with good performance. The results are in accordance with the previous findings in large scale neural networks (Dauphin et al., 2014; Saxe et al., 2014; Choromanska et al., 2015).

In summary, we conclude that:

1) Overlap singularities lead to genuine local minima and elimination singularities lead to saddle points. The plateau phenomena are mainly caused by the elimination singularities.

2) The elimination singularities have a more significant impact to the learning processes than the overlap singularities. The overlap singularities mainly influence the learning dynamics of neural networks with low dimension. The large scale networks predominantly suffer from elimination singularities (saddle point case) and local minima with high error have rare influence.

Future research should pay more attention to the elimination singularities, and special treatments should be designed both for the traditional feedforward neural networks and deep neural networks to deal with the existence of singularities.

Acknowledgments

The authors would like to thank Professor S. Amari for his very constructive comments and suggestions. This work is partially supported by the Data Science and Artificial Intelligence Center (DSAIR) at the Nanyang Technological University. We would like to acknowledge support for this project from the National Science Foundation of China (NSFC) under Grant 61374006, 61773118 and 61703100, Natural Science Foundation of Jiangsu under Grant BK20170692, Jiangsu Planned Projects for Postdoctoral Research Funds under Grant 1601001A and Fundamental Research Funds for the Central Universities.

Appendix A. Proof of Theorem 1

By substituting $f(\boldsymbol{x},\boldsymbol{\theta}) = \sum_{i=1}^{k} w_i \phi(\boldsymbol{x},\boldsymbol{J}_i)$ into Eq.(4), we have

$$F(\boldsymbol{\theta}) = (F_{ij})_{(2k)\times(2k)}. \tag{A-1}$$

From Eq.(5), we have:

$$y - f_0(\boldsymbol{x}) = \varepsilon \sim \mathcal{N}(0,\sigma_0^2), \tag{A-2}$$

then

$$\frac{1}{\sqrt{2\pi}\sigma_0}\int_{-\infty}^{+\infty}\exp\left(-\frac{(y-f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right)\mathrm{d}y = \frac{1}{\sqrt{2\pi\sigma_0}}\int_{-\infty}^{+\infty}\exp(-\frac{\varepsilon^2}{2\sigma_0^2})\mathrm{d}\varepsilon = 1. \tag{A-3}$$

Thus,

$$\begin{aligned}\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle &= \left(\sqrt{2\pi}\right)^{-n}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty}\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right)\\ &\quad\times\frac{1}{\sqrt{2\pi}\sigma_0}\exp\left(-\frac{(y-f_0(\boldsymbol{x}))^2}{2\sigma_0^2}\right)\mathrm{d}y\mathrm{d}\boldsymbol{x}\\ &= \left(\sqrt{2\pi}\right)^{-n}\int_{-\infty}^{+\infty}\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{x}\right)\mathrm{d}\boldsymbol{x}.\end{aligned} \tag{A-4}$$

From results in Eq.(B.6)(Wei and Amari, 2008), we have:

$$\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle = C(\boldsymbol{J}_i,\boldsymbol{J}_j). \tag{A-5}$$

Then we calculate $\left\langle\phi(\boldsymbol{x},\boldsymbol{J}_j)\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\right\rangle$ and $\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_j)}{\partial\boldsymbol{J}_j^T}\right\rangle$:

$$\left\langle\phi(\boldsymbol{x},\boldsymbol{J}_j)\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\right\rangle = \frac{\partial}{\partial\boldsymbol{J}_i}\langle\phi(\boldsymbol{x},\boldsymbol{J}_i)\phi(\boldsymbol{x},\boldsymbol{J}_j)\rangle = C(\boldsymbol{J}_i,\boldsymbol{J}_j)B(\boldsymbol{J}_i,\boldsymbol{J}_j). \tag{A-6}$$

$$\begin{aligned}\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_j)}{\partial\boldsymbol{J}_j^T}\right\rangle &= \frac{\partial}{\partial\boldsymbol{J}_j^T}\left\langle\frac{\partial\phi(\boldsymbol{x},\boldsymbol{J}_i)}{\partial\boldsymbol{J}_i}\phi(\boldsymbol{x},\boldsymbol{J}_j)\right\rangle\\ &= \frac{C(\boldsymbol{J}_i,\boldsymbol{J}_j)}{\sigma^2(\sigma^2+2)}\left(\mathbf{I}_n + (\boldsymbol{J}_j - (\sigma^2+1)\boldsymbol{J}_i)B^T(\boldsymbol{J}_i,\boldsymbol{J}_j)\right).\end{aligned} \tag{A-7}$$

From Eq.(A-1) and Eqs.(A-5)-(A-7), we can obtain the results in Theorem 1. ■

Appendix B. Teacher and Student Parameters of RBF Networks of General Case

Case 1 (Fast convergence):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} 2.4872 & -1.8617 & -3.8418 & 3.2283 & -3.4445 & 2.4688 & -1.7230 & -3.8000 & 2.5460 & 3.6434 \\ -0.8149 & 3.4688 & 1.3114 & 3.8362 & -1.5388 & -2.2370 & -3.9184 & 0.1137 & 0.6776 & 0.0255 \end{bmatrix},$$

$\boldsymbol{v} = [2.5148, -2.6292, -2.9227, 0.3614, -0.1168, -1.1332, -2.4091, 1.4182, 0.6103, 1.9928]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 0.9014 & 2.1329 & -1.6206 & -2.9885 & 3.7966 & -2.5606 & 2.6824 & -0.3329 & 2.2608 & -3.8733 \\ 2.0291 & -3.2296 & -3.3590 & 1.2210 & -2.8917 & 0.7866 & 2.7598 & 2.7933 & 0.0625 & 1.7027 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [3.0312, -3.2829, -2.5060, -0.7531, 1.1339, 2.5729, 0.7212, 1.2064, 3.5641, -0.4348]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.0234 & 2.8324 & -1.7438 & -3.6897 & 3.5900 & -3.7365 & 2.9589 & -0.2505 & 2.5353 & -3.7203 \\ 4.3690 & -2.6481 & -3.9085 & 0.9678 & -2.8629 & 0.2693 & 2.8289 & 3.8682 & -0.7019 & 1.4435 \end{bmatrix},$$

$\boldsymbol{w} = [2.4010, -2.8088, -2.4106, -1.5482, 1.1832, 1.7415, 0.4593, 0.1452, 2.6930, -0.8301]$.

Case 2 (**Overlap singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} 2.6147 & -1.9288 & -2.0314 & -1.6500 & 3.6916 & 0.0861 & -3.9215 & 1.8375 & 0.4179 & -2.8322 \\ 2.3125 & -1.8576 & 0.4307 & -3.5281 & -3.3686 & 2.5186 & 3.0892 & 0.5955 & -1.7836 & -1.3025 \end{bmatrix},$$

$\boldsymbol{v} = [3.6158, -0.3521, 1.9219, 1.9590, -3.3921, -3.9902, 2.8331, -0.5973, 1.8690, -1.9287]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} -2.5084 & 3.9684 & 0.3200 & 3.9631 & -0.8001 & -1.1329 & -0.4638 & 2.5393 & -2.0853 & -2.8973 \\ 3.3105 & 3.4805 & -3.9175 & -0.2992 & 2.0027 & -3.3396 & 2.6410 & -0.7964 & -1.1499 & -1.0947 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [-1.4711, 1.7100, -3.1748, -2.3028, 3.0340, -1.9053, -1.8244, 3.8838, -1.0591, -3.4191]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -2.2057 & 3.7966 & 0.4940 & 3.9223 & -2.3099 & -1.3950 & 0.0860 & 4.0588 & -2.2064 & -3.0322 \\ 3.3668 & 3.3314 & -4.9285 & -0.7460 & 3.4631 & -4.7794 & 2.5181 & -0.7819 & -1.7263 & -1.4229 \end{bmatrix},$$

$\boldsymbol{w} = [-1.6246, 1.7315, -3.0298, -2.4627, 2.4273, -1.5311, -3.9856, 3.4951, -0.5322, -2.2387]$.

Case 3 (**Elimination singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -3.8655 & -1.9977 & -0.9693 & -0.6684 & 0.8680 & 1.0865 & 1.1128 & 2.2843 & 3.0604 & 3.7053 \\ -0.2720 & -3.6433 & 0.3606 & -1.3656 & -1.6714 & 3.0623 & -0.1191 & 1.5496 & 0.5356 & -0.5962 \end{bmatrix},$$

$\boldsymbol{v} = [-0.3246, 1.3070, 2.5382, 1.4447, 1.8875, -1.5562, 3.3401, 2.5875, -1.1534, 0.8754]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 2.5721 & -2.0317 & -3.0520 & 2.2606 & -3.7871 & -3.4318 & -3.4161 & -3.3289 & -2.9387 & 2.9206 \\ -0.9149 & 2.3647 & 1.9332 & 0.3286 & 2.6422 & -2.5984 & 2.3670 & -3.9398 & -2.0198 & -0.9915 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [0.7242, 2.1926, -2.5681, -2.1687, -0.5425, 1.2113, -0.1622, -3.9397, -2.9404, -2.3096]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -3.9125 & -3.6358 & -3.5306 & -3.4661 & -3.4104 & -3.3225 & -0.9603 & 1.0969 & 3.6156 & 4.4242 \\ 2.7119 & -2.5288 & 2.7775 & -4.0849 & 2.3727 & -2.4674 & 0.3054 & -0.1894 & 0.3816 & -1.9401 \end{bmatrix},$$

$\boldsymbol{w} = [-0.4462, -2.7326, -2.0156, -3.9234, 0.1763, 1.3621, 2.5836, 3.4922, -1.3202, -1.8928]$.

Case 4 (**Output weight 0**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -2.9683 & -3.2510 & 1.8798 & 2.8565 & -0.9188 & -2.3089 & 1.2504 & -2.2858 & -2.3246 & 3.5718 \\ -1.2588 & 0.5826 & 0.5730 & 2.3925 & -2.0164 & -0.0036 & -1.2349 & -2.2290 & 3.8091 & 3.6664 \end{bmatrix},$$

$\boldsymbol{v} = [-3.8032, 3.7781, 0.2144, 0.4885, -3.8106, 3.0554, -2.3919, 2.8606, -2.1003, -3.9480]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} -1.6664 & 1.7394 & -1.7774 & -1.6198 & 0.5730 & 3.1482 & 0.9029 & 0.6478 & 1.7433 & -3.7033 \\ 3.3900 & 0.1411 & -2.1530 & 2.1560 & 1.9557 & -2.7220 & -1.9711 & 0.8890 & -2.1818 & 1.4784 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [1.6267, -1.2823, 1.1560, -0.6281, -1.0531, 1.8734, 3.4009, -2.9951, 1.6401, 2.4469]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} -1.9184 & 1.2261 & -2.6233 & -1.8287 & 0.4141 & 3.7659 & 1.4521 & 2.7995 & 2.8480 & -2.4240 \\ 4.0887 & -1.2480 & -2.6963 & 2.8370 & 2.8316 & -3.0412 & -4.6956 & 3.7300 & -3.3875 & 0.1493 \end{bmatrix},$$

$\boldsymbol{w} = [1.1474, -2.3942, 1.8534, -0.1541, 0.0024, 1.7245, 2.9327, -1.5772, 0.9781, 3.7390]$.

Case 5 (**Overlap and elimination singularity**):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -0.0872 & 2.5761 & -2.5050 & -2.7155 & 2.3981 & 3.6351 & -2.5770 & 1.7904 & -0.9167 & -0.1812 \\ 2.5066 & 0.0111 & -2.3643 & 0.6209 & -2.1824 & -2.0130 & 2.3249 & 1.0783 & -2.6681 & -0.9122 \end{bmatrix},$$

$\boldsymbol{v} = [-1.5554,\ -0.9298,\ -0.2798,\ 2.5378,\ 3.0691,\ 3.3465,\ -0.7750,\ -1.5487,\ 3.6799,\ -3.2292]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 2.4527 & 3.3154 & 0.3844 & -2.5047 & 2.8795 & 3.2582 & -2.7824 & 1.0668 & 1.1586 & -3.0520 \\ -2.2713 & -0.8962 & -3.8169 & 1.9851 & -3.4619 & -1.8005 & 3.6875 & 0.7938 & -1.7594 & -2.5302 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [0.7493,\ 1.1685,\ -0.2370,\ -3.8941,\ 3.3049,\ -2.4357,\ 0.7329,\ -0.5840,\ 1.2373,\ 0.8759]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.4494 & 3.5238 & -0.9825 & -3.1353 & 3.5427 & 4.7567 & -2.8540 & 1.8474 & 2.4020 & -3.2640 \\ -2.1973 & -1.9146 & -2.8404 & 3.1545 & -3.4283 & -1.8511 & 3.5258 & 0.9770 & -2.2118 & -2.9316 \end{bmatrix},$$

$\boldsymbol{w} = [1.5556,\ 1.7289,\ 4.0780,\ -3.5320,\ 3.1846,\ -2.1169,\ 0.8460,\ -1.6949,\ 1.6490,\ 0.5044]$.

Case 6 (Elimination singularity and output weight 0):
The teacher parameters are:

$$\boldsymbol{t} = \begin{bmatrix} -1.0613 & -3.8576 & 3.4049 & 0.5341 & -2.6745 & -2.2680 & 3.4429 & -2.8965 & -3.3202 & -0.4482 \\ -1.5431 & 3.7297 & 3.7183 & -2.7340 & -3.0857 & 2.1992 & 1.9510 & -1.8745 & 0.8943 & 3.8086 \end{bmatrix},$$

$\boldsymbol{v} = [-3.2804,\ 3.1900,\ 2.8246,\ -0.3471,\ 3.8166,\ -1.0072,\ -2.4852,\ 3.4311,\ 3.9074,\ 3.0455]$.

The initial student parameters are:

$$\boldsymbol{J}^{(0)} = \begin{bmatrix} 1.8378 & -2.5054 & 0.2820 & -2.4732 & 0.2117 & -1.6407 & -3.1161 & 1.1045 & 1.9858 & -2.9216 \\ 0.7437 & 2.2043 & -2.1807 & 2.6032 & 1.0711 & 2.6934 & -0.3361 & 3.6327 & 1.7780 & -3.6724 \end{bmatrix},$$

$\boldsymbol{w}^{(0)} = [2.2498,\ -3.4754,\ 1.2272,\ -1.1583,\ 0.6268,\ -0.7964,\ 0.9334,\ 1.1823,\ -3.4548,\ 2.2123]$.

The final student parameters are:

$$\boldsymbol{J} = \begin{bmatrix} 2.8022 & -3.7679 & -1.0605 & -2.4646 & 0.2115 & -2.1265 & -3.2653 & -0.3505 & 2.9145 & -2.8238 \\ 1.6773 & 3.3496 & -1.5449 & 2.4451 & 1.1889 & 2.0453 & 0.8777 & 3.5703 & 1.7208 & -1.9812 \end{bmatrix},$$

$\boldsymbol{w} = [2.2651,\ -3.1769,\ -3.2902,\ -0.7017,\ -0.0004,\ -0.4704,\ 3.3391,\ 1.4123,\ -3.2949,\ 3.5688]$.

References

S. Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

S. Amari and H. Nagaoka. *Methods of Information Geometry*. AMS and Oxford University, New York, USA, 2000.

S. Amari and T. Ozeki. Differential and algebraic geometry of multilayer perceptrons. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer System*, E84(A(1)):31–38, 2001.

S. Amari, H. Park, and K. Fukumizu. Adaptive method of realizing natural gradient learning for multilayer perceptrons. *Neural Computation*, 12(6):1399–1409, 2000.

S. Amari, H. Park, and T. Ozeki. Singularities affect dynamics of learning in neuromanifolds. *Neural Computation*, 18(5):1007–1065, 2006.

S. Amari, T. Ozeki, F. Cousseau, and H. Wei. Dynamics of learning in hierarchical models – singularity and milnor attractor. *Second International Conference on Cognitive Neurodynamics*, pages 3–9, 2009.

M. Aoyagi. Stochastic complexity and generalization error of a restricted Boltzmann machine in Bayesian estimation. *Journal of Machine Learning Research*, 11:1243–1272, 2010.

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.

M. Biehl and H. Schwarze. Learning by on-line gradient descent. *Journal of Physics A: Mathematical and General*, 28(3):643–656, 1995.

A. Borji and L. Itti. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):185–207, 2013.

A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surface of multilayer networks. *18th International Conference on Artificial Intelligence and Statistics(AISTATS 2015)*, pages 192–204, 2015.

F. Cousseau, T. Ozeki, and S. Amari. Dynamics of learning in multilayer perceptrons near singularities. *IEEE Transactions on Neural Networks*, 19(8):1313–1328, 2008.

Y. N. Dauphin, R. Pascanu, C. Gulcehre, K. Cho, S. Ganguli, and Y. Bengio. Identifying and attacking the saddle point problem in high-dimensional non-convex optimization. *Advances in Neural Information Processing Systems (NIPS)*, pages 2933–2941, 2014.

D. Erhan, P. A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent. The difficulty of training deep architectures and the effect of unsupervised pre-training. *12th International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 153–160, 2009.

K. Fukumizu. A regularity condition of information matrix of a multilayer perceptron network. *Neural Networks*, 9(5):871–879, 1996.

K. Fukumizu and S. Amari. Local minima and plateaus in hierarchical structure of multilayer perceptrons. *Neural Networks*, 13(3):317–327, 2000.

I. Goodfellow, O. Vinyals, and A. M. Saxe. Qualitatively characterizing neural network optimization problems. *URL: http://arxiv. org/abs/1412.6544*, 2014.

I. Goodfellow, Y. Bengio, and A. Courville. Deep learning. *MIT Press URL: http://www.deeplearningbook.org*, 2016.

C. Gulcehre, J. Sotelo, M. Moczulski, and Y. Bengio. A robust adaptive stochastic gradient method for deep learning. *International Joint Conference on Neural Networks (IJCNN2017)*, pages 125–132, 2017.

W. Guo, H. Wei, J. Zhao, and K. Zhang. Averaged learning equations of error-function-based multilayer perceptrons. *Neural Computing & Applications*, 25(3-4):825–832, 2014.

W. Guo, H. Wei, J. Zhao, and K. Zhang. Theoretical and numerical analysis of learning dynamics near singularity in multilayer perceptrons. *Neurocomputing*, 151:390–400, 2015.

T. Heskes. On "natural" learning and pruning in multilayered perceptrons. *Neural Computation*, 12(4):881–901, 2000.

G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

G. Lee, Y. Tai, and J. Kim. Deep saliency with encoded low level distance map and high level features. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 660–668, 2016.

Z. C. Lipton. Stuck in a what? adventures in weight space. *URL: https://arxiv.org/abs/1602.07320*, 2016.

T. Mononen. A case study of the widely applicable Bayesian information criterion and its optimality. *Statistics and Computing*, 25(5):929–940, 2015.

T. Nitta. Local minima in hierarchical structures of complex-valued neural networks. *Neural Networks*, 43:1–7, 2013.

T. Nitta. Learning dynamics of a single polar variable complex-valued neuron. *Neural Computation*, 27(5):1120–1141, 2015.

H. Park and T. Ozeki. Singularity and slow convergence of the EM algorithm for Gaussian mixtures. *Neural Processing Letters*, 29(1):45–59, 2009.

H. Park, S. Amari, and K. Fukumizu. Adaptive natural gradient learning algorithms for various stochastic models. *Neural Networks*, 13(7):755–764, 2000.

H. Park, M. Inoue, and M. Okada. Online learning dynamics of multilayer perceptrons with unidentifiable parameters. *Journal of Physics A: Mathematical and General*, 36 (47):11753–11764, 2003.

R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. *URL: http://arxiv.org/abs/1301.3584v7*, 2014.

M. Rattray, D. Saad, and S. Amari. Natural gradient descent for on-line learning. *Physical Review Letters*, 81(24):5461–5464, 1998.

D. Saad and A.Solla. Exact solution for online learning in multilayer neural networks. *Physical Review Letters*, 74(21):4337–4340, 1995.

A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *URL: http://arXiv preprint arXiv:1312.6120*, 2014.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117, 2015.

H. van Hasselt, A. Guez, M. Hessel, and D. Silver. Learning functions across many orders of magnitudes. *URL: http://arXiv preprint arXiv:1602.07714*, 2016.

S. Watanabe. Algebraic analysis for non-identifiable learning machines. *Neural Computation*, 13(4):899–933, 2001a.

S. Watanabe. Algebraic geometrical methods for hierarchical learning machines. *Neural Works*, 14(8):1049–1060, 2001b.

S. Watanabe. Almost all learning machines are singular. *IEEE Symposium on Foundations of Computational Intelligence*, pages 383–388, 2007.

S. Watanabe. Asymptotic equivalence of Bayes cross validation and widely applicable information criterion in singular learning theory. *Journal of Machine Learning Research*, 11:3571–3594, 2010.

S. Watanabe. A widely applicable Bayesian information criterion. *Journal of Machine Learning Research*, 14:867–897, 2013.

H. Wei and S. Amari. Dynamics of learning near singularities in radial basis function networks. *Neural Networks*, 21(7):989–1005, 2008.

H. Wei, Q. Li, and W. Song. Gradient learning dynamics of radial basis function networks. *Control Theory & Applications*, 24(3):356–360, 2007.

H. Wei, J. Zhang, F. Cousseau, T. Ozeki, and S. Amari. Dynamics of learning near singularities in layered networks. *Neural Computation*, 20(34):813–843, 2008.

Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1155–1162, 2013.

J. Zhang, J. Ding, and J. Yang. Exploiting global rarity, local contrast and central bias for salient region learning. *Neurocomputing*, 144:569–580, 2014.

J. Zhang, K. A. Ehinger, H. Wei, K. Zhang, and J. Yang. A novel graph-based optimization framework for salient object detection. *Pattern Recognition*, 64:39–50, 2017.

J. Zhao, H. Wei, C. Zhang, W. Li, W. Guo, and K. Zhang. Natural gradient learning algorithms for RBF networks. *Neural Computation*, 27(2):481–505, 2015a.

R. Zhao, W. Ouyang, H. Li, and X. Wang. Daliency detection by multi-context deep learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1265–1274, 2015b.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 05/16; Revised 11/17; Published 08/18

A Two-Stage Penalized Least Squares Method for Constructing Large Systems of Structural Equations

Chen Chen* chen1167@stat.purdue.edu
Min Ren* ren80@stat.purdue.edu
Min Zhang minzhang@stat.purdue.edu
Dabao Zhang zhangdb@stat.purdue.edu
Department of Statistics
Purdue University
West Lafayette, IN 47907, USA

Editor: Xiaotong Shen

Abstract

We propose a two-stage penalized least squares method to build large systems of structural equations based on the instrumental variables view of the classical two-stage least squares method. We show that, with large numbers of endogenous and exogenous variables, the system can be constructed via consistent estimation of a set of conditional expectations at the first stage, and consistent selection of regulatory effects at the second stage. While the consistent estimation at the first stage can be obtained via the ridge regression, the adaptive lasso is employed at the second stage to achieve the consistent selection. This method is computationally fast and allows for parallel implementation. We demonstrate its effectiveness via simulation studies and real data analysis.

Keywords: graphical model, high-dimensional data, reciprocal graphical model, simultaneous equation model, structural equation model

1. Introduction

We consider a linear system with p endogenous and q exogenous variables. With a sample of n observations from this system, we denote the observed values of endogenous and exogenous variables by $\mathbf{Y}_{n\times p} = (\mathbf{Y}_1, \cdots, \mathbf{Y}_p)$ and $\mathbf{X}_{n\times q} = (\mathbf{X}_1, \cdots, \mathbf{X}_q)$, respectively. The interactions among endogenous variables and the direct causal effects by exogenous variables can be described by a system of structural equations,

$$\mathbf{Y} = \mathbf{Y}\boldsymbol{\Gamma} + \mathbf{X}\boldsymbol{\Psi} + \boldsymbol{\epsilon}, \tag{1}$$

where the $p \times p$ matrix $\boldsymbol{\Gamma}$ has zero diagonal elements and contains regulatory effects, the $q \times p$ matrix $\boldsymbol{\Psi}$ contains causal effects, and $\boldsymbol{\epsilon}$ is an $n \times p$ matrix of error terms. We assume that $\mathbf{X}$ and $\boldsymbol{\epsilon}$ are independent of each other, and each component of $\boldsymbol{\epsilon}$ is independently distributed as normal with zero mean while rows of $\boldsymbol{\epsilon}$ are identically distributed.

With gene expression levels and genotypic values as endogenous and exogenous variables, respectively, the model (1) has been used to represent a gene regulatory network with each equation modeling the regulatory genetic effects as well as the causal genomic effects from cis-eQTL (i.e., expression quantitative trait loci located within the regions of their target genes) on a given gene, see Xiong *et al.* (2004), and Liu *et al.* (2008), among others. Genetical genomics experiments, which collect genome-wide gene expressions and genotypic values, have been widely undertaken to construct gene regulatory networks (Jansen and Nap, 2001; Schadt *et al.*, 2003). However, fitting a system of structural equations in (1) to genetical genomics data for the purpose of revealing a whole-genome gene regulatory network is still hindered by lack of an effective statistical method which addresses issues brought by large numbers of endogenous and exogenous variables.

Several efforts have been made to construct the system (1) with genetical genomics data. Xiong *et al.* (2004) proposed to use a genetic algorithm to search for genetic networks which minimize the Akaike Information Criterion (AIC; Akaike, 1974), and Liu *et al.* (2008) instead proposed to minimize the Bayesian Information Criterion (BIC; Schwartz, 1978) and its modification (Broman and Speed, 2002) for the optimal genetic networks. Both AIC and BIC are applicable to inferring networks for only a small number of endogenous variables. For a large system with many endogenous and exogenous variables, Cai *et al.* (2013) proposed to maximize a penalized likelihood to construct a sparse system. However, it is computationally formidable to fit a large system based on the likelihood function of the complete model. Logsdon and Mezey (2010) instead proposed to apply the adaptive lasso (Zou, 2006) to fitting each structural equation separately, and then recover the network relying on additional assumption on unique exogenous variables. However, Cai *et al.* (2013) demonstrated its inferior performance via simulation studies, which is consistent with our conclusion.

Instead of the full information model specified in (1), we seek to establish the large system via constructing a large number of limited information models, each for one endogenous variable (Schmidt, 1976). For example, when the k-th endogenous variable is concerned, we focus on the k-th structural equation in (1) which models the regulatory effects of other endogenous variables and direct causal effects of exogenous variables, and ignore the system structures contained in other structural equations, leading to the following limited-information model,

$$\begin{cases} \mathbf{Y}_k = \mathbf{Y}_{-k}\boldsymbol{\gamma}_k + \mathbf{X}\boldsymbol{\psi}_k + \boldsymbol{\epsilon}_k, \\ \mathbf{Y}_{-k} = \mathbf{X}\boldsymbol{\pi}_{-k} + \boldsymbol{\xi}_{-k}. \end{cases} \tag{2}$$

Here $\mathbf{Y}_{-k}$ refers to $\mathbf{Y}$ excluding the k-th column, $\boldsymbol{\gamma}_k$ refers to the k-th column of $\boldsymbol{\Gamma}$ excluding the diagonal zero, and $\boldsymbol{\psi}_k$ and $\boldsymbol{\epsilon}_k$ refer to the k-th columns of $\boldsymbol{\Psi}$ and $\boldsymbol{\epsilon}$ respectively. The second part of the model (2) is from the following reduced model by excluding the k-th reduced-form equation, with $\boldsymbol{\pi} = \boldsymbol{\Psi}(\mathbf{I} - \boldsymbol{\Gamma})^{-1}$ and $\boldsymbol{\xi} = \boldsymbol{\epsilon}(\mathbf{I} - \boldsymbol{\Gamma})^{-1}$,

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\pi} + \boldsymbol{\xi}. \tag{3}$$

In a classical low-dimensional setting, applying the ordinary least squares method to the first equation in (2) leads to underestimated $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_k$ due to correlated $\mathbf{Y}_{-k}$ and $\boldsymbol{\epsilon}_k$. Instead, the reduced-form equations in (2) are fitted to obtain least squares estimator $\hat{\boldsymbol{\pi}}_{-k}$ of $\boldsymbol{\pi}_{-k}$, and least squares estimators of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_k$ are further obtained by regressing $\mathbf{Y}_k$ against $\hat{\mathbf{Y}}_{-k} = \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}$ and $\mathbf{X}$. This procedure is widely known as the two-stage least squares

*. The first two authors contribute equally.

(2SLS) method which can produce consistent estimates of the parameters when the system is identifiable. The 2SLS estimator was originally proposed by Theil (1953a,b, 1961) and, independently, Basmann (1957), and can be restated as the instrumental variables estimator (Reiersøl, 1941, 1945).

As in a typical genetical genomics experiment, we are interested in constructing a large system with the number of endogenous variables p possibly larger than the sample size n. Such a high-dimensional and small sample size data set makes it infeasible to directly apply the 2SLS method. Indeed, $p \geq n$ may result in perfect fits of reduced-form equations at the first stage, which implies that we regress against the observed values of endogenous variables at the second stage and therefore obtain ordinary least squares estimates of the parameters. It is well known that such ordinary least squares estimates are inconsistent. Furthermore, constructing a large system demands, at the second stage, selecting regulatory endogenous variables among massive candidates, i.e., variable selection in fitting high-dimensional linear models.

In the setting of selecting instrumental variables (IVs) among a large number of candidates, L_1 regularized least squares estimators have been recently proposed to replace the ordinary least squares estimator at the first stage of 2SLS (Belloni *et al.*, 2012; Lin *et al.*, 2015; Zhu, 2015). Belloni *et al.* (2012) applied lasso-based methods to select IVs and obtain consistent estimations at the first stage when the first stage is approximately sparse. For sparse instrumental variables models, Zhu (2015) proposed to replace with lasso-based methods at both stages of 2SLS and Lin *et al.* (2015) considered the representative L_1 regularization methods and a class of concave regularization methods for both stages. All of these methods assume that each endogenous variable is only associated to a relatively small set of exogenous variables, i.e., each row of $\boldsymbol{\pi}$ in (3) only has a small set of nonzero components.

Here we consider to construct a general system of structural equations, which allows us to model nonrecursive or even cyclic relationships between endogenous variables. With the instrumental variables view of the two-stage approach, we observed that successful identification and consistent estimation of model parameters rely on consistent estimation of a set of conditional expectations which are optimal instruments. Therefore, establishing the system (1) in a high-dimensional setting is contingent on obtaining consistent estimation of these conditional expectations at the first stage, and effectively selecting and estimating of regulatory effects out of a large number of candidates at the second stage. Accordingly, we propose a two-stage penalized least squares (2SPLS) method to fit regularized linear models at each stage, with L_2 regularized linear models at the first stage and L_1 regularized linear models at the second stage.

The proposed method addresses three challenging issues in constructing a large system of structural equations, i.e., memory capacity, computational time, and statistical power. First, the limited information models are considered to develop the algorithm. In this way, we avoid working with full information models which may consist of many subnetworks and involve a massive number of endogenous variables. Second, allowing us to fit one linear model for each endogenous variable at each stage makes the algorithm computationally fast. It also makes it feasible to parallelize the large number of model fittings at each stage. Finally, the oracle properties of the resultant estimates show that the proposed method can achieve optimal power in identifying and estimating regulatory effects. Furthermore,

the efficient computation makes it feasible to use the bootstrap method to evaluate the significance of regulatory effects.

The rest of this paper is organized as follows. First, we state an identifiable model in Section 2. Section 3 revisits the instrumental variables view on the classical 2SLS method, which motivates our development of the 2SPLS method in Section 4. We show in Section 5 the theoretical properties of the estimates from 2SPLS, with the proof included in the Appendix. Simulation studies are carried out in Section 6 to evaluate the performance of 2SPLS. An application to a real data set to infer a yeast gene regulatory network is presented in Section 7. We conclude this paper with a discussion in Section 8.

2. The Identifiable Model

We follow the practice of constructing system (1) in analyzing genetical genomics data (Logsdon and Mezey, 2010; Cai *et al.*, 2013), and assume that each endogenous variable is affected by a unique set of exogenous variables, that is, the structural equation in (2) has known zero elements of $\boldsymbol{\psi}_k$. Explicitly, we use $\mathcal{S}_k$ to denote the set of row indices of known nonzero elements in $\boldsymbol{\psi}_k$. Then we have known sets $\mathcal{S}_k, k = 1, 2, \cdots, p$, which dissect the set $\{1, 2, \cdots, q\}$. We explicitly state this assumption in the below.

Assumption A. $\mathcal{S}_k \neq \emptyset$ for $k = 1, \cdots, p$, but $\mathcal{S}_j \cap \mathcal{S}_k = \emptyset$ as long as $j \neq k$.

The above assumption satisfies the rank condition (Schmidt, 1976), which is a sufficient condition for model identification. Since each $\boldsymbol{\psi}_k$ has a set of known zero components, from this point forward we ignore them and rewrite the structural equation in the model (2) as,

$$\mathbf{Y}_k = \mathbf{Y}_{-k}\boldsymbol{\gamma}_k + \mathbf{X}_{\mathcal{S}_k}\boldsymbol{\psi}_{\mathcal{S}_k} + \boldsymbol{\epsilon}_k, \qquad \boldsymbol{\epsilon}_k \sim N(\mathbf{0}, \sigma_k^2 \mathbf{I}_n), \tag{4}$$

where $\mathbf{X}_{\mathcal{S}_k}$ refers to $\mathbf{X}$ including only columns indicated by $\mathcal{S}_k$, and $\boldsymbol{\psi}_{\mathcal{S}_k}$ refers to $\boldsymbol{\psi}_k$ including only elements indicated by $\mathcal{S}_k$.

3. The Instrumental Variables View of the Two-Stage Least Squares Method

Because $\mathbf{Y}_{-k}$ and $\boldsymbol{\epsilon}_k$ are correlated, fitting merely the model (4) results in biased estimates of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$. However, the following two sets of variables are independent,

$$\begin{cases} \mathbf{Z}_{-k} = E[\mathbf{Y}_{-k}|\mathbf{X}] = \mathbf{X}\boldsymbol{\pi}_{-k}, \\ \boldsymbol{\varepsilon}_k = \boldsymbol{\epsilon}_k + \boldsymbol{\xi}_{-k}\boldsymbol{\gamma}_k. \end{cases}$$

Consequently, consistent estimates of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$ can be obtained by applying least squares method to the following model,

$$\mathbf{Y}_k = \mathbf{Z}_{-k}\boldsymbol{\gamma}_k + \mathbf{X}_{\mathcal{S}_k}\boldsymbol{\psi}_{\mathcal{S}_k} + \boldsymbol{\varepsilon}_k. \tag{5}$$

Observing $\mathbf{Y}_{-k}$ instead of $\mathbf{Z}_{-k} = E[\mathbf{Y}_{-k}|\mathbf{X}]$ naturally leads to application of the instrumental variables method (Reiersøl, 1941, 1945), that is, replacing $\mathbf{Z}_{-k} = \mathbf{X}\boldsymbol{\pi}_{-k}$ with its estimate $\hat{\mathbf{Z}}_{-k} = \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}$ in fitting the linear model (5). When a $\sqrt{n}$-consistent least squares

estimator of $\boldsymbol{\pi}_j$ is obtained by fitting each equation in (3) for $j = 1, \cdots, p$, the resultant estimators of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$ are exactly the 2SLS estimators by Theil (1953a,b, 1961) and Basmann (1957).

Suppose that the matrix $\mathbf{X}$ satisfies the assumption in the below. It is easy to prove that, in a low-dimensional setting, we can obtain consistent estimators for the model (5) with any consistent estimate of $\boldsymbol{\pi}_{-k}$.

Assumption B. $n^{-1}\mathbf{X}^T\mathbf{X} \to \mathbf{C}$, where $\mathbf{C}$ is a positive definite matrix.

Proposition 3.1 *Suppose Assumptions A and B are satisfied for the system (1) with fixed $p \ll n$ and $q \ll n$. When there exists a consistent estimator $\hat{\boldsymbol{\pi}}_{-k}$ of $\boldsymbol{\pi}_{-k}$, the ordinary least squares estimators of $(\boldsymbol{\gamma}_k, \boldsymbol{\psi}_{\mathcal{S}_k})$ obtained by regressing $\mathbf{Y}_k$ against $(\mathbf{X}\hat{\boldsymbol{\pi}}_{-k}, \mathbf{X}_{\mathcal{S}_k})$ are also consistent.*

The above instrumental variables view implies that the conditional expectation $\mathbf{Z}_{-k} = E[\mathbf{Y}_{-k}|\mathbf{X}]$ serves as the optimal instrument for $\mathbf{Y}_{-k}$. Although, in a low-dimensional setting, any consistent estimator $\hat{\boldsymbol{\pi}}_{-k}$ leads to the instrument $\hat{\mathbf{Z}}_{-k} = \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}$, an efficient estimate of $\boldsymbol{\pi}_{-k}$ should be used to produce efficient estimates of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$. In the following section, we build up on this view and construct the high-dimensional system (1) by first fitting high-dimensional linear models to consistently estimate the conditional expectations of endogenous variables given exogenous variables.

4. The Two-Stage Penalized Least Squares Method

To construct the limited-information model (2), we can obtain consistent estimates of the conditional expectations of endogenous variables given exogenous variables by fitting high-dimensional linear models, and then conduct a high-dimensional variable selection following our view on the model (5). Accordingly, we propose a two-stage penalized least squares (2SPLS) procedure to construct each model in (2) so as to establish the large system (1).

4.1 The Method

At the first stage, we use the ridge regression to fit each reduced-form equation in (3) to obtain consistent estimates of the conditional expectations of endogenous variables given exogenous variables, that is, for each $j = 1, 2, \cdots, p$, we obtain the ridge regression estimator of $\boldsymbol{\pi}_j$ by minimizing the following penalized sum of squares,

$$\|\mathbf{Y}_j - \mathbf{X}\boldsymbol{\pi}_j\|_2^2 + \tau_j\|\boldsymbol{\pi}_j\|_2^2, \tag{6}$$

where $\|\cdot\|_2$ is the L_2 norm, and $\tau_j > 0$ is a tuning parameter that controls the strength of the penalty. The solution to the minimization problem is $\hat{\boldsymbol{\pi}}_j = (\mathbf{X}^T\mathbf{X} + \tau_j\mathbf{I})^{-1}\mathbf{X}^T\mathbf{Y}_j$, which leads to a consistent estimate of $\mathbf{Z}_j$,

$$\hat{\mathbf{Z}}_j = \mathbf{P}_{\tau_j}\mathbf{Y}_j,$$

where $\mathbf{P}_{\tau_j} = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau_j\mathbf{I})^{-1}\mathbf{X}^T$. With a proper choice of τ_j, the ridge regression has a good estimation performance as shown in the next section.

At the second stage, we replace $\mathbf{Z}_{-k}$ with $\hat{\mathbf{Z}}_{-k}$ in model (5) to derive estimates of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$, specifically, we minimize the following penalized error squares to obtain estimates of $\boldsymbol{\gamma}_k$ and $\boldsymbol{\psi}_{\mathcal{S}_k}$,

$$\frac{1}{2}\|\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k - \mathbf{X}_{\mathcal{S}_k}\boldsymbol{\psi}_{\mathcal{S}_k}\|_2^2 + \lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|, \tag{7}$$

where $|\boldsymbol{\gamma}_k|$ denotes componentwise absolute value of $\boldsymbol{\gamma}_k$, $\boldsymbol{\omega}_k$ is a known weight vector, and $\lambda_k > 0$ is a tuning parameter.

Minimizing for $\boldsymbol{\psi}_{\mathcal{S}_k}$ in (7) leads to

$$\hat{\boldsymbol{\psi}}_{\mathcal{S}_k} = (\mathbf{X}_{\mathcal{S}_k}^T\mathbf{X}_{\mathcal{S}_k})^{-1}\mathbf{X}_{\mathcal{S}_k}^T(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k),$$

where $\mathbf{X}_{\mathcal{S}_k}$ is usually of low dimension, and the above least squares estimator of $\boldsymbol{\psi}_{\mathcal{S}_k}$ is easy to obtain.

Plugging $\hat{\boldsymbol{\psi}}_{\mathcal{S}_k}$ into (7), we can solve the following minimization problem to obtain an estimate of $\boldsymbol{\gamma}_k$,

$$\hat{\boldsymbol{\gamma}}_k = \arg\min_{\boldsymbol{\gamma}_k}\left\{\frac{1}{2}(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)^T\mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k) + \lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|\right\}, \tag{8}$$

where $\mathbf{H}_k = \mathbf{I} - \mathbf{X}_{\mathcal{S}_k}(\mathbf{X}_{\mathcal{S}_k}^T\mathbf{X}_{\mathcal{S}_k})^{-1}\mathbf{X}_{\mathcal{S}_k}^T$, this is equivalent to a variable selection problem in regressing $\mathbf{H}_k\mathbf{Y}_k$ against high-dimensional $\mathbf{H}_k\hat{\mathbf{Z}}_{-k}$. We will resort to adaptive lasso to select nonzero components of $\boldsymbol{\gamma}_k$ and estimate them. Specifically, picking up a $\delta > 0$ and obtaining $\tilde{\boldsymbol{\gamma}}_k$ as a $\sqrt{n}$-consistent estimate of $\boldsymbol{\gamma}_k$, we calculate the weight vector $\boldsymbol{\omega}_k$ with components inversely proportional to components of $|\tilde{\boldsymbol{\gamma}}_k|^\delta$. The above minimization problem (8) is a convex optimization problem which is computationally efficient.

4.2 Tuning Parameter Selection

In this method, we need to select tuning parameters at each stage. At the first stage, we propose to choose each τ_j in (6) by the method of generalized cross-validation (GCV; Golub *et al.*, 1979), that is,

$$\tau_j = \arg\min_{\tau>0} G_j(\tau) = \arg\min_{\tau>0}\frac{(\mathbf{Y}_j - \mathbf{P}_\tau\mathbf{Y}_j)^T(\mathbf{Y}_j - \mathbf{P}_\tau\mathbf{Y}_j)}{(n - \mathrm{tr}\{\mathbf{P}_\tau\})^2}.$$

It is a rotation-invariant version of ordinary cross-validation, and leads to an approximately optimal estimate of the conditional expectation $\mathbf{Z}_j$. At the second stage, the tuning parameter λ_k in (8) is obtained via K-fold cross validation.

5. Theoretical Properties

5.1 The Number of Endogenous Variables is Fixed

As an extension of the classical 2SLS method to high dimensions, the proposed 2SPLS method also has some good theoretical properties. In this section, we will show that the 2SPLS estimates enjoy the oracle properties. As the second-stage estimation relies on the

ridge estimates $\hat{\mathbf{Z}}_{-k}$ obtained from the first stage, we start with the theoretical properties of $\hat{\mathbf{Z}}_{-k}$.

As mentioned previously, each τ_j in (6) is obtained by GCV. Interestingly, as stated by Golub *et al.* (1979), such a τ_j is closely related to the one minimizing

$$T_j(\tau) = (\mathbf{Z}_j - \mathbf{P}_\tau \mathbf{Y}_j)^T (\mathbf{Z}_j - \mathbf{P}_\tau \mathbf{Y}_j).$$

We have the following result similar to Theorem 2 of Golub *et al.* (1979).

Theorem 5.1 *Suppose that all components of $\boldsymbol{\pi}_j$ are i.i.d. with mean zero and variance $\sigma^2_{\boldsymbol{\pi}}$, then*

$$\arg\min_{\tau>0} E\left[E[G_j(\tau)|\boldsymbol{\pi}_j]\right] = \arg\min_{\tau>0} E\left[E[T_j(\tau)|\boldsymbol{\pi}_j]\right] = \sigma^2_{\boldsymbol{\xi}_j}/\sigma^2_{\boldsymbol{\pi}},$$

where $\sigma^2_{\boldsymbol{\xi}_j}$ is the variance component of $\boldsymbol{\xi}_j$ in model (2).

This theorem implies that the GCV estimate $\hat{\mathbf{Z}}_j = \mathbf{P}_{\tau_j}\mathbf{Y}_j$ is approximately the optimal estimate of the conditional expectation $\mathbf{Z}_j$; furthermore, as the optimal tuning parameter approximates a constant determined by the variance components ratio, we make the following assumption on τ_j.

Assumption C. $\tau_j/\sqrt{n} \to 0$ as $n \to \infty$, for $j = 1, \cdots, p$.

We then have the following properties on $\hat{\mathbf{Z}}_{-k}$.

Theorem 5.2 *For $k = 1, \ldots, p$, let $\mathbf{M}_k = \boldsymbol{\pi}^T_{-k}(\mathbf{C} - \mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}^{-1}_{\mathcal{S}_k,\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k\bullet})\boldsymbol{\pi}_{-k}$ where each $\mathbf{C}_{\mathcal{S}_r\mathcal{S}_c}$ is a submatrix of $\mathbf{C}$ identified with row indices in $\mathcal{S}_r$ and column indices in $\mathcal{S}_c$ (the dot implies all rows or columns). Then, under Assumptions A, B, and C,*

a. $n^{-1}\hat{\mathbf{Z}}^T_{-k}\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \to_p \mathbf{M}_k$, *as* $n \to \infty$;

b. $n^{-1/2}(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \to_d N(\mathbf{0}, \sigma^2_k\mathbf{M}_k)$, *as* $n \to \infty$.

Since $n^{-1}\mathbf{Z}^T_{-k}\mathbf{H}_k\mathbf{Z}_{-k} \to \mathbf{M}_k$, Theorem 5.2.a states that $\hat{\mathbf{Z}}^T_{-k}\mathbf{H}_k\hat{\mathbf{Z}}_{-k}$ is a good approximation to $\mathbf{Z}^T_{-k}\mathbf{H}_k\mathbf{Z}_{-k}$. On the other hand, $\mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)$ is the error term in regressing $\mathbf{H}_k\mathbf{Y}_k$ against $\mathbf{H}_k\hat{\mathbf{Z}}_{-k}$, and Theorem 5.2.b implies that $n^{-1}(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \to_d 0$. Thus $\hat{\mathbf{Z}}_{-k}$ results in regression errors with good properties, i.e., the error effects on the 2SPLS estimators will vanish when the sample size gets sufficiently large.

In summary, the above theorem indicates that $\hat{\mathbf{Z}}_{-k}$ behaves the same way as $\mathbf{Z}_{-k}$ asymptotically, which makes it reasonable to replace $\mathbf{Z}_{-k}$ with $\hat{\mathbf{Z}}_{-k}$ at the second stage. Denote the j-th elements of $\boldsymbol{\gamma}_k$ and $\hat{\boldsymbol{\gamma}}_k$ as γ_{kj} and $\hat{\gamma}_{kj}$, respectively. Then, the properties of $\hat{\mathbf{Z}}_{-k}$ in Theorem 5.2, together with the oracle properties of the adaptive lasso, will lead to the following oracle properties of our proposed estimates.

Theorem 5.3 *(Oracle Properties) Let $\mathcal{A}_k = \{j : \gamma_{kj} \neq 0, j \neq k\}$ and $\hat{\mathcal{A}}_k = \{j : \hat{\gamma}_{kj} \neq 0, j \neq k\}$. Further index both rows and columns of $\mathbf{M}_k$ with $1, \cdots, k-1, k+1, \cdots, p$, and let $\mathbf{M}_{k,\mathcal{A}_k}$ be the submatrix of $\mathbf{M}_k$ identified with both row and column indices in $\mathcal{A}_k$. Suppose that $\lambda_k/\sqrt{n} \to 0$ and $\lambda_k n^{(\delta-1)/2} \to \infty$. Then, under Assumptions A, B, and C, the estimates from the proposed 2SPLS method satisfy the following properties,*

a. Consistency in variable selection: $\lim_{n\to\infty} P(\hat{\mathcal{A}}_k = \mathcal{A}_k) = 1$;

b. Asymptotic normality: $\sqrt{n}(\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k} - \boldsymbol{\gamma}_{k,\mathcal{A}_k}) \to_d N(\mathbf{0}, \sigma^2_k\mathbf{M}^{-1}_{k,\mathcal{A}_k})$, *as* $n \to \infty$.

It is worth mentioning that Theorem 5.2 plays an essential role in establishing the oracle properties of 2SPLS. In fact, as long as the properties in Theorem 5.2 hold true for the first-stage estimates of $\mathbf{Z}_{-k}$, the oracle properties can be expected from the adaptive lasso (Zou, 2006) at the second stage. On the other hand, we can also generalize the second-stage regularization to a wide class of regularization methods (Fan and Li, 2001; Huang *et al.*, 2011; Zhang, 2010), the theoretical properties, of which, can still be inherited due to the results in Theorem 5.2.

5.2 The Number of Endogenous Variables is Divergent

In this section, we investigate the theoretical properties of 2SPLS with a divergent p. That is, per Assumption A, both p and q may grow with sample size n at the the same order. The theoretical properties will be described by a prespecified sequence $f_n = o(n)$ but $f_n \to \infty$.

We first update Assumptions B and C for the divergent p and q.

Assumption B′. Both p and q grow at the same order of $o(n)$, i.e., $p \asymp q = o(n)$. Furthermore, the singular values of $\mathbf{I} - \boldsymbol{\Gamma}$ are positively bounded from below, and there exist positive constants c_1 and c_2 such that, for any vector δ with $\|\delta\|_2 = 1$, $c_1 \geq n^{-1/2}\|\mathbf{X}\delta\|_2 \geq c_2$.

Assumption C′. $r_{nk} \triangleq \tau^2_k \|\pi_k\|^2_2/n = o(n)$.

We have the following properties on the ridge regression estimator of $\boldsymbol{\pi}_k$ from the first stage.

Theorem 5.4 *Under Assumptions A, B′, and C′, for each ridge regression estimator $\hat{\boldsymbol{\pi}}_k$, there exist constants C_1 and C_2 such that, with probability at least $1 - e^{-f_n}$,*

(a) $\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|^2_2 \leq C_1(r_{nk} \vee q \vee f_n)/n$;

(b) $n^{-1}\|\mathbf{X}(\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k)\|^2_2 \leq C_2(r_{nk} \vee q \vee f_n)/n$.

Denote $r_{\max} = \max_{1\leq k\leq p} r_{nk}$. Then the system-wise losses in both $\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|^2_2$ and $n^{-1}\|\mathbf{X}(\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k)\|^2_2$ have upper bounds in the same order as $(r_{\max} \vee q \vee f_n)/n$, with probability at least $1 - e^{-(f_n - \log(p))}$. With $p = o(n)$, we henceforth select f_n to dominate $log(p)$, i.e. $f_n - \log(p) \to \infty$, to guarantee the well-controlled losses over the whole system.

Denote $\mathcal{A}_k = \{j : \gamma_{kj} \neq 0, j \neq k\}$. Indexing all rows and columns with only $j = 1, \cdots, k-1, k+1, \cdots, p$, we define the restricted eigenvalue for a $(p-1)\times(p-1)$ matrix $\mathbf{M}$ as

$$\phi_k(\mathbf{M}) = \min\left\{n^{-1/2}\|\mathbf{M}\gamma\|_2\|\gamma_{\mathcal{A}_k}\|^{-1}_2 : \|\gamma_{\mathcal{A}^c_k}\|_1 \leq 3\|\gamma_{\mathcal{A}_k}\|_1\right\}.$$

We further define $\|\cdot\|_\infty$ and $\|\cdot\|_{-\infty}$ to be the maximum and minimum absolute values of the components of a vector, respectively. For a matrix, $\|\cdot\|_\infty$ is defined to be the maximum absolute row sum of the matrix.

We further make the following assumption on the tuning parameter λ_k of the adaptive lasso at the second stage.

Assumption D. The adaptive tuning parameter λ_k is at the same order as $\|\omega_k\|_{-\infty}^{-1}\|\mathbf{\Gamma}\|_1 \|\boldsymbol{\pi}\|_1 \sqrt{n(r_{\max} \vee q \vee f_n)\log p}$.

We then have the consistency property of estimator $\hat{\boldsymbol{\gamma}}_k$.

Theorem 5.5 *(Estimation Consistency) Suppose that, for each node k, both inequalities $\|\omega_{k,\mathcal{A}_k}\|_\infty \|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1} \leq 1$ and $\sqrt{(r_{max} \vee q \vee f_n)/n} + c_1\|\boldsymbol{\pi}\|_1 \leq \sqrt{c_1^2\|\boldsymbol{\pi}\|_1^2 + \phi_0^2/64C_2|\mathcal{A}_k|}$ hold, and there exists a positive constant ϕ_0 such that $\phi_k(\mathbf{H}_k\mathbf{X}\boldsymbol{\pi}_{-k}) \geq \phi_0$. Denote $h_n = (\|\mathbf{\Gamma}\|_1^2 \wedge 1)\left[(\frac{n}{q}\|\boldsymbol{\pi}\|_1^2) \wedge (r_{max} \vee q \vee f_n)\right]\log p$. Under Assumptions A, B′, C′, and D, there exist constants $C_3 > 0$ and $C_4 > 0$ such that, with probability at least $1 - e^{-C_3 h_n + \log(4pq)} - e^{-f_n + \log(p)}$, each 2SPLS estimator $\hat{\boldsymbol{\gamma}}_k$ satisfies that*

1. $\|\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}_k\|_1 \leq 8C_4 \frac{\|\omega_{k,\mathcal{A}_k}\|_\infty \|\boldsymbol{\pi}\|_1 \|\mathbf{\Gamma}\|_1}{\phi_0^2 \|\omega_k\|_{-\infty}} |\mathcal{A}_k| \sqrt{\frac{(r_{max} \vee q \vee f_n)\log p}{n}}$;
2. $n^{-1}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}_k)\right\|_2^2 \leq \frac{C_4^2\|\omega_{k,\mathcal{A}_k}\|_\infty^2 \|\boldsymbol{\pi}\|_1^2 \|\mathbf{\Gamma}\|_1^2}{\phi_0^2\|\omega_k\|_{-\infty}^2} |\mathcal{A}_k| \frac{(r_{max} \vee q \vee f_n)\log p}{n}$.

Note that the system-wide upper bounds, defined by replacing $|\mathcal{A}_k|$ with $\max_k |\mathcal{A}_k|$, can also be achieved with probability at least $1 - e^{-C_3 h_n + \log(4q) + 2\log(p)} - e^{-f_n + 2\log(p)}$.

Let $W_k = diag\{\boldsymbol{\omega}_k\}$ and $V_k = (v_{ij})_{(p-1)\times(p-1)} \triangleq \frac{1}{n}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{H}_k\mathbf{X}\boldsymbol{\pi}_{-k}$. Further denote $W_{k,\mathcal{A}_k} = diag\{\boldsymbol{\omega}_{k,\mathcal{A}_k}\}$, $W_{k,\mathcal{A}_k^c} = diag\{\boldsymbol{\omega}_{k,\mathcal{A}_k^c}\}$, $V_{k,21} = (v_{ij})_{i\in\mathcal{A}_k^c, j\in\mathcal{A}_k}$, $V_{k,11} = (v_{ij})_{i\in\mathcal{A}_k, j\in\mathcal{A}_k}$, and $\theta_k = \left\|V_{k,11}^{-1}W_{k,\mathcal{A}_k}\right\|_\infty$. We then have the following selection property.

Theorem 5.6 *(Selection Consistency) Suppose that, for each node k, $V_{k,11}$ is invertible, and $\sqrt{(r_{max} \vee q \vee f_n)/n} + c_1\|\boldsymbol{\pi}\|_1 \leq \sqrt{c_1^2\|\boldsymbol{\pi}\|_1^2 + \min(\phi_0^2/64, \zeta(4-\zeta)^{-1}\|\boldsymbol{\omega}_k\|_{-\infty}/\theta_k)/(C_2|\mathcal{A}_k|)}$. Further assume that there exists a positive constant $\zeta \in (0,1)$ such that $\min_{j\in\mathcal{A}_k}|\gamma_{kj}| > \frac{2\lambda_k\theta_k}{n(2-\zeta)}$ and $\left\|W_{k,\mathcal{A}_k^c}^{-1}V_{k,21}V_{k,11}^{-1}W_{k,\mathcal{A}_k}\right\|_\infty < 1-\zeta$. Under Assumptions A, B′, C′, and D, there exists a 2SPLS estimator $\hat{\gamma}_k$ satisfying that, with probability at least $1 - e^{-C_5 h_n + \log(4pq)} - e^{-f_n + \log(p)}$ for some constant $C_5 > 0$, $\hat{\mathcal{A}}_k = \mathcal{A}_k$ with $\hat{\mathcal{A}}_k = \{j : \hat{\gamma}_{kj} \neq 0, j \neq k\}$.*

6. Simulation Studies

We conducted simulation studies to compare 2SPLS with the adaptive lasso based algorithm (AL) by Logsdon and Mezey (2010), and the sparsity-aware maximum likelihood algorithm (SML) by Cai *et al.* (2013). To investigate whether it is necessary to select instrumental variables at the first stage as proposed in Belloni *et al.* (2012), Lin *et al.* (2015), and Zhu (2015), we also consider a method which replaces the ridge regression at the first stage of 2SPLS with the adaptive lasso, that is, the two-stage adaptive lasso (2SAL) method. Both acyclic networks and cyclic networks were simulated, each involving 300 endogenous variables. Each endogenous variable was simulated to have, on average, one regulatory effect for sparse networks, or three regulatory effects for dense networks. The regulatory effects were independently simulated from a uniform distribution over $(-1,-0.5)\cup(0.5,1)$. To allow the use of AL and SML, every endogenous variable in the same network was

simulated to have the same number (either one or three) of nonzero exogenous effects (EEs) by the exogenous variables, with all effects equal to one. Each exogenous variable was simulated to take values 0, 1 and 2 with probabilities 0.25, 0.5 and 0.25, respectively, emulating genotypes of an F2 cross in a genetical genomics experiment. All error terms were independently simulated from $N(0, 0.1^2)$, and the sample size n varied from 100 to 1,000. For each network setup, we simulated 100 data sets and applied all four algorithms to calculate the power and false discovery rate (FDR).

For inferring acyclic networks, the power and FDR of the four different algorithms are plotted in Figure 1. 2SPLS has greater power than the other three algorithms to infer both sparse and dense acyclic networks when the sample size is small or moderate. When the sample size is large, 2SPLS, SML, and 2SAL are comparable for constructing both sparse and dense acyclic networks. In any case, AL has much lower power than other methods. Specifically, AL provides power as low as under 10% when the sample size is small, and its power is still under 50% even when the sample size increases to 1,000. On the other hand, 2SPLS provides power over 80% for small sample sizes, and over 90% for moderate to large sample sizes.

As shown in Figure 1, 2SPLS controls the FDR under 20% except for the case which has three available EEs with small sample sizes ($n = 100$). Although SML controls the FDR as low as under 5% for sparse acyclic networks when the sample sizes are large, it reports large FDRs when the sample sizes are small. For example, when the sample sizes are under 200, SML reports FDR over 40% for dense acyclic networks. In general, both 2SPLS and SML outperform AL and 2SAL in terms of FDR. Only in the case when inferring sparse acyclic networks with one available EE from data sets of moderate or large sample sizes, AL and 2SAL report FDR lower than 2SPLS.

Plotted in Figure 2 are the power and FDR of the four different algorithms when inferring cyclic networks. Similar to the results on acyclic networks, 2SPLS has greater power than SML and AL across all sample sizes and has lower FDR when the sample size is small. 2SPLS has greater power than 2SAL in most scenarios and has much lower FDR than 2SAL except for the case when inferring sparse cyclic networks from data sets of large sample sizes. SML provides power competitive to 2SPLS for sparse cyclic networks, but its power is much lower than that of 2SPLS for dense cyclic networks. Similar to the case of acyclic networks, SML reports much higher FDR for inferring dense networks from data sets with small sample sizes though it reports small FDR when the sample sizes are large. 2SAL reports the highest FDR, especially for networks with three available EEs.

Although not performing as well as 2SPLS, 2SAL reports competitive power to SML when inferring either acyclic or cyclic networks. For the acyclic sparse network with one EE, 2SAL can control FDR at a similar level to 2SPLS because each endogenous variable may be associated to a very small set of exogenous variables in (3). However, we observed high FDR of 2SAL in Figure 1.b for the acyclic sparse network with three EEs which triples the average number of exogenous variables associated to each endogenous variable. The similar phenomenon of 2SAL appears in Figure 2.b for the cyclic sparse networks. The dense networks also triple the average number of regulatory effects for each endogenous variable, which implies an increased number of exogenous variables associated to each endogenous variable in (3). Therefore, we unsurprisingly observed even higher FDR of 2SAL in Figure 1.d and Figure 2.d, where the FDR is over 0.8. In summary, variable selection at

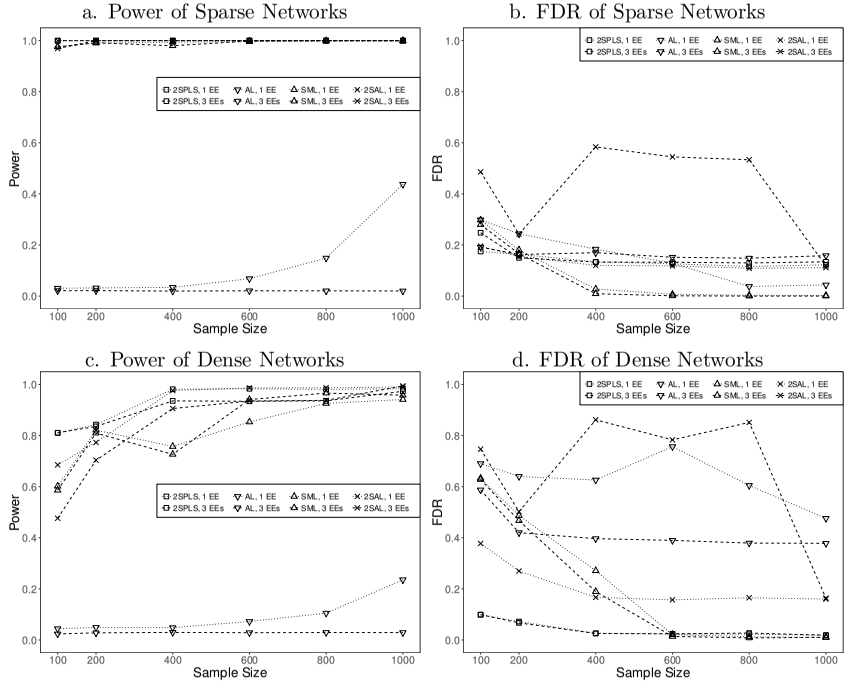


Figure 1: Performance of 2SPLS, AL, SML, and 2SAL when identifying regulatory effects in acyclic networks with one EE or three EEs.

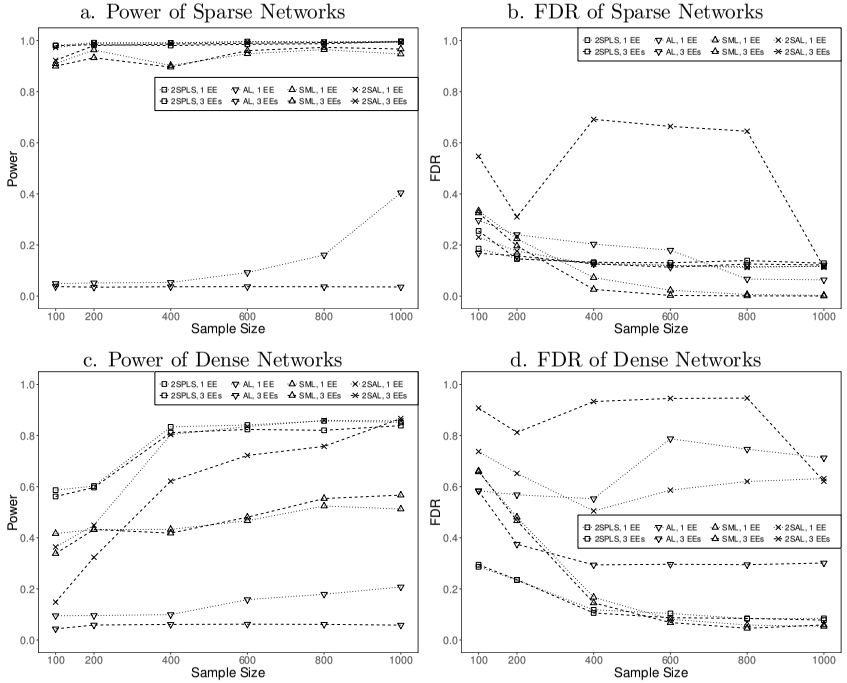


Figure 2: Performance of 2SPLS, AL, SML, and 2SAL when identifying regulatory effects in cyclic networks with one EE or three EEs.

the first stage seems work well when each endogenous variable is associated to a small set of exogenous variables in (3), but may compromise the identification of regulatory effects at the second stage when the number of exogenous variables associated to an endogenous variable increases.

Both 2SPLS and 2SAL are two-stage methods developed based on the limited-information model (2), instead of the full-information model used by SML, leading to fast computation and potential implementation of parallel computing. To demonstrate the computational advantage of 2SPLS and 2SAL, we recorded the computing time of all algorithms when inferring the same networks from small data sets ($n = 100$). Each algorithm analyzed the same data set using only one CPU in a server with Quad-Core AMD Opteron^{TM} Processor 8380. Reported in Table 1 are the running times of all four algorithms for inferring different networks. AL is the fastest although it performs with the least power. The running time of 2SPLS usually doubles or triples that of AL, but the computation time of 2SAL generally triples that of 2SPLS because 2SAL employed K-fold cross-validation to choose the tuning

parameter at the first stage. SML is the slowest algorithm which generally takes more than 40 times longer than 2SPLS to infer different networks. In particular, SML is almost 200 times slower than 2SPLS when inferring acyclic sparse networks.

	Acyclic				Cyclic			
	Sparse		Dense		Sparse		Dense	
	1 EE	3 EEs	1 EE	3 EEs	1 EE	3 EEs	1 EE	3 EEs
2SPLS	1303	1332	1127	1112	1297	1337	1125	1165
AL	405	652	404	637	443	659	430	781
SML	258875	195739	58509	43118	49393	58716	67949	68081
2SAL	3239	4726	3398	5357	3135	4681	3686	5651

Table 1: The running time (in seconds) of inferring networks from a data set with $n = 100$.

The robustness of 2SPLS was also evaluated from different aspects: (i) its robustness to different noise levels by doubling or even quadrupling the error variance; (ii) its robustness to non-normality of error terms by simulating errors sampled from a t-distribution, i.e., $t(3)$; (iii) its robustness to uncertainty in the connections between exogenous and endogenous variables by simulating three exogenous effects for each endogenous variable (to emulate the genetical genomics experiment, the three exogenous variables are correlated with correlation coefficients at 0.8, and have effects at 1, 0.5, and -0.3, respectively) but including only one exogenous variable with the strongest estimated effects for each endogenous variable; (iv) its robustness to existence of hub nodes by simulating networks with six hub nodes having five regulatory effects on average while other endogenous variables having on average one regulatory effect for sparse networks, or three regulatory effects for dense networks. All networks include 300 endogenous variables, and the networks with errors following $N(0, 0.01)$ are the same as those shown in Figure 1. As shown in Figure 3, the 2SPLS method demonstrated robust power while the FDR was slightly affected when the error variance doubled. When the error variance quadrupled, a higher FDR was reported as expected. With errors from $t(3)$, we observed similar power and slightly increased FDR of 2SPLS, which confirms the robustness of 2SPLS to non-normality. The uncertainty in the connections between exogenous and endogenous variables had almost no effect on the power of 2SPLS, and only slightly increased the FDR in constructing sparse networks. The existence of hub nodes rarely affected construction of dense networks, but decreased the FDR in constructing sparse networks. Overall, the performance of 2SPLS is remarkable in demonstrating robustness under a variety of realistic data structures.

7. Real Data Analysis

We analyzed a yeast data set with 112 segregants from a cross between two strains BY4716 and RM11-la (Brem and Kruglyak, 2005). A total of 5,727 genes were measured for their expression values, and 2,956 markers were genotyped. Each marker within a genetic region (including 1kb upstream and downstream regions) was evaluated for its association with the corresponding gene expression, yielding 722 genes with marginally significant cis-eQTL (p-value < 0.05). The set of cis-eQTL for each gene was filtered to control a pairwise correlation under 0.90, and then further filtered to keep up to three cis-eQTL which have the strongest association with the corresponding gene expression.

With 112 observations of 722 endogenous variables and 732 exogenous variables, we applied 2SPLS to infer the gene regulatory network in yeast. The constructed network includes 7,300 regulatory effects in total. To evaluate the reliability of constructed gene regulations, we generated 10,000 bootstrap data sets (each with $n = 112$) by randomly sampling the original data with replacement, and applied 2SPLS to each data set to infer the gene regulatory network. Among the 7,300 regulatory effects, 323 effects were repeatedly identified in more than 80% of the 10,000 data sets, and Figure 4 shows the three largest subnetworks formed by these 323 effects. Specifically, the largest subnetwork consists of 22 endogenous variables and 26 regulatory effects, the second largest one includes 14 endogenous variables and 18 regulatory effects, and the third largest one has 11 endogenous variables and 16 regulatory effects.

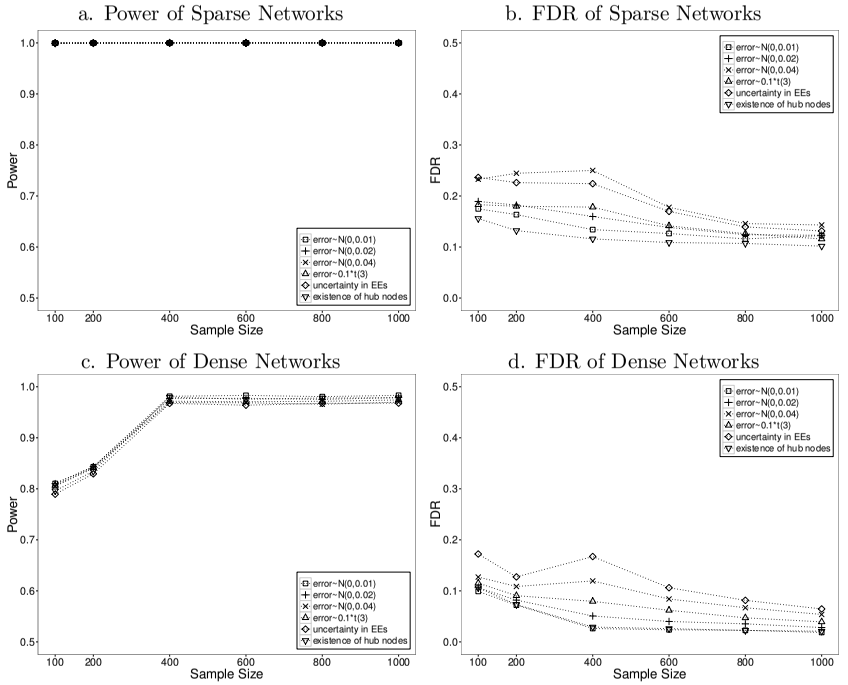


Figure 3: Performance of 2SPLS in robustness tests when identifying regulatory effects in acyclic networks with one EE.

A gene-enrichment analysis with DAVID (Huang *et al.*, 2009) showed that the three subnetworks are enriched in different gene clusters (controlling p-values from Fisher's exact tests under 0.01). A total of six gene clusters are enriched with genes from the first subnetwork, and four of them are related to either methylation or methyltransferase. Six of 22 genes in the first subnetwork are found in a gene cluster which is related to none-coding RNA processing. The second subnetwork is enriched in nine gene clusters. While three of the clusters are related to electron, one cluster includes half of the genes from the second subnetwork and is related to oxidation reduction. The third subnetwork is also enriched in nine different gene clusters, with seven clusters related to proteasome.

A total of 18 regulations were constructed from each of the 10,000 bootstrap data sets, and are shown in Figure 5. There are seven pairs of genes which regulate each other. It is interesting to observe that all regulatory genes up-regulate the target genes except two genes, namely, YCL018W and YEL021W.

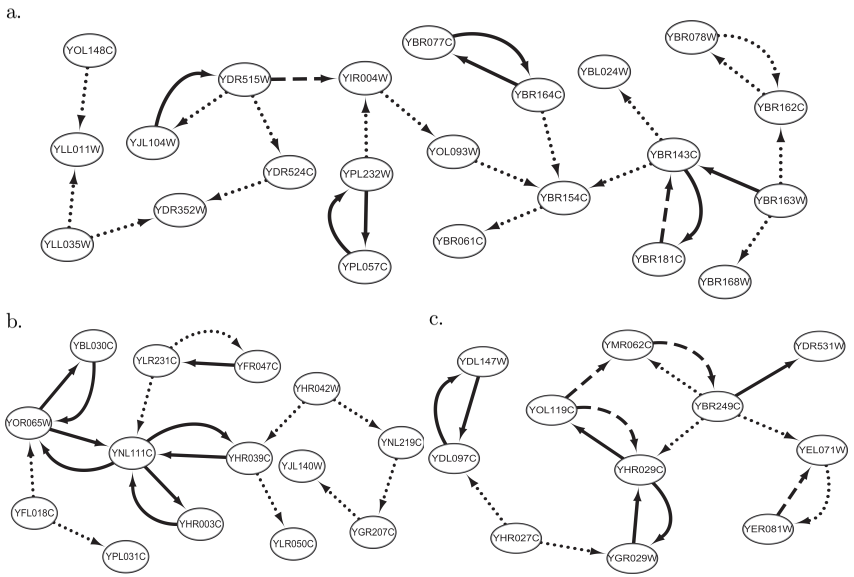


Figure 4: Three gene regulatory subnetworks in yeast (the dotted, dashed, and solid arrows implied that the corresponding regulations were constructed respectively from over 80%, 90%, and 95% of the bootstrap data sets).

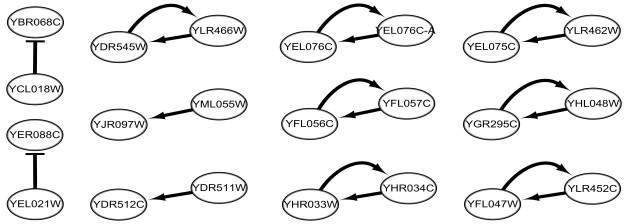


Figure 5: The yeast gene regulatory subnetworks constructed in each of 10,000 bootstrap data sets (with arrow- and bar-headed lines implying up and down regulations, respectively).

8. Discussion

In a classical setting with small numbers of endogenous/exogenous variables, constructing a system of structural equations has been well studied since Haavelmo (1943, 1944). Anderson and Rubin (1949) first proposed to estimate the parameters of a single structural equation with the limited information maximum likelihood estimator. Later, Theil (1953a,b, 1961) and Basmann (1957) independently developed the 2SLS estimator, which is the simplest and most common estimation method for fitting a system of structural equations. However, genetical genomics experiments usually collect data in which both the number of endogenous variables and the number of exogenous variables can be very large, invalidating the classical methods for building gene regulatory networks. It is noteworthy that, although each structural equation modeling gene regulation has few exogenous variables, the genome-wide gene regulatory network consists of a large number of structural equations and therefore has a large number of exogenous variables.

The instrumental variables view of 2SLS sheds light on the consistency of 2SLS estimators which is guaranteed by good estimation of the conditional expectations of endogenous variables given exogenous variables. For large systems, we proposed to estimate these conditional expectations via ridge regression coupled with GCV so as to address possible overfitting issues brought by a large number of exogenous variables. We obtained approximately optimal estimation of these conditional expectations at the first stage. At the second stage, we could adopt results from high-dimensional variable selection, e.g., Fan and Li (2001), Zou (2006), Zhang (2010), and Huang *et al.* (2011), to consistently identify and further estimate the regulatory effects of the endogenous variables. As a high-dimensional extension of the classical 2SLS method, the 2SPLS method is also computationally fast and easy to implement. As shown in constructing a genome-wide gene regulatory network of yeast, the high computational efficiency of 2SPLS allows us to employ the bootstrap method to calculate the p-values of the regulatory effects.

Our simulation studies show a seemingly counterintuitive result that our moment-based method 2SPLS provides higher power than the likelihood-based method SML, because the maximum likelihood method is usually the most efficient method, and dominates moment methods. However, as evidenced in Bollen (1996) and Kennedy (1985) (p.134), 2SLS can perform better than the maximum likelihood method in small samples. Furthermore, SML is not a pure likelihood method but rather a penalized likelihood method, and 2SPLS is not a pure moment method but rather a penalized moment method. Therefore, the theoretical advantage of likelihood methods over moment methods may not carry over to comparing penalized likelihood methods versus penalized moment methods. In fact, SML uses an L_1 penalty to penalize nonzero regulatory effects, but 2SPLS employs an L_2 penalty on the regression coefficients of the reduced models at the first stage and an L_1 penalty on the regulatory effects at the second stage. We conjecture that the different choice of penalty terms may also distinguish the two different methods as shown in the advantage of the elastic net (Zou and Hastie, 2005) over lasso (Tibshirani, 1996).

Although applicable to diverse fields, our development of 2SPLS is motivated by constructing gene regulatory networks using genetical genomics data. The algorithm is applicable to any population-based studies with either experimental crosses or natural populations. Assumption A means that each gene under investigation has at least one unique polymor-

phism from its cis-eQTL, which can be detected with classical eQTL mapping methods, e.g., Kendziorski *et al.* (2006), Gelfond *et al.* (2007), and Jia and Xu (2007). Trans-eQTL (i.e., eQTL outside the regions of their target genes) hold the key to our understanding of gene regulation because their indirect regulations are likely caused by interactions among genes. When the gene regulatory network is modeled with a system of structural equations, classical eQTL mapping methods essentially identify both cis-eQTL and trans-eQTL involved in each reduced-form equation in the reduced model (3). Nonetheless, it is challenging, if not impossible, to recover a large system from the reduced model.

An alternative strategy to construct the whole system is to build undirected graphs first (Spirtes *et al.*, 2001; Shipley, 2002; de la Fuente *et al.*, 2004) and then locally orient the edges in the graphs (Aten *et al.*, 2008; Neto *et al.*, 2008). While constructing a small network is much easier and more robust than constructing a large system, we here intend to construct large networks, such as whole-genome gene regulatory networks from genetical genomics data. Furthermore, application of the alternative strategy is contingent on whether the underlying system is composed of unconnected subsystems, because ignoring the regulatory effects from other genes outside a subset of genes may lead to false regulatory interaction (Neto *et al.*, 2008; de la Fuente *et al.*, 2004). Instead, 2SPLS allows to construct a subset of structural equations inside the whole system, ignoring many other structural equations. Therefore, we can apply 2SPLS to investigate the interactive regulation among a subset of genes as well as how these genes are regulated by others.

It is evidenced in different species that effects of trans-eQTL are weaker than those of cis-eQTL and trans-eQTL are more difficult to identify than cis-eQTL (Schadt *et al.*, 2003; Dixon *et al.*, 2007). On the other hand, a system of structural equations modeling genome-wide gene regulation may induce a large number of trans-eQTL to each reduced-form equation in (3). While constructing the system is contingent on the accuracy of predicting each endogenous variable on the basis of the corresponding reduced-form equation in (3), the weak effects of a large number of trans-eQTL privilege the use of ridge regression at the first stage of 2SPLS for constructing gene regulatory networks (Frank and Friedman, 1993). By comparing 2SPLS with 2SAL, our simulation studies demonstrated the superiority of using ridge regression over the adaptive lasso at the first stage. In fact, when some genes have a relatively large number of trans-eQTL, selecting variables at the first stage may compromise the identification of regulatory effects at the second stage.

Acknowledgments

We thank the action editor and four anonymous reviewers for their helpful comments. This work is partially supported by NSF CAREER award IIS-0844945, NIH R03CA211831, and the Cancer Care Engineering project at the Oncological Science Center of Purdue University.

Appendix A: Proof of Theorem 5.2

a. Since $\tau_j/\sqrt{n} \to 0$ for any $1 \le j \le p$, the different choice of τ_j for each j does not affect the following asymptotic property involving τ_j,

$$n(\mathbf{X}^T\mathbf{X} + \tau_j\mathbf{I})^{-1} \to \mathbf{C}. \tag{9}$$

Without loss of generality, we assume $\tau_1 = \tau_2 = \cdots = \tau_p = \tau$. Then $\hat{\mathbf{Z}}_{-k} = \mathbf{P}_\tau \mathbf{Y}_{-k}$.

$$\begin{aligned} n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} &= n^{-1}(\mathbf{X}\boldsymbol{\pi}_{-k} + \boldsymbol{\xi}_{-k})^T\mathbf{P}_\tau^T\mathbf{H}_k\mathbf{P}_\tau(\mathbf{X}\boldsymbol{\pi}_{-k} + \boldsymbol{\xi}_{-k}) \\ &= n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} + n^{-1}\boldsymbol{\xi}_{-k}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \\ &\quad + n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k} + n^{-1}\boldsymbol{\xi}_{-k}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k} \end{aligned}$$

We will consider the asymptotic property of each of the above four terms.

First, $n^{-1}\mathbf{X}^T\mathbf{X} \to \mathbf{C}$ implies that

$$n^{-1}\mathbf{X}^T\mathbf{H}_k\mathbf{X} = n^{-1}\mathbf{X}^T\{\mathbf{I} - \mathbf{X}_{\mathcal{S}_k}(\mathbf{X}_{\mathcal{S}_k}^T\mathbf{X}_{\mathcal{S}_k})^{-1}\mathbf{X}_{\mathcal{S}_k}^T\}\mathbf{X} \to \mathbf{C} - \mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet}. \tag{10}$$

The above result and (9) easily lead to the following result,

$$\begin{aligned} & n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \\ &= n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau\mathbf{I})^{-1}\mathbf{X}^T\mathbf{H}_k\mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\pi}_{-k} \\ &\to \boldsymbol{\pi}_{-k}^T(\mathbf{C} - \mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet})\boldsymbol{\pi}_{-k} = \mathbf{M}_k. \end{aligned} \tag{11}$$

The other three terms approaching to zero directly follows that $n^{-1}\boldsymbol{\xi}_{-k}^T\mathbf{X} \to_p \mathbf{0}$. Thus, $\frac{1}{n}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \to_p \mathbf{M}_k$.

b. Since $\mathbf{H}_k(\mathbf{Y}_k - \mathbf{Y}_{-k}\boldsymbol{\gamma}_k) = \mathbf{H}_k\boldsymbol{\epsilon}_k$, we have

$$\begin{aligned} & n^{-1/2}(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \\ &= n^{-1/2}\{(\mathbf{Y}_k - \mathbf{Y}_{-k}\boldsymbol{\gamma}_k) + (\mathbf{I} - \mathbf{P}_\tau)\mathbf{Y}_{-k}\boldsymbol{\gamma}_k\}^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k} \\ &= n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{Y}_{-k} + n^{-1/2}\boldsymbol{\gamma}_k^T\{(\mathbf{I} - \mathbf{P}_\tau)\mathbf{Y}_{-k}\}^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{Y}_{-k}. \end{aligned}$$

In the following, we will prove that the second term approaches to zero, and the first term asymptotically approaches to the required distribution, i.e.,

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{Y}_{-k} \to_d N(\mathbf{0}, \sigma_k^2\mathbf{M}_k). \tag{12}$$

We notice that

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \sim N(\mathbf{0}, n^{-1}\sigma_k^2\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k}).$$

Following (11), we have

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \to_d N(\mathbf{0}, \sigma_k^2\mathbf{M}_k). \tag{13}$$

Because of (10) and

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{X} \sim N(\mathbf{0}, n^{-1}\sigma_k^2\mathbf{X}^T\mathbf{H}_k\mathbf{X}),$$

we have

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{X} \to_d N(\mathbf{0}, \sigma_k^2(\mathbf{C} - \mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet})).$$

Since $n^{-1}\boldsymbol{\xi}_{-k}^T\mathbf{X} \to_p \mathbf{0}$, we can apply Slutsky's theorem and obtain that

$$n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k} = n^{-1/2}\boldsymbol{\epsilon}_k^T\mathbf{H}_k\mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{\xi}_{-k} \to_p \mathbf{0}.$$

Pooling the above result and (13) leads to the asymptotic distribution in (12).

To prove that the second term asymptotically approaches to zero, we further partition it as follows,

$$\begin{aligned} & n^{-1/2}\boldsymbol{\gamma}_k^T\{(\mathbf{I}-\mathbf{P}_\tau)\mathbf{Y}_{-k}\}^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{Y}_{-k} \\ = \;& n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} + n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \\ & + n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k} + n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k}. \end{aligned}$$

It suffices to prove each of these four parts asymptotically approaches to zero.

First, notice that

$$\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau) = \tau(\mathbf{X}^T\mathbf{X}+\tau\mathbf{I})^{-1}\mathbf{X}^T,$$

we have

$$\begin{aligned} & n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k} \\ = \;& n^{-1/2}\tau\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T(\mathbf{X}^T\mathbf{X}+\tau\mathbf{I})^{-1}\mathbf{X}^T\mathbf{H}_k\mathbf{X}(\mathbf{X}^T\mathbf{X}+\tau\mathbf{I})^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\pi}_{-k} \to \mathbf{0}, \end{aligned} \tag{14}$$

which follows (10) and that $\tau/\sqrt{n}\to 0$ as $n\to\infty$.

Because $\mathbf{C}_{\mathcal{S}_k\bullet}\mathbf{C}^{-1}\mathbf{C}_{\bullet\mathcal{S}_k} = \mathbf{C}_{\mathcal{S}_k\mathcal{S}_k}$, we have

$$(\mathbf{C}-\mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet})\mathbf{C}^{-1}(\mathbf{C}-\mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet}) = \mathbf{C}-\mathbf{C}_{\bullet\mathcal{S}_k}\mathbf{C}_{\mathcal{S}_k,\mathcal{S}_k}^{-1}\mathbf{C}_{\mathcal{S}_k\bullet},$$

which implies that

$$\begin{aligned} & n^{-1/2}\mathbf{X}^T\mathbf{P}_\tau^T\mathbf{H}_k^T(\mathbf{I}-\mathbf{P}_\tau)^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X} \\ = \;& n^{-1}\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X} - 2n^{-1}\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau\mathbf{X} + n^{-1}\mathbf{X}^T\mathbf{P}_\tau\mathbf{H}_k\mathbf{P}_\tau^2\mathbf{H}_k\mathbf{P}_\tau\mathbf{X} \to \mathbf{0}. \end{aligned}$$

Since $\mathrm{Var}(\boldsymbol{\xi}_{-k}\boldsymbol{\gamma}_k)$ is proportional to an identity matrix, the above result leads to that

$$\mathrm{Var}\left(n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k}\right)\to\mathbf{0},$$

which implies that

$$n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\mathbf{X}\boldsymbol{\pi}_{-k}\to_p\mathbf{0}. \tag{15}$$

Similarly, we can prove that, for each $\boldsymbol{\xi}_j$,

$$\mathrm{Var}\left(n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_j\right)\to\mathbf{0},$$

which implies that

$$n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\pi}_{-k}^T\mathbf{X}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k}\to_p\mathbf{0}. \tag{16}$$

Note that

$$n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k} = \left\{n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{X}\right\}\left\{(\mathbf{X}^T\mathbf{X}+\tau\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{\xi}_{-k}\right\}.$$

Since

$$n^{-1}\mathbf{X}^T\mathbf{H}_k(\mathbf{I}-\mathbf{P}_\tau)(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{X}\to\mathbf{0},$$

we have

$$\mathrm{Var}\left(n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{X}\right)\to\mathbf{0}.$$

Therefore,

$$n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{X}\to_p\mathbf{0},$$

which, together with $(\mathbf{X}^T\mathbf{X}+\tau\mathbf{I})^{-1}\mathbf{X}^T\boldsymbol{\xi}_{-k}\to_p\mathbf{0}$, leads to that

$$n^{-1/2}\boldsymbol{\gamma}_k^T\boldsymbol{\xi}_{-k}^T(\mathbf{I}-\mathbf{P}_\tau)\mathbf{H}_k\mathbf{P}_\tau\boldsymbol{\xi}_{-k}\to_p\mathbf{0}. \tag{17}$$

Pooling (14), (15), (16) and (17), we have proved that $n^{-1/2}\boldsymbol{\gamma}_k^T\{(\mathbf{I}-\mathbf{P}_\tau)\mathbf{Y}_{-k}\}^T\mathbf{H}_k\mathbf{P}_\tau\mathbf{Y}_{-k}\to_p\mathbf{0}$, which concludes the proof.

Appendix B: Proof of Theorem 5.3

Let $\psi_n(\boldsymbol{\mu}) = \|\mathbf{H}_k\mathbf{Y}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\boldsymbol{\gamma}_k+\boldsymbol{\mu}/\sqrt{n})\|_2^2+\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k+\boldsymbol{\mu}/\sqrt{n}|$. Let $\hat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}}\psi_n(\boldsymbol{\mu})$, then $\hat{\boldsymbol{\gamma}}_k = \boldsymbol{\gamma}_k+\hat{\boldsymbol{\mu}}/\sqrt{n}$ or $\hat{\boldsymbol{\mu}} = \sqrt{n}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)$. Note that $\psi_n(\boldsymbol{\mu})-\psi_n(\mathbf{0}) = V_n(\boldsymbol{\mu})$, where

$$\begin{aligned} V_n(\boldsymbol{\mu}) \;=\; & \boldsymbol{\mu}^T(n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k})\boldsymbol{\mu} - 2n^{-1/2}(\mathbf{Y}_k-\hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k)^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\boldsymbol{\mu} \\ & + n^{-1/2}\lambda_k\boldsymbol{\omega}_k^T\times\sqrt{n}(|\boldsymbol{\gamma}_k+n^{-1/2}\boldsymbol{\mu}|-|\boldsymbol{\gamma}_k|). \end{aligned}$$

Denote the j-th elements of $\boldsymbol{\omega}_k$ and $\boldsymbol{\mu}$ as ω_{kj} and μ_j, respectively.

If $\gamma_{kj}\neq 0$, then $\omega_{kj}\to_p|\gamma_{kj}|^{-\delta}$ and $\sqrt{n}(|\gamma_{kj}+\mu_j/\sqrt{n}|-|\gamma_{kj}|)\to_p\mu_j\mathrm{sign}(\gamma_{kj})$. By Slutsky's theorem, we have $(\lambda_k/\sqrt{n})\omega_{kj}\sqrt{n}(|\gamma_{kj}+\mu_j/\sqrt{n}|-|\gamma_{kj}|)\to_p 0$. If $\gamma_{kj}=0$, then $\sqrt{n}(|\gamma_{kj}+\mu_j/\sqrt{n}|-|\gamma_{kj}|) = |\mu_j|$ and $(\lambda_k/\sqrt{n})\omega_{kj} = (\lambda_k/\sqrt{n})n^{\delta/2}(|\sqrt{n}\tilde{\gamma}_{kj}|)^{-\delta}$, where $\sqrt{n}\tilde{\gamma}_{kj} = O_p(1)$. Thus,

$$n^{-1/2}\lambda_k\boldsymbol{\omega}_k^T\times n^{1/2}(|\boldsymbol{\gamma}_k+n^{-1/2}\boldsymbol{\mu}|-|\boldsymbol{\gamma}_k|)\to_p\begin{cases}0, & \text{if } \|\boldsymbol{\mu}_{\mathcal{A}_k^c}\|_2=0;\\ \infty, & \text{otherwise.}\end{cases}$$

Hence, following Theorem 5.2 and Slutsky's theorem, we see that $V_n(\boldsymbol{\mu})\to_d V(\boldsymbol{\mu})$ for every $\boldsymbol{\mu}$, where

$$V(\boldsymbol{\mu}) = \begin{cases}\boldsymbol{\mu}_{\mathcal{A}_k}^T\mathbf{M}_{k,\mathcal{A}_k}\boldsymbol{\mu}_{\mathcal{A}_k}-2\boldsymbol{\mu}_{\mathcal{A}_k}^T\mathbf{W}_{k,\mathcal{A}_k}, & \text{if } \|\boldsymbol{\mu}_{\mathcal{A}_k^c}\|_2=0;\\ \infty, & \text{otherwise.}\end{cases}$$

$V_n(\boldsymbol{\mu})$ is convex, and the unique minimizer of $V(\boldsymbol{\mu})$ is $(\mathbf{M}_{k,\mathcal{A}_k}^{-1}\mathbf{W}_{k,\mathcal{A}_k},\mathbf{0})^T$. Following the epi-convergence results of Geyer (1994) and Fu and Knight (2000), we have

$$\begin{cases}\hat{\boldsymbol{\mu}}_{\mathcal{A}_k}\to_d\mathbf{M}_{k,\mathcal{A}_k}^{-1}\mathbf{W}_{k,\mathcal{A}_k},\\ \hat{\boldsymbol{\mu}}_{\mathcal{A}_k^c}\to_d\mathbf{0}.\end{cases}$$

Since $\mathbf{W}_{k,\mathcal{A}_k}\sim N(\mathbf{0},\sigma_k^2\mathbf{M}_{k,\mathcal{A}_k})$, we indeed have proved the asymptotic normality.

Now we show the consistency in variable selection. $\forall j\in\mathcal{A}_k$, the asymptotic normality indicates that $\hat{\gamma}_{kj}\to_p\gamma_{kj}$, thus $P(j\in\hat{\mathcal{A}}_k)\to 1$. Then it suffices to show that $\forall j\notin\mathcal{A}_k$, $P(j\in\hat{\mathcal{A}}_k)\to 0$.

When $j \in \hat{\mathcal{A}}_k$, by the KKT normality conditions, we know that $\hat{\mathbf{Z}}_j^T \mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\hat{\gamma}_k) = \lambda_k \omega_{kj}$. Note that $\lambda_k \omega_{kj}/\sqrt{n} \to_p \infty$, whereas $\hat{\mathbf{Z}}_j^T \mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\hat{\gamma}_k)/\sqrt{n} = (\hat{\mathbf{Z}}_j^T \mathbf{H}_k \hat{\mathbf{Z}}_{-k}/n) \times \sqrt{n}(\gamma_k - \hat{\gamma}_k) + \hat{\mathbf{Z}}_j^T \mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\gamma_k)/\sqrt{n}$. Following Theorem 5.2 and the asymptotic normality, $\hat{\mathbf{Z}}_j^T \mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\hat{\gamma}_k)/\sqrt{n}$ asymptotically follows a normal distribution. Thus, $P(j \in \hat{\mathcal{A}}_k) \le P(\hat{\mathbf{Z}}_j^T \mathbf{H}_k(\mathbf{Y}_k - \hat{\mathbf{Z}}_{-k}\hat{\gamma}_k) = \lambda_k \omega_{kj}) \to 0$. Then we have proved the consistency in variable selection.

Appendix C: Proof of Theorem 5.4

Denote $\lambda_{min}(\mathbf{M})$ and $\lambda_{max}(\mathbf{M})$ the minimum and maximum eigenvalues of matrix $\mathbf{M}$, respectively. Follow Assumption B′ to assume that the singular values of matrix $\mathbf{I} - \mathbf{\Gamma}$ are positively bounded from below by a constant c. Further denote $\tilde{\sigma}_k^2 = \mathrm{var}(\boldsymbol{\xi}_k)$, and $\sigma_{p\max}^2 = \max_{1 \le k \le p}(\sigma_k^2)$. Noting that $\boldsymbol{\xi} = \boldsymbol{\epsilon}(\mathbf{I} - \mathbf{\Gamma})^{-1}$, we have $\tilde{\sigma}_k^2 \le \sigma_{p\max}^2/c$.

(a) From the ridge regression, we have the following closed form solution,

$$\hat{\boldsymbol{\pi}}_k = (\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}\mathbf{X}^T\mathbf{Y}_k = (\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}\mathbf{X}^T\mathbf{X}\boldsymbol{\pi}_k + (\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}\mathbf{X}^T\boldsymbol{\xi}_k.$$

Note that

$$\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k = -\tau_k(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}\boldsymbol{\pi}_k + (\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}\mathbf{X}^T\boldsymbol{\xi}_k = \boldsymbol{\mu} + A_k^T\boldsymbol{\xi}_k,$$

where $\boldsymbol{\mu} = -\tau_k(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}$ and $A_k = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-1}$. Then we have

$$\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|_2^2 = \underbrace{\boldsymbol{\mu}^T\boldsymbol{\mu}}_{T_1} + \underbrace{2\boldsymbol{\mu}^T A_k^T\boldsymbol{\xi}_k}_{T_2} + \underbrace{\boldsymbol{\xi}_k^T A_k A_k^T\boldsymbol{\xi}_k}_{T_3}. \tag{18}$$

Via the singular value decomposition of $\mathbf{X}$, we can have the decomposition $\mathbf{X}^T\mathbf{X} = \mathbf{P}^T\mathbf{U}\mathbf{P}$, where $\mathbf{P}$ is a unitary matrix and matrix $\mathbf{U}$ is a diagonal matrix with non-negative diagonal elements u_i. Therefore,

$$(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-2} = \mathbf{P}^T(\mathbf{U} + \tau_k I_q)^{-2}\mathbf{P}.$$

Following Assumption B′, we have $\lambda_{min}(\mathbf{X}^T\mathbf{X}) > c_2^2 n$ and $\lambda_{max}(\mathbf{X}^T\mathbf{X}) < c_1^2 n$, which implies that $u_i \asymp n$ for all i. Therefore,

$$T_1 = \tau_k^2\boldsymbol{\pi}_k^T\mathbf{P}^T(\mathbf{U} + \tau_k I_q)^{-2}\mathbf{P}\boldsymbol{\pi}_k = \sum_{i=1}^{q}\frac{\tau_k^2 a_{ik}^2}{(u_i + \tau_k)^2} = \mathcal{O}(\tau_k^2\|\boldsymbol{\pi}_k\|_2^2/n^2) = \mathcal{O}(r_{nk}/n), \tag{19}$$

where a_{ik} is the i-th element of $\mathbf{a}_k = \mathbf{P}\boldsymbol{\pi}_k$ with $\|\mathbf{a}_k\|_2 = \|\boldsymbol{\pi}_k\|_2$.

For the term T_2, we have that

$$E[T_2] = 0, \qquad \mathrm{Var}(T_2) = 4\tilde{\sigma}_k^2\boldsymbol{\mu}^T A_k^T A_k\boldsymbol{\mu}.$$

By the classical Gaussian tail probability, we have

$$\mathbb{P}(T_2 \le t) \ge 1 - \exp\left\{-t^2/\left(8\tilde{\sigma}_k^2\boldsymbol{\mu}^T A_k^T A_k\boldsymbol{\mu}\right)\right\}.$$

Note that,

$$\boldsymbol{\mu}^T A_k^T A_k\boldsymbol{\mu} = \tau_k^2\boldsymbol{\pi}_k^T\mathbf{P}^T(\mathbf{U} + \tau_k I_q)^{-2}\mathbf{U}(\mathbf{U} + \tau_k I_q)^{-2}\mathbf{P}\boldsymbol{\pi}_k = \sum_{i=1}^{q}\frac{\tau_k^2 u_i a_{ik}^2}{(u_i + \tau_k)^4} = \mathcal{O}(\tau_k^2\|\boldsymbol{\pi}_k\|_2^2/n^3).$$

Letting $t = \sqrt{8\tilde{\sigma}_k^2\boldsymbol{\mu}^T A_k^T A_k\boldsymbol{\mu}(f_n + \log 2)}$, we have, with probability at least $1 - e^{-f_n}/2$,

$$T_2 = \mathcal{O}(\sqrt{r_{nk} f_n}/n). \tag{20}$$

For the term T_3, we can invoke the Hanson-Wright inequality (Rudelson and Vershynin, 2013) to have, for some constant $t_1 > 0$,

$$P(T_3 \le E[T_3] + t) \ge 1 - \exp\left\{-t_1\min\left(\frac{t^2}{\tilde{\sigma}_k^4\left\|A_k A_k^T\right\|_F^2}, \frac{t}{\tilde{\sigma}_k^2\left\|A_k A_k^T\right\|_{op}}\right)\right\},$$

where $\|\cdot\|_{op} = \max_{x \ne 0}\|\cdot x\|_2/\|x\|_2$ is the operator norm and $\|\cdot\|_F$ is the Frobenius norm.

Since

$$A_k A_k^T = \mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-2}\mathbf{X}^T = \mathbf{X}\mathbf{P}^T(\mathbf{U} + \tau_k I_q)^{-2}\mathbf{P}\mathbf{X}^T,$$

we have

$$\begin{aligned}
E[T_3] &= \tilde{\sigma}_k^2\mathrm{trace}(A_k A_k^T) = \tilde{\sigma}_k^2\mathrm{trace}(\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X} + \tau_k I_q)^{-2}) \\
&= \tilde{\sigma}_k^2\mathrm{trace}(\mathbf{U}(\mathbf{U} + \tau_k I_q)^{-2}) = \sum_{i=1}^{q}\frac{\tilde{\sigma}_k^2 u_i}{(u_i + \tau_k)^2} = \mathcal{O}(\tilde{\sigma}_k^2 q/n), \\
\left\|A_k A_k^T\right\|_F^2 &= \mathrm{trace}(A_k A_k^T A_k A_k^T) = \mathrm{trace}(A_k^T A_k A_k^T A_k) \\
&= \mathrm{trace}(P^T\mathbf{U}(\mathbf{U} + \tau_k I_q)^{-2}U(U + \tau_k I_q)^{-2}) = \sum_{i=1}^{q}\frac{u_i^2}{(u_i + \tau_k)^4} = \mathcal{O}(q/n^2), \\
\left\|A_k A_k^T\right\|_{op} &= \mathcal{O}(\lambda_{max}\left(\mathbf{X}\mathbf{X}^T\right)/n^2) = \mathcal{O}(n^{-1}).
\end{aligned}$$

Letting $t = \max\left(\sqrt{\tilde{\sigma}_k^4\left\|A_k A_k^T\right\|_F^2(f_n + \log 2)/t_1}, \tilde{\sigma}_k^2\left\|A_k A_k^T\right\|_{op}(f_n + \log 2)/t_1\right)$, we obtain that, with probablity at least $1 - e^{-f_n}/2$,

$$T_3 = \mathcal{O}(q/n) + \mathcal{O}(\sqrt{f_n q}/n) + \mathcal{O}(f_n/n). \tag{21}$$

Collecting the bounds in (19), (20), and (21), we conclude that there exist a positive constant C_1 such that, with probability at least $1 - e^{-f_n}$,

$$\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|_2^2 \le C_1(r_{nk} \vee q \vee f_n)/n.$$

(b) Similar to (18), we have

$$\|\mathbf{X}(\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k)\|_2^2 = \underbrace{\boldsymbol{\mu}^T\mathbf{X}^T\mathbf{X}\boldsymbol{\mu}}_{T_4} + \underbrace{2\boldsymbol{\mu}^T\mathbf{X}^T\mathbf{X}A_k^T\boldsymbol{\xi}_k}_{T_5} + \underbrace{\boldsymbol{\xi}_k^T A_k\mathbf{X}^T\mathbf{X}A_k^T\boldsymbol{\xi}_k}_{T_6}.$$

For the term T_4, we have

$$\begin{aligned} T_4 &= \tau_k^2 \mathbf{a}_k^T \mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{a}_k \\ &= \tau_k^2 \sum_{i=1}^{q} \frac{u_i a_{ik}^2}{(u_i+\tau_k)^2} = \mathcal{O}(\tau_k^2 \left\|\boldsymbol{\pi}_k\right\|_2^2/n) = \mathcal{O}(r_{nk}). \end{aligned} \tag{22}$$

For the term T_5, by the classical Gaussian tail inequality, we have

$$P\left(T_5 \le t\right) \ge 1 - \exp\left\{-t^2/(2\mathrm{Var}(T_5))\right\},$$

where

$$\begin{aligned} \mathrm{Var}(T_5) &= 4\tilde{\sigma}_k^2 \boldsymbol{\mu}^T \mathbf{X}^T \mathbf{X} A_k^T A_k \mathbf{X}^T \mathbf{X}\boldsymbol{\mu} \\ &= 4\tilde{\sigma}_k^2 \tau_k^2 \mathbf{a}_k^T (\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{a}_k \\ &= 4\tilde{\sigma}_k^2 \tau_k^2 \sum_{i=1}^{q} \frac{u_i^3 a_{ik}^2}{(u_i+\tau_k)^4} = \mathcal{O}(\tilde{\sigma}_k^2 \tau_k^2 \left\|\boldsymbol{\pi}_k\right\|_2^2/n). \end{aligned}$$

Taking $t = \sqrt{2\mathrm{Var}(T_5)(f_n + \log 2)}$, we can obtain that, with probability at least $1 - e^{-f_n}/2$,

$$T_5 = \mathcal{O}(\sqrt{r_{nk} f_n}). \tag{23}$$

For the term T_6, by the Hanson-Wright inequality, we have, for some constant $t_2 > 0$,

$$P(T_6 \le \mathbb{E}(T_6) + t) \ge 1 - \exp\left\{-t_2 \min\left(\frac{t^2}{\tilde{\sigma}_k^4 \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_F^2}, \frac{t}{\tilde{\sigma}_k^2 \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_{op}}\right)\right\}.$$

Similar to managing the term T_3 in (a), we have

$$\begin{aligned} E[T_6] &= \tilde{\sigma}_k^2 \mathrm{trace}(A_k \mathbf{X}^T \mathbf{X} A_k^T) = \tilde{\sigma}_k^2 \mathrm{trace}(U(U+\tau_k I_q)^{-1} U (U+\tau_k I_q)^{-1}) \\ &= \tilde{\sigma}_k^2 \sum_{i=1}^{q} \frac{u_i^2}{(u_i+\tau_k)^2} = \mathcal{O}(\tilde{\sigma}_k^2 q), \\ \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_F^2 &= \mathrm{trace}(A_k \mathbf{X}^T \mathbf{X} A_k^T A_k \mathbf{X}^T \mathbf{X} A_k^T) = \mathrm{trace}(\mathbf{X}^T \mathbf{X} A_k^T A_k \mathbf{X}^T \mathbf{X} A_k^T A_k) \\ &= \mathrm{trace}(\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}\mathbf{U}(\mathbf{U}+\tau_k I_q)^{-1}) \\ &= \sum_{i=1}^{q} \frac{u_i^4}{(u_i+\tau_k)^4} = \mathcal{O}(q), \\ \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_{op} &= \left\|\mathbf{X}(\mathbf{X}^T\mathbf{X}+\tau_k I_q)^{-1}\mathbf{X}^T\mathbf{X}(\mathbf{X}^T\mathbf{X}+\tau_k I_q)^{-1}\mathbf{X}^T\right\|_{op} \\ &= \mathcal{O}(\lambda_{\max}\left(\mathbf{X}\mathbf{X}^T\mathbf{X}\mathbf{X}^T\right)/n^2) = \mathcal{O}(1). \end{aligned}$$

Letting $t = \max\left(\sqrt{\tilde{\sigma}_k^4 \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_F^2 (f_n + \log 2)/t_2}, \tilde{\sigma}_k^2 \left\|A_k \mathbf{X}^T \mathbf{X} A_k^T\right\|_{op} (f_n + \log 2)/t_2\right)$, we have that, with probability at least $1 - e^{-f_n}/2$,

$$T_6 = \mathcal{O}(q) + \mathcal{O}(\sqrt{q f_n}) + \mathcal{O}(f_n). \tag{24}$$

Collecting the bounds in (22), (23), and (24), we conclude that there exists a positive constant C_2 such that, with probability at least $1 - e^{-f_n}$,

$$n^{-1} \left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k)\right\|_2^2 \le C_2(r_{nk} \vee q \vee f_n)/n.$$

Appendix D: Proof of Theorem 5.5

Let

$$g_n = C_2\left(r_{\max} \vee q \vee f_n\right)/n + 2c_1 C_2 \left\|\boldsymbol{\pi}\right\|_1 \sqrt{\left(r_{\max} \vee q \vee f_n\right)/n}.$$

We will first prove some lemmas, and then proceed to prove Theorem 5.5.

Lemma 1 *Suppose that there exists a positive constant ϕ_0 such that $\phi_k(\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_{-k}) \ge \phi_0$ for all k. If*

$$\sqrt{(r_{max} \vee q \vee f_n)/n} + c_1 \left\|\boldsymbol{\pi}\right\|_1 \le \sqrt{c_1^2 \left\|\boldsymbol{\pi}\right\|_1^2 + \phi_0^2/(64 C_2 |\mathcal{A}_k|)} \tag{25}$$

then, with probability at least $1 - e^{-(f_n - \log p)}$, we have $\phi_k(\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}) \ge \phi_0/2$.

Proof Note that the inequality (25) implies that $g_n \le \frac{\phi_0^2}{64|\mathcal{A}_k|}$. Then, for any index i and j, we first investigate the bound of

$$\begin{aligned} &(\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_i)^T (\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_j) - (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_i)^T (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_j) \\ &= \underbrace{(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)^T \mathbf{X}^T \mathbf{H}_k \mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)}_{T_7} + \underbrace{(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)^T \mathbf{X}^T \mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_j}_{T_8} + \underbrace{(\mathbf{X}\boldsymbol{\pi}_i)^T \mathbf{H}_k \mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)}_{T_9}. \end{aligned}$$

Note that $\lambda_{max}(\mathbf{H}_k) = 1$. By Theorem 5.4, we have, with probability at least $1 - e^{-f_n}$,

$$\begin{aligned} |T_7| &\le \left\|\mathbf{H}_k \mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\right\|_2 \times \left\|\mathbf{H}_k \mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\right\|_2 \\ &\le \lambda_{max}(\mathbf{H}_k) \times \left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\right\|_2 \times \left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\right\|_2 \le C_2\left(r_{\max} \vee q \vee f_n\right). \end{aligned} \tag{26}$$

Following that $\left\|\mathbf{X}\boldsymbol{\pi}_j\right\|_2 \le c_1\sqrt{n}\left\|\boldsymbol{\pi}_j\right\|_2$, we have,

$$\begin{aligned} |T_8| &\le \left\|\mathbf{X}\boldsymbol{\pi}_j\right\|_2 \times \left\|\mathbf{H}_k \mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\right\|_2 \le c_1\sqrt{n}\left\|\boldsymbol{\pi}_j\right\|_2 \times \left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\right\|_2 \\ &\le c_1 C_2 \left\|\boldsymbol{\pi}\right\|_1 \sqrt{n\left(r_{\max} \vee q \vee f_n\right)}. \end{aligned} \tag{27}$$

Similarly, we have,

$$|T_9| \le c_1\sqrt{n}\left\|\boldsymbol{\pi}_i\right\|_2 \left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\right\|_2 \le c_1 C_2 \left\|\boldsymbol{\pi}\right\|_1 \sqrt{n\left(r_{\max} \vee q \vee f_n\right)}. \tag{28}$$

Putting together the bounds in (26), (27), and (28), we have, with probability at least $1 - e^{-f_n}$,

$$|(\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_i)^T (\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_j) - (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_i)^T (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_j)| \le n g_n. \tag{29}$$

By definition, for any set $\mathcal{A}_k$ and any β, we have

$$\left\|\beta\right\|_1^2 \le \left(\left\|\beta_{\mathcal{A}_k^c}\right\|_1 + \left\|\beta_{\mathcal{A}_k}\right\|_1\right)^2 \le \left(3\sqrt{|\mathcal{A}_k|}\left\|\beta_{\mathcal{A}_k}\right\|_2 + \sqrt{|\mathcal{A}_k|}\left\|\beta_{\mathcal{A}_k}\right\|_2\right)^2 = 16|\mathcal{A}_k|\left\|\beta_{\mathcal{A}_k}\right\|_2^2.$$

We then have, with probability at least $1 - pe^{-f_n}$,

$$\begin{aligned} &|\beta^T((\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_{-k})^T (\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}) - (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_{-k})^T (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_{-k}))\beta| / (n\left\|\beta_{\mathcal{A}_k}\right\|_2^2) \\ &\le \left\|\beta\right\|_1^2 \left\|\beta_{\mathcal{A}_k}\right\|_2^{-2} \max_{i,j} |(\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_i)^T (\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_j) - (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_i)^T (\mathbf{H}_k \mathbf{X}\boldsymbol{\pi}_j)|/n \\ &\le 16|\mathcal{A}_k| \times g_n \le 16|\mathcal{A}_k| \times \phi_0^2/\left(64|\mathcal{A}_k|\right) = \phi_0^2/4, \end{aligned}$$

which implies that $\phi_k(\mathbf{H}_k \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}) \ge \phi_0/2$. ■

Lemma 2 *(Basic Inequality) Let random vector* $\boldsymbol{J}_k = 2n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\boldsymbol{\epsilon}_k - 2n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k} - \mathbf{Y}_{-k})\boldsymbol{\gamma}_k$ *and* $W_k^{-1} = diag(\boldsymbol{w}_k^{-1})$*, then, for the event*

$$\mathcal{J}_k(\lambda_k) = \left\{ \left\| W_k^{-1}\boldsymbol{J}_k \right\|_\infty \le \lambda_k/n \right\},$$

there exists a constant $C_3 > 0$ *such that*

$$P(\mathcal{J}_k(\lambda_k)) \ge 1 - e^{-C_3 h_n + \log(4pq)} - e^{-f_n + \log(p)}.$$

Furthermore, concurring with the random vector $\boldsymbol{J}_k$*, we have the following basic inequality,*

$$n^{-1}\left\| \mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}_k) \right\|_2^2 + 2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\hat{\boldsymbol{\gamma}}_k| \le 2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k| + \boldsymbol{J}_k^T|\hat{\boldsymbol{\gamma}}_k - \boldsymbol{\gamma}_k|. \tag{30}$$

Proof With $\mathbf{Y}_{-k} = \mathbf{X}\boldsymbol{\pi}_{-k} + \boldsymbol{\xi}_{-k}$ and $\hat{\mathbf{Z}}_{-k} = \mathbf{X}\hat{\boldsymbol{\pi}}_{-k}$, we have

$$\begin{aligned}
\boldsymbol{J}_k &= 2n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\boldsymbol{\epsilon}_k - 2n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k} - \mathbf{Y}_{-k})\boldsymbol{\gamma}_k \\
&= 2n^{-1}\hat{\boldsymbol{\pi}}_{-k}^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k - \frac{2}{n}\hat{\boldsymbol{\pi}}_{-k}^T\mathbf{X}^T\mathbf{H}_k(\mathbf{X}\hat{\boldsymbol{\pi}}_{-k} - \mathbf{X}\boldsymbol{\pi}_{-k} - \boldsymbol{\xi}_{-k})\boldsymbol{\gamma}_k \\
&= \underbrace{2n^{-1}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k}_{T_{10}} + \underbrace{2n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k}_{T_{11}} + \underbrace{2n^{-1}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\xi}_{-k}\boldsymbol{\gamma}_k}_{T_{12}} \\
&\quad + \underbrace{2n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\xi}_{-k}\boldsymbol{\gamma}_k}_{T_{13}} \underbrace{-2n^{-1}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\mathbf{X}^T\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})\boldsymbol{\gamma}_k}_{T_{14}} \\
&\quad \underbrace{-2n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})\boldsymbol{\gamma}_k}_{T_{15}}.
\end{aligned}$$

Denote $\mathbf{X} = (X_{\cdot 1}, X_{\cdot 2}, \ldots, X_{\cdot q})$, then $X_{\cdot j}^T X_{\cdot j} = n$ due to standardization. With $\sigma_{pmax}^2 = \max_{1\le k\le p}\sigma_k^2$, we have $\mathrm{Var}(X_{\cdot j}^T\mathbf{H}_k\boldsymbol{\epsilon}_k) = X_{\cdot j}^T\mathbf{H}_k X_{\cdot j}\sigma_k^2 \le n\sigma_k^2 \le n\sigma_{pmax}^2$. Further let, for some constant $t_\lambda > 0$,

$$\lambda_k = t_\lambda\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\Gamma}\|_1\|\boldsymbol{\pi}\|_1\sqrt{n(r_{\max}\vee q\vee f_n)\log p}.$$

By the Gaussian tail inequality, we have

$$\begin{aligned}
P\left(\left\|W_k^{-1}T_{10}\right\|_\infty \ge \lambda_k/(6n)\right) &\le P\left(\|T_{10}\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&= P\left(\left\|2n^{-1}(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\left\|(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\right\|_\infty \times \left\|2n^{-1}\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\left\|2n^{-1}\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n\delta_\pi)\right) \\
&\le q\exp\left\{-\lambda_k^2\|\omega_k\|_{-\infty}^2/(288n\sigma_{pmax}^2\delta_\pi^2)\right\} = q\cdot p^{-\frac{n}{q}t_3\|\Gamma\|_1^2\|\pi\|_1^2},
\end{aligned}$$

where $t_3 = t_\lambda^2/\left(288C_1\sigma_{pmax}^2\right)$ and

$$\delta_\pi = \max_k\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|_1 \le \max_k\sqrt{q}\,\|\hat{\boldsymbol{\pi}}_k - \boldsymbol{\pi}_k\|_2 = \sqrt{C_1 q(r_{\max}\vee q\vee f_n)/n}.$$

Similarly, letting $t_4 = t_\lambda/\left(288\sigma_{pmax}^2\right)$, we have

$$\begin{aligned}
P\left(\left\|W_k^{-1}T_{11}\right\|_\infty \ge \lambda_k/(6n)\right) &\le P\left(\|T_{11}\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&= P\left(\left\|2n^{-1}\boldsymbol{\pi}_{-k}^T\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le \mathbb{P}\left(\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty\left\|2n^{-1}\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\left\|2n^{-1}\mathbf{X}^T\mathbf{H}_k\boldsymbol{\epsilon}_k\right\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty^{-1}/(6n)\right) \\
&\le q\exp\left\{-\lambda_k^2\|\omega_k\|_{-\infty}^2\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty^{-2}/(288n\sigma_{pmax}^2)\right\} \\
&= q\cdot p^{-t_4\|\Gamma\|_1^2(r_{\max}\vee q\vee f_n)}.
\end{aligned}$$

Let $\tilde{\sigma}_{pmax}^2 = \max_k\mathrm{Var}(\boldsymbol{\xi}_k)$ and $t_5 = t_\lambda/\left(288C_1\tilde{\sigma}_{pmax}^2\right)$. For the term T_{12}, we have

$$\begin{aligned}
P\left(\left\|W_k^{-1}T_{12}\right\|_\infty \ge \lambda_k/(6n)\right) &\le P\left(\|T_{12}\|_\infty \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\left\|(\hat{\boldsymbol{\pi}}_{-k} - \boldsymbol{\pi}_{-k})^T\right\|_\infty\left\|2n^{-1}\mathbf{X}^T\mathbf{H}_k\boldsymbol{\xi}_{-k}\boldsymbol{\gamma}_k\right\|_1 \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\delta_\pi\max_{i,j}\left|2n^{-1}\mathbf{x}_i^T\mathbf{H}_k\boldsymbol{\xi}_j\right|\|\boldsymbol{\gamma}_k\|_1 \ge \lambda_k\|\omega_k\|_{-\infty}/(6n)\right) \\
&\le P\left(\max_{i,j}\left|2n^{-1}\mathbf{x}_i^T\mathbf{H}_k\boldsymbol{\xi}_j\right| \ge \lambda_k\|\omega_k\|_{-\infty}\|\boldsymbol{\gamma}_k\|_1^{-1}/(6n\delta_\pi)\right) \\
&\le qp\exp\left\{-\lambda_k^2\|\omega_k\|_{-\infty}^2\tilde{\sigma}_{pmax}^{-2}\delta_\pi^{-2}\|\boldsymbol{\gamma}_k\|_1^{-2}/(288n)\right\} = qp^{1-t_5\|\pi\|_1^2 n/q}.
\end{aligned}$$

Letting $t_6 = t_\lambda/\left(288\tilde{\sigma}_{pmax}^2\right)$, we similarly have

$$\begin{aligned}
&P\left(\left\|W_k^{-1}T_{13}\right\|_\infty \ge \lambda_k/(6n)\right) \\
&\le qp\exp\left\{-\lambda_k^2\tilde{\sigma}_{pmax}^{-2}\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty^{-2}\|\boldsymbol{\gamma}_k\|_1^{-2}/(288n)\right\} = qp^{1-t_6(r_{max}\vee q\vee f_n)}.
\end{aligned}$$

When t_λ is sufficiently large, say $t_\lambda \ge 6C_2\|\boldsymbol{\pi}\|_1^{-1}\sqrt{(r_{max}\vee q\vee f_n)/(n\log p)}$, we have

$$\begin{aligned}
\left\|W_k^{-1}T_{14}\right\|_\infty &\le n^{-1}\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\gamma}_k\|_1\max_{i,j}|(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)^T\mathbf{X}^T\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)| \\
&\le n^{-1}\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\gamma}_k\|_1\max_{i,j}\|\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\|_2\|\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\|_2 \\
&\le n^{-1}\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\gamma}_k\|_1\max_{i,j}\lambda_{\max}(\mathbf{H}_k)\|\mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\|_2\|\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\|_2 \\
&\le n^{-1}\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\gamma}_k\|_1\max_{i,j}\|\mathbf{X}(\hat{\boldsymbol{\pi}}_i - \boldsymbol{\pi}_i)\|_2\|\mathbf{X}(\hat{\boldsymbol{\pi}}_j - \boldsymbol{\pi}_j)\|_2 \\
&\le C_2\|\omega_k\|_{-\infty}^{-1}\|\boldsymbol{\gamma}_k\|_1 n^{-1}(r_{\max}\vee q\vee f_n) \\
&\le \left\{\lambda_k/(6n)\right\}\times\left\{6C_2 t_\lambda^{-1}\|\boldsymbol{\pi}\|_1^{-1}\sqrt{n^{-1}(\log p)^{-1}(r_{max}\vee q\vee f_n)}\right\} \le \lambda_k/(6n).
\end{aligned}$$

Similarly, when $t_\lambda \geq 12\sqrt{C_2/\log p}$,

$$\begin{aligned}
\left\|W_k^{-1}T_{15}\right\|_\infty &\leq 2n^{-1}\left\|\gamma_k\right\|_1\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty\left\|\omega_k\right\|_{-\infty}^{-1}\max_{i,j}|X_{\cdot i}^T\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_j-\boldsymbol{\pi}_j)| \\
&\leq 2n^{-1/2}\left\|\gamma_k\right\|_1\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty\left\|\omega_k\right\|_{-\infty}^{-1}\max_j\left\|\mathbf{H}_k\mathbf{X}(\hat{\boldsymbol{\pi}}_j-\boldsymbol{\pi}_j)\right\|_2 \\
&\leq 2n^{-1/2}\left\|\gamma_k\right\|_1\left\|\boldsymbol{\pi}_{-k}^T\right\|_\infty\left\|\omega_k\right\|_{-\infty}^{-1}\max_j\left\|\mathbf{X}(\hat{\boldsymbol{\pi}}_j-\boldsymbol{\pi}_j)\right\|_2 \\
&\leq \left\{\lambda_k/(6n)\right\}\times\left\{12t_\lambda^{-1}\sqrt{C_2/\log p}\right\}\leq\lambda_k/(6n).
\end{aligned}$$

Putting together all the above results, we have, for some constant $C_3>0$,

$$P(\mathcal{J}_k(\lambda_k))\geq 1-e^{-C_3h_n+\log(4pq)}-e^{-f_n+\log(p)}.$$

Concurring with the random vector $\boldsymbol{J}_k$, we have the following inequality based on the optimality of $\hat{\boldsymbol{\gamma}}_k$,

$$\left\|\mathbf{H}_k\mathbf{Y}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k\right\|_2+2\lambda_k\boldsymbol{\omega}_k^T|\hat{\boldsymbol{\gamma}}_k|\leq\left\|\mathbf{H}_k\mathbf{Y}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k\right\|_2+2\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|. \tag{31}$$

With $\mathbf{H}_k\mathbf{Y}_k=\mathbf{H}_k\mathbf{Y}_{-k}\boldsymbol{\gamma}_k+\mathbf{H}_k\boldsymbol{\epsilon}_k$, we also have

$$\begin{aligned}
&\left\|\mathbf{H}_k\mathbf{Y}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k\right\|_2^2 \\
&= \left\|\mathbf{H}_k\mathbf{Y}_{-k}\boldsymbol{\gamma}_k+\mathbf{H}_k\boldsymbol{\epsilon}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k\right\|_2^2 \\
&= \|\mathbf{H}_k\boldsymbol{\epsilon}_k\|_2^2-2\boldsymbol{\epsilon}_k^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k-\mathbf{Y}_{-k}\boldsymbol{\gamma}_k)+\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k-\mathbf{H}_k\mathbf{Y}_{-k}\boldsymbol{\gamma}_k\right\|_2^2 \\
&= \|\mathbf{H}_k\boldsymbol{\epsilon}_k\|_2^2-2\boldsymbol{\epsilon}_k^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}\hat{\boldsymbol{\gamma}}_k-\mathbf{Y}_{-k}\boldsymbol{\gamma}_k)+\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2^2 \\
&\quad+\left\|\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}-\mathbf{Y}_{-k})\boldsymbol{\gamma}_k\right\|_2^2+2\boldsymbol{\gamma}_k^T(\hat{\mathbf{Z}}_{-k}-\mathbf{Y}_{-k})^T\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k), \qquad (32)
\end{aligned}$$

$$\begin{aligned}
&\left\|\mathbf{H}_k\mathbf{Y}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k\right\|_2^2 \\
&= \left\|\mathbf{H}_k\mathbf{Y}_{-k}\boldsymbol{\gamma}_k+\mathbf{H}_k\boldsymbol{\epsilon}_k-\mathbf{H}_k\hat{\mathbf{Z}}_{-k}\boldsymbol{\gamma}_k\right\|_2^2 \\
&= \|\mathbf{H}_k\boldsymbol{\epsilon}_k\|_2^2+\left\|\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}-\mathbf{Y}_{-k})\boldsymbol{\gamma}_k\right\|_2^2-2\boldsymbol{\epsilon}_k^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}-\mathbf{Y}_{-k})\boldsymbol{\gamma}_k. \qquad (33)
\end{aligned}$$

Combining the equations (31), (32), and (33), we obtain that

$$\begin{aligned}
&n^{-1}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2^2+2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\hat{\boldsymbol{\gamma}}_k| \\
&\leq 2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|+\left(\frac{2}{n}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k\boldsymbol{\epsilon}_k-2n^{-1}\hat{\mathbf{Z}}_{-k}^T\mathbf{H}_k(\hat{\mathbf{Z}}_{-k}-\mathbf{Y}_{-k})\boldsymbol{\gamma}_k\right)^T(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k) \\
&= 2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|+\boldsymbol{J}_k^T(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k),
\end{aligned}$$

which concludes the proof. ■

By the basic inequality we just proved above and condition on the event $\mathcal{J}_k(\lambda_k)$, we have that

$$\begin{aligned}
n^{-1}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2^2 &\leq 2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\boldsymbol{\gamma}_k|-2n^{-1}\lambda_k\boldsymbol{\omega}_k^T|\hat{\boldsymbol{\gamma}}_k|+\boldsymbol{J}_k^T(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k) \\
&\leq 2n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k}^T|\boldsymbol{\gamma}_{k,\mathcal{A}_k}|-2n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}|-2n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k^c}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}| \\
&\quad+\boldsymbol{J}_{k,\mathcal{A}_k^c}^T(\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c})+\boldsymbol{J}_{k,\mathcal{A}_k}^T(\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}) \\
&\leq 2n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}|-2n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k^c}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}| \\
&\quad+n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k^c}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}|+n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}| \\
&\leq 3n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}|-n^{-1}\lambda_k\boldsymbol{\omega}_{k,\mathcal{A}_k^c}^T|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}| \\
&\leq 3n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k}\|_\infty\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_1-n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}\right\|_1,
\end{aligned}$$

which implies that

$$n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}\right\|_1\leq 3n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k}\|_\infty\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_1. \tag{34}$$

Note that $\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}\leq 1$, we have that

$$\begin{aligned}
&\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k^c}-\boldsymbol{\gamma}_{k,\mathcal{A}_k^c}\right\|_1 \\
&\leq 3\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_1\leq 3\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_1. \qquad (35)
\end{aligned}$$

On the other hand, following Lemma 1, we have, with $C_4=6t_\lambda$,

$$\begin{aligned}
n^{-1}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2^2 &\leq 3n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k}\|_\infty\sqrt{|\mathcal{A}_k|}\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_2 \\
&\leq 3n^{-1}\lambda_k\|\omega_{k,\mathcal{A}_k}\|_\infty\sqrt{|\mathcal{A}_k|}\times 2n^{-1/2}\phi_0^{-1}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2 \\
&\leq 36n^{-2}\phi_0^{-2}\|\omega_{k,\mathcal{A}_k}\|_\infty^2|\mathcal{A}_k|\lambda_k^2 \\
&= C_4^2\phi_0^{-2}\|\omega_k\|_{-\infty}^{-2}\|\omega_{k,\mathcal{A}_k}\|_\infty^2\left\|\boldsymbol{\pi}\right\|_1^2\left\|\boldsymbol{\Gamma}\right\|_1^2|\mathcal{A}_k|(r_{\max}\vee q\vee f_n)\log p/n.
\end{aligned}$$

Employing the inequality (34), along with $\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}\leq 1$, we have

$$\begin{aligned}
\|\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k\|_1 &\leq \left(3\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}+1\right)\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_1 \\
&\leq \left(3\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}+1\right)\sqrt{|\mathcal{A}_k|}\left\|\hat{\boldsymbol{\gamma}}_{k,\mathcal{A}_k}-\boldsymbol{\gamma}_{k,\mathcal{A}_k}\right\|_2 \\
&\leq \left(6\|\omega_{k,\mathcal{A}_k}\|_\infty\|\omega_{k,\mathcal{A}_k^c}\|_{-\infty}^{-1}+2\right)\sqrt{|\mathcal{A}_k|}\times n^{-1/2}\left\|\mathbf{H}_k\hat{\mathbf{Z}}_{-k}(\hat{\boldsymbol{\gamma}}_k-\boldsymbol{\gamma}_k)\right\|_2\phi_0^{-1} \\
&\leq 8C_4\times\|\omega_{k,\mathcal{A}_k}\|_\infty\left\|\boldsymbol{\pi}\right\|_1\left\|\boldsymbol{\Gamma}\right\|_1\phi_0^{-2}\|\omega_k\|_{-\infty}^{-1} \\
&\quad\times|\mathcal{A}_k|\sqrt{(r_{\max}\vee q\vee f_n)\log p/n}.
\end{aligned}$$

Since we condition on event $\mathcal{J}_k(\lambda_k)$, the above prediction and estimation bounds hold with probability at least $1-e^{-C_3h_n+\log(4pq)}-e^{-f_n+\log(p)}$.

Appendix E: Proof of Theorem 5.6

Denote $\hat{V}_k = (\hat{v}_{ij})_{(p-1)\times(p-1)} \triangleq n^{-1}\hat{\boldsymbol{\pi}}_{-k}^T \mathbf{X}^T \mathbf{H}_k \mathbf{X} \hat{\boldsymbol{\pi}}_{-k}$, $\hat{V}_{k,21} = (\hat{v}_{ij})_{i\in\mathcal{A}_k^c, j\in\mathcal{A}_k}$, and $\hat{V}_{k,11} = (\hat{v}_{ij})_{i\in\mathcal{A}_k, j\in\mathcal{A}_k}$. The proof of Theorem 5.6 will be presented after the following lemma.

Lemma 3 *Assume that, for each node i, the following inequality holds.*

$$\begin{aligned}&\sqrt{(r_{max} \vee q \vee f_n)/n} + c_1 \|\boldsymbol{\pi}\|_1 \\ \leq\ & \sqrt{c_1^2 \|\boldsymbol{\pi}\|_1^2 + \min(\phi_0^2/64, \zeta(4-\zeta)^{-1}\|\boldsymbol{\omega}_k\|_{-\infty}/\theta_i)/(C_2|\mathcal{A}_k|)}. \end{aligned} \tag{36}$$

Under the assumptions and conditions of Theorem 5.6, we have that, with probability at least $1 - pe^{-f_n}$,

$$\left\| W_{k,\mathcal{A}_k^c}^{-1} \left(\hat{V}_{k,21} \hat{V}_{k,11}^{-1} \right) W_{k,\mathcal{A}_k} \right\|_\infty \leq 1 - \zeta/2.$$

Proof Following Theorem 5.4, we have, with probability at least $1 - pe^{-f_n}$,

$$n^{-1} \max_{i,j} \left| (\mathbf{H}_k \mathbf{X} \hat{\boldsymbol{\pi}}_i)^T (\mathbf{H}_k \mathbf{X} \hat{\boldsymbol{\pi}}_j) - (\mathbf{H}_k \mathbf{X} \boldsymbol{\pi}_i)^T (\mathbf{H}_k \mathbf{X} \boldsymbol{\pi}_j) \right| \leq g_n.$$

The inequality (36) implies that $\theta_k \|\omega_{k,\mathcal{A}_k}\|_{-\infty}^{-1} |\mathcal{A}_k| g_n \leq \zeta/(4-\zeta)$, we have

$$\left\| W_{k,\mathcal{A}_k}^{-1} (\hat{V}_{k,11} - V_{k,11}) \right\|_\infty \leq \|\omega_{k,\mathcal{A}_k}\|_{-\infty}^{-1} |\mathcal{A}_k| g_n \leq \zeta / \{(4-\zeta)\theta_k\}.$$

Similarly we have that

$$\left\| W_{k,\mathcal{A}_k^c}^{-1} (\hat{V}_{k,21} - V_{k,21}) \right\|_\infty \leq \zeta / \{(4-\zeta)\theta_k\}.$$

Applying the matrix inversion error bound in Horn and Johnson (2012), we obtain

$$\begin{aligned} &\left\| \hat{V}_{k,11}^{-1} W_{k,\mathcal{A}_k} \right\|_\infty \leq \left\| V_{k,11}^{-1} W_{k,\mathcal{A}_k} \right\|_\infty + \left\| \hat{V}_{k,11}^{-1} W_{k,\mathcal{A}_k} - V_{k,11}^{-1} W_{k,\mathcal{A}_k} \right\|_\infty \\ \leq\ & \theta_k + \theta_k \left\| W_{k,\mathcal{A}_k}^{-1} (\hat{V}_{k,11} - V_{k,11}) \right\|_\infty \left(1 - \theta_k \left\| W_{k,\mathcal{A}_k}^{-1} (\hat{V}_{k,11} - V_{k,11}) \right\|_\infty \right)^{-1} \theta_k \\ \leq\ & \theta_k (4-\zeta)/(4-2\zeta). \end{aligned}$$

Therefore,

$$\begin{aligned} &\left\| W_{k,\mathcal{A}_k^c}^{-1} \left(\hat{V}_{k,21} \hat{V}_{k,11}^{-1} - V_{k,21} V_{k,11}^{-1} \right) W_{k,\mathcal{A}_k} \right\|_\infty \\ \leq\ & \left\| W_{k,\mathcal{A}_k^c}^{-1} \left(\hat{V}_{k,21} - V_{k,21} \right) (\hat{V}_{k,11}^{-1}) W_{k,\mathcal{A}_k} \right\|_\infty \\ & + \left\| W_{k,\mathcal{A}_k^c}^{-1} V_{k,21} V_{k,11}^{-1} W_{k,\mathcal{A}_k} W_{k,\mathcal{A}_k}^{-1} \left(\hat{V}_{k,11} - V_{k,11} \right) (\hat{V}_{k,11}^{-1}) W_{k,\mathcal{A}_k} \right\|_\infty \\ \leq\ & \left\| W_{k,\mathcal{A}_k^c}^{-1} \left(\hat{V}_{k,21} - V_{k,21} \right) \right\|_\infty \left\| (\hat{V}_{k,11}^{-1}) W_{k,\mathcal{A}_k} \right\|_\infty \\ & + \left\| W_{k,\mathcal{A}_k^c}^{-1} V_{k,21} V_{k,11}^{-1} W_{k,\mathcal{A}_k} \right\|_\infty \left\| W_{k,\mathcal{A}_k}^{-1} \left(\hat{V}_{k,11} - V_{k,11} \right) \right\|_\infty \left\| (\hat{V}_{k,11}^{-1}) W_{k,\mathcal{A}_k} \right\|_\infty \\ \leq\ & \zeta/2, \end{aligned}$$

which implies that $\left\| W_{k,\mathcal{A}_k^c}^{-1} (\hat{V}_{k,21} \hat{V}_{k,11}^{-1}) W_{k,\mathcal{A}_k} \right\|_\infty \leq 1 - \zeta/2.$ ■

By the optimality of $\hat{\gamma}_k$, it must satisfy the KKT condition as follows,

$$-2n^{-1}(\mathbf{H}_k \hat{\mathbf{Z}}_{-k})^T (\mathbf{H}_k \mathbf{Y}_k - \mathbf{H}_k \hat{\mathbf{Z}}_{-k} \hat{\gamma}_k) + 2n^{-1} \lambda_k W_k \alpha_k = 0, \tag{37}$$

where $\|\alpha_k\|_\infty \leq 1$ and $\alpha_{kj} I[\hat{\gamma}_{kj} \neq 0] = sign(\hat{\gamma}_{kj})$. Plug in the equation $\mathbf{H}_k \mathbf{Y}_k = \mathbf{H}_k \mathbf{Y}_{-k} \boldsymbol{\gamma}_k + \mathbf{H}_k \epsilon_k$, we can have that

$$\begin{aligned} \mathbf{H}_k \mathbf{Y}_k - \mathbf{H}_k \hat{\mathbf{Z}}_{-k} \hat{\gamma}_k &= \mathbf{H} \mathbf{Y}_{-k} \gamma_k + \mathbf{H}_k \epsilon_k - \mathbf{H}_k \hat{\mathbf{Z}}_{-k} \hat{\gamma}_k \\ &= \mathbf{H}_k \epsilon_k + \mathbf{H}_k \mathbf{Y}_{-k} \gamma_k - -\mathbf{H}_k \hat{\mathbf{Z}}_{-k} \gamma_k + \mathbf{H}_k \hat{\mathbf{Z}}_{-k} \gamma_k - \mathbf{H}_k \hat{\mathbf{Z}}_{-k} \hat{\gamma}_k \\ &= \mathbf{H}_k \epsilon_k - \mathbf{H}_k (\hat{\mathbf{Z}}_{-k} - \mathbf{Y}_{-k}) \gamma_k - \mathbf{H}_k \hat{\mathbf{Z}}_{-k} (\hat{\gamma}_k - \gamma_k). \end{aligned} \tag{38}$$

Combining (37) and (38), we can get that

$$2\hat{V}_k (\hat{\gamma}_k - \gamma_k) - \boldsymbol{J}_k = -2\lambda_k W_k \alpha_k / n, \tag{39}$$

where $\boldsymbol{J}_k = 2n^{-1} \hat{\mathbf{Z}}_{-k}^T \mathbf{H}_k \boldsymbol{\epsilon}_k - 2n^{-1} \hat{\mathbf{Z}}_{-k}^T \mathbf{H}_k (\hat{\mathbf{Z}}_{-k} - \mathbf{Y}_{-k}) \boldsymbol{\gamma}_k$. For an estimator satisfying $\hat{\gamma}_{k,\mathcal{A}_k^c} = \gamma_{k,\mathcal{A}_k^c} = 0$, the above equation implies that

$$\begin{cases} 2\hat{V}_{k,11} (\hat{\gamma}_{k,\mathcal{A}_k} - \gamma_{k,\mathcal{A}_k}) - \boldsymbol{J}_{k,\mathcal{A}_k} = -\lambda_k W_{k,\mathcal{A}_k} \alpha_{k,\mathcal{A}_k} / n, \\ 2\hat{V}_{k,21} (\hat{\gamma}_{k,\mathcal{A}_k} - \gamma_{k,\mathcal{A}_k}) - \boldsymbol{J}_{k,\mathcal{A}_k^c} = -\lambda_k W_{k,\mathcal{A}_k^c} \alpha_{k,\mathcal{A}_k^c} / n. \end{cases} \tag{40}$$

Manipulating the above equations, we have that

$$\begin{aligned} \hat{\gamma}_{k,\mathcal{A}_k} - \gamma_{k,\mathcal{A}_k} &= 2^{-1} \hat{V}_{k,11}^{-1} (\boldsymbol{J}_{k,\mathcal{A}_k} - \lambda_k W_{k,\mathcal{A}_k}^T \alpha_{k,\mathcal{A}_k}) \\ &= 2^{-1} \hat{V}_{k,11}^{-1} W_{k,\mathcal{A}_k} (W_{k,\mathcal{A}_k}^{-1} \boldsymbol{J}_{k,\mathcal{A}_k} - \lambda_k \alpha_{k,\mathcal{A}_k}). \end{aligned} \tag{41}$$

Following the similar strategy in proving Lemma 2, we can prove that there exists a constant $C_5 > 0$ such that $\left\| W_k^{-1} \boldsymbol{J}_k \right\|_\infty \leq 2\lambda_k \zeta / \{n(4-\zeta)\}$ with probability at least $1 - e^{-C_5 h_n + \log(4pq)} - e^{-f_n + \log(p)}$. Therefore, with $\|\alpha_{k,\mathcal{A}_k}\|_\infty \leq 1$, we have that

$$\begin{aligned} &\|\hat{\gamma}_{k,\mathcal{A}_k} - \gamma_{k,\mathcal{A}_k}\|_\infty \leq 2^{-1} \left\| \hat{V}_{k,11}^{-1} W_{k,\mathcal{A}_k} \right\|_\infty \left(\left\| W_{k,\mathcal{A}_k}^{-1} \boldsymbol{J}_{k,\mathcal{A}_k} \right\|_\infty + 2n^{-1}\lambda_k \right) \\ \leq\ & \{\theta_k (4-\zeta)/(2-\zeta)\} \times \{4/(4-\zeta)\} \times \{2\lambda_k/n\} = 2\lambda_k \theta_k / \{n(2-\zeta)\} \leq \min_{j\in\mathcal{A}_k} |\gamma_{kj}|. \end{aligned}$$

The above inequality implies that $sign(\hat{\gamma}_{k,\mathcal{A}_k}) = sign(\gamma_{k,\mathcal{A}_k})$.

Combining (40) and (41), we can also verify that

$$\begin{aligned} &\left\| W_{k,\mathcal{A}_k^c}^{-1} \hat{V}_{k,21} (\hat{V}_{k,11})^{-1} (\boldsymbol{J}_{k,\mathcal{A}_k} - 2\lambda_k W_{k,\mathcal{A}_k} \alpha_{k,\mathcal{A}_k} / n) - W_{k,\mathcal{A}_k^c}^{-1} J_{k,\mathcal{A}_k^c} \right\|_\infty \\ \leq\ & \left\| W_{k,\mathcal{A}_k^c}^{-1} \hat{V}_{k,21} (\hat{V}_{k,11})^{-1} W_{k,\mathcal{A}_k} \right\|_\infty \left(\left\| W_{k,\mathcal{A}_k}^{-1} \boldsymbol{J}_k \right\|_\infty + 2\lambda_k / n \right) + \left\| W_{k,\mathcal{A}_k^c}^{-1} J_{k,\mathcal{A}_k^c} \right\|_\infty \\ \leq\ & (1 - \zeta/2)(4/(4-\zeta)) 2\lambda_k/n + \zeta/(4-\zeta) 2\lambda_k/n = 2\lambda_k/n. \end{aligned}$$

Therefore, there exists an estimator $\hat{\gamma}_k$ satisfying the KKT condition (39) as well as $sign(\hat{\gamma}_k) = sign(\gamma_k)$ which implies $\hat{\mathcal{A}}_k = \mathcal{A}_k$.

References

Hirotugu Akaike. A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6): 716-723, 1974.

Theodore W. Anderson and Herman Rubin. Estimation of the parameters of a single equation in a complete system of stochastic equations. *The Annals of Mathematical Statistics*, 20(1): 46-63, 1949.

Jason E. Aten, Tova F. Fuller, Aldons J. Lusis, and Steve Horvath. Using genetic markers to orient the edges in quantitative trait networks: The NEO software. *BMC Systems Biology*, 2: 34, 2008.

Robert L. Basmann. A generalized classical method of linear estimation of coefficients in a structural equation. *Econometrica*, 25(1): 77-83, 1957.

Alexandre Belloni, Daniel Chen, Victor Chernozhukov, and Christian Hansen. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrics*, 80(6): 2369-2429, 2012.

Kenneth A. Bollen. An alternative two stage least squares (2SLS) estimator for latent variable equations. *Psychometrika*, 61(1): 109-121, 1996.

Rachel B. Brem and Leonid Kruglyak. The landscape of genetic complexity across 5,700 gene expression traits in yeast. *Proceedings of the National Academy of Sciences of the United States of America*, 102(5): 1572-1577, 2005.

Karl W. Broman and Terence P. Speed. A model selection approach for the identification of quantitative trait loci in experimental crosses. *Journal of the Royal Statistical Society, Series B*, 64(4): 641-656, 2002.

Xiaodong Cai, Juan Andrés Bazerque, and Georgios B. Giannakis. Inference of gene regulatory networks with sparse structural equation models exploiting genetic perturbations. *PLoS Computational Biology*, 9(5): e1003068, 2013.

Alberto de la Fuente, Nan Bing, Ina Hoeschele, and Pedro Mendes. Discovery of meaningful associations in genomic data using partial correlation coefficients. *Bioinformatics*, 20(18): 3565-3574, 2004.

Anna L. Dixon, Liming Liang, Miriam F. Moffatt, Wei Chen, Simon Heath, Kenny C. C. Wong, Jenny Taylor, Edward Burnett, Ivo Gut, Martin Farrall, G Mark Lathrop, Gonçalo R. Abecasis, and William O. C. Cookson. A genome-wide association study of global gene expression. *Nature Genetics*, 39(10): 1202-1207, 2007.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456): 1348-1360, 2001.

Ildiko E. Frank and Jerome H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35(2): 109-135, 1993.

Wenjiang Fu and Keith Knight. Asymptotics for lasso-type estimators. *The Annals of Statistics*, 28(5): 1356-1378, 2000.

Jonathan A. L. Gelfond, Joseph G. Ibrahim, and Fei Zou. Proximity model for expression quantitative trait loci (eQTL) detection. *Biometrics*, 63(4): 1108-1116, 2007.

Charles J. Geyer. On the asymptotics of constrained M-estimation. *The Annals of Statistics*, 22(4): 1993-2010, 1994.

Gene H. Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2): 215-223, 1979.

Trygve Haavelmo. The statistical implications of a system of simultaneous equations. *Econometrica*, 11(1): 1-12, 1943.

Trygve Haavelmo. The probability approach in econometrics. *Econometrica*, 12: S1-S115, 1944.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

Da Wei Huang, Brad T. Sherman, and Richard A. Lempicki. Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists. *Nucleic Acids Research*, 37(1): 1-13, 2009.

Jian Huang, Shuangge Ma, Hongzhe Li, and Cun-Hui Zhang. The sparse Laplacian shrinkage estimator for high-dimensional regression. *The Annals of Statistics*, 39(4): 2021-2046, 2011.

Ritsert C. Jansen and Jan-Peter Nap. Genetical genomics: the added value from segregation. *Trends in Genetics*, 17(7): 388-391, 2001.

Zhenyu Jia and Shizhong Xu. Mapping quantitative trait loci for expression abundance. *Genetics*, 176(1): 611-623, 2007.

Christina Kendziorski, Meng Chen, Ming Yuan, Hong Lan, and Alan D. Attie. Statistical methods for expression quantitative trait loci (eQTL) mapping. *Biometrics*, 62(1): 19-27, 2006.

Peter E. Kennedy. *A Guide to Econometrics*. Cambridge, MA: MIT Press, 1985.

Wei Lin, Rui Feng, and Hongzhe Li. Regularization methods for high-dimensional instrumental variables regression with an application to genetical genomics. *Journal of the American Statistical Association*, 110(509): 270-288, 2015.

Bing Liu, Alberto de la Fuente, and Ina Hoeschele. Gene network inference via structural equation modeling in genetical genomics experiments. *Genetics*, 178(3): 1763-1776, 2008.

Benjamin A. Logsdon and Jason Mezey. Gene expression network reconstruction by convex feature selection when incorporating genetic perturbations. *PLoS Computational Biology*, 6(12): e1001014, 2010.

Elias Chaibub Neto, Christine T. Ferrara, Alan D. Attie, and Brian S. Yandell. Inferring causal phenotype networks from segregating populations. *Genetics*, 179(2): 1089-1100, 2008.

Olav Reiersøl. Confluence analysis by means of lag moments and other methods of confluence analysis. *Econometrica*, 9(1): 1-24, 1941.

Olav Reiersøl. Confluence analysis by means of instrumental sets of variables. *Arkiv for Mathematik, Astronomi och Fysik*, 32A(4), 1945.

Mark Rudelson and Roman Vershyninand. Hanson-Wright inequality and sub-gaussian concentration. *Electronic Communications in Probability*, 18: 1-9, 2013.

Eric E. Schadt, Stephanie A. Monks, Thomas A. Drake, Aldons J. Lusis, Nam Che, Veronica Colinayo, Thomas G. Ruff, Stephen B. Milligan, John R. Lamb, Guy Cavet, Peter S. Linsley, Mao Mao, Roland B. Stoughton, and Stephen H. Friend. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422: 297-302, 2003.

Peter Schmidt. *Econometrics*. New York: Marcel Dekker, 1976.

Gideon Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2): 461-464, 1978.

Bill Shipley. *Cause and Correlation in Biology: A User's Guide to Path Analysis, Structural Equations and Causal Inference*. New York: Cambridge University Press, 2002.

Peter Spirtes, Clark Glymour, and Richard Scheines. *Causation, Prediction, and Search*. Boston: The MIT Press, 2001.

Henri Theil. Repeated least-squares applied to complete equation systems. *Mimeo. The Hague: Central Planning Bureau.*, 1953a.

Henri Theil. Estimating and simultaneous correlation in complete equation systems. *Mimeo. The Hague: Central Planning Bureau.*, 1953b.

Henri Theil. *Economic Forecasts and Policy*. Amsterdam: North Holland, 1961.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B*, 58(1): 267-288, 1996.

Momiao Xiong, Jun Li, and Xiangzhong Fang. Identification of genetic networks. *Genetics*, 166(2): 1037-1052, 2004.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of Statistics*, 38(2): 894-942, 2010.

Ying Zhu. Sparse linear models and l_1-regularized 2SLS with high-dimensional endogenous regressors and instruments. *MPRA* Paper No. 65703, 2015.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476): 1418-1429, 2006.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67(2): 301-320, 2005.

Journal of Machine Learning Research 19 (2018) 1–34 Submitted 10/16; Revised 8/18; Published 8/18

Approximate Submodularity and its Applications: Subset Selection, Sparse Approximation and Dictionary Selection*

Abhimanyu Das†
ABHI.DAS@GMAIL.COM
Google

David Kempe‡
DAVID.M.KEMPE@GMAIL.COM
Department of Computer Science
University of Southern California

Editor: Jeff Bilmes

Abstract

We introduce the *submodularity ratio* as a measure of how "close" to submodular a set function f is. We show that when f has submodularity ratio γ, the greedy algorithm for maximizing f provides a $(1 - e^{-\gamma})$-approximation. Furthermore, when γ is bounded away from 0, the greedy algorithm for minimum submodular cover also provides essentially an $O(\log n)$ approximation for a universe of n elements.

As a main application of this framework, we study the problem of selecting a subset of k random variables from a large set, in order to obtain the best linear prediction of another variable of interest. We analyze the performance of widely used greedy heuristics; in particular, by showing that the submodularity ratio is lower-bounded by the smallest $2k$-sparse eigenvalue of the covariance matrix, we obtain the strongest known approximation guarantees for the Forward Regression and Orthogonal Matching Pursuit algorithms.

As a second application, we analyze greedy algorithms for the dictionary selection problem, and significantly improve the previously known guarantees. Our theoretical analysis is complemented by experiments on real-world and synthetic data sets; in particular, we focus on an analysis of how tight various spectral parameters and the submodularity ratio are in terms of predicting the performance of the greedy algorithms.

1. Introduction

Over the past 10–15 years, submodularity has established itself as one of the workhorses of the Machine Learning community. A function f mapping sets to real numbers is called *submodular* if $f(S \cup \{v\}) - f(S) \geq f(T \cup \{v\}) - f(T)$ whenever $S \subseteq T$. One of the most popular consequences of submodularity is that greedy algorithms perform quite well for maximizing the function subject to a cardinality constraint. Specifically, suppose that f is non-negative, monotone, and submodular, and consider the algorithm that, for k iterations, adds the element x_{i+1} that has largest marginal gain $f(S_i \cup \{x_{i+1}\}) - f(S_i)$ with respect to the current set S_i. By a classic result of Nemhauser et al. (1978), this algorithm guarantees that the final set achieves a function value within a factor $1 - 1/e$ of the optimum set S^* of cardinality k.

This approximation guarantee has been applied in a large number of settings; see, e.g., a survey in (Krause and Golovin, 2014). Of course, greedy algorithms are also popular when the objective function is not submodular. Typically, when f is not submodular, the greedy algorithm, though perhaps still useful in practice, will not provide theoretical performance guarantees. However, one might suspect that when f is "close to" submodular, then the performance of the greedy algorithm should degrade gracefully.

In the present article (Section 2), we formalize this intuition by defining a measure of "approximate submodularity" which we term *submodularity ratio*, and denote by γ. We prove that when a function f has submodularity ratio γ, the greedy algorithm gives a $(1 - e^{-\gamma})$-approximation; in particular, whenever γ is bounded away from 0, the greedy algorithm guarantees a solution within a constant factor of optimal. We also show that for the complementary *Minimum Submodular Cover* problem, where the goal is to find the smallest set S with $f(S) \geq C$ for a given value C, the greedy algorithm gives essentially an $O(\log n)$ approximation when γ is bounded away from 0.

Subset Selection for Regression. To illustrate the usefulness of the approximate submodularity framework, we analyze greedy algorithms for the problem of *Subset Selection for Regression*: select a subset of k variables from a given set of n observation variables which, taken together, "best" predict another variable of interest. This problem has many applications ranging from feature selection, sparse learning and dictionary selection in machine learning, to sparse approximation and compressed sensing in signal processing. From a machine learning perspective, the variables are typically features or observable attributes of a phenomenon, and we wish to predict the phenomenon using only a small subset from the high-dimensional feature space. In signal processing, the variables usually correspond to a collection of dictionary vectors, and the goal is to parsimoniously represent another (output) vector. For many practitioners, the prediction model of choice is linear regression, and the goal is to obtain a linear model using a small subset of variables, to minimize the mean square prediction error or, equivalently, maximize the squared multiple correlation R^2 (Johnson and Wichern, 2002; Miller, 2002).

Thus, we formulate the Subset Selection problem for Regression as follows: Given the (normalized) covariances between n variables X_i (which can in principle be observed) and a variable Z (which is to be predicted), select a subset of $k \ll n$ of the variables X_i and a linear prediction function of Z from the selected X_i that maximizes the R^2 fit. (A formal definition is given in Section 3.) The covariances are usually obtained empirically from detailed past observations of the variable values.

The above formulation is known (see, e.g., (Das and Kempe, 2008)) to be equivalent to the problem of *sparse approximation* over dictionary vectors: the input consists of a dictionary of n feature vectors $\mathbf{x_i} \in \mathbb{R}^m$, along with a target vector $\mathbf{z} \in \mathbb{R}^m$, and the goal is to select at most k vectors whose linear combination best approximates $\mathbf{z}$. The pairwise

*. A preliminary version was included in the proceedings of ICML 2011 under the title "Submodular Meets Spectral: Greedy Algorithms for Subset Selection, Sparse Approximation and Dictionary Selection."

†. Work done while the author was at the University of Southern California, supported in part by NSF grant DDDAS-TMRP 0540420.

‡. Supported in part by NSF CAREER award 0545855, and NSF grant DDDAS-TMRP 0540420.

covariances of the previous formulation are then exactly the inner products of the dictionary vectors.[1]

This problem is **NP**-hard (Natarajan, 1995), so no polynomial-time algorithms are known to solve it optimally for all inputs. Two approaches are frequently used for approximating such problems: greedy algorithms (Miller, 2002; Tropp, 2004; Gilbert et al., 2003; Zhang, 2008) and convex relaxation schemes (Tibshirani, 1996; Candès et al., 2005; Tropp, 2006; Donoho, 2005). For our formulation, a disadvantage of convex relaxation techniques is that they do not provide explicit control over the target sparsity level k of the solution; additional effort is needed to tune the regularization parameter.

Greedy algorithms are widely used in practice for subset selection problems; for example, they are implemented in all commercial statistics packages. They iteratively add or remove variables based on simple measures of fit with Z. Two of the most well-known and widely used greedy algorithms are the subject of our analysis: Forward Regression (Miller, 2002) and Orthogonal Matching Pursuit (Tropp, 2004). (These algorithms are defined formally in Section 3).

Our main result is that using the approximate submodularity framework, approximation guarantees much stronger than all previously known bounds can be obtained quite immediately. Specifically, we show that the relevant submodularity ratio for the R^2 objective is lower-bounded by the smallest $(2k)$-sparse eigenvalue $\lambda_{\min}(C, 2k)$ of the covariance matrix C of the observation variables. Combined with our general bounds for approximately submodular functions, this immediately implies a $(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)})$-approximation guarantee for Forward Regression. For Orthogonal Matching Pursuit, a similar analysis leads to a somewhat weaker guarantee of essentially $(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})$. In a precise sense, our analysis thus shows that the less singular C (or its small principal submatrices) are, the "closer to" submodular the R^2 objective. Previously, Das and Kempe (2008) had shown that R^2 is truly submodular when there are no "conditional suppressor" variables; however, the latter is a much stronger condition.

Most previous results for greedy subset selection algorithms (e.g., (Gilbert et al., 2003; Tropp, 2004; Das and Kempe, 2008)) had been based on coherence of the input data, i.e., the maximum correlation μ between any pair of variables. Small coherence is an extremely strong condition, and the bounds usually break down when the coherence is $\omega(1/k)$. On the other hand, most bounds for greedy and convex relaxation algorithms for sparse recovery are based on a weaker sparse-eigenvalue or Restricted Isometry Property (RIP) condition (Zhang, 2009, 2008; Lozano et al., 2009; Zhou, 2009; Candès et al., 2005). However, these results apply to a different objective: minimizing the difference between the actual and estimated coefficients of a sparse vector. Simply extending these results to the subset selection problem adds a dependence on the largest k-sparse eigenvalue and only leads to weak additive bounds.

Dictionary Selection. As a second illustration of the approximate submodularity framework, we obtain much tighter theoretical performance guarantees for greedy algorithms for dictionary selection (Krause and Cevher, 2010). In the *Dictionary Selection problem*, we are given s target vectors, and a candidate set V of feature vectors. The goal is to select a set

1. For this reason, the dimension m of the feature vectors only affects the problem indirectly, via the accuracy of the estimated covariance matrix.

$D \subset V$ of at most d feature vectors, which will serve as a *dictionary* in the following sense. For each of the target vectors, the best $k < d$ vectors from D will be selected and used to achieve a good R^2 fit; the goal is to maximize the average R^2 fit for all of these vectors. (A formal definition is given in Section 4.) The problem of finding a dictionary of basis functions for sparse representation of signals has several applications in machine learning and signal processing. Krause and Cevher (2010) showed that greedy algorithms for dictionary selection can perform well in many instances, and proved additive approximation bounds for two specific algorithms, $\mathrm{SDS}_{\mathrm{MA}}$ and $\mathrm{SDS}_{\mathrm{OMP}}$ (defined in Section 4). Our approximate submodularity framework directly yields stronger multiplicative approximation guarantees.

Our theoretical analysis is complemented by experiments comparing the performance of the greedy algorithms and a baseline convex-relaxation algorithm for subset selection on two real-world data sets and a synthetic data set. We also evaluate the submodularity ratio of these data sets and compare it with other spectral parameters: while the input covariance matrices are close to singular, the submodularity ratio actually turns out to be significantly larger.

While the submodularity ratio is always *lower-bounded* by the smallest (sparse) eigenvalue, our experiments reveal that this lower bound can be loose. This happens when there are small (sparse) eigenvalues, but the predictor variable is not badly aligned with their eigenspace. Hence, computing the submodularity ratio explicitly (although it appears computationally intensive to do so) can lead to stronger post hoc approximation guarantees. In this context, we also discuss ways in which a more careful analysis of the greedy algorithms allows significantly stronger post hoc approximation guarantees.

Our main contributions can be summarized as follows:

1. We introduce (in Section 2) the notion of the submodularity ratio as a predictor of the performance of greedy algorithms. We show that a submodularity ratio of γ leads to a $(1-\mathrm{e}^{-\gamma})$-approximation guarantee for the greedy algorithm for maximum coverage. For the minimum cover probem, we show essentially a $\frac{\log n}{\gamma}$ approximation guarantee for the greedy algorithm.

2. Using the approximate submodularity framework, in Section 3, we obtain the strongest known theoretical performance guarantees for greedy algorithms for subset selection. In particular, we show that the Forward Regression and OMP algorithms are within a $1-\mathrm{e}^{-\gamma}$ factor and $1-\mathrm{e}^{-(\gamma\cdot\lambda_{\min})}$ factor of the optimal solution, respectively (where the γ and λ terms are appropriate submodularity and sparse-eigenvalue parameters).

3. Again using the approximate submodularity framework, in Section 4, we obtain the strongest known theoretical guarantees for algorithms for dictionary selection, improving on the results of Krause and Cevher (2010). In particular, we show that the $\mathrm{SDS}_{\mathrm{MA}}$ algorithm is within a factor $\frac{\gamma}{\lambda_{\max}}(1-\frac{1}{\mathrm{e}})$ of optimal.

4. We evaluate our theoretical bounds for subset selection by running greedy and L1-relaxation algorithms on real-world and synthetic data, and show how the various submodular and spectral parameters correlate with the performance of the algorithms in practice.

1.1 Related and Subsequent Work

We provide an overview of related work both in the context of subset selection (and its variants) and in submodular optimization, as well as a discussion of work that appeared subsequent to the conference version of the present article.

1.1.1 Subset Selection and Sparse Recovery

There has been a lot of related work in the statistics, machine learning and signal processing communities on problems with sparsity constraints (such as sparse recovery, compressed sensing, sparse approximation and feature selection).

In sparse recovery, one is given an $n \times m$ dictionary ϕ of m vectors in $\mathbb{R}^n$ (where $n < m$), along with another vector $y \in \mathbb{R}^n$. It is known that y has some sparse representation in terms of k vectors of ϕ, up to a small noise term ϵ, and the goal is to recover the coefficients α given y, ϕ and ϵ. There has been a lot of recent interest in greedy and convex relaxation techniques for the sparse recovery problems, both in the noiseless and noisy setting. For L1 relaxation techniques, Tropp (2006) showed conditions based on the coherence (i.e., the maximum correlation between any pair of variables) of the dictionary that guaranteed near-optimal recovery of a sparse signal. In (Candès et al., 2005; Donoho, 2005), it was shown that if the target signal is truly sparse, and the dictionary obeys a Restricted Isometry Property (RIP), then L1 relaxation can almost exactly recover the true sparse signal. Other results (Zhao and Yu, 2006; Zhou, 2009) also prove conditions under which L1 relaxation can recover a sparse signal. Though related, the above results are not directly applicable to our subset selection formulation, since the goal in sparse recovery is to recover the true coefficients of the sparse signal, as opposed to our problem of minimizing the prediction error of an arbitrary signal subject to a specified sparsity level.

For greedy sparse recovery, Zhang (2008, 2009) and Lozano et al. (2009) provided conditions based on sparse eigenvalues under which Forward Regression and Forward-Backward Regression can recover a sparse signal. As with the L1 results for sparse recovery, the objective function analyzed in these papers is somewhat different from that in our subset selection formulation; furthermore, these results are intended mainly for the case when the predictor variable is truly sparse. Simply extending these results to our problem formulation gives weaker, additive bounds and requires stronger conditions than our results.

The papers by Das and Kempe (2008), Gilbert et al. (2003) and Tropp et al. (2003); Tropp (2004) analyzed greedy algorithms for sparse approximation, which as mentioned previously is equivalent to our subset selection formulation presented in this work. In particular, they obtained a $1 + \Theta(\mu^2 k)$ multiplicative approximation guarantee for the mean square error objective and a $1 - \Theta(\mu k)$ guarantee for the R^2 objective, whenever the coherence μ of the dictionary is $O(1/k)$. These results are thus weaker than those presented here, since they do not apply to instances with even moderate correlations of $\omega(1/k)$.

Other analysis of greedy methods includes the work of Natarajan (1995), which proved a bicriteria approximation bound for minimizing the number of vectors needed to achieve a given prediction error.

As mentioned earlier, the paper by Krause and Cevher (2010) analyzed greedy algorithms for the dictionary selection problem, which generalizes subset selection to prediction of multiple variables. They too use a notion of approximate submodularity to provide additive approximation guarantees. Since their analysis is for a more general problem than subset selection, applying their results directly to the subset selection problem predictably gives much weaker bounds than those presented in this paper for subset selection. Furthermore, even for the general dictionary selection problem, our techniques can be used to significantly improve their analysis of greedy algorithms and obtain tighter multiplicative approximation bounds (as shown in Section 4).

In general, we note that the performance bounds for greedy algorithms derived using the coherence parameter are usually the weakest, followed by those using the Restricted Isometry Property, then those using sparse eigenvalues, and finally those using the submodularity ratio. (We show an empirical comparison of these parameters in Section 5.)

1.1.2 Submodular Maximization and Curvature

In the context of submodular maximization, the celebrated result of Nemhauser et al. (1978) proved that the greedy algorithm obtained a $(1 - 1/e)$-approximation for maximizing any monotone, submodular set function subject to a uniform matroid. The same guarantee was obtained by Calinescu et al. (2011) for an arbitrary matroid constraint, using a continuous variant of the greedy algorithm.

While we are not aware of prior work on defining a notion of how far a function is from being submodular (or analyzing greedy algorithms for such functions), there is a well-known notion of curvature (Conforti and Cornuéjols, 1984; Vondrák, 2010) that captures how far a submodular function is from being *modular*. In particular, the *total curvature* of a submodular set function is defined as $c = 1 - \min_{S, j \notin S} \frac{f_S(j)}{f_\emptyset(j)}$, where $f_S(j) = f(S \cup j) - f(S)$. (Additional related notions include average curvature and monotonicity ratio; see (Iyer, 2015) for a discussion.) Intuitively c measures how far away f is from being modular, and is equal to 0 if f is modular. Conforti and Cornuéjols (1984) analyzed the greedy algorithm for submodular maximization in terms of the c parameter, and showed a $\frac{1}{c}(1 - e^{-c})$ approximation for a uniform matroid. The result was extended to an arbitrary matroid by Vondrák (2010), and an improved guarantee of $(1 - c/e)$ was obtained recently by Sviridenko et al. (2015). Curvature was also used by Iyer et al. (2013) to obtain improved bounds for submodular function approximation, PMAC-learning and submodular minimization.

Another notion of approximate modularity was recently proposed by Chierichetti et al. (2015), who defined a function to be ϵ-approximately modular if it satisfies all the modularity requirements to within an ϵ additive error. Chierichetti et al. (2015) analyzed how close (in the l_∞ distance) any approximately modular function can be to a modular function.

Note that both the notions of total curvature and approximate modularity are different from the submodularity ratio proposed in this paper, which measures how far a set function is from being submodular.

1.1.3 Subsequent Work

Subsequent to our work introducing the submodularity ratio, several papers have used this notion for analyzing greedy algorithms for machine learning applications. Das et al. (2012) proposed diversity-promoting spectral regularizers for feature selection, and used the submodularity ratio to analyze a hybrid greedy and local search algorithm for the diverse feature selection problem. Grubb and Bagnell (2012) analyzed greedy algorithms for learning an

ensemble of *anytime predictors* that automatically trade computation time with predictive accuracy. Using the submodularity ratio, the authors provide an approximation guarantee for the performance of their ensemble algorithm. Kusner et al. (2014) analyzed greedy methods for training a tree of classifiers for feature-cost sensitive learning, and show that the objective function for obtaining a cost-sensitive tree of classifiers is approximately submodular. Qian et al. (2015) proposed a Pareto optimization approach for subset selection in sparse regression and analyzed the performance of their algorithm using the submodularity ratio.

Most directly following up on our initial work is a recent result of Elenberg et al. (2018) that extends our analysis of greedy algorithms for subset selection from the linear regression setting to arbitrary Generalized Linear Models. The main result is a lower bound on any function's submodularity ratio in terms of its restricted strong convexity and smoothness parameters, which can then be used to obtain approximation guarantees for greedy feature selection algorithms.

2. Approximate Submodularity

We begin by defining our notion of approximate submodularity, and explaining its relationship with the traditional notion of submodularity. Then, we show that approximation results for greedy algorithms degrade gracefully as the function becomes less and less submodular.

2.1 Submodularity Ratio

We introduce the notion of submodularity ratio for a general set function, which captures "how close" to submodular the function is. Let $\mathcal{X}$ be a universe of elements, and Let $f : 2^{\mathcal{X}} \to \mathbb{R}^+$ be a non-negative set function.

Definition 1 (Monotonicity, Submodularity) *1. f is* monotone *iff $f(S) \leq f(T)$ whenever $S \subseteq T$.*

2. f is submodular *iff $f(S \cup \{x\}) - f(S) \geq f(T \cup \{x\}) - f(T)$ whenever $S \subseteq T$.*

Our definition of the submodularity ratio smoothly interpolates between functions that are submodular and those that are far from so.

Definition 2 (Submodularity Ratio) *The* submodularity ratio *of a monotone function f with respect to a set U and a parameter $k \geq 1$ is*

$$\gamma_{U,k}(f) = \min_{L \subseteq U, S:|S| \leq k, S \cap L = \emptyset} \frac{\sum_{x \in S} f(L \cup \{x\}) - f(L)}{f(L \cup S) - f(L)}, \tag{1}$$

where we define $0/0 := 1$. Thus, the submodularity ratio captures how much more f can increase by adding any subset S of size k to L, compared to the combined benefits of adding its individual elements to L. That Definition 2 generalizes submodularity is captured by the following proposition.

Proposition 3 *f is submodular if and only if $\gamma_{U,k} \geq 1$ for all U and k.*

Proof. First, assume that $\gamma_{U,k} \geq 1$ for all U and k. By choosing $k = 2$ and $S = \{x, y\}$ in Equation (1), we obtain that $f(L \cup \{x\}) + f(L \cup \{y\}) \geq f(L \cup \{x, y\}) + f(L)$, or, rearranged, $f(L \cup \{x\}) - f(L) \geq f(L \cup \{x, y\}) - f(L \cup \{y\})$. Now, when we have two sets S and $T = S \cup \{x_1, x_2, \ldots, x_k\}$, define $S_i := S \cup \{x_1, \ldots, x_i\}$ for $0 \leq i \leq k$. Setting $L = S_i$ now gives us that $f(S_i \cup \{x\}) - f(S_i) \geq f(S_{i+1} \cup \{x\}) - f(S_{i+1})$. Induction on i now completes the proof.

Conversely, assume that f is submodular. In Equation (1), let $S = \{x_1, \ldots, x_k\}$ and $S_i = \{x_1, \ldots, x_i\}$, and write a telescoping series $f(L \cup S) - f(L) = \sum_{i=0}^{k-1} f(L \cup S_{i+1}) - f(L \cup S_i)$. By submodularity of f, we can bound

$$f(L \cup S_{i+1}) - f(L \cup S_i) \;=\; f(L \cup S_i \cup \{x_{i+1}\}) - f(L \cup S_i) \;\leq\; f(L \cup \{x_{i+1}\}) - f(L),$$

which gives us a lower bound of 1 on the ratio. ■

Remark 4 *The submodularity ratio is defined as a minimum over exponentially many values, and in general, it is* **NP***-hard to compute exactly (more recently, Bai and Bilmes (2018) showed that it cannot be computed in polynomial time in the value oracle model). This is a property it shares with the well-known Restricted Isometry Property (RIP) (Candès and Tao, 2005): computing the RIP of a matrix is essentially equivalent to computing the expansion of a graph, yet the guarantees for sparse approximation algorithms are frequently expressed in terms of the RIP.*

Whether one can efficiently approximate the submodularity ratio to within non-trivial factors is an interesting open question. Approximating it would allow one to at least derive post hoc approximation guarantees, i.e., to give the user guarantees on the approximation quality for the specific instance that was solved. In the appendix, we discuss some (fairly strong) assumptions under which one can derive non-trivial lower bounds on the submodularity ratio.

Typically, rather than computing the submodularity ratio on a given instance, one would use problem-specific insights to derive a priori lower bounds on the submodularity ratio in terms of quantities that are easier to compute exactly or approximately. For example, in the primary application studied here (linear regression), the submodularity ratio is lower-bounded by the (easy to compute) smallest eigenvalue of the covariance matrix, and more tightly bounded by the (not so easy to compute) smallest $2k$-sparse eigenvalue of the covariance matrix. Recently, Elenberg et al. (2018) showed how to derive similar lower bounds for a more general class of linear objective functions. We anticipate that similar types of bounds can be obtained for other classes of objectives.

2.2 The Greedy Algorithm for Maximum Coverage

Probably the most widely used fact about (monotone) submodular functions is that a simple greedy algorithm approximately optimizes the function subject to a cardinality constraint.[2] This is a celebrated result by Nemhauser et al. (1978). Specifically, Nemhauser et al. (1978) analyzed the following algorithm.

Let S^{NG} be the final set S_k returned by the algorithm. The following theorem of Nemhauser et al. (1978) is widely used in the Machine Learning and related communities:

2. Many other algorithmic optimization problems are easier for submodular function. Some of them are discussed in Section 6.

Algorithm 1 The Nemhauser Greedy Algorithm for a non-negative, monotone, and submodular set function f on a universe $\mathcal{X}$.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i+1 = 1, 2, \ldots$ **do**
3: Let $x_{i+1} \in \mathcal{X}$ be an element maximizing $f(S_i \cup \{x_{i+1}\})$, and set $S_{i+1} = S_i \cup \{x_{i+1}\}$.
4: Output S_k.

Theorem 5 (Nemhauser et al. (1978)) *The set S^{NG} returned by the Nemhauser Greedy Algorithm guarantees that $f(S^{NG}) \geq (1 - \frac{1}{e}) \cdot f(S_k^*)$, where S_k^* is the set maximizing $f(S)$ among all size-k sets S.*

The centerpiece of our algorithmic analysis is a generalization of Theorem 5 to approximately submodular functions.

Theorem 6 *Let f be a nonnegative, monotone set function, and OPT be the maximum value of f obtained by any set of size k. Then, the set S^{NG} selected by the Nemhauser Greedy Algorithm has the following approximation guarantee:*

$$f(S^{NG}) \geq \left(1 - \mathrm{e}^{-\gamma_{S^{NG},k}(f)}\right) \cdot OPT.$$

Notice that for submodular functions, because $\gamma_{S^{\mathrm{NG}},k}(f) \geq 1$, our theorem recovers the result of Nemhauser et al. (1978) as a special case.

Proof. We carry out the analysis in somewhat more generality than needed here, since most of it will be useful in Section 2.3. Let k be the number of iterations that Algorithm 1 was run, and S_i^{NG} the set of elements greedily chosen in the first i iterations. Let S_i^{NG} be the set of variables chosen by the Nemhauser Greedy Algorithm (Algorithm 1) in the first i iterations. Define $A(i) = f(S_i^{\mathrm{NG}}) - f(S_{i-1}^{\mathrm{NG}})$ to be the gain obtained from the variable chosen by the algorithm in iteration i. Then $f(S^{\mathrm{NG}}) = \sum_{i=1}^{k} A(j)$.

For simplicity of notation, we write $f(x/S)$ to denote $f(\{x\} \cup S) - f(S)$, and $f(T/S)$ to denote $f(T \cup S) - f(S)$, for any element $x \in \mathcal{X}$ and sets S and T. We will also write $\gamma_{S^{\mathrm{NG}},k}$ to denote $\gamma_{S^{\mathrm{NG}},k}(f)$.

Let S^* be some (optimum) set of k^* variables, achieving a value of (at least) C. Let $S_i = S^* \setminus S_i^{\mathrm{NG}}$. By monotonicity of f and the fact that $S_i \cup S_i^{\mathrm{NG}} \supseteq S^*$, we have that $f(S_i \cup S_i^{\mathrm{NG}}) \geq C$. We will show that at least one of the $x \in S_i$ is a good candidate in iteration $i+1$ of the algorithm. First, the joint contribution of S_i, conditioned on the set S_i^{NG}, must be fairly large: $f(S_i/S_i^{\mathrm{NG}}) = f(S_i \cup S_i^{\mathrm{NG}}) - f(S_i^{\mathrm{NG}}) \geq C - f(S_i^{\mathrm{NG}})$. Using Definition 2, as well as $S_i^{\mathrm{NG}} \subseteq S^{\mathrm{NG}}$ and $|S_i| \leq k^*$,

$$\sum_{x \in S_i} f(x/S_i^{\mathrm{NG}}) \geq \gamma_{S_i^{\mathrm{NG}},|S_i|} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \gamma_{S^{\mathrm{NG}},k^*} \cdot f(S_i/S_i^{\mathrm{NG}}).$$

Let $\hat{x} \in \mathrm{argmax}_{x \in S_i} f(x/S_i^{\mathrm{NG}})$ maximize $f(\hat{x}/S_i^{\mathrm{NG}})$. Then we get that

$$f(\hat{x}/S_i^{\mathrm{NG}}) \geq \frac{\gamma_{S^{\mathrm{NG}},k^*}}{|S_i|} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot f(S_i/S_i^{\mathrm{NG}}).$$

Since the $\hat{x}$ above was a candidate to be chosen in iteration $i+1$, and the algorithm chose a variable x_{i+1} such that $f(x_{i+1}/S_i^{\mathrm{NG}}) \geq f(x/S_i^{\mathrm{NG}})$ for all $x \notin S_i^{\mathrm{NG}}$, we obtain that

$$A(i+1) \geq \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot f(S_i/S_i^{\mathrm{NG}}) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - f(S_i^{\mathrm{NG}})) \;\geq\; \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - \sum_{j=1}^{i} A(j)).$$

We will use the above inequality to prove by induction on t that

$$C - \sum_{i=1}^{t} A(i) \leq C \cdot (1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*})^t \;\leq\; C \cdot \mathrm{e}^{-\gamma_{S^{\mathrm{NG}},k^*} \cdot \frac{t}{k^*}}. \tag{2}$$

The base case is clearly true for $t = 0$. Suppose that the inequality is true after t iterations. Then, at iteration $t+1$, we have

$$\begin{aligned} C - \sum_{i=1}^{t+1} A(i) &= C - \sum_{i=1}^{t} A(i) - A(t+1) \\ &\leq C - \sum_{i=1}^{t} A(i) - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot (C - \sum_{i=1}^{t} A(i)) \\ &= (C - \sum_{i=1}^{t+1} A(i)) \cdot \left(1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*}\right) \\ &\leq C \cdot \left(1 - \frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*}\right)^{t+1}, \end{aligned}$$

thus completing the inductive proof. Using Inequality(2) with $k = k^*, t = k-1$ and $C = \mathrm{OPT}$, we obtain that

$$f(S^{\mathrm{NG}}) = \sum_{i=1}^{k} A(i) \;\geq\; \mathrm{OPT} \cdot \left(1 - \mathrm{e}^{-\gamma_{S^{\mathrm{NG}},k}}\right).$$

This completes the proof of the approximation guarantee. ∎

Remark 7 *As the submodularity ratio goes to 0, the approximation guarantee of Theorem 6 deteriorates and becomes 0 in the limit. This is not surprising: in the limit, the definition does not place any restrictions on the function f. Without any restrictions on f, not only can the greedy algorithm perform arbitrarily poorly, but the same may be true for* any *efficient algorithm, since f might be a function that is provably hard to approximate to within any non-trivial factor.*

Indeed, the goal of Theorem 6 is not to provide a universal approximation guarantee, but rather to outline conditions under which running the greedy algorithm comes with provable approximation guarantees. Practitioners run greedy algorithms routinely without any guarantees, and the submodularity ratio may provide guidance under what conditions doing so has theoretical justification, even when the objective function f is not submodular.

2.3 The Greedy Algorithm for Minimum Submodular Cover

The "complementary" problem to submodular function maximization is minimum submodular cover, where the goal is to find a smallest set S with $f(S) \geq C$, a given target value. The name derives from one of the most common instance of submodular functions: coverage functions.[3] Here, the elements x correspond to sets, and the function value f is the size of the union of the selected sets. In the Maximum Coverage Problem, the goal is to maximize the size of the union by selecting k sets, and in the Minimum Set Cover Problem, the goal is to cover all elements selecting as few sets as possible.

For both problems, the greedy algorithm (Algorithm 1) provides essentially best possible guarantees. The only difference is the termination condition: for maximum coverage, the algorithm is terminated when k sets are selected, while for minimum cover, the algorithm is terminated when all elements (or a given number) have been covered. For the Minimum Set Cover Problem, the greedy algorithm achieves a $\ln n$ approximation, which is best possible unless $\mathbf{P} = \mathbf{NP}$. For more general monotone submodular functions, the results are somewhat less clean to express, but are summarized by the following theorem of Wolsey (1982).

Theorem 8 (Theorem 1 of Wolsey (1982)) *Let f be nonnegative, monotone and submodular, and let $n = |\mathcal{X}|$. For any given C, let $k^*(C)$ be the size of the smallest set $S \subseteq V$ such that $f(S) \geq C$. Let k be the size of the set S^{NG} selected by Algorithm 1 when run until $f(S) \geq C$. Then,*

$$k \leq \left(1 + \log\left(\frac{C}{C - f(S^{NG}_{k-1})}\right)\right) \cdot k^*(C),$$

where S^{NG}_{k-1} is the set selected by Algorithm 1 after $k-1$ iterations.

If f is integer valued, then

$$k \leq (1 + \log(\theta)) \cdot k^*,$$

where $\theta = \max_{x \in \mathcal{X}} f(x)$ is the maximum value of the set function obtained by a single element.

We show that Theorem 8, too, extends gracefully to approximately submodular functions f.

Theorem 9 *Let f be a nonnegative and monotone function, and let $n = |\mathcal{X}|$. For any given C, let $k^*(C)$ be the size of the smallest set $S \subseteq V$ such that $f(S) \geq C$. Let k be the size of the set S^{NG} selected by Algorithm 1 when run until $f(S) \geq C$. Then,*

$$k \leq 1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \cdot \log\left(\frac{C}{C - f(S^{NG}_{k-1})}\right) \cdot k^*(C),$$

where S^{NG}_{k-1} is the set selected by Algorithm 1 after $k-1$ iterations.

3. A characterization of coverage functions in terms of functional properties akin to submodularity is given by Salek et al. (2010).

Proof. We use the same notation as in the proof of Theorem 6. For notational convenience, write $k^* = k^*(C)$. Let k be the number of iterations taken by Algorithm 1, so that $f(S^{\mathrm{NG}}_k) \geq C$ and $f(S^{\mathrm{NG}}_{k-1}) < C$. Thus $f(S^{\mathrm{NG}}) = \sum_{j=1}^{k} A(j)$.

Let S^* be a smallest set (i.e., $|S^*| = k^*$) with $f(S^*) \geq C$. Substituting $t = k-1$ into Equation (2) and solving for k, we obtain that

$$k \leq 1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{C}{C - f(S^{\mathrm{NG}}_{k-1})}\right) \cdot k^*,$$

as claimed. ∎

As with Wolsey's result for submodular functions, the bounds can be improved when f is integer-valued.

Theorem 10 *Assume that f is integer-valued, in addition to all conditions (and notation) of Theorem 9. Let $\theta = \max_{x \in \mathcal{X}} f(x)$ is the maximum value of the set function obtained by any single element. Then, the number k of elements selected by Algorithm 1 satisfies*

$$k \leq 1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \cdot \log(C) \cdot k^*(C),$$
$$k \leq \left(1 + \frac{1}{\gamma_{S^{NG},k^*(C)}(f)} \log\left(\frac{\theta}{\gamma_{\emptyset,k^*}(f)}\right)\right) \cdot k^*(C).$$

Proof. The first result follow directly from Theorem 9, because $C - f(S^{\mathrm{NG}}_{k-1}) \geq 1$ for integer-valued functions.

For the second result, substitute $t = \frac{k^*}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{f(S^*)}{k^*}\right)$ into Inequality (2) to obtain that

$$C - f(S^{\mathrm{NG}}_t) \leq C \cdot \mathrm{e}^{-\frac{\gamma_{S^{\mathrm{NG}},k^*}}{k^*} \cdot t} \;\leq\; k^*.$$

Because f is a monotone and integer-valued, $f(S^{\mathrm{NG}}_i) - f(S^{\mathrm{NG}}_{i-1}) \geq 1$ for all remaining iterations i, and it takes at most k^* additional iterations to reach a value of C. Hence,

$$k \leq t + k^* \;=\; \left(1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log(C/k^*)\right) \cdot k^* \;\leq\; \left(1 + \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \cdot \log\left(\frac{\theta}{\gamma_{\emptyset,k^*}(f)}\right)\right) \cdot k^*.$$

The inequality $C/k^* \leq \theta/\gamma_{\emptyset,k^*}(f)$ is directly from Definition 2. ∎

The same techniques can be used to obtain the following bicriteria approximation guarantee below. The bicriteria guarantees are similar in spirit to, for instance, (Krause and Golovin, 2014, Theorem 1.5). We believe that similar results for submodular functions are folklore among researchers, though we are unaware of a reference stating precisely the form we give here.

Theorem 11 *For any $\epsilon \in (0,1)$, if Algorithm 1 is run until $f(S^{NG}) \geq (1-\epsilon) \cdot C$, the size of the set S^{NG} that is returned is at most $\frac{1}{\gamma_{S^{NG},k^*}(f)} \log(\frac{1}{\epsilon}) \cdot k^*(C)$.*

Proof. For the proof, simply substitute $t = \frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)} \log(\frac{1}{\epsilon}) \cdot k^*(C)$ into Inequality (2). ■

A particularly clean corollary of this theorem is obtained when $\epsilon = 1/\mathrm{e}$. In that case, we obtain a $(1 - 1/\mathrm{e})$ approximation by increasing the set size by a factor $\frac{1}{\gamma_{S^{\mathrm{NG}},k^*}(f)}$. Thus, instead of a smooth degradation of the customary $(1 - 1/\mathrm{e})$ approximation guarantee, we can choose a smooth increase in the size of the set that the greedy algorithm is allowed to select, and thus retain the customary $(1 - 1/\mathrm{e})$ approximation, even for functions that are only approximately submodular.

3. Subset Selection for Regression

As our first and main application of the approximate submodularity framework, we analyze greedy algorithms for subset selection in regression. The goal in subset selection is to estimate a *predictor variable* Z using linear regression on a small subset from the set of *observation variables* $\mathcal{X} = \{X_1, \ldots, X_n\}$. We use $\mathrm{Var}[X_i]$, $\mathrm{Cov}[X_i, X_j]$ and $\rho(X_i, X_j)$ to denote the variance, covariance and correlation of random variables, respectively. By appropriate normalization, we can assume that all the random variables have expectation 0 and variance 1. The matrix of covariances between the X_i and X_j is denoted by C, with entries $c_{i,j} = \mathrm{Cov}[X_i, X_j]$. Similarly, we use $\mathbf{b}$ to denote the covariances between Z and the X_i, with entries $b_i = \mathrm{Cov}[Z, X_i]$. Formally, the *Subset Selection* problem can now be stated as follows:

Definition 12 (Subset Selection) *Given pairwise covariances among all variables, as well as a parameter k, find a set $S \subset \mathcal{X}$ of at most k variables X_i and a linear predictor $Z' = \sum_{i \in S} \alpha_i X_i$ of Z, maximizing the* squared multiple correlation *(Diekhoff, 2002; Johnson and Wichern, 2002)*

$$R^2_{Z,S} = \frac{\mathrm{Var}[Z] - \mathbb{E}\left[(Z - Z')^2\right]}{\mathrm{Var}[Z]}.$$

R^2 is a widely used measure for the goodness of a statistical fit; it captures the fraction of the variance of Z explained by variables in S. Because we assumed Z to be normalized to have variance 1, it simplifies to $R^2_{Z,S} = 1 - \mathbb{E}\left[(Z - Z')^2\right]$.

For a set S, we use C_S to denote the submatrix of C with row and column set S, and $\mathbf{b}_S$ to denote the vector with only entries b_i for $i \in S$. For notational convenience, we frequently do not distinguish between the index set S and the variables $\{X_i \mid i \in S\}$. Given the subset S of variables used for prediction, the optimal regression coefficients α_i are well known to be $\mathbf{a}_S = (\alpha_i)_{i \in S} = C_S^{-1} \cdot \mathbf{b}_S$ (see, e.g., (Johnson and Wichern, 2002)), and hence $R^2_{Z,S} = \mathbf{b}_S^\intercal C_S^{-1} \mathbf{b}_S$. Thus, the subset selection problem can be phrased as follows: Given C, $\mathbf{b}$, and k, select a set S of at most k variables to maximize $R^2_{Z,S} = \mathbf{b}_S^\intercal (C_S^{-1}) \mathbf{b}_S$.[4]

Many of our results are phrased in terms of eigenvalues of the covariance matrix C and its submatrices. Covariance matrices are positive semidefinite, so their eigenvalues are real

4. We assume throughout that C_S is non-singular. For some of our results, an extension to singular matrices is possible using the Moore-Penrose generalized inverse.

and non-negative (Johnson and Wichern, 2002). We denote the eigenvalues of a positive semidefinite matrix A by $\lambda_{\min}(A) = \lambda_1(A) \leq \lambda_2(A) \leq \cdots \leq \lambda_n(A) = \lambda_{\max}(A)$. We use $\lambda_{\min}(C, k) = \min_{S:|S|=k} \lambda_{\min}(C_S)$ to refer to the smallest eigenvalue of any $k \times k$ submatrix of C (i.e., the smallest k-sparse eigenvalue), and similarly $\lambda_{\max}(C, k) = \max_{S:|S|=k} \lambda_{\max}(C_S)$.[5] We also use $\kappa(C, k)$ to denote the largest condition number (the ratio of the largest and smallest eigenvalue) of any $k \times k$ submatrix of C. This quantity is strongly related to the Restricted Isometry Property in (Candès et al., 2005). We also use $\mu(C) = \max_{i \neq j} |c_{i,j}|$ to denote the *coherence*, i.e., the maximum absolute pairwise correlation between the X_i variables. Recall the L_2 vector and matrix norms: $\|\mathbf{x}\|_2 = \sqrt{\sum_i |x_i|^2}$, and $\|A\|_2 = \lambda_{\max}(A) = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$. We also use $\|\mathbf{x}\|_0 = |\{i \mid x_i \neq 0\}|$ to denote the sparsity of a vector $\mathbf{x}$.

The Rayleigh-Ritz representation for $\|A\|_2$ is useful in bounding $\lambda_{\min}(A)$, as for any vector $\mathbf{x}$, we have $\lambda_{\min}(A) \leq \frac{\|A\mathbf{x}\|_2}{\|\mathbf{x}\|_2}$.

The part of a variable Z that is not correlated with the X_i for all $i \in S$, i.e., the part that cannot be explained by the X_i, is called the *residual* (see (Diekhoff, 2002)), and defined as $\mathrm{Res}(Z, S) = Z - \sum_{i \in S} \alpha_i X_i$.

3.1 Approximate Submodularity of R^2

The key insight enabling our analysis is a bound on the submodularity ratio of the R^2 function. To avoid notational clutter, when we are specifically concerned with the R^2 objective defined on the variables X_i, we omit the function name in the definition of the submodularity ratio, and simply write

$$\gamma_{U,k} = \min_{L \subseteq U, S: |S| \leq k, S \cap L = \emptyset} \frac{\sum_{i \in S} (R^2_{Z,L \cup \{X_i\}} - R^2_{Z,L})}{R^2_{Z,S \cup L} - R^2_{Z,L}} = \min_{L \subseteq U, S: |S| \leq k, S \cap L = \emptyset} \frac{(\mathbf{b}_S^L)^\intercal \mathbf{b}_S^L}{(\mathbf{b}_S^L)^\intercal (C_S^L)^{-1} \mathbf{b}_S^L},$$

where C^L and $\mathbf{b}^L$ are the normalized covariance matrix and normalized covariance vector corresponding to the set $\{\mathrm{Res}(X_1, L), \mathrm{Res}(X_2, L), \ldots, \mathrm{Res}(X_n, L)\}$.

Our key lemma can now be stated as follows:

Lemma 13 $\gamma_{U,k} \geq \lambda_{\min}(C, k + |U|) \geq \lambda_{\min}(C)$.

For all our analysis in this paper, we will use $|U| = k$, and hence $\gamma_{U,k} \geq \lambda_{\min}(C, 2k)$. Thus, the smallest $2k$-sparse eigenvalue is a lower bound on this submodularity ratio; as we show later, it is often a weak lower bound.

Before proving Lemma 13, we first introduce two lemmas that relate the eigenvalues of a normalized covariance matrix with those of its submatrices.

Lemma 14 *Let C be the covariance matrix of n zero-mean random variables $X_1, X_2, \ldots, X_n$, each of which has variance at most 1. Let C_ρ be the corresponding correlation matrix of the n random variables, that is, C_ρ is the covariance matrix of the variables after they are normalized to have unit variance. Then $\lambda_{\min}(C) \leq \lambda_{\min}(C_\rho)$.*

5. Computing $\lambda_{\min}(C, k)$ is NP-hard. In Appendix A we describe how to efficiently approximate the values for some scenarios.

Proof. Since C_ρ is obtained by normalizing the variables such that they have unit variance, we get $C_\rho = D^\intercal C D$, where D is a diagonal matrix with diagonal entries $d_i = \frac{1}{\sqrt{\mathrm{Var}[X_i]}}$.

Since both C_ρ and C are positive semidefinite, we can perform Cholesky factorization to get lower-triangular matrices A_ρ and A such that $C = AA^\intercal$ and $C_\rho = A_\rho A_\rho^\intercal$. Hence $A_\rho = D^\intercal A$.

Let $\sigma_{\min}(A)$ and $\sigma_{\min}(A_\rho)$ denote the smallest singular values of A and A_ρ, respectively. Also, let $\mathbf{v}$ be the singular vector corresponding to $\sigma_{\min}(A_\rho)$. Then,

$$\|A\mathbf{v}\|_2 = \|D^{-1}A_\rho\mathbf{v}\|_2 \;\le\; \|D^{-1}\|_2\|A_\rho\mathbf{v}\|_2 \;=\; \sigma_{\min}(A_\rho)\|D^{-1}\|_2 \;\le\; \sigma_{\min}(A_\rho),$$

where the last inequality follows since

$$\|D^{-1}\|_2 \;=\; \max_i \frac{1}{d_i} \;=\; \max_i \sqrt{\mathrm{Var}[X_i]} \;\le\; 1.$$

Hence, by the Courant-Fischer theorem, $\sigma_{\min}(A) \le \sigma_{\min}(A_\rho)$, and consequently, $\lambda_{\min}(C) \le \lambda_{\min}(C_\rho)$. ∎

Lemma 15 *Let $\lambda_{\min}(C)$ be the smallest eigenvalue of the covariance matrix C of n random variables $X_1, X_2, \ldots, X_n$, and $\lambda_{\min}(C')$ be the smallest eigenvalue of the $(n-1)\times(n-1)$ covariance matrix C' corresponding to the $n-1$ random variables $\mathrm{Res}(X_1, X_n), \ldots, \mathrm{Res}(X_{n-1}, X_n)$. Then $\lambda_{\min}(C) \le \lambda_{\min}(C')$.*

Proof. Let λ_i and λ'_i denote the eigenvalues of C and C', respectively. Also, let $c'_{i,j}$ denote the entries of C'. Using the definition of the residual, we get that

$$c'_{i,j} = \mathrm{Cov}[\mathrm{Res}(X_i, X_n), \mathrm{Res}(X_j, X_n)] \;=\; c_{i,j} - \frac{c_{i,n}c_{j,n}}{c_{n,n}},$$

$$c'_{i,i} = \mathrm{Var}[\mathrm{Res}(X_i, X_n)] \;=\; c_{i,i} - \frac{c_{i,n}^2}{c_{n,n}}.$$

Defining $D = \frac{1}{c_{n,n}} \cdot [c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal \cdot [c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]$, we can write $C_{\{1,\ldots,n-1\}} = C' + D$. To prove $\lambda_1 \le \lambda'_1$, let $\mathbf{e}' = [e'_1, \ldots, e'_{n-1}]^\intercal$ be the eigenvector of C' corresponding to the eigenvalue λ'_1, and consider the vector $\mathbf{e} = [e'_1, e'_2, \ldots, e'_{n-1}, -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}]^\intercal$. Then, $C \cdot \mathbf{e} = [\begin{smallmatrix}\mathbf{y}\\ 0\end{smallmatrix}]$, where

$$\begin{aligned}\mathbf{y} &= -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}[c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal + C_{\{1,\ldots,n-1\}} \cdot \mathbf{e}' \\ &= -\frac{1}{c_{n,n}}\sum_{i=1}^{n-1} e'_i c_{i,n}[c_{1,n}, c_{2,n}, \ldots, c_{n-1,n}]^\intercal + D \cdot \mathbf{e}' + C' \cdot \mathbf{e}' \\ &= C' \cdot \mathbf{e}'.\end{aligned}$$

Thus, $C \cdot \mathbf{e} = [\lambda'_1 e'_1, \lambda'_1 e'_2, \ldots, \lambda'_1 e'_{n-1}, 0]^\intercal = \lambda'_1[e'_1, e'_2, \ldots, e'_{n-1}, 0]^\intercal \le \lambda'_1 \|\mathbf{e}\|_2$, which by Rayleigh-Ritz bounds implies that $\lambda_1 \le \lambda'_1$. ∎

Using the above two lemmas, we now prove Lemma 13.

Proof of Lemma 13. Since

$$\frac{(\mathbf{b}_S^L)^\intercal (C_S^L)^{-1}\mathbf{b}_S^L}{(\mathbf{b}_S^L)^\intercal \mathbf{b}_S^L} \le \max_{\mathbf{x}} \frac{\mathbf{x}^\intercal (C_S^L)^{-1}\mathbf{x}}{\mathbf{x}^\intercal\mathbf{x}} \;=\; \lambda_{\max}((C_S^L)^{-1}) \;=\; \frac{1}{\lambda_{\min}(C_S^L)},$$

we can use Definition 2 to obtain that

$$\gamma_{U,k} \ge \min_{(L\subseteq U, S: |S|\le k, S\cap L=\emptyset)} \lambda_{\min}(C_S^L).$$

Next, we relate $\lambda_{\min}(C_S^L)$ with $\lambda_{\min}(C_{L\cup S})$, using repeated applications of Lemmas 14 and 15. Let $L = \{X_1, \ldots, X_\ell\}$; for each i, define $L_i = \{X_1, \ldots, X_i\}$, and let $C^{(i)}$ be the covariance matrix of the random variables $\{\mathrm{Res}(X, L \setminus L_i) \mid X \in S \cup L_i\}$, and $C_\rho^{(i)}$ the covariance matrix after normalizing all its variables to unit variance. Then, Lemma 14 implies that for each i, $\lambda_{\min}(C^{(i)}) \le \lambda_{\min}(C_\rho^{(i)})$, and Lemma 15 shows that $\lambda_{\min}(C_\rho^{(i)}) \le \lambda_{\min}(C^{(i-1)})$ for each $i > 0$. Combining these inequalities inductively for all i, we obtain that

$$\lambda_{\min}(C_S^L) = \lambda_{\min}(C_\rho^{(0)}) \;\ge\; \lambda_{\min}(C^{(\ell)}) \;=\; \lambda_{\min}(C_{L\cup S}) \;\ge\; \lambda_{\min}(C, |L\cup S|).$$

Finally, since $|S| \le k$ and $L \subseteq U$, we obtain $\gamma_{U,k} \ge \lambda_{\min}(C, k + |U|)$. ∎

3.2 Forward Regression

We now use our approximate submodularity framework along with the result of Lemma 13 to achieve theoretical performance bounds for Forward Regression and Orthogonal Matching Pursuit, which are widely used in practice. We also analyze the Oblivious algorithm, one of the simplest greedy algorithms for subset selection. Throughout the remainder of this section, we use $\mathrm{OPT} = \max_{S:|S|=k} R^2_{Z,S}$ to denote the optimum R^2 value achievable by any set of size k.

We begin with an analysis of Forward Regression, which is the standard algorithm used by many researchers in medical, social, and economic domains.[6]

Algorithm 2 The Forward Regression (also called Forward Selection) algorithm.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i + 1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $R^2_{Z,S_i\cup\{X_{i+1}\}}$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
4: Output S_k.

Notice that Forward Regression is exactly the special case of the general Nemhauser Greedy Algorithm (Algorithm 1) applied to the R^2 objective.

Our main result is the following theorem.

6. There is some inconsistency in the literature about naming of greedy algorithms. Forward Regression is sometimes also referred to as Orthogonal Matching Pursuit (OMP). We choose the nomenclature consistent with Miller (2002) and Tropp (2004).

Theorem 16 *The set S^{FR} selected by Forward Regression has the following approximation guarantees:*

$$\begin{aligned} R^2_{Z,S^{FR}} &\geq (1-\mathrm{e}^{-\gamma_{S^{FR},k}})\cdot OPT \\ &\geq (1-\mathrm{e}^{-\lambda_{\min}(C,2k)})\cdot OPT \\ &\geq (1-\mathrm{e}^{-\lambda_{\min}(C,k)})\cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)})\cdot OPT. \end{aligned}$$

The first inequality is just an application of Theorem 6 to the R^2 objective, and the second inequality follows directly from Lemma 13 by noticing that $|S^{\mathrm{FR}}| = k$. Thus, our proof will focus on the third inequality, which relates the performance measured with respect to the smallest k-sparse eigenvalue to that measured with respect to the smallest $2k$-sparse eigenvalue. We begin with a general lemma that bounds the amount by which the R^2 value of a set and the sum of R^2 values of its elements can differ.

Lemma 17 *Let C and $\mathbf{b}$ be the covariance matrix and covariance vector corresponding to a predictor variable Z and a set S of random variables $X_1, X_2, \ldots, X_n$ that are normalized to have zero mean and unit variance. Then,*

$$\frac{1}{\lambda_{\max}(C)}\sum_{i=1}^{n} R^2_{Z,X_i} \leq R^2_{Z,\{X_1,\ldots,X_n\}} \leq \frac{1}{\gamma_{\emptyset,n}}\sum_{i=1}^{n} R^2_{Z,X_i} \leq \frac{1}{\lambda_{\min}(C)}\sum_{i=1}^{n} R^2_{Z,X_i}.$$

Proof. Let the eigenvalues of C^{-1} be $\lambda'_1 \leq \lambda'_2 \leq \ldots \leq \lambda'_n$, with corresponding orthonormal eigenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}$. We write $\mathbf{b}$ in the basis $\{\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}\}$ as $\mathbf{b} = \sum_i \beta_i \mathbf{e_i}$. Then,

$$R^2_{Z,\{X_1,\ldots,X_n\}} = \mathbf{b}^\intercal C^{-1}\mathbf{b} = \sum_i \beta_i^2 \lambda'_i.$$

Because $\lambda'_1 \leq \lambda'_i$ for all i, we get $\lambda'_1 \sum_i \beta_i^2 \leq R^2_{Z,\{X_1,\ldots,X_n\}}$, and $\sum_i \beta_i^2 = \mathbf{b}^\intercal\mathbf{b} = \sum_i R^2_{Z,X_i}$, because the length of the vector $\mathbf{b}$ is independent of the basis it is written in. Also, by definition of the submodularity ratio, $R^2_{Z,\{X_1,\ldots,X_n\}} \leq \frac{\sum_i \beta_i^2}{\gamma_{\emptyset,n}}$. Finally, because $\lambda'_1 = \frac{1}{\lambda_{\max}(C)}$, and using Lemma 13, we obtain the result. ∎

The next lemma relates the optimal R^2 value using k elements to the optimal R^2 using $k' < k$ elements.

Lemma 18 *For each k, let $S^*_k \in \mathrm{argmax}_{|S|\leq k} R^2_{Z,S}$ be an optimal subset of at most k variables. Then, for any $k' = \Theta(k)$ such that $\frac{1}{\lambda_{\min}(C,k)} < k' < k$, we have that $R^2_{Z,S^*_{k'}} \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{k'}{k})^{1/\lambda_{\min}(C,k)})$, for large enough k. In particular, $R^2_{Z,S^*_{k/2}} \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)})$, for large enough k.*

Proof. We first prove that $R^2_{Z,S^*_{k-1}} \geq (1-\frac{1}{k\lambda_{\min}(C,k)})R^2_{Z,S^*_k}$. Let $T = \mathrm{Res}(Z, S^*_k)$; then, $\mathrm{Cov}[X_i, T] = 0$ for all $X_i \in S^*_k$, and $Z = T + \sum_{X_i\in S^*_k} \alpha_i X_i$, where $\alpha = (\alpha_i) = C^{-1}_{S^*_k} \cdot \mathbf{b}_{S^*_k}$ are

the optimal regression coefficients. We write $Z' = Z - T$. For any $X_j \in S^*_k$, by definition of R^2, we have that

$$R^2_{Z',S^*_k\setminus\{X_j\}} = 1 - \frac{\alpha_j^2 \mathrm{Var}[X_j]}{\mathrm{Var}[Z']} = 1 - \frac{\alpha_j^2}{\mathrm{Var}[Z']};$$

in particular, this implies that $R^2_{Z',S^*_{k-1}} \geq 1 - \frac{\alpha_j^2}{\mathrm{Var}[Z']}$ for all $X_j \in S^*_k$.

Focus now on j minimizing α_j^2, so that $\alpha_j^2 \leq \frac{\|\alpha\|_2^2}{k}$. As in the proof of Lemma 17, by writing α in terms of an orthonormal eigenbasis of $C_{S^*_k}$, one can show that $|\alpha^\intercal C_{S^*_k}\alpha| \geq \|\alpha\|_2^2 \lambda_{\min}(C_{S^*_k})$, or $\|\alpha\|_2^2 \leq \frac{|\alpha^\intercal C_{S^*_k}\alpha|}{\lambda_{\min}(C_{S^*_k})}$. Furthermore, $\alpha^\intercal C_{S^*_k}\alpha = \mathrm{Var}[\sum_{X_i\in S^*_k} \alpha_i X_i] = \mathrm{Var}[Z']$, so $R^2_{Z',S^*_{k-1}} \geq 1 - \frac{1}{k\lambda_{\min}(C_{S^*_k})}$. Finally, by definition, $R^2_{Z',S^*_k} = 1$, so

$$\frac{R^2_{Z,S^*_{k-1}}}{R^2_{Z,S^*_k}} \geq \frac{R^2_{Z',S^*_{k-1}}}{R^2_{Z',S^*_k}} \geq 1 - \frac{1}{k\lambda_{\min}(C_{S^*_k})} \geq 1 - \frac{1}{k\lambda_{\min}(C,k)}.$$

Now, applying this inequality repeatedly, we get

$$R^2_{Z,S^*_{k'}} \geq R^2_{Z,S^*_k} \cdot \prod_{i=k'+1}^{k} (1 - \frac{1}{i\lambda_{\min}(C,i)}).$$

Let $t = \lceil 1/\lambda_{\min}(C,k)\rceil$, so that the previous bound implies $R^2_{Z,S^*_{k'}} \geq R^2_{Z,S^*_k} \cdot \prod_{i=k'+1}^{k} \frac{i-t}{i}$. Most of the terms in the product telescope, giving us a bound of $R^2_{Z,S^*_k} \cdot \prod_{i=1}^{t} \frac{k'-t+i}{k-t+i}$. Since $\prod_{i=1}^{t} \frac{k'-t+i}{k-t+i}$ converges to $(\frac{k'}{k})^t$ with increasing k (keeping t constant), we get that for large k,

$$R^2_{Z,S^*_{k'}} \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{k'}{k})^t) \geq R^2_{Z,S^*_k} \cdot \Theta((\frac{k'}{k})^{1/\lambda_{\min}(C,k)}).$$

This completes the proof. ∎

Using the above lemmas, we now prove the main theorem.

Proof of Theorem 16. As mentioned earlier, the first inequality is a direct corollary of Theorem 6, obtained by replacing f with the R^2 function. The second inequality follows directly from Lemma 13 and the fact that $|S^{\mathrm{FR}}| = k$.

By applying the above result after $k/2$ iterations, we obtain $R^2_{Z,S^{\mathrm{NG}}_{k/2}} \geq (1-\mathrm{e}^{-\lambda_{\min}(C,k)})\cdot R^2_{Z,S^*_{k/2}}$. Now, using Lemma 18 and monotonicity of R^2, we get

$$R^2_{Z,S^{\mathrm{NG}}_k} \geq R^2_{Z,S^{\mathrm{NG}}_{k/2}} \geq (1-\mathrm{e}^{-\lambda_{\min}(C,k)})\cdot\Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)})\cdot R^2_{Z,S^*_k},$$

proving the third inequality. ∎

Algorithm 3 The Orthogonal Matching Pursuit algorithm.

1: Initialize $S_0 = \emptyset$.
2: **for** each iteration $i+1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $|\mathrm{Cov}[\mathrm{Res}(Z, S_i), X_{i+1}]|$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
4: Output S_k.

3.3 Orthogonal Matching Pursuit

The second greedy algorithm we analyze is Orthogonal Matching Pursuit (OMP), frequently used in signal processing domains.

By applying similar techniques as in the previous section, we can also obtain approximation bounds for OMP. We start by proving the following lemma that lower-bounds the variance of the residual of a variable.

Lemma 19 *Let A be the $(n+1) \times (n+1)$ covariance matrix of the normalized variables $Z, X_1, X_2, \ldots, X_n$. Then* $\mathrm{Var}[\mathrm{Res}(Z, \{X_1, \ldots, X_n\})] \geq \lambda_{\min}(A)$.

Proof. The matrix A is of the form $A = \begin{pmatrix} 1 & \mathbf{b}^\intercal \\ \mathbf{b} & C \end{pmatrix}$. We use $A[i,j]$ to denote the matrix obtained by removing the i^{th} row and j^{th} column of A, and similarly for C. Recalling that the (i,j) entry of C^{-1} is $\frac{(-1)^{i+j}\det(C[i,j])}{\det(C)}$, and developing the determinant of A by the first row and column, we can write

$$\begin{aligned}
\det(A) &= \sum_{j=1}^{n+1} (-1)^{1+j} a_{1,j} \det(A[1,j]) \\
&= \det(C) + \sum_{j=1}^{n} (-1)^j b_j \det(A[1, j+1]) \\
&= \det(C) + \sum_{j=1}^{n} (-1)^j b_j \sum_{i=1}^{n} (-1)^{i+1} b_i \det(C[i,j]) \\
&= \det(C) - \sum_{j=1}^{n} \sum_{i=1}^{n} (-1)^{i+j} b_i b_j \det(C[i,j]) \\
&= \det(C)(1 - \mathbf{b}^\intercal C^{-1} \mathbf{b}).
\end{aligned}$$

Therefore, using that $\mathrm{Var}[Z] = 1$,

$$\mathrm{Var}[\mathrm{Res}(Z, \{X_1, \ldots, X_n\})] \;=\; \mathrm{Var}[Z] - \mathbf{b}^\intercal C^{-1} \mathbf{b} \;=\; \frac{\det(A)}{\det(C)}.$$

Because $\det(A) = \prod_{i=1}^{n+1} \lambda_i^A$ and $\det(C) = \prod_{i=1}^{n} \lambda_i^C$, and $\lambda_1^A \leq \lambda_1^C \leq \lambda_2^A \leq \lambda_2^C \leq \ldots \leq \lambda_{n+1}^A$ by the eigenvalue interlacing theorem, we get that $\frac{\det(A)}{\det(C)} \geq \lambda_1^A$, proving the lemma. ∎

The above lemma, along with an analysis similar to the proof of Theorem 16, can be used to prove the following approximation bounds for OMP:

Theorem 20 *The set S^{OMP} selected by orthogonal matching pursuit has the following approximation guarantees:*

$$\begin{aligned}
R^2_{Z,S^{OMP}} &\geq (1 - \mathrm{e}^{-(\gamma_{S^{OMP},k} \cdot \lambda_{\min}(C,2k))}) \cdot OPT \\
&\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \cdot OPT \\
&\geq (1 - \mathrm{e}^{-\lambda_{\min}(C,k)^2}) \cdot \Theta((\frac{1}{2})^{1/\lambda_{\min}(C,k)}) \cdot OPT.
\end{aligned}$$

Proof. We begin by proving the first inequality. Using notation similar to that in the proof of Theorem 16, we let S_k^* be the optimum set of k variables, S_i^{OMP} the set of variables chosen by OMP in the first i iterations, and $S_i = S_k^* \setminus S_i^{\mathrm{OMP}}$. For each $X_j \in S_i$, let $X'_j = \mathrm{Res}(X_j, S_i^{\mathrm{OMP}})$ be the residual of X_j conditioned on S_i^{OMP}, and write $S'_i = \{X'_j \mid X_j \in S_i\}$.

Consider some iteration $i+1$ of OMP. We will show that at least one of the X'_i is a good candidate in this iteration. Let ℓ maximize R^2_{Z,X'_ℓ}, i.e., $\ell \in \mathrm{argmax}_{(j:X'_j \in S'_i)} R^2_{Z,X'_j}$. By Lemma 19,

$$\mathrm{Var}[X'_\ell] \geq \lambda_{\min}(C_{S_G^i \cup \{X'_\ell\}}) \;\geq\; \lambda_{\min}(C, 2k).$$

The OMP algorithm chooses a variable X_m to add which maximizes $|\mathrm{Cov}[\mathrm{Res}(Z, S_G^i), X_m]|$. Thus, X_m maximizes

$$\mathrm{Cov}[\mathrm{Res}(Z, S_G^i), X_m]^2 = \mathrm{Cov}[Z, \mathrm{Res}(X_m, S_G^i)]^2 \;=\; R^2_{Z,\mathrm{Res}(X_m,S_G^i)} \cdot \mathrm{Var}[\mathrm{Res}(X_m, S_G^i)].$$

In particular, this implies

$$\begin{aligned}
R^2_{Z,\mathrm{Res}(X_m,S_G^i)} &\geq R^2_{Z,X'_\ell} \cdot \frac{\mathrm{Var}[X'_\ell]}{\mathrm{Var}[\mathrm{Res}(X_m, S_G^i)]} \\
&\geq R^2_{Z,X'_\ell} \cdot \frac{\lambda_{\min}(C,2k)}{\mathrm{Var}[\mathrm{Res}(X_m, S_G^i)]} \;\geq\; R^2_{Z,X'_\ell} \cdot \lambda_{\min}(C,2k),
\end{aligned}$$

because $\mathrm{Var}[\mathrm{Res}(X_m, S_G^i)] \leq 1$. As in the proof of Theorem 6, $R^2_{Z,X'_\ell} \geq \frac{\gamma_{S^{\mathrm{OMP}},k}}{k} \cdot R^2_{Z,S'_i}$, so $R^2_{Z,\mathrm{Res}(X_m,S_G^i)} \geq R^2_{Z,S'_i} \cdot \frac{\lambda_{\min}(C,2k) \cdot \gamma_{S^{\mathrm{OMP}},k}}{k}$. With the same definition of $A(i)$ as in the proof of Theorem 6, we get that $A(i+1) \geq \frac{\lambda_{\min}(C,2k)\gamma_{S^{\mathrm{OMP}},k}}{k} \cdot (\mathrm{OPT} - \sum_{j=1}^{i} A(j))$. An inductive proof now shows that

$$R^2_{Z,S_G} \;=\; \sum_{i=1}^{k} A(i) \;\geq\; (1 - \mathrm{e}^{-\lambda_{\min}(C,2k) \cdot \gamma_{S^{\mathrm{OMP}},k}}) \cdot R^2_{Z,S_k^*}.$$

The proofs of the other two inequalities follow the same pattern as the proof for Forward Regression. ∎

3.4 Oblivious Algorithm

As a baseline, we also consider a greedy algorithm which completely ignores C and simply selects the k variables individually most correlated with Z.

Lemma 17 immediately implies a simple bound for the oblivious algorithm:

Algorithm 4 The oblivious algorithm.

1: Sort the X_i by non-increasing b_i values.
2: Return $\{X_1, X_2, \ldots, X_K\}$.

Theorem 21 *The set S^{OBL} selected by the oblivious algorithm has the following approximation guarantees:*

$$R^2_{Z,S^{OBL}} \;\geq\; \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot OPT \;\geq\; \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)} \cdot OPT.$$

Proof. Let S be the set chosen by the oblivious algorithm, and S^*_k the optimum set of k variables. By definition of the oblivious algorithm, $\sum_{i\in S} R^2_{Z,X_i} \geq \sum_{i\in S^*_k} R^2_{Z,X_i}$, so using Lemma 17, we obtain that

$$R^2_{Z,S} \geq \frac{\sum_{i\in S} R^2_{Z,X_i}}{\lambda_{\max}(C,k)} \;\geq\; \frac{\sum_{i\in S^*_k} R^2_{Z,X_i}}{\lambda_{\max}(C,k)} \;\geq\; \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} R^2_{Z,S^*_k}.$$

The second inequality of the theorem follows directly from Lemma 13. ■

4. Dictionary Selection Bounds

To demonstrate the wider applicability of the approximate submodularity framework, we next obtain a tighter analysis for two greedy algorithms for the dictionary selection problem, introduced by Krause and Cevher (2010).

The Dictionary Selection problem generalizes the Subset Selection problem by considering s predictor variables $Z_1, Z_2, \ldots, Z_s$. The goal is to select a dictionary D of d observation variables, to optimize the average R^2 fit for the Z_i using at most k vectors from D for each. Formally, the Dictionary Selection problem is defined as follows:

Definition 22 (Dictionary Selection) *Given all pairwise covariances among the Z_j and X_i variables, as well as parameters d and k, find a set D of at most d variables from $\{X_1, \ldots, X_n\}$ maximizing*

$$F(D) = \sum_{j=1}^{s} \max_{S\subset D, |S|=k} R^2_{Z_j,S}.$$

4.1 The Algorithm $\mathrm{SDS_{MA}}$

The $\mathrm{SDS_{MA}}$ algorithm generalizes the oblivious greedy algorithm to the problem of Dictionary Selection. It replaces the $R^2_{Z_j,S}$ term in Definition 22 with its modular approximation $f(Z_j, S) = \sum_{i\in S} R^2_{Z_j,X_i}$. Thus, it greedily tries to maximize the function $\hat{F}(D) = \sum_{j=1}^{s} \max_{S\subset D,|S|=k} f(Z_j, S)$, over sets D of size at most d; the inner maximum can be computed efficiently using the oblivious algorithm.

Using Lemma 17, we obtain the following multiplicative approximation guarantee for $\mathrm{SDS_{MA}}$:

Algorithm 5 The SDS_{MA} algorithm for dictionary selection.

1: Initialize $D_0 = \emptyset$.
2: **for** each iteration $i + 1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $\hat{F}(D \cup \{X_m\})$, and set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
4: Output D_d.

Theorem 23 *Let D^{MA} be the dictionary selected by the SDS_{MA} algorithm, and D^* the optimum dictionary of size $|D| \leq d$, with respect to the objective $F(D)$ from Definition 22. Then,*

$$F(D^{MA}) \geq \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)}(1-\frac{1}{\mathrm{e}}) \cdot F(D^*) \;\geq\; \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)}(1-\frac{1}{\mathrm{e}}) \cdot F(D^*).$$

Proof. Let $\hat{D}$ be a dictionary of size d maximizing $\hat{F}(D)$. Because $f(Z_j, S)$ is monotone and modular in S, $\hat{F}$ is a monotone, submodular function. Hence, using the submodularity results of Nemhauser et al. (1978) and the optimality of $\hat{D}$ for $\hat{F}$,

$$\hat{F}(D^{\mathrm{MA}}) \geq \hat{F}(\hat{D}) \cdot (1-\frac{1}{\mathrm{e}}) \;\geq\; \hat{F}(D^*) \cdot (1-\frac{1}{\mathrm{e}}).$$

Now, by applying Lemma 17 for each Z_j, it is easy to show that $\hat{F}(D^*) \geq \gamma_{\emptyset,k} \cdot F(D^*)$, and similarly $\hat{F}(D^{\mathrm{MA}}) \leq \lambda_{\max}(C,k) \cdot F(D^{\mathrm{MA}})$. Thus we get $F(D^{\mathrm{MA}}) \geq \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)}(1-\frac{1}{\mathrm{e}})F(D^*)$.

The second part now follows from Lemma 13. ■

Note that these bounds significantly improve the previous additive approximation guarantee obtained by Krause and Cevher (2010): $F(D^{\mathrm{MA}}) \geq (1-\frac{1}{\mathrm{e}}) \cdot F(D^*) - (2-\frac{1}{\mathrm{e}}) \cdot k \cdot \mu(C)$. In particular, when $\mu(C) > \Theta(1/k)$, i.e., even just one pair of variables has moderate correlation, the approximation guarantee of Krause and Cevher becomes trivial.

4.2 The Algorithm $\mathrm{SDS_{OMP}}$

We also obtain a multiplicative approximation guarantee for the greedy $\mathrm{SDS_{OMP}}$ algorithm, introduced by Krause and Cevher for dictionary selection. Our bounds for $\mathrm{SDS_{OMP}}$ are much stronger than the additive bounds obtained by Krause and Cevher. However, for both our results and theirs, the performance guarantees for $\mathrm{SDS_{OMP}}$ are much weaker than those for $\mathrm{SDS_{MA}}$.

The $\mathrm{SDS_{OMP}}$ algorithm generalizes the Orthogonal Matching Pursuit algorithm for subset selection to the problem of dictionary selection. In each iteration, it adds a new element to the currently selected dictionary by using Orthogonal Matching Pursuit to approximate the estimation of $\max_{|S|=k} R^2_{Z_j,S}$.

We now show how to obtain a multiplicative approximation guarantee for $\mathrm{SDS_{OMP}}$. The following definitions are key to our analysis; the first two are from Definition 22 and

Algorithm 6 The *SDS$_{OMP}$* algorithm for dictionary selection.

1: Initialize $D_0 = \emptyset$.
2: **for** each iteration $i+1 = 1, 2, \ldots$ **do**
3: Let X_{i+1} be a variable maximizing $\sum_{j=1}^{s} R^2_{Z_j, S_{\mathrm{OMP}}(D_i \cup \{X_{i+1}\}, Z_j, k)}$ where $S_{\mathrm{OMP}}(D, Z, k)$ denotes the set selected by Orthogonal Matching Pursuit for predicting Z using k variables from D.
4: Set $S_{i+1} = S_i \cup \{X_{i+1}\}$.
5: Output D_d.

Theorem 23.

$$F(D) = \sum_{j=1}^{s} \max_{S \subset D, |S|=k} R^2_{Z_j,S},$$

$$\hat{F}(D) = \sum_{j=1}^{s} \max_{S \subset D, |S|=k} f(Z_j, S),$$

$$\tilde{F}(D) = \sum_{j=1}^{s} R^2_{Z_j, S_{\mathrm{OMP}}(D, Z_j, k)}.$$

We first prove the following lemma about approximating the function $\hat{F}(D)$ by $\tilde{F}(D)$:

Lemma 24 *For any set D, we have that*

$$\frac{(1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})}{\lambda_{\max}(C,k)} \cdot \hat{F}(D) \leq \tilde{F}(D) \ \leq \ \frac{\hat{F}(D)}{\gamma_{\emptyset,k}}.$$

Proof. Using Theorem 20 and Lemma 17 and summing up over all the Z_j terms, we obtain that

$$\tilde{F}(D) \geq (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \cdot F(D) \ \geq \ (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2}) \frac{\hat{F}(D)}{\lambda_{\max}(C,k)}.$$

Similarly, using Lemma 17 and the fact that $\max_{S \subset D, |S|=k} R^2_{Z_j,S} \geq R^2_{Z_j, S_{OMP}(D, Z_j, k)}$, we have

$$\hat{F}(D) \geq \gamma_{\emptyset,k} \cdot F(D) \ \geq \ \gamma_{\emptyset,k} \cdot \tilde{F}(D). \ \blacksquare$$

Using the above lemma, we now prove the following bound for $\mathrm{SDS}_{\mathrm{OMP}}$:

Theorem 25 *Let D^{OMP} be the dictionary selected by the SDS$_{OMP}$ algorithm, and D^* the optimum dictionary of size $|D| \leq d$, with respect to the objective $F(D)$ from Definition 22. Then,*

$$F(D^{OMP}) \geq F(D^*) \cdot \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1} \ \geq \ F(D^*) \cdot \frac{\lambda_{\min}(C,k)}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1},$$

where $p = \frac{1}{\lambda_{\max}(C,k)} \cdot (1 - \mathrm{e}^{-\lambda_{\min}(C,2k)^2})$.

Proof. Let $\hat{D}$ be the dictionary of size d that maximizes $\hat{F}(D)$. We first prove that $\hat{F}(D^{\mathrm{OMP}})$ is a good approximation to $\hat{F}(\hat{D})$.

Let S_i^{NG} be the variables chosen by $\mathrm{SDS}_{\mathrm{OMP}}$ after i iterations. Define $S_i = \hat{D} \setminus S_i^{\mathrm{NG}}$. By monotonicity of $\hat{F}$, we have that $\hat{F}(S_i \cup S_i^{\mathrm{NG}}) \geq \hat{F}(\hat{D})$.

Let $\hat{X} \in S_i$ be the variable maximizing $\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\})$, and similarly $\tilde{X} \in S_i$ be the variable maximizing $\tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\})$.

Since $\hat{F}$ is a submodular function, it is easy to show (using an argument similar to the proof of Theorem 16) that $\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) - \hat{F}(S_i^{\mathrm{NG}}) \geq \frac{\hat{F}(\hat{D}) - \hat{F}(S_i^{\mathrm{NG}})}{d}$.

Now, using Lemma 24 above, and the optimality of $\tilde{X}$ for $\tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\})$, we obtain that

$$\frac{1}{\gamma_{\emptyset,k}} \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \geq \tilde{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \ \geq \ \tilde{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) \ \geq \ p \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}).$$

Thus, $\hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) \geq p \cdot \gamma_{\emptyset,k} \cdot \hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\})$, or

$$\hat{F}(S_i^{\mathrm{NG}} \cup \{\tilde{X}\}) - \hat{F}(S_i^{\mathrm{NG}}) \geq p \cdot \gamma_{\emptyset,k} \cdot (\hat{F}(S_i^{\mathrm{NG}} \cup \{\hat{X}\}) - \hat{F}(S_i^{\mathrm{NG}})) - (1 - p \cdot \gamma_{\emptyset,k}) \hat{F}(S_i^{\mathrm{NG}}).$$

Define $A(i) = \hat{F}(S_i^{\mathrm{NG}}) - \hat{F}(S_{i-1}^{\mathrm{NG}})$ to be the gain, with respect to $\hat{F}$, obtained from the variable chosen by $\mathrm{SDS}_{\mathrm{OMP}}$ in iteration i. Then $\hat{F}(D^{\mathrm{OMP}}) = \sum_{i=1}^{d} A(i)$. From the preceding paragraphs, we obtain

$$A(i+1) \geq \frac{p \cdot \gamma_{\emptyset,k}}{d} \cdot (\hat{F}(\hat{D}) - (1 + \frac{d}{p \cdot \gamma_{\emptyset,k}} - d) \sum_{j=1}^{i} A(j)).$$

Since the above inequality holds for each iteration $i = 1, 2, \ldots, d$, a simple inductive proof shows that

$$\hat{F}(\hat{D}) - \sum_{i=1}^{d} A(i) \leq \hat{F}(\hat{D}) \cdot (1 - \frac{p\gamma_{\emptyset,k}}{d})^d + (d - dp\gamma_{\emptyset,k}) \cdot \sum_{i=1}^{d} A(i).$$

Rearranging the terms and simplifying, we get that

$$\hat{F}(D^{\mathrm{OMP}}) = \sum_{i=1}^{d} A(i) \ \geq \ \hat{F}(\hat{D}) \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - dp\gamma_{\emptyset,k} + 1} \ \geq \ \hat{F}(D^*) \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - dp\gamma_{\emptyset,k} + 1},$$

where the last inequality is due to the optimality of $\hat{D}$ for $\hat{F}$.

Now, using Lemma 17 for each Z_j term, it can be easily seen that $\hat{F}(D^*) \geq \gamma_{\emptyset,k} \cdot F(D^*)$. Similarly, using Lemma 3.3 on the set D^{OMP}, we have $F(D^{\mathrm{OMP}}) \geq \frac{1}{\lambda_{\max}(C,k)} \cdot \hat{F}(D^{\mathrm{OMP}})$.

Using the above inequalities, we therefore get the desired bound

$$F(D^{\mathrm{OMP}}) \geq F(D^*) \cdot \frac{\gamma_{\emptyset,k}}{\lambda_{\max}(C,k)} \cdot \frac{(1 - \mathrm{e}^{-(p \cdot \gamma_{\emptyset,k})})}{d - d \cdot p \cdot \gamma_{\emptyset,k} + 1}.$$

The second inequality of the Theorem now follows directly from Lemma 13. $\blacksquare$

5. Experiments

In this section, we evaluate Forward Regression (FR) and OMP empirically, on two real-world and one synthetic data set. We compare the two algorithms against an optimal solution (OPT), computed using exhaustive search, the oblivious greedy algorithm (OBL), and the L1-regularization/Lasso (L1) algorithm (in the implementation of Koh et al. (2008)). Beyond the algorithms' performance, we also compute the various spectral parameters from which we can derive lower bounds. Specifically, these are

1. the submodularity ratio: $\gamma_{S^{\mathrm{FR}},k}$, where S^{FR} is the subset selected by forward regression.
2. the smallest sparse eigenvalues $\lambda_{\min}(C,k)$ and $\lambda_{\min}(C,2k)$. (In some cases, computing $\lambda_{\min}(C,2k)$ was not computationally feasible due to the problem size.)
3. the sparse inverse condition number $\kappa(C,k)^{-1}$. As mentioned earlier, the sparse inverse condition number $\kappa(C,k)$ is strongly related to the Restricted Isometry Property in (Candès et al., 2005).
4. the smallest eigenvalue $\lambda_{\min}(C) = \lambda_{\min}(C,n)$ of the entire covariance matrix.

The aim of our experiments is twofold: First, we wish to evaluate which among the submodular and spectral parameters are good predictors of the performance of greedy algorithms in practice. Second, we wish to highlight how the theoretical bounds for subset selection algorithms reflect on their actual performance. Our analytical results predict that Forward Regression should outperform OMP, which in turn outperforms Oblivious. For Lasso, it is not known whether strong multiplicative bounds, like the ones we proved for Forward Regression or OMP, can be obtained.

5.1 Data Sets

Because several of the spectral parameters (as well as the optimum solution) are **NP**-hard to compute, we restrict our experiments to data sets with $n \leq 30$ features, from which $k \leq 8$ are to be selected. We stress that the greedy algorithms themselves are very efficient, and the restriction on data set sizes is only intended to allow for an adequate evaluation of the results.

Each data set contains $m > n$ samples, from which we compute the empirical covariance matrix (analogous to the Gram matrix in sparse approximation) between all observation variables and the predictor variable; we then normalize it to obtain C and $\mathbf{b}$. We evaluate the performance of all algorithms in terms of their R^2 fit; thus, we implicitly treat C and $\mathbf{b}$ as the ground truth, and also do not separate the data sets into training and test cases.

Our data sets are the *Boston Housing Data*, a data set of *World Bank Development Indicators*, and a synthetic data set generated from a distribution similar to the one used by Zhang (2008). The *Boston Housing Data* (available from the UCI Machine Learning Repository) is a small data set frequently used to evaluate ML algorithms. It comprises $n = 15$ features (such as crime rate, property tax rates, etc.) and $m = 516$ observations. Our goal is to predict housing prices from these features. The *World Bank Data* (available from `http://databank.worldbank.org`) contains an extensive list of socio-economic and

health indicators of development, for many countries and over several years. We choose a subset of $n = 29$ indicators for the years 2005 and 2006, such that the values for all of the $m = 65$ countries are known for each indicator. (The data set does not contain all indicators for each country.) We choose to predict the average life expectancy for those countries.

To perform tests in a controlled fashion, we also generate random instances from a known distribution similar to one used by Zhang (2008): There are $n = 29$ features, and $m = 100$ data points are generated from a joint Gaussian distribution with moderately high correlations of 0.6. The target vector is obtained by generating coefficients uniformly from 0 to 10 along each dimension, and adding noise with variance $\sigma^2 = 0.1$. Notice that the target vector is not truly sparse. As for the other two data sets, the covariances are then taken to be the empirical ones of the generated data. The plots we show are the average R^2 values for 20 independent runs of the experiment.

5.2 Results

We run the different subset selection algorithms for values of k from 2 through 8, and plot the R^2 values for the selected sets. When including all of the features, the R^2 value is close to 1 in all data sets, implying that nearly all of the variance in the function to be predicted can be explained by the features.

Figures 1, 3 and 5 show the results for the three data sets. The main insight is that on all data sets, Forward Regression performs optimally or near-optimally, and OMP is only slightly worse. This is despite the fact that (as we discuss shortly) the spectral properties would not necessarily predict such near-optimal performance. Lasso performs somewhat worse on all data sets, and, not surprisingly, the baseline oblivious algorithm performs even worse. The last fact implies that the optimal solution is non-trivial in that it must account for correlation between the observation variables. The order of performance of the greedy algorithms match the order of the strength of the theoretical bounds we derived for them.

On the World Bank data (Figure 3), all algorithms perform quite well with just 2–3 features already. The main reason is that adolescent birth rate is by itself highly predictive of life expectancy, so the first feature selected by all algorithms already contributes high R^2 value.

Figures 2, 4 and 6 show the different spectral quantities for the data sets, for varying values of k. Both of the real-world data sets are nearly singular, as evidenced by the small $\lambda_{\min}(C)$ values. In fact, the near-singularities manifest themselves for small values of k already; in particular, since $\lambda_{\min}(C,2)$ is already small, we observe that there are pairs of highly correlated observations variables in the data sets. Thus, the bounds on approximation we would obtain by considering merely $\lambda_{\min}(C,k)$ or $\lambda_{\min}(C,2k)$ would be quite weak. Notice, however, that they are still quite a bit stronger than the inverse condition number $\kappa(C,k)^{-1}$: this bound — which is closely related to the RIP property frequently at the center of sparse approximation analysis — takes on much smaller values, and thus would be an even looser bound than the eigenvalues.

On the other hand, the submodularity ratios $\gamma_{S^{\mathrm{FR}},k}$ for all the data sets are much larger than the other spectral quantities (almost 5 times larger, on average, than the corresponding $\lambda_{\min}(C)$ values). Notice that unlike the other quantities, the submodularity ratios are not

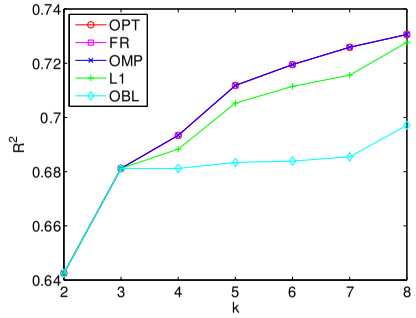


Figure 1: Boston Housing R^2

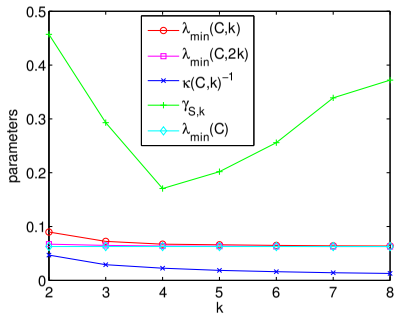


Figure 2: Boston Housing parameters

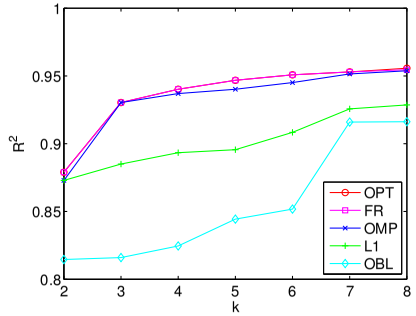


Figure 3: World Bank R^2

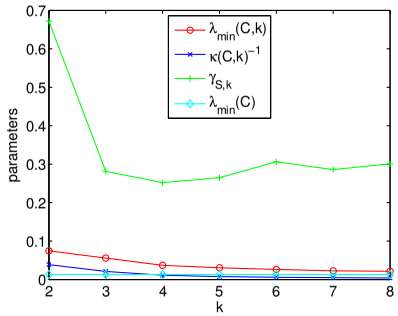


Figure 4: World Bank parameters

monotonically decreasing in k — this is due to the dependency of $\gamma_{S^{\mathrm{FR}},k}$ on the set S^{FR}, which is different for every k.

The discrepancy between the small values of the eigenvalues and the good performance of all algorithms shows that bounds based solely on eigenvalues can sometimes be loose. Significantly better bounds are obtained from the submodularity ratio $\gamma_{S^{\mathrm{FR}},k}$, which takes on values above 0.2, and significantly larger in many cases. While not entirely sufficient to explain the performance of the greedy algorithms, it shows that the near-singularities of C do not align unfavorably with $\mathbf{b}$, and thus do not provide an opportunity for strong supermodular behavior that adversely affects greedy algorithms.

The synthetic data set we generated is somewhat further from singular, with $\lambda_{\min}(C) \approx 0.11$. However, the same patterns persist: the simple eigenvalue based bounds, while somewhat larger for small k, still do not fully predict the performance of greedy algorithms, whereas the submodularity ratio here is close to 1 for all values of k. This shows that the near-singularities do not at all provide the possibility of strongly supermodular benefits of sets of variables. Indeed, the plot of R^2 values on the synthetic data is concave, an indicator of submodular behavior of the function.

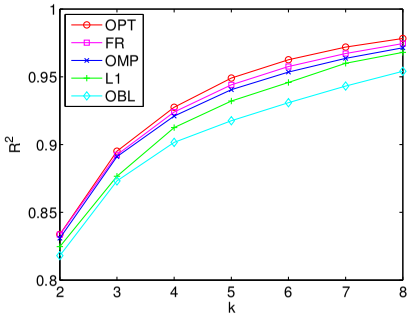


Figure 5: Synthetic Data R^2

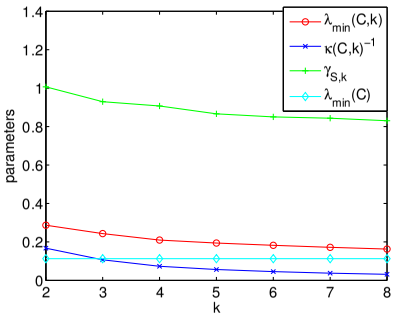


Figure 6: Synthetic Data parameters

The above observations suggest that bounds based on the submodularity ratio are better predictors of the performance of greedy algorithms, followed by bounds based on the sparse eigenvalues, and finally those based on the condition number or RIP property.

5.3 Narrowing the gap between theory and practice

Our theoretical bounds, though much stronger than previous results, still do not fully predict the observed near-optimal performance of Forward Regression and OMP on the real-world datasets. In particular, for Forward Regression, even though the submodularity ratio is less than 0.4 for most cases, implying a theoretical guarantee of roughly $1 - e^{-0.4} \approx 33\%$, the algorithm still achieves near-optimal performance. While gaps between worst-case bounds and practical performance are commonplace in algorithmic analysis, they also suggest that there is scope for further improving the analysis, by looking at more fine-grained parameters.

Indeed, a slightly more careful analysis of the proof of Theorem 16 and our definition of the submodularity ratio reveals that we do not really need to calculate the submodularity ratio over all sets S of size k while analyzing the greedy steps of Forward Regression. We can ignore sets S whose submodularity ratio is low, but whose marginal contribution to the current R^2 is only a small fraction (say, at most ϵ). This is because the proof of Theorem 16 shows that for each iteration $i+1$, we only need to consider the submodularity ratio for the set $S_i = S_k^* \setminus S_i^{\mathrm{NG}}$, where S_i^{NG} is the set selected by the greedy algorithm after i iterations, and S_k^* is the optimal k-subset. Thus, if $R^2_{Z,S_i \cup S_i^{\mathrm{NG}}} \leq (1+\epsilon) \cdot R^2_{Z,S_i^{\mathrm{NG}}}$, then the currently selected set must already be within a factor $\frac{1}{1+\epsilon}$ of optimal.

By carefully pruning such sets (using $\epsilon = 0.2$) while calculating the submodularity ratio, we see that the resulting values of $\gamma_{S^{\mathrm{FR}},k}$ are much higher (more than 0.8), thus significantly reducing the gap between the theoretical bounds and experimental results. Table 1 shows the values of $\gamma_{S^{\mathrm{FR}},k}$ obtained using this method.

The results suggest an interesting direction for future work: namely, to characterize for which sets the submodular behavior of R^2 really matters.

Data Set	$k=2$	$k=3$	$k=4$	$k=5$	$k=6$	$k=7$	$k=8$
Boston	0.9	0.91	1.02	1.21	1.36	1.54	1.74
World Bank	0.8	0.81	0.81	0.81	0.94	1.19	1.40

Table 1: Improved estimates for submodularity ratio

6. Discussion and Concluding Remarks

In this paper, we defined a notion of approximate submodularity. We showed that it naturally captures the performance degradation of the greedy algorithm. As a concrete application of the framework, we connected the submodularity ratio with spectral parameters of the covariance matrix to obtain the strongest known approximation guarantees for the Forward Selection and Orthogonal Matching Pursuit algorithms for regression. As a second example, we gave improved approximation guarantees for known greedy algorithms for dictionary selection.

We believe that our techniques for analyzing greedy algorithms using a notion of "approximate submodularity" are not specific to subset selection and dictionary selection, and could also be used to analyze other problems in compressed sensing and sparse recovery. A natural further direction is hence to identify other applications of the approximate submodularity technique.

While approximation guarantees for the greedy algorithm are perhaps the most widely used consequence of submodularity, they are far from the only one. Some other useful consequences include the following:

1. In *valid utility games* (Vetta, 2002), where utility functions are essentially submodular and interact with each other in certain ways, equilibria always achieve high social welfare.

2. A monotone submodular function can be approximately maximized subject to a Knapsack constraint (Sviridenko, 2004), Matroid constraint (Vondrák, 2008) or combinations thereof (e.g., (Chekuri et al., 2011)).

3. If the function f is submodular, but not necessarily monotone, it can be approximately maximized, with or without a cardinality constraint. Without a cardinality constraint, it can also be exactly minimized.

It would be desirable to verify whether some of these results gracefully degrade when the submodularity ratio is bounded away from 0. The third property (optimization of non-monotone submodular functions) seems unlikely to carry over, as our definition was targeted

at monotone submodular functions. This raises the natural question of whether there is a more general definition of approximate submodularity that retains the positive results of the present work while also yielding an analogue to some or all of the above properties.

Our bicriteria approximation guarantees, trading off a maximization of coverage against a minimization of cost, could be generalized to more general constraints. For instance, Iyer and Bilmes (Iyer and Bilmes, 2013) give bicriteria approximation guarantees for maximizing a submodular function subject to a submodular cost constraint, or minimizing a submodular function subject to a submodular coverage constraint. It is natural to ask whether similar guarantees can be obtained for approximately submodular functions.

As discussed in Remark 4, it is open how well one can approximate the submodularity ratio of a given function f in general; being able to do so would allow one to obtain approximation guarantees at least for specific instances. Alternatively, it may be possible to establish approximation hardness results for computing the submodularity ratio.

The approximation guarantees of the greedy algorithm are worst when the covariance matrix is singular, or close to singular. When the covariance matrix is estimated from data (rather than explicitly given), the natural variance in data generated from joint distributions may keep it from being too close to singular. A detailed investigation would constitute an interesting direction for future work, though to be useful, it would have to provide a lower bound of $\omega(1/\log n)$ on the smallest (sparse) eigenvalue.

Acknowledgments

We would like to thank Andreas Krause, Fei Sha and several anonymous referees for their helpful feedback. This work was supported in part by NSF grant 0540420 (DDDAS-TMRP).

References

Wenruo Bai and Jeffrey A. Bilmes. Greed is still good: Maximizing monotone submodular+supermodular functions, 2018. https://arxiv.org/pdf/1801.07413.pdf.

Gruia Calinescu, Chandra Chekuri, Martin Pál, and Jan Vondrák. Maximizing a submodular set function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6): 1740–1766, 2011.

Emmanuel J. Candès and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Emmanuel J. Candès, Justin Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

Chandra Chekuri, Jan Vondrák, and Rico Zenklusen. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *Proc. 43rd ACM Symp. on Theory of Computing*, pages 783–792, 2011.

Flavio Chierichetti, Abhimanyu Das, Anirban Dasgupta, and Ravi Kumar. Approximate modularity. In *Proc. 56th IEEE Symp. on Foundations of Computer Science*, pages 1143–1162, 2015.

Michele Conforti and Gérard Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the rado-edmonds theorem. *Discrete Applied Mathematics*, 7:251–274, 1984.

Abhimanyu Das and David Kempe. Algorithms for subset selection in linear regression. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 97–108, 2008.

Abhimanyu Das, Anirban Dasgupta, and Ravi Kumar. Selecting diverse features via spectral regularization. In *Proc. 26th Advances in Neural Information Processing Systems*, pages 1592–1600, 2012.

George M. Diekhoff. *Statistics for the Social and Behavioral Sciences*. Brown & Benchmark, 2002.

David L. Donoho. For most large underdetermined systems of linear equations, the minimal 11-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics*, 59:1207–1223, 2005.

Ethan R. Elenberg, Rajiv Khanna, Alexandros G. Dimakis, and Sahand Negahban. Restricted strong convexity implies weak submodularity. *Annals of Statistics*, 2018. To appear.

Anna Gilbert, S. Muthu Muthukrishnan, and Martin Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proc. 14th ACM-SIAM Symp. on Discrete Algorithms*, pages 243–252, 2003.

Alexander Grubb and J. Andrew Bagnell. Speedboost: Anytime prediction with uniform near-optimality. In *Proc. 15thIntl. Conf. on Artificial Intelligence and Statistics*, pages 458–466, 2012.

Rishabh K. Iyer. *Submodular Optimization and Machine Learning: Theoretical Results, Unifying and Scalable Algorithms, and Applications*. PhD thesis, University of Washington, 2015.

Rishabh K. Iyer and Jeff A. Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *Proc. 27th Advances in Neural Information Processing Systems*, pages 2436–2444, 2013.

Rishabh K. Iyer, Stefanie Jegelka, and Jeff A. Bilmes. Curvature and optimal algorithms for learning and minimizing submodular functions. In *Proc. 27th Advances in Neural Information Processing Systems*, pages 2742–2750, 2013.

Richard A. Johnson and Dean W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, 2002.

Kwangmoo Koh, Seung-Jean Kim, and Stephen Boyd. l1_ls: Simple Matlab Solver for l1-regularized Least Squares Problems, 2008. http://www.stanford.edu/ boyd/l1_ls.

Andreas Krause and Volkan Cevher. Submodular dictionary selection for sparse representation. In *Proc. 27th Intl. Conf. on Machine Learning*, pages 567–574, 2010.

Andreas Krause and Daniel Golovin. Submodular function maximization. In Lucas Bordeaux, Youssef Hamadi, and Pushmeet Kohli, editors, *Tractability: Practical Approaches to Hard Problems*, pages 71–104. Cambridge University Press, 2014.

Matt J. Kusner, Wenlin Chen, Quan Zhou, Zhixiang Xu, Kilian Q. Weinberger, and Yixin Chen. Feature-cost sensitive learning with submodular trees of classifiers. In *Proc. 28th AAAI Conf. on Artificial Intelligence*, pages 1949–1945, 2014.

Aurelie C. Lozano, Grzegorz Swirszcz, and Naoki Abe. Grouped orthogonal matching pursuit for variable selection and prediction. In *Proc. 23rd Advances in Neural Information Processing Systems*, pages 1150–1158, 2009.

Alan J. Miller. *Subset Selection in Regression*. Chapman and Hall, second edition, 2002.

Balas K. Natarajan. Sparse approximation solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of the approximations for maximizing submodular set functions. *Mathematical Programming*, 14:265–294, 1978.

Chao Qian, Yang Yu, and Zhi-Hua Zhou. Subset selection by pareto optimization. In *Proc. 29th Advances in Neural Information Processing Systems*, pages 1774–1782, 2015.

Mahyar Salek, Shahin Shayandeh, and David Kempe. You share, I share: Network effects and economic incentives in P2P file-sharing systems. In *Proc. 6th Workshop on Internet and Network Economics (WINE)*, pages 354–365, 2010.

Maxim Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Oper. Res. Lett.*, 32(1):41–43, 2004.

Maxim Sviridenko, Jan Vondrák, and Justin Ward. Optimal approximation for submodular and supermodular optimization with bounded curvature. In *Proc. 26th ACM-SIAM Symp. on Discrete Algorithms*, pages 1134–1148, 2015.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

Joel Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50:2231–2242, 2004.

Joel Tropp. Just relax: Convex programming methods for identifying sparse signals. *IEEE Transactions on Information Theory*, 51:1030–1051, 2006.

Joel Tropp, Anna Gilbert, S. Muthu Muthukrishnan, and Martin Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *Proc. IEEE-ICIP*, pages 37–40, 2003.

Adrian Vetta. Nash equlibria in competitive societies with applications to facility location, traffic routing and auctions. In *Proc. 43rd IEEE Symp. on Foundations of Computer Science*, pages 416–425, 2002.

Jan Vondrák. Optimal approximation for the submodular welfare problem in the value oracle model. In *Proc. 40th ACM Symp. on Theory of Computing*, pages 67–74, 2008.

Jan Vondrák. Submodularity and curvature: the optimal algorithm. *RIMS Kokyuroku Bessatsu*, B23:253–266, 2010.

Laurence A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982.

Tong Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Proc. 22nd Advances in Neural Information Processing Systems*, pages 1921–1928, 2008.

Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine Learning Research*, 7:2451–2457, 2006.

Shuheng Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In *Proc. 23rd Advances in Neural Information Processing Systems*, pages 2304–2312, 2009.

Appendix A. Estimating $\lambda_{\min}(C,k)$

Several of our approximation guarantees are phrased in terms of $\lambda_{\min}(C,k)$. Finding the exact value of $\lambda_{\min}(C,k)$ is **NP**-hard in general; here, we show how to estimate lower and upper bounds. Let $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$ be the eigenvalues of C, and $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_n}$ the corresponding eigenvectors. A first simple bound can be obtained directly from the eigenvalue interlacing theorem: $\lambda_1 \leq \lambda_{\min}(C,k) \leq \lambda_{n-k+1}$.

One case in which good lower bounds on $\lambda_{\min}(C,k)$ can possibly be obtained is when only a small (constant) number of the λ_i are small. The following lemma allows a bound in terms of any λ_j; however, since the running time by the implied algorithm is exponential in j, and the quality of the bound depends on λ_j, it is useful only in the special case when $\lambda_j \gg 0$ for a small constant j.

Lemma 26 *Let V_j be the vector space spanned by the eigenvectors $\mathbf{e_1}, \mathbf{e_2}, \ldots, \mathbf{e_j}$, and define*

$$\beta_j = \max_{\mathbf{y} \in V_j, \mathbf{x} \in \mathbb{R}^n, \|\mathbf{y}\|_2 = \|\mathbf{x}\|_2 = 1, \|\mathbf{x}\|_0 \leq k} |\mathbf{x} \cdot \mathbf{y}|.$$

Then, $\lambda_{\min}(C,k) \geq \lambda_{j+1} \cdot (1 - \beta_j)$.

Proof. Let $\mathbf{x}' \in \mathbb{R}^n, \|\mathbf{x}'\|_2 = 1, \|\mathbf{x}\|_0 \leq k$ be an eigenvector corresponding to $\lambda_{\min}(C,k)$. Let α_i be the coefficients of the representation of $\mathbf{x}'$ in terms of the $\mathbf{e_i}$: $\mathbf{x}' = \sum_{i=1}^n \alpha_i \mathbf{e_i}$. Thus, $\sum_{i=1}^n \alpha_i^2 = 1$, and we can write

$$\lambda_{\min}(C,k) \;=\; \mathbf{x'^T} C \mathbf{x}' \;=\; \sum_{i=1}^{n} \alpha_i^2 \lambda_i \;\geq\; \lambda_{j+1}(1 - \sum_{i=1}^{j} \alpha_i^2).$$

Since $\sum_{i=1}^j \alpha_i^2$ is the length of the projection of $\mathbf{x}$ onto V_j, we have

$$\sum_{i=1}^{j} \alpha_i^2 \;=\; \max_{\mathbf{y} \in V_j, \|\mathbf{y}\|_2 = 1} |\mathbf{x}' \cdot \mathbf{y}| \;\leq\; \max_{\mathbf{y} \in V_j, \mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_2 = \|\mathbf{y}\|_2 = 1, \|\mathbf{x}\|_0 \leq k} |\mathbf{y} \cdot \mathbf{x}|,$$

completing the proof. ■

Since all the λ_j can be computed easily, the crux in using this bound is finding a good bound on β_j. Next, we show a PTAS (Polynomial-Time Approximation Scheme) for approximating β_j, for any constant j.

Lemma 27 *For every $\epsilon > 0$, there is a $1 - \epsilon$ approximation for calculating β_j, running in time $O((\frac{n}{\epsilon})^j)$.*

Proof. Any vector $\mathbf{y} \in V_j$ with $\|\mathbf{y}\|_2 = 1$ can be written as $\mathbf{y} = \sum_{i=1}^j \eta_i \mathbf{e_i}$ with $\eta_i \in [-1, 1]$ for all i. The idea of our algorithm is to exhaustively search over all $\mathbf{y}$, as parametrized by their η_i entries. To make the search finite, the entries are discretized to multiples of $\delta = \epsilon \cdot \sqrt{k/(nj)}$. The total number of such vectors to search over is $(2/\delta)^j \leq (n/\epsilon)^j$.

Let $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ attain the maximum in the definition of β_j, and write $\hat{\mathbf{y}} = \sum_{i=1}^j \hat{\eta}_i \mathbf{e_i}$. For each i, let η_i be $\hat{\eta}_i$, rounded to the nearest multiple of δ, and $\mathbf{y} = \sum_{i=1}^j \eta_i \mathbf{e_i}$. Then, $\|\hat{\mathbf{y}} - \mathbf{y}\|_2 \leq \|\delta \sum_{i=1}^j \mathbf{e_j}\|_2 = \delta\sqrt{j}$.

The vector $\mathbf{x}' = \text{argmax}_{\mathbf{x} \in \mathbb{R}^n, \|\mathbf{x}\|_2 = 1, \|\mathbf{x}\|_0 \leq k} |\mathbf{y} \cdot \mathbf{x}|$ is of the following form: Let I be the set of k indices i such that $|y_i|$ is largest, and $\gamma = \sqrt{\sum_{i \in I} y_i^2}$. Then, $x'_i = 0$ for $i \notin I$ and $x'_i = y_i/\gamma$ for $i \in I$. Notice that given $\mathbf{y}$, we can easily find $\mathbf{x}'$, and because $|\hat{\mathbf{x}} \cdot \mathbf{y}| \leq |\mathbf{x}' \cdot \mathbf{y}| \leq |\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}|$, we have

$$\frac{||\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}| - |\mathbf{x}' \cdot \mathbf{y}||}{|\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}|} \;\leq\; \frac{||\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}| - |\hat{\mathbf{x}} \cdot \mathbf{y}||}{|\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}|} \;\leq\; \frac{\|\hat{\mathbf{x}}\|_2 \|\hat{\mathbf{y}} - \mathbf{y}\|_2}{|\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}|} \;\leq\; \frac{\delta\sqrt{j}}{|\hat{\mathbf{x}} \cdot \hat{\mathbf{y}}|} \;\leq\; \delta\sqrt{jn/k}.$$

The last inequality follows since the sum of the k largest entries of $\hat{\mathbf{y}}$ is at least $k/\sqrt{n}$, so by setting $x_i = 1/\sqrt{k}$ for each of those coordinates, we can attain at least an inner product of $\sqrt{k/n}$, and the inner product with $\hat{\mathbf{x}}$ cannot be smaller.

The value output by the exhaustive search over all discretized values is at least $|\mathbf{x}' \cdot \mathbf{y}|$, and thus within a factor of $1 - \frac{\delta\sqrt{j}n}{k} = 1 - \epsilon$ of the maximum value, attained by $\hat{\mathbf{x}}, \hat{\mathbf{y}}$. ■

Journal of Machine Learning Research 19 (2018) 1-62 Submitted 12/16; Revised 11/17; Published 8/18

A Hidden Absorbing Semi-Markov Model for Informatively Censored Temporal Data: Learning and Inference

Ahmed M. Alaa[†] AHMEDMALAA@UCLA.EDU
[†]*Electrical Engineering Department*
University of California, Los Angeles (UCLA)
Los Angeles, CA 90095-1594, USA

Mihaela van der Schaar[*,†] MIHAELA.VANDERSCHAAR@ENG.OX.AC.UK
[*]*Department of Engineering Science*
University of Oxford
Parks Road, Oxford OX1 3PJ, UK

Editor: Edoardo M. Airoldi

Abstract

Modeling continuous-time physiological processes that manifest a patient's evolving clinical states is a key step in approaching many problems in healthcare. In this paper, we develop the *Hidden Absorbing Semi-Markov Model* (HASMM): a versatile probabilistic model that is capable of capturing the modern electronic health record (EHR) data. Unlike existing models, the HASMM accommodates irregularly sampled, temporally correlated, and informatively censored physiological data, and can describe non-stationary clinical state transitions. Learning the HASMM parameters from the EHR data is achieved via a novel *forward-filtering backward-sampling* Monte-Carlo EM algorithm that exploits the knowledge of the end-point clinical outcomes (informative censoring) in the EHR data, and implements the E-step by sequentially sampling the patients' clinical states in the reverse-time direction while conditioning on the future states. Real-time inferences are drawn via a forward-filtering algorithm that operates on a virtually constructed discrete-time *embedded Markov chain* that mirrors the patient's continuous-time state trajectory. We demonstrate the prognostic utility of the HASMM in a critical care prognosis setting using a real-world dataset for patients admitted to the Ronald Reagan UCLA Medical Center. In particular, we show that using HASMMs, a patient's clinical deterioration can be predicted 8-9 hours prior to intensive care unit admission, with a 22% AUC gain compared to the Rothman index, which is the state-of-the-art critical care risk scoring technology.

Keywords: Hidden Semi-Markov Models, Medical Informatics, Monte Carlo methods.

1. Introduction

Modeling the clinical conditions of a patient using evidential physiological data is a ubiquitous problem that arises in many healthcare settings, including disease progression modeling (Schulam and Saria (2015); Mould (2012); Wang et al. (2014); Jackson et al. (2003); Sweeting et al. (2010); Liu et al. (2015)) and critical care prognosis (Moreno et al. (2005); Matos et al. (2006); Yoon et al. (2016); Hoiles and van der Schaar (2016); Alaa et al. (2016)). Accurate physiological modeling in these settings confers an *instrumental value* that manifests in the ability to provide early diagnosis, individualized treatments and timely interventions (e.g. early warning systems in hospital wards (Yoon et al. (2016)), early diagnosis for Scleroderma patients (Varga et al. (2012); Alaa and van der Schaar (2016)), early detection of a progressing breast cancer (Bartkova et al. (2005)), etc). Physiological modeling also confers an *epistemic value* that manifests in the knowledge extracted from data about the progression and severity phases of a disease (Stelfox et al. (2012))), or the short-term dynamics of the physiological behavior of critically ill patients (Li-wei et al. (2013)).

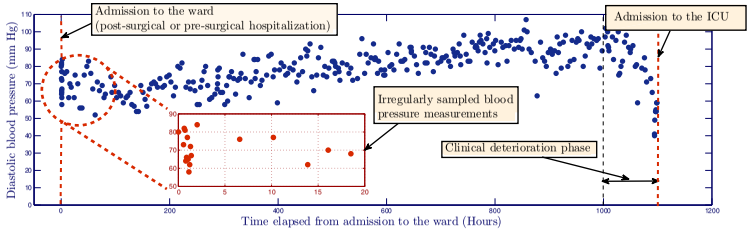


Figure 1: An episode of the diastolic blood pressure measurements (as recorded in the EHR) for a patient hospitalized in a regular ward for 50 days and then admitted to the ICU after the ward staff realized she is clinically deteriorating.

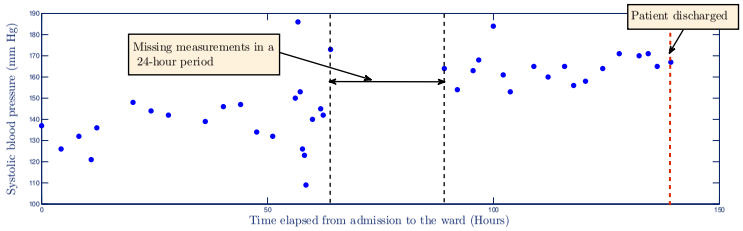


Figure 2: An episode of the systolic blood pressure measurements for a patient hospitalized in a regular ward for 6 days and then discharged home by the ward staff. Measurements are missing in a 24-hour period during the patient's stay in the ward.

The recent availability of data in the electronic health records (EHR)[1] creates a promising horizon for establishing rich and complex physiological models (Gunter and Terry (2005)). Modern EHRs comprise *episodic* data records for individual (anonymized) patients; every patient's episode is a temporal sequence of clinical findings (e.g. visual field index for Glaucoma patients (Liu et al. (2015)), CD4 cell counts for HIV-infected patients (Guihenneuc-Jouyaux et al. (2000)), etc), lab test results (e.g. white cell blood count for

1. A recent data brief from the Office for National Coordinator (ONC) for healthcare technology shows that the adoption of EHR in US hospitals exhibited a spectacular increase from 9.4% in 2008, 27.6% in 2011, to 75.5% in 2014 (Charles et al. (2015)).

post-operative patients under immunosuppressive drugs (Cholette et al. (2012)), etc), or vital signs (e.g. blood pressure and O_2 saturation (Yoon et al. (2016))). The time span of these episodes may be as short as few days in short-term hospitalization episodes (e.g. patients with solid tumors, hematological malignancies or neutropenia who are hospitalized in regular wards before or after a surgery (Kause et al. (2004); Hogan et al. (2012); Kirkland et al. (2013))), or as long as few years in longitudinal episodes (e.g. chronic obstructive pulmonary disease may evolve from a mild Stage I to a very severe Stage IV over a time span of 10 years (Pedersen et al. (2011); Wang et al. (2014))). **In this paper, we develop a versatile time-series model that provides means for accurate real-time risk prognostication of adverse clinical outcomes.** Other applications of the model include but are not limited to modeling default data in quantitative finance (Giampieri et al. (2005)), and fault detection in general dynamic systems (Smyth (1994)). In the next Subsection, we expose our modeling rationale and challenges posed by the structure of modern EHR data. We conclude this Section by summarizing our contributions in Subsection 1.2.

1.1 Modeling Rationale and Challenges

1.1.1 Rationale

Previous physiological models have branched into two different modeling paths with respect to the way a patient's clinical states are defined. One strand of literature adopts *fully observable models*; these models assume that clinical states are quantifiable via *observable* clinical markers or disease severity measures (e.g. PFVC in Scleroderma (Schulam and Saria (2015)), GFR in kidney disease (Eddy and Neilson (2006)), etc). Another strand of literature adopts *latent variable models*, which assume that clinical states are latent and manifest only through proximal, noisy physiological measurements. Table 1 lists some notable previous works that fall under each modeling category[2].

Table 1: Modeling methodologies in previous works.

Methodology	Previous Works
Fully observable models	• HIV (Dessie (2014); Foucher et al. (2005)) • Chronic kidney diseases (Eddy and Neilson (2006)) • Scleroderma (Schulam and Saria (2015)) • ICU (Ghassemi et al. (2015)).
Latent variable models	• Alzheimer (Chen and Zhou (2011)) • HIV (Guihenneuc-Jouyaux et al. (2000)) • Glaucoma (Liu et al. (2015)) • Comorbidities (Wang et al. (2014)).

2. While the models in (Schulam and Saria (2015)) and (Ghassemi et al. (2015)) involve latent variables that designate patient subtypes, the clinical states in both works are considered to be captured via observable bio-markers (PFVC in the former and the Cerebrovascular Autoregulation index in the latter).

Our modeling choice is to go with a latent variable model for the following reasons:

- In a wide range of problems, a concrete clinical marker that can be directly used as a surrogate for the patient's true clinical condition is **not** available. This is especially true in critical care settings where no solid definition or measure of a "clinical state" exists (Li-wei et al. (2013)). Previous works that adopted a clinical risk score as a surrogate for the clinical state in critical care settings have found that other physiological features, when augmented with the clinical risk score, still hold a significant predictive power with respect to end-point clinical outcomes (Ghassemi et al. (2015)). This implies that a clinical risk score or a severity of illness measure (such as APACHE II, SAPS and SOFA (Knaus et al. (1991); Subbe et al. (2001))) is not a sufficient measure of a patient's true clinical condition, and hence cannot be reliably modeled as an observable clinical state.

- The same line of argument extends to disease progression models: (Jackson et al. (2003)) has shown that significant modeling gain can be attained by treating clinical markers and diagnostic assessments as noisy manifest variables for the patient's true clinical state rather than defining a clinical state in terms of those markers.

- For various chronic disease, such as HIV, Scleroderma, and kidney disease, progression stages are well defined in terms of observable clinical markers (CD4 cell count, PFVC and GFR). However, a latent variable model can help validate and assess the current domain knowledge-based clinical practice guidelines by learning alternative, data-driven guidelines. Other diseases, such as COPD, have their progression stages manifesting only through symptoms (e.g. chronic bronchitis, emphysema and chronic airway obstruction (Wang et al. (2014))), which may or may not accurately reflect the disease's true state, and hence a latent variable model is necessary.

- Conclusive clinical markers that reveal a patient's true state may be available only occasionally in a patient's longitudinal episode. For instance, in a breast cancer progression setting, most of the data points associated with a patient's longitudinal episode would be imaging test results (e.g. BI-RADS scores of a mammogram or an MRI (Gail and Mai (2010); Taghipour et al. (2013))), which are noisy markers for the existence of a tumor, whereas a conclusive biopsy result that truly reveals whether the patient is in a preclinical or clinical breast cancer state may not be available because the patient did not undergo a biopsy test.

- A fully observable model does not provide diagnostic utility since it assumes that an already observable clinical marker provides an immediate, domain-knowledge-based diagnosis for the patient. Contrarily, a latent variable model leaves room for diagnoses to be learned from evidential data by learning the association between physiological evidence and clinical states, which may help inform and improve clinical practice.

1.1.2 Challenges

Hidden Markov Models (HMMs) and their variants have been widely deployed as temporal latent variable models for dynamical systems (Smyth (1994); Zhang et al. (2001); Giampieri et al. (2005); Genon-Catalot et al. (2000); Ghahramani and Jordan (1997)). Such models

have achieved considerable success in various applications, such as topic modeling (Gruber et al. (2007)), speaker diarization (Fox et al. (2011b)), and speech recognition (Rabiner (1989)). However, the nature of the clinical setting, together with the format of the modern EHR data pose the following set of serious challenges that confound classical HMM models:

(A) Non-stationarity: Recently developed disease progression models, such those in (Wang et al. (2014)) and (Liu et al. (2015)), use conventional stationary Markov chain models. In particular, they assume that state transition probabilities are independent of time. However, this assumption is seriously at odds with even casual observational studies which show that the probability of transiting from the current state to another state depends on the time spent in the current state (Lagakos et al. (1978); Huzurbazar (2004); Gillaizeau et al. (2015)). This effect, which violates the memorylessness assumptions adopted by continuous-time Markovian models, was verified in patients who underwent renal transplantation (Foucher et al. (2007, 2008)), patients who are HIV infected (Joly and Commenges (1999); Dessie (2014); Foucher et al. (2005)), and patients with chronic obstructive pulmonary disease (Bakal et al. (2014)).

(B) Irregularly spaced observations: The times at which the clinical findings of a patient (vital signs or lab tests) are observed is controlled either by clinicians (in the case of hospitalized inpatients), or by the patient's visit times (in the case of a chronic disease follow up). The time interval between every two measurements may vary from one patient to another, and may also vary for the same patient within her episode. This is reflected in the structure of the episodes in the EHR records, as shown in Figure 1 and 2. Figure 1 depicts an actual diastolic blood pressure episode for a patient hospitalized in a regular ward for 1200 hours (50 days)[3]. The patient's stay in the ward was concluded with an admission to the ICU after the ward staff realized she was clinically deteriorating. As we can see, the blood pressure measurements in the first 20 hours were initially taken with a rate of 1 sample per hour, and then later the rate changed to 1 sample every 5 hours[4]. While some recent works have argued for the parametrization of time in longitudinal data via the natural event sequence (Hripcsak et al. (2015)), it is often the case that the sampling times are themselves informative of the patients' clinical well-being (Alaa et al. (2017)). Thus, a direct application of a regular, discrete-time HMM (e.g. the models in (Murphy (2002); Fox et al. (2011b,a); Rabiner (1989); Yu (2010); Matos et al. (2006); Guihenneuc-Jouyaux et al. (2000))) will not suffice for jointly describing the latent states and observations, and hence ensuring accurate inferences.

(C) Discrete observations of a continuous-time phenomena: A patient's physiological signals and latent states evolve in continuous time; however, the observed physiological measurements are gathered at discrete time steps that can differ from one physiological signal to another. (One alternative view of such a structure is to think of a time series with irregularly sampled multidimensional measurements and with missing data in every measurement (Lipton et al. (2016)). We do not address data that is missing **not** at random

3. A detailed description for the data involved in this paper is provided in Section 5.
4. While Figure 1 illustrates a short-term episode for a critical care patient, similar effects are experienced in longitudinal episodes for patients with chronic disease (see Figure 4 in (Wang et al. (2014))).

in this paper.) The intervals between observed measurements can vary quite significantly; as we can see in Figure 2, the systolic blood pressure for a patient who stayed in a ward for 140 hours exhibits an entire day without measurements [5]. This means that the patient may encounter multiple hidden state transitions without any associated observed data. These effects make learning and inference problems more complicated since the inference algorithms need to consider potential unobserved trajectories of state evolution between every two timestamps. This challenge has been recently addressed in (Nodelman et al. (2012); Wang et al. (2014); Liu et al. (2015)), but only on the basis of memoryless Markov chain models for the hidden states, for which tractable inferences that rely on the solutions to Chapman-Kolmogorov equations can be executed. However, incorporating non-stationarity in state transitions (i.e. addressing challenge (A)) would make the problem of reasoning about a continuous-time process through discrete observations much more complicated.

(D) Lack of supervision: The episodes in the EHR may be labeled with the aid of domain knowledge (e.g. the stages and symptoms of some chronic diseases, such as chronic kidney disease (Eddy and Neilson (2006)), are known to clinicians and may be provided in the EHR). However, in many cases, including the case of (post or pre-operative) critical care, we do not have access to any labels for the patients' states. Hence, unsupervised learning approaches need to be used for learning model parameters from EHR episodes. While unsupervised learning of discrete-time HMMs has been extensively studied and is well understood (e.g. the Baum-Welch EM algorithm is predominant in such settings (Zhang et al. (2001); Yu (2010); Rabiner (1989))), the problem of unsupervised learning of continuous-time models for which both the patient's states and state transition times are hidden is far less understood, and indeed far more complicated.

(E) Censored observations: Episodes in the EHR are usually terminated by an informative intervention or event, such as death, ICU admission, discharge, etc. This is known as *informative censoring* (Scharfstein and Robins (2002); Huang and Wolfe (2002); Link (1989)). Unlike classical HMM settings where training sets comprise fixed length, or arbitrarily-censored, HMM sequence instances, a typical EHR dataset would comprise a set of episodes with different durations, and the duration of each episodes is itself informative of the state trajectory. Learning in such settings requires algorithms that can efficiently compute the likelihood of observing a set of episodes conditioned on their durations and terminating states.

1.2 Summary of Contributions

In order to address the challenges above, we develop a new model –which we call the *Hidden Absorbing Semi-Markov Model* (HASMM)– as a versatile generative model for a patient's (physiological) episode as recorded in the EHR. The HASMM captures non-stationary transitions for a patient's clinical state via a continuous-time semi-Markov model with explicitly specified state sojourn time distributions. Informative censoring is captured via absorbing states that designate clinical endpoint outcomes (e.g. cardiac arrest, mortality, recovery,

5. This may have resulted due to the patient undergoing a surgery or an intervention, or because the EHR recording system accidentally did not receive the data from the clinicians during that day.

etc); entering an absorbing state of an HASMM stimulates censoring events (e.g. clinical deterioration leads to an ICU admission which terminates the physiological observations for a monitored patient in a ward, etc). Observable variables are modeled via a multi-task Gaussian process (Bonilla et al. (2007)), for which the observation times (i.e. follow up visits, vital sign gathering, lab tests, etc) are modeled as a point process. Using multi-task Gaussian process with state-dependent hyper-parameters, an HASMM accounts for both correlations among different physiological variables, in addition to the temporal correlations among the observation variables that are generated by the same hidden state during its sojourn period. In that sense, an HASMM is a segment model (Ostendorf et al. (1996)) and also a *state-switching* model (Fox et al. (2011a))).

To allow for real-time inference of a patient's state, we develop a forward-filtering HASMM inference algorithm that can estimate a patient's latent state using her history of irregularly sampled physiological measurements. The inference algorithm operates by constructing a virtual, discrete-time *embedded Markov chain* that fully describes the patient's state transitions at observation times. The embedded Markov chain is constructed in an offline stage by solving a system of *Volterra integral equations of the second kind* using the *successive approximation* method; the solution to this system of equations, which parallels the Chapman-Kolmogorov equations in ordinary Markov chains, describe the HASMM's semi-Markovian state transitions as observed at arbitrarily selected discrete timestamps.

Offline learning of the HASMM model parameters from patients' episodes in an EHR is a daunting task. Since the HASMM is a continuous-time model, we cannot directly use the classical Baum-Welch EM algorithms for learning its parameters (Rabiner (1989)). Moreover, the semi-Markovianity of an HASMM yields an intractable integral in the E-step of the Expectation-Maximization (EM) formulation. Since the HASMM's state transitions are not captured by the conventional continuous-time Markov chain transition rate matrices, we cannot make use of the *Expm* and *Unif* methods that were used in (Hobolth and Jensen (2011)), and more recently in (Liu et al. (2015)) for evaluating the integrals involved in the E-step of learning continuous-time HMMs. To address this challenge, we develop a novel *forward-filtering backward-sampling Monte Carlo EM* (FFBS-MCEM) algorithm that approximates the integral involved in the E-step by efficiently sampling the latent clinical trajectories conditioned on observations in the EHR by exploiting the informative censoring of the patients' episodes. The FFBS-MCEM algorithm samples the latent clinical states of every (offline) patient episode in the EHR as follows: it starts from the known clinical endpoints, and sequentially samples the patient's states by traversing in the reverse-time direction while conditioning on the future states, and then uses the sampled state trajectories to evaluate a Monte Carlo approximation for the E-step.

2. The Hidden Absorbing Semi-Markov Model (HASMM)

2.1 Abstract Model

We start by describing the HASMM's hidden state evolution process, and then we describe the structure of its observable variables.

2.1.1 Hidden States

We consider a filtered probability space $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{R}_+}, \mathbb{P})$, over which a continuous-time stochastic process $X(t)$ is defined on $t \in \mathbb{R}_+$. The process $X(t)$ corresponds to a temporal trajectory of the patient's hidden clinical states, which take on values from a finite *state-space* $\mathcal{X} = \{1, 2, \ldots, N\}$. Because the process $X(t)$ takes on only finitely many values, it can be decomposed in the form[6]

$$X(t) = \sum_n X_n \cdot \mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}, \tag{1}$$

where $(X(t))_{t\in\mathbb{R}_+}$ is a càdlàg path, $X_n \in \mathcal{X}$, and the interval $[\tau_n, \tau_{n+1})$ is the time interval accommodating the n^{th} hidden state. Every path $(X(t))_{t\in\mathbb{R}_+}$ on the stochastic basis $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{R}_+}, \mathbb{P})$ is a *semi-Markov path* (Janssen and De Dominicis (1984); Durrett (2010)), where the *sojourn time* of state n, which we denote as $S_n = \tau_{n+1} - \tau_n$, is drawn from a *state-specific* distribution $v_j(S_n = s\,|\lambda_j) = d\mathbb{P}(S_n = s\,|X_n = j)$, with λ_j being a state-specific *duration parameter* associated with state $j \in \mathcal{X}$. Unlike ordinary time-homogeneous semi-Markov transitions, in which the transition probabilities among states are assumed to be constant conditioned on there being a transition from the current state (Gillaizeau et al. (2015); Murphy (2002); Johnson and Willsky (2013); Yu (2010); Dewar et al. (2012); Guédon (2007)), our model accounts for *duration-dependent* semi-Markov transitions, i.e. we have that

$$\mathbb{P}(X_{n+1} = j | X_n = i, S_n = s) = g_{ij}(s), \tag{2}$$

where $g_{ij} : \mathbb{R}_+ \to [0, 1], \forall i, j \in \mathcal{X}$ is a *transition function* for which $\frac{\partial g_{ij}(s)}{\partial s}$ is well defined, and $\sum_{j=1}^{N} g_{ij}(s) = 1, \forall s \in \mathbb{R}_+, i \in \mathcal{X}$.

Now consider the bi-variate (renewal) process $(X_n, S_n)_{n\in\mathbb{N}_+}$, which comprises the sequence of states and sojourn times. Semi-Markovianity of $X(t)$ implies that $(X_n, S_n)_{n\in\mathbb{N}_+}$ satisfies the following condition on its transition probabilities

$$\begin{aligned}\mathbb{P}(X_{n+1} = j, S_n \leq s\,|\,\mathcal{F}_{\tau_n}) &= \mathbb{P}(X_{n+1} = j, S_n \leq s\,|\,X_n = i)\\ &= \mathbb{P}(X_{n+1} = j\,|\,X_n = i, S_n \leq s) \cdot \mathbb{P}(S_n \leq s\,|\,X_n = i)\\ &= \mathbb{E}_{S_n}\left[g_{ij}(S_n)\,|\,S_n \leq s\right] \cdot V_i(s\,|\,\lambda_i) = \bar{g}_{ij}(s) \cdot V_i(s\,|\,\lambda_i),\end{aligned} \tag{3}$$

where $\{X_n = i\} \in \mathcal{F}_{\tau_n}$, $V_i(.)$ is the cumulative distribution function of state i's sojourn time, and $\bar{g}_{ij}(s)$ is the probability mass function that reflects the probability that a patient's next state being j given that she was at state i and her sojourn time in i is less than (or equal to) s. Based on (3), we define the *semi-Markov transition kernel* as a matrix-valued function $\mathbf{Q} : \mathbb{R}_+ \to [0, 1]^{N\times N}$ that describes the dynamics of $X(t)$ in continuous time, with entries $\mathbf{Q}(s) = (Q_{ij}(s))_{i,j\in\mathcal{X}}$ that are given by

$$Q_{ij}(s) = \bar{g}_{ij}(s) \cdot V_i(s\,|\,\lambda_i). \tag{4}$$

The initial state X_1 is random[7], and the initial state distribution is given by $\mathbf{p}^o = [p_1^o, \ldots, p_N^o]^T$, where $p_j^o = \mathbb{P}(X(0) = j)$, and $\sum_{j=1}^{N} p_j^o = 1$.

6. By convention, we set $\tau_1 = 0$.
7. We do not consider left-censored observations in this model.

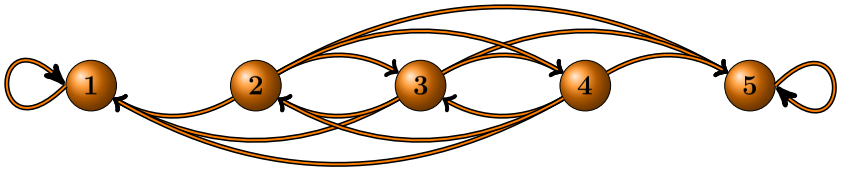


Figure 3: The Markov chain model for a 5-state HASMM.

We assume that whenever the patient enters either state 1 or state N, she remains there forever[8]. Therefore, we model states $\{1, N\}$ as *absorbing states*, whereas we model the remaining states in $\mathcal{X} \setminus \{1, N\}$ as *transient states* that represent intermediate levels of risk. We define and interpret states 1 and N as follows:

- **State** 1 is denoted as the *safe state*, and represents the state at which the patient is at minimum (or no) risk (e.g. clinically stable post-operative patient, etc).
- **State** N is denoted as the *catastrophic state*, and represents the state at which the patient is at severe risk or encounters an adverse event (e.g. a very severe stage of a chronic disease (Bakal et al. (2014)), a cardiac or respiratory arrest (Subbe et al. (2001)), mortality (Knaus et al. (1991)), etc).

We do not assume that the transient states are ordered linearly in terms of clinical risk. Following the assumptions in (Murphy (2002); Johnson and Willsky (2013)), we eliminate the self-transitions for all transient states by setting $g_{ii}(s) = 0, Q_{ii}(s) = 0, \forall s \in \mathbb{R}_+, i \in \mathcal{X} \setminus \{1, N\}$, whereas we restrict the transitions from states 1 and N to self-transitions only, i.e. $g_{ii}(s) = 1, i \in \{1, N\}$. Figure 3 depict the Markov chain for the sequence $\{X_n\}_{n\in\mathbb{N}_+}$.

We define $\mathcal{A}_1$ as the event that the path $(X(t))_{t\in\mathbb{R}_+}$ is absorbed in the safe state 1, i.e. $\mathcal{A}_1 = \{\lim_{t\to\infty} X(t) = 1\}$, and $\mathcal{A}_N$ as the event that $(X(t))_{t\in\mathbb{R}_+}$ is absorbed in the catastrophic state N, i.e. $\mathcal{A}_N = \{\lim_{t\to\infty} X(t) = N\}$. Since $(X(t))_{t\in\mathbb{R}_+}$ is an absorbing semi-Markov chain[9], we know that $\mathbb{P}(\mathcal{A}_1 \vee \mathcal{A}_N) = 1$, and since the events $\mathcal{A}_1$ and $\mathcal{A}_N$ are mutually exclusive, it follows that $\mathbb{P}(\mathcal{A}_N) = 1 - \mathbb{P}(\mathcal{A}_1)$. The quantity $\mathbb{P}(\mathcal{A}_N)$ describes a patient's prior risk of ending in the catastrophic state, whereas $\mathbb{P}(\mathcal{A}_N \,|\mathcal{F}_t\,)$ describes the patient's posterior risk of ending in the catastrophic state having observed its evolution history up to time t[10]. Define T_s as an $\mathcal{F}$-stopping time representing the absorption time

8. The model can be easily extended to accommodate an arbitrary number of competing absorbing states.
9. We assume that the transition functions $g_{i1}(s)$ and $g_{iN}(s)$ for any transient state i is non-zero for every s. Hence, it follows that $(X(t))_{t\in\mathbb{R}_+}$ is an absorbing semi-Markov chain since it has 2 absorbing states, each of which can be visited starting from any other state (Durrett (2010)).
10. In the clinical applications under consideration, transient states can be ordered by their respective relative risks of encountering event $\mathcal{A}_N$ in the subsequent transitions, i.e. in a 5-state chain, it is more likely for the patient to be absorbed in state 5 in the future when it is in state 4 than when it is in state 3. For instance, it is more likely for a patient's chronic obstructive pulmonary disease that is currently assessed to have a severity degree of GOLD1 (mild severity as defined in the GOLD standard Pedersen et al. (2011)) to progress (in the near future) to a severity degree of GOLD2 (moderate) rather than GOLD3 (severe).
11. This means that the sampling times are uninformative of the latent states; which simplifies the inference problem. The HASMM model can be extended to incorporate a state-dependent sampling process using a Cox process (Lando (1998)) or a Hawkes process (Hawkes and Oakes (1974)) to modulate the point-process intensity;however, such an extension would result in a significantly harder inference problem. A good discussion on conditional intensity models can be found in (Qin and Shelton (2015)).
12. Note that what is observed is a sequence of sampling times $\mathcal{T}$, the elements of which are not labeled by the corresponding state indexes, for that the states are latent, i.e. the sets $\mathcal{T}_n$ are unobserved partitions of $\mathcal{T}$.

of the path $(X(t))_{t\in\mathbb{R}_+}$ in either state 1 or state N, i.e.

$$T_s = \inf\{t \in \mathbb{R}_+ : X(t) \in \{1, N\}\}.$$

Finally, we define K as the (random) number of state realizations in the sequence $\{X_n\}_{n=1}^K$ up to the stopping time T_s, which has to be concluded by either state 1 or N, e.g. when $|\mathcal{X}| = 4$, the sequences $\{1\}, \{4\}, \{2, 3, 2, 3, 4\}$, and $\{3, 2, 1\}$ are valid, random-length realizations of $\{X_n\}_{n=1}^K$, and each represents a certain state evolution trajectory for the patient.

2.1.2 Observations and Censoring

The path $(X(t))_{t\in\mathbb{R}_+}$ is unobservable; what is observable is a corresponding process $(Y(t))_{t\in\mathbb{R}_+}$ on $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in\mathbb{R}_+}, \mathbb{P})$, the values of which are drawn from an *observation-space* $\mathcal{Y}$, and whose distributional properties are dependent on the latent states' path $(X(t))_{t\in\mathbb{R}_+}$. The observable process $(Y(t))_{t\in\mathbb{R}_+}$ can be put in the form

$$Y(t) = \sum_n Y_n(t) \cdot \mathbf{1}_{\{\tau_n \le t < \tau_{n+1}\}}, \tag{5}$$

where $(Y(t))_{t\in\mathbb{R}_+}$ is a càdlàg path, comprising a sequence of function-valued variables $\{Y_n(t)\}_{n=1}^K$, with $Y_n : [\tau_n, \tau_{n+1}) \to \mathcal{Y}$. Even though the path $(Y(t))_{t\in\mathbb{R}_+}$ is accessible, only a sequence of irregularly spaced samples of it is observed over time, and is denoted by $\{Y(t_m)\}_{t_m\in\mathcal{T}}$, where $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$ is the set of observed measurements, and M is the total number of such measurements. We say that the process is censored if $M < \infty$; typical episodes in an EHR are censored: observations stop at some point of time due to a release from care, an ICU admission, mortality, etc.

The sampling times in $\mathcal{T}$ represent the times at which a patient with a chronic disease took clinical tests (i.e. time intervals in $\mathcal{T}$ spans years), or the times at which clinicians have gathered vital signs for a monitored critically ill patient in a hospital ward (i.e. time intervals in $\mathcal{T}$ span days or hours). We assume that the sampling times in $\mathcal{T}$ are drawn from a *point-process* $\Phi = \sum_{m\in\mathbb{N}_+} \delta_{t_m}$, with δ_t being the Dirac measure. The point-process Φ is assumed to be independent of the latent states path[11]. Define $\mathcal{T}_n$ as the set of M_n samples that are gathered during the interval[12] $[\tau_n, \tau_{n+1})$, i.e. $\mathcal{T}_n = \{t_m : t_m \in \mathcal{T}, t_m \in [\tau_n, \tau_{n+1})\}$, $M_n = |\mathcal{T}_n|$, and $\sum_n M_n = M$. Since $\mathcal{T}_n$ could possibly be empty ($\mathcal{T}_n = \emptyset$), some states can have no corresponding observations (i.e. an inpatient may exhibit a transition to a deteriorating state during the night, even though her blood pressure were not measured during the night.

Recall the illustration in Figure 2).

The paths $\{Y_n(t)\}_{n=1}^K$ are assumed to be conditionally independent given the hidden sequence of states $\{X_n\}_{n=1}^K$, and hence we have that

$$\{Y(t_m)\}_{t_m\in\mathcal{T}_n} \perp\!\!\!\perp \{Y(t_m)\}_{t_m\in\mathcal{T}_{n+1}} \,|\, X_n, X_{n+1}\,, \forall n \in \{1,2,\ldots,K-1\}.$$

The observed samples generated under every state X_n and sampled at the times in $\mathcal{T}_n$ are drawn from $\mathcal{Y}$ according to a distribution $\mathbb{P}(\{Y(t_m)\}_{t_m\in\mathcal{T}_n}|X_n = j, \Theta_j)$, where Θ_j is an *emission parameter* that controls the distributional properties of the observations generated under state j.

The number of observation samples is finite: the observed sequence is *censored* at some point of time, which we call the censoring time T_c, after which no more observation samples are available. Censoring reflects an external intervention/event that terminated the observation sequence, i.e. death, intensive care unit (ICU) admission, etc. Censoring is *informative* (Scharfstein and Robins (2002);Huang and Wolfe (2002);Link (1989)), because the censoring time is correlated with the absorption time T_s, and T_s strictly precedes T_c (in an almost sure sense). That is, T_c is an $\mathcal{F}$-stopping time that is given by $T_c = T_s + S_K$, i.e. once the patient enters state 1 or state N, the observations stop after the patient's sojourn time in that state (i.e. observations stop after a time S_K from the entrance in the absorbing state). Therefore, the duration distributions $v_1(s|\lambda_1)$ and $v_N(s|\lambda_N)$ of states 1 and N are used to determine the censoring times conditioned on the chain $\{X_n\}_{n=1}^K$ being absorbed at time T_s.

Every sample from the HASMM is an episode comprising a random-length sequence of hidden states $\{X_n\}_{n=1}^K$, and a random-length sequence of observations $\{Y(t_m)\}_{m=1}^M$ together with the associated observation times. We only observe $\{Y(t_m)\}_{m=1}^M$; the path of latent states $X(t)$, the number of realized states K, the association between observations and states (i.e. the sets $\mathcal{T}_n$) are all unobserved, which makes the inference problem very challenging, but captures the realistic EHR data format and the associated inferential hurdles. In the next subsection, we specify the model's generative process and present an algorithm to generate episodic samples from an HASMM.

2.2 Model Specification and Generative Process

As have been discussed in Subsection 2.1, the hidden and observables variables of an HASMM can be listed as follows:

- **Hidden variables:** The hidden states sequence $\{X_n\}_{n=1}^K$ and the states' sojourn times $\{S_n\}_{n=1}^K$ (or equivalently, the transition times $\{\tau_n\}_{n=1}^K$).

- **Observable variables:** The observed episode $\{Y(t_m)\}_{m=1}^M$ and the associated sampling times $\mathcal{T} = \{t_m\}_{m=1}^M$.

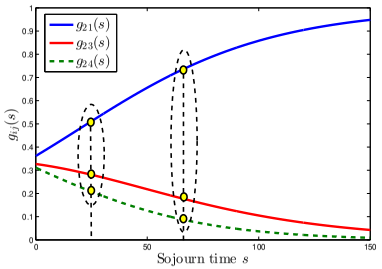


Figure 4: Exemplary transition functions $(g_{2j})_{j=1}^4$ for a 4-state HASMM.

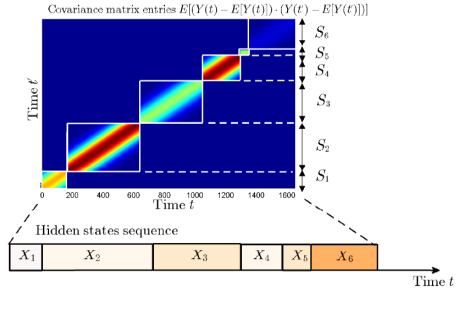


Figure 5: Depiction for the correlation structure of the observable variables for an underlying state sequence $\{X_n\}_{n=1}^6$.

The HASMM model parameters that generate both the hidden and observable variables are encompassed in the parameter set Γ, i.e.

$$\Gamma = \left(\underbrace{N}_{\text{State cardinality}}, \underbrace{\lambda = \{\lambda_j\}_{j=1}^N}_{\text{State duration}}, \underbrace{\mathbf{p}^o}_{\text{Initial states}}, \underbrace{\mathbf{Q} = \{Q_{ij}(s)\}_{i,j=1}^N}_{\text{Transitions}}, \underbrace{\mathbf{\Theta} = \{\Theta_j\}_{j=1}^N}_{\text{Emission}} \right).$$

The total number of parameters in an HASMM is $N^2 + 3N(E+1)$.

2.2.1 Distributional specifications for the hidden variables

We model the state sojourn time of each state $i \in \mathcal{X}$ via a Gamma distribution. The selection of a Gamma distribution ensures that the generative process encompasses ordinary continuous-time Markov models for the path $(X(t))_{t\in\mathbb{R}_+}$, since the exponential distribution[13] is a special case of the Gamma distribution (Durrett (2010)). Thus, if the underlying physiology of the patient is naturally characterized by memoryless state transitions, this will be automatically learned from the data via the parameters of the Gamma distribution. The sojourn time distribution for state i is given by

$$v_i(s|\lambda_i = \{\lambda_{i,s}, \lambda_{i,r}\}) = \frac{1}{\Gamma(\lambda_{i,s})} \cdot \lambda_{i,r}^{\lambda_{i,s}} \cdot s^{\lambda_{i,s}-1} \cdot e^{-s\cdot\lambda_{i,r}}, s \geq 0,$$

where $\lambda_{i,s} > 0$ and $\lambda_{i,r} > 0$ are the shape and rate parameters of the Gamma distribution respectively.

Now we specify the structure of the transition kernel $\mathbf{Q}(s) = (Q_{ij}(s))_{i,j}, i,j \in \mathcal{X}$. Recall from (4) that the each element in the transition kernel matrix can be written as $\mathbb{E}_S[g_{ij}(S)|S \leq s] \cdot V_i(s|\lambda_i)$. Having specified the distribution $v_i(s|\lambda_i)$ as a Gamma distribution, it remains to specify the function $g_{ij}(s)$ in order to construct the elements of $\mathbf{Q}(s)$.

13. Note that a semi-Markov chain reduces to a Markov chain if the sojourn times are exponentially distributed.

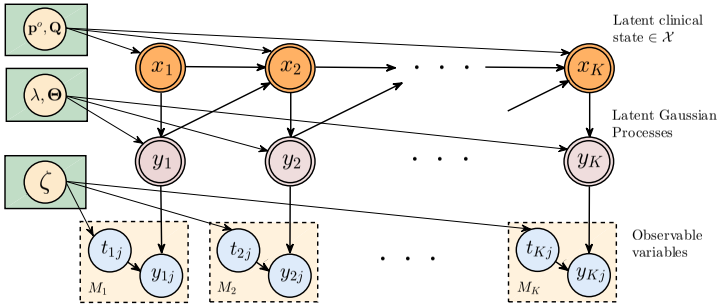


Figure 6: A basic graphical model for the HASMM. The arrow between the (function-valued) variable y_n and the latent state X_{n+1} designates the dependence of state transitions on the sojourn time of the previous state (duration-dependence).

The transition functions $(g_{ij}(s))_{i,j}$ are given by *multinomial logistic* functions as follows

$$\begin{aligned} g_{ij}(s) &= \frac{e^{(\eta_{ij}+\beta_{ij}\cdot s)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}\cdot s)}}, \forall i \neq j, i \notin \{1, N\} \\ g_{ii}(s) &= 0, \forall i \in \{2, \ldots, N-1\}, \\ g_{ii}(s) &= 1, \forall i \in \{1, N\}, \end{aligned} \tag{6}$$

where $\eta_{ij}, \beta_{ij} \in \mathbb{R}_+$. The parameters $(\eta_{ij})_{j=1}^{N}$ determine the baseline values for the transition probability mass from state i to state j, i.e. $g_{ij}(0)$, whereas the parameters β_{ij} controls the dependence of the transition probability mass on the sojourn time[14]. If $\beta_{ij} = 0$, then we have that $g_{ij}(s) = g_{ij}(0) = \frac{e^{\eta_{ij}}}{\sum_{k=1}^{N} e^{\eta_{ik}}}, \forall s \in \mathbb{R}_+$, i.e. the transition probability out of state i remains constant irrespective of the sojourn time in that state. In the limit when s goes to infinity, β_{ij} dominates the functional form in (6), and we have that $g_{ij}(\infty) = \arg\max_j \beta_{ij}$. Figure 4 depicts exemplary transition functions for a 4-state HASMM.

14. Similar effects for the sojourn time on the transition probabilities has been demonstrated in the progression of breast cancer from healthy to preclinical states in (Taghipour et al. (2013)), where age (the main risk factor for breast cancer) was shown to affect the probability of progressing across the states of healthy to preclinical, clinical and death. These effects may be also prevailing in other diseases, or in critical care settings where the length of time during which a patient stays clinically stable may imply that the patient is more likely to transit to a more healthy state in the future. Through the HASMM model, we can recognize whether or not this effect is evident in the EHR data, i.e. whether the transition function reflects an underlying homogeneous (if $g_{ij}(s)$ is independent of s) or duration-dependent transitions by learning the parameter β_{ij}. Moreover, the parameter β_{ij} is defined per state; the HASMM model can capture scenarios where transitions are duration-independent from some states, but are duration-dependent from others.

2.2.2 Distributional specifications for the observable variables

As explained in Subsection 2.1, the observable process $Y(t)$ can be decomposed as $Y(t) = \sum_{n=1}^{K} Y_n(t) \cdot \mathbf{1}_{\{\tau_n \leq t < \tau_{n+1}\}}$, where the paths $(Y_n(t))_{n=1}^{K}$ are conditionally independent given the state sequence $\{X_n\}_{n=1}^{K}$. Since observations are drawn from $Y(t)$ at arbitrarily, and irregularly spaced time instances $\mathcal{T}$, we have to model the distributional properties of $Y(t)$ in continuous time. We model every path $Y_n(t)$ defined over $[\tau_n, \tau_{n+1})$ as a segment drawn from a multi-task Gaussian Process (GP), with a hyper-parameter set Θ_i that depends on the corresponding latent state $X_n = i$ (Rasmussen (2006); Bonilla et al. (2007)). The input to the multi-task GP is the time variable and the output is the set of physiological variables at a certain point of time. The GP associated with every state $X_n = i$ is parametrized by a constant mean function $m_i(t) = m_i$, a *squared-exponential* covariance kernel $k_i(t, t') = \sigma_i^2 e^{-\frac{1}{2\ell_i^2}||t-t'||^2}$, and a "free-form" covariance matrix Σ_i between the different physiological measurements (Bonilla et al. (2007)). Thus, for a E-dimensional physiological stream $Y(t) = (Y^1(t), \ldots, Y^E(t))$, the observations for state i are generated as follows

$$\left\langle Y_i^l(t) \cdot Y_i^v(t') \right\rangle = \Sigma_i(l, v) \cdot k_i(t, t'), \qquad \{Y_i^l(t)\}_{t \in \mathcal{T}, 1 \leq l \leq E} \sim \mathcal{N}(m_i(t), \mathbf{\Sigma}_i),$$

where $\mathbf{\Sigma}_i(l, v, t, t') = \left\langle Y_i^l(t) \cdot Y_i^v(t') \right\rangle$. The GP hyper-parameters associated with state i are given by $\Theta_i = (m_i, \sigma_i, \Sigma_i, \ell_i)$, i.e. $Y_n(t)|X_n = i \sim \mathcal{GP}(\Theta_i)$.
We note that the HASMM model is a *segment model* (Ostendorf et al. (1996); Murphy (2002); Yu (2010); Guédon (2007)), i.e. observation samples that are defined within the sojourn time of the same state are correlated, but observation samples in different states are independent. **The segmental nature of the model allows for easily handling irregular sampling of temporally correlated observation at the cost of introducing discontinuities of the observed data at the state transition times; in all clinical settings of interest, capturing temporal correlations of irregular observations is crucial whereas the continuity of observations is of less relevance.** The model can also be viewed as a state-switching model, but for which the transition dynamics do not need to be linear as in (Georgatzis et al. (2016); Fox et al. (2011a)), but rather depend on the covariance kernel $k_i(t, t')$. Figure 5 depicts the correlation structure of the observable variables in terms of the covariance matrix of a discrete version of $Y(t)$ generated under a specific hidden state sequence. We can see that conditioned on the hidden state sequence, the covariance matrix is a block diagonal matrix, where the sizes of the blocks are random and are determined by the states' sojourn times.

Figure. 6 depicts the graphical model for an HASMM. In Figure. 6, the variables y_n are function-valued and correspond to the finite-duration, continuous-time functions $\{Y_n(t)\}_{t \in \mathcal{T}_n}$. **The arrow between the (function-valued) y_n and the latent state x_{n+1} designates the dependence of the transition probabilities on the state sojourn time (i.e. the domain over which y_n is non-zero).** In Appendix A, we present an algorithm (`GenerateHASMM(`Γ`)`) for sampling episodes from an HASMM with a hyper-parameter set Γ; Figure 7 depicts an exemplary episode sampled via Algorithm 7.

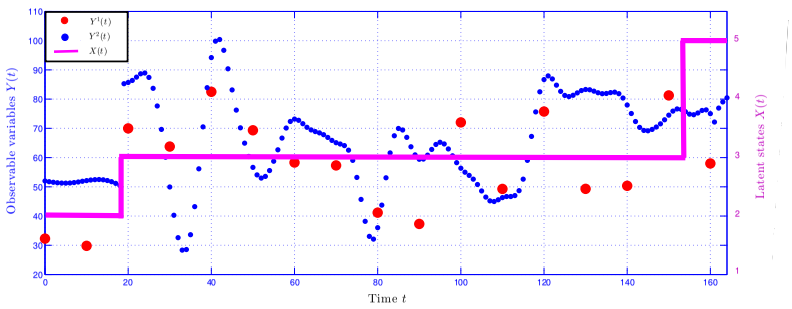


Figure 7: An episode generated by `GenerateHASMM(Γ)` with $N = 5$. The realized hidden state sequence (upper) is $\{2, 3, 5\}$, and is absorbed in state 5. The physiological stream $(Y^1(t), Y^2(t))$ is 2-dimensional and stream $Y^2(t)$ is sampled more intensely than $Y^1(t)$.

3. Inference in Hidden Absorbing Semi-Markov Models

In this Section, we develop an online algorithm that carries out diagnostic and prognostic inferences for a monitored patient's episode in real-time. Given an ongoing realization of an episode $\{y(t_1), y(t_2), \ldots, y(t_m)\}$ at time t_m (before the censoring time T_c), and the HASMM model parameter Γ that has generated this realization, we aim at carrying out the following inference tasks:

- **Diagnosis:** Infer the patient's current clinical state, i.e. compute

$$\mathbb{P}(X(t_m) = j \,|\, Y(t_1) = y(t_1), \ldots, Y(t_m) = y(t_m), \Gamma), \forall j \in \mathcal{X}.$$

- **Prognostic Risk Scoring:** Compute the patient's risk of absorption in the catastrophic state, i.e.

$$\mathbb{P}(\mathcal{A}_N \,|\, Y(t_1) = y(t_1), \ldots, Y(t_m) = y(t_m), \Gamma).$$

In the rest of this Section, we drop the conditioning on Γ for notational brevity. The first inference task corresponds to disease severity estimation for patients with chronic disease, or clinical acuity assessment for critical care patients. The second task corresponds to risk scoring for future adverse events for patients who have been monitored for some period of time, i.e. the risk of developing a future preclinical or clinical breast cancer state (Gail and Mai (2010)), the risk of clinical deterioration for post-operative patients in wards (Rothman et al. (2013)), the risk of mortality for ICU patients (Knaus et al. (1985)), etc.

3.1 Challenges facing the HASMM Inference Tasks

The inference tasks discussed in the previous Subsection are confronted with 3 main challenges –listed hereunder– that hinder the direct deployment of classical forward-backward message-passing routines.

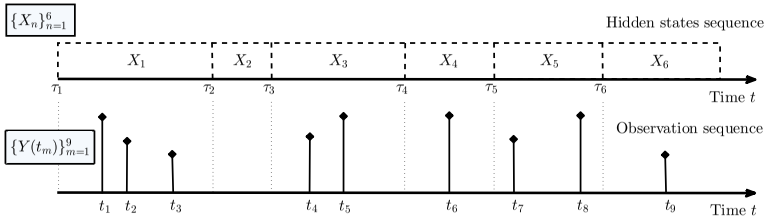


Figure 8: An exemplary HASMM episode with 6 hidden state realizations and 9 observed samples.

1. In addition to the clinical states $\{X_n\}_{n=1}^K$ being unobserved, the transition times among the states, $\{\tau_n\}_{n=1}^K$, are also unobserved (i.e. we do not know the time at which the patient's state changed). Thus, unlike the discrete-time models in (Murphy (2002); Johnson and Willsky (2013); Yu (2010); Dewar et al. (2012); Guédon (2007)), in which we know that the underlying states switch sequentially in a (known) one-to-one correspondence with the observations, in an HASMM the association between states and observations is unknown. Figure 8 depicts an exemplary HASMM episode with 6 realized states and 9 observations samples; in this realization, the association between the observations $\{Y(t_1), Y(t_2), Y(t_3)\}$ and state X_1 is hidden. The importance of reasoning about the hidden transition times is magnified by the duration-dependence of the transition probabilities that govern the sequence $\{X_n\}_{n=1}^K$.

2. Since observations are made at random and arbitrary time instances, some transitions may not be associated with any evidential data. That is, as it is the case for state X_2 in Figure 8, there is no guarantee that for every state X_n, an observation is drawn during its occupancy, i.e. $[\tau_n, \tau_{n+1})$. In a practical setting, the inference algorithm should be able to reason about the state trajectories even in silence periods that come with no observations (recall the example in Figure 2 where observations of a critical care patient's systolic blood pressure stop for an entire day). Hence, one cannot directly discretize the time variable and use the discrete-time HMM inference algorithms (e.g. the algorithms in (Rabiner (1989))) since in that case we would exhibit time steps that come with no associated observations, and with potential state transitions.

3. The HASMM model assumes that observations that belong to the same state are correlated (e.g. in Figure 8, each of the subset of observations $\{Y(t_1), Y(t_2), Y(t_3)\}$, $\{Y(t_4), Y(t_5)\}$ and $\{Y(t_7), Y(t_8)\}$ are not drawn independently conditioned on the latent state since they are sampled from a GP), thus we cannot use the variable-duration and explicit-duration HSMM inference algorithms in (Murphy (2002); Johnson and Willsky (2013); Yu (2010); Guédon (2007)), as those assume that all observations are conditionally independent given the latent states. Our model is closer to a segment-HSMM model (Yu (2010); Guédon (2007)), but with irregularly spaced observations and an underlying duration-dependent state evolution process, which requires a different construction of the forward messages.

In the following Subsection, we develop a forward filtering algorithm that deals with episodes generated from an HASMM and addresses the above challenges.

3.2 The HASMM Forward Filtering Algorithm

Given a realization of an episode $\{y(t_1), y(t_2), \ldots, y(t_m)\}$ at time t_m, the posterior probability of the patient's current clinical state $X(t_m)$ is given by[15]

$$\mathbb{P}(X(t_m) = j \,|\, y(t_1), \ldots, y(t_m), \mathcal{T}) = \frac{d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}{d\mathbb{P}(y(t_1), \ldots, y(t_m)|\mathcal{T})} = \frac{d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}{\sum_{j'=1}^{N} d\mathbb{P}(X(t_m) = j', y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})}. \quad (7)$$

The above application of Bayes' rule implies that, given the observation times $\mathcal{T}$, computing the joint probability density $d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})$ suffices for computing the posterior probability of the patient's clinical states. As it is the case for the conventional HMM setting, we denote these joint probabilities as the *forward messages* $\alpha_m(j \,|\, \mathcal{T}) = d\mathbb{P}(X(t_m) = j, y(t_1), \ldots, y(t_m) \,|\, \mathcal{T})$.

Since the HASMM is a segment model, the conventional notion of the forward messages $\alpha_m(j \,|\, \mathcal{T})$ does not suffice for constructing the forward filtering algorithm since we need to account for the latent correlation structures between the (conditionally-dependent) observations (Murphy (2002)). To that end, we define $\alpha_m(j, w \,|\, \mathcal{T})$ as the forward message for the j^{th} state at the m^{th} observation time (i.e. t_m) *with a lag* w as follows

$$\alpha_m(j, w \,|\, \mathcal{T}) = d\mathbb{P}(X(t_m) = j, \{t_u\}_{u=m-w+1}^{m} \in \mathcal{T}_n, t_{m-w} \in \mathcal{T}_{n'}, \{y(t_u)\}_{u=1}^{m} \,|\, \mathcal{T}), \quad (8)$$

for some $n, n^{'} \in \mathbb{N}_+$, and $n \neq n^{'}$. That is, the forward message $\alpha_m(j, w \,|\, \mathcal{T})$ is simply the joint probability that the current state is j, that the associated observations are $(y(t_1), \ldots, y(t_m))$, and that the current state has lasted for the last w measurements. For notational brevity, denote the event $\left\{\{t_u\}_{u=m-w+1}^{m} \in \mathcal{T}_n, t_{m-w} \in \mathcal{T}_{n'}\right\}$ as $\psi(m, w)$. Thus, $\alpha_m(j, w \,|\, \mathcal{T})$ can be written as

$$\alpha_m(j, w \,|\, \mathcal{T}) = \sum_{i=1}^{N} \sum_{w'=1}^{m-w} d\mathbb{P}(X(t_m) = j, \psi(m, w), X(t_{m-w}) = i, \psi(m - w, w^{'}), \{y(t_u)\}_{u=1}^{m} \,|\, \mathcal{T}),$$

which can be decomposed using the conditional independence properties of the states, observable variables and sojourn times as follows

$$d\mathbb{P}(X(t_m) = j, \psi(m, w), X(t_{m-w}) = i, \psi(m - w, w^{'}), \{y(t_u)\}_{u=1}^{m}) =$$

$$d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j, \psi(m, w)) \times \underbrace{\mathbb{P}(X(t_m) = j \,|\, X(t_{m-w}) = i, \psi(m - w, w^{'}))}_{p_{ij}(t_m - t_{m-w}, \psi(m-w, w'))} \times$$

$$\underbrace{\mathbb{P}(\psi(m, w) \,|\, X(t_m) = j)}_{V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j)} \times \underbrace{d\mathbb{P}(X(t_{m-w}) = i, \psi(m - w, w^{'}), \{y(t_u)\}_{u=1}^{m-w})}_{\alpha_{m-w}(i, w')}, \quad (9)$$

15. We use the notation $d\mathbb{P}$ to denote a probability density defined with respect to $(\Omega, \mathcal{F}, \mathbb{P})$.

where we have dropped the conditioning on $\mathcal{T}$ for notational brevity. The first term, $d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j, \psi(m, w))$, is the probability density of the observable variables in $\{y(t_u)\}_{u=m-w+1}^{m}$ conditioned on the hidden state being $X(t_m) = j$ and that the time instances $\{t_u\}_{u=m-w+1}^{m}$ reside in the sojourn time of $X(t_m) = j$. The second term, $p_{ij}(t_m - t_{m-w}, \psi(m - w, w^{'}))$, is the *interval transition probability*, i.e. the probability that the state sequence transits to state j after a period $t_m - t_{m-w}$, given that its sojourn time in state $X(t_{m-w}) = i$ at t_m is at least $t_m - t_{m-w+1}$, and at most $t_m - t_{m-w-w'}$. The third term is the probability that the sojourn time in state $X(t_m) = j$ is between $t_m - t_{m-w+1}$ and $t_m - t_{m-w}$, whereas the fourth term, $\alpha_{m-w}(i, w^{'})$, is the $(m-w)^{th}$ forward message with a lag of $w^{'}$. Thus, we can write the m^{th} forward message with a lag w as follows

$$\alpha_m(j, w) = d\mathbb{P}(\{y(t_u)\}_{u=m-w+1}^{m} \,|\, X(t_m) = j) \times$$

$$\sum_{i=1}^{N} \sum_{w'=1}^{m-w} p_{ij}(t_m - t_{m-w}, \psi(m - w, w^{'})) \cdot (V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j)) \cdot \alpha_{m-w}(i, w^{'}). \quad (10)$$

As we can see in (10), one can express $\alpha_m(j, w)$ using a recursive formula that makes use of the older forward messages $\{\alpha_{m-w}(i, w^{'})\}_{w=1}^{m}$, where $\alpha_o(i, w^{'}) = 0$, which allows for an efficient dynamic programming algorithm to infer the patient's clinical state in real-time.

The construction of the forward messages in (10) parallels the structure of forward message-passing in segment-HSMM (See Section 1.2 in (Murphy (2002)) and Section 4.2.2 in (Yu (2010))), but with the following differences. In (10), the time interval between every two observation samples is irregular, which reflects in the correlation between the observations in $\{y(t_u)\}_{u=m-w+1}^{m}$ (depends on the covariance kernel of the GP, and the probability of the current latent state's sojourn time being encompassing the most recent w samples, i.e. $(V_j(t_m - t_{m-w}|\lambda_j) - V_j(t_m - t_{m-w+1}|\lambda_j))$. However, the most challenging ingredient of the forward message is the interval transition probability $p_{ij}(t_m - t_{m-w}, \psi(m - w, w^{'}))$. This is because unlike the discrete-time HSMM models in (Murphy (2002); Yu (2010)), which exhibit transitions only at discrete time steps that are always accompanied with evidential observations, i.e. no hidden transitions can occur between observation samples, and the transitions among hidden states are duration-independent, in an HASMM, transitions can occur at arbitrary time instances, multiple transitions can occur between two observation samples, and transitions are duration-dependent.

In order to evaluate the term $p_{ij}(t_m - t_{m-w}, \psi(m - w, w^{'}))$, we construct a virtual (discrete-time) trivariate *embedded Markov chain* $\{X(t_m), t_{m-w}, t_{m-w+1}\}$, the transition probabilities of which are equal to the interval transition probabilities. In the recent work in (Liu et al. (2015)), a similar embedded Markov chain analysis was conducted for a CT-HMM (Continuous-time HMM), but for which the underlying state evolution process was assumed to be a duration-independent ordinary Markov chain for which the expressions for the interval transition probabilities are readily available by virtue of the exponential distributions of the memoryless state sojourn times.

Recall from Subsection 2.1.1 that the semi-Markov kernel of the hidden state sequence $\{X_n\}_{n=1}^K$ is defined as $Q_{ij}(\tau) = \mathbb{P}(X_{n+1} = j, S_n \leq \tau | X_n = i)$, i.e. the probability that the sequence transits from state i to state j given that the sojourn time in i is less than or equal to τ. Theorem 1 establishes the methodology for computing the interval transition probabilities $p_{ij}(t_m - t_{m-w}, \psi(m-w, w'))$ using the parameters of an HASMM. In Theorem 1, we define $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ as a matrix-valued function $\tilde{\mathbf{P}} : \mathcal{S} \rightarrow [0,1]^{N \times N}$, $\mathcal{S} = \{(\tau, \underline{s}, \bar{s}) : \tau \in \mathbb{R}_+, \bar{s} \in \mathbb{R}_+, \underline{s} \leq \bar{s}\}$, the entries of which are given by

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \underbrace{\left[\begin{array}{c|c|c|c} \tilde{p}_{11}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{21}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{N1}(\tau, \underline{s}, \bar{s}) \\ \tilde{p}_{12}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{22}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{N2}(\tau, \underline{s}, \bar{s}) \\ \vdots & \vdots & & \vdots \\ \tilde{p}_{1N}(\tau, \underline{s}, \bar{s}) & \tilde{p}_{2N}(\tau, \underline{s}, \bar{s}) & \cdots & \tilde{p}_{NN}(\tau, \underline{s}, \bar{s}) \end{array}\right]}_{\text{Size } N \times N \text{ matrix}}.$$

In addition, we define a *truncated semi-Markov kernel* as

$$\bar{Q}_{ij}(\tau, \underline{s}, \bar{s}) = \int_{s=\underline{s}}^{\bar{s}} (\bar{g}_{ij}(\tau + s) - \bar{g}_{ij}(s)) \cdot \frac{V_i(\tau + s|\lambda_i) - V_i(s|\lambda_i)}{1 - V_i(s|\lambda_i)} \cdot dV_i(s|\lambda_i),$$

a scalar-valued function $\bar{Q}_i(\tau, \underline{s}, \bar{s}) = \sum_{j \in \mathcal{X} \setminus \{i\}} \bar{Q}_{ij}(\tau, \underline{s}, \bar{s})$, and a matrix-valued function

$$\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s}) = \underbrace{\left[\begin{array}{c|c|c|c} 0 & \bar{Q}_{21}(\tau, \underline{s}, \bar{s}) & \cdots & \bar{Q}_{N1}(\tau, \underline{s}, \bar{s}) \\ \bar{Q}_{12}(\tau, \underline{s}, \bar{s}) & 0 & \cdots & \bar{Q}_{N2}(\tau, \underline{s}, \bar{s}) \\ \vdots & \vdots & & \vdots \\ \bar{Q}_{1N}(\tau, \underline{s}, \bar{s}) & \bar{Q}_{2N}(\tau, \underline{s}, \bar{s}) & \cdots & 0 \end{array}\right]}_{\text{Size } N \times N \text{ matrix}}.$$

Theorem 1 (Interval transition probabilities) *Let $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ be the solution to the following integral equation*

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathbf{I}_{N \times N} - \text{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \int_{u=0}^{\tau} \frac{\partial \bar{\mathbf{Q}}(u, \underline{s}, \bar{s})}{\partial u} \times \tilde{\mathbf{P}}(\tau - u, 0, 0)\, du, \tag{11}$$

for the three independent variables $(\tau, \underline{s}, \bar{s}) \in \mathcal{S}$. Then, the interval transition probability p_{ij} is given by $p_{ij}(t_m - t_{m-w}, \psi(m-w, w')) = \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}), \forall i, j \in \mathcal{X}$, at $\tau = t_m - t_{m-w}$, $\underline{s} = t_m - t_{m-w+1}$, and $\bar{s} = t_m - t_{m-w+w'}$.

Proof *See Appendix B.* ■

Theorem 1 follows from a *first-step analysis* that is akin to the derivation of the conventional Chapman-Kolmogorov equations in ordinary Markov chains (Kulkarni (1996)). The integral equation in (11) is a (matrix-valued) non-homogeneous *Volterra integral equation of the second kind* (Polyanin and Manzhirov (2008)). It can be easily demonstrated that a closed-form solution that hinges on conventional kernel methods cannot be obtained. Hence, we

resort to a numerical method in order to solve (11) for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$, $\forall (\tau, \underline{s}, \bar{s}) \in \mathcal{S}$. Before presenting the numerical method, we reformulate (11) as follows

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathbf{I}_{N \times N} - \text{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \left(\frac{\partial \bar{\mathbf{Q}}(., \underline{s}, \bar{s})}{\partial u} \star \tilde{\mathbf{P}}(., 0, 0)\right)(\tau), \tag{12}$$

where $\star$ is an element-wise convolution operator. (12) follows from (11) by the fact that the integral in (11) is a convolution integral; (12) can be expressed as follows

$$\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})), \tag{13}$$

where the (functional) operator $\mathcal{B}\{\mathbf{Q}\}(\tilde{\mathbf{P}})$ is given by

$$\begin{aligned} &\mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})) = \\ &\mathbf{I}_{N \times N} - \text{diag}\left(\bar{Q}_1(\tau, \underline{s}, \bar{s}), \ldots, \bar{Q}_N(\tau, \underline{s}, \bar{s})\right) + \mathscr{F}^{-1}\left\{\mathscr{F}\left\{\frac{\partial \bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})}{\partial \tau}\right\} \cdot \mathscr{F}\left\{\tilde{\mathbf{P}}(\tau, 0, 0)\right\}\right\}, \end{aligned} \tag{14}$$

where $\mathscr{F}$ is the Fourier transform operator, and the transforms in (14) are all taken with respect to τ.

The solution to (13) can be obtained via the *successive approximation* method (Opial (1967)) as follows. We initialize the function $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ with the truncated semi-Markov kernel[16] $\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})$, and then iteratively apply the operator $\mathcal{B}(.)$ to obtain a new value for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ until convergence. That is, the successive approximation procedure goes as follows

$$\begin{aligned} &\tilde{\mathbf{P}}^o(\tau, \underline{s}, \bar{s}) = \bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s}) \\ &\text{While } \left\| \tilde{\mathbf{P}}^z(\tau, \underline{s}, \bar{s}) - \tilde{\mathbf{P}}^{z-1}(\tau, \underline{s}, \bar{s}) \right\|_\infty > \epsilon \\ &\tilde{\mathbf{P}}^z(\tau, \underline{s}, \bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau, \underline{s}, \bar{s})\}(\tilde{\mathbf{P}}^{z-1}(\tau, \underline{s}, \bar{s})). \end{aligned} \tag{15}$$

The following Theorem establishes the validity of the procedure in (15) as a solver for (13). Before presenting the statement of Theorem 2, we define the function space $\mathcal{P}$ as follows

$$\mathcal{P} = \left\{ \tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) : \ \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}) \in [0,1], \sum_j \tilde{p}_{ij}(\tau, \underline{s}, \bar{s}) = 1, \tilde{p}_{ij}(0, \underline{s}, \bar{s}) = \delta_{ij}, (\tau, \underline{s}, \bar{s}) \in \mathcal{S} \right\},$$

where δ_{ij} is the Kronecker delta function.

Theorem 2 (Convergence of successive approximations) *The functional $\mathcal{B}\{\bar{\mathbf{Q}}\}(\tilde{\mathbf{P}})$ has a unique fixed-point $\tilde{\mathbf{P}}^*$ in $\mathcal{P}$, and the successive approximation procedure in (15) always converges to the fixed point, i.e. $\tilde{\mathbf{P}}^\infty(\tau, \underline{s}, \bar{s}) = \tilde{\mathbf{P}}^*(\tau, \underline{s}, \bar{s})$, starting from any initial value $\tilde{\mathbf{P}}^o(\tau, \underline{s}, \bar{s}) \in \mathcal{P}$.*

Proof *See Appendix C.* ■

16. This is a reasonable initialization since the entries of the semi-Markov kernel correspond to interval transition probabilities conditioned on there being no intermediate transitions on the way from state i to state j.

Algorithm 1 Constructing a look-up table of interval transition probabilities

procedure TransitionLookUp(Γ, ϵ)
  Input: HASMM parameters Γ and precision ϵ
  Output: A look-up table $[\tilde{p}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})]_{i,j,a,b,c}$
  Set the values of A, B and C (number of steps), $\Delta\tau$ (step sizes)
  for $a = 1$ to A, $b = 1$ to B, $c = 1$ to C **do**
    $g^{\tau}_{ij}(a\Delta\tau) \leftarrow \sum_{x=1}^{a} \frac{e^{(\eta_{ij}+\beta_{ij}x\Delta\tau)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}x\Delta\tau)}} \left(\frac{1}{\Gamma(\lambda_{i,s})\,\lambda_{i,r}^{\lambda_{i,s}}} (x\Delta\tau)^{\lambda_{i,s}-1} e^{-\frac{x\Delta\tau}{\lambda_{i,r}}} \right) \Delta\tau$
    $g^{s}_{ij}(a\Delta s) \leftarrow \sum_{x=1}^{a} \frac{e^{(\eta_{ij}+\beta_{ij}x\Delta s)}}{\sum_{k=1}^{N} e^{(\eta_{ik}+\beta_{ik}x\Delta s)}} \left(\frac{1}{\Gamma(\lambda_{i,s})\,\lambda_{i,r}^{\lambda_{i,s}}} (x\Delta s)^{\lambda_{i,s}-1} e^{-\frac{x\Delta s}{\lambda_{i,r}}} \right) \Delta s$
    $\bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \sum_{x=b}^{c} \frac{(g^{\tau}_{ij}(a\Delta\tau) - g^{s}_{ij}(x\Delta s))\,(V_i(a\Delta\tau|\lambda_i) - V_i(x\Delta s|\lambda_i))}{1 - V_i(x\Delta s|\lambda_i)}\, v_i(x\Delta s|\lambda_i)$
  end for
  $e = \epsilon + 1$
  $z \leftarrow 1$
  $\tilde{p}^{(o)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$, $\forall a, b, c, i, j$.
  while $e > \epsilon$ **do**
    $CQ_{i,j,k}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow$
      $\text{IFFT}\left(\text{FFT}\left(\text{diff}\left(\bar{Q}_{ik}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})\right)\right), \text{FFT}\left(\tilde{p}^{(z-1)}_{jk}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})\right)\right),$
    $\tilde{p}^{(z)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow \delta_{ij}\,\bar{Q}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) + \sum_{k=1}^{N} CQ_{i,j,k}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$
    $\tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) = \left[\tilde{p}^{(z)}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}))\right]_{i,j,a,b,c}$
    $e \leftarrow \left\| \tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) - \tilde{\mathbf{P}}^{(z-1)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \right\|_{\infty}$
    $z \leftarrow z + 1$
  end while
  return $\tilde{\mathbf{P}}^{(z)}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$
end procedure

It is important to note that we do not need to solve for $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$ during real-time inference. Instead, we create a look-up table comprising a discretized version of $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s}) = [\tilde{p}_{ij}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}))]_{i,j,a,b,c}$, and then we query this table when performing real-time inference for monitored patients. Hence, efficient and fast inferences can be provided for critical care patients for whom prompt diagnoses are necessary for the efficacy of clinical interventions. Algorithm 1 shows a pseudocode for constructing a look-up table of interval transition probabilities, `TransitionLookUp`(Γ, ϵ), which takes as an input the parameter set Γ and a precision level ϵ (to control the termination of the successive approximation iterations), and outputs the interval transitions look-up table $\tilde{\mathbf{P}}(\tau, \underline{s}, \bar{s})$. In Algorithm 1, FFT and IFFT refer to the fast Fourier transform operation and its inverse, respectively, and "diff(.)" refers to a numerical differentiation operation.

Now that we have constructed the algorithm `TransitionLookUp` to compute the interval transition probabilities in the look-up table $\tilde{\mathbf{P}}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s})$, we can implement a forward-filtering inference algorithm using dynamic programming (by virtue of the recursive formula in (10)). In particular, the posterior probability of the patient's current clinical state in

Algorithm 2 Forward filtering inference

procedure ForwardFilter(Γ, $\{y(t_w)\}_{w=1}^{m}$, ϵ)
  Input: Observed samples $\{y(t_w)\}_{w=1}^{m}$, HASMM parameters Γ, and precision ϵ
  Output: The posterior state distribution $\{\mathbb{P}(X(t_m) = j \,|\, \{y(t_w)\}_{w=1}^{m})\}_{j=1}^{N}$
  $\tilde{\mathbf{P}}(a\Delta\tau, b\Delta\underline{s}, c\Delta\bar{s}) \leftarrow$ `TransitionLookUp`(Γ, ϵ)
  $\alpha_1(j,1) = d\mathbb{P}(y(t_1) \,|\, X(t_1) = j)\, \sum_{i=1}^{N} \tilde{p}_{ij}(t_1, 0, 0) \cdot p_i^{o},\ \forall j \in \mathcal{X}$
  for $j = 1$ **to** N
  for $z = 2$ to m **do**
    for $w = 1$ to z **do**
      $a^*(z,w) = \arg\min_a |t_z - t_{z-w} - a\Delta\tau|$
      $b^*(z,w) = \arg\min_b |t_z - t_{z-w+1} - b\Delta\underline{s}|$
      $c^*(z,w,w') = \arg\min_c \left|t_z - t_{z-w-w'} - c\Delta\bar{s}\right|$
      $\alpha_z(j,w) = d\mathbb{P}(\{y(t_u)\}_{u=z-w+1}^{z} \,|\, X(t_z) = j)\, \sum_{i=1}^{N} \sum_{w'=1}^{z-w} \alpha_{z-w}(i,w') \times$

$$\tilde{p}_{ij}(a^*(z,w)\Delta\tau, b^*(z,w)\Delta\underline{s}, c^*(z,w,w')\Delta\bar{s}) \times (V_j(t_z - t_{z-w}|\lambda_j) - V_j(t_z - t_{z-w+1}|\lambda_j))$$

    end for
  end for
  $\mathbb{P}(X(t_m) = j \,|\, \{y(t_u)\}_{u=1}^{m}) = \frac{\sum_{w=1}^{m} \alpha_m(j,w)}{\sum_{k=1}^{N} \sum_{w=1}^{m} \alpha_m(k,w)}$
  return $\{\mathbb{P}(X(t_m) = j \,|\, \{y(t_w)\}_{w=1}^{m})\}_{j=1}^{N}$
end procedure

terms of the forward messages can be written as

$$\mathbb{P}(X(t_m) = j \,|\, y(t_1), \ldots, y(t_m)) = \frac{\sum_{w=1}^{m} \alpha_m(j,w)}{\sum_{k=1}^{N} \sum_{w=1}^{m} \alpha_m(k,w)}. \tag{16}$$

Algorithm 2, `ForwardFilter`, implements real-time inference of a patient's clinical state given a sequence of measurements $\{y(t_1), \ldots, y(t_m)\}$. In Algorithm 2, we invoke `TransitionLookUp` initially to construct the look-up table of transition probabilities, but in practice, the look-up table can be constructed in an offline stage once the HASMM parameter set Γ is known.

The number of computations can be reduced by limiting the lags w for every forward message $\alpha_m(j,w)$ to a maximum of W lags. Ignoring the computations involved in evaluating the GP likelihoods, the complexity of `ForwardFilter` is $\mathcal{O}(mWN + mN^2)$. Since evaluating the GP likelihoods is cubic in the number of observations, the worst caset complexity of `ForwardFilter` is $\mathcal{O}((mWN + mN^2)W^3)$. (In most practical clinical settings of interest, the number of observations W can be restricted to include the most recent few samples.)

3.3 Prognostic risk scoring using an HASMM

Diagnostic inference, e.g. estimating the patient's current state after a screening test, can be conducted by a direct application of the forward filtering algorithm presented in the previous Subsection. Prognostic risk scoring plays an important role in designing screening guidelines

(Gail and Mai (2010)), acute care interventions (Knaus et al. (1985)) and surgical decisions (Foucher et al. (2007)). A risk score is a measure for the patient's risk of encountering an adverse event (abstracted as state N in our model) at any future time step starting from time t_m. That is, the patient's risk score at time t_m can be formulated as

$$\begin{aligned} R(t_m) &= \mathbb{P}(\mathcal{A}_N \,|\, \{y(t_u)\}_{u=1}^{m}, \Gamma) \\ &= 1 - \mathbb{P}(X(\infty) = N \,|\, \{y(t_u)\}_{u=1}^{m}, \Gamma), \end{aligned} \tag{17}$$

which can be computed using the outputs of `TransitionLookUp` and `ForwardFilter` as follows

$$R(t_m) = \sum_{j=1}^{N} \tilde{p}_{jN}(A, 0, 0) \cdot \frac{\sum_{w=1}^{m} \alpha_m(j, w)}{\sum_{k=1}^{N} \sum_{w=1}^{m} \alpha_m(k, w)}. \tag{18}$$

Therefore, the procedures `TransitionLookUp` and `ForwardFilter` suffice for executing both the diagnostic and prognostic inference tasks.

4. Learning Hidden Absorbing Semi-Markov Models

In Section 3.1, we developed an inference algorithm that can handle diagnostic and prognostic tasks for patients in real-time assuming that the true HASMM parameter set Γ is known. In practice, the parameter set Γ is not known, and has to be learned from an offline EHR dataset $\mathcal{D}$ that comprises D episodes for previously hospitalized or monitored patients, i.e.

$$\mathcal{D} = \left\{ \{y_m^d, t_m^d\}_{m=1}^{M^d}, T_c^d, l^d \right\}_{d=1}^{D},$$

where $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ are the observable variables and sampling times for the d^{th} episode, T_c^d is the episode's censoring time, and $l^d \in \{1, N\}$ is a label for the realized absorbing state.

We note that unlike the conventional HMM learning setting (Rabiner (1989); Zhang et al. (2001); Nodelman et al. (2012)), the episodes are not of equal-length as the observations for every episode stop at a random, but informative, censoring time. Thus, the patient's state trajectory does not manifest only in the observable time series, i.e. $\{y_m^d, t_m^d\}_{m=1}^{M^d}$, but also in the episode's censoring variables $\{T_c^d, l^d\}$. In this Section, we develop an efficient algorithm, which we call the *forward-filtering backward-sampling Monte Carlo EM* (FFBS-MCEM) algorithm, that computes the Maximum Likelihood (ML) estimate of Γ given an informatively censored dataset $\mathcal{D}$, i.e. $\Gamma^* = \arg\max_\Gamma \Lambda(\mathcal{D} \,|\, \Gamma)$, where $\Lambda(\mathcal{D} \,|\, \Gamma) = d\mathbb{P}(\mathcal{D} \,|\, \Gamma)$ is the likelihood of the dataset $\mathcal{D}$ given the parameter set Γ. We start by presenting the learning setup in Section 4.1, and then we present the FFBS-MCEM algorithm Section 4.3.

4.1 The Learning Setup

We focus on the challenging scenario when no domain knowledge or diagnostic assessments for the patients' latent states are provided in the dataset [17] $\mathcal{D}$ (with the exception of the absorbing state which is declared by the variable l^d), i.e. the learning setup is an *unsupervised* one. For such a scenario, the main challenge in constructing the ML estimator Γ^* resides in the hiddenness of the patients' state trajectories in the training dataset $\mathcal{D}$; the dataset $\mathcal{D}$ contains only the sequence of observable variables, their respective observation times, the episode's censoring time and the state in which the trajectory was absorbed. If the patients' latent state trajectories $(X(t))_{t\in\mathbb{R}_+}$ were observed in $\mathcal{D}$, the ML estimation problem $\Gamma^* = \arg\max_\Gamma \mathbb{P}(\mathcal{D} \,|\, \Gamma)$ would have been straightforward; the hiddenness of $(X(t))_{t\in\mathbb{R}_+}$ entails the need for marginalizing over the space of all possible latent trajectories conditioned on the observed variables, which is a hard task even for conventional CT-HMM models (Liu et al. (2015); Nodelman et al. (2012); Leiva-Murillo et al. (2011); Metzner et al. (2007)).

We start by writing the complete likelihood, i.e. the likelihood of an HASMM with a parameter set Γ to generate both the hidden states trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ and the observable variables $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ for the d^{th} episode in the dataset $\mathcal{D}$ as follows

$$\begin{aligned} &d\mathbb{P}\left(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,\middle|\, \Gamma \right) = \mathbb{P}(x_1^d|\Gamma) \cdot d\mathbb{P}(s_1^d|x_1^d, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_1^d}|x_1^d, \Gamma) \times \\ &\prod_{n=2}^{k^d} \mathbb{P}(x_n^d \,|\, x_{n-1}^d, s_{n-1}^d, \Gamma) \cdot d\mathbb{P}(s_n^d \,|\, x_n^d, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^d, \Gamma), \end{aligned} \tag{19}$$

where k^d is the number of states that realized in episode d from $t = 0$ until absorption. The factorization in (19) follows from the conditional independence properties of the HASMM variables (see Figure 6). Since we cannot observe the latent states trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$, the ML estimator deals with the expected likelihood $\Lambda(\mathcal{D} \,|\, \Gamma)$, which is evaluated by marginalizing the complete likelihood over the latent states trajectories, i.e.

$$\begin{aligned} \Lambda(\mathcal{D} \,|\, \Gamma) &= \mathbb{E}_{x(t)|\mathcal{D},\Gamma} \left[\prod_{d=1}^{D} d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma) \right] \\ &= \prod_{d=1}^{D} \mathbb{E}_{x^d(t)|\mathcal{D},\Gamma} \left[d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma) \right] \\ &= \prod_{d=1}^{D} \int d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma) \cdot d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \mathcal{D}, \Gamma), \end{aligned} \tag{20}$$

where the expectation is taken with respect to the latent trajectory conditioned on the observed dataset $\mathcal{D}$, **which contains the information on every episode's censoring time T_c^d and terminating state l^d**. The integral in (20) can be further decomposed as follows

$$\begin{aligned} \Lambda(\mathcal{D} \,|\, \Gamma) = \prod_{d=1}^{D} \int & \mathbb{P}(x_1^d|\Gamma) \cdot d\mathbb{P}(s_1^d|x_1^d, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_1^d}|x_1^d, \Gamma) \cdot d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \mathcal{D}, \Gamma) \\ & \times \prod_{n=2}^{k^d} \mathbb{P}(x_n^d \,|\, x_{n-1}^d, s_{n-1}^d, \Gamma) \cdot d\mathbb{P}(s_n^d \,|\, x_n^d, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^d, \Gamma). \end{aligned} \tag{21}$$

17. For some settings, such as chronic kidney disease progression estimation (Eddy and Neilson (2006)), the EHR records may include some anchors or assessments to the latent states over time. A simpler version of the learning algorithm proposed in this Section can be used to deal with such datasets. In critical care settings, it is more common that the EHR records are not labeled with any clinical state assessments over time (Yoon et al. (2016)).

4.2 Challenges Facing the HASMM Learning Task

The problem of learning the HASMM parameters by maximizing the likelihood function in (21) is obstructed by various obstacles that hinder the deployment of off-the-shelf learning algorithms; we list these challenges hereunder.

1. Finding the ML estimate Γ^* by direct maximization of $\Lambda(\mathcal{D}\,|\,\Gamma)$ is not viable due to the intractability of the integral in (21), i.e. $\Lambda(\mathcal{D}\,|\,\Gamma)$ has no analytic maximizer. The difficulty of evaluating the expected likelihood $\Lambda(\mathcal{D}\,|\,\Gamma)$ follows from the need to average the complete likelihood over a complicated posterior density function for the latent state trajectory.

2. Direct adoption of the conventional Baum-Welch implementation of the EM algorithm as a solution to the intractable problem of maximizing the expected likelihood in (21) –as has been applied in HMMs (Rabiner (1989)), HSMMs (Murphy (2002)), EDHMMs and VDHMMs (Yu (2010)– is not possible for the HASMM setting. This is due to the intractability of the integral involved in the E-step; a problem that is also faced by other continuous-time models (Liu et al. (2015); Nodelman et al. (2012)). However, these models assumed Markovian state trajectories, in which case the implementation of the E-step boils down to computing the expected state durations and transition counts as sufficient statistics for estimating the latent trajectories[18] (e.g. see Equations (12) and (13) in (Liu et al. (2015))). This simplification, which follows from the plausible properties of the Markov chain's transition rate matrix, does not materialize for semi-Markovian transitions. Further complications are introduced by the duration-dependence of the state-transitions and the segmental nature of the observables.

3. Learning an informatively censored dataset would naturally benefit from the information conveyed in the censoring variables $\{T_c^d, l^d\}$. However, the availability of censoring information leads to more complicated posterior density expressions for the latent state trajectories, which complicates the job of any analytic, variational or Monte Carlo based inference method one would use to infer the latent state trajectories[19].

In the following Section, we present a learning algorithm that addresses the above challenges, and provides insights into general settings in which informatively censored time series data are to be dealt with.

18. Different approaches have been developed in the literature for computing these quantities: (Wang et al. (2014)) assumes that the transition rate matrix is diagonalizable, and hence utilize a closed-form estimator for the transition rates, whereas (Liu et al. (2015)) uses the *Expm* and *Unif* methods (originally developed in (Hobolth and Jensen (2011))) to evaluate the integrals of the transition matrix exponential. Unfortunately, none of these methods could be utilized for computing the proximal log-likelihood of an HASMM due to the semi-Markovianity of the state trajectory (i.e. state-durations are not exponentially distributed as it is the case in (Liu et al. (2015); Nodelman et al. (2012); Hobolth and Jensen (2011); Wang et al. (2014))).

19. Note that the censoring information are only available in the model training (learning) phase since we deal with an offline batch of data through which we can see the full patients' episodes, whereas real-time inference, discussed in the previous Section, does not take advantage of any external censoring information.

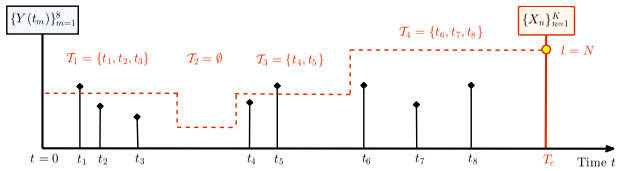


Figure 9: An episode that comprised 8 observable samples, censored at time T_c, and absorbed in state N (catastrophic state). The dashed state trajectory is a trajectory that could have generated the observables with a positive probability. Computing the proximal log-likelihood requires averaging over infinitely many paths that could have generated the observables with a positive probability.

4.3 The Forward-filtering Backward-sampling Monte Carlo EM Algorithm

4.3.1 Expectation-Maximization

As in the case of classical discrete and continuous-time HMMs, we address the first challenge stated in Section 4.2 by using the EM algorithm (Liu et al. (2015); Nodelman et al. (2012)). The iterative EM algorithm starts with an initial guess $\hat{\Gamma}^o$ for the parameter set, and maximizes a proxy for the log-likelihood in the z^{th} iteration as follows:

- **E-step:** $U(\Gamma;\hat{\Gamma}^{z-1}) = \sum_{d=1}^{D} \mathbb{E}\left[\log(\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d}\,|\,\Gamma))\middle|\ \mathcal{D}, \hat{\Gamma}^{z-1}\right]$.
- **M-step:** $\hat{\Gamma}^z = \arg\max_\Gamma U(\Gamma;\hat{\Gamma}^{z-1})$.

The E-step computes the *proximal expected log-likelihood* $U(\Gamma;\hat{\Gamma}^{z-1})$, which entails evaluating the following integral

$$U(\Gamma;\hat{\Gamma}^{z-1}) = \sum_{d=1}^{D} \int \log(d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}, \{y_m^d, t_m^d\}_{m=1}^{M^d}\,|\,\Gamma)) \cdot d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}\,|\,\mathcal{D}, \hat{\Gamma}^{z-1}),$$

where $d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}\,|\,\mathcal{D}, \hat{\Gamma}^{z-1}) = d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d}\,|\,\{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d, \hat{\Gamma}^{z-1})$. That is, the proximal expected log-likelihood $U(\Gamma;\hat{\Gamma}^{z-1})$ is computed by marginalizing the likelihood of the observed samples of the d^{th} episodes $\{y_m^d, t_m^d\}_{m=1}^{M^d}$ over all potential latent paths $(x^d(t))_{t\in\mathbb{R}_+}$ that are censored at time T_c^d and absorbed in state l^d. Figure 9 depicts a set of observables $(\{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d)$ for one episode, and a potential latent path $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ that could have generated such observables. Computing $U(\Gamma;\hat{\Gamma}^{z-1})$ requires averaging over the posterior density of the latent paths conditional on an observable episode.

4.3.2 "The Only Good Monte Carlo is a Dead Monte Carlo"

Since computing $U(\Gamma;\hat{\Gamma}^{z-1})$ does not admit a closed-form solution, as mentioned earlier in the second challenge stated in Section 4.2, we resort to a Monte Carlo approach for approximating the integral involved in the E-step (Caffo et al. (2005)). That is, in the z^{th} iteration of the EM algorithm, we draw G random trajectories $(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}})_{g=1}^{G}$ for every episode

d, and use those trajectories to construct a Monte Carlo approximation $\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1})$ for the proximal log-likelihood function $U(\Gamma;\hat{\Gamma}^{z-1})$. Sample trajectories are drawn from the posterior density of the latent states' trajectory conditional on the the observable variables (including the censoring information). That is to say, the g^{th} sample trajectory $\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}$ is drawn as follows

$$\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} \sim d\mathbb{P}(\{x_n^d, s_n^d\}_{n=1}^{k^d} \,|\, \{y_m^d, t_m^d\}_{m=1}^{M^d}, x^d(T_c^d) = l^d, \hat{\Gamma}^{z-1}), \tag{22}$$

for $g \in \{1,\ldots,G\}$. Hence, the proximal log-likelihood $U(\Gamma;\hat{\Gamma}^{z-1})$ can be approximated via a Monte Carlo estimate $\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1})$ as follows

$$\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1}) \triangleq \sum_{d=1}^{D}\frac{1}{G}\sum_{g=1}^{G}\log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)). \tag{23}$$

It follows from the *Glivenko-Cantelli* Theorem (Durrett (2010)) that

$$||\, U(\Gamma;\hat{\Gamma}^{z-1}) - \hat{U}_G(\Gamma;\hat{\Gamma}^{z-1})\,||_\infty = \sup_{\Gamma} |\, U(\Gamma;\hat{\Gamma}^{z-1}) - \hat{U}_G(\Gamma;\hat{\Gamma}^{z-1})\,| \to 0 \;\text{ a.s.},$$

and hence the Monte Carlo implementation of the E-step becomes more accurate as the sample size G increases. Sampling trajectories from the posterior distribution specified in (22) in order to obtain a Monte Carlo estimate for $U(\Gamma;\hat{\Gamma}^{z-1})$ is not a straight forward task; the sampler needs to jointly sample the states and their sojourn times taking into account the duration-dependent transitions among states, and that the number of variables sampled (number of states) $k^{d,g}$ in each trajectory is itself random.

Since there is no straightforward method that can generate samples for the random state trajectory $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ from the joint posterior density in (22), the normative solution for such a problem is to resort to a Markov Chain Monte Carlo (MCMC) method such as Metropolis-Hastings or Gibbs sampling (Carter and Kohn (1994)). Since the number of state and sojourn time variables, k^d, is itself random, one can even resort to a reversible jump MCMC method (Green and Hastie (2009)) in order to generate the samples for $\{x_n^d, s_n^d\}_{n=1}^{k^d}$. At this point of our analysis, we invoke the classical aphorism with which we titled this Subsection: *"The Only Good Monte Carlo is a Dead Monte Carlo"* (Trotter and Tukey (1956)). By this quote, Trotter meant to advocate the view that sophisticated Monte Carlo methods should be avoided whenever possible; whenever an integral is analytically tractable, or whenever some analytic insights can be exploited to built simpler samplers, doing so should be preferred to an expensive Monte Carlo method. MCMCs are indeed expensive: they mix very slowly and they generate correlated samples. Adopting an MCMC to generate random state trajectories in every iteration of the EM algorithm and for every episode in $\mathcal{D}$ is beyond affordable. Fortunately, in the rest of this Section we show that an efficient sampler that generates independent samples of $\{x_n^d, s_n^d\}_{n=1}^{k^d}$ and for which the run-time is geometrically distributed can be constructed by capitalizing on the censoring information and utilizing some insights from the literature on sequential Monte Carlo smoothing (Godsill et al. (2004)).

4.3.3 The Forward-filtering Backward-sampling Recipe

The availability of the censoring information for every episode d in $\mathcal{D}$, together with the inherent non-linearity of the semi-Markovian transition dynamics encourage the development of a *forward-filtering backward-sampling* (FFBS) Monte Carlo algorithm[20] that goes in the reverse-time direction of every episode by starting from the censoring instance, and sequentially sampling the latent states conditioned on the (sampled) future trajectory (Godsill et al. (2004)). That is, unlike the *generative process* (described by the routine `GenerateHASMM`(Γ)) which uses the knowledge of Γ to generate sample trajectories by drawing an initial state and then sequentially going forward in time and sampling future states until absorption, the *inferential process* naturally goes the other way around: it exploits informative censoring by starting from the (known) final absorbing state (and censoring time), and sequentially samples a trajectory by traversing backwards in time and conditioning on the future.

We start constructing our forward-filter backward-sampler by first formulating the posterior density of the latent trajectory $\{x_n, s_n\}_{n=1}^{k}$ (from which we sample the G trajectories in the z^{th} iteration of the FFBS-MCEM algorithm as shown in (22)) as follows

$$\begin{aligned} &d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l,\, \hat{\Gamma}^{z-1}) = \\ &d\mathbb{P}(s_k \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) \cdot \prod_{n=1}^{k-1} d\mathbb{P}(x_n, s_n \,|\, \underbrace{\{x_{n'}, s_{n'}\}_{n'=n+1}^{k}}_{\text{Future trajectory}},\, \{y_m, t_m\}_{m=1}^{M},\, T_c), \end{aligned} \tag{24}$$

where the conditioning on the $(z-1)^{th}$ guess of the parameter set, $\hat{\Gamma}^{z-1}$ and the episode index d are suppressed for notational convenience. The formulation in (24) decomposes the posterior density of the latent trajectory $\{x_n, s_n\}_{n=1}^{k}$ into factors in which the likelihood of every state n is conditioned on the future trajectory starting from n (i.e. the states x_{n+1} up to the absorbing states, together with their corresponding sojourn times). The posterior density in (24) can be further decomposed as follows

$$\begin{aligned} &d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) = d\mathbb{P}(s_k \,|\, x_k = l, s_k < T_c) \times \\ &\prod_{n=1}^{k-1} d\mathbb{P}(x_n, s_n \,|\, \{x_{n'}, s_{n'}\}_{n'=n+1}^{k}, s_n < \underbrace{T_c - (s_{n+1} + \ldots + s_k)}_{\text{Elapsed time in the episode}},\, \{y_m, t_m\}_{m=1}^{M}), \end{aligned} \tag{25}$$

which, using the conditional independence properties of the HASMM (see Figure 6), can be simplified as follows

$$\begin{aligned} &d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M},\, x(T_c) = l) = \\ &d\mathbb{P}(s_k \,|\, x_k = l,\, s_k < T_c) \cdot \mathbb{P}(x_1 \,|\, x_2,\, s_1 = T_c - (s_2 + \ldots + s_k),\, \{(y_m, t_m) : t_m \in \mathcal{T}_1\}) \times \\ &\prod_{n=2}^{k-1} d\mathbb{P}(x_n,\, s_n \,|\, x_{n+1},\, s_n < \underbrace{T_c - (s_{n+1} + \ldots + s_k)}_{\text{Elapsed time in the episode}},\, \underbrace{\{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\}}_{\text{Observable variables up to state } n}). \end{aligned} \tag{26}$$

20. The methods used in this Section are also known in the literature as *sequential Monte Carlo* or *particle filtering* methods (Godsill et al. (2004)).

From (26), we can see that for the last state in every episode, i.e. state k, we already know that $x_k = l$, and hence the randomness is only in the last state's sojourn time $s_k = l$. Contrarily, for the first state, we know that conditioned on the sojourn times of the "future states" $(s_2, \ldots, s_k)$, the sojourn time of state x_1 is equal to $T_c - \sum_{n'=2}^{k} s_{n'}$ almost surely, and hence the randomness is only in the initial state realization x_1. Generally, (26) says that a sufficient statistic for the n^{th} state and sojourn time is the future trajectory (starting from state $n+1$) summarized by: the next state, i.e. x_{n+1}, the observable variables up to state n, and the time elapsed in the episode up to state n, i.e. the duration of state n cannot exceed the difference between the censoring time T_c and the sojourn time of the future trajectory that stems from state $n+1$. This is captured by the last factor in (26), which explicitly specifies the likelihood of a joint realization for a state and its sojourn time conditioned on the future trajectory. Using Bayes' rule, we can further represent the last factor in (26) in terms of familiar quantities that are directly derived from the HASMM model parameters as follows

$$\begin{aligned} &d\mathbb{P}(x_n, s_n \,|\, x_{n+1}, s_n < T_c - (s_{n+1} + \ldots + s_K), \{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\}) \\ &\propto \underbrace{\mathbb{P}(x_n \,|\, \{(y_m, t_m) : t_m \in \mathcal{T} / \cup_{n'=n+1}^{k} \mathcal{T}_{n'}\})}_{\text{Forward message}} \times \underbrace{d\mathbb{P}(s_n \,|\, x_n, s_n < T_c - (s_{n+1} + \ldots + s_k))}_{\text{Truncated sojourn time distribution}} \\ &\quad \times \underbrace{\mathbb{P}(x_{n+1} \,|\, x_n, s_n)}_{\text{Transition function}} . \end{aligned} \tag{27}$$

Thus, a sampler for the latent states trajectories can be constructed using the forward messages, the HASMM's transition functions $(g_{ij}(s))_{i,j}$, and the sojourn time distributions. A compact representation for the factors in (27) is given by

(Forward messages) $\mathbb{P}(X_n = j \,|\, \{y_{m'}, t_{m'}\}_{m'=1}^{m}, \hat{\Gamma}^{z-1}) = \alpha_m^{z-1}(j)$, $\forall 1 \le m \le M$, $j \in \mathcal{X}$.

(Transition functions) $\mathbb{P}(X_{n+1} = j \,|\, X_n = i, S_n = s, \hat{\Gamma}^{z-1}) = g_{ij}^{z-1}(s), i, j \in \mathcal{X}$,

(Truncated sojourn times) $d\mathbb{P}(S_n = s \,|\, X_n = j, S_n < \bar{s}) = \dfrac{v_j(s|\hat{\lambda}_j^{z-1}) \cdot \mathbf{1}_{\{s<\bar{s}\}}}{V_j(\bar{s}|\hat{\lambda}_j^{z-1})}, j \in \mathcal{X}$.

Given the representations above, we can write the last factor in (26) in the z^{th} iteration of the EM algorithm as follows

$$d\mathbb{P}(x_n, s_n \,|\, x_{n+1}, s_n < \bar{s}, \{y_m, t_m\}, \hat{\Gamma}^{z-1}) \propto \alpha_m^{z-1}(x_n) \cdot g_{x_n, x_{n+1}}^{z-1}(s_n) \cdot \frac{v_{x_n}(s_n|\hat{\lambda}_{x_n}^{z-1}) \cdot \mathbf{1}_{\{s_n \le \bar{s}\}}}{V_{x_n}(\bar{s}|\hat{\lambda}_{x_n}^{z-1})}. \tag{28}$$

From the factor decomposition in (27), we can see that informative censoring allows us to construct a sampler for the latent state trajectories that operates sequentially in the reverse time direction by sampling from the posterior probability of every state n given the future trajectory of states that starts from state $n+1$. From (28), we note that the posterior density of the latent states conditioned on the future trajectory, from which sequential sampling is conducted, can be explicitly decomposed in terms of the HASMM parameters.

A complete recipe for the forward-filtering backward-sampling procedure for sampling trajectories from the posterior density $d\mathbb{P}(\{x_n, s_n\}_{n=1}^{k} \,|\, \{y_m, t_m\}_{m=1}^{M}, x(T_c) = l, \hat{\Gamma}^{z-1})$ using the decomposition in (27) and the posterior density in (28) is provided as follows:

- **Forward filtering pass:**
  For every episode in $\mathcal{D}$, compute the forward messages $\{\alpha_m^{z-1}(j)\}$ for all time instances $t_m \in \mathcal{T}$ using the current estimate for the parameter set $\hat{\Gamma}^{z-1}$, i.e. invoke the routine `ForwardFilter`$(\hat{\Gamma}^{z-1}, \{y_m, t_m\}_{m=1}^{M}, \epsilon)$.

- **Backward sampling pass:**
  For every episode in $\mathcal{D}$, carry out the following steps:

  1. Set a dummy *placeholder index* as $k^{\#} = 1$ and set $u_{k^{\#}} = l$.
  2. Sample a Bernoulli random variable $B_{k^{\#}} \sim \text{Bernoulli}(\mathbb{P}(k = k^{\#} \,|\, \{u_k, w_k\}_{k=1}^{k^{\#}-1}, l))$.
  3. If $B_{k^{\#}} = 0$ and $k^{\#} > 1$, sample a bi-variate random variable $(u_{k^{\#}}, w_{k^{\#}})$ using the routines `TRSampler` and `BARSampler` as follows

     $$(u_{k^{\#}}, w_{k^{\#}}) \sim \frac{1}{\mathcal{U}} \cdot \alpha_m^{z-1}(u_{k^{\#}}) \cdot g_{u_{k^{\#}}, u_{k^{\#}-1}}^{z-1}(w_{k^{\#}}) \cdot \frac{v_{u_{k^{\#}}}(w_{k^{\#}}|\hat{\lambda}_{u_{k^{\#}}}^{z-1}) \cdot \mathbf{1}_{\{w_{k^{\#}} \le \bar{s}\}}}{V_{u_{k^{\#}}}(\bar{s}|\hat{\lambda}_{u_{k^{\#}}}^{z-1})},$$

     where $\mathcal{U} = \sum_u \int_w \alpha_m^{z-1}(u) \cdot g_{u, u_{k^{\#}-1}}^{z-1}(w) \cdot \frac{v_u(w|\hat{\lambda}_u^{z-1}) \cdot \mathbf{1}_{\{w \le \bar{s}\}}}{V_u(\bar{s}|\hat{\lambda}_u^{z-1})}$, $\bar{s} = T_c - \sum_{n'=1}^{k^{\#}-1} w_{n'}$, and $m = \arg\max_{m'} \{\mathcal{T} : t_{m'} \le \bar{s}\}$.

     If $B_{k^{\#}} = 0$ and $k^{\#} = 1$, then sample $w_{k^{\#}}$ as follows

     $$w_{k^{\#}} \sim \frac{v_{u_{k^{\#}}}(w_{k^{\#}}|\hat{\lambda}_{u_{k^{\#}}}^{z-1}) \cdot \mathbf{1}_{\{w_{k^{\#}} \le T_c\}}}{V_{u_{k^{\#}}}(T_c|\hat{\lambda}_{u_{k^{\#}}}^{z-1})}.$$

  4. If $B_{k^{\#}} = 1$, then set $w_{k^{\#}} = \bar{s}$. If $k^{\#} > 1$, then sample $u_{k^{\#}}$ as follows

     $$u_{k^{\#}} \sim \frac{d\mathbb{P}(\{y_{m'}, t_{m'}\}_{m'=1}^{m} \,|\, u_{k^{\#}}) \cdot g_{u_{k^{\#}}, u_{k^{\#}-1}}(\bar{s}) \cdot v_{u_{k^{\#}}}(\bar{s}|\hat{\lambda}_{u_{k^{\#}}}^{z-1}) \cdot \hat{p}_{u_{k^{\#}}}^{o,z-1}}{\sum_u d\mathbb{P}(\{y_{m'}, t_{m'}\}_{m'=1}^{m} \,|\, u) \cdot g_{u, u_{k^{\#}-1}}(\bar{s}) \cdot v_u(\bar{s}|\hat{\lambda}_u^{z-1}) \cdot \hat{p}_u^{o,z-1}}.$$

  5. If $B_{k^{\#}} = 0$, then increment the placeholder index $k^{\#}$ and go to step 2 and repeat the consequent steps.
  6. If $B_{k^{\#}} = 1$, then set $k = k^{\#}$ and terminate the sampling process. Set the sampled trajectory by swapping the bi-variate sequence $(u_{k^{\#}}, w_{k^{\#}})$ as follows: $(x_n, s_n) = (u_{k^{\#}-n+1}, w_{k^{\#}-n+1}), \forall n \in \{1, \ldots, k^{\#}\}$.

The forward-filtering backward-sampling procedure constitutes of a forward pass in which we compute the forward messages for all the data points in $\mathcal{D}$ using the dynamic programming algorithms presented in Section 3, and a backward pass in which these forward messages are used to sample latent state trajectories. The backward sampling procedure for every episode goes as follows. We start from the censoring time at which we know what

Algorithm 3 Truncated Rejection Sampler

procedure TRSAMPLER(Γ, u, $\bar{s}$)
  Input: A parameter set Γ, a state u and a truncation threshold $\bar{s}$
  Output: A random variable s
  $k \leftarrow 0$
  while $k = 0$ **do**
    $s \sim v_u(s|\lambda_u)$
    Accept s and set $k \leftarrow 1$ if $s < \bar{s}$. Reject s otherwise.
  end while
  return s
end procedure

Algorithm 4 Bivariate Adaptive Rejection Sampler

procedure BARSAMPLER($\{\alpha(j)\}_{j=1}^{N}$, Γ, $u^{'}$, $\bar{w}$)
  Input: A set of N forward messages $\{\alpha(j)\}_{j=1}^{N}$, parameter set Γ, and a state $u^{'}$
  Output: A bivariate conditional random variable $(u, w)|u^{'}$
  $k \leftarrow 0$
  while $k = 0$ **do**
    $u \sim$ Multinomial$(\alpha(1), \ldots, \alpha(N))$
    $w =$ `TRSampler`$(\Gamma, u, \bar{w})$
    $\tilde{u} \sim$ Multinomial$(g_{u1}(w), \ldots, g_{uN}(w))$
    Accept (u, w) and set $k \leftarrow 1$ if $\tilde{u} = u^{'}$. Reject (u, w) otherwise.
  end while
  return (u, w)
end procedure

state has actually materialized, i.e. the absorbing state. Since we do not know the number of states in the state trajectory, we initialize a placeholder index $k^{\#} = 1$ as an index for the absorbing state, and increment it whenever a new state is sampled. We start the sampling procedure as follows. Given the the censoring variables and the observable time series, we sample the sojourn time of the last state (the absorbing state): this is sampled from a truncated sojourn time distribution, with a truncation threshold at T_c, and a point mass at T_c with an assigned measure that is equal to the posterior probability of the absorbing state being the initial state as depicted in Figure 10. This is implemented by first sampling a Bernoulli random variable $B_{k^{\#}}$ with a success probability equal to the posterior probability of the absorbing state being the initial state, and then sampling the truncated sojourn time if $B_{k^{\#}} = 0$ using the simple rejection sample executed by the routine `TRSampler` which is provided in Algorithm 3. Having sampled the last state's sojourn time, we sample the penultimate state and its sojourn time jointly using the routine `BARSampler` (Algorithm 4) as depicted in Figure 11. The routine `BARSampler` uses a sampling algorithm, that we call the *bi-variate adaptive rejection sampler*, which jointly samples the current state and its sojourn time given the next state as follows. First, a state is sampled from a Multinomial distribution with probability masses equal to the forward messages. Next, given the sam-

Algorithm 5 A sampler for latent state trajectories

procedure BACKWARDSAMPLING($\Gamma, \{\{\alpha_m^o(j)\}_{m,j}\}, \{y_m, t_m\}_{m=1}^{M}, x(T_c) = l$)
  Input: Parameter Γ, forward messages, observables, and censoring information
  Output: A sampled latent state trajectory $\{x_n, s_n\}_{n=1}^{k}$
  $k^{\#} \leftarrow 1$, $u_{k^{\#}} \leftarrow l$, $B_{k^{\#}} \sim$ Bernoulli$(\mathbb{P}(k = k^{\#}|\{y_m, t_m\}_{m=1}^{M}, x(T_c) = l))$
  if $B_{k^{\#}} = 0$ **then**
    $w_{k^{\#}} =$ `TRSampler`$(\Gamma, u_{k^{\#}}, T_c)$
    $k^{\#} \leftarrow k^{\#} + 1$
  else
    $w_{k^{\#}} = T_c$, $k = 1$, $\{x_1, s_1\} \leftarrow \{u_{k^{\#}}, w_{k^{\#}}\}$
    Terminate `BackwardSampling`.
  end if
  while $k^{\#} > 0$ **do**
    $B_{k^{\#}} \sim$ Bernoulli$(\mathbb{P}(k = k^{\#}|\{u_k, w_k\}_{k=1}^{k^{\#}-1}, \{y_m, t_m\}_{m=1}^{M}))$
    $\bar{s} = T_c - \sum_{n'=1}^{k^{\#}-1} w_{k^{\#}}$
    if $B_{k^{\#}} = 0$ **then**
      $(u_{k^{\#}}, w_{k^{\#}}) \leftarrow$ `BARSampler`$(\{\alpha_m^o(j)\}_j, \Gamma, u_{k^{\#}-1}, \bar{s})$
      $k^{\#} \leftarrow k^{\#} + 1$
    else
      Sample the initial state $u_{k^{\#}}$, set $w_{k^{\#}} \leftarrow \bar{s}$
      $\{x_n, s_n\} = \{u_{k^{\#}-n+1}, w_{k^{\#}-n+1}\}$, $\forall n \in \{1, \ldots, k^{\#}\}$
      $k^{\#} \leftarrow -1$
    end if
  end while
  return $\{x_n, s_n\}_{n=1}^{k}$
end procedure

pled state, we sample a sojourn time from the truncated sojourn time distribution. Finally, given the sampled state and the sampled sojourn time, we sample a dummy state from a Multinomial whose masses are equal to the transition functions, and we accept the sample only if the sampled dummy state is equal to the next state. It can be easily proven that `BARSampler` generates samples that are equal in distribution to the true state trajectory.

The backward-sampling procedure operates sequentially by invoking the `BARSampler` to generate new state and sojourn times samples conditional on the previously sampled (future) states. The process terminates whenever $B_{k^{\#}} = 1$, i.e. a state is sampled as an "initial state". The routine `BackwardSampling` (Algorithm 5) implements the overall backward-sampling procedure for every episode in $\mathcal{D}$. **The computational complexity of the `BackwardSampling` routine is dominated by the computation of the GP likelihood (Step 4 in the sampling procedure described above), which is cubic in the number of observations ($\mathcal{O}(W^3)$). The computations in the `BackwardSampling` procedure scales only linearly with the number of states N.**

Note that, unlike the slowly mixing MCMC methods, the backward-sampling algorithm can generate the latent state trajectory in an efficient manner, i.e. the run-time of the

Algorithm 6 Forward-filtering Backward-sampling Monte Carlo EM Algorithm

procedure `FFBS-MCEM`($\mathcal{D}$, G, ϵ)
  Input: A dataset $\mathcal{D}$, number of Monte Carlo samples G, and a precision level ϵ
  Output: An estimate $\hat{\Gamma}$ for the HASMM parameters
  Set an initial value $\hat{\Gamma}^o$ for the HASMM parameters
  $\{\alpha_m^{d,o}\}_{m=1}^{M_d} = \texttt{ForwardFilter}(\hat{\Gamma}^o, \{y_m^d, t_m^d\}_{m=1}^{M_d}, \epsilon), \forall 1 \leq d \leq D$ ▷ Forward pass
  for $d = 1$ to D **do** ▷ Backward pass: sample G latent state trajectories
    for $g = 1$ to G **do**
      $\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} = \texttt{BackwardSampling}(\hat{\Gamma}^o, \{y_m^d, t_m^d\}_{m=1}^{M_d}, x^d(T_c^d) = l^d)$
    end for
  end for
  $z \leftarrow 1$
  $E \leftarrow \epsilon + 1$
  while $E > \epsilon$ **do**
    $I_{d,g}^{z-1} \leftarrow d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} | \hat{\Gamma}^{z-1}) / d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} | \hat{\Gamma}^o)$ ▷ Importance weights
    $\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} | \Gamma)) \cdot \frac{I_{d,g}^{z-1}}{G}$ ▷ E-step
    $\hat{\Gamma}^z = \arg\max_\Gamma \hat{U}_G(\Gamma; \hat{\Gamma}^{z-1})$ ▷ M-step
    $z \leftarrow z + 1$
  end while
  return $\hat{\Gamma} = \hat{\Gamma}^z$
end procedure

backward-sampling algorithm is stochastically dominated by a geometrically-distributed random variable with a success probability that, other than in a pathological HASMM parameter settings, would not be close to zero. Moreover, since `BackwardSampling` generates independent samples, no wasteful burn-in sampling iterations are involved in the FFBS-MCEM operation. We provide a pseudocode for the overall operation of the `FFBS-MCEM` algorithm in Algorithm 6. We omit the standard EM operations for the sake of brevity. The details of the M-step is provided in Appendix D.

In Algorithm 6, we avoid the need for running the routine `BackwardSampling` in every iteration of the EM algorithm by re-using the sampled trajectories based on the initial parameter guess $\hat{\Gamma}^o$ through the usage of importance weights in the E-step. That is, in the z^{th} iteration of the EM algorithm, we implement the E-step as follows (Booth and Hobert (1999))

$$\hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} | \Gamma)) \cdot \underbrace{\frac{d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} | \hat{\Gamma}^{z-1})}{d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}} | \hat{\Gamma}^o)}}_{\text{Importance weights}}.$$

This implementation for the E-step offers a tremendous advantage in the computational cost of `FFBS-MCEM`. By using importance weights, we need to compute the forward messages and sample the latent state trajectories only once, and then reuse the sampled trajectories in all the subsequent EM iterations.

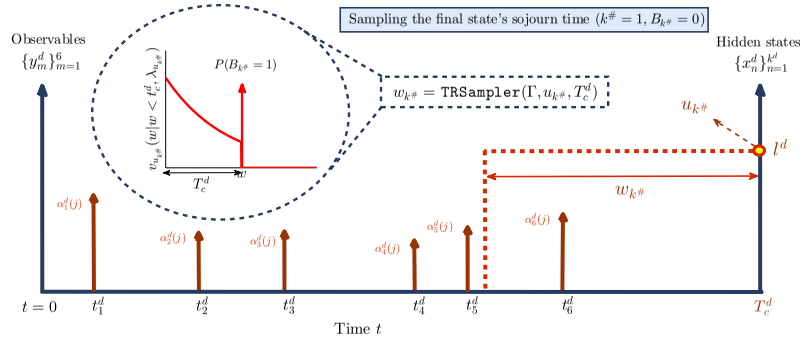


Figure 10: Depiction of the backward sampling pass for the last state of an episode d.

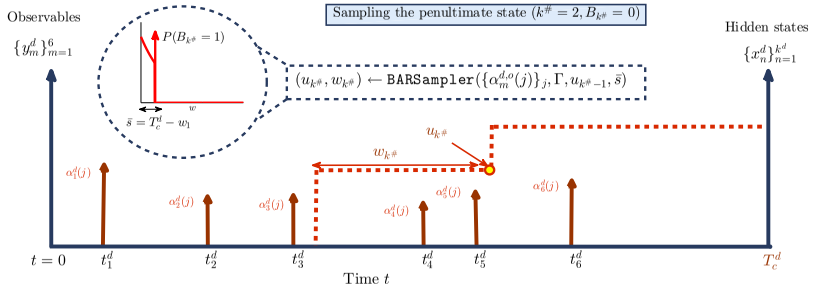


Figure 11: Depiction of the backward sampling pass for the penultimate state after having sampled the last state as depicted in the Figure above.

5. Experiments: Intensive Care Unit Prognostication

We investigate the utility of the HASMM in the setting of ICU prognostication; we use the HASMM as a model for the physiology of critically ill patients in regular hospital wards who are monitored for various vital signs and lab tests. Through the HASMM, we construct a risk score (based on the analysis in Section 3.3) that assesses the risk of clinical deterioration for the monitored patients, which allows for timely ICU admission whenever clinical decompensation is detected. Risk scoring in hospital wards and ICU admission management is a pressing problem with a huge social and clinical impact: qualitative medical studies have suggested that up to 50% of cardiac arrests on general wards could be prevented by earlier transfer to the ICU (Hershey and Fisher (1982)). Since over 200,000 in-hospital cardiac arrests occur in the U.S. each year with a mortality rate of 75% (Merchant et al. (2011)), improved patient monitoring and vigilant care in wards enabled by the HASMM would translate to a large number of lives saved yearly.

5.1 Evidence of the Clinical Utility of Early ICU Admission

Throughout this Section, we will evaluate the clinical utility of our model by investigating both the *accuracy* and *timeliness* of the real-time risk scores that the model computes. This approach has been the standard approach for evaluating the clinical utility of risk scores in retrospective clinical cohort studies that deal with critical care data (Pirracchio et al. (2015); Rothman et al. (2013). More accuracy and timeliness translates to a necessarily improved clinical outcomes; this fact has been confirmed by a large number of medical studies (Cardoso et al. (2011); Johnson et al. (2013); Hershey and Fisher (1982)). For instance, in (Cardoso et al. (2011)), it was shown that each hour of waiting in the ward was independently associated with a 1.5% increased risk of mortality in the ICU.

We stress that knowing the exact magnitude of the improvement in clinical utility (in terms of the reduction in the incidence rates of adverse outcomes) upon using our model is not possible since all available datasets are observational in nature. That is, it is impossible to answer the question of "what would have happened to the patient in the ICU had she been admitted earlier?". Estimation of such counterfactual outcomes is also not viable due to the highly imbalanced nature of the data and the wide variety of possible adverse outcomes in the ICU, some of which are not available in our dataset. Evaluating the clinical utility in terms of the reduction in the incidence rates of adverse outcomes is only possible through an actual clinical trial. Hence, after consulting with our medical collaborators and following the clinical literature on observational studies, we rely on the accuracy and timeliness metrics as proxies for the clinical utility.

5.2 Data

5.2.1 The Patients' Cohort

Experiments were conducted on a heterogeneous cohort of 6,094 episodes for patients who were hospitalized in Ronald Reagan UCLA medical center during the period between March 3^{rd}, 2013 to March 29^{th}, 2016. The patients' population is heterogeneous: we considered admissions to all the floors and units in the medical center, those include the acute care pediatrics unit, cardiac observation unit, cardiothoracic unit, hematology and stem cell transplant unit and the liver transplant service. Patients admitted to those floors (or wards) are post-operative or pre-operative critically ill patients who are vulnerable to adverse clinical outcomes that may require an impending ICU transfer. The cohort comprised patients with a wide variety of ICD-9 codes and medical conditions, including leukemia, hypertension, septicemia, sepsis, abdomen and pelvis, pneumonia, and renal failure. Table 2 shows the distribution of the most common ICD-9 codes in the patient cohort together with the corresponding medical conditions . The notable heterogeneity of the cohort suggests that the results presented in this Section are generalizable to different cohorts extracted from different hospitals. Every patient in the cohort is associated with a set of 21 (temporal) physiological streams comprising a set of vital signs and lab tests that are listed in Table 2. The physiological measurements are gathered over time during the patient's stay in the ward, and they manifest -in a subtle fashion- the patient's clinical state. The physiological measurements are collected over irregularly spaced time intervals (usually ranging from 1

Table 2: Characteristics of the patient cohort under study

Physiological data	
Vital signs	**Lab tests**
Diastolic blood pressure Eye opening Glasgow coma scale score Heart rate Respiratory rate Temperature O_2 Device Assistance O_2 Saturation Best motor response Best verbal response Systolic blood pressure	Chloride Glucose Urea Nitrogen White blood cell count Creatinine Hemoglobin Platelet Count Potassium Saturation Sodium Total CO_2

ICD-9 codes	
(786.05)	Shortness of Breath
(401.9)	Hypertension
(38.9)	Septicemia
(995.91)	Sepsis
(789)	Abdomen and pelvis
(780.6)	Fever
(486)	Pneumonia
(584.9)	Renal failure
(599)	Urethra and urinary attack
(780.97)	Altered mental status
(285.9)	Anemia
(786.5)	Chest pain
(585)	Chronic renal failure
(780.79)	Malaise and fatigue
(578)	Gastrointestinal hemorrhage
(428)	Heart failure
(427.31)	Atrial fibrillation
(787.01)	Nausea

Baseline Patient Characteristics (with 95% CI)
• Gender distribution (Male percentage)
(Training: 50.31% ± 1.4% - Testing: 51.16% ± 2.92%)
• Transfers from other hospitals
(Training: 11.88% ± 0.94% - Testing: 11.08% ± 1.95%)
• Average age
(Training: 58.9 ± 0.55 years - Testing: 59.37 ± 1.11 years)
• Patients with chemotherapy
(Training: 0.688% ± 0.272% - Testing: 1.558% ± 0.9%)
• Patients with stem cell transplants
(Training: 0.121% ± 0.8% - Testing: 0.008% ± 0.004%)

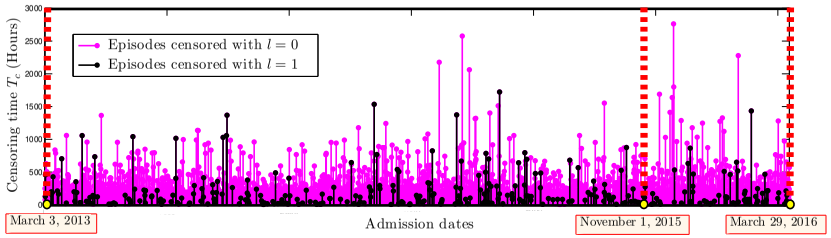


Figure 12: Visualization for the episodes' censoring information.

to 4 hours); for each physiological time series, we have access to the times at which each value was gathered.

5.2.2 Inclusion and Exclusion Criteria

In all the experiments hereafter, we split the patient cohort into a training set and a testing set. In the training set, we included a total of 4,939 patients admitted to the medical center in the period between March 3^{rd}, 2013 to November 1^{st}, 2015; the testing set comprises 1,155 patients admitted in the period between November 1^{st}, 2015 to March 29^{th}, 2016. This split of the data allows us to assess the performance under the realistic scenario when a certain algorithm learns from the data available up to a certain date, and then is used to assess the risk for patients admitted in future dates. In Table 2, we show statistics for the patients' baseline static features (e.g. gender, age, etc) in both the training and testing sets; as we can see, the characteristics of the patients admitted in the period (March 2013 - November 2015) has not significantly changed from those admitted in the period (November 2015 - March 2016). We have verified this fact using a two-sample t-test through which we compared the expected values of the baseline co-variates in both the training and testing sets. This means that the hospital's management policy with respect to the patients' acceptance and triaging has not significantly changed across the two time periods, and hence whatever is learned from the training data can be sensibly applied to the testing data. **We have excluded all patients who underwent a preplanned ICU admission from the dataset since those patients did not actually experience clinical deterioration, but were transferred routinely to the ICU after a surgery.**

5.2.3 Informative Censoring

All the patient episodes in the cohort were informatively censored. That is, for every patient in the cohort, we know the following information:

• **The censoring time (T_c):** the length of stay of each patient in the ward is recorded in the dataset, and hence we have access to the HASMM's censoring time variable T_c. The average hospitalization time (or censoring time) in the cohort is 157 hours and 34 minutes (6.5 days). The patient episodes' censoring times ranged from 4 hours to 2,672 hours.

• **The absorbing clinical state (l):** with the help of experts from the division of pulmonary and critical care medicine at Ronald Reagan UCLA medical center, we set the value of the variable l (absorbing state) for every patient's episode based on the clinicians' interventions as reported in the dataset. That is, as advised by our medical collaborators, we assigned the label $l = 1$ to every patient who was admitted to the ICU and underwent an intervention in the ICU (e.g. ventilator, drug, etc), or was reported to exhibit a cardiac or respiratory arrest (before or after the ICU transfer). According to the medical experts, those patients have experienced "clinical deterioration" as their absorbing state, and would have benefited from an earlier admission to the ICU. We assigned the label $l = 0$ to all patients who were discharged home after the clinician's in charge realized they were clinically stable. Since the readmission rate at the UCLA medical center is quite low, our medical collaborators believe that the labels $l = 1$ and $l = 0$ represent an accurate representation for the patients' true absorbing clinical states upon censoring.

Patient episodes with the absorbing state $l = 0$ had an average censoring time of 155 hours, whereas those with $l = 1$ had an average censoring time of 204 hours. The percentage of episodes with an absorbing state $l = 1$ was $4.98\% \pm 0.64\%$ in the training period (March 2013 - November 2015), and was $5.19\% \pm 1.44\%$ in the testing period (November 2015 - March 2016). A two-sample t-test reveals that the censoring information (distributions of T_c and l) has not significantly changed from the training to testing periods, which suggests that the HASMM learned from the training data can be sensibly applied to the testing data. Figure 12 visualizes the informative censoring information over the time period between March 2013 and March 2016. Every patient episode, starting at a certain admission date, is represented by its censoring time (hospitalization time); light colored episodes are ones that were absorbed in the clinical stability state ($l = 0$), whereas dark colored ones were absorbed in the clinical deterioration state ($l = 1$).

5.3 Baseline Algorithms

We compare our model with other baseline early warning methods. The comparisons involve both state-of-the-art clinical risk scores that are currently used in various healthcare facilities around the world, in addition to benchmark machine learning algorithms. The details of the baselines are provided in the following subsections.

5.3.1 State-of-the-art Clinical Risk Scores

We have conducted comparisons with the most prominent clinical risk scores currently deployed in major healthcare facilities. We list the clinical risk scores involved in our comparisons hereunder.

(i) **Modified Early Warning System (MEWS)**: a risk scoring scheme used currently by many healthcare facilities and rapid response teams to quickly assess the severity of illness of a hospitalized patient (Morgan et al. (1997)). The score ranges from 0 to 3 and is based on the following cardinal vital signs: systolic blood pressure, respiratory rate, SaO_2, temperature, and heart rate.

(ii) **Sequential Organ Failure Assessment (SOFA)**: a risk score (ranging from 1 to 4) that is used to determine the extent of a hospitalized patient's respiratory, cardiovascular, hepatic, coagulation, renal and neurological organ function in the ICU (Vincent et al. (1996)).

(iii) **Acute Physiology and Chronic Health Evaluation (APACHE II)**: a risk scoring system (an integer score from 0 to 71) for predicting mortality of patients in the ICU (Knaus et al. (1991)). The score is based on 12 physiological measurements, including creatinine, white blood cell count, and glasgow coma scale.

(iv) **Rothman Index**: a regression-based data-driven risk score that utilizes physiological data to predict mortality, 30-days readmission, and ICU admissions for patients in regular wards (Rothman et al. (2013)). The Rothman index is the state-of-the-art risk score for regular ward patients and is currently used in more than 70 hospitals in the US, including the Houston Methodist hospital in Texas and the Yale-New Haven hospital in Connecticut (Landro (2015)). At the time of conducting these experiments, the Rothman index was also deployed in the Ronald Reagan UCLA medical center.

We implemented the MEWS, SOFA, APACHE II and Rothman scores according to the specifications in (Vincent et al. (1996); Knaus et al. (1991); Rothman et al. (2013)). Note that while the SOFA and APACHE II scores are usually deployed for patients in the ICU, both scores have been recently shown to provide a prognostic utility for predicting clinical deterioration for patients in regular wards (Yu et al. (2014)), and hence we consider both scores in our comparisons. All the features used by these scores were also fed to the machine learning baselines.

5.3.2 Machine Learning Algorithms

In order to demonstrate the modeling gain of HASMMs, we make comparisons with 12 competing machine learning algorithms. We list all the baseline models hereunder.

- **Random forest.**
- **Logistic regression.**
- **XGBoost.**
- **AdaBoost.**
- **Bagging.**
- **Least absolute shrinkage and selection operator (LASSO).**
- **Deep Neural Networks (DNN) trained with BFGS.**
- **DNN trained with ADAM.**
- **Recurrent Neural Networks (RNN) trained with BFGS.**
- **RNN trained with ADAM.**

- **Hidden Markov Models (HMM) with Gaussian emissions.**
- **Multi-task Gaussian process (MTGP).**

Recently, notable works have built on ideas from deep learning and deep hierarchical models to construct survival predictors that learn from time-to-event data: examples of such models are those in (Katzman et al. (2016)) and (Ranganath et al. (2016)). Unfortunately, these models do not apply directly to our setting for two reasons. First, the models therein are not well-suited for handling irregularly sampled follow-ups and computing survival curves in a dynamic fashion. Second, and more importantly, our main focus is to predict whether or not a patient will exhibit clinical deterioration in the future, and not estimating survival curves with respect to a single endpoint event. Hence, our problem is technically equivalent to survival analysis with two "competing risks" (Prentice et al. (1978)), with the competing risks being *ICU admission* and *hospital discharge*, and thus the problem cannot be directly cast to the standard survival analysis setting tackled in (Katzman et al. (2016)) and (Ranganath et al. (2016)).

In order to ensure that the censoring information is properly utilized by all the discriminative predictors (Random forest, Logistic regression, XGBoost, AdaBoost, Bagging, LASSO, MTGP, and DNN), we train every predictor by constructing a training dataset that comprises the physiological data gathered within a temporal window before the censoring event (ICU admission or patient discharge), and using the censoring information (i.e. the variable l) as the labels. The size of this window is a hyper-parameter that is tuned separately for every predictor. For the testing data, the predictors are applied sequentially to a sliding window of every patient's episode, and the predictor's output is considered as the patient's real-time risk score. We used Python's `Sklearn` library (Pedregosa et al. (2011)) for training the Random forest, Logistic regression, XGBoost, AdaBoost, Bagging, LASSO and DNN predictors, and the `GPy library` (group (2012)) for training the MTGP predictor. The RNN models were implemented in `TensorFlow` (Abadi et al. (2016)).

Although RNNs are not clinically interpretable, they have been frequently applied to the problem of clinical time series prediction, and the recent work in (Che et al. (2016)) have considered RNNs to predict mortality in the ICU using the MIMIC dataset (Saeed et al. (2002)). We have trained an RNN with 5 hidden layers, and 10 neurons with each layer, using both the *Broyden-Fletcher-Goldfarb-Shanno* (BFGS) algorithm[21], where gradients are computed using the *Backpropagation Through Time* algorithm (Werbos (1990)). We have also trained an RNN model using the ADAM optimizer (Kingma and Ba (2014)). All the training time series were temporally aligned via the endpoint censoring information, and training was accomplished via 1000 iterations of the gradient descent algorithm. A top layer with a squashing sigmoid function was used to map the RNN hidden states to a risk score between 0 and 1 at each point in time. The DNNs are implemented as multi-layer perceptrons, the hyper-parameters of which (number of layers and hidden units) are optimized using grid search.

21. We have also tried the Levenberg-Marquardt algorithm, but the network learned by BFGS offered a significantly better performance.

We used the Baum-Welch algorithm for learning the HMM (Murphy et al. (2001)); the informative censoring information was incorporated by including two absorbing states for clinical stability ($l = 0$) and deterioration ($l = 1$), and informing the forward-backward algorithm with the labeled states at the end of every episode. We tried many initializations for the HMM parameters and picked the initialization that led to the maximum likelihood for the training dataset. The complete data log likelihood after 100 EM iterations was -1.25$\times$ 10^7. In real-time, a patient's risk score at every point of time is computed by first applying forward filtering to obtain the posterior probability of the patient's states, and then averaging over the distribution of the absorbing states. Using the Bayesian Information Criterion, we selected an HMM model with 4 latent states.

For the multi-task Gaussian process, we used the free-form parametrization (intrinsic coregionalization model) in (Bonilla et al. (2007)), and used the gradient method to learn the parameters of two Gaussian process models: one for patients with $l = 0$, and one for patients with $l = 1$. The risk score for a patient's risk score is computed as the test statistic of a sequential hypothesis test that is based on the two learned Gaussian process models. This differs from the static simulation setting in (Ghassemi et al. (2015)) were predictions are issued in a one-shot fashion using only the data obtained within 24 hours after a patient's admission.

We used the correlated feature selection algorithm to select the physiological stream for every predictor (Yu and Liu (2003)). **To ensure a fair comparison, we did not include the static (background) co-variates in any predictor, including the HASMM, since they are not used by the clinical risk scores.**

5.4 Results

5.4.1 Performance metrics

In order to assess the performance of every algorithm, we compute each algorithm's risk score $R(t)$ at every point of time in every patient's episode. We only use the patient episodes in the testing set for performance evaluation. The risk score that is based on an HASMM is evaluated as discussed in Section 3.3. We emulate the ICU admission decisions by setting a threshold on the risk score $R(t)$ above which a patient is identified as "clinically deteriorating". The accuracy of such decisions are assessed via the following performance metrics: true positive rate (TPR), positive predictive value (PPV) and timeliness. These performance metrics are formally defined as follows:

$$\text{TPR} = \frac{\#\,\text{patients with } l = 1 \text{ and } R(t) \text{ exceeding threshold for some } t < Tc}{\#\,\text{patients with } l = 1},$$

$$\text{PPV} = \frac{\#\,\text{patients with } l = 1 \text{ and } R(t) \text{ exceeding threshold for some } t < Tc}{\#\,\text{patients with } R(t) \text{ exceeding threshold for some } t < Tc},$$

and

$$\text{Timeliness} = \mathbb{E}\left[\text{Time at which } R(t) \text{ exceeds threshold} - T_c \mid R(t) \text{ exceeds threshold}, l = 1\right].$$

The three performance metrics described above evaluate the different risk scoring algorithms in terms of their detection power, false alarm rate, and timeliness in detecting clinical deterioration. We sweep the threshold value of every risk scoring algorithm and report the AUC of the TPR vs. PPV ROC curve. All results reported hereafter are statistically significant (p-value < 0.001).

The usage of precision and recall (TPR and PPV) instead of the conventional (TPR and FPR) metrics is driven by the following motives. Since we are using our model as an alarm system, our algorithm only picks patients who are believed to be deterioration (patients with label 1), and we are not identifying stable patients (i.e. there is no well-defined "true negative" count). The performance of an algorithm in this particular "information retrieval" setting is more sensibly assessed via precision and recall. That is, we are trying to identify as many deteriorating patients as possible (TPR) and avoid overwhelming the ward staff with many false alarms (PPV). The "true negative" count does not play an important role in our setting. We also note that the ward's patient cohort has a significant class imbalance (the ICU admission rate is around 5%). Hence, we are typically trying to identify a small number of deteriorating patients in a large pool of stable patients. In such an unbalanced cohort, it is significantly more difficult to achieve a good PPV (PPV = TP/(TP + FP)) than a low false positive rate (FPR = FP/(FP + TN)) for a fixed TPR, since most patients are already clinically stable and therefore the false positive and true negative counts (which are counted in the stable population) will naturally be significantly larger than the true positive counts. Hence, the FPR rates (and consequently the AUC values) may look deceptively high, but they are not truly reflective of the "usefulness" of the algorithm. This is because one can still have numerous false alarms, the quantification of which is distorted by the large true negative rates that results mainly because of the fact most patients are stable. Due to reasons above, the area under the TPR vs. PPV curve has been recently identified by the critical care community as being a more sensible measure of accuracy (Romero-Brufau et al. (2015)).

While the traditional AUCROC metric can be interpreted as the probability of miss-ranking two instances with positive and negative classes, the area under the TPR vs. PPV curve can be interpreted as a measure of how well an algorithm can identify the positive classes in a pool of instances with a negative class (Davis and Goadrich (2006)). **Note that random guessing yields an area under TPR vs. FPR curve of 0.5; in the case of the TPR vs. PPV curve, and given the definitions above, random guessing yields and AUC that is equal to the fraction of instances with a positive class. That is, in our dataset, the area under the TPR vs. PPV curve for random guessing is as small as 0.05.**

5.4.2 Learning the HASMM

We applied the FFBS-MCEM algorithm to the training episodes in order to estimate the parameter set Γ. Based on the Bayesian information criterion, we have selected a model with 4 clinical states, i.e. $\mathcal{X} = \{1, 2, 3, 4\}$. State 1 is the clinical stability state, whereas state 4 is the clinical deterioration state. We ran 100 MCEM iterations and used $\hat{\Gamma}^{100}$ as the

estimate for Γ. The parameter set Γ was initialized randomly using uniform distributions that cover each parameter's admissible bounds.

We discretized the time domain into steps of 1 hour while computing the elements of the look-up table holding the values of the tensor $\tilde{\mathbf{P}}$. With a granular 1-hour discretization of the time horizon, the Gaussian covariance matrix was found to be ill-conditioned for many patient episodes. To ensure the numerical stability of the computations involving the Gaussian process likelihood functions, we used the Moore-Penrose pseudo-inverse for the covariance matrix instead of direct matrix inversion. The function `TransitionLookUp` was invoked once before running the MCEM iterations, and its run time was 2 minutes and 15 seconds on a dual-core 3 GHz machine. The function `ForwardFilter` was invoked 150,852 times (all data points in all patients' episodes in both the training and testing sets), and its overall run time was 3 hours and 50 minutes (on a dual-core 3 GHz machine). The run time for every risk score update for a single patient is less than 1 second, which implies that the algorithm can efficiently prompt quick risk assessments if implemented on a machine with a reasonable computational power.

From the learned HASMM, we were able to extract the following "medical concept" out of the training data. The patients' clinical state space $\mathcal{X} = \{1, 2, 3, 4\}$ comprises the following 4 states:

- **State 1: clinical stability.**
- **State 2: type-1 critical state.**
- **State 3: type-2 critical state.**
- **State 4: clinical deterioration.**

As implied by the model, states 1 and 4 are absorbing states: once the patient is believed to be in state 1, the clinicians should release her from care, whereas exhibiting clinical state 4 should be treated with an admission to the ICU. States 2 and 3 are critical states that require the patient to stay under vigilant care in the ward. The two states are different ways to manifest "criticality". We characterize the properties of the four clinical states in the rest of this subsection.

Figures 13-16 depict the different characteristics of the four clinical states. In Figure 13, we plot a bipartite correlation graph that shows the correlations among the relevant physiological streams in the different clinical states. These graphs were constructed by computing the *Pearson correlation coefficient* $\sigma_{Y^l Y^v} = \frac{\text{COV}(Y^l, Y^v)}{\sigma_{Y^l} \cdot \sigma_{Y^v}}$ using the entries of the multi-task Gaussian process covariance matrix $\mathbf{\Sigma}$. An edge is connected between every two features for whom the Pearson correlation coefficient exceeds 0.1, i.e. $\sigma_{Y^l Y^v} > 0.1$. As we can see, different physiological variables become less or more correlated in the different clinical states. For instance, only the clinically stable patients experience significant correlations between their urea Nitrogen and the diastolic blood pressure; the Pearson coefficient between those variables becomes insignificant in the other states. Clinicians can use this piece of information, extracted solely from the data, to construct simple tests for clinical stability by computing

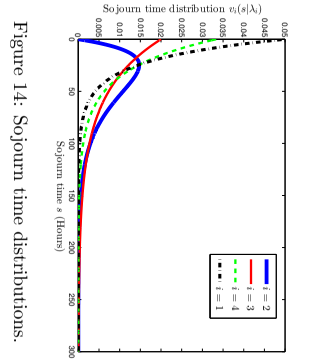


Figure 14: Sojourn time distributions.

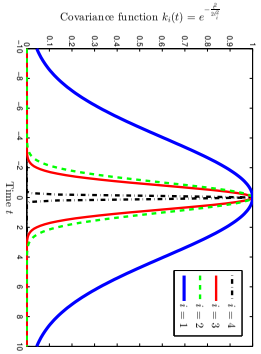


Figure 15: Covariance functions.

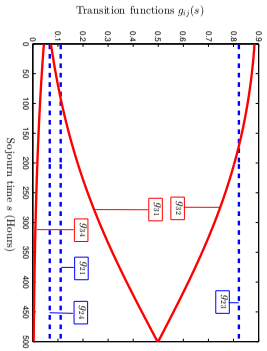


Figure 16: Transition functions.

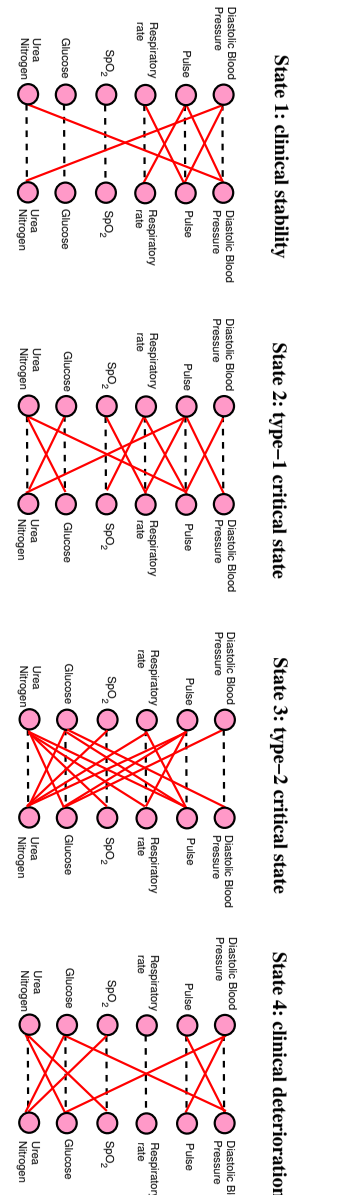


Figure 13: Correlations between the patients' physiological streams in the different clinical states.

the correlations between blood pressure and urea Nitrogen for a hospitalized patient before deciding to discharge her. Generally speaking, we observe that the critical, transient states display more correlations among the physiological streams than the clinical stability and deterioration states. In particular, the type-2 critical state has most of the physiological streams being strongly correlated. We speculate that the reason behind these strong correlations is that some kinds of interventions (e.g. drugs, mechanical pumps, ventilators, etc) applied to hospitalized patients affect all the physiological streams simultaneously; and hence we believe that type-1 and type-2 critical state patients are hospitalized patients with and without clinical interventions. We will examine this claim when we retrieve information about interventions and the time they were applied from the Ronald Reagan medical center; such information was not available at the time of conducting these experiments.

Figure 14 shows the sojourn time distributions for the four states. Recall that the "sojourn time" of an absorbing state (state 1 or 4) is defined as the time between entering the state and the censoring time; such a time interval corresponds to the clinicians' policy with respect to patient discharge and ICU admission. That is, the sojourn time of an absorbing state is not a natural physiological quantity, but it rather reflects the speed with which patients are released from care or receive leveraged level of care. The sojourn time distribution for state i is an exponential distribution if the shape parameter $\lambda_{i,s} = 1$. The sojourn time distributions for states 2 and 3 significantly deviate from an exponential distribution of an ordinary, memoryless Markov model, which supports our assumption of semi-Markovianity. As we can see in Figure 14, the sojourn time distribution is not concentrated around 0 and hence is radically different from an exponential distribution (the estimated shape parameter is $\lambda_{2,s} = 2.25$). State 2 is the state with the largest first moment for the sojourn time distribution: this means that most patients in the ward exhibit this state and hence it is the most relevant for predictions. Figure 14 clearly shows that this state is not memoryless. State 3 exhibits a sojourn time distribution that is concentrated around 0; however, its shape parameter is $\lambda_{2,s} = 0.55$, and hence cannot be adequately modeled by a memoryless process.

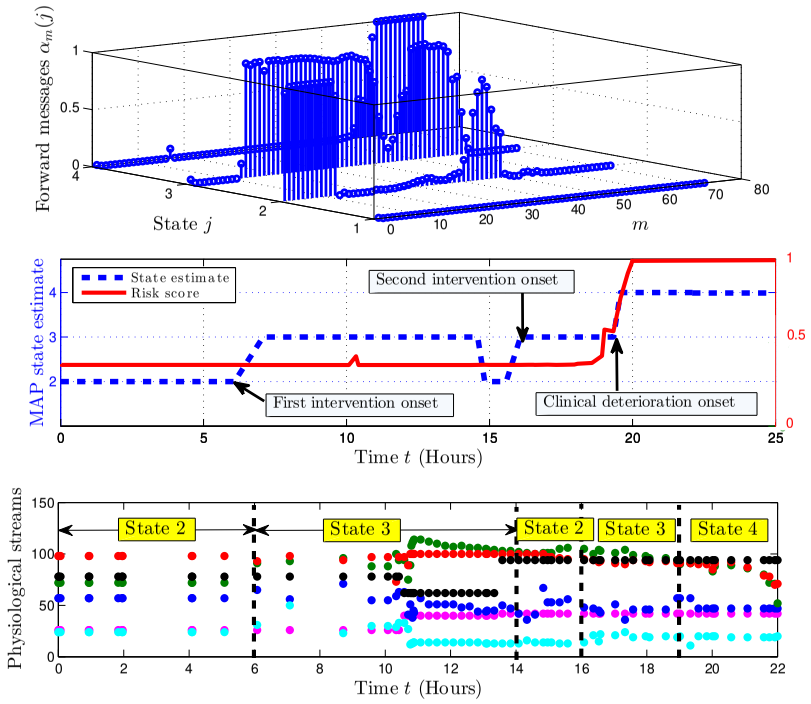


Figure 17: Depiction for the episode of a clinically deteriorating patient. (The physiological streams are color coded as follows: Diastolic blood pressure is in green, systolic blood pressure is in red, blood urea nitrogen is in cyan, and heart rate is in black, respiratory rate is in purple.)

Figure 15 displays the covariance function $k_i(t, t')$ for the 4 clinical states; the state-specific covariance function quantifies the physiological streams' temporal correlations in a particular clinical conditions. Knowing such correlation patterns are useful for deciding the frequency with which nurses and clinicians should collect physiological measurements over time for different patients in different clinical conditions (Alaa and van der Schaar (2016)). We observed that, as one would expect, the temporal correlations increase when the patient becomes more stable; the temporal correlation is greatest in state 1 and smallest in state 4. This means that one would expect deteriorating patients to experience more physiological fluctuations over time. We also note that physiological stream for which the constant mean function differed significant among the clinical state was the urea Nitrogen. The level of urea Nitrogen increases significantly when the patient is in a more risky state; the average blood urea nitrogen is 11.7 milligrams per deciliter (mg/dL) in state 1, 23.8 mg/dL in state 2, 41.1 mg/dL in state 3 and 64.9 mg/dL in state 4. This is consistent with medical domain knowledge and recent discoveries in the area of critical care medicine; in (Beier et al. (2011)), it was shown that there is a substantial evidence that that elevated urea Nitrogen can be associated with all cause mortality in a heterogeneous critically ill population.

Figure 16 depicts the transition functions g_{ij} out of the transient states 2 and 3 as a function of the sojourn time in those states. We note that the transition probabilities are almost a constant function of sojourn time for patients in state 2 ($\beta_{2j} \approx 0$), whereas the duration-dependence is more significant ($\beta_{3j} > 0$); as the sojourn time in state 3 increases, the transition probabilities become more biased towards state 1. This reinforces our hypothesis that state 3 corresponds to patients for whom interventions were applied. That is, as time passes for a patient in state 3 after receiving an intervention, her chances for recovery (transiting to state 1) increases.

Now we illustrate the real-time operation of the inference algorithm as it computes risk score over time by focusing on an episode of a particular patient who was hospitalized for 1 day and then admitted to the ICU. As shown in Figure 17 (top), the inference algorithm

computes the forward messages whenever new physiological measurements become available. Using the forward messages, the algorithm can display the maximum a posteriori (MAP) state estimates to the clinicians over time. As we can see in Figure 17 (middle), the patient under consideration was in clinical state 2 (type-1 critical state) at the time of admission to the ward. After 6 hours, the patient switched to state 3 (type-2 critical state), probably due to a clinical intervention. After around 9 hours, the patient switched back to the type-1 critical state for a brief 2-hour period, before switching to the type-2 critical state (probably due to a second intervention). Our algorithm was able to detect clinical deterioration (state 4) conclusively (through both the MAP state estimate and the risk score) more than 6 hours before the clinicians actually sent the patient to the ICU. Had the clinicians used the algorithm for monitoring that patient, they would have been able to send the patient to the ICU 6 hours early, allowing for a potentially much more efficient therapeutic intervention in intensive care. In Figure 17 (bottom), we plot the patient's physiological stream and tag the different time intervals with the corresponding clinical state estimates. The clinicians can rely on these clinically interpretable tags to describe the patient's states at each point of time rather than using a high-dimensional, and potentially inexpressive set of physiological measurements.

5.4.3 Performance comparisons

Since we focus on the AUC for the TPR vs. PPV performance, the AUC values are nominally less than that for the TPR vs. FPR curves. The AUC values in the TPR vs. PPV analyses are usually less than 0.5, whereas in the TPR vs. FPR analysis they can reach 0.8 (Rothman et al. (2013)). As mentioned earlier, random guessing yields an area under the TPR vs. PPV curve that is as small as 0.05. Table 3 reports the AUC and timeliness (in hours) for: ♡ HASMM, ♣ sequential (sliding-window) classification benchmarks, ♠ deep learning algorithms, ★ HMMs and ♢ clinical risk scores. As we can see, all the machine learning algorithms significantly outperform the state-of-the-art clinical risk scores (Rothman, MEWS, APACHE and SOFA). The reason behind the significant performance gain of the HASMM as compared to the clinical risk scores is that it incorporates the patients' history when updating the forward messages (as shown in Figure 17), and reasons about the future trajectory when computing the risk score (as discussed in Section 3.3). Clinical risk scores are instantaneous in that they map the current physiological measurements to a risk score without considering the previously measured physiological variables, and hence they are vulnerable to high false alarm rates (low PPV). Moreover, the clinical risk scores do not reason about the future trajectory given the current physiological measurements, and hence they display a sluggish risk signal that fail to quickly cope with subtle clinical deterioration.

With the exception of MTGPs, all the competing machine learning baselines are incapable of handling irregularly sampled data. Hence, for the baselines, we discretized the time domain into steps of 1 hour and interpolated the missing samples using zero-order-hold filtering. (We have tried cubic spline interpolation as well but this yield worse accuracies for the baselines.) The timeliness values reported in Table 3 are evaluated for the operating point for which the TPR is 50% and the PPV is 35%; this operating point was decided by

Table 3: Performance comparisons for various algorithms.

		AUC	Timeliness (hours)
♡	**HASMM**	**0.489**	**8 hrs 34 mins**
♣	Random Forest	0.362	4 hrs 21 mins
	Logistic Regression	0.271	4 hrs 36 mins
	XGBoost	0.374	7 hrs 6 mins
	AdaBoost	0.323	6 hrs 52 mins
	Bagging	0.293	6 hrs 31 mins
	MTGP	0.365	6 hrs 44 mins
	LASSO	0.261	5 hrs 21 mins
♠	RNN-BFGS	0.293	7 hrs 48 mins
	RNN-ADAM	0.311	8 hrs 6 mins
	DNN-BFGS	0.426	7 hrs 21 mins
	DNN-ADAM	0.366	6 hrs 21 mins
★	HMM	0.321	8 hrs 39 mins
♢	Rothman	0.251	—
	MEWS	0.180	—
	SOFA	0.131	—
	APACHE	0.143	—

our medical collaborators as an acceptable balance between predictive accuracy and alarm fatigue. Non of the clinical risk scores were able to achieve the desired operating point at any timeliness level.

We can see from Table 3 that HASMMs outperforms conventional HMMs; this is a consequence of incorporating temporal correlations and semi-Markovian state transitions, which more accurately describe the patient's physiology. This manifests in the sojourn time distributions in Figure 14, which largely deviate from the exponential distribution adopted by an HMM, and also manifests in the temporal correlation patterns in Figure 15, which largely deviate from the Dirac-delta function and are clearly discriminative of the different states. RNNs trained via BFGS and ADAM did not display high predictive power, probably due to the relatively limited sample size of the patient cohort under study. DNNs operating on a sliding window provided a competitive predictive accuracy: the most competitive was a DNN trained with BFGS and had its hyper-parameters (number of layers and hidden units) optimized using grid search, outperforming the different ensemble methods involved in the comparisons (XGBoost, Random Forest, AdaBoost, and Bagging).

As expected, algorithms that involved a principled time series models provided the most timely predictions as they not only evaluate the current measurements but also forecast the future. Conventional HMMs, due to their memoryless nature, provided the most timely pre-

dictions, slightly exceeding the timeliness of our model with a tight margin of 4.8 minutes. This comes at a huge false alarms' cost, manifesting in a relatively poor AUC of 0.32. This means that using a conventional HMM for risk scoring can provide decent timeliness for the patients who are identified as deteriorating, but would miss a large number of patients who will go unidentified. Contrarily, our model provides excellent timeliness together with high accuracy.

We stress that not only the proposed model outperforms the competing models in terms of accuracy, but also unlike these models, it provides a clinically interpretable model that can be used for understanding the nuances of the complex critical care setting and guiding clinical practice and ward management policies (see Figure 17). This epistemic value cannot be obtained from any of the black-box predictors, including RNNs and DNNs.

5.4.4 Controlled Analysis of the HASMM Performance

Table 3 demonstrates the performance gains achieved by our model, but it does not show the individual contributions of the different elements of the model in achieving these gains. In Table 4, we report the results of a controlled analysis of our model by investigating the impact of removing individual modeling aspects and assessing the resulting performance. To this end, we create three versions of our model, listed below, where each version lacks one of the modeling aspects:

- HASMM*: this version of the model does not capture the temporal correlations in the observations. We implement this model by forcing the length-scale parameter ℓ for all the observations to be infinite in all the iterations of the FFBS-EM algorithm. The resulting observation model corresponds to a model with independent Gaussian emissions at each time step.
- HASMM**: this version of the model adopts an exponential distribution for the sojourn times of all states. We implement this model by forcing the shape parameter $\lambda_{i,s}$ for every state i to be equal to 1.
- HASMM***: this version of the model adopts duration-independent transitions for all states. We implement this model by forcing the parameters β_{ij} to be equal to 0 for all i and j.

Table 4: A controlled analysis of the HASMM modeling aspects.

	AUC
HASMM	0.489
HASMM*	0.425
HASMM**	0.445
HASMM***	0.462

As we can see in Table 4, every aspect of the HASMM model contributes to its predictive capacity. Removing temporal correlations from our model (HASMM*) caused the biggest

drop in the AUC. This shows the importance of capturing temporal correlations (i.e. physiological trends) in predicting the endpoint outcomes. The model HASMM* still significantly outperforms a standard HMM, mainly because it captures semi-Markovianity, whereas an HMM exhibits memoryless transitions, which significantly increases the false alarms. As we can see from the performance of the HASMM**, semi-Markovian transitions are instrumental in improving the AUC performance, mainly because of their role in reducing the false alarms by mitigating rapid transitions in the state process. The HASMM***, which removes duration dependence, is also inferior to the full HASMM in terms of accuracy. It is important to note that the model HASMM*** captures temporal trends in irregularly sampled data, which cannot be achieved via simpler auto-regressive HMM models that operate in discrete time.

It is important to note that our model is not developed with the exclusive goal of predicting clinical outcomes; the epistemic value of interpreting the model parameters are of great importance for managing the complex (and rather poorly understood) critical care environment. Hence, while Table 4 shows that all modeling aspects contribute to the predictive accuracy, it is worth mentioning that all of these aspects contribute to the extracted clinical knowledge as well. (This clinical knowledge is summarized in Figures 13-17.) For instance, modeling the duration dependence is not just aimed at improving the model's accuracy, but is also crucial for understating how should a clinician schedule the therapeutic interventions for a certain patient over time even during the same clinical state. A concrete example for clinical knowledge extracted from our model is the knowledge that the HASMM learned about state 3. Having learned from the model that blood urea nitrogen is predictive of clinical deterioration (see Subsection 5.4.2) and is relatively high in clinical state 3, the clinician can use the information on duration dependence in 16 to manage the administration and timing of drugs, such as Allopurinol and Aminoglycoside antibiotics, that may increase the urea nitrogen.

6. Conclusions

We developed a versatile model, which we call the Hidden Absorbing Semi-Markov Model (HASMM), for clinical time series data which accurately represents physiological data in modern EHRs. The HASMM can deal with irregularly sampled, temporally correlated, and informatively censored physiological data with non-stationary clinical state transitions. We also proposed an efficient Monte Carlo EM learning algorithms that is based on particle filtering, and developed an inference algorithm that can effectively carry out real-time inferences. We have shown, using a real-world dataset for patients admitted to the Ronald Reagan UCLA Medical Center, that HASMMs provide a significant gain in critical care prognosis when utilized for constructing an early warning and risk scoring system.

. Acknowledgments

We would like to thank Dr. Scott Hu (Division of Pulmonary and Critical Care Medicine, Department of Medicine, David Geffen School of Medicine, UCLA) for providing us with the clinical data and the appropriate medical background and insights used in Section 5. We also thank Mr. Jinsung Yoon for his valuable help with the simulations in Section 5.

This research was funded by grants from the Office of Naval Research (ONR) and NSF ECCS 1462245.

Appendix A. An Algorithm for Sampling episodes from an HASMM

Algorithm 7 Sampling episodes from an HASMM

procedure GENERATEHASMM(Γ)
  Input: HASMM model parameters $\Gamma = (N, \lambda, \mathbf{p}^o, \mathbf{Q}(s), \mathbf{\Theta}, \zeta)$
  Output: An episode $(\{X_n\}_{n=1}^K, \{\tau_n\}_{n=1}^K, \{Y(t_m)\}_{m=1}^M, \{t_m\}_{m=1}^M)$
  $\tau_1 \leftarrow 0$, $k \leftarrow 1$, $\mathcal{T} \sim$ Poisson(ζ) ▷ Initializations
  $x_1 \sim$ Multinomial($p_1^o, p_2^o, \ldots, p_N^o$) ▷ Sample an initial latent state
  $s_1 \sim$ Gamma($\lambda_{x_1,s}, \lambda_{x_1,r}$), $\tau_2 \leftarrow \tau_1 + s_1$
  $\mathcal{T}_1 = \{t \in \mathcal{T} : \tau_1 \leq t \leq \tau_2\}$
  while $x_k \notin \{1, N\}$ **do** ▷ Sample latent states until absorption
    $x_{k+1} \sim$ Multinomial($g_{x_k 1}(s_k), g_{x_k 2}(s_k), \ldots, g_{x_k N}(s_k)$)
    $s_{k+1} \sim$ Gamma $\left(\lambda_{x_{k+1},s}, \lambda_{x_{k+1},r}\right)$, $\tau_{k+2} \leftarrow \tau_{k+1} + s_{k+1}$
    $\mathcal{T}_{k+1} = \{t \in \mathcal{T} : \tau_{k+1} \leq t \leq \tau_{k+2}\}$
    $\{y(t_m)\}_{t_m \in \mathcal{T}_{k+1}} \sim \mathcal{GP}(\Theta_{x_{k+1}})$ ▷ Sample observations from a Gaussian Process
    $k \leftarrow k + 1$
  end while
  return $(\{x_n\}_{n=1}^K, \{\tau_n\}_{n=1}^K, \{y(t_m)\}_{m=1}^M, \{t_m\}_{m=1}^M)$
end procedure

Appendix B. Proof of Theorem 1

We start by rewriting (11) as follows:

$$\begin{bmatrix} \tilde{p}_{11}(\tau,\underline{s},\bar{s}) & \ldots & \tilde{p}_{1N}(\tau,\underline{s},\bar{s}) \\ \vdots & \ddots & \vdots \\ \tilde{p}_{N1}(\tau,\underline{s},\bar{s}) & \ldots & \tilde{p}_{NN}(\tau,\underline{s},\bar{s}) \end{bmatrix} = \begin{bmatrix} 1-\bar{Q}_1(\tau,\underline{s},\bar{s}) & \ldots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \ldots & 1-\bar{Q}_N(\tau,\underline{s},\bar{s}) \end{bmatrix} + \int_{u=0}^{\tau} \left(\frac{\partial}{\partial u} \begin{bmatrix} \bar{Q}_{11}(u,\underline{s},\bar{s}) & \ldots & \bar{Q}_{1N}(u,\underline{s},\bar{s}) \\ \vdots & \ddots & \vdots \\ \bar{Q}_{N1}(u,\underline{s},\bar{s}) & \ldots & \bar{Q}_{NN}(u,\underline{s},\bar{s}) \end{bmatrix} \right) \times \begin{bmatrix} \tilde{p}_{11}(\tau-u,0,0) & \ldots & \tilde{p}_{1N}(\tau-u,0,0) \\ \vdots & \ddots & \vdots \\ \tilde{p}_{N1}(\tau-u,0,0) & \ldots & \tilde{p}_{NN}(\tau-u,0,0) \end{bmatrix} du. \tag{29}$$

Starting with the left hand side, we can use a first-step analysis to write every term $\tilde{p}_{ij}(\tau,\underline{s},\bar{s})$ as follows

$$\begin{aligned} \tilde{p}_{ij}(\tau,\underline{s},\bar{s}) &= \mathbb{P}(X(t+\tau)=j|X(t)=i, \underline{s} \leq S(t) \leq \bar{s}) \\ &= \delta_{ij}\left(\mathbb{P}(S_i<\tau|X(t)=i,\underline{s}\leq S(t)\leq\bar{s})\right)+ \\ &\int_{u=0}^{\tau} \mathbb{P}(X(t+u)=k|X(t)=i,\underline{s}\leq S(t)\leq\bar{s}) \cdot \mathbb{P}(X(t+\tau)=j|X(t+u)=k)\,du \\ &= \delta_{ij}\left(1-\bar{Q}_i(\tau,\underline{s},\bar{s})\right)+ \\ &\int_{u=0}^{\tau} \mathbb{P}(X(t+u)=k|X(t)=i,\underline{s}\leq S(t)\leq\bar{s}) \cdot \mathbb{P}(X(t+\tau-u)=j|X(t)=k)\,du \\ &= \delta_{ij}\left(1-\bar{Q}_i(\tau,\underline{s},\bar{s})\right) + \int_{u=0}^{\tau} \frac{\partial}{\partial u} \sum_{k\neq i} \bar{Q}_{ik}(u,\underline{s},\bar{s}) \cdot \tilde{p}_{kj}(\tau-u,0,0)\,du, \end{aligned} \tag{30}$$

$\forall i,j \in \mathcal{X}$, where $S(t)$ is the time elapsed in state $X(t)$, and S_i is the sojourn time of state i. The integral equation in (30) can be written in a matrix form as in the right hand side of (29), and hence the Theorem follows.

Appendix C. Proof of Theorem 2

Recall that the operation

$$\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau,\underline{s},\bar{s})\}(\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}))$$

can be written as

$$\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}) = \mathbf{I}_{N\times N} - \text{diag}\left(\bar{Q}_1(\tau,\underline{s},\bar{s}),\ldots,\bar{Q}_N(\tau,\underline{s},\bar{s})\right) + \left(\frac{\partial\bar{\mathbf{Q}}(.,\underline{s},\bar{s})}{\partial u} \star \tilde{\mathbf{P}}(.,0,0)\right)(\tau).$$

Now consider n applications of the operator $\mathcal{B}(.)$, we have that

$$\begin{aligned} \left(\frac{\partial\bar{\mathbf{Q}}(.,\underline{s},\bar{s})}{\partial u_1} \star \ldots \star \frac{\partial\bar{\mathbf{Q}}(.,\underline{s},\bar{s})}{\partial u_n} \star \tilde{\mathbf{P}}(.,0,0)\right)(\tau) &\leq N^n \cdot \int_0^{\tau}\int_0^{\tau-u_{n-1}} \ldots \int_0^{\tau-u_1} du_1\,du_2\ldots,du_n \\ &= N^n \cdot \frac{\tau^n}{n!}. \end{aligned} \tag{31}$$

Thus, for every $\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}) \in \mathcal{P}$ and every $\bar{\mathbf{Q}}(\tau,\underline{s},\bar{s}) < 1$, there exists n such that $\mathcal{B}^n\{.\}(.)$ is a contraction mapping. Therefore, the operation $\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}) = \mathcal{B}\{\bar{\mathbf{Q}}(\tau,\underline{s},\bar{s})\}(\tilde{\mathbf{P}}(\tau,\underline{s},\bar{s}))$ has a unique fixed point that can be reached via $n \in \mathbb{N}$ successive approximations.

Appendix D. The M-step of the FFBS-MCEM Algorithm

The proximal likelihood function at the z^{th} iteration is given by

$$\hat{U}_G(\Gamma;\hat{\Gamma}^{z-1}) = \sum_{d,g} \log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d}|\,\Gamma)) \cdot \frac{I_{d,g}^{z-1}}{G},$$

where the likelihood inside the logarithm can be factorized as follows

$$d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma) = \mathbb{P}(x_1^{d,g}|\Gamma) \cdot d\mathbb{P}(s_1^{d,g}|x_1^{d,g}, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_1^d} | x_1^{d,g}, \Gamma) \times$$
$$\prod_{n=2}^{k^{d,g}} \mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma) \cdot d\mathbb{P}(s_n^{d,g} \,|\, x_n^{d,g}, \Gamma) \cdot d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma),$$

and hence the log-likelihood is given by

$$\log(d\mathbb{P}(\{x_n^{d,g}, s_n^{d,g}\}_{n=1}^{k^{d,g}}, \{y_m^d, t_m^d\}_{m=1}^{M^d} \,|\, \Gamma)) = \log(\mathbb{P}(x_1^{d,g}|\Gamma)) + \sum_{n=2}^{k^{d,g}} \log(\mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma))$$
$$+ \sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(s_n^{d,g} \,|\, x_n^{d,g}, \Gamma)) + \sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma)).$$

The updated parameter set $\hat{\Gamma}^z$ is obtained by maximizing:

$$\hat{\Gamma}^z = \arg\max_{\Gamma} \hat{U}_G(\Gamma; \hat{\Gamma}^{z-1}).$$

Updating the initial state distribution is straightforwardly conducted as follows

$$\hat{p}_k^{o,z} = \frac{1}{G \cdot D} \sum_{d,g} \mathbf{1}_{\{x_1^{d,g}=k\}} \cdot \frac{I_{d,g}^{z-1}}{G}.$$

For the k^{th} state sojourn time distribution parameters (shape and rate parameters for the Gamma distribution), we solve the optimization problem by computing the Maximum Likelihood Estimate (MLE) based on the observed sojourn times in the sampled trajectories as follows

$$\mathcal{N}_k = \{(d, g, n) : x_n^{d,g} = k\},$$
$$\Xi_k = \log\left(\frac{1}{|\mathcal{N}_k|} \sum_{(d,g,n)\in\mathcal{N}_k} s_n^{d,g}\right) - \frac{1}{|\mathcal{N}_k|} \sum_{(d,g,n)\in\mathcal{N}_k} \log\left(s_n^{d,g}\right),$$
$$\hat{\lambda}_{k,s}^z = \frac{3 - \Xi_k + \sqrt{(\Xi_k - 3)^2 - 24\Xi_k}}{12\Xi_k},$$
$$\hat{\lambda}_{k,r}^z = \frac{\hat{\lambda}_{k,s}^z |\mathcal{N}_k|}{\sum_{(d,g,n)\in\mathcal{N}_k} s_n^{d,g}}.$$

The optimization problem is intractable for the rest of the parameters, and hence we resort to approximate solutions. For the transition parameters, we maximize the term $\sum_{n=2}^{k^{d,g}} \log(\mathbb{P}(x_n^{d,g} \,|\, x_{n-1}^{d,g}, s_{n-1}^{d,g}, \Gamma))$ using the successive approximations, whereas for the GP parameters, we maximize the term $\sum_{n=1}^{k^{d,g}} \log(d\mathbb{P}(\{y_m^d, t_m^d\}_{t_m^d \in \mathcal{T}_n^d} \,|\, x_n^{d,g}, \Gamma))$ via conjugate gradient descent.

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *OSDI*, volume 16, pages 265–283, 2016.

Ahmed M. Alaa and Mihaela van der Schaar. Balancing suspense and surprise: Timely decision making with endogenous information acquisition. In *Advances in Neural Information Processing Systems*, pages 2910–2918, 2016.

Ahmed M Alaa, Jinsung Yoon, Scott Hu, and Mihaela van der Schaar. Personalized risk scoring for critical care prognosis using mixtures of gaussian processes. *arXiv preprint arXiv:1610.08853*, 2016.

Ahmed M. Alaa, Scott Hu, and Mihaela van der Schaar. Learning from clinical judgments: Semi-markov-modulated marked hawkes processes for risk prognosis. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Jeffrey A Bakal, Finlay A McAlister, Wei Liu, and Justin A Ezekowitz. Heart failure re-admission: measuring the ever shortening gap between repeat heart failure hospitalizations. *PloS one*, 9(9):e106494, 2014.

Jirina Bartkova, Zuzana Hořejší, Karen Koed, Alwin Krämer, Frederic Tort, Karsten Zieger, Per Guldberg, Maxwell Sehested, Jahn M Nesland, Claudia Lukas, et al. Dna damage response as a candidate anti-cancer barrier in early human tumorigenesis. *Nature*, 434 (7035):864–870, 2005.

Kevin Beier, Sabitha Eppanapally, Heidi S Bazick, Domingo Chang, Karthik Mahadevappa, Fiona K Gibbons, and Kenneth B Christopher. Elevation of bun is predictive of long-term mortality in critically ill patients independent of'normal'creatinine. *Critical care medicine*, 39(2):305, 2011.

Edwin V Bonilla, Kian M Chai, and Christopher Williams. Multi-task gaussian process prediction. In *Advances in neural information processing systems*, pages 153–160, 2007.

James G Booth and James P Hobert. Maximizing generalized linear mixed model likelihoods with an automated monte carlo em algorithm. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(1):265–285, 1999.

Brian S Caffo, Wolfgang Jank, and Galin L Jones. Ascent-based monte carlo expectation–maximization. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):235–251, 2005.

Lucienne TQ Cardoso, Cintia MC Grion, Tiemi Matsuo, Elza HT Anami, Ivanil AM Kauss, Ludmila Seko, and Ana M Bonametti. Impact of delayed admission to intensive care units on mortality of critically ill patients: a cohort study. *Critical Care*, 15(1):R28, 2011.

Chris K Carter and Robert Kohn. On gibbs sampling for state space models. *Biometrika*, 81(3):541–553, 1994.

Dustin Charles, Meghan Gabriel, and JaWanna Henry. Electronic capabilities for patient engagement among us non-federal acute care hospitals: 2012-2014. *The Office of the National Coordinator for Health Information Technology*, 2015.

Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *arXiv preprint arXiv:1606.01865*, 2016.

Baojiang Chen and Xiao-Hua Zhou. Non-homogeneous markov process models with informative observations with an application to alzheimer's disease. *Biometrical Journal*, 53 (3):444–463, 2011.

Jill M Cholette, Kelly F Henrichs, George M Alfieris, Karen S Powers, Richard Phipps, Sherry L Spinelli, Michael Swartz, Francisco Gensini, L Eugene Daugherty, Emily Nazarian, et al. Washing red blood cells and platelets transfused in cardiac surgery reduces post-operative inflammation and number of transfusions: Results of a prospective, randomized, controlled clinical trial. *Pediatric critical care medicine: a journal of the Society of Critical Care Medicine and the World Federation of Pediatric Intensive and Critical Care Societies*, 13(3), 2012.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proceedings of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.

Zelalem Getahun Dessie. Multi-state models of hiv/aids by homogeneous semi-markov process. *American Journal of Biostatistics*, 4(2):21, 2014.

Michael Dewar, Chris Wiggins, and Frank Wood. Inference in hidden markov models with explicit state duration distributions. *IEEE Signal Processing Letters*, 19(4):235–238, 2012.

Rick Durrett. *Probability: theory and examples*. Cambridge university press, 2010.

Allison A Eddy and Eric G Neilson. Chronic kidney disease progression. *Journal of the American Society of Nephrology*, 17(11):2964–2966, 2006.

Yohann Foucher, Eve Mathieu, Philippe Saint-Pierre, J Durand, and J Daures. A semi-markov model based on generalized weibull distribution with an illustration for hiv disease. *Biometrical journal*, 47(6):825, 2005.

Yohann Foucher, Magali Giral, Jean-Paul Soulillou, and Jean-Pierre Daures. A semi-markov model for multistate and interval-censored data with multiple terminal events. application in renal transplantation. *Statistics in medicine*, 26(30):5381–5393, 2007.

Yohann Foucher, M Giral, JP Soulillou, and JP Daures. A flexible semi-markov model for interval-censored data and goodness-of-fit testing. *Statistical methods in medical research*, 2008.

Emily Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. Bayesian nonparametric inference of switching dynamic linear models. *IEEE Transactions on Signal Processing*, 59(4):1569–1585, 2011a.

Emily B Fox, Erik B Sudderth, Michael I Jordan, and Alan S Willsky. A sticky hdp-hmm with application to speaker diarization. *The Annals of Applied Statistics*, pages 1020–1056, 2011b.

Mitchell H Gail and Phuong L Mai. Comparing breast cancer risk assessment models. *Journal of the National Cancer Institute*, 102(10):665–668, 2010.

Valentine Genon-Catalot, Thierry Jeantheau, Catherine Larédo, et al. Stochastic volatility models as hidden markov models and statistical applications. *Bernoulli*, 6(6):1051–1079, 2000.

Konstantinos Georgatzis, Christopher KI Williams, and Christopher Hawthorne. Input-output non-linear dynamical systems applied to physiological condition monitoring. *Journal of Machine Learning Research*, 2016.

Zoubin Ghahramani and Michael I Jordan. Factorial hidden markov models. *Machine learning*, 29(2-3):245–273, 1997.

Marzyeh Ghassemi, Marco AF Pimentel, Tristan Naumann, Thomas Brennan, David A Clifton, Peter Szolovits, and Mengling Feng. A multivariate timeseries modeling approach to severity of illness assessment and forecasting in icu with sparse, heterogeneous clinical data. In *Proceedings of the... AAAI Conference on Artificial Intelligence. AAAI Conference on Artificial Intelligence*, volume 2015, page 446. NIH Public Access, 2015.

Giacomo Giampieri, Mark Davis, and Martin Crowder. Analysis of default data using hidden markov models. *Quantitative Finance*, 5(1):27–34, 2005.

Florence Gillaizeau, Etienne Dantan, Magali Giral, and Yohann Foucher. A multistate additive relative survival semi-markov model. *Statistical methods in medical research*, page 0962280215586456, 2015.

Simon J Godsill, Arnaud Doucet, and Mike West. Monte carlo smoothing for nonlinear time series. *Journal of the american statistical association*, 99(465):156–168, 2004.

Peter J Green and David I Hastie. Reversible jump mcmc. *Genetics*, 155(3):1391–1403, 2009.

Sheffield ML group. Gpy: A gaussian process framework in python. 2012.

Amit Gruber, Yair Weiss, and Michal Rosen-Zvi. Hidden topic markov models. In *AISTATS*, volume 7, pages 163–170, 2007.

Yann Guédon. Exploring the state sequence space for hidden markov and semi-markov chains. *Computational Statistics & Data Analysis*, 51(5):2379–2409, 2007.

Chantal Guihenneuc-Jouyaux, Sylvia Richardson, and Ira M Longini. Modeling markers of disease progression by a hidden markov process: application to characterizing cd4 cell decline. *Biometrics*, 56(3):733–741, 2000.

Tracy D Gunter and Nicolas P Terry. The emergence of national electronic health record architectures in the united states and australia: models, costs, and questions. *Journal of medical Internet research*, 7(1):e3, 2005.

Alan G Hawkes and David Oakes. A cluster process representation of a self-exciting process. *Journal of Applied Probability*, pages 493–503, 1974.

CharlesO Hershey and Linda Fisher. Why outcome of cardiopulmonary resuscitation in general wards is poor. *The Lancet*, 319(8262):31–34, 1982.

Asger Hobolth and Jens Ledet Jensen. Summary statistics for endpoint-conditioned continuous-time markov chains. *Journal of Applied Probability*, pages 911–924, 2011.

Helen Hogan, Frances Healey, Graham Neale, Richard Thomson, Charles Vincent, and Nick Black. Preventable deaths due to problems in care in english acute hospitals: a retrospective case record review study. *BMJ quality & safety*, pages bmjqs–2012, 2012.

William Hoiles and Mihaela van der Schaar. A non-parametric learning method for confidently estimating patient's clinical state and dynamics. In *Advances in Neural Information Processing Systems*, pages 2020–2028, 2016.

George Hripcsak, David J Albers, and Adler Perotte. Parameterizing time in electronic health record studies. *Journal of the American Medical Informatics Association*, 22(4): 794–804, 2015.

Xuelin Huang and Robert A Wolfe. A frailty model for informative censoring. *Biometrics*, 58(3):510–520, 2002.

Aparna V Huzurbazar. Multistate models, flowgraph models, and semi-markov processes. 2004.

Christopher H Jackson, Linda D Sharples, Simon G Thompson, Stephen W Duffy, and Elisabeth Couto. Multistate markov models for disease progression with classification error. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 52(2):193–209, 2003.

Jacques Janssen and R De Dominicis. Finite non-homogeneous semi-markov processes: Theoretical and computational aspects. *Insurance: Mathematics and Economics*, 3(3): 157–165, 1984.

Daniel W Johnson, Ulrich H Schmidt, Edward A Bittner, Benjamin Christensen, Retsef Levi, and Richard M Pino. Delay of transfer from the intensive care unit: a prospective observational study of incidence, causes, and financial impact. *Critical Care*, 17(4):R128, 2013.

Matthew J Johnson and Alan S Willsky. Bayesian nonparametric hidden semi-markov models. *Journal of Machine Learning Research*, 14(Feb):673–701, 2013.

Pierre Joly and Daniel Commenges. A penalized likelihood approach for a progressive three-state model with censored and truncated data: Application to aids. *Biometrics*, 55(3): 887–890, 1999.

Jared Katzman, Uri Shaham, Jonathan Bates, Alexander Cloninger, Tingting Jiang, and Yuval Kluger. Deep survival: A deep cox proportional hazards network. *arXiv preprint arXiv:1606.00931*, 2016.

Juliane Kause, Gary Smith, David Prytherch, Michael Parr, Arthas Flabouris, Ken Hillman, et al. A comparison of antecedents to cardiac arrests, deaths and emergency intensive care admissions in australia and new zealand, and the united kingdom—the academia study. *Resuscitation*, 62(3):275–282, 2004.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Lisa L Kirkland, Michael Malinchoc, Megan O'Byrne, Joanne T Benson, Deanne T Kashiwagi, M Caroline Burton, Prathibha Varkey, and Timothy I Morgenthaler. A clinical deterioration prediction tool for internal medicine patients. *American Journal of Medical Quality*, 28(2):135–142, 2013.

William A Knaus, Elizabeth A Draper, Douglas P Wagner, and Jack E Zimmerman. Apache ii: a severity of disease classification system. *Critical care medicine*, 13(10):818–829, 1985.

William A Knaus, Douglas P Wagner, Elizabeth A Draper, Jack E Zimmerman, Marilyn Bergner, Paulo G Bastos, Carl A Sirio, Donald J Murphy, Ted Lotring, and Anne Damiano. The apache iii prognostic system. risk prediction of hospital mortality for critically ill hospitalized adults. *Chest Journal*, 100(6):1619–1636, 1991.

Vidyadhar G Kulkarni. *Modeling and analysis of stochastic systems*. CRC Press, 1996.

Stephan W Lagakos, Charles J Sommer, and Marvin Zelen. Semi-markov models for partially censored data. *Biometrika*, 65(2):311–317, 1978.

David Lando. On cox processes and credit risky securities. *Review of Derivatives research*, 2(2-3):99–120, 1998.

Laura Landro. Hospitals find new ways to monitor patients 24/7. *The Wall Street Journal*, 2015.

Jose Leiva-Murillo, AA Rodrguez, and E Baca-Garca. Visualization and prediction of disease interactions with continuous-time hidden markov models. In *NIPS 2011 Workshop on Personalized Medicine*, 2011.

H Lehman Li-wei, Shamim Nemati, Ryan P Adams, George Moody, Atul Malhotra, and Roger G Mark. Tracking progression of patient state of health in critical care using inferred shared dynamics in physiological time series. In *2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7072–7075. IEEE, 2013.

William A Link. A model for informative censoring. *Journal of the American Statistical Association*, 84(407):749–752, 1989.

Zachary C Lipton, David Kale, and Randall Wetzel. Directly modeling missing data in sequences with rnns: Improved classification of clinical time series. In *Machine Learning for Healthcare Conference*, pages 253–270, 2016.

Yu-Ying Liu, Shuang Li, Fuxin Li, Le Song, and James M Rehg. Efficient learning of continuous-time hidden markov models for disease progression. In *Advances in neural information processing systems*, pages 3600–3608, 2015.

Sergio Matos, Surinder S Birring, Ian D Pavord, and H Evans. Detection of cough signals in continuous audio recordings using hidden markov models. *IEEE Transactions on Biomedical Engineering*, 53(6):1078–1083, 2006.

Raina M Merchant, Lin Yang, Lance B Becker, Robert A Berg, Vinay Nadkarni, Graham Nichol, Brendan G Carr, Nandita Mitra, Steven M Bradley, Benjamin S Abella, et al. Incidence of treated cardiac arrest in hospitalized patients in the united states. *Critical care medicine*, 39(11):2401, 2011.

Philipp Metzner, Illia Horenko, and Christof Schütte. Generator estimation of markov jump processes based on incomplete observations nonequidistant in time. *Physical Review E*, 76(6):066702, 2007.

Rui P Moreno, Philipp GH Metnitz, Eduardo Almeida, Barbara Jordan, Peter Bauer, Ricardo Abizanda Campos, Gaetano Iapichino, David Edbrooke, Maurizia Capuzzo, Jean-Roger Le Gall, et al. Saps 3-from evaluation of the patient to evaluation of the intensive care unit. part 2: Development of a prognostic model for hospital mortality at icu admission. *Intensive care medicine*, 31(10):1345–1355, 2005.

RJM Morgan, F Williams, and MM Wright. An early warning scoring system for detecting developing critical illness. *Clin Intensive Care*, 8(2):100, 1997.

DR Mould. Models for disease progression: new approaches and uses. *Clinical Pharmacology & Therapeutics*, 92(1):125–131, 2012.

Kevin Murphy et al. The bayes net toolbox for matlab. *Computing science and statistics*, 33(2):1024–1034, 2001.

Kevin P Murphy. Hidden semi-markov models (hsmms). *unpublished notes*, 2, 2002.

Uri Nodelman, Christian R Shelton, and Daphne Koller. Expectation maximization and complex duration distributions for continuous time bayesian networks. *arXiv preprint arXiv:1207.1402*, 2012.

Zdzisław Opial. Weak convergence of the sequence of successive approximations for nonexpansive mappings. *Bulletin of the American Mathematical Society*, 73(4):591–597, 1967.

Mari Ostendorf, Vassilios V Digalakis, and Owen A Kimball. From hmm's to segment models: A unified view of stochastic modeling for speech recognition. *IEEE Transactions on speech and audio processing*, 4(5):360–378, 1996.

Soren Erik Pedersen, Suzanne S Hurd, Robert F Lemanske, Allan Becker, Heather J Zar, Peter D Sly, Manuel Soto-Quiroz, Gary Wong, and Eric D Bateman. Global strategy for the diagnosis and management of asthma in children 5 years and younger. *Pediatric pulmonology*, 46(1):1–17, 2011.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(Oct):2825–2830, 2011.

Romain Pirracchio, Maya L Petersen, Marco Carone, Matthieu Resche Rigon, Sylvie Chevret, and Mark J van der Laan. Mortality prediction in intensive care units with the super icu learner algorithm (sicula): a population-based study. *The Lancet Respiratory Medicine*, 3(1):42–52, 2015.

Andrei D Polyanin and Alexander V Manzhirov. *Handbook of integral equations*. CRC press, 2008.

Ross L Prentice, John D Kalbfleisch, Arthur V Peterson Jr, Nancy Flournoy, Vern T Farewell, and Norman E Breslow. The analysis of failure times in the presence of competing risks. *Biometrics*, pages 541–554, 1978.

Zhen Qin and Christian R Shelton. Auxiliary gibbs sampling for inference in piecewise-constant conditional intensity models. In *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence*, 2015.

Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

Rajesh Ranganath, Adler Perotte, Noémie Elhadad, and David Blei. Deep survival analysis. In *Machine Learning for Healthcare Conference*, pages 101–114, 2016.

Carl Edward Rasmussen. Gaussian processes for machine learning. 2006.

Santiago Romero-Brufau, Jeanne M Huddleston, Gabriel J Escobar, and Mark Liebow. Why the c-statistic is not informative to evaluate early warning scores and what metrics to use. *Critical Care*, 19(1):285, 2015.

Michael J Rothman, Steven I Rothman, and Joseph Beals. Development and validation of a continuous measure of patient condition using the electronic medical record. *Journal of biomedical informatics*, 46(5):837–848, 2013.

Mohammed Saeed, Christine Lieu, Greg Raber, and Roger G Mark. Mimic ii: a massive temporal icu patient database to support research in intelligent patient monitoring. In *Computers in Cardiology, 2002*, pages 641–644. IEEE, 2002.

Daniel O Scharfstein and James M Robins. Estimation of the failure time distribution in the presence of informative censoring. *Biometrika*, 89(3):617–634, 2002.

Peter Schulam and Suchi Saria. A framework for individualizing predictions of disease trajectories by exploiting multi-resolution structure. In *Advances in Neural Information Processing Systems*, pages 748–756, 2015.

Padhraic Smyth. Hidden markov models for fault detection in dynamic systems. *Pattern recognition*, 27(1):149–164, 1994.

Henry T Stelfox, Brenda R Hemmelgarn, Sean M Bagshaw, Song Gao, Christopher J Doig, Cheri Nijssen-Jordan, and Braden Manns. Intensive care unit bed availability and outcomes for hospitalized patients with sudden clinical deterioration. *Archives of internal medicine*, 172(6):467–474, 2012.

CP Subbe, M Kruger, P Rutherford, and L Gemmel. Validation of a modified early warning score in medical admissions. *Qjm*, 94(10):521–526, 2001.

MJ Sweeting, VT Farewell, and D De Angelis. Multi-state markov models for disease progression in the presence of informative examination times: An application to hepatitis c. *Statistics in medicine*, 29(11):1161–1174, 2010.

S Taghipour, D Banjevic, AB Miller, N Montgomery, AKS Jardine, and BJ Harvey. Parameter estimates for invasive breast cancer progression in the canadian national breast screening study. *British journal of cancer*, 108(3):542–548, 2013.

Hale F Trotter and John W Tukey. Conditional monte carlo for normal samples. In *Symposium on Monte Carlo Methods*, pages 64–79. Wiley, 1956.

John Varga, Christopher P Denton, and Fredrick M Wigley. *Scleroderma: From pathogenesis to comprehensive management*. Springer Science & Business Media, 2012.

J-L Vincent, Rui Moreno, Jukka Takala, Sheila Willatts, Arnaldo De Mendonça, Hajo Bruining, CK Reinhart, PeterM Suter, and LG Thijs. The sofa (sepsis-related organ failure assessment) score to describe organ dysfunction/failure. *Intensive care medicine*, 22(7):707–710, 1996.

Xiang Wang, David Sontag, and Fei Wang. Unsupervised learning of disease progression models. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 85–94. ACM, 2014.

Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

J Yoon, A Alaa, S Hu, and M van der Schaar. Forecasticu: A prognostic decision support system for timely prediction of intensive care unit admission. pages 1680–1689, 2016.

Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

Shun Yu, Sharon Leung, Moonseong Heo, Graciela J Soto, Ronak T Shah, Sampath Gunda, and Michelle Ng Gong. Comparison of risk prediction scoring systems for ward patients: a retrospective nested case-control study. *Critical Care*, 18(3):1, 2014.

Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

Yongyue Zhang, Michael Brady, and Stephen Smith. Segmentation of brain mr images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE transactions on medical imaging*, 20(1):45–57, 2001.

Journal of Machine Learning Research 19 (2018) 1-66 Submitted 1/17; Revised 12/17; Published 08/18

Can We Trust the Bootstrap in High-dimensions? The Case of Linear Models

Noureddine El Karoui nkaroui@berkeley.edu, n.elkaroui@criteo.com
Criteo AI Lab
32 Rue Blanche
75009 Paris, France
and
Department of Statistics
University of California
Berkeley, CA 94270, USA

Elizabeth Purdom epurdom@stat.berkeley.edu
Department of Statistics
University of California
Berkeley, CA 94270, USA

Editor: Guy Lebanon

Abstract

We consider the performance of the bootstrap in high-dimensions for the setting of linear regression, where $p < n$ but p/n is not close to zero. We consider ordinary least-squares as well as robust regression methods and adopt a minimalist performance requirement: can the bootstrap give us good confidence intervals for a single coordinate of β (where β is the true regression vector)?

We show through a mix of numerical and theoretical work that the bootstrap is fraught with problems. Both of the most commonly used methods of bootstrapping for regression—residual bootstrap and pairs bootstrap—give very poor inference on β as the ratio p/n grows. We find that the residual bootstrap tend to give anti-conservative estimates (inflated Type I error), while the pairs bootstrap gives very conservative estimates (severe loss of power) as the ratio p/n grows. We also show that the jackknife resampling technique for estimating the variance of $\hat{\beta}$ severely overestimates the variance in high dimensions.

We contribute alternative procedures based on our theoretical results that result in dimensionality adaptive and robust bootstrap methods.

Keywords: Bootstrap, high-dimensional inference, random matrices, resampling

1. Introduction

The bootstrap (Efron, 1979) is a ubiquitous tool in applied statistics, allowing for inference when very little is known about the properties of the data-generating distribution. The bootstrap is a powerful tool in applied settings because it does not make the strong assumptions common to classical statistical theory regarding this data-generating distribution. Instead, the bootstrap resamples the observed data to create an estimate, $\hat{F}$, of the unknown data-generating distribution, F. The distribution $\hat{F}$ then forms the basis of further inference.

Since its introduction, a large amount of research has explored the theoretical properties of the bootstrap, improvements for estimating F under different scenarios, and how to most effectively estimate different quantities from $\hat{F}$ (see the pioneering Bickel and Freedman, 1981 for instance and many many more references in the book-length review of Davison and Hinkley, 1997, as well as van der Vaart, 1998 for a short summary of the modern point of view on these questions). Other resampling techniques exist of course, such as subsampling, m-out-of-n bootstrap, and jackknifing, all of which have been studied and much discussed (see Efron, 1982; Hall, 1992; Politis et al., 1999; Bickel et al., 1997; and Efron and Tibshirani, 1993 for a practical introduction).

An important limitation for the bootstrap is the quality of $\hat{F}$. The standard bootstrap estimate of F based on the empirical distribution of the data may be a poor estimate when the data has a non-trivial dependency structure, when the quantity being estimated, such as a quantile, is sensitive to the discreteness of $\hat{F}$, or when the functionals of interest are not smooth (see e.g., Bickel and Freedman, 1981 for a classic reference, as well as Beran and Srivastava, 1985 or Eaton and Tyler, 1991 in the context of multivariate statistics).

An area that has received less attention is the performance of the bootstrap in high dimensions and this is the focus of our work. In particular, we consider the setting of standard linear models where data y_i are drawn from the linear model

$$\forall i, y_i = \beta' X_i + \epsilon_i \ , 1 \leq i \leq n \ , \quad \text{where } X_i \in \mathbb{R}^p \ .$$

We are interested in the bootstrap or resampling properties of the estimator defined as

$$\widehat{\beta}_\rho = \text{argmin}_{b \in \mathbb{R}^p} \sum_{i=1}^{n} \rho(y_i - X_i' b) \ , \text{ where } \rho \text{ is a convex function.}$$

We consider the two standard methods for resampling to create a bootstrap distribution in this setting. The first is *pairs resampling*, where bootstrap samples are drawn from the empirical distribution of the pairs (y_i, X_i). The second resampling method is *residual resampling*, where the bootstrapped data consists of $y_i^* = \widehat{\beta}_\rho' X_i + \hat{\epsilon}_i^*$, where $\hat{\epsilon}_i^*$ is drawn from the empirical distribution of the estimated residuals, e_i. We also consider the jackknife, a resampling method focused specifically on estimating the variance of functionals of $\widehat{\beta}_\rho$. These three methods are extremely flexible for linear models regardless of the method of fitting β or the error distribution of the ϵ_i.

The high dimensional setting: $\boldsymbol{p/n \to \kappa \in (0,1)}$ In this work we call a high-dimensional setting one where the number of predictors, p, is of the same order of magnitude as the number of observations, n, formalized mathematically by assuming that $p/n \to \kappa \in (0,1)$. Several reasons motivate our theoretical study in this regime. The asymptotic behavior of the estimate $\widehat{\beta}_\rho$ is known to depend heavily on whether one makes the classical theoretical assumption that $p/n \to 0$ or instead assumes $p/n \to \kappa \in (0,1)$ (see Section 1.2 and AppendixA and references therein). But from the standpoint of practical usage on moderate-sized data sets (i.e., n and p both moderately sized with $p < n$), it is not always obvious which assumption is justified. Working in the high-dimensional regime of $p/n \to \kappa \in (0,1)$ captures better the complexity encountered even in reasonably low-dimensional practice than using the classical assumption $p/n \to 0$. In fact, asymptotic predictions based on the

high-dimensional assumption can work surprisingly well in very low-dimension (see Johnstone, 2001). Furthermore, in these high-dimensional settings, where much is still unknown theoretically, the bootstrap is a natural and compelling alternative to asymptotic analysis.

Another motivation for our investigation is that of very large scale applications (Chapelle et al., 2014; Criteo, 2017; Langford et al., 2007), where one might resort to subsampling methods or recent variants like the bag-of-little-bootstraps (Kleiner et al., 2014) for uncertainty assessment. Subsampling is also very commonly used in this setting for simple computational speed-up. In such cases, even if one had started with a data set where $p \ll n$, after subsampling one often ends up with p comparable to n on the subsamples where bootstrap-like computations are performed. It is therefore important to know if the bootstrap and other resampling plans perform well when p is comparable to n.

Defining success: accurate inference on β_1 The common theoretical definition of whether the bootstrap "works" is that the bootstrap distribution of the entire bootstrap estimate $\widehat{\beta}^*_\rho$ converges conditionally almost surely to the sampling distribution of the estimator $\widehat{\beta}_\rho$ (see e.g., van der Vaart, 1998). The work of Bickel and Freedman (1983) on the residual bootstrap for least squares regression, which we discuss in the background section 1.2, shows that this theoretical requirement is not fulfilled even for the simple problem of least squares regression.

In this paper, we choose to focus only on accurate inference for the projection of our parameter on a pre-specified direction v. More specifically, we concentrate only on whether the bootstrap gives accurate confidence intervals for $v'\beta$. We think that this is the absolute minimal requirement we can ask of a bootstrap inferential method, as well as one that is meaningful from the standpoint of applied statistics. This is of course a much less stringent requirement than performing well on complicated functionals of the whole parameter vector, which is the implicit demand of standard definitions of bootstrap success. For this reason, we focus throughout the exposition on inference for β_1 (the first element of β) as an example of a pre-defined direction of interest (where β_1 corresponds to choosing $v = e_1$, the first canonical basis vector).

We note that considering the asymptotic behavior of $v'\beta$ as $p/n \to \kappa \in (0, 1)$ implies that $v = v(p)$ changes with p. By "pre-defined" we will mean simply a deterministic sequence of directions $v(p)$. We will continue to suppress the dependence on p in writing v in what follows for the sake of clarity.

1.1 Organization and Main Results of the Paper

In Section 2 we demonstrate that in high dimensions residual-bootstrap resampling results in extremely poor inference on the coordinates of β_ρ with error rates much higher than the reported Type I error. We show that the error in inference based on residual bootstrap resampling is due to the fact that the distribution of the residuals e_i are a poor estimate of the distribution of ϵ_i; we further illustrate that common methods of standardizing the e_i do not solve the problem for general ρ. We propose two new dimension-adaptive methods of residual resampling that appear promising for use in bootstrapping linear models. We also provide some theoretical results for the behavior of this method as $p/n \to 1$.

In Section 3 we examine *pairs-bootstrap resampling* and show that confidence intervals based on bootstrapping the pairs also perform very poorly. Unlike in the residual-bootstrap

case discussed in Section 2, the confidence intervals obtained from the pairs-bootstrap are instead conservative to the point of being non-informative. This results in a dramatic loss of power. We prove in the case of L_2 loss, i.e., $\rho(x) = x^2$, that the variance of the bootstrapped $v'\widehat{\beta}^*$ is greater than that of $v'\widehat{\beta}$, leading to the overly conservative performance we see in simulations. We demonstrate that a different resampling scheme we propose can provide accurate confidence intervals in moderately high dimensions.

In Section 4, we discuss another resampling scheme, the jackknife. We focus on the jackknife estimate of variance and show that it has similarly poor behavior in high dimensions. In the case of L_2 loss with Gaussian design matrices, we further prove that the jackknife estimator over estimates the variance of our estimator by a factor of $1/(1-p/n)$; we also provide corrections for other losses that improve the jackknife estimate of variance in moderately high dimensions.

We rely on simulation results to demonstrate the practical impact of the failure of the bootstrap. The settings for our simulations and corresponding theoretical analyses are idealized, without many of the common settings of heteroskedasticity, dependency, outliers and so forth that are known to be a problem for bootstrapping. This is intentional, since even these idealized settings are sufficient to demonstrate that the standard bootstrap methods have poor performance. For brevity, we give only brief descriptions of the simulations in what follows; detailed descriptions can be found in AppendixD.1.

Similarly, we focus on the basic implementations of the bootstrap for linear models. While there are many proposed alternatives (often for specific loss functions or types of data), the standard methods we study are most commonly used and recommended in practice. Furthermore, to our knowledge none of the alternative bootstrap methods we have seen specifically address the underlying theoretical problems that appear in high dimensions without making low-dimensional assumptions about either the design matrix or the sparsity of β, and therefore are likely to suffer from the same fate as standard methods. We note that in truly large scale applications, sparsity assumptions are not always made by practitioners (Chapelle et al., 2014; Langford et al., 2007; Criteo, 2017) and it is hence natural to study the performance of estimators outside of sparse settings. We have also experimented with more complicated ways to build confidence intervals (e.g., bias correction methods), but have found their performance to be erratic in high-dimension and offer no improvement.

We first give some background regarding the bootstrap and estimation of linear models in high dimensions before presenting our new results.

1.2 Background: Inference Using the Bootstrap

We consider the setting $y_i = \beta' X_i + \epsilon_i$, where $E(\epsilon_i) = 0$ and $\mathrm{var}(\epsilon_i) = \sigma^2_\epsilon$. The vector β is estimated as minimizing the average loss,

$$\widehat{\beta}_\rho = \mathrm{argmin}_{b\in\mathbb{R}^p} \sum_{i=1}^{n} \rho(y_i - X_i' b), \tag{1}$$

where ρ defines the loss function for a single observation. The function ρ is assumed to be convex in all the paper. Common choices are $\rho(x) = x^2$, i.e., least-squares, $\rho(x) = |x|$, which defines L_1 regression, or Huber$_k$ loss where $\rho(x) = (x^2/2)1_{|x|<k} + (k|x| - k^2/2)1_{|x|\geq k}$.

Bootstrap methods are used in order to estimate the distribution of the estimate $\widehat{\beta}_\rho$ under the true data-generating distribution, F. The bootstrap estimates this distribution with the distribution obtained when the data is drawn from an estimate $\hat{F}$ of F. Following standard convention, we designate this bootstrapped estimator $\widehat{\beta}^*_\rho$ to note that this is an estimate of β using loss function ρ when the data-generating distribution is known to be exactly equal to $\hat{F}$. Since $\hat{F}$ is completely specified, we can in principle exactly calculate the distribution of $\widehat{\beta}^*_\rho$ and use it as an approximation of the distribution of $\widehat{\beta}_\rho$ under F. In practice, we simulate B independent draws of size n from the distribution $\hat{F}$ and perform inference based on the empirical distribution of $\widehat{\beta}^{*b}_\rho$, $b = 1, \ldots, B$.

In bootstrap inference for the linear model, there are two common methods for re-sampling, which results in different estimates $\hat{F}$. In the first method, called the residual bootstrap, $\hat{F}$ is an estimate of the conditional distribution of y_i given β and X_i. In this case, the corresponding resampling method consists of resampling ϵ^*_i from an estimate of the distribution of ϵ and forming data $y^*_i = X_i'\widehat{\beta}_\rho + \epsilon^*_i$, from which $\widehat{\beta}^*_\rho$ is computed. This method of bootstrapping assumes that the linear model is correct for the mean of y (i.e., that $\mathbf{E}\,(y_i) = X_i'\beta$); it also assumes fixed X_i design vectors because the sampling is conditional on the X_i. In the second method, called pairs bootstrap, $\hat{F}$ is an estimate of the joint distribution of the vector $(y_i, X_i) \in R^{p+1}$ given by the empirical joint distribution of $\{(y_i, X_i)\}_{i=1}^n$; the corresponding resampling method resamples the pairs (y_i, X_i). This method makes no assumption about the mean structure of y and, by resampling the X_i, also does not condition on the values of X_i. For this reason, pairs resampling is often considered to be more generally applicable than residuals resampling (see e.g., Davison and Hinkley, 1997).

1.3 Background: High-dimensional Inference of Linear Models

Recent research shows that $\widehat{\beta}_\rho$ has very different asymptotic properties when p/n has a limit κ that is bounded away from zero than it does in the classical setting where $p/n \to 0$ (see e.g., Huber, 1973; Huber and Ronchetti, 2009; Portnoy, 1984, 1985, 1986, 1987; Mammen, 1989 for $\kappa = 0$; El Karoui et al., 2013 for $\kappa \in (0, 1)$). A simple example is that the vector $\widehat{\beta}_\rho$ is no longer consistent in Euclidean norm when $\kappa > 0$. We should be clear, however, that projections on fixed non-random directions such as we consider, i.e., $v'\widehat{\beta}_\rho$, are $\sqrt{n}$ consistent for $v'\beta$, even when $\kappa > 0$. In particular, the coordinates of $\widehat{\beta}_\rho$ are $\sqrt{n}-$consistent for the coordinates of β (El Karoui et al., 2013, Lemma 1). Hence, in practice the estimator $\widehat{\beta}_\rho$ is still a reasonable quantity to consider.

Bootstrap in high-dimensional linear models Very interesting work exists already in the literature about bootstrapping regression estimators when p is allowed to grow with n (Shorack, 1982; Wu, 1986; Mammen, 1989, 1992, 1993; Parzen et al., 1994; Koenker, 2005, Section 3.9). With a few exceptions, this work has been in the classical, low-dimensional setting where either p is held fixed or p grows slowly relative to n (i.e., $\kappa = 0$ in our notation). For instance, in Mammen (1993), it is shown that under mild technical conditions and assuming that $p^{1+\delta}/n \to 0$, $\delta > 0$, the pairs bootstrap distribution of linear contrasts $v'(\widehat{\beta}^*_\rho - \widehat{\beta}_\rho)$ is in fact very close to the sampling distribution of $v'(\widehat{\beta}_\rho - \beta)$ with high-probability, when using least-squares. Other results, such as Shorack (1982) and Mammen (1989),

also allow for increasing dimensions, for example in the case of linear contrasts in robust regression, by making assumptions on the diagonal entries of the hat matrix. In our context, these assumptions would be satisfied only if $p/n \to 0$. Hence those interesting results do not apply to the present study. We also note that Hall (1992, p. 167) contains cautionary notes about using the bootstrap in high-dimension.

While there has not been much theoretical work on the bootstrap in the setting where $p/n \to \kappa \in (0, 1)$, one early work of Bickel and Freedman (1983) considered bootstrapping scaled residuals for least-squares regression when $\kappa > 0$. They show that when $p/n \to \kappa \in (0, 1)$, there exists a data-dependent direction c, such that $c'\widehat{\beta}^*_{LS}$ does not have the correct asymptotic distribution (Bickel and Freedman, 1983, Theorem 3.1, p.39), i.e., its distribution is not conditionally in probability close to the sampling distribution of $c'\widehat{\beta}_{LS}$. Furthermore, they show that when the errors in the model are Gaussian, under the assumption that the diagonal entries of the hat matrix are not all close to a constant, the empirical distribution of the residuals is a scaled-mixture of Gaussian, which is not close to the original error distribution.

As we previously explained, in this work we instead only consider inference for *predefined* contrasts $v'\beta$. The important and interesting problems pointed out in Bickel and Freedman (1983) disappear if we focus on fixed, non-data-dependent projection directions. Hence, our work complements the work of Bickel and Freedman (1983) and is not redundant with it.

There has been some recent interest in residual bootstrap methods for penalized likelihood methods in high-dimensions (often proposed for the case when $p >> n$), for example lasso estimates (Chatterjee and Lahiri, 2010, 2011), adaptive lasso estimates (Chatterjee and Lahiri, 2013), de-biased lasso estimates (Belloni et al., 2015; Dezeure et al., 2017), and ridge regression (Lopes, 2014). These bootstrap results make the assumption of sparsity of some form, generally in terms of the number of non-zero components of β, but in the case of Lopes (2014) by the assumption that the design matrix X is nearly low-rank. As explained previously, our work is focused on a very different line of inquiry: the case of a comparatively diffuse signal in β, where there is no reduction of the high-dimensional problem to a low-dimensional approximation.

The role of the distribution of X An important consideration in interpreting theoretical work on linear models in high dimensions is the role of the design matrix X. In classical asymptotic theory, the results can be stated conditionally on X so that the assumptions can be stated in terms of conditions that can be evaluated on any observed design matrix X. In the high dimensional setting, the available theoretical tools do not yet allow for an asymptotic analysis conditional on X; instead the results make assumptions about the distribution of the entries of X. Theoretical work in the nascent literature for the high dimensional setting usually allows for a fairly general class of distributions for the individual elements of X_i and can handle covariance between the predictor variables. However, the X_i's are generally considered i.i.d., which limits the ability of any X_i to be too influential in the fit of the model (see AppendixA for more detail). For discussion of limitations of the corresponding models for statistical purposes, see Diaconis and Freedman (1984); Hall et al. (2005); El Karoui (2009).

1.4 Notations and Default Conventions

When referring to the Huber loss in a numerical context, we refer (unless otherwise noted) to the default implementation in the `rlm` package in R, where the transition from quadratic to linear behavior is at $k = 1.345$. We call X the design matrix and $\{X_i\}_{i=1}^n$ its rows. We have $X_i \in \mathbb{R}^p$. β denotes the true regression vector, i.e., the population parameter. $\widehat{\beta}_\rho$ refers to the estimate of β using loss ρ; from this point on, however, we will often drop the ρ and refer to simply $\widehat{\beta}$. The i-th residual is denoted as e_i, i.e., $e_i = y_i - X_i'\widehat{\beta}$. Throughout the paper, we assume that the linear model holds, i.e., $y_i = X_i'\beta + \epsilon_i$ for some fixed $\beta \in \mathbb{R}^p$ and that ϵ_i's are i.i.d. with mean 0 and $\text{var}(\epsilon_i) = \sigma_\epsilon^2$. We call G the distribution of ϵ. When we need to stress the impact of the error distribution on the distribution of $\widehat{\beta}_\rho$, we will write $\widehat{\beta}_\rho(G)$ or $\widehat{\beta}_\rho(\epsilon)$ to denote our estimate of β obtained assuming that ϵ_i's are i.i.d. G.

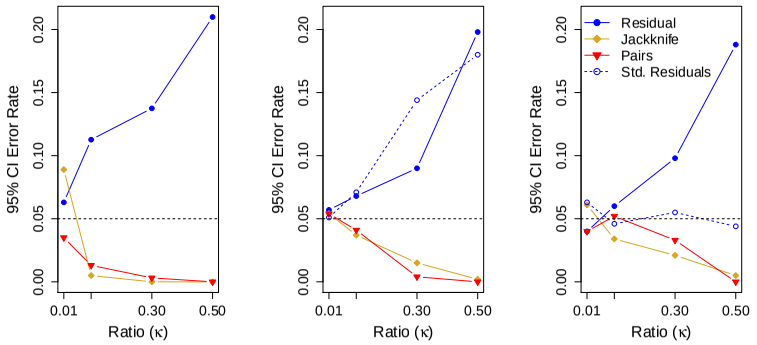
We denote generically by $\kappa = \lim_{n\to\infty} p/n$. We restrict ourselves to $\kappa \in (0,1)$. The standard notation $\widehat{\beta}_{(i)}$ refers to the leave-one-out estimate of $\widehat{\beta}$ where the i-th pair (y_i, X_i) is excluded from the regression, and $\tilde{e}_{i(i)} \triangleq y_i - X_i'\widehat{\beta}_{(i)}$ is the i-th predicted error (based on the leave-one-out estimate of $\widehat{\beta}$). We also use the notation $\tilde{e}_{j(i)} \triangleq y_j - X_j'\widehat{\beta}_{(i)}$. The hat matrix is of course $H = X(X'X)^{-1}X'$. o_P denotes a "little-oh" in probability, a standard notation (see van der Vaart, 1998). When we say that we work with a Gaussian design with covariance Σ, we mean that $X_i \overset{iid}{\backsim} \mathcal{N}(0,\Sigma)$. Throughout the paper, the loss function ρ is assumed to be convex, $\mathbb{R} \mapsto \mathbb{R}^+$. We use the standard notation $\psi = \rho'$. We finally assume that ρ is such that there is a unique solution to the robust regression problem—an assumption that applies to all classical losses in the context of our paper.

2. Residual Bootstrap

We first focus on the method of bootstrap resampling where $\hat{F}$ is the conditional distribution $y|\widehat{\beta}, X$. In this case the distribution of $\widehat{\beta}^*$ under $\hat{F}$ is formed by independent resampling of ϵ_i^* from an estimate $\hat{G}$ of the distribution G that generated ϵ_i. Then new data y_i^* are formed as $y_i^* = X_i'\widehat{\beta} + \epsilon_i^*$ and the model is fitted to this new data to get $\widehat{\beta}^*$. Generally the estimate of the error distribution, $\hat{G}$, is taken to be empirical distribution of the observed residuals, so that the ϵ_i^* are found by sampling with replacement from the e_i.

Yet, even a cursory evaluation of e_i in the simple case of least-squares regression ($\rho(x) = x^2$) reveals that the empirical distribution of the e_i may be a poor approximation to the error distribution of ϵ_i. In particular, it is well known that e_i has variance equal to $\sigma_\epsilon^2(1-h_i)$ where h_i is the ith diagonal element of the hat matrix. This problem becomes particularly pronounced in high dimensions. For instance, if $X_i \overset{iid}{\backsim} \mathcal{N}(0,\Sigma)$, $h_i = p/n + o_P(1)$ so that e_i has variance approximately $\sigma_\epsilon^2(1-p/n)$, i.e., generally much smaller than the true variance of ϵ for $\lim p/n > 0$. This fact is also true when assuming more general distributions for the design matrix X (see e.g.,Wachter, 1978; Haff, 1979; Silverstein, 1995; Pajor and Pastur, 2009; El Karoui and Koesters, 2011, where the main results of some of these papers require minor adjustments to get the approximation of h_i we just mentioned).

In Figure 1, we plot the error rate of 95% bootstrap confidence intervals based on resampling from the residuals for different loss functions, based on a simulation when the entries of X are i.i.d. $\mathcal{N}(0,1)$ and $\epsilon \sim N(0,1)$. Even in this idealized situation, as the ratio of



(a) L_1 loss (b) Huber loss (c) L_2 loss

Figure 1: **Performance of 95% confidence intervals of β_1 :** Here we show the coverage error rates for 95% confidence intervals for $n = 500$ based on applying common resampling-based methods to simulated data: pairs bootstrap (red), residual bootstrap (blue), and jackknife estimates of variance (yellow). These bootstrap methods are applied with three different loss functions shown in the three plots above: (a) L_1, (b) Huber, and (c) L_2. For L_2 and Huber loss, we also show the performance of methods for standardizing the residuals before bootstrapping described in the text (blue, dashed line). If accurate, all of these methods should have an error rate of 0.05 (shown as a horizontal black line). Error rates above 5% correspond to anti-conservative methods. Error rates below 5% correspond to conservative methods. The error rates are based on 1,000 simulations, with $N(0,1)$ error, and entries of the design matrix i.i.d $N(0,1)$; see the description in Appendix D.1 for more details. The exact values plotted here are given in Table A-1 in Appendix I.

p/n increases the error rate of the confidence intervals in least squares regression increases well beyond the expected 5%: we observe error rates of 10-15% for $p/n = 0.3$ and approximately 20% for $p/n = 0.5$. We see similar error rates for other robust-regression methods, such as L_1 and Huber loss, and also for different error distributions and distributions of X (Supplementary Figures A-1 and A-2). We explain some of the reasons for these problems in Subsection 2.2 below.

2.1 Bootstrapping from Corrected Residuals

While resampling directly from the uncorrected residuals is widespread and often given as a standard bootstrap procedure (e.g., Koenker, 2005; Chernick, 1999), the discrepancy between the distribution of ϵ_i and e_i has spurred more refined recommendations in the

case of least-squares: form corrected residuals $r_i = e_i/\sqrt{1-h_i}$ and sample the ϵ_i^* from the empirical distribution of the $r_i - \bar{r}$ (see e.g., Davison and Hinkley, 1997).

This correction is known to exactly align the variance of r_i with that of ϵ_i regardless of the design vectors X_i or the true error distribution, using simply the fact that the hat matrix is a rank $min(n,p)$ orthogonal projection matrix. We see that for L_2 loss it corrects the error in bootstrap inference in our simulations (Figure 1). This is not so surprising, given that with L_2 loss, the error distribution G impacts the inference on β only through σ_ϵ^2, in the case of homoskedastic errors (see Section 2.4 for much more detail).

However, this adjustment of the residuals is a correction specific to the least-squares problem. Similar corrections for robust estimation procedures using a loss function ρ are given by McKean et al. (1993) with standardized residuals r_i given by,

$$r_i = \frac{e_i}{\sqrt{1-dh_i}}, \text{ where } d = \frac{2\sum e_j'\psi(e_j')}{\sum \psi(e_j')} - \frac{\sum \psi(e_j')^2}{(\sum \psi(e_j'))^2}, \tag{2}$$

where h_i is the i-th diagonal entry of the hat matrix, $e_j' = e_j/s$, s is a estimate of σ, and ψ is the derivative of ρ, assuming ψ is a bounded and odd function (see Davison and Hinkley, 1997 for a complete description of its implementation for the bootstrap and McKean et al., 1993 for a full description of the regularity conditions).

Unlike the correction for L_2 loss mentioned earlier, however, the scaling described in Equation (2) for the residuals is an approximate variance correction, and the approximation depends on assumptions that do not hold true in higher dimensions. The error rate of confidence intervals in our simulations based on this rescaling show no improvement in high dimensions over that of simple bootstrapping of the residuals. This could be explained by the fact that standard perturbation analytic methods used for the analysis of M-estimators in low-dimension, which are at the heart of the correction in Equation (2), fail in high-dimensions.

2.2 Understanding the Behavior of the Residual Bootstrap

This misbehavior of the residual bootstrap can be explained by the fact that in high-dimension, the residuals tend to have a very different distribution from that of the true errors. Their distributions differ not only in simple properties, such as their variances, but in more general aspects, such as their marginal distributions. To make these statements precise, we make use of the previous work of El Karoui et al. (2013) and El Karoui (2013). These papers do not discuss bootstrap or resampling issues, but rather are entirely focused on providing asymptotic theory for the behavior of $\widehat{\beta}_\rho$ as $p/n \to \kappa \in (0,1)$; in the course of doing so, they characterize the asymptotic relationship of e_i to ϵ_i in high-dimensions. We make use of this relationship to characterize the behavior of the residual bootstrap and to suggest an alternative estimates of $\hat{G}$ for bootstrap resampling.

Behavior of residuals in high-dimensional regression We now summarize the asymptotic relationship between e_i and ϵ_i in high-dimensions given in the above cited work (see AppendixA for a more detailed and technical summary). Let $\widehat{\beta}_{(i)}$ be the estimate of β based on fitting the linear model of Equation (1) without using observation i, and $\tilde{e}_{j(i)}$ be the error of observation j from this model (the leave-one-out or predicted error), i.e., $\tilde{e}_{j(i)} = y_j - X_j'\widehat{\beta}_{(i)}$

For simplicity of exposition, X_i is assumed to have an elliptical distribution, i.e., $X_i = \lambda_i\Gamma_i$, where $\Gamma_i \sim N(0,\Sigma)$, and λ_i is a scalar random variable independent of Γ_i with $\mathbf{E}\left(\lambda_i^2\right) = 1$. For simplicity in restating their results, we will assume $\Sigma = \mathrm{Id}_p$, but equivalent statements can be made for arbitrary Σ; similar results also apply when $\Gamma_i = \Sigma^{1/2}\xi_i$, with ξ_i having i.i.d. non-Gaussian entries, satisfying a few technical requirements (see AppendixA).

With this assumption on X_i, for any sufficiently smooth loss function ρ and any size dimension where $p/n \to \kappa < 1$, the relationship between the i-th residual e_i and the true error ϵ_i can be summarized as,

$$\tilde{e}_{i(i)} = \epsilon_i + |\lambda_i|\|\widehat{\beta}_{\rho(i)} - \beta\|_2 Z_i + \mathrm{o}_P(u_n) \tag{3}$$

$$e_i + c_i\lambda_i^2\psi(e_i) = \tilde{e}_{i(i)} + \mathrm{o}_P(u_n) \tag{4}$$

where Z_i is a random variable distributed $N(0,1)$ and independent of ϵ_i. The variable u_n refers to a sequence of numbers tending to 0. The quantities c_i, λ_i and $\|\widehat{\beta}_{\rho(i)} - \beta\|_2$ are all of order 1, i.e., they are not close to 0 in general in the high-dimensional setting. The scalar c_i is given as $\frac{1}{n}\mathrm{trace}\left(S_i^{-1}\right)$, where $S_i = \frac{1}{n}\sum_{j\neq i}\psi'(\tilde{e}_{j(i)})X_jX_j'$. For p,n large the c_i's are approximately equal and $\|\widehat{\beta}_{\rho(i)} - \beta\|_2 \simeq \|\widehat{\beta}_\rho - \beta\|_2 \simeq \mathbf{E}\left(\|\widehat{\beta}_\rho - \beta\|_2\right)$; furthermore $c_i\lambda_i^2$ can be approximated by $X_i'S_i^{-1}X_i/n$. Note that when ρ is either non-differentiable at all points (L_1) or not twice differentiable (Huber), arguments can be made that make these expressions valid, using for instance the notion of sub-differential for ψ (Hiriart-Urruty and Lemaréchal, 2001).

Interpretation of Equations (3) *and* (4) Equation (3) means that the marginal distribution of the leave-i-th-out predicted error, $\tilde{e}_{i(i)}$, is asymptotically a convolution of the true error, ϵ_i, and an independent scale mixture of Normals. Furthermore, Equation (4) means that the i-th residual e_i can be understood as a non-linear transformation of $\tilde{e}_{i(i)}$. As we discuss below, these relationships are qualitatively very different from the classical case $p/n \to 0$.

2.2.1 Consequence for the Residual Bootstrap

We apply these results to the question of the residual bootstrap to give an understanding of why bootstrap resampling of the residuals can perform so badly in high-dimension. The distribution of the e_i is far removed from that of the ϵ_i, and hence bootstrapping from the residuals effectively amounts to sampling errors from a distribution that is very different from the original error distribution, ϵ.

The impact of these discrepancies for bootstrapping is not equivalent for all dimensions, error distributions, or loss functions. It depends on the constant c_i and the risk, $\|\widehat{\beta}_{\rho(i)} - \beta\|_2$, both of which are highly dependent on the dimensions of the problem, the distribution of the errors and the choice of loss function. We now discuss some of these issues.

Least Squares regression In the case of least squares regression, the relationships given in Equation (3) are exact, i.e., $u_n = 0$. Further, $\psi(x) = x$, and $c_i = h_i/(1-h_i)$, giving the well known linear relationship $e_i = (1-h_i)\tilde{e}_{i(i)}$ (see, e.g., the standard reference Weisberg, 2014). This linear relationship is exact regardless of dimension, though the dimensionality aspects are captured by h_i. This expression can be used to show that asymptotically $\mathbf{E}\left(\sum_{i=1}^n e_i^2\right) = \sigma_\epsilon^2(n-p)$, when ϵ_i's have the same variance. Hence, sampling at random

from the residuals results in a distribution that underestimates the variance of the errors by a factor $1 - p/n$. The corresponding bootstrap confidence intervals are then naturally too small, and hence the error rate increases far from the nominal 5% - as we observed in Figure 1c.

More general robust regression The situation is much more complicated for general robust regression estimators. One clear implication of Equations (3) and (4) is that simply rescaling the residuals e_i should not in general result in an estimated error distribution $\hat{G}$ that will have similar properties to those of G. The relationship between the residuals and the errors is very non-linear in high-dimensions. This is why in what follows we will propose to work with leave-one-out predicted errors $\tilde{e}_{i(i)}$ instead of the residuals e_i.

The classical case of $\boldsymbol{p/n \to 0}$*:* In this setting, $c_i \to 0$ and therefore Equation (3) shows that the residuals e_i are approximately equal in distribution to the predicted errors, $\tilde{e}_{i(i)}$. Similarly, $\widehat{\beta}_\rho$ is L_2 consistent when $p/n \to 0$, so $\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2 \to 0$ and Equation (4) gives $\tilde{e}_{i(i)} \simeq \epsilon_i$. Hence, the residuals should be fairly close to the true errors in the model when p/n is small. This dimensionality assumption is key to many theoretical analyses of robust regression, and underlies the derivation of corrected residuals r_i of McKean et al. (1993) given in Equation (2) above for losses other than L_2.

2.3 Alternative Residual Bootstrap Procedures

We propose two methods for improving the performance of confidence intervals obtained through the residual bootstrap. Both do so by providing alternative estimates of $\hat{G}$ from which bootstrap errors ϵ_i^* can be drawn. They estimate a $\hat{G}$ appropriate for the setting of high-dimensional data by accounting for relationship of the distribution of ϵ and $\tilde{e}_{i(i)}$.

Method 1: Deconvolution The relationship in Equation (3) says that the distribution of $\tilde{e}_{i(i)}$ is a convolution of the correct G distribution and a normal distribution. This suggests applying techniques for deconvolving a signal from Gaussian noise. Specifically, we propose the following bootstrap procedure: **1)** calculate the predicted errors, $\tilde{e}_{i(i)}$; **2)** estimate the variance of the normal (i.e., $|\lambda_i|\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2$); **3)** deconvolve in $\tilde{e}_{i(i)}$ the error term ϵ_i from the normal term; **4)** Use the resulting estimate $\hat{G}$ to draw errors ϵ_i^* for residual bootstrapping.

Deconvolution problems are known to be very difficult (see Fan, 1991, Theorem 1 p. 1260, that gives $1/\log(n)^\alpha$ rates of convergence when convolving with a Gaussian distribution). The resulting deconvolved errors are likely to be quite noisy estimates of ϵ_i. However, it is possible that while individual estimates are poor, the distribution of the deconvolved errors is estimated well-enough to form a reasonable $\hat{G}$ for the bootstrap procedure.

We used the deconvolution algorithm in the `decon` package in R (Wang and Wang, 2011) to estimate the distribution of ϵ_i. The deconvolution algorithm requires knowledge of the variance of the Gaussian that is convolved with the ϵ_i, i.e., estimation of $|\lambda_i|\|\widehat{\beta}_{\rho(i)} - \beta\|_2$ term. In what follows, we assume a Gaussian design, i.e., $\lambda_i = 1$, so that we need to estimate only the term $\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2$. An estimation strategy for the more general setting of $|\lambda_i| \neq 1$ is presented in AppendixB.5. We use the fact that $\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2 \simeq \|\widehat{\beta}_\rho - \beta\|_2^2$ for all i and estimate $\|\widehat{\beta}_{\rho(i)} - \beta\|_2$ as $\widehat{var}(\tilde{e}_{i(i)}) - \hat{\sigma}_\epsilon^2$, where $\widehat{var}(\tilde{e}_{i(i)})$ is the empirical variance of the $\tilde{e}_{i(i)}$ and $\hat{\sigma}_\epsilon^2$ is an estimate of the variance of G, which we discuss below. We note

that the deconvolution strategy we employ makes assumptions of homoskedastic errors ϵ_i's, which is true in our simulations but may not be true in practice. See AppendixB for details regarding the implementation of Method 1.

Method 2: Bootstrapping from standardized $\tilde{e}_{i(i)}$ A simpler alternative is bootstrapping from the predicted error terms, $\tilde{e}_{i(i)}$, without deconvolution. Specifically, we propose to bootstrap from a scaled version of $\tilde{e}_{i(i)}$,

$$\tilde{r}_{i(i)} = \frac{\hat{\sigma}_\epsilon}{\sqrt{\widehat{var}(\tilde{e}_{i(i)})}}\tilde{e}_{i(i)},$$

where $\widehat{var}(\tilde{e}_{i(i)})$ is the standard estimate of the variance of $\tilde{e}_{i(i)}$ and $\hat{\sigma}_\epsilon$ is an estimate of σ_ϵ. This scaling aligns the first two moments of $\tilde{e}_{i(i)}$ with those of ϵ_i. On the face of it, resampling from $\tilde{r}_{i(i)}$ seems problematic, since Equation (3) demonstrates that $\tilde{e}_{i(i)}$ does not have the same distribution as ϵ_i, even if the first two moments are the same. However, as we demonstrate in simulations, this distributional mismatch appears to have limited practical effect on our bootstrap confidence intervals.

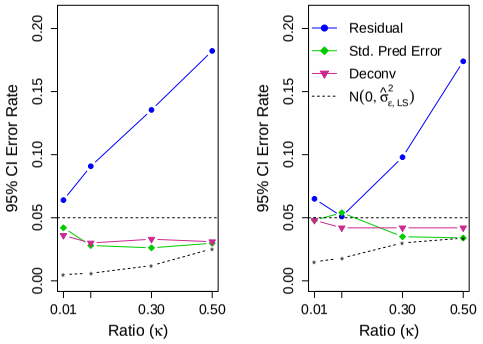
Estimation of σ_ϵ^2 Both methods described above require an estimator of σ_ϵ that is consistent regardless of dimension and error distribution. As we have explained earlier, for general ρ we cannot rely on the observed residuals e_i nor on $\tilde{e}_{i(i)}$ for estimating σ_ϵ (see Equations (3) and (4)). The exception is the standard estimate of σ_ϵ^2 from least-squares regression, i.e., $\rho(x) = x^2$,

$$\widehat{\sigma}_{\epsilon,LS}^2 = \frac{1}{n-p}\sum_i e_{i,L_2}^2.$$

$\widehat{\sigma}_{\epsilon,LS}^2$ is a consistent estimator of σ^2 for any error distribution G, assuming i.i.d. errors and mild moment requirements. In implementing the two alternative residual-bootstrap methods described above, we use $\widehat{\sigma}_{\epsilon,LS}$ as our estimate of σ_ϵ, including for bootstrapping robust regression where $\rho(x) \neq x^2$.

Performance in bootstrap inference In Figure 2 we show the error rate of confidence intervals based on the two residual-bootstrap methods we proposed above. We see that both methods control the Type I error, unlike bootstrapping directly from the residuals, and that both methods are conservative. There is little difference between the two methods with this sample size ($n = 500$), though with $n = 100$, we observe the deconvolution performance to be worse in L_1 (data not shown).

The deconvolution strategy, however, depends on the distribution of the design matrix, which in these simulations we assumed was Gaussian (so we did not have to estimate λ_i's). For elliptical designs ($\lambda_i \neq 1$), the error rate of the deconvolution method described above, with no adaptation for the design, was similar to that of uncorrected residuals in high dimensions (i.e., > 0.25 for $p/n = 0.5$). Individual estimates of λ_i might improve the deconvolution strategy, but this problem points to the general reliance of the deconvolution method on precise knowledge about the design matrix. The bootstrap using standardized predicted errors, on the other hand, had a Type I error for an elliptical design only slightly higher than the target 0.05 (around 0.07, data not shown), suggesting that it might be less sensitive to the properties of the design matrix.



(a) L_1 loss (b) Huber loss

Figure 2: **Bootstrap based on predicted errors:** We plotted the error rate of 95% confidence intervals for the alternative bootstrap methods described in Section 2.3: bootstrapping from standardized predicted errors (green) and from deconvolution of predicted error (magenta). We demonstrate its improvement over the standard residual bootstrap (blue) for (a) L_1 loss and (b) Huber loss. The error distribution is double exponential, but otherwise the simulations parameters are as in Figure 1. The error rates on confidence intervals based on bootstrapping from a $N(0, \hat{\sigma}^2_{\epsilon,LS})$ (dashed curve) are as a lower bound on the problem. For the precise error rates see Appendix, Table A-3.

Given our previous discussion of the behavior of $\tilde{e}_{i(i)}$, it is somewhat surprising that resampling from the distribution of $\tilde{r}_{i(i)}$ performed well in our simulations. Clearly a few cases exist where $\tilde{r}_{i(i)}$ should work well as an approximation of ϵ_i. We have already noted that as $p/n \to 0$, the effect of the convolution with the Gaussian disappears since $\|\widehat{\beta}_\rho - \beta\| \to 0$; in this case both e_i and $\tilde{r}_{i(i)}$ should be good estimates of ϵ_i. Similarly, in the case $\epsilon_i \sim N(0, \sigma^2)$, Equation (3) tells us that $\tilde{e}_{i(i)}$ are also asymptotically marginally normally distributed, so that correcting the variance should result in $\tilde{r}_{i(i)}$ having the same distribution as ϵ_i, at least when $X_{i,j}$ are i.i.d.

Surprisingly, for larger p/n we do not see a deterioration of the performance of bootstrapping from $\tilde{r}_{i(i)}$. This is unexpected, since as $p/n \to 1$ the risk $\|\widehat{\beta}_\rho - \beta\|_2^2$ grows to be much larger than σ_ϵ^2 (a claim we will make more precise in the next section); together with Equation (3), this implies that $\tilde{r}_{i(i)}$ is essentially distributed $N(0, \widehat{\sigma}^2_{\epsilon,LS})$ as $p/n \to 1$ regardless of the original distribution of ϵ_i. This is confirmed in Figure 2 where we superimpose the results of bootstrap confidence intervals from when we simply estimate $\hat{G}$ with $N(0, \hat{\sigma}^2_{\epsilon,LS})$; we see the Type I error rate of the confidence intervals based on bootstrapping from $\tilde{r}_{i(i)}$ do indeed approach that of $N(0, \hat{\sigma}^2_{\epsilon,LS})$. Putting these two pieces of information

together leads to the conclusion that as $p/n \to 1$ we can estimate $\hat{G}$ simply as $N(0, \hat{\sigma}_{\epsilon,LS})$ regardless of the actual distribution of ϵ.

In the next section we give some theoretical results that seek to understand this phenomenon.

2.4 Behavior of the Risk of $\widehat{\beta}$ When $\kappa \to 1$

In the previous section we saw even if the distribution of the bootstrap errors ϵ_i^*, given by $\hat{G}$, is not close to that of G, we can sometime get accurate bootstrap confidence intervals. For example, in least squares Equation (3) makes clear that even the standardized residuals, r_i, do not have the same marginal distribution as ϵ_i, yet they still provide accurate bootstrap confidence intervals in our simulations. We would like to understand for what choice of distributions $\hat{G}$ will we see the same performance in our bootstrap confidence intervals of $\widehat{\beta}_1$?

When working conditional on X as in residual resampling, the statistical properties of $(\widehat{\beta}^* - \widehat{\beta})$ differ from that of $(\widehat{\beta} - \beta)$ only because the errors are drawn from a different distribution: $\hat{G}$ rather than G. Then to understand whether the distribution of $\widehat{\beta}_1^*$ matches that of $\widehat{\beta}_1$ we can ask, what are the distributions of errors, G, that yield the same distribution for the resulting $\widehat{\beta}_1(G)$? In this section, we narrow our focus on understanding not the entire distribution of $\widehat{\beta}_1$, but only its variance. We do so because under assumptions on the design matrix X, it is known that $\widehat{\beta}_1$ is asymptotically normally distributed. This is true for both the classical setting of $\kappa = 0$ and the high-dimensional setting of $\kappa \in (0, 1)$ (see AppendixA for a review of these results and a more technical discussion). Our previous question is then reduced to understanding which distributions G give the same var $\left(\widehat{\beta}_1(G)\right)$.

In the setting of least squares, it is clear that the only property of $\epsilon_i \overset{iid}{\sim} G$ that matters for the variance of $\widehat{\beta}_{1,L_2}$ is σ_ϵ^2, since var $\left(\widehat{\beta}_{1,L_2}\right) = (X'X)^{-1}(1,1)\sigma_\epsilon^2$. For general ρ, if we assume $p/n \to 0$, then var $\left(\widehat{\beta}_{1,\rho}\right)$ will depend on features of G beyond the first two moments (specifically through $\mathbf{E}\left(\psi^2(\epsilon)\right)/[\mathbf{E}\left(\psi'(\epsilon)\right)]^2$, (Huber, 1973)). If we assume instead $p/n \to \kappa \in (0,1)$, then it has been shown (El Karoui et al., 2013) that var $\left(\widehat{\beta}_{1,\rho}(G)\right)$ depends on G only by the effect of G on the squared risk of the vector $\widehat{\beta}_\rho(G)$, i.e., through $\mathbf{E}\left(\|\widehat{\beta}_\rho(G) - \beta\|_2^2\right)$ (for the convenience of the reader we give a review of these results, which are a bit scattered in the literature, in AppendixA).

For this reason, in the setting of $p/n \to \kappa \in (0,1)$, we need to characterize the risk of $\widehat{\beta}_\rho$ to understand when different distributions of ϵ result in the same variance of $\widehat{\beta}_{1,\rho}$. In what follows, we denote by $r_\rho^2(\kappa; G)$ the asymptotic squared risk of $\widehat{\beta}_\rho(G)$ as p and n tend to ∞,

$$r_\rho^2(\kappa; G) = \lim_{n,p\to\infty, \frac{p}{n}\to\kappa} E||\widehat{\beta}_\rho(G) - \beta||^2 \, .$$

The dependence of $r_\rho^2(\kappa; G)$ on ϵ is characterized, under appropriate technical conditions on X, ρ and ϵ_i's, by a system of two non-linear equations (El Karoui et al., 2013). Specifically, if we define $\hat{z}_\epsilon = \epsilon + r_\rho(\kappa; G)Z$, where $Z \sim \mathcal{N}(0,1)$ is independent of ϵ, and ϵ has the same distribution G as the ϵ_i's, then there exists a constant c such that the pair of positive,

finite, and deterministic scalars $(c, r_\rho(\kappa;G))$ satisfy the following system of equations:

$$\begin{aligned} \mathbf{E}\left((\mathrm{prox}(c\rho))'(\hat{z}_\epsilon)\right) &= 1-\kappa\ , \\ \kappa r_\rho^2(\kappa;G) &= \mathbf{E}\left([\hat{z}_\epsilon - \mathrm{prox}(c\rho)(\hat{z}_\epsilon)]^2\right)\ . \end{aligned} \tag{5}$$

In this system, $\mathrm{prox}(c\rho)$ refers to Moreau's proximal mapping of the convex function $c\rho$ (see Moreau, 1965; Hiriart-Urruty and Lemaréchal, 2001).

It is therefore not entirely trivial to characterize those distributions Γ for which $r_\rho^2(\kappa;G) = r_\rho^2(\kappa;\Gamma)$. In the following theorem, however, we show that as $\kappa \to 1$, $r_\rho^2(\kappa;G)$ converges to a constant that depends only on σ_ϵ^2. This implies that when $\kappa \to 1$, two different error distributions that have the same variances will result in estimators $\widehat{\beta}_{1,\rho}$ with the same variance. Before stating our theorem formally, however, we will review the necessary assumptions for the system of equations in (5) to hold. For a precise statement of the assumptions, see El Karoui (2017)

Assumptions for Equation 5: The proof of (5) provided in El Karoui (2013) assumes that the X_i's have mean 0, $\mathrm{cov}\,(X_i) = \mathrm{Id}_p$, and they satisfy sub-Gaussian concentration assumptions (with constants dependent on n). El Karoui (2013) further assumes that the ϵ_i have a unimodal density, are independent from the X_{ij}, $\sup_{1\leq i\leq n}|\epsilon_i| = \mathrm{O}_P(\mathrm{polyLog}(n))$, and that similiar bounds also hold for a few moments of ϵ_i (the number of such moments depends on the loss function ρ). Log-concave densities such as those corresponding to double exponential or Gaussian errors used in the current paper fall within the scope of this theorem. The reader interested in generalizations and truly heavy-tailed situation is referred to El Karoui (2017) and references therein.

The loss function ρ is assumed by El Karoui (2013) (in the unpenalized case) to be non-negative, twice differentiable, strongly convex, non-linear, taking value 0 at 0, and with a derivative that grows at most polynomially at infinity and a second derivative that is locally Lipschitz, with local Lipschitz constant that grow at most polynomially at infinity. It should be noted that distributions with sufficiently many moments, the condition of strong convexity of ρ can be obtained by adding $\delta x^2/2$ to the initial ρ, with δ "small", e.g., $\delta = 10^{-10}$, and that modification will change very little or anything to the estimator. Furthermore, the requirement of strong convexity of ρ, though superficially limiting, is likely an artifact of the proof, where the main motivation was log-concave distributions with an eye towards optimality (Bean et al., 2013). In fact, the theoretical predictions of (5) were verified numerically in El Karoui et al. (2011) outside of the assumptions stated above, and the predictions of Equation (5) were found to be very accurate in simulations even for non-smoothed ℓ_1 and Huber losses with certain error distributions.

We now state the theorem formally; see AppendixE for the proof of this statement.

Theorem 1 *Suppose we are working with robust regression estimators with loss ρ, and $p/n \to \kappa$. Suppose that $r_\rho^2(\kappa;G)$ is characterized by the system of equations in* (5)*. Then,*

$$\lim_{\kappa\to 1}\frac{1-\kappa}{\sigma_\epsilon^2}r_\rho^2(\kappa;G) = 1\ ,$$

provided ρ is additionally differentiable near 0 and $\rho'(x) \sim x$ near 0.

Implications for the Bootstrap For the purposes of the residual-bootstrap, Theorem 1 implies that different methods of estimating the residual distribution $\hat{G}$ will result in similar residual-bootstrap confidence intervals as $p/n \to 1$, if $\hat{G}$ has the same variance. This agrees with our simulations, where both of our proposed bootstrap strategies set the variance of $\hat{G}$ equal to $\widehat{\sigma}^2_{\epsilon,LS}$ and both had similar performance in our simulations for large p/n. Furthermore, as we noted, for p/n closer to 1, they both had similar performance to a bootstrap procedure that simply sets $\hat{G} = \mathcal{N}(0, \widehat{\sigma}^2_{\epsilon,LS})$ (Figure 2) (see also AppendixA.3 for further discussion of residual bootstrap methods which draw from the "wrong" distribution, i.e., forms of wild bootstraps (Wu, 1986)).

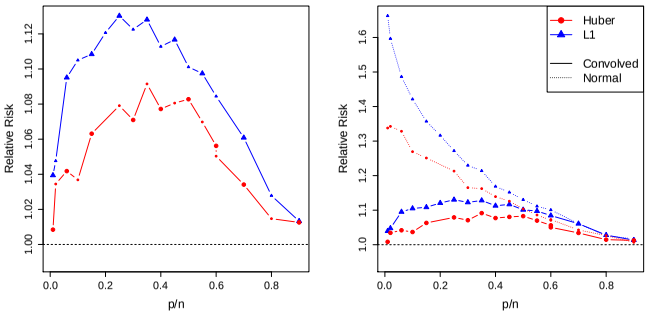
We return specifically to the bootstrap based on $\tilde{r}_{i(i)}$, the standardized predicted errors. Equation (3) tells us that the marginal distribution of $\tilde{e}_{i(i)}$ is a convolution of the distribution of ϵ_i and a normal, with the variance of the normal governed by the term $\|\widehat{\beta}_\rho - \beta\|_2$. Theorem 1 makes rigorous our previous assertion that as $p/n \to 1$, the normal term will dominate and the marginal distribution of $\tilde{e}_{i(i)}$ will approach normality, regardless of the distribution of ϵ. However, Theorem 1 also implies that as $p/n \to 1$, inference for the coordinates of β will be increasingly less reliant on features of the error distribution beyond the variance, implying that our standardized predited errors, $\tilde{r}_{i(i)}$, will still result in an estimate $\hat{G}$ that will give accurate confidence intervals. Conversely, as $p/n \to 0$ classical theory tells us that the inference of β relies heavily on the distribution G beyond the first two moments, but in that case the distribution of $\tilde{r}_{i(i)}$ approaches the correct distribution as we explained earlier. So bootstrapping from the marginal distribution of $\tilde{r}_{i(i)}$ also makes sense when p/n is small.

For κ between these two extremes it is difficult to theoretically predict the risk of $\widehat{\beta}_\rho(\hat{G})$ when the distribution $\hat{G}$ is given by resampling from the $\tilde{r}_{i(i)}$. We turn to numerical simulations to evaluate this risk. Specifically, for $\epsilon_i \sim G$, we simulated data that is a convolution of G and a normal with variance equal to $r_\rho^2(\kappa;G)$; we then scale this simulated data to have variance σ_ϵ^2. The scaled data are the ϵ_i^* and we refer to the distribution of ϵ_i^* as the convolution distribution, denoted G_{conv}. Then, G_{conv} is the asymptotic version of the marginal distribution of the standardized predicted errors, $\tilde{r}_{i(i)}$, used in our bootstrap method proposed above.

In Figure 3 we plot for both Huber loss and L_1 loss the average risk $r_\rho(\kappa;G_{conv})$ (i.e., errors given by G_{conv}) relative to the average risk $r_\rho(\kappa;G)$ (i.e., errors distributed according to G), where G has a double exponential distribution. We also plot the relative average risk $r_\rho(\kappa;G_{norm})$, where $G_{norm} = N(0,\sigma_\epsilon^2)$. As predicted by Theorem 1, for κ close to 1, $r_\rho(\kappa;G_{conv})/r_\rho(\kappa;G)$ and $r_\rho(\kappa;G_{norm})/r_\rho(\kappa;G)$ converge to 1. Conversely, as $\kappa \to 0$, $r_\rho(\kappa;G_{norm})/r_\rho(\kappa;G)$ diverges dramatically from 1, while $r_\rho(\kappa;G_{conv})/r_\rho(\kappa;G)$ approaches 1, as expected. For Huber, the divergence of $r_\rho(\kappa;G_{conv})/r_\rho(\kappa;G)$ from 1 is at most 8%, but the difference is larger for L_1 (12%), probably due to the fact that the convolution with a normal error has a larger effect on the risk for L_1.

3. Pairs Bootstrap

As described above, estimating the distribution $\hat{F}$ from the empirical distribution of (y_i, X_i) (*pairs bootstrapping*) is generally considered the most general and widely applicable method of bootstrapping, allowing for the linear model to be incorrectly specified (i.e., $\mathbf{E}\,(y_i)$ is not a linear function of X_i). It is also considered to be slightly more conservative compared



(a) Relative risk of G_{conv} to G (b) Relative risk of G_{conv} and G_{norm} to G

Figure 3: **Relative Risk of $\widehat{\beta}$ for scaled predicted errors vs original errors - population version:** (a) Plotted with a solid lines are the ratios of the average risk of $\widehat{\beta}(G_{conv})$ to the average risk of $\widehat{\beta}(G)$ for Huber and L_1 loss. (b) shows the same plot, but added to the plot (dotted lines) is the relative risk of $\widehat{\beta}(G)$ when the errors are distributed $G_{norm} = \mathcal{N}(0, \sigma_\epsilon^2)$. For both figures, the y-axis gives the relative risk, and the x-axis is the ratio p/n, with n fixed at 500. Blue/triangle plotting symbols indicate L_1 loss; red/circle plotting symbols indicate Huber loss. The average risk is calculated over 500 simulations, where the design matrix X has Gaussian entries. The "true" error distribution G is the standard Laplacian distribution with $\sigma_\epsilon^2 = 2$. Each simulation uses the standard estimate of σ_ϵ^2 from the generated ϵ_i's. $r_\rho(\kappa; G)$ was computed using a first run of simulations with $\epsilon_i \overset{iid}{\backsim} G$. The Huber loss in this plot is Huber_1 and not the default $\text{Huber}_{1.345}$ of the `rlm` function.

to bootstrapping from the residuals. In the case of random design, it makes also a lot of intuitive sense to use the pairs bootstrap, since resampling the predictors might be interpreted as mimicking the data generating process.

However, as in residual bootstrap, it is clear that the pairs bootstrap will have problems, at least in quite high dimensions. In fact, when resampling the X_i's from $\hat{F}$, the number of times a certain vector X_{i_0} is picked has asymptotically Poisson(1) distribution. So the expected number of different vectors appearing in the bootstrapped design matrix X^* is $n(1 - 1/e)$. When p/n is large, with increasingly high probability the bootstrapped design matrix X^* will no longer be of full rank. For example, if $p/n > (1 - 1/e) \approx 0.63$ then with probability tending to one as $n \to \infty$, the bootstrapped design matrix X^* is singular, even when the original design matrix X is of rank $p < n$. Bootstrapping the pairs in that situation makes little statistical sense (El Karoui, 2010, Subsection 4.4; Zheng et al., 2014).

For smaller ratios of p/n, we evaluate the performance of pairs bootstrapping on simulated data. We see that the performance of the bootstrap for inference also declines

dramatically as the dimension increases, becoming increasingly conservative (Figure 1). In pairs bootstrapping, the error rates of 95%-confidence-intervals drop far below the nominal 5%, and are essentially zero for the ratio of $p/n = 0.5$. Like residual bootstrap, this overall trend is seen for all the settings we simulated under (Supplemental Figures A-1, A-2). For L_1 loss, even ratios as small as 0.1 yield incredibly conservative bootstrap confidence intervals for $\widehat{\beta}_1$, with the error rate dropping to less than 0.01. For Huber and L_2 losses, the severe loss of power in our simulations starts for ratios of 0.3.

A minimal requirement for the distribution of the bootstrapped data to give reasonable inferences is that the variance of the bootstrap estimator $\widehat{\beta}_1^*$ needs to be a good estimate of the variance of $\widehat{\beta}_1$. This is not the case in high-dimensions. In Figure 5 we plot the ratio of the variance of $\widehat{\beta}_1^*$ to the variance of $\widehat{\beta}_1$ evaluated over simulations. We see that for $p/n = 0.3$ and design matrices X with i.i.d. $\mathcal{N}(0, 1)$ entries, the average variance of $\widehat{\beta}_1^*$ roughly overestimates the true variance of $\widehat{\beta}_1$ by a factor 1.3 in the case of least-squares; for Huber and L_1 the bootstrap estimate of variance is roughly twice as large as it should be.

In the case of least-squares, we can further quantify this loss in power by comparing the size of the bootstrap confidence intervals to the size of the correct confidence interval based on theoretical results (Figure 4). We see that even for ratios κ as small as 0.1, the confidence intervals for some design matrices X were 15% larger for pairs bootstrap than the correct size (e.g., the case of elliptical distributions where λ_i is exponential). For much higher dimensions of $\kappa = 0.5$, the simple case of i.i.d. normal entries for the design matrix gives intervals that are 80% larger than needed; for the elliptical distributions we simulated, the width of the bootstrap confidence interval was as much as 3.5 times larger than that of the correct confidence interval. Furthermore, as we can see in Figure 1, least-squares regression represents the best case scenario; L_1 and Huber will have even worse loss of power and at smaller values of κ.

3.1 Theoretical Analysis for Least-Squares

In the setting of least-squares, we can for some distributions of the design matrix X theoretically determine the asymptotic expectation of the variance of $v'\widehat{\beta}^*$ and show that it is a severe over-estimate of the true variance of $v'\widehat{\beta}$.

We first setup some notation for the theorem that follows. Define $\widehat{\beta}_w$ as the result of regressing y on X with random weight w_i for each observation (y_i, X_i). In other words,

$$\widehat{\beta}_w = \text{argmin}_{u \in \mathbb{R}^p} \sum_{i=1}^{n} w_i (y_i - X_i' u)^2 \, .$$

We assume that the weights are independent of $\{y_i, X_i\}_{i=1}^n$ and define $\widehat{\beta}_w^*$ to be the random variable with distribution equal to that of $\widehat{\beta}_w$ conditional on the data $\{y_i, X_i\}_{i=1}^n$, i.e., $\widehat{\beta}_w^* \overset{\mathcal{L}}{=} \widehat{\beta}_w | \{y_i, X_i\}_{i=1}^n$. For the standard pairs bootstrap, the distribution of $\widehat{\beta}^*$ from resampling from the pairs (y_i, X_i) is equivalent to the distribution of $\widehat{\beta}_w^*$, where w is drawn from a multinomial distribution with expectation $1/n$ for each entry. In which case, the variance of $v'\widehat{\beta}_w^*$ refers to the standard bootstrap estimate of variance given by the distribution of $v'\widehat{\beta}^*$ over repeated resampling from the pairs (y_i, X_i).

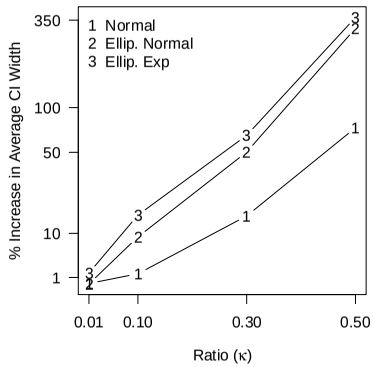


Figure 4: **Comparison of width of 95% confidence intervals of β_1 for L_2 loss:** Here we demonstrate the increase in the width of the confidence interval due to pairs bootstrapping. Shown on the y-axis is the percent increase of the average confidence interval width based on simulation ($n = 500$), as compared to the average for the standard confidence interval based on normal theory in L_2; the percent increase is plotted against the ratio $\kappa = p/n$ (x-axis). Shown are three different choices in simulating the entries of the design matrix X: (1) $X_{ij} \sim N(0,1)$ (2) elliptical X_{ij} with $\lambda_i \sim N(0,1)$ and (3) elliptical X_{ij} with $\lambda_i \sim Exp(\sqrt{2})$. The methods of simulation are the same as described in Figure 1; exact values are given in Table A-2 in AppendixI.

We have the following result for the expected value of the bootstrap variance of any contrast $v'\widehat{\beta}^*_w$ where v is deterministic, assuming independent weights with a Gaussian design matrix X and some mild conditions on the distribution of the w's.

Theorem 2 *Let the weights $(w_i)_{i=1}^n$ be i.i.d. and without loss of generality that $\mathbf{E}(w_i) = 1$; we suppose that the w_i's have 8 moments and for all i, $w_i > \eta > 0$. Suppose X_i's are i.i.d. $\mathcal{N}(0,\Sigma)$, Σ is positive definite and the vector v is deterministic with $\|v\|_2 = 1$.*

Suppose $\widehat{\beta}$ is obtained by solving a least-squares problem and $y_i = X_i'\beta + \epsilon_i$, ϵ_i's being i.i.d. mean 0, with $\mathrm{var}(\epsilon_i) = \sigma_\epsilon^2$.

If $\lim p/n = \kappa < 1$ then the expected variance of the bootstrap estimator, asymptotically as $n \to \infty$, is given by

$$p\frac{\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_w\right)\right)}{v'\Sigma^{-1}v} = p\frac{\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}_w|\{y_i,X_i\}_{i=1}^n\right)\right)}{v'\Sigma^{-1}v} \to \sigma_\epsilon^2\left[\frac{\kappa}{1-\kappa-f(\kappa)} - \frac{1}{1-\kappa}\right],$$

where $f(\kappa) = \mathbf{E}\left(\frac{1}{(1+cw_i)^2}\right)$ and c is the unique solution of $\mathbf{E}\left(\frac{1}{1+cw_i}\right) = 1-\kappa$.

We note that $\mathbf{E}\left(\frac{1}{(1+cw_i)^2}\right) \geq \left[\mathbf{E}\left(\frac{1}{1+cw_i}\right)\right]^2 = (1-\kappa)^2$ - where the first inequality comes from Jensen's inequality, and therefore the expression we give for the asymptotic limit of the expected bootstrap variance is non-negative. For a proof of this theorem and a consistent estimator of this limit, see AppendixF.

In light of previous work on model robustness issues in high-dimensional statistics (see e.g., (Diaconis and Freedman, 1984; Hall et al., 2005; El Karoui, 2009, 2010)), it is natural to ask whether the central results of Theorem 2 still apply when X_i is not Gaussian but has an elliptical distribution. The formula in Theorem 2 does not apply directly to this latter case. However, the proof given in AppendixF extends to that setting, and we refer the interested reader to the AppendixF.1 where we give the necessary details of how to change the formulas and proof to encompass the elliptical case (we do not provide them in rigorous mathematical detail in this work as they are substantially more cumbersome than those in Theorem 2 and do not give enough additional insights to justify inclusion). On the other hand, a number of the quantities appearing in the proof of Theorem 2 will converge to the same limit as that given in Theorem 2 when i.i.d. Gaussian predictors are replaced by i.i.d. predictors with mean 0 and variance 1 and sufficiently many moments (an example being bounded random variables). Thus the results we present here should be fairly robust to changing i.i.d. normality assumptions for the entries of the design matrix X, but again the technical work necessary for making this rigorous is beyond the scope of this paper.

Implications for Pairs Bootstrap In the standard pairs bootstrap, the weights are chosen according to a Multinomial$(n, 1/n)$ distribution. This violates two conditions in the previous theorem: independence of w_i's and the condition $w_i > 0$. In AppendixF.2, we give the technical details for how to extend the proof of Theorem 2 to multinomial weights. In what follows, however, we use i.i.d. Poisson(1) weights, which asymptotically and marginally correspond to the Multinomial$(n, 1/n)$ weights, to develop intuition about the bootstrap. In this case, we can apply the formula in Theorem 2 to explain why pairs bootstrap confidence intervals perform poorly in high-dimensions, at least for least squares regression with Gaussian design matrix.

When $X_i \overset{iid}{\backsim} \mathcal{N}(0,\Sigma)$, it is well known in the least-squares case that the quantity $p \text{ var}\left(v'\widehat{\beta}\right)/v'\Sigma^{-1}v$ converges asymptotically to $\kappa/(1-\kappa)\sigma_\epsilon^2$ (this can be shown through simple Wishart computations Haff, 1979; Mardia et al., 1979). If the variance of $v'\widehat{\beta}_w^*$ converged to the variance of $v'\widehat{\beta}$, we should be able to equate this latter quantity to the limit given in Theorem 2, i.e.,

$$\left[\frac{\kappa}{1-\kappa-f(\kappa)}-\frac{1}{1-\kappa}\right]=\frac{\kappa}{1-\kappa}\,,$$

and hence should have

$$f(\kappa)=\mathbf{E}\left(\frac{1}{(1+cw_i)^2}\right)=\frac{1-\kappa}{1+\kappa}\,.$$

However, this relationship does not hold for most weight distributions, and in particular does not hold for weights following a Poisson(1) distribution (which asymptotically corresponds to the standard pairs bootstrap, as explained above). Thus the pairs bootstrap does not correctly estimate the variance of $v'\widehat{\beta}$. In Figure 5a we calculate the theoretical predictions of $\mathbf{E}\left(\text{var}\left(\widehat{\beta}_w^*\right)\right)$ given by Theorem 2 (using Poisson(1) weights and $\Sigma=\text{Id}_p$), and we compare them to the asymptotic variance of $\widehat{\beta}_1$ given by $\kappa/(1-\kappa)\sigma_\epsilon^2/p$. We see that Theorem 2 predicts that the pairs bootstrap overestimates the variance of the estimator by a factor that ranges from 1.2 to 3 as κ varies between 0.3 and 0.5. These theoretical predictions correspond to the level of overestimation of the variance seen in our bootstrap simulations (Figure 5b).

3.2 Alternative Weight Distributions for Resampling

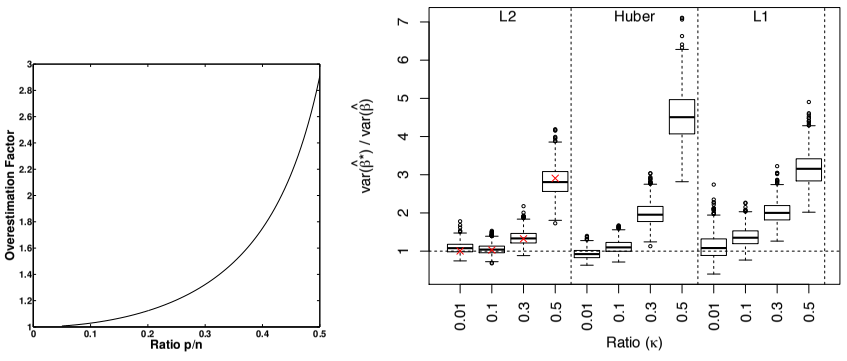
The formula given in Theorem 2 suggests that resampling from a distribution $\hat{F}$ defined using weights other than i.i.d. Poisson(1) (or, equivalently for our asymptotics, Multinominal(n,1/n)) should give us better bootstrap estimators than using the standard pairs bootstrap. In fact, we should require, at least, that the bootstrap expected variance of these estimators match the correct variance $\text{var}\left(v'\widehat{\beta}\right)=\kappa/(1-\kappa)\sigma_\epsilon^2/p$ (for the Gaussian design, when $\Sigma=\text{Id}_p$). We focus our discussion on the case $\Sigma=\text{Id}_p$; see AppendixC for the case $\Sigma\neq\text{Id}_p$.

We note that if we use $w_i=1,\ \forall i$, the bootstrap variance will be 0, since with such a resampling scheme the resampled data set is always the original data set. On the other hand, we have seen that with $w_i\sim\text{Poisson}(1)$, the expected bootstrap variance was too large compared to $\kappa/(1-\kappa)\sigma_\epsilon^2/p$. Hence, we tried to find alternative weights via calculating a parameter α such that if

$$w_i \overset{iid}{\backsim} 1-\alpha+\alpha\text{Poisson}(1)\,, \tag{6}$$

the expected bootstrap variance would match the theoretical value of $\kappa/(1-\kappa)\sigma_\epsilon^2/p$.

We numerically solved this problem to find $\alpha(\kappa)$ (for details of computation see AppendixC). We then used these values and performed bootstrap resampling using the weights defined in Equation (6). We evaluated bootstrap estimate of $\text{var}\left(\widehat{\beta}_1\right)$ as well as the confidence interval coverage of the true β_1. We find that this adjustment of the weights in



(a) L_2 (Theoretical) (b) All (Simulated)

Figure 5: **Factor by which standard pairs bootstrap over-estimates the variance:** (a) plotted is the ratio of the value of the expected bootstrap variance computed from Theorem 2 using Poisson(1) weights to the asymptotic variance $\kappa/(1-\kappa)\sigma_\epsilon^2$. (b) boxplots of the ratio of the bootstrap variance of $\widehat{\beta}_1^*$ to the variance $\widehat{\beta}_1$, as calculated over 1000 simulations (i.e., $\text{var}\left(\widehat{\beta}\right)$ is estimated across simulated design matrices X, and not conditional on X). The theoretical prediction for the mean of the distribution from Theorem 2 is marked with a 'X' for L_2 regression. Simulations were performed with normal design matrix X and normal error ϵ_i with values of $n=500$. For the median values of each boxplot, see Table A-6 in AppendixI.

estimating $\hat{F}$ results in accurate bootstrap estimates of variance and appropriate levels of confidence interval coverage (Table 1).

However, small changes in the choice of α can result in fairly large changes in $\mathbf{E}\left(\text{var}\left(v'\widehat{\beta}_w|X,\epsilon\right)\right)$. For instance, for $\kappa=0.5$, using the value of $\alpha=0.95$ which is close to the correct value of $\alpha(0.5)=0.92$ results in an expected bootstrap variance roughly 30% larger than it should be.

Moreover, this strategy for finding a good weight distribution requires knowing a great deal about the distribution of the design matrix. Hence the work we just presented on finding new weight distributions for bootstrapping is a proof of principle that alternative weighting schemes could be used for pairs bootstrapping in high-dimension, but important practical details would depend strongly on the statistical model that is assumed. This is in sharp contrast with the low-dimensional situation, where a unique and model-free bootstrap resampling technique works in a broad variety of situations.

	κ			
	.1	.2	.3	.5
α	.9875	.9688	.9426	.9203
Error Rate of 95% CIs	0.051	0.06	0.061	0.057
Ratio of Variances	1.0119	1.0236	0.9931	0.9992

Table 1: **Summary of weight-adjusted bootstrap simulations for L_2 :** Given are the results of performing bootstrap resampling for $n = 500$ according to the estimate of $\hat{F}$ given by the weights in Equation (6). "Error Rate of 95% CIs" denotes the percent of bootstrap confidence intervals that did not contain the correct value of the parameter β_1. "Ratio of Variances" gives the ratio of the empirical expected bootstrap variance over our simulations divided by the theoretical value $\sigma_\epsilon^2\kappa/(1-\kappa)$. Results are based on 1000 simulations, with a Gaussian random design and errors distributed as double exponential.

4. The Jackknife

In the context we are investigating, where we know that the distribution of $\widehat{\beta}_1$ is asymptotically normal, it is natural to ask whether we could simply use the jackknife to estimate the variance of $\widehat{\beta}_1$. The jackknife relies on leave-one-out procedures to estimate var $\left(\widehat{\beta}_1\right)$. More specifically, for a fixed vector v, the jackknife estimate of var $\left(v'\widehat{\beta}\right)$ is given by:

$$\widehat{var}_{JACK}(v'\widehat{\beta}) = \frac{n-1}{n}\sum_{i=1}^{n}(v'[\widehat{\beta}_{(i)} - \tilde{\beta}])^2 \tag{7}$$

where $\tilde{\beta} = \frac{1}{n}\sum_{i=1}^{n}\widehat{\beta}_{(i)}$. The case of $\widehat{\beta}_1$ corresponds to picking $v = e_1$, i.e., the first canonical basis vector. The Efron-Stein inequality guarantees in general that the expectation of the jackknife estimate of variance gives an upper-bound on the variance of the statistic under consideration (Efron and Stein, 1981).

Given the problems we just documented with the pairs bootstrap, it is natural to ask whether confidence intervals based on the jackknife estimate of variance perform better than pairs bootstrap intervals in high-dimensions. The jackknife is known to have problems (Efron, 1982 or Koenker, 2005, p.105), but the reliance of the jackknife on leave-one-out estimates $\widehat{\beta}_{(i)}$ might suggest it could be more robust to dimensionality issues than other methods.

Empirical findings As in the pairs bootstrap case, simulations show that confidence intervals based on the jackknife estimate of variance lead to extremely poor inference for β_1 (Figure 1) and that the jackknife dramatically overestimates the variance of $\widehat{\beta}_1$ (Figure 6). For L_2 and Huber loss, the jackknife estimate of variance is 10-15% too large for $p/n = 0.1$, and for $p/n = 0.5$ the jackknife estimate of variance is 2-2.5 times larger than it should be. In the case of L_1 loss, the jackknife variance is completely erratic, even in low dimensions; this is not completely surprising given the known problems with the jackknife for the median (Koenker,

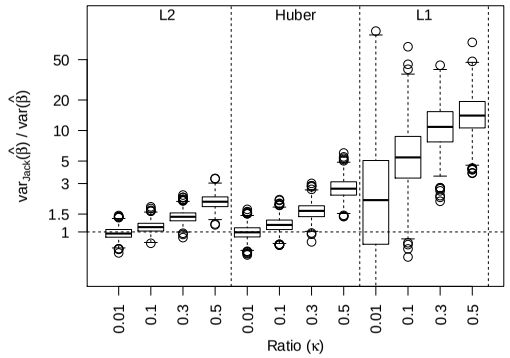


Figure 6: **Factor by which jackknife over-estimates the variance:** boxplots of the ratio of the jackknife estimate of the variance $\widehat{\beta}_1$ to the variance of $\widehat{\beta}_1$ as calculated over 1000 simulations. Simulations were with normal design matrix X and normal error ϵ_i with values of $n = 500$. Note that because the L_1 jackknife estimates so wildly overestimate the variance, in order to put all the methods on the same plot the boxplot of ratio is on log-scale; y-axis labels give the corresponding ratio to which the log values correspond. For the median values of each boxplot, see Table A-6 in AppendixI.

2005). Even for $p/n = 0.01$, the estimate is not unbiased for L_1, with median estimates twice as large as they should be and enormous variance in the estimates of variance. Higher dimensions only worsen the behavior with jackknife estimates being 15 times larger than they should.

4.1 Theoretical Results

Again, in the case of least-squares regression with a Gaussian design matrix, we can theoretically evaluate the behavior of the jackknife. The proof of the following theorem is given in AppendixG (when the observations have covariance Id) and in AppendixH (to show how to extend the results to general covariance).

Theorem 3 *Let us call var_{JACK} the jackknife estimate of variance of $v'\widehat{\beta}$ given in* (7)*, where v is any deterministic vector with $\|v\|_2 = 1$. Suppose the design matrix X is such that $X_i \overset{iid}{\backsim} \mathcal{N}(0,\Sigma)$, $\widehat{\beta}$ is computed using least-squares, and the errors ϵ have a variance. Then we have, as $n, p \to \infty$ and $p/n \to \kappa < 1$,*

$$\frac{\mathbf{E}\left(var_{JACK}\right)}{\text{var}\left(v'\widehat{\beta}\right)} \to \frac{1}{1-\kappa}\,.$$

As in Theorem 2, the proof of Theorem 3 is based on random matrix techniques where further technical work should allow an extension for the entries of $X_{i,j}$ to be i.i.d. from a distribution other than Gaussian, provided $X_{i,j}$'s have sufficiently many moments. This is also beyond the scope of our work, but interested readers can see AppendixG.3 for more details.

Correcting the Jackknife in Least Squares Theorem 3 implies that scaling the jackknife estimate of variance by multiplying it by $1-p/n$ will result in an estimate of var $\left(\widehat{\beta}_1\right)$ with the correct expectation; simulations shown in Figure 7 confirm that confidence intervals based on this corrected estimate of variance yield correct confidence intervals for least-squares estimates of $\widehat{\beta}$ when the design matrix X is Gaussian. However this scaling factor is not robust to violations of these assumptions. In particular when the X matrix follows an elliptical distribution the correction of $1-p/n$ from Theorem 3 gives little improvement even when the loss is still L_2 (Figure 7).

Corrections for more general settings For the more general setting of an elliptical design matrix X and loss function ρ, preliminary computations suggest an alternative result. Let $\mathcal{S}$ be the random matrix defined by

$$\mathcal{S} = \frac{1}{n}\sum_{i=1}^{n} \psi'(e_i) X_i X_i'.$$

Then in our asymptotic regime, and when $\Sigma = \mathrm{Id}_p$, preliminary heuristic calculations suggest that we can estimate the amount by which $\mathbf{E}\left(var_{JACK}\right)$ overestimates the variance of $\widehat{\beta}_1$ by $\mathbf{E}\left(\hat{\gamma}\right)$, where

$$\hat{\gamma} \triangleq \frac{\mathrm{trace}\left(\mathcal{S}^{-2}\right)/p}{\left[\mathrm{trace}\left(\mathcal{S}^{-1}\right)/p\right]^2}\,. \tag{8}$$

Note that when applied to least-squares regression with $X \sim \mathcal{N}(0, Id_p)$ this conforms to our result in Theorem 3. Theoretical considerations suggest that in our asymptotics, for smooth ρ, $\hat{\gamma} \simeq \mathbf{E}\left(\hat{\gamma}\right)$, which suggests a data-driven correction to the jackknife estimate of variance; however that correction depends having information about the distribution of the design matrix.

Equation (8) assumes that the loss function can be twice differentiated, which is not the case for either Huber or L_1 loss. In the case of non-differentiable ρ and ψ, we can use appropriate regularizations to make sense of those functions. For $\rho = \mathrm{Huber}_k$, i.e., a Huber function that transitions from quadratic to linear at $|x| = k$, ψ' should be understood as $\psi'(x) = 1_{|x|\leq k}$. For L_1 loss, ψ' should be understood as $\psi'(x) = 1_{x=0}$.

In Figure 7 we show simulation results for confidence intervals created based on rescaling the jackknife estimate of variance by $\mathbf{E}\left(\hat{\gamma}\right)$ defined in Equation (8). In the case of least-squares with an elliptical design matrix, this correction—which directly uses the distribution of the observed X matrix—leads to a definite improvement in our jackknife confidence intervals. Similarly, for the Huber loss we see a definite improvement as compared to the standard jackknife estimate, as well as an improvement over the simpler correction of $1-p/n$ that would be appropriate for squared error loss.

It should be noted that the quality of this proposed correction seems to depend on how smooth is the function ψ. In particular, even using the previous interpretations, the

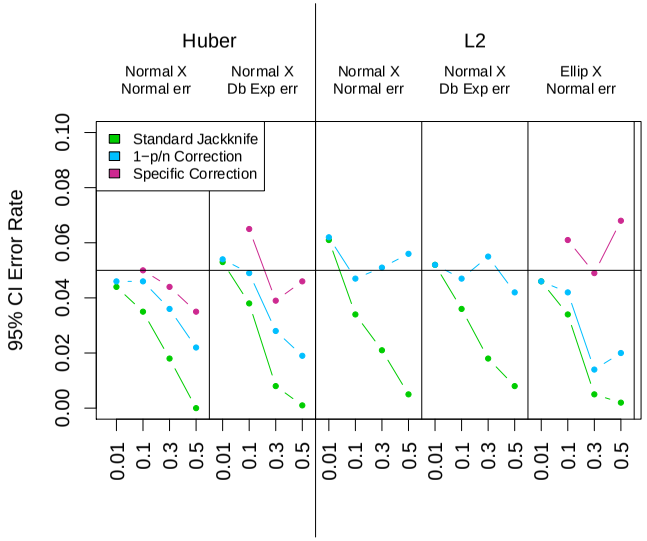


Figure 7: **Rescaling jackknife estimate of variance:** Shown are the error rates for confidence intervals for different re-scalings of the jackknife estimate of variance: the standard jackknife estimate (green); re-scaling using $1-p/n$ as given in Theorem 3 for the L_2 case with normal design matrix X (blue); and re-scaling based on the heuristic in Equation (8) for those settings not covered by the assumptions of Theorem 3 (magenta). The Huber loss in this plot is Huber_1 rather than the default $\mathrm{Huber}_{1.345}$; Huber_1 is further from L_2 than $\mathrm{Huber}_{1.345}$ and therefore better shows the improvement gained by using the heuristic in Equation (8).

correction does not perform well for L_1 (at least for $n = 1000$ and $\kappa = 0.1, 0.3, 0.5$, data not shown) - though as we mentioned Figure 6 shows that jackknifing in L_1-regression is probably not a good idea; see also Koenker (2005, Section 3.9).

We also note that the assumption of cov $(X_i) = \mathrm{Id}_p$ is essential to the jackknife correction proposed in Equation (8). Let $\widehat{\beta}_\rho(\Sigma)$ denotes our estimator of β when cov $(X_i) = \Sigma$. $\hat{\gamma}$ in equation (8) is an estimate of

$$R(\widehat{\beta}_\rho) = \frac{\mathbf{E}\left(var_{JACK}(v'\widehat{\beta}_\rho(\mathrm{Id}_p))\right)}{\mathrm{var}\left(v'\widehat{\beta}_\rho(\mathrm{Id}_p)\right)}.$$

Invariance arguments applied to cov (X_i) can be used to show that when X_i has an elliptical distribution, then $R(\widehat{\beta}_\rho(\mathrm{Id}_p)) = R(\widehat{\beta}_\rho(\Sigma))$ for all convex loss functions ρ (see AppendixH).

However, even though this population quantity is unchanged when $\mathrm{cov}\,(X_i) = \Sigma$ changes, the estimate $\hat{\gamma}$ we give above depends crucially on knowing that $\mathrm{cov}\,(X_i) = \mathrm{Id}_p$ and cannot be used as-is when $\Sigma \neq \mathrm{Id}_p$.

5. Conclusion

In this paper, we have studied various resampling plans in the high-dimensional setting where p/n is not close to zero. One of our main findings is that the two most widely-used and advocated bootstraps will yield either highly conservative or highly anti-conservative confidence intervals. This is in sharp contrast to the low-dimensional setting where p is fixed and $n \to \infty$ or $p/n \to 0$. Under various assumptions underlying our simulations, we explained theoretically the phenomena we were observing in our numerical work.

Beside our theoretical contributions, we propose improved and dimension-adaptive bootstrap methods for both pairs and residual bootstraps, as well as jackknife corrections, that give confidence intervals with approximately correct coverage probability. These methods are novel dimensionality-robust resampling techniques for linear models. The resulting modified pairs bootstrap gives a principled method for pairs bootstrapping regardless of the value of $p/n < 1$, and avoids the problem of non-invertible bootstrapped design matrices X^* that commonly result from the standard pairs bootstrap. The most promising of our proposed resampling schemes is our proposed residual bootstrapping method that resamples from appropriately scaled predicted errors. This bootstrap routine performed well without distribution-specific corrections that some of our other methods require. It has the greatest potential to be a general-purpose bootstrap method for linear models in high dimensions.

This work has focused on estimation of the linear model, where there are theoretical benchmarks. The real practical power of the bootstrap lays in giving the ability to perform inference in complicated settings involving sophisticated statistical procedures for which we do not even begin to have theoretical results for the behavior of our estimators. Yet our work shows that even for the simple case of inference in the linear model and for the simplest inferential question, the two most common and natural resampling techniques perform very poorly in only moderately high-dimensions. More importantly, these two equally intuitive methods have completely divergent statistical behavior with one being extremely conservative and the other anti-conservative. This casts serious doubts about the reliability, interpretability and accuracy of inferential statements made through generic resampling methods in moderate and high dimensions. This is troubling for more complicated problems where resampling techniques are the only inference tools currently available. Our findings also suggest that appropriate resampling methods for high-dimensional problems may not be able to rely on generic resampling procedures but rather need to be tailored to the statistical problem of interest. This raises many interesting new theoretical and methodological questions for the future.

Acknowledgments

The authors gratefully acknowledge grants NSF DMS-1026441, NSF DMS-0847647 (CAREER), and NSF DMS-1510172 and support of the ENS-CFM Data Science Chair. They would also like to thank Peter Bickel and Jorge Banuelos for discussions. N. El Karoui would like to thank Criteo AI Lab for providing a great and very stimulating research environment.

Appendices

Notations : in these appendices, we use e_i to denote the i-th residual, i.e., $e_i = y_i - X_i'\widehat{\beta}$. We use $\tilde{e}_{i(i)}$ to denote the i-th prediction error, i.e., $\tilde{e}_{i(i)} = y_i - X_i'\widehat{\beta}_{(i)}$, where $\widehat{\beta}_{(i)}$ is the estimate of $\widehat{\beta}$ with the i-th pair (y_i, X_i) left out. We assume that the linear model holds so that $y_i = X_i'\beta + \epsilon_i$. We assume that the errors ϵ_i are i.i.d with mean 0.

Appendix A. Technical Background on Existing Literature on Robust Regression

Recall that we consider

$$\widehat{\beta}_\rho = \mathrm{argmin}_{u\in\mathbb{R}^p} \sum_{i=1}^n \rho(y_i - X_i'u) \ , \ \text{where } y_i = \epsilon_i + X_i'\beta \ .$$

The ϵ_i's are assumed to be i.i.d with mean 0 here.

A.1 Classical Results and Asymptotic Normality

Least-squares In this case $\rho(x) = x^2/2$ and we have of course

$$\widehat{\beta}_{LS} - \beta = (X'X)^{-1}X'\epsilon \ .$$

Hence,

$$\mathrm{cov}\left(\widehat{\beta}_{LS}\right) = (X'X)^{-1}\mathrm{var}\left(\epsilon\right) \ .$$

Robust regression We recall the classic result of Huber (Huber, 1973) and (Huber and Ronchetti, 2009), Chapter 7: when p is fixed and $n \to \infty$, the limiting covariance of $\widehat{\beta}_\rho$ is, with a slight abuse of notation,

$$\mathrm{cov}\left(\widehat{\beta}_\rho\right) = \frac{1}{n}\left(\frac{X'X}{n}\right)^{-1}\frac{\mathbf{E}\left(\psi^2(\epsilon)\right)}{\left[\mathbf{E}\left(\psi'(\epsilon)\right)\right]^2} \ .$$

See also the papers Portnoy (1984, 1985, 1987); Mammen (1989) for the situation where $p \to \infty$ and $p/n \to 0$ at various rates.

Asymptotic normality questions and impact on confidence intervals: $\boldsymbol{p/n \to 0}$ In the case of least-squares, the Lindeberg-Feller theorem (Stroock, 1993) guarantees that under mild conditions on the $p \times n$ matrix X, the coordinates of $\widehat{\beta}_{LS}$ are asymptotically Normal. Similarly if the $1 \times n$ vector $v'(X'X)^{-1}X'$ satisfy the conditions of the the Lindeberg-Feller theorem, then $v'(\widehat{\beta}_{LS} - \beta)$ is asymptotically normal. Similarly, under mild conditions on X, the classic papers mentioned above guarantee asymptotic normality of the coordinates of $\widehat{\beta}_\rho$ when $p/n \to 0$. In these cases, the width of confidence intervals for the coordinates of β are hence only dependent asymptotically on the variance of the coordinates of $\widehat{\beta}_\rho$.

A.2 Summary of recent results on high-dimensional robust regression

We summarize in this section the key results we use from the recent papers El Karoui et al. (2011); El Karoui et al. (2013); El Karoui (2013, 2017). The third paper is a mathematically rigorous version of the heuristic arguments of the first two; the first paper is the long-form version of the second one. Those papers are concerned with the asymptotic properties of $\widehat{\beta}_\rho$ when $p/n \to \kappa \in (0,1)$. The predictor vectors X_i's are assumed to be random and independent, with $X_i = \lambda_i\Sigma^{1/2}\tilde{X}_i$, where $\tilde{X}_i$ has i.i.d (not necessarily Gaussian) entries with mean 0 and variance 1. λ_i's are independent random variables with $\mathbf{E}\left(\lambda_i^2\right) = 1$. (The

$n \times p$ design matrix X is full rank with probability 1. Σ has only positive eigenvalues.) X_i's are independent of ϵ_i's.

Role of $\mathrm{cov}\,(X_i) = \Sigma$ It is shown in these papers that, if $\widehat{\beta}(\beta;\Sigma)$ is the regression vector corresponding to the situation where $y_i = X_i'\beta + \epsilon_i$ and $\mathrm{cov}\,(X_i) = \Sigma$ for all i,

$$\widehat{\beta}_\rho(\beta;\Sigma) = \beta + \Sigma^{-1/2}\widehat{\beta}(0;\mathrm{Id}_p)\ .$$

This follows from a simple change of variable. It also means that to understand the properties of $\widehat{\beta}_\rho(\beta;\Sigma)$, it is enough to understand the "null case" $\beta = 0$ and $\Sigma = \mathrm{Id}_p$.

Consequence for leave-one-out-predicted errors The result we just mentioned has an important consequence for our leave-one-out predicted error, i.e $\tilde{e}_{i(i)} = y_i - X_i'\widehat{\beta}_{(i)}$: $\tilde{e}_{i(i)}(\beta;\Sigma) = \tilde{e}_{i(i)}(0;\mathrm{Id}_p)$. In other words, we can assume without loss of generality that $\beta = 0$ and $\Sigma = \mathrm{Id}_p$ when working with leave-one-out-predicted errors.

A non-asymptotic and exact stochastic representation in the elliptical case When $X_i \overset{iid}{\backsim} \lambda_i \upsilon_i$, where $\upsilon_i \sim \mathcal{N}(0,\Sigma)$ and λ_i is a random variable independent of υ_i, it is shown that

$$\widehat{\beta}_\rho(\beta;\Sigma) \overset{\mathcal{L}}{=} \beta + \|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2 \Sigma^{-1/2} u\ ,$$

where u is uniformly distributed on the unit sphere in $\mathbb{R}^p$ and $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ is independent of u. $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ is simply the norm of $\widehat{\beta}_\rho$ when $\beta = 0$ and $\mathrm{cov}\,(X_i) = \mathrm{Id}_p$. Note that u has the stochastic representation $u \overset{\mathcal{L}}{=} Z_p/\|Z_p\|_2$, where $Z_p \sim \mathcal{N}(0,\mathrm{Id}_p)$.

Consequence of the previous representation for large p Since $\|Z_p\|_2$ has χ_p distribution, it is clear that as $p \to \infty$, if v is a deterministic vector,

$$\sqrt{p}\frac{v'(\widehat{\beta}_\rho(\beta;\Sigma) - \beta)}{\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2} \Longrightarrow \mathcal{N}(0, v'\Sigma^{-1}v)\ ,$$

where $\Longrightarrow$ denotes weak convergence of distributions. Hence, provided $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ and $v'\Sigma^{-1}v$ remain bounded, $v'\widehat{\beta}_\rho(\beta;\Sigma)$ is $\sqrt{p}$-consistent for $v'\beta$.

Properties of $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ It is shown, under various technical assumptions, that as p and n tend to infinity with $p/n \to \kappa$, the variance of the random variable $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ goes to zero. Hence, for practical matters, $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ can be considered non-random. In particular, that implies that

$$\sqrt{p}v'(\widehat{\beta}_\rho(\beta;\Sigma) - \beta) \text{ is approximately Normal as } p/n \to \kappa\ .$$

Of great importance is the characterization of $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$, since it will affect the width of confidence intervals. It can be characterized, in the case where $\lambda_i = 1$ (see the papers for the case $\lambda_i \neq 1$) in the following way: $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2 \to r_\rho(\kappa)$. The non-random scalar $r_\rho(\kappa)$ can be characterized through a system of two non-linear equations, involving another constant, c. The pair of positive and deterministic scalars $(c, r_\rho(\kappa))$ satisfy: if $\hat{z}_\epsilon = \epsilon + r_\rho(\kappa)Z$, where $Z \sim \mathcal{N}(0,1)$ is independent of ϵ, and ϵ has the same distribution as ϵ_i's:

$$\left\{ \begin{array}{rl} \mathbf{E}\left((\mathrm{prox}(c\rho))'(\hat{z}_\epsilon)\right) & = 1-\kappa\ , \\ \kappa r_\rho^2(\kappa) & = \mathbf{E}\left([\hat{z}_\epsilon - \mathrm{prox}(c\rho)(\hat{z}_\epsilon)]^2\right)\ . \end{array} \right. \tag{9}$$

In this system, $\mathrm{prox}(c\rho)$ refers to Moreau's proximal mapping of the convex function $c\rho$ - see Moreau (1965) or Hiriart-Urruty and Lemaréchal (2001). (The system is rigorously shown in El Karoui (2013) under the assumption that the X_i's have i.i.d entries with mean 0 and variance 1, as well as a few other minor requirements; these assumptions are satisfied when $X_{i,j}$ have a Gaussian distribution, or are bounded, or do not have heavy tails, the latter requiring appeal to various truncation arguments. Another proof of the validity of this system, which first appeared in El Karoui et al. (2011), can be found in Donoho and Montanari (2013). That proof is limited to the case of X_i's having i.i.d Gaussian entries.) The assumptions on ϵ_i's and ρ are relatively mild. See El Karoui (2017) for the latest, handling the situation where ϵ_i's have for instance a Cauchy distribution. We note that some of the results in El Karoui (2013) are stated with ρ strongly convex (and ϵ_i's having many moments). While the proof in that paper suggests several ways of removing this assumption, it is also possible to change ρ in to $\rho + \eta x^2/2$ with η very small (e.g $\eta = 10^{-100}$) to satisfy this technical assumption and change essentially nothing to the statistical problem at hand.

Consequences for the distribution of $\widehat{\beta}_1$ *or other contrasts of interest* In our simulation setup, the previous results imply that the distribution of $\widehat{\beta}_1$ (or any other coordinates or contrasts $v'\widehat{\beta}$ for v deterministic) is asymptotically normal. In the case where $\Sigma = \mathrm{Id}_p$, the variance of $\sqrt{p}(\widehat{\beta}_1 - \beta_1)$ is roughly $\mathcal{N}(0, r_\rho^2(\kappa))$. See Bean et al. (2013) and its supplementary material for a longer discussion and questions related to building confidence intervals.

Asymptotic normality questions and impact on confidence intervals: $\boldsymbol{p/n \to \kappa \in (0,1)}$ Because we know that, in the Gaussian design case, the coordinates of $\widehat{\beta}_\rho$ are asymptotically normal, the width of these intervals is completely determined by the variance of the coordinates of $\widehat{\beta}_\rho$. We explain above how these variances depend on the distribution of ϵ and the loss function ρ: basically through $\|\widehat{\beta}(\rho;\mathrm{Id})\|_2$ and hence $r_\rho(\kappa)$. Therefore, as was the case in the low-dimensional situation, the variance of the coordinates of $\widehat{\beta}_\rho$ can be used as a proxy for the width of the confidence interval in the high-dimensional case where $p/n \to \kappa$, $0 < \kappa < 1$.

In (Bean et al., 2013), these asymptotic normality results are used to create confidence intervals for $v'\beta$ in the Gaussian design case: if $z_{1-\alpha/2}$ is the $(1-\alpha/2)$-quantile of the Gaussian distribution a $100(1-\alpha)\%$ confidence interval for $v'\beta$ is

$$v'\widehat{\beta} \pm \frac{z_{1-\alpha/2}}{\sqrt{p}}\hat{r}\sqrt{(1-p/n)v'\widehat{\Sigma}^{-1}v}\ ,$$

where $\hat{r}$ is a consistent estimator of $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$. In (Bean et al., 2013), it is said without more precision that leave-one-techniques can be used to come up with $\hat{r}$; we propose in the current paper estimates $\hat{r}$ based on leave-one-out predicted errors that can therefore be used for the purpose of building those confidence intervals. (See Section 2.3 in the main paper)

Leave-one-out approximations for $\widehat{\beta}$ It is shown in the aforementioned papers that

$$\widehat{\beta} \simeq \widehat{\beta}_{(i)} + \frac{1}{n}S_i^{-1}X_i\psi(e_i)\ ,$$

where $\simeq$ means that we are neglecting a quantity that is negligible for all our mathematical and statistical purposes (see the papers for very precise bounds on the quantity we are

neglecting). This approximation is the key to the approximations in Equations (3) and (4) which we use in the main paper. Recall that $S_i = \frac{1}{n}\sum_{j\neq i}\psi'(\tilde{e}_{j(i)})X_jX_j'$.

A.3 Consequences for the residual bootstrap

We call $\{\epsilon_i^*\}_{i=1}^n$ the estimated errors used in the residual bootstrap. When doing a residual bootstrap, we are effectively sampling from a model with fixed design X, "true β" taken to be equal to $\widehat{\beta}_\rho$ and i.i.d errors sampled according to the empirical distribution of the $\{\epsilon_i^*\}_{i=1}^n$. As a shortcut, we call this distribution ϵ^* in what follows. We call $\widehat{\beta}_\rho^*$ the bootstrapped version of $\widehat{\beta}$.

Case $\boldsymbol{p/n \to 0}$ Naturally, the classic results mentioned above imply that the distribution of $v'(\widehat{\beta}_\rho^* - \widehat{\beta}_\rho)$ is going to be asymptotically normal (under mild conditions on X that are satisfied in our simulations); the variance of the coordinates of $\widehat{\beta}_\rho^*$, on the other hand depends on $\frac{\mathbf{E}\left(\psi^2(\epsilon^*)\right)}{[\mathbf{E}(\psi'(\epsilon^*))]^2}$. Hence, even if the distribution of the estimated errors ϵ^* is very different from that of the "true" errors, ϵ, the residual bootstrap may work very well: indeed, if ϵ and ϵ^* have two very different distribution but

$$\frac{\mathbf{E}\left(\psi^2(\epsilon^*)\right)}{[\mathbf{E}\left(\psi'(\epsilon^*)\right)]^2} = \frac{\mathbf{E}\left(\psi^2(\epsilon)\right)}{[\mathbf{E}\left(\psi'(\epsilon)\right)]^2},$$

using a residual bootstrap with "the wrong error distribution", ϵ^*, will give us bootstrap confidence intervals of the right width. An important question then becomes, when p/n is small: what class of distributions ϵ^* is such that $\frac{\mathbf{E}\left(\psi^2(\epsilon^*)\right)}{[\mathbf{E}(\psi'(\epsilon^*))]^2} = \frac{\mathbf{E}\left(\psi^2(\epsilon)\right)}{[\mathbf{E}(\psi'(\epsilon))]^2}$, as this class defines all acceptable error distributions from the point of view of our residual bootstrap.

Case $\boldsymbol{p/n \to \kappa \in (0,1)}$ We note that at this point in the case $p/n \to \kappa \in (0,1)$ we are not aware of central limit theorems for the coordinates of $\widehat{\beta}$ that are valid conditional on the design matrix X. However, it is expected that such theorems will hold if the design matrix results from a draw of a random design matrix similar to the ones we consider (with very high-probability with respect to the sampling of the design matrix). The discussions above make then clear that the key quantity to describe the width of the residual bootstrap confidence intervals becomes the risk $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p;\epsilon^*)\|_2$, i.e the risk $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p)\|_2$ when the error distribution is ϵ^*. A "good" error distribution is therefore one for which $r_\rho(\kappa;\epsilon^*) \simeq r_\rho(\kappa;\epsilon)$. (We used the notation $r_\rho(\kappa;\epsilon) = \lim_{n\to\infty}\|\widehat{\beta}_\rho(0;\mathrm{Id}_p;\epsilon)\|$, when $p/n \to \kappa$.)

The case of least squares Let us call $\hat{G}_{n,p}$ the distribution of the errors we use in our residual bootstrap. We assume that $\hat{G}_{n,p}$ has mean 0. Let us call $w' = v'(X'X)^{-1}X'$ - where we choose to not index v and w by p for the sake of clarity. v is a deterministic sequence of p-dimensional vectors. Assume that w and $\hat{G}_{n,p}$ satisfy the conditions of the Linderberg-Feller theorem for triangular arrays, and that $\lim_{n\to\infty}\mathrm{var}\left(\hat{G}_{n,p}\right) = \sigma_\epsilon^2$. Then the Lindeberg-Feller theorem guarantees that

$$\frac{v'(\widehat{\beta}^* - \widehat{\beta})}{\|w\|} \Longrightarrow \mathcal{N}(0,\sigma_\epsilon^2)\ .$$

Note that it also guarantees, under the same assumptions on w that

$$\frac{v'(\widehat{\beta} - \beta)}{\|w\|} \Longrightarrow \mathcal{N}(0,\sigma_\epsilon^2)\ .$$

These results do not depend on the size of κ, the limit of the ratio p/n.

Informally, what this means is that provided that the entries of w are all relatively small, that $\hat{G}_{n,p}$ has mean 0 and var $\left(\hat{G}_{n,p}\right)$ is close to σ_ϵ^2, then bootstrapping from the residuals in least-squares works for approximating the distribution $v'(\widehat{\beta} - \beta)$.

Conclusion for the purposes of the main paper In our discussions we use $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p;\epsilon^*)\|$ and its closeness to its value under the correct error distribution, $\|\widehat{\beta}_\rho(0;\mathrm{Id}_p;\epsilon)\|$, as a proxy to understand a priori the quality of residual bootstrap confidence intervals when using ϵ^* to sample the errors instead of ϵ. The previous discussion explains why we do so. Our numerical work in Section 2.3 of the main text shows numerically that this yields valuable insights. This is why our discussion in Section 2.4 is focused on understanding $\|\widehat{\beta}(0;\mathrm{Id}_p;\epsilon)\|_2$ for various error distributions. In particular, Theorem 1 shows that when p/n is close to 1, if ϵ^* has approximately the same two first moments as ϵ, $\|\widehat{\beta}(0;\mathrm{Id}_p;\epsilon^*)\|/\|\widehat{\beta}(0;\mathrm{Id}_p;\epsilon)\| \simeq 1$. This explains why the scaled $\tilde{r}_{i(i)}$ is probably a good error distribution ϵ^* to use in the residual bootstrap when κ is close to 0 or 1. We note that when κ is close to 1, $\tilde{r}_{i(i)}$ gives an error distribution that is in general very different from the distribution of ϵ. Our numerical work of Section 2.3 shows that it is nonetheless a good error distribution from the point of view of the residual bootstraps we consider.

Appendix B. Deconvolution Bootstrap

In the main text, we considered situations where our predictors X_i are i.i.d with an elliptical distribution and assume for instance that $X_i = \lambda_i\xi_i$, where $\xi_i \sim \mathcal{N}(0,\Sigma)$ and λ_i are i.i.d scalar random variables with $\mathbf{E}\left(\lambda_i^2\right) = 1$. As described in the main text, if X is elliptical, $\tilde{e}_{i(i)}$ is a convolution of the correct G distribution and a Normal distribution,

$$\tilde{e}_{i(i)} \simeq \epsilon_i + \tilde{Z}_i,$$

where

$$\tilde{Z}_i \overset{iid}{\backsim} \mathcal{N}(0,\lambda_i^2\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2)$$

and are independent of ϵ_i.

We proposed in Section 2.3 of the main text an alternative bootstrap method based on using deconvolution techniques to estimate G (Method 1). Specifically, we proposed the following bootstrap procedure:

1. Calculate the predicted errors, $\tilde{e}_{i(i)}$
2. Estimate $|\lambda_i|\|\widehat{\beta}_{\rho(i)} - \beta\|_2$ (the standard deviation of the $\tilde{Z}_i$)
3. Deconvolve in $\tilde{e}_{i(i)}$ the error term ϵ_i from the $\tilde{Z}_i$ term ;
4. Use the resulting estimates of G as the estimate of $\hat{G}$ in residual bootstrapping.

B.1 Estimating $\|\widehat{\beta}_\rho - \beta\|$ and the Variance of the Z_i

Deconvolution methods that deconvolve ϵ from the $\tilde{Z}_i$ require an estimate of the variance of the $\tilde{Z}_i$. Equation (3) gives the variance as $\lambda_i^2\|\widehat{\beta}_{\rho(i)} - \beta\|_2^2$, and we need to estimate this quantity from the data. We use the approximation

$$\|\widehat{\beta}_{\rho(i)} - \beta\|_2 \simeq \|\widehat{\beta}_\rho - \beta\|_2.$$

See AppendixA and references therein for justification of this approximation.

Furthermore, as we note in the main text, in our implementation of this deconvolution in simulations we assume $X \sim \mathcal{N}(0, Id_p)$ so that $\lambda_i = 1$ (see Section B.5 below for estimating λ_i in the elliptical case). This means we are estimating the variance of $\tilde{Z}_i$ as $\|\widehat{\beta} - \beta\|_2^2$ for all i. We estimate this as

$$\widehat{var}(\tilde{Z}_i) = \widehat{var}(\tilde{e}_{i(i)}) - \hat{\sigma}_\epsilon^2,$$

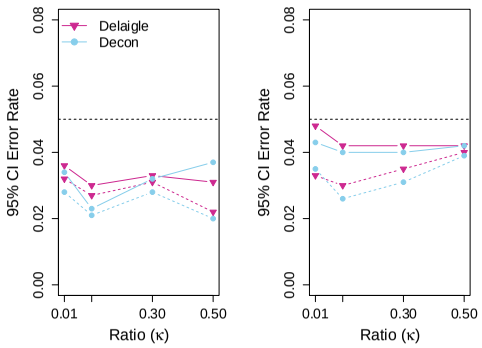
where $\widehat{var}(\tilde{e}_{i(i)})$ is the standard estimate of variance and $\hat{\sigma}_\epsilon^2$ is the estimate of variance from the least squares fit, $\widehat{\sigma}_{\epsilon,LS}^2$, defined in the main text.

In the case where $\widehat{var}(\tilde{e}_{i(i)}) \leq \hat{\sigma}_\epsilon^2$, we do not do a deconvolution, but simply bootstrap from the $\tilde{e}_{i(i)}$. This is generally only the case when p/n is quite small.

B.2 Estimating $\hat{G}$

We used the deconvolution algorithm in the `decon` package in R (Wang and Wang, 2011) to estimate the distribution of ϵ_i. Deconvolution algorithms require selection of a bandwidth in the kernels that make up the functional basis of the estimate. The appropriate bandwidth parameter in deconvolution problems is tied intrinsically to the use of the estimate, with optimal bandwidths depending on what functional of the distribution is wanted (e.g. the pdf versus the cdf). Moreover, the optimal bandwidth depends on the distribution of $\tilde{Z}_i$ with which the signal is being convolved. Ultimately, our procedure resamples from the distribution $\hat{G}$, requiring estimates of $G^{-1}(y)$, and the distribution of $\tilde{Z}_i$ is Gaussian. There is no specific theory for the optimal bandwidth in this setting (though see the work of Hall and Lahiri (2008) for optimal bandwidth selection for estimations of the quantiles of $\hat{G}$ if the $\tilde{Z}_i$ are distributed according to a distribution whose characteristic function decays polynomially at infinity - see Assumption (A.11) on p.2133; this is clearly violated in our case where $\tilde{Z}_i$ are normally distributed.)

We used the bandwidth estimation procedure `bw.dboot2` provided in the package `decon`. Delaigle (2014) outlines problems in the estimation of bandwidth parameter in `decon`; specifically that the implementation in `decon` of existing bandwidth estimation procedures does not match their published descriptions. `bw.dboot2` was not one of the bandwidth procedures with these discrepancies. However, we also compared our results with a bandwidth selected via the bandwidth selection method of Delaigle and Gijbels (2002, 2004) and used the R code implementation provided by the authors on `http://www.ms.unimelb.edu.au/~aurored/links.html#Code`. The two different choices in bandwidth, however, had little effect on the coverage of the bootstrap confidence intervals (Supplemental Figure 8). The results in Figure 2 in the main text make use of the bandwidth parameter of Delaigle and Gijbels (2002, 2004).



(a) L_1 loss (b) Huber loss

Figure 8: **Different bandwidths for Method 1:** We plotted the error rate of 95% confidence intervals for the deconvolution bootstrap (Method 1) using two different choices of bandwith: the bw.dboot2 in `decon` (light blue) or that of Delaigle and Gijbels (2002, 2004) (maroon). The solid lines refer to bootstrapping by drawing $\{\epsilon_i^*\}_{i=1}^n$ as i.i.d draws from $\hat{G}$; the dashed lines refer to $\{\epsilon_i^*\}_{i=1}^n$ drawn from repeated resampling of a single draw $(\{\hat{\epsilon}_i\}_{i=1}^n)$ from $\hat{G}$. See section B.4 below. Note that the y-axis for these plots is different than that shown in the main text.

For both bandwidth selections, we estimated the cdf using the function `DeconCdf` provided in the `decon` package and provided the bandwidth parameters described above. We specified the error distribution as 'Normal' and set the variance of $\tilde{Z}_i$ as described above in Section B.1. The number of grid points for evaluating the cdf (the 'ngrid' argument) was set to be the number needed to get a space of 0.01 across the range of observed $\tilde{e}_{i(i)}$, with a lower bound of 512 grid points (the default of 'ngrid' given by the `DeconCdf` function). Other options were set to the default of `DeconCdf`.

B.3 Random Draws from $\hat{G}$

The end result of the `DeconCdf` function was values of the $\hat{G}$ evaluated at specific grid points x. The resulting $\hat{G}(x)$ was not always guaranteed to be ≤ 1 nor monotonically decreasing; this is likely due to the fact that use of higher-order kernels estimates (which is standard practice in deconvolution literature) does not constrain the estimate be a proper density. Furthermore, the tail ends of the cdf are based on little data and unlikely to be reliable, as well as having problems either non-monotonicity or extending beyond the boundaries of $(0, 1)$. We truncated the left tail of $\hat{G}(x)$ to be within 0.001 by finding the largest such x_0 such that $\hat{G}(x_0) \leq 0.001$ and setting $\hat{G}(x) = 0.001$ for $x \leq x_0$; and we similarly trimmed the right tail based on $1 - 0.001$. We then calculated the differences $d_i = \hat{G}(x_i) - \hat{G}(x_{i-1})$

and for $d_i < 0$ set $d_i = 0$. We then defined a monotone cdf based on the cumulative sum of the d_i,

$$C(x_j) = \sum_{i=1}^{j} d_i.$$

We then renormalized the values $C(x_j)$ so that they extend from 0 to 1, giving the final monotone estimate of $\hat{G}(x_j)$ as

$$\hat{G}(x_j) = \frac{C(x_j) - \min_i C(x_i)}{\max_i C(x_i) - \min_i C(x_i)}$$

To randomly sample from $\hat{G}$, we needed to be able to evaluate $\hat{G}$ for all x. We did this by linearly interpolating between the $\hat{G}(x_j)$ values. In what follows, we consider the values $\hat{G}(x)$ based on this smoothed and monotone version of the original output of the `DeconCdf` function.

We create random draws from $\hat{G}$ by drawing random variables U_i from a $Unif(0,1)$ and calculating $E_i = \hat{G}(U_i)$. We further centered and standardized the draws E_j from $\hat{G}$ to get

$$\epsilon_j^* = (E_j - mean_i(E_j)) \frac{\widehat{\sigma}_{\epsilon,LS}}{\sqrt{\text{var}\,(E_i)}}$$

so that the resulting ϵ_j^* have mean zero and variance $\widehat{\sigma}^2_{\epsilon,LS}$. This was done because the variance of $\hat{G}$ was not guaranteed to have the correct variance, despite the fact we had to pre-specify the variance in the deconvolution call. Ensuring the correct moments of ϵ_j^* was a critical component for reasonable coverage of the bootstrap confidence intervals. When we did not standardize the results and just took the draws from E_j, the resulting bootstrap confidence intervals became more and more conservative as p/n grew. This again highlights the results of Theorem 1 – the variance of $\hat{G}$ is the most important feature of the distribution in order to have accurate confidence intervals.

B.4 Bootstrap Estimates $\widehat{\beta}^*$ from $\hat{G}$

We used $\hat{G}$ to create bootstrap errors, $\{\epsilon_i^*\}_{i=1}^n$ in two ways. For the first method we estimated $\{\epsilon_i^*\}_{i=1}^n$ as i.i.d draws from $\hat{G}$, and repeatedly drew such samples from $\hat{G}$, B times. In the second method, we drew one single estimate $\{\hat{\epsilon}_i\}_{i=1}^n$ as i.i.d draws from $\hat{G}$ and then created $\{\epsilon_i^*\}_{i=1}^n$ from resampling from the empirical distribution of the $\{\hat{\epsilon}_i\}_{i=1}^n$, and repeated this resampling from the empirical distribution of $\{\hat{\epsilon}_i\}_{i=1}^n$ B times. For both methods, we then calculated $\widehat{\beta}^*$ from the data (X_i, y_i^*) where $y_i^* = X_i'\widehat{\beta} + \epsilon_i^*$, as in the standard residual bootstrap. The first method seems to do slightly better in simulations, see Figure 8.

B.5 Estimation of λ_i^2

To extend the deconvolution bootstrapping method to the elliptical case when $p/n \to \kappa \in (0,1)$, one needs to be able to estimate λ_i, at least up to sign. In which case, one could estimate individually the variance of $\tilde{Z}_i$ and feed these individual estimates into the deconvolution method described above.

We recall a simple proposal from the paper (El Karoui, 2010) to solve this problem. Specifically, the author proposes to use

$$\widehat{\lambda_i}^2 = \frac{\|X_i^2\|_2/p}{\frac{1}{p}\text{trace}\left(\widehat{\Sigma}\right)} = \frac{\|X_i\|_2^2}{\frac{1}{n}\sum_{i=1}^n \|X_i\|_2^2},$$

where $\widehat{\Sigma} = \frac{1}{n}\sum_{i=1}^n X_i X_i'$. Under mild conditions on Σ and λ_i, it can be shown that when $n \to \infty$ and $p/n \to \kappa \in (0,\infty)$

$$\sup_{1 \le i \le n} |\lambda_i^2 - \widehat{\lambda_i^2}| \to 0 \text{ in probability.}$$

The intuition and proof are as follows. Concentration of measure arguments (Ledoux, 2001) show that if $\xi_i \sim \mathcal{N}(0,\Sigma)$, $\|\xi_i\|^2/p \simeq \text{trace}\,(\Sigma)/p$ and hence $\|X_i\|^2/p \simeq \lambda_i^2 \text{trace}\,(\Sigma)/p$. The law of large numbers and a little bit of further technical work then imply that $\frac{1}{n}\sum_{i=1}^n \|X_i\|^2/p \simeq \mathbf{E}\left(\lambda_i^2\right)\text{trace}\,(\Sigma)/p = \text{trace}\,(\Sigma)/p$.

Appendix C. Alternative Weight Distributions for Pairs Bootstrap

The formula given in Theorem 2 suggests resampling from a distribution $\hat{F}$ defined using weights other than i.i.d. Poisson(1). An acceptable weight distribution is such that the variance of the resampled estimator is equal to the variance of the sampling distribution of the original estimator. In Section 3.2 we consider the least-squares estimator where the variance, in the case where $\Sigma = \text{Id}_p$ is asymptotically $\kappa/(1-\kappa)\sigma_\epsilon^2/p$.

In the main text, we proposed to use

$$w_i \overset{iid}{\backsim} 1 - \alpha + \alpha\text{Poisson}(1)$$

We determined α numerically so that

$$\left[\kappa \frac{1}{1-\kappa-\mathbf{E}_{w_i}\left[\frac{1}{(1+cw_i)^2}\right]} - \frac{1}{1-\kappa}\right] = \frac{\kappa}{1-\kappa}. \tag{10}$$

This was done via a simple dichotomous search for α over the interval $[0,1]$. Our initial α was .95. We specified a tolerance of 10^{-2} for the results reported in the paper in Table 1. This means that we stopped the algorithm when the ratio of the two terms in Equation (10) was within 1% of 1. We used a sample size of 10^6 to estimate all the expectations. Table 2 gives the values of $\alpha(\kappa)$ found.

C.1 Case $\Sigma \neq \text{Id}_p$

In the case where $\Sigma \neq \text{Id}$, both $\mathbf{E}\left(\text{var}\left(v'\widehat{\beta}^*\right)\right)$ and $\text{var}\left(v'\widehat{\beta}\right)$ depend on $v'\Sigma^{-1}v$. It is therefore natural to ask how we could estimate this quantity. If we are able to do so, it is clear that we could follow the same strategy as above to find α from the data. Standard Wishart results (Mardia et al. (1979), Theorem 3.4.7) give that

$$\frac{v'\Sigma^{-1}v}{v'\widehat{\Sigma}^{-1}v} \sim \frac{\chi^2_{n-p}}{n-1} \to (1-\kappa) \text{in probability.}$$

κ	0.05	0.10	0.15	0.20	0.25
$\alpha(\kappa)$	0.9938	0.9875	0.9812	0.9688	0.9562
κ	0.30	0.35	0.40	0.45	0.50
$\alpha(\kappa)$	0.9426	0.9352	0.9277	0.9222	0.9203

Table 2: Values of $\alpha(\kappa)$ to use to fix the variance estimation issue in high-dimensional pairs-bootstrap

This of course suggests using $(1-p/n)v'\widehat{\Sigma}^{-1}v$ as an estimator of $v'\Sigma^{-1}v$ and solves the question we were discussing above.

However, we note that since

$$\frac{\mathbf{E}\left(\text{var}\left(v'\widehat{\beta}^*\right)\right)}{\text{var}\left(v'\widehat{\beta}\right)}$$

does not depend on Σ when the design is Gaussian or Elliptical, the same α should work regardless of Σ, provided it is positive definite. In particular, an acceptable weight distribution for resampling as defined above could be computed by assuming $\Sigma = \text{Id}_p$ and would work for any positive definite Σ.

Appendix D. Description of Numerics

Here we describe the implementation of various computational numerics used in the paper.

D.1 Simulation Description

In the simulations described in the paper, we explored variations in the distribution of the design matrix X, the error distribution, the loss function, the sample size (n), and the ratio of $\kappa = p/n$, detailed below.

All results in the paper were based upon $1,000$ replications of our simulation routine for each combination of these values. Each simulation consisted of

1. Simulation of data matrix X, $\{\epsilon_i\}_{i=1}^n$ and construction of data $y_i = X'\beta + \epsilon_i$. However, for our simulations, $\beta = 0$ (without loss of generality for the results, which are shift equivariant), so $y_i = \epsilon_i$.

2. Estimate $\hat{\beta}$ using the corresponding loss function. For L_2 this was via the `lm` command in R, for Huber via the `rlm` command in the `MASS` package with default settings ($k = 1.345$) (Venables and Ripley, 2002), and for L_1 via an internal program making use of MOSEK optimization package and accessed in R using the `Rmosek` package (MOSEK, 2014). The internal L_1 program was checked to give the same results as the `rq` function that is part of the R package `quantreg` (Koenker, 2013), but was much faster for simulations.

3. Bootstrapping according to the relevant bootstrap procedure (using the `boot` package) and estimating $\hat{\beta}^*$ for each bootstrap sample. Each bootstrap resampling consisted of $R = 1,000$ bootstrap samples, the minimum generally suggested for 95% confidence intervals (Davison and Hinkley, 1997). For jackknife resampling and for calculating leave-one-out prediction errors $\tilde{e}_{i(i)}$, we wrote an internal function that left out each observation in turn and recalculated $\hat{\beta}_{(i)}$.

4. Construction of confidence intervals for $\hat{\beta}_1$. For bootstrap resampling, we used the function `boot.ci` in the `boot` package to calculate confidence intervals. We calculated "basic", "percentile", "normal", and "BCA" confidence intervals (see help of `boot.ci` and Davison and Hinkley (1997) for details about each of these), but all results shown in the manuscript rely on only the percentile method. The percentile method calculates the boundaries of the confidence intervals as the estimates of 2.5% and 97.5% percentiles of $\hat{\beta}_1^*$ (note that the estimate is not exactly the *observed* 2.5% and 97.5% of $\hat{\beta}_1^*$, since there is a correction term for estimating the percentile, again see Davison and Hinkley (1997)). For the jackknife confidence intervals, the confidence interval calculated was a standard normal confidence interval ($\pm 1.96\sqrt{\widehat{var}_{Jack}(\hat{\beta}_1)}$)

D.2 Values of Simulation Parameters

Design Matrix For the design matrix X, we considered the following designs for the distribution of an element X_{ij} of the matrix X, ensuring that the vectors X_i had covariance Id_p in all cases :

- Normal: X_{ij} are i.i.d $N(0,1)$
- Double Exp: X_{ij} are i.i.d. double exponential with variance 1.
- Elliptical: $X_{ij} \sim \lambda_i Z_{ij}$ where the Z_{ij} are i.i.d $N(0,1)$ and the λ_i are i.i.d according to
  - $\lambda_i \sim Exp(\sqrt{2})$ (i.e. mean $1/\sqrt{2}$), so $\mathbf{E}\left(\lambda_i^2\right) = 1$
  - $\lambda_i \sim N(0,1)$
  - $\lambda_i \sim Unif(\sqrt{\frac{12}{13}}0.5, \sqrt{\frac{12}{13}}1.5)$ so that $\mathbf{E}\left(\lambda_i^2\right) = 1$

Error Distribution We used two different distributions for the i.i.d errors ϵ_i: $N(0,1)$ and standard double exponential (with variance 2).

Dimensions We simulated from $n = 100$, 500, and $1,000$ though we showed only $n = 500$ in our results for simplicity. Except where noted, no significant difference in the results was seen for varying sample size. The ratio κ was simulated at $0.01, 0.1, 0.3, 0.5$.

D.3 Details of Additional Numerics

In Tables A-1 to A-5 in AppendixI we give the precise numerical results from our simulations that are plotted in both the main text and supplementary figures.

Calculating Correction Factors for Jackknife We computed these quantities using the formula we mentioned in the text and Matlab. We solve the associated regression problems with `cvx` (Grant and Boyd, 2014, 2008), running `Mosek` (ApS, 2015) as our optimization

engine. We used $n = 500$ and $1,000$ simulations to compute the mean of the quantities we were interested in.

Relative Risk (Figure 3) In Figure 3 we plot for both Huber loss and L_1 loss the average risk $r_\rho(\kappa; G_{conv})$ (i.e. errors given by G_{conv}) relative to the average risk $r_\rho(\kappa; G)$ (i.e. errors distributed according to G), where G has a double exponential distribution. We also plot the relative average risk $r_\rho(\kappa; G_{norm})$, where $G_{norm} = N(0, \sigma_\epsilon^2)$. The values for Figure 3 were generated with Matlab, using cvx and Mosek, as described above. We picked $n = 500$ and did 500 simulations. p was taken in (5, 10, 30, 50, 75, 100, 125, 150, 175, 200, 225, 250, 275, 300, 350, 400, 450). We used our simulations for the case of the original errors to estimate $\mathbf{E}\left(\|\widehat{\beta} - \beta\|_2\right)$. We used this estimate in our simulation under the convolved error distribution. The Gaussian error simulations were made with $\mathcal{N}(0, 2)$ to match the variance of the double exponential distribution.

Calculation of amount of variance overestimated in pairs bootstrap (Figure 5a)

In Figure 5a, we plot the theoretical factor by which the pairs bootstrap overestimates the actual variance of $\beta_{1\rho}$. This figure was generated by assuming Poisson(1) weights and computing deterministically the expectations of interest. This was easy since if $W \sim$ Poisson(1), $P(W = k) = \frac{\exp(-1)}{k!}$.

We truncated the expansion of the expectation at $K = 100$, so we neglected terms of order 1/100! or lower only. The constant c was found by dichotomous search, with tolerance 10^{-6} for matching the equation $\mathbf{E}\left(1/(1 + Wc)\right) = 1 - p/n$. Once c was found, we approximated the expectation in Theorem 2 in the same fashion as we just described.

Once we had computed the quantity appearing in Theorem 2, we divided it by $\kappa/(1-\kappa)$. We repeated these computations for $\kappa = .05$ to $\kappa = .5$ by increments of 10^{-3} to produce our figure.

Appendix E. Proof of Theorem 1 (Residual bootstrap, p/n close to 1)

Proof

Recall the system describing the asymptotic limit of $\|\widehat{\beta}_\rho - \beta\|$ when $p/n \to \kappa$ and the design matrix has i.i.d mean 0, variance 1 entries, is, under some conditions on ϵ_i's and some mild further conditions on the design (see Section A above): $\|\widehat{\beta}_\rho - \beta\| \to r_\rho(\kappa)$ and the pair of positive and deterministic scalars $(c, r_\rho(\kappa))$ satisfy: if $\hat{z}_\epsilon = \epsilon + r_\rho(\kappa)Z$, where $Z \sim \mathcal{N}(0, 1)$ is independent of ϵ, and ϵ has the same distribution as ϵ_i's:

$$\begin{cases} \mathbf{E}\left((\text{prox}(c\rho))'(\hat{z}_\epsilon)\right) &= 1 - \kappa \ , \\ \kappa r_\rho^2(\kappa) &= \mathbf{E}\left([\hat{z}_\epsilon - \text{prox}(c\rho)(\hat{z}_\epsilon)]^2\right) \ . \end{cases}$$

In this system, prox$(c\rho)$ refers to Moreau's proximal mapping of the convex function $c\rho$ - see Moreau (1965) or Hiriart-Urruty and Lemaréchal (2001).

We first give an informal argument to "guess" the correct values of various quantities of interest, namely c and of course, $r_\rho(\kappa)$.

Note that when $|x| \ll c$, and when $\psi(x) \sim x$ at 0, prox$(c\rho)(x) \simeq \frac{x}{1+c}$. Hence, $x -$ prox$(c\rho)(x) \simeq xc/(1 + c)$. (Note that as long as $\psi(x)$ is linear near 0, we can assume that $\psi(x) \sim x$, since the scaling of ρ by a constant does not affect the performance of the estimators.)

We see that $1 - \kappa \simeq 1/(1 + c)$, so that $c \simeq \kappa/(1 - \kappa)$ - assuming for a moment that we can apply the previous approximations in the system. Hence, we have

$$\kappa r_\rho(\kappa)^2 \simeq (c/(1 + c))^2[r_\rho(\kappa)^2 + \sigma_\epsilon^2] \simeq \kappa^2[r_\rho(\kappa)^2 + \sigma_\epsilon^2] \ .$$

We can therefore conclude (informally at this point) that

$$r_\rho(\kappa)^2 \sim \frac{\sigma_\epsilon^2 \kappa}{1 - \kappa} \sim \frac{\sigma_\epsilon^2}{1 - \kappa} \ .$$

Once these values are guessed, it is easy to verify that $r_\rho(\kappa) \ll c$ and hence all the manipulations above are valid if we plug these two expressions in the system driving the performance of robust regression estimators described above. We note that our argument is not circular: we just described a way to guess the correct result. Once this has been done, we have to make a verification argument to show that our guess was correct.

In this particular case, the verification is done as follows: we can rewrite the expectations as integrals and split the domain of integration into $(-\infty, -s_\kappa)$, $(-s_\kappa, s_\kappa)$, (s_κ, ∞), with $s_\kappa = (1 - \kappa)^{-3/4}$. Using our candidate values for c and $r_\rho(\kappa)$, we see that the corresponding $\widehat{z}_\epsilon$ has extremely low probability of falling outside the interval $(-s_\kappa, s_\kappa)$ - recall that $1 - \kappa \to 0$. Coarse bounding of the integrands outside this interval shows the corresponding contributions to the expectations are negligible at the scales we consider. On the interval $(-s_\kappa, s_\kappa)$, we can on the other hand make the approximations for prox$(c\rho)(x)$ we discussed above and integrate them. That gives us the verification argument we need, after somewhat tedious but simple technical arguments. (Note that the method of propagation of errors in analysis described in (Miller, 2006) works essentially in a similar a-posteriori-verification fashion. Also, s_κ could be picked as $(1 - \kappa)^{-(1/2+\delta)}$ for any $\delta \in (0, 1/2)$ and the arguments would still go through.) ■

Appendix F. Proof of Theorem 2 (Expected Variance of the Pairs Bootstrap Estimator)

In this section, we compute the expected variance of the bootstrap estimator.

We recall that for random variables T, Γ, we have

$$\text{var}\left(T\right) = \text{var}\left(\mathbf{E}\left(T|\Gamma\right)\right) + \mathbf{E}\left(\text{var}\left(T|\Gamma\right)\right) \ .$$

In our case, $T = v'\widehat{\beta}_w$, the projection of the regression estimator $\widehat{\beta}_w$ obtained using the random weights w on the contrast vector v. Γ represents both the design matrix and the errors. We assume without loss of generality that $\|v\|_2 = 1$.

Hence,

$$\text{var}\left(v'\widehat{\beta}_w\right) = \text{var}\left(v'\mathbf{E}\left(\widehat{\beta}_w|\Gamma\right)\right) + \mathbf{E}\left(\text{var}\left(v'\widehat{\beta}_w|\Gamma\right)\right) \ .$$

In plain English, the variance of $v'\widehat{\beta}_w$ is equal to the variance of the bagged estimator plus the expectation of the variance of the bootstrap estimator (where we randomly weight observation (y_i, X_i) with weight w_i).

As explained in Section H, we can study without loss of generality the case where $\Sigma = \mathrm{Id}_p$ and $\beta = 0$. This is what we do in this proof. Further the rotational invariance arguments we give in Section H mean that we can focus on the case $v = e_p$,the p-th canonical basis vector, without loss of generality.

We consider the case where $X_i \overset{iid}{\backsim} \mathcal{N}(0, \mathrm{Id}_p)$. This allows us to work with results in El Karoui et al. (2011); El Karoui et al. (2013), El Karoui (2013).

Notational simplification To make the notation lighter, in what follows in this proof we use the notation $\widehat{\beta}$ for $\widehat{\beta}_w$. There are no ambiguities as we are always using a weighted version of the estimator and hence this simplification should not create any confusion.

In particular, we have, using the derivation of Equation (9) in El Karoui et al. (2013) and noting that in the least-squares case all approximations in that paper are actually exact equalities,

$$\widehat{\beta}_p = \hat{c}\frac{\sum_{i=1}^n w_i X_i(p) e_{i,[p]}}{p} \ .$$

$e_{i,[p]}$ here are the residuals based on the first $p-1$ predictors, when $\beta = 0$. We note that, under our assumptions on X_i's and w_i's, $\hat{c} = \frac{1}{n}\mathrm{trace}\left(S_w^{-1}\right) + \mathrm{o}_{L_2}(1)$, where $S_w = \frac{1}{n}\sum_{i=1}^n w_i X_i X_i'$. It is known from work in random matrix theory (see e.g El Karoui (2009)) that $\frac{1}{n}\mathrm{trace}\left(S_w^{-1}\right)$ is asymptotically deterministic in the situation under investigation with our assumptions on w and X, i.e $\frac{1}{n}\mathrm{trace}\left(S_w^{-1}\right) = c + \mathrm{o}_{L_2}(1)$, where $c = \mathbf{E}\left(\frac{1}{n}\mathrm{trace}\left(S_w^{-1}\right)\right)$.

We also recall the residuals representation from El Karoui et al. (2013), which are exact in the case of least-squares : namely here,

$$\widehat{\beta} - \widehat{\beta}_{(i)} = \frac{w_i}{n} S_i^{-1} X_i \psi(e_i) \ ,$$

which implies that, with $S_i = \frac{1}{n}\sum_{j\neq i} w_j X_j X_j'$,

$$\tilde{e}_{i(i)} = e_i + w_i \frac{X_i' S_i^{-1} X_i}{n}\psi(e_i) \ .$$

In the case of least-squares, $\psi(x) = x$, so that

$$e_i = \frac{\tilde{e}_{i(i)}}{1 + w_i c_i} \ ,$$

where

$$c_i = \frac{X_i' S_i^{-1} X_i}{n} \ .$$

These equalities also follow from simple linear algebra since we are in the least-squares case. We note that $c_i = c + \mathrm{o}_P(1)$, where c is deterministic, as explained in e.g El Karoui (2010), El Karoui (2013). Furthermore, here the approximation holds in L_2 because of our assumptions on w's and existence of moments for the inverse Wishart distribution - see e.g Haff (1979). As explained in El Karoui (2013), the same is true for $c_{i,[p]}$ which is the same quantity computed using the first $(p-1)$ coordinates of X_i, vectors we denote generically by V_i. We can rewrite

$$\widehat{\beta}_p = \hat{c}\frac{\sum_{i=1}^n w_i X_i(p)\frac{\tilde{e}_{i(i),[p]}}{1+w_i c_{i,[p]}}}{p} \ .$$

Let us call $\widehat{b}$ the bagged estimate. We note that $\tilde{e}_{i(i),[p]}$ is independent of w_i and so is $c_{i,[p]}$. We have already seen that $\hat{c}$ is close to a constant, c. So taking expectation with respect to the weights, we have, if $w_{(i)}$ denotes $\{w_j\}_{j\neq i}$, and using independence of the weights,

$$\widehat{b}_p = \frac{1}{p}\sum_{i=1}^n \mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right) X_i(p)\mathbf{E}_{w_{(i)}}\left(\tilde{e}_{i(i),[p]}\right)\left[1 + \mathrm{o}_{L_2}(1)\right] \ .$$

Now the last term is of course the prediction error for the bagged problem, i.e

$$\mathbf{E}_{w_{(i)}}\left(\tilde{e}_{i(i),[p]}\right) = \epsilon_i - V_i'(\widehat{g}_{(i)} - \gamma)$$

where $\widehat{g}_{(i)}$ is the bagged estimate of $\hat{\gamma}$ and $\hat{\gamma}$ is the regression vector obtained by regressing y_i on the first $p-1$ coordinates of X_i. (Recall that in these theoretical considerations we are assuming that $\beta = 0$, without loss of generality.)

So we have, since we can work in the null case where $\gamma = 0$ (without loss of generality),

$$\widehat{b}_p = \frac{1}{p}\sum_{i=1}^n \mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right) X_i(p)\left[\epsilon_i - V_i'\widehat{g}_{(i)}\right](1 + \mathrm{o}_{L_2}(1)) \ .$$

Hence,

$$\mathbf{E}\left(p\widehat{b}_p^2\right) = \frac{1}{p}\sum_{i=1}^n \left[\mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right)\right]^2 (\sigma_\epsilon^2 + \mathbf{E}\left(\|\widehat{g}_{(i)}\|_2^2\right))(1 + \mathrm{o}(1)) \ .$$

Now, in expectation, using e.g El Karoui (2013), $\mathbf{E}\left(\|\widehat{g}_{(i)}\|_2^2\right)(1 + \mathrm{o}(1)) = \mathbf{E}\left(\|\widehat{b}\|_2^2\right) = p\mathbf{E}\left(\widehat{b}_p^2\right)$. The last equality comes from the fact that all coordinates play a symmetric role in this problem, so they are all equal in law.

Now, recall that according to e.g El Karoui et al. (2013), top-right equation on p. 14562, or El Karoui (2010)

$$\frac{1}{n}\sum_{i=1}^n \frac{1}{1+cw_i} = 1 - \frac{p}{n} + \mathrm{o}_{L_2}(1) \ ,$$

since the previous expression effectively relates $\mathrm{trace}\left(D_w X(X' D_w X)^{-1} X'\right)$ to $n-p$, the rank of the corresponding "hat matrix".

Since $\frac{cw_i}{1+cw_i} = 1 - \frac{1}{1+cw_i}$, we see that

$$\mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right) = \frac{p}{n} + \mathrm{o}(1) \ .$$

Hence, for the bagged estimate, we have the equation

$$\mathbf{E}\left(\|\widehat{b}\|_2^2\right) = \frac{p}{n}\left(\sigma^2 + \mathbf{E}\left(\|\widehat{b}\|_2^2\right)\right)(1 + \mathrm{o}(1)) \ .$$

We conclude that

$$\mathbf{E}\left(\|\widehat{b}\|_2^2\right) = (1 + \mathrm{o}(1))\frac{\kappa}{1-\kappa}\sigma^2 \ .$$

Note that $\frac{\kappa}{1-\kappa}\sigma^2 = \mathbf{E}\left(\|\widehat{\beta}_{sLS}\|_2^2\right)$, where the latter is the standard (i.e non-weighted) least squares estimator.

We note that the rotational invariance argument given in El Karoui et al. (2011); El Karoui et al. (2013) still apply here, so that we have the

$$\widehat{b} - \beta \stackrel{\mathcal{L}}{=} \|\widehat{b} - \beta\| u \ ,$$

where u is uniform on the sphere and independent of $\|\widehat{b} - \beta\|$ (recall that this simply comes from the fact that if X_i is changed into OX_i, where O is orthogonal, $\widehat{b}$ is changed into $O\widehat{b}$ - and we then apply invariance arguments coming from rotational invariance of the distribution of X_i). Therefore,

$$\mathrm{var}\left(v'(\widehat{b} - \beta)\right) = \frac{\|v\|^2}{p}\mathbf{E}\left(\|\widehat{b} - \beta\|_2^2\right) \ .$$

So we conclude that

$$p\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}_w|\Gamma\right)\right) = p\mathrm{var}\left(v'\widehat{\beta}_w\right) - \frac{\kappa}{1-\kappa}\sigma^2\|v\|_2^2 + \mathrm{o}(1) \ .$$

Now, the quantity $\mathrm{var}\left(v'\widehat{\beta}_w\right)$ is well understood. The rotational invariance arguments we mentioned before give that

$$\mathrm{var}\left(v'\widehat{\beta}_w\right) = \frac{\|v\|_2^2}{p}\mathbf{E}\left(\|\widehat{\beta}_w - \beta\|_2^2\right) \ .$$

In fact, using the notation D_w for the diagonal matrix with $D_w(i,i) = w_i$, since

$$\widehat{\beta}_w - \beta = (X'D_wX)^{-1}X'D_w\epsilon \ ,$$

we see that

$$\mathbf{E}\left(\|\widehat{\beta}_w - \beta\|_2^2\right) = \sigma_\epsilon^2\mathbf{E}\left(\mathrm{trace}\left((X'D_wX)^{-2}X'D_{w^2}X\right)\right) \ .$$

(Note that under mild conditions on ϵ, X and w, we also have $\|\widehat{\beta}_w - \beta\|_2^2 = \mathbf{E}\left(\|\widehat{\beta}_w - \beta\|_2^2\right) + \mathrm{o}_{L_2}(1)$ - owing to concentration results for quadratic forms of vectors with independent entries; see Ledoux (2001).)

We now need to simplify this quantity.

Analytical simplification of trace $\left((X'D_wX)^{-2}X'D_{w^2}X\right)$ Of course,

$$\mathrm{trace}\left((X'D_wX)^{-2}X'D_{w^2}X\right) = \mathrm{trace}\left(D_wX(X'D_wX)^{-2}X'D_w\right) = \sum_{i=1}^n w_i^2X_i'(X'D_wX)^{-2}X_i \ .$$

Hence, if $\widehat{\Sigma}_w = \frac{1}{n}\sum_{i=1}^n w_iX_iX_i' \triangleq \frac{w_i}{n}X_iX_i' + \widehat{\Sigma}_{(i)}$, we have

$$\mathrm{trace}\left((X'D_wX)^{-2}X'D_{w^2}X\right) = \frac{1}{n}\sum_{i=1}^n w_i^2\frac{X_i'\widehat{\Sigma}^{-2}X_i}{n} \ .$$

Call $\widehat{\Sigma}(z) = \widehat{\Sigma} - z\mathrm{Id}_p$. Using the identity

$$(\widehat{\Sigma} - z\mathrm{Id}_p)(\widehat{\Sigma} - z\mathrm{Id}_p)^{-1} = \mathrm{Id}_p \ ,$$

we see, after taking traces, that (Silverstein (1995))

$$\frac{1}{n}\sum_{i=1}^n w_iX_i'(\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}X_i - z\mathrm{trace}\left((\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}\right) = p \ .$$

We call, for $z \in \mathbb{C}$, $c(z) = \frac{1}{n}\mathrm{trace}\left((\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}\right)$ and $c_i(z) = X_i'(\widehat{\Sigma}_{(i)} - z\mathrm{Id}_p)^{-1}X_i$, provided z is not an eigenvalue of $\widehat{\Sigma}$.

Differentiating with respect to z and taking $z = 0$ (we know here that $\widehat{\Sigma}$ is non-singular with probability 1, so this does not create a problem), we have

$$\frac{1}{n}\sum_{i=1}^n w_iX_i'\widehat{\Sigma}^{-2}X_i - \mathrm{trace}\left(\widehat{\Sigma}^{-1}\right) = 0 \ .$$

Also, since, by the Sherman-Morrison-Woodbury formula (Horn and Johnson (1990)),

$$X_i'\widehat{\Sigma}(z)^{-1}X_i = \frac{X_i'\widehat{\Sigma}_{(i)}(z)^{-1}X_i}{1 + w_i\frac{1}{n}X_i'\widehat{\Sigma}_{(i)}(z)^{-1}X_i} \ ,$$

we have, after differentiating,

$$\frac{1}{n}X_i'\widehat{\Sigma}^{-2}X_i = \frac{c_i'(0)}{[1 + w_ic_i(0)]^2} \ ,$$

where of course $c_i'(0) = X_i'\widehat{\Sigma}_{(i)}^{-2}X_i$. Hence,

$$\frac{1}{n}\sum_{i=1}^n w_i^2\frac{1}{n}X_i'\widehat{\Sigma}^{-2}X_i = \frac{1}{n}\sum_{i=1}^n w_i^2\frac{c_i'(0)}{[1 + w_ic_i(0)]^2} = c'(0)\frac{1}{n}\sum_{i=1}^n\frac{w_i^2}{[1 + w_ic(0)]^2} \ .$$

(Note that the arguments given in e.g El Karoui (2010) or El Karoui and Koesters (2011) for why $c_i(z) = c(z)(1 + \mathrm{o}_P(1))$ extend easily to c_i' and c' given our assumptions on w's and the fact that these functions have simple interpretations in terms of traces of powers of inverses of certain well-behaved - under our assumptions - matrices.)

Going back to

$$\frac{1}{n}\sum_{i=1}^n w_iX_i'(\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}X_i - z\mathrm{trace}\left((\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}\right) = p \ ,$$

and using the previously discussed identity

$$\frac{w_i}{n}X_i'(\widehat{\Sigma} - z\mathrm{Id}_p)^{-1}X_i = 1 - \frac{1}{1 + w_ic_i(z)} \ ,$$

we have

$$n - \sum_{i=1}^n\frac{1}{1 + w_ic_i(z)} - znc(z) = p \ .$$

In other words,

$$1-\kappa=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{1+w_ic_i(z)}+zc(z)\ .$$

Now,

$$\begin{aligned}c(z)\frac{1}{n}\sum_{i=1}^{n}\frac{w_i}{1+w_ic(z)}&=\frac{1}{n}\sum_{i=1}^{n}(1-\frac{1}{1+w_ic(z)})\\&=\frac{1}{n}\sum_{i=1}^{n}(1-\frac{1}{1+w_ic_i(z)})+\eta(z)\\&=\kappa+zc(z)+\eta(z)\ ,\end{aligned}$$

where $\eta(z)$ is such that $\eta(z)=\mathrm{o}_P(1)$ and $\eta'(z)=\mathrm{o}_P(1)$ (η has an explicit expression which allows us to verify these claims). Therefore, by differentiation, and after simplifications,

$$\frac{1}{n}\sum\left[\frac{w_i}{1+w_ic(0)}\right]^2c'(0)=\kappa\frac{c'(0)}{[c(0)]^2}-1+\mathrm{o}_P(1)\ .$$

Hence,

$$\mathrm{trace}\left((X'D_wX)^{-2}X'D_{w^2}X\right)=\left[\kappa\frac{\mathrm{trace}\left(\widehat{\Sigma}_w^{-2}\right)/n}{[\mathrm{trace}\left(\widehat{\Sigma}_w^{-1}\right)/n]^2}-1\right]+\mathrm{o}_P(1)\ .$$

The fact that we can take expectations on both sides of this equation and that $\mathrm{o}_P(1)$ is in fact $\mathrm{o}_{L_2}(1)$ come from our assumptions about w_i's - especially the fact that they are independent and bounded away from 0 - and properties of the inverse Wishart distribution.

Conclusion We can now conclude that a consistent estimator of the expected variance of the bootstrap estimator is

$$\frac{\|v\|_2^2}{p}\sigma_\epsilon^2\left[\kappa\frac{\mathrm{trace}\left(\widehat{\Sigma}_w^{-2}\right)/n}{[\mathrm{trace}\left(\widehat{\Sigma}_w^{-1}\right)/n]^2}-\frac{1}{1-\kappa}\right]\ .$$

Using the fact that

$$1-\kappa=\frac{1}{n}\sum_{i=1}^{n}\frac{1}{1+w_ic(z)}+zc(z)\ ,$$

we see that, since $\frac{1}{n}\mathrm{trace}\left(\widehat{\Sigma}_w^{-2}\right)=c'(0)$,

$$\frac{1}{n}\mathrm{trace}\left(\widehat{\Sigma}_w^{-2}\right)=\frac{c(0)}{\frac{1}{n}\sum_{i=1}^{n}w_i/(1+w_ic(0))^2}\ .$$

We further note that asymptotically, when w_i are i.i.d and satisfy our assumptions, $c(0)\to c$, which solves:

$$\mathbf{E}_{w_i}\left[\frac{1}{1+w_ic}\right]=1-\kappa\ .$$

Hence, asymptotically, when w_i's are i.i.d and satisfy our assumptions, we have

$$\frac{\mathrm{trace}\left(\widehat{\Sigma}_w^{-2}\right)/n}{[\mathrm{trace}\left(\widehat{\Sigma}_w^{-1}\right)/n]^2}\to\frac{1}{c\mathbf{E}_{w_i}[w_i/(1+w_ic)^2]}\ .$$

Since $cw_i/(1+cw_i^2)=1/(1+cw_i)-1/(1+cw_i)^2$, we finally see that

$$\begin{aligned}c\mathbf{E}_{w_i}\left[\frac{w_i}{(1+w_ic)^2}\right]&=\mathbf{E}_{w_i}\left[\frac{1}{1+cw_i}\right]-\mathbf{E}_{w_i}\left[\frac{1}{(1+cw_i)^2}\right]\ ,\\&=1-\kappa-\mathbf{E}_{w_i}\left[\frac{1}{(1+cw_i)^2}\right]\ .\end{aligned}$$

So asymptotically, the expected bootstrap variance is equivalent to, when $\|v\|_2=1$,

$$\frac{\sigma_\epsilon^2}{p}\left[\kappa\frac{1}{1-\kappa-\mathbf{E}\left(\frac{1}{(1+cw_i)^2}\right)}-\frac{1}{1-\kappa}\right]\ ,$$

where $\mathbf{E}\left(\frac{1}{1+cw_i}\right)=1-\kappa$.

In particular, when $w_i=1$, we see, unsurprisingly, that the above quantity is 0, as it should, given that the bootstrapped estimate does not change when resampling.

We finally make note of a technical point, that is addressed in papers such as El Karoui (2010, 2013) and on which we rely here by using those papers. Essentially, theoretical considerations regarding quantities such as $\frac{1}{p}\mathrm{trace}\left(\widehat{\Sigma}_w^{-k}\right)$ are easier to handle by working rather with $\frac{1}{p}\mathrm{trace}\left((\widehat{\Sigma}_w+\tau\mathrm{Id}_p)^{-k}\right)$, for some $\tau>0$. In the present context, it is easy to show (and done in those papers) that this approximation allows us to take the limit - even in expectation - for $\tau\to 0$ in all the expressions we get for $\tau>0$ and that that limit is indeed $\mathbf{E}\left(\frac{1}{p}\mathrm{trace}\left(\widehat{\Sigma}_w^{-k}\right)\right)$. Technical details rely on using the first resolvent identity (Kato, 1995), using moment properties of inverse Wishart distributions and using the fact that w_i's are bounded below.

F.1 Extension: Elliptical Design

In this case, we have $\tilde{X}_i=\lambda_iX_i$, where $X_i\sim\mathcal{N}(0,\mathrm{Id}_p)$ and $y_i=\epsilon_i+\tilde{X}_i'\beta$. We assume $\lambda_i\neq 0$ for all i, $\mathbf{E}\left(\lambda_i^2\right)=1$, λ_i's are i.i.d and bounded away from 0.

We can go through the proof of Theorem 2 and make necessary adjustments.

Of course, we have

$$\mathbf{E}\left(\|\widehat{\beta}_w-\beta\|_2^2\right)=\sigma_\epsilon^2\mathbf{E}\left(\mathrm{trace}\left((\tilde{X}'D_w\tilde{X})^{-2}\tilde{X}'D_{w^2}\tilde{X}\right)\right)\ .$$

If we reformulate this expression in terms of X we get

$$\mathbf{E}\left(\|\widehat{\beta}_w-\beta\|_2^2\right)=\sigma_\epsilon^2\mathbf{E}\left(\mathrm{trace}\left((X'D_{\lambda^2w}X)^{-2}X'D_{\lambda^2w^2}X\right)\right)\ .$$

So this quantity is affected by the distribution of λ_i's; hence the risk of $\widehat{\beta}_w$ is different in the Gaussian and elliptical design case.

The other important part of the proof is the computation of the risk of the bagged estimator. In this case, earlier work in random matrix theory (e.g El Karoui (2009); El Karoui and Koesters (2011)) shows that we can use the approximations

$$c_i \simeq \lambda_i^2 c \, ,$$

where $c = \lim_{n,p\to\infty} \mathbf{E}\left(\frac{1}{n}\text{trace}\left(S^{-1}\right)\right)$, where $S = \frac{1}{n}\sum_{i=1}^n \lambda_i^2 w_i X_i X_i'$, i.e $S = \frac{1}{n}X' D_{\lambda^2 w} X$.

If we call

$$g(\lambda_i^2) = \mathbf{E}_{w_i}\left(\frac{c\lambda_i^2}{1+c\lambda_i^2 w_i}\right) \, ,$$

we see by keeping track of changes in the earlier proof that we have asymptotically

$$\mathbf{E}\left(\|\hat{b}\|_2^2\right) = \frac{\mathbf{E}_{\lambda_i}(\lambda_i^2 g^2(\lambda_i^2))}{\kappa - \mathbf{E}_{\lambda_i}(\lambda_i^4 g^2(\lambda_i^2))}\sigma_\epsilon^2 \, .$$

The same arguments we used before give that

$$\mathbf{E}\left(g(\lambda_i^2)\right) = \kappa \, .$$

Based on this information, we can compute $\mathbf{E}\left(\text{var}\left(v'\widehat{\beta}_w^*\right)\right)$ as we had in the proof of Theorem 2 and compare it to var $\left(v'\widehat{\beta}\right)$. The expressions do not seem to simplify much further however in this case, by contrast to the Gaussian design case where $\lambda_i = 1$ for all i. (For instance, when λ_i=1 for all i's, $g(\lambda_i) = g(1) = \kappa$ and we recover the results of Theorem 2.)

Importantly, the characteristics of the distribution of λ_i that affect $\mathbf{E}\left(\text{var}\left(v'\widehat{\beta}_w^*\right)\right)$ go beyond $\mathbf{E}\left(\lambda_i^2\right)$. And hence the expression we gave in Theorem 2 won't apply directly to the elliptical case.

F.2 Extension: Multinomial$(n, 1/n)$ weights

A natural question is whether the computations we have made can be extended to w_i's that are i.i.d *Poisson*(1) and/or Multinomial$(n, 1/n)$, as in the standard bootstrap.

In both cases, technical issues arise because with asymptotically negligible but non-zero probability, the matrix $X' D_w X$ may be of rank less than p. This can handled in several ways. A simple one is to replace the weights w_i by $w_i(\tau) = \tau + (1-\tau)w_i$ and study the problem when $\tau \to 0$.

Beyond that technicality, an important question is whether one can handle the fact that the weights are dependent in the multinomial case. For quantities of the type $\frac{1}{n}\text{trace}\left((X' D_w X)^{-1}\right)$, it was argued in El Karoui (2010) that one could ignore the dependency issue and treat the problem as if the weights where i.i.d Po(1). This type of arguments would be easy to extend where we need them here, for instance in quantities that arise in the computation of $\mathbf{E}\left(\|\widehat{\beta}_w - \beta\|_2^2\right)$ or to show that we can write $c_i = c + \text{o}_P(1)$, where c is deterministic.

The remaining question is therefore the characterization of the risk of the bagged estimator. We have, with a slight modification with respect to the case of independent weights,

$$\hat{b}_p = \frac{1}{p}\sum_{i=1}^n X_i(p)\left[\mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right)\epsilon_i - V_i'\mathbf{E}_w\left(\hat{\gamma}_{(i)}\frac{cw_i}{1+cw_i}\right)\right](1+\text{o}_P(1)) \, .$$

As before, $\mathbf{E}_{w_i}\left(\frac{cw_i}{1+cw_i}\right) = p/n + \text{o}(1)$. The problem is the dependence between $\hat{\gamma}_{(i)}$ and w_i. The rotational invariance arguments we invoked before still hold, so that $\hat{\gamma}_i = \|\hat{\gamma}_{(i)}\|_2 u$, where u is uniform on the unit sphere and independent of $\|\hat{\gamma}_i\|_2$. It is also independent of V_i, since $\hat{\gamma}_{(i)}$ is the leave-one-out estimate of γ. The same rotational invariance arguments hold for the bagged estimate $\mathbf{E}_w \hat{\gamma}_{(i)}\frac{cw_i}{1+cw_i}$. Hence, after a little bit of work we see that

$$\mathbf{E}\left([V_i'\mathbf{E}_w\left(\hat{\gamma}_{(i)}\frac{cw_i}{1+cw_i}\right)]^2\right) = \mathbf{E}\left(\|\mathbf{E}_w\left(\hat{\gamma}_{(i)}\frac{cw_i}{1+cw_i}\right)\|_2^2\right) \, .$$

Using the fact that $w_{(i)}|w_i \sim \text{Multinomial}(n-w_i, 1/(n-1))$, the only real technical hurdle is to show that $\mathbf{E}_{w_i}\hat{\gamma}_{(i)}$ is asymptotically deterministic and independent of w_i. A strategy for this is to create a coupling: one can compare $\hat{\gamma}_{(i)}$ to $\hat{\mathfrak{g}}_{(i)}$, where $\hat{\mathfrak{g}}_{(i)}$ is computed using a $n-1$ dimensional vector of weights with distribution Multinomial$(n-1, 1/(n-1))$ - i.e running $w_i - 1$ multinomial trials after having obtained $w_{(i)}$ (the case $w_i = 0$ is easy to handle separately). Clearly, the distribution of $\hat{\mathfrak{g}}_{(i)}$ is independent of w_i, by construction. On the other hand, a bit of work on top of the leave-one-observation-out expansions shows that $\|\hat{\mathfrak{g}}_{(i)} - \hat{\gamma}_{(i)}\|_2^2$ is roughly of size at most $w_i^2/n \to 0$. Furthermore, $\|\mathbf{E}_w(\hat{\mathfrak{g}}_{(i)}) - \mathbf{E}_w(\hat{\gamma}_{(i)})\|_2 \to 0$ for the same reason. This suggests that further technical work along those lines will give that

$$\mathbf{E}_w\left(\hat{\gamma}_{(i)}\frac{cw_i}{1+cw_i}\right) \simeq \mathbf{E}_w\left(\hat{\gamma}_{(i)}\right)\mathbf{E}_w\left(\frac{cw_i}{1+cw_i}\right) \, ,$$

where $\simeq$ means that the approximation is valid in Euclidean norm. The same coupling arguments will give that

$$\|\mathbf{E}_w\left(\hat{\gamma}_{(i)}\right)\| \simeq \|\hat{b}\| \, ,$$

where $\hat{b}$ is the bagged estimator. This will yield the same results as in the i.i.d Po(1) case.

Numerical results We verified that our theoretical results (i.e Theorem 2) hold for Poisson(1) weights in limited simulations (note that in this case $w_i = 0$ is possible). For Gaussian design matrix, double exponential errors, and ratios $\kappa = .1, .3, .5$ we found that the ratio of the observed bootstrap expected variance of $\widehat{\beta}_1^*$ to our theoretical prediction using Poisson(1) weights was 1.0027, 1.0148, and 1.0252, respectively (here $n = 500$, and there were $R = 1000$ bootstrap resamples for each of 1000 simulations).

Appendix G. Proof of Theorem 3 (Jackknife Variance)

As explained in Section H, we can study without loss of generality the case where $\Sigma = \text{Id}_p$ and $\beta = 0$. This is what we do in this proof.

We study it in details in the least-squares case, and postpone a detailed analysis of the robust regression case to future studies.

According to the approximations in El Karoui et al. (2013), which are exact for least squares, or classic results Weisberg (2014) we have:

$$\widehat{\beta} - \widehat{\beta}_{(i)} = \frac{1}{n}\widehat{\Sigma}_{(i)}^{-1} X_i e_i \ .$$

Recall also that

$$e_i = \frac{\tilde{e}_{i(i)}}{1 + \frac{1}{n} X_i' \widehat{\Sigma}_{(i)}^{-1} X_i} \ .$$

Hence,

$$v'(\widehat{\beta} - \widehat{\beta}_{(i)}) = \frac{1}{n} v' \widehat{\Sigma}_{(i)}^{-1} X_i \frac{\tilde{e}_{i(i)}}{1 + \frac{1}{n} X_i' \widehat{\Sigma}_{(i)}^{-1} X_i} \ .$$

Hence,

$$n \sum_{i=1}^{n} [v'(\widehat{\beta} - \widehat{\beta}_{(i)})]^2 = \frac{1}{n} \sum_{i=1}^{n} \frac{[v' \widehat{\Sigma}_{(i)}^{-1} X_i \tilde{e}_{i(i)}]^2}{[1 + \frac{1}{n} X_i' \widehat{\Sigma}_{(i)}^{-1} X_i]^2} \ .$$

Note that at the denominator, we have

$$\begin{aligned} 1 + \frac{1}{n} X_i' \widehat{\Sigma}_{(i)}^{-1} X_i &= 1 + \frac{1}{n} \text{trace}\left(\widehat{\Sigma}^{-1}\right) + \text{o}_P(1) \ , \\ &= 1 + \frac{p}{n} \frac{1}{1 - p/n} + \text{o}_P(1) = \frac{1}{1 - p/n} + \text{o}_P(1) \ . \end{aligned}$$

by appealing to standard results about concentration of high-dimensional Gaussian random variables, and standard results in random matrix theory and classical multivariate statistics (see Mardia et al. (1979); Haff (1979)). By the same arguments, this approximation works not only for each i but for all $1 \leq i \leq n$ at once. The approximation is also valid in expectation, using results concerning Wishart matrices found for instance in Mardia et al. (1979).

For the numerator, we see that

$$T_i = v' \widehat{\Sigma}_{(i)}^{-1} X_i \tilde{e}_{i(i)} = v' \widehat{\Sigma}_{(i)}^{-1} X_i (\epsilon_i - X_i'(\widehat{\beta}_{(i)} - \beta)) \ .$$

Since ϵ_i is independent of X_i and $\widehat{\Sigma}_{(i)}$, we see that

$$\mathbf{E}\left(T_i^2\right) = \mathbf{E}\left(\epsilon_i^2\right) \mathbf{E}\left((v' \widehat{\Sigma}_{(i)}^{-1} X_i)^2\right) + \mathbf{E}\left([X_i'(\widehat{\beta}_{(i)} - \beta)]^2 [v' \widehat{\Sigma}_{(i)}^{-1} X_i]^2\right) \ .$$

If α and β are fixed vectors, $\alpha' X_i$ and $\beta' X_i$ are Gaussian random variables with covariance $\alpha'\beta$, since we are working under the assumption that $X_i \sim \mathcal{N}(0, \text{Id}_p)$. It is easy to check that if Z_1 and Z_2 are two Gaussian random variables with covariance γ and respective variances σ_1^2 and σ_2^2, we have

$$\mathbf{E}\left((Z_1 Z_2)^2\right) = \sigma_1^2 \sigma_2^2 + 2\gamma^2 \ .$$

We conclude that

$$\mathbf{E}\left((a' X_i)^2 (b' X_i)^2\right) = \|a\|_2^2 \|b\|_2^2 + 2(a'b)^2 \ .$$

We note that

$$\mathbf{E}\left([v' \widehat{\Sigma}_{(i)}^{-1} X_i]^2\right) = \mathbf{E}\left(v' \widehat{\Sigma}_{(i)}^{-2} v\right) \ .$$

Classic Wishart computations give (Haff (1979), p.536 (iii)) that as $n, p \to \infty$,

$$\mathbf{E}\left(\widehat{\Sigma}_{(i)}^{-2}\right) = \left[\frac{1}{(1 - p/n)^3} + \text{o}(1)\right] \text{Id}_p \ .$$

Hence, in our asymptotics,

$$\mathbf{E}\left((v' \widehat{\Sigma}_{(i)}^{-1} X_i)^2\right) \to \frac{1}{(1 - p/n)^3} \|v\|_2^2 \ .$$

We also note that

$$\mathbf{E}_\epsilon\left[(v' \widehat{\Sigma}_{(i)}^{-1} \widehat{\beta}_{(i)})^2\right] = \frac{1}{n} v' \widehat{\Sigma}_{(i)}^{-3} v \ .$$

Hence,

$$\mathbf{E}\left((v' \widehat{\Sigma}_{(i)}^{-1} \widehat{\beta}_{(i)})^2\right) = \text{o}(1) \text{in our asymptotics} \ .$$

Therefore,

$$\mathbf{E}\left(T_1^2\right) = \frac{1}{(1 - p/n)^3} \|v\|_2^2 \sigma_\epsilon^2 (1 + \frac{p/n}{1 - p/n}) + \text{o}(1)$$

since $\mathbf{E}\left(\|\widehat{\beta}_{(i)} - \beta\|_2^2\right) = \sigma_\epsilon^2 \frac{p/n}{1 - p/n} + \text{o}(1)$.

When $v = e_1$, we therefore have

$$\mathbf{E}\left(T_1^2\right) = \sigma_\epsilon^2 \frac{1}{(1 - p/n)^4} + \text{o}(1) \ .$$

Therefore, in that situation,

$$\mathbf{E}\left(n \sum_{i=1}^{n} (v'(\widehat{\beta}_{(i)} - \widehat{\beta})^2)\right) = \sigma_\epsilon^2 \frac{1}{(1 - p/n)^2} + \text{o}(1) \ .$$

In other words,

$$\mathbf{E}\left(\sum_{i=1}^{n} (v'(\widehat{\beta}_{(i)} - \widehat{\beta})^2)\right) = \left[\frac{1}{1 - p/n} + \text{o}(1)\right] \text{var}\left(\widehat{\beta}_1\right)$$

G.1 Dealing with Centering

Let us call $\widehat{\beta}_{(\cdot)} = \frac{1}{n} \sum_{i=1}^{n} \widehat{\beta}_{(i)}$. We have previously studied the properties of $\sum_{i=1}^{n} ([v'(\widehat{\beta} - \widehat{\beta}_{(i)})]^2)$ and now need to show that the same results apply to $\sum_{i=1}^{n} ([v'(\widehat{\beta}_{(\cdot)} - \widehat{\beta}_{(i)})]^2)$.

To show that replacing $\widehat{\beta}$ by $\widehat{\beta}_{(\cdot)}$ does not affect the result, we consider the quantity

$$n^2 [v'(\widehat{\beta} - \widehat{\beta}_{(\cdot)})]^2 \ .$$

Since $\widehat{\beta} - \widehat{\beta}_{(i)} = \frac{1}{n} \widehat{\Sigma}_{(i)}^{-1} X_i e_i$, we have

$$\widehat{\beta} - \widehat{\beta}_{(\cdot)} = \frac{1}{n^2} \sum_{i=1}^{n} \widehat{\Sigma}_{(i)}^{-1} X_i e_i \ .$$

Hence,

$$n^2[v'(\widehat{\beta}-\widehat{\beta}_{(\cdot)})]^2 = \left[\frac{1}{n}\sum_{i=1}^n v'\widehat{\Sigma}_{(i)}^{-1}X_i(\epsilon_i - X_i'(\widehat{\beta}-\beta))\right]^2 .$$

A simple variance computation gives that $\frac{1}{n}\sum_{i=1}^n v'\widehat{\Sigma}_{(i)}^{-1}X_i\epsilon_i \to 0$ in L^2, since each term has mean 0 and the variance of the sum goes to 0.

Recall now that

$$\widehat{\Sigma}^{-1}X_i = \frac{\widehat{\Sigma}_{(i)}^{-1}X_i}{1+c_i} ,$$

where all c_i's are equal to $p/n/(1-p/n)+\mathrm{o}_P(1)$. Let us call $\mathsf{c} = p/n/(1-p/n)$.

We conclude that

$$\frac{1}{n}\sum_{i=1}^n v'\widehat{\Sigma}_{(i)}^{-1}X_iX_i'(\widehat{\beta}-\beta) = v'(\widehat{\beta}-\beta)(1+\mathsf{c}+\mathrm{o}(1)) .$$

When v is given, we clearly have $v'(\widehat{\beta}-\beta) = \mathrm{o}_P(p^{-1/2})$, given the distribution of $\widehat{\beta}-\beta$ under our assumptions on X_i's and ϵ_i's. So we conclude that

$$n^2[v'(\widehat{\beta}-\widehat{\beta}_{(\cdot)})]^2 \to 0 \text{ in probability} .$$

Because we have enough moments, the previous result is also true in expectation.

G.2 Putting Everything Together

The jackknife estimate of variance of $v'\widehat{\beta}$ is up to a factor going to 1

$$\begin{aligned}\frac{n}{n-1}\mathrm{JACK}(\mathrm{var}\left(v'\widehat{\beta}\right)) &= \sum_{i=1}^n [(v'\widehat{\beta}_{(i)}-\widehat{\beta}_{(\cdot)})]^2 \\ &= \sum_{i=1}^n [(v'\widehat{\beta}_{(i)}-\widehat{\beta})]^2 + n[v'(\widehat{\beta}-\widehat{\beta}_{(\cdot)})]^2 .\end{aligned}$$

Our previous analyses therefore imply (using $v=e_1$) that

$$\frac{n}{n-1}\mathbf{E}\left(\mathrm{JACK}(\mathrm{var}\left(\widehat{\beta}_1\right))\right) = \left[\frac{1}{1-p/n}+\mathrm{o}(1)\right]\mathrm{var}\left(\widehat{\beta}_1\right) .$$

This completes the proof of Theorem 3

G.3 Extension: More Involved Designs and Different Loss Functions

Our approach could be used to analyze similar problems in the case of elliptical designs. However, in that case, it seems that the factor that will appear in quantifying the amount by which the variance is mis-estimated will depend in general on the ellipticity parameters. We refer to El Karoui (2013) for computations of quantities such as $v'\widehat{\Sigma}^{-2}v$ in that case, which are of course essential to measuring mis-estimation.

We obtained the possible correction we mentioned in the paper for these more general settings following the ideas used in the rigorous proof we just gave, as well as approximation

arguments given in El Karoui et al. (2013) and justified rigorously in El Karoui (2013). Checking fully rigorously all the approximations we made in this Jackknife computation would require a very large amount of technical work, and since this is tangential to our main interests in this paper, we postpone that to a future work of a more technical nature.

It is also clear, since all these results and the proof we just gave rely on random matrix techniques, that a similar analysis could be carried out in the case where $X_{i,j}$ are i.i.d with a non-Gaussian distribution, provided that distribution has enough moments (see e.g Pajor and Pastur (2009) or El Karoui and Koesters (2011) for examples of such techniques, actually going beyond the case of i.i.d entries for the design matrix). The main issues in carrying out this program seem to be technical and not conceptual at this point, so we leave this problem to possible future work.

Appendix H. Going from $\Sigma = \mathrm{Id}_p$ to $\Sigma \neq \mathrm{Id}_p$

As discussed in Section A, we have

$$\widehat{\beta}_\rho(y_i;X_i;\epsilon_i) - \beta = \Sigma^{-1/2}\widehat{\beta}_\rho(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) ,$$

In other words, $\widehat{\beta}_\rho(\tilde{y}_i;\Sigma^{-1/2}X_i;\epsilon_i)$ is the robust regression estimator in the null case where $\beta=0$ and X_i is replaced by $\tilde{X}_i = \Sigma^{-1/2}X_i$. Of course, if $\mathrm{cov}\,(X_i)=\Sigma$, $\mathrm{cov}\left(\tilde{X}_i\right) = \mathrm{Id}_p$.

H.1 Consequences for the Jackknife

Naturally the same equality applies to leave-one-out estimators. So, with the notations of Equation (7) in the main text, we have, when $\mathrm{span}(\{X_i\}_{i=1}^n) = \mathbb{R}^p$ and Σ is positive definite,

$$(v'[\widehat{\beta}_{(i)}-\tilde{\beta}])^2 = (v'\Sigma^{-1/2}[\widehat{\beta}_{(i)}(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) - \tilde{\beta}(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i)])^2 .$$

Let us call $\widehat{\beta}_\rho(\Sigma;\beta)$ our robust regression estimator when $\mathrm{cov}\,(X_i)=\Sigma$ and $\mathbf{E}\,(y_i|X_i) = X_i'\beta$. It is clear from the previous display that the properties of $var_{JACK}(v'\widehat{\beta}_\rho(\Sigma;\beta))$ are the same as those of $var_{JACK}(v'\Sigma^{-1/2}\widehat{\beta}_\rho(\mathrm{Id}_p;0))$. So understanding the null case is enough to understand the general case, which is why we focus on the null case in our computations.

Furthermore, by the same arguments, we have

$$\mathrm{var}\left(v'\widehat{\beta}_\rho(y_i;X_i;\epsilon_i)\right) = \mathrm{var}\left(v'\Sigma^{-1/2}\widehat{\beta}_\rho(\mathrm{Id}_p;0)\right) .$$

So we have

$$\frac{var_{JACK}(v'\widehat{\beta}_\rho(\Sigma;\beta))}{\mathrm{var}\left(v'\widehat{\beta}_\rho(\Sigma;\beta)\right)} = \frac{var_{JACK}(v'\Sigma^{-1/2}\widehat{\beta}_\rho(\mathrm{Id}_p;0))}{\mathrm{var}\left(v'\Sigma^{-1/2}\widehat{\beta}_\rho(\mathrm{Id}_p;0)\right)} .$$

Calling $u_1 = \Sigma^{-1/2}v/\|\Sigma^{-1/2}v\|$, we see that u_1 is a unit vector. And we finally see that

$$\frac{var_{JACK}(v'\widehat{\beta}_\rho(\Sigma;\beta))}{\mathrm{var}\left(v'\widehat{\beta}_\rho(\Sigma;\beta)\right)} = \frac{var_{JACK}(u_1'\widehat{\beta}_\rho(\mathrm{Id}_p;0))}{\mathrm{var}\left(u_1'\widehat{\beta}_\rho(\mathrm{Id}_p;0)\right)} .$$

Hence, characterizing $\frac{var_{JACK}(v'\widehat{\beta}_\rho(\mathrm{Id}_p;0))}{\mathrm{var}(v'\widehat{\beta}_\rho(\mathrm{Id}_p;0))}$ for all fixed unit vectors v characterizes

$$\frac{var_{JACK}(v'\widehat{\beta}_\rho(\Sigma;\beta))}{\mathrm{var}\left(v'\widehat{\beta}_\rho(\Sigma;\beta)\right)}$$

for all β and invertible Σ. This is why our proof is focused on the null case $\Sigma = \mathrm{Id}_p$ and $\beta = 0$.

H.2 Consequences for the Pairs Bootstrap

Let us call D_w the diagonal matrix with (i,i)-entry $D(i,i) = w_i$. We consider only the case where $w_i > 0$, so we do not have to consider the case where fewer than p X_i's are assigned positive weights - which would result in $\widehat{\beta}_\rho$ being ill-defined (since infinitely many solutions would then be feasible).

In particular, for least squares, we have in our setting

$$\widehat{\beta}_w = (X'D_wX)^{-1}X'D_wY = \beta + (X'D_wX)^{-1}X'D_w\epsilon \ .$$

More generally, by a simple change of variables, since $w_i > 0$ and $\mathrm{span}(\{X_i\}_{i=1}^n) = \mathbb{R}^p$, when Σ is invertible,

$$\widehat{\beta}_{w,\rho}(y_i;\{X_i\}_{i=1}^n;\epsilon_i) - \beta = \Sigma^{-1/2}\widehat{\beta}_{w,\rho}(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) \ .$$

If b_ρ is the corresponding bagged estimate, obtained by averaging $\widehat{\beta}_{w,\rho}$ over w's, we also have

$$b_\rho(y_i;\{X_i\}_{i=1}^n;\epsilon_i) - \beta = \Sigma^{-1/2}b_\rho(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) \ .$$

Hence, we also have

$$\widehat{\beta}_{w,\rho}(y_i;\{X_i\}_{i=1}^n;\epsilon_i) - b_\rho(y_i;\{X_i\}_{i=1}^n;\epsilon_i) = \Sigma^{-1/2}\left[\widehat{\beta}_{w,\rho}(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) - b_\rho(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i)\right]$$

We further note that since $y_i = \epsilon_i + X_i'\beta$, $y_i = \epsilon_i + (\Sigma^{-1/2}X_i)'\Sigma^{1/2}\beta$ and hence

$$\widehat{\beta}_{w,\rho}(y_i;\Sigma^{-1/2}X_i;\epsilon_i) = \Sigma^{1/2}\beta + \widehat{\beta}_{w,\rho}(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) \ .$$

The previous equation clearly implies that, if v is a fixed vector and $u_1 = \Sigma^{-1/2}v$

$$\begin{aligned} &v'(\widehat{\beta}^*_\rho(y_i;\{X_i\}_{i=1}^n;\beta) - b_\rho(y_i;\{X_i\}_{i=1}^n;\beta)) \\ &= u_1'\left[\widehat{\beta}^*_\rho(y_i;\Sigma^{-1/2}X_i;\epsilon_i) - b_\rho(y_i;\Sigma^{-1/2}X_i;\epsilon_i)\right] \ , \\ &= u_1'\left[\widehat{\beta}^*_\rho(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i) - b_\rho(\epsilon_i;\Sigma^{-1/2}X_i;\epsilon_i)\right] \ . \end{aligned}$$

We note that if $\mathrm{cov}\,(X_i) = \Sigma$, the last line in the previous display corresponds to the bootstrap distribution of our estimator in the null case where $\beta = 0$ and $\Sigma = \mathrm{Id}_p$, but v has been replaced by $u_1 = \Sigma^{-1/2}v$. This shows that understanding the bootstrap properties of $v'(\widehat{\beta}^*_\rho - b_\rho)$ in the null case $\mathrm{cov}\,(X_i) = \Sigma$ and $\beta = 0$ gives the result we seek in the general case of $\Sigma \neq \mathrm{Id}_p$ and $\beta \neq 0$. (Here we centered our estimator around the bagged estimator, because it is natural when computing bootstrap variances. The arguments above show that many other centering choices are possible, however.)

The last small issue that one needs to handle is the fact that our computations are done for v with unit norm and u_1 may not have unit norm. This is easily handled by simply scaling by the deterministic $\|u_1\|$. In particular, it is easy to see through simple scaling arguments that

$$\frac{\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_\rho(\Sigma;\beta)\right)\right)}{\mathrm{var}\left(v'\widehat{\beta}_\rho(\Sigma;\beta)\right)} = \frac{\mathbf{E}\left(\mathrm{var}\left(\tilde{u_1}'\widehat{\beta}^*_\rho(\mathrm{Id}_p;0)\right)\right)}{\mathrm{var}\left(\tilde{u}_1'\widehat{\beta}_\rho(\mathrm{Id}_p;0)\right)} \ ,$$

where $\tilde{u}_1 = u_1/\|u_1\|$ has unit norm.

H.3 Rotational Invariance Arguments and Consequences

Motivated by the arguments in the previous two subsections, we now consider the null case where $\beta = 0$ and $\mathrm{cov}\,(X_i) = \mathrm{Id}_p$. Note that then $y_i = \epsilon_i$. Also, if X_i is replaced by OX_i, where O is an orthogonal matrix, and $\widehat{\beta}$ is replaced by $O\widehat{\beta}$. In other words,

$$\widehat{\beta}_\rho(\epsilon_i;\{OX_i\}_{i=1}^n;\epsilon_i) = O\widehat{\beta}_\rho(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i) \ ,.$$

Note that when the design matrix is such that $OX_i \overset{\mathcal{L}}{=} X_i$ for all i (i.e the distribution of X_i's is invariant by rotation),

$$\widehat{\beta}_\rho(\epsilon_i;\{OX_i\}_{i=1}^n;\epsilon_i) \overset{\mathcal{L}}{=} \widehat{\beta}_\rho(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i) \ .$$

When $w_i > 0$ for all i, we see that exactly the same arguments apply to $\widehat{\beta}_{w,\rho}(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i)$ and hence $\widehat{\beta}^*_\rho(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i)$. In particular, for any orthogonal matrix O, since $X_i \overset{\mathcal{L}}{=} OX_i$,

$$\begin{aligned} \mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_\rho(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i)\right)\right) &= \mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_\rho(\epsilon_i;\{OX_i\}_{i=1}^n;\epsilon_i)\right)\right) \\ &= \mathbf{E}\left(\mathrm{var}\left(v'O\widehat{\beta}^*_\rho(\epsilon_i;\{X_i\}_{i=1}^n;\epsilon_i)\right)\right) \ . \end{aligned}$$

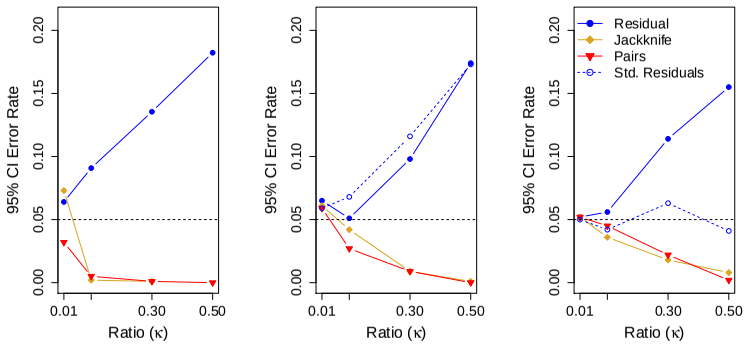
This implies that for any unit vector v, we have, if e_1 is the first canonical basis vector,

$$\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_\rho(\epsilon_i;X_i;\epsilon_i)\right)\right) = \mathbf{E}\left(\mathrm{var}\left(e_1'\widehat{\beta}^*_\rho(\epsilon_i;X_i;\epsilon_i)\right)\right) \ .$$

Indeed, we just need to take O to be such that $O'v = e_1$ to prove the above result.

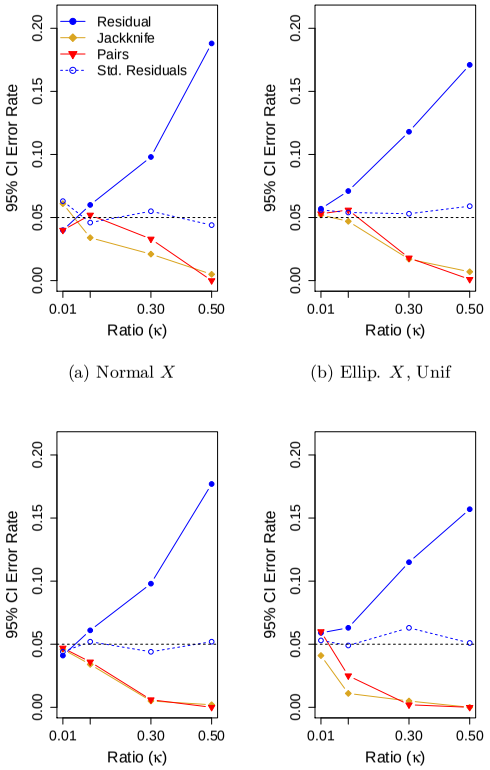
In the case where X_i's are i.i.d $\mathcal{N}(0,\mathrm{Id}_p)$, we do have $X_i \overset{\mathcal{L}}{=} OX_i$, so the arguments above apply. Therefore, to understand $\mathbf{E}\left(\mathrm{var}\left(v'\widehat{\beta}^*_\rho\right)\right)$ in this case it is sufficient to understand $\mathbf{E}\left(\mathrm{var}\left(e_1'\widehat{\beta}^*_\rho\right)\right)$. This latter case is the case tackled in the proof of Theorem 2. (These rotational invariance arguments are closely related to those in El Karoui et al. (2013).)

Appendix I. Supplementary Tables & Figures



(a) L_1 loss (b) Huber loss (c) L_2 loss

Figure A-1: **Performance of 95% confidence intervals of β_1 (double exponential error):** Here we show the coverage error rates for 95% confidence intervals for $n = 500$ with the error distribution being double exponential (with $\sigma^2 = 2$) and i.i.d. normal entries of X. See the caption of Figure 1 for more details.



(a) Normal X (b) Ellip. X, Unif

(c) Ellip. X, $N(0,1)$ (d) Ellip. X, Exp

Figure A-2: **Performance of 95% confidence intervals of β_1 for L_2 loss (elliptical design X):** Here we show the coverage error rates for 95% confidence intervals for $n = 500$ with different distributions of the design matrix X using ordinary least squares regression: (a) $N(0,1)$, (b) elliptical with $\lambda_i \sim U(\sqrt{\frac{12}{13}}0.5, \sqrt{\frac{12}{13}}1.5)$, (c) elliptical with $\lambda_i \sim N(0,1)$, and (d) elliptical with $\lambda_i \sim Exp(\sqrt{2})$. In all of these plots, the error is distributed $N(0,1)$ and the loss is L_2. See the caption of Figure 1 for additional details.

	Residual	Jackknife	Pairs
κ =0.01	0.063	0.089	0.035
κ =0.1	0.113	0.005	0.013
κ =0.3	0.137	0.000	0.003
κ =0.5	0.210	0.000	0.000

(a) L_1 loss

	Residual	Jackknife	Pairs
κ =0.01	0.057	0.054	0.054
κ =0.1	0.068	0.037	0.041
κ =0.3	0.090	0.015	0.004
κ =0.5	0.198	0.002	0.000

(b) Huber loss

	Residual	Jackknife	Pairs
κ =0.01	0.040	0.061	0.040
κ =0.1	0.060	0.034	0.052
κ =0.3	0.098	0.021	0.033
κ =0.5	0.188	0.005	0.000

(c) L_2 loss

Table A-1: **Error rate of 95% confidence intervals of β_1 for $n = 500$** This table gives the exact error rates plotted in Figure 1. $\kappa = p/n$ indicates the ratio of p/n used in the simultation for this and future tables. See Figure 1's caption for more details.

	Normal	Ellip. Normal	Ellip. Exp
κ =0.01	1.001	1.001	1.017
κ =0.1	1.016	1.090	1.156
κ =0.3	1.153	1.502	1.655
κ =0.5	1.737	3.123	3.635

Table A-2: **Ratio of CI Width of Pairs compared to Standard.** This table gives the ratio of the average width of the confidence intervals from pairs bootstrapping to the average for the standard interval given by theoretical results, i.e. using $var(\hat{\beta}) = \sigma^2(X'X)^{-1}$ and creating standard confidence interval. These values were used for Figure 4 in the text.

	Residual	Std. Pred Error	Deconv
κ =0.01	0.064	0.042	0.031
κ =0.1	0.091	0.028	0.018
κ =0.3	0.135	0.026	0.022
κ =0.5	0.182	0.030	0.035

(a) L_1 loss

	Residual	Std. Pred Error	Deconv
κ =0.01	0.065	0.048	0.036
κ =0.1	0.051	0.054	0.039
κ =0.3	0.098	0.035	0.037
κ =0.5	0.174	0.034	0.036

(b) Huber loss

Table A-3: **Error rate of 95% confidence intervals using predicted errors.** This table gives the exact error rates plotted in Figure 2. See figure caption for more details.

	Residual	Jackknife	Pairs
κ =0.01	0.064	0.073	0.032
κ =0.1	0.091	0.002	0.005
κ =0.3	0.135	0.001	0.001
κ =0.5	0.182		0.000

(a) L_1 loss

	Residual	Jackknife	Pairs
κ =0.01	0.065	0.061	0.059
κ =0.1	0.051	0.042	0.027
κ =0.3	0.098	0.009	0.009
κ =0.5	0.174	0.001	0.000

(b) Huber loss

	Residual	Jackknife	Pairs
κ =0.01	0.052	0.052	0.052
κ =0.1	0.056	0.036	0.045
κ =0.3	0.114	0.018	0.022
κ =0.5	0.155	0.008	0.002

(c) L_2 loss

Table A-4: **Error rate of 95% confidence intervals of β_1 for double exponential error** This table gives the exact error rates plotted in Figure A-1. See figure caption for more details.

	Residual	Jackknife	Pairs
κ =0.01	0.057	0.052	0.053
κ =0.1	0.071	0.047	0.056
κ =0.3	0.118	0.017	0.018
κ =0.5	0.171	0.007	0.001

(a) Ellipical, Unif

	Residual	Jackknife	Pairs
κ =0.01	0.041	0.046	0.047
κ =0.1	0.061	0.034	0.036
κ =0.3	0.098	0.005	0.006
κ =0.5	0.177	0.002	0.000

(b) Elliptical, Normal

	Residual	Jackknife	Pairs
κ =0.01	0.059	0.041	0.060
κ =0.1	0.063	0.011	0.025
κ =0.3	0.115	0.005	0.002
κ =0.5	0.157	0.000	0.000

(c) Elliptical, Exp

Table A-5: **Error rate of 95% confidence intervals of β_1 for elliptical design** X This table gives the exact error rates plotted in Figure A-2. See figure caption for more details.

	L2	Huber	L1
κ =0.01	0.964	0.991	2.060
κ =0.1	1.115	1.173	5.432
κ =0.3	1.411	1.613	10.862
κ =0.5	1.986	2.671	14.045

(a) Jackknife

	L2	Huber	L1
κ =0.01	1.078	0.923	1.081
κ =0.1	1.041	1.098	1.351
κ =0.3	1.333	1.954	2.001
κ =0.5	2.808	4.507	3.156

(b) Pairs Bootstrap

Table A-6: **Over estimation of variance for Pairs bootstrap and Jackknife** This table gives the median values of the boxplots plotted in Figures 5 and 6. See relevant figure captions for more details.

References

M. ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 7.1 (Revision 28).*, 2015. URL `http://docs.mosek.com/7.1/toolbox/index.html`.

D. Bean, P. J. Bickel, N. El Karoui, and B. Yu. Optimal M-estimation in high-dimensional regression. *Proceedings of the National Academy of Sciences*, 110(36):14563–14568, 2013.

A. Belloni, V. Chernozhukov, and K. Kato. Uniform post-selection inference for least absolute deviation regression and other Z-estimation problems. *Biometrika*, 102(1):77–94, March 2015.

R. Beran and M. S. Srivastava. Bootstrap tests and confidence regions for functions of a covariance matrix. *Ann. Statist.*, 13(1):95–115, 1985.

P. J. Bickel and D. A. Freedman. Bootstrapping regression models with many parameters. In *A Festschrift for Erich L. Lehmann*, Wadsworth Statist./Probab. Ser., pages 28–48. Wadsworth, Belmont, Calif., 1983.

P. J. Bickel, F. Götze, and W. R. van Zwet. Resampling fewer than n observations: gains, losses, and remedies for losses. *Statist. Sinica*, 7(1):1–31, 1997. Empirical Bayes, sequential analysis and related topics in statistics and probability (New Brunswick, NJ, 1995).

P. J. Bickel and D. A. Freedman. Some asymptotic theory for the bootstrap. *Ann. Statist.*, 9(6):1196–1217, 1981.

O. Chapelle, E. Manavoglu, and R. Rosales. Simple and scalable response prediction for display advertising. *ACM Trans. Intell. Syst. Technol.*, 5(4):61:1–61:34, December 2014.

A. Chatterjee and S. N. Lahiri. ASYMPTOTIC PROPERTIES OF THE RESIDUAL BOOTSTRAP FOR LASSO ESTIMATORS. *Proceedings of the American Mathematical Society*, 138(12):4497–4509, December 2010.

A. Chatterjee and S. N. Lahiri. Rates of convergence of the Adaptive LASSO estimators to the Oracle distribution and higher order refinements by the bootstrap. *The Annals of Statistics*, 41(3):1232–1259, June 2013.

A. Chatterjee and S. Lahiri. Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106(494):608–625, 2011.

M. R. Chernick. *Bootstrap Methods: A Practitioner's Guide*. Wiley, 1999.

Criteo. Criteo publicly available datasets, 2017. URL `http://research.criteo.com/outreach/`.

A. C. Davison and D. V. Hinkley. *Bootstrap methods and their application*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1997.

A. Delaigle and I. Gijbels. Estimation of integrated squared density derivatives from a contaminated sample. *Journal of the Royal Statistical Society, B*, 64:869–886, 2002.

A. Delaigle and I. Gijbels. Practical bandwidth selection in deconvolution kernel density estimation. *Computational Statistics and Data Analysis*, 45:249 – 267, 2004.

A. Delaigle. Nonparametric kernel methods with errors-in-variables: constructing estimators, computing them, and avoiding common mistakes. *Aust. N. Z. J. Stat.*, 56(2): 105–124, 2014.

R. Dezeure, P. Bühlmann, and C.-H. Zhang. High-dimensional simultaneous inference with the bootstrap. *TEST*, 26(4):685–719, October 2017.

P. Diaconis and D. Freedman. Asymptotics of graphical projection pursuit. *Ann. Statist.*, 12(3):793–815, 1984.

D. Donoho and A. Montanari. High dimensional robust m-estimation: Asymptotic variance via approximate message passing. *arXiv:1310.7320*, 2013.

M. L. Eaton and D. E. Tyler. On Wielandt's inequality and its application to the asymptotic distribution of the eigenvalues of a random symmetric matrix. *Ann. Statist.*, 19(1):260–271, 1991.

B. Efron. Bootstrap methods: another look at the jackknife. *Ann. Statist.*, 7(1):1–26, 1979.

B. Efron. *The jackknife, the bootstrap and other resampling plans*, volume 38 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa., 1982.

B. Efron and C. Stein. The jackknife estimate of variance. *Ann. Statist.*, 9(3):586–596, 1981.

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

N. El Karoui. Concentration of measure and spectra of random matrices: Applications to correlation matrices, elliptical distributions and beyond. *The Annals of Applied Probability*, 19(6):2362–2405, December 2009.

N. El Karoui. High-dimensionality effects in the Markowitz problem and other quadratic programs with linear constraints: risk underestimation. *Ann. Statist.*, 38(6):3487–3566, 2010.

N. El Karoui. On the realized risk of high-dimensional markowitz portfolios. *SIAM Journal in Financial Mathematics*, 4(1), 2013.

N. El Karoui. Asymptotic behavior of unregularized and ridge-regularized high-dimensional robust regression estimators: rigorous results. *arXiv:1311.2445*, 2013. ArXiv:1311.2445.

N. El Karoui. On the impact of predictor geometry on the performance on high-dimensional ridge-regularized generalized robust regression estimators. *Probability Theory and Related Fields*, 2017.

N. El Karoui and H. Koesters. Geometric sensitivity of random matrix results: consequences for shrinkage estimators of covariance and related statistical methods. *Submitted to Bernoulli*, 2011. Available at arXiv:1105.1404 (68 pages).

N. El Karoui, D. Bean, P. Bickel, C. Lim, and B. Yu. On robust regression with high-dimensional predictors. Technical Report 811, UC, Berkeley, Department of Statistics, 2011. Originally submitted as manuscript AoS1111-009. Not under consideration anymore.

N. El Karoui, D. Bean, P. J. Bickel, C. Lim, and B. Yu. On robust regression with high-dimensional predictors. *Proceedings of the National Academy of Sciences*, 2013.

J. Fan. On the optimal rates of convergence for nonparametric deconvolution problems. *Ann. Statist.*, 19(3):1257–1272, 1991.

M. Grant and S. Boyd. Graph implementations for nonsmooth convex programs. In V. Blondel, S. Boyd, and H. Kimura, editors, *Recent Advances in Learning and Control*, Lecture Notes in Control and Information Sciences, pages 95–110. Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

L. R. Haff. An identity for the Wishart distribution with applications. *J. Multivariate Anal.*, 9(4):531–544, 1979.

P. Hall. *The bootstrap and Edgeworth expansion*. Springer Series in Statistics. Springer-Verlag, New York, 1992.

P. Hall and S. Lahiri. Estimation of distributions, moments and quantiles in deconvolution problems. *The Annals of Statistics*, 36(5):2110–2134, 2008.

P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *J. R. Stat. Soc. Ser. B Stat. Methodol.*, 67(3):427–444, 2005.

J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of convex analysis*. Grundlehren Text Editions. Springer-Verlag, Berlin, 2001. Abridged version of ıt Convex analysis and minimization algorithms. I [Springer, Berlin, 1993; MR1261420 (95m:90001)] and ıt II [ibid.; MR1295240 (95m:90002)].

R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, Cambridge, 1990. Corrected reprint of the 1985 original.

P. J. Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *Ann. Statist.*, 1:799–821, 1973.

P. J. Huber and E. M. Ronchetti. *Robust statistics*. Wiley Series in Probability and Statistics. John Wiley & Sons Inc., Hoboken, NJ, second edition, 2009.

I. Johnstone. On the distribution of the largest eigenvalue in principal component analysis. *Ann. Statist.*, 29(2):295–327, 2001.

T. Kato. *Perturbation theory for linear operators*. Classics in Mathematics. Springer-Verlag, Berlin, 1995. Reprint of the 1980 edition.

A. Kleiner, A. Talwalkar, P. Sarkar, and M. I. Jordan. A scalable bootstrap for massive data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(4): 795–816, 2014.

R. Koenker. *Quantile regression*, volume 38 of *Econometric Society Monographs*. Cambridge University Press, Cambridge, 2005.

R. Koenker. *quantreg: Quantile Regression*, 2013. URL `http://CRAN.R-project.org/package=quantreg`. R package version 5.05.

J. Langford, L. Li, and A. Strehl, 2007. URL `https://github.com/JohnLangford/vowpal_wabbit/wiki`.

M. Ledoux. *The concentration of measure phenomenon*, volume 89 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2001.

M. Lopes. A Residual Bootstrap for High-Dimensional Regression with Near Low-Rank Designs. In *Advances in Neural Information Processing Systems NIPS*, pages 3239–3247, 2014.

E. Mammen. Asymptotics with increasing dimension for robust regression with applications to the bootstrap. *Ann. Statist.*, 17(1):382–400, 1989.

E. Mammen. Bootstrap, wild bootstrap, and asymptotic normality. *Probab. Theory Related Fields*, 93(4):439–455, 1992.

E. Mammen. Bootstrap and wild bootstrap for high-dimensional linear models. *Ann. Statist.*, 21(1):255–285, 1993.

K. V. Mardia, J. T. Kent, and J. M. Bibby. *Multivariate analysis*. Academic Press [Harcourt Brace Jovanovich Publishers], London, 1979. Probability and Mathematical Statistics: A Series of Monographs and Textbooks.

J. W. McKean, S. J. Sheather, and T. P. Hettmansperger. The Use and Interpretation of Residuals Based on Robust Estimation. *Journal of the American Statistical Association*, 88(424):1254–1263, December 1993.

P. D. Miller. *Applied asymptotic analysis*, volume 75 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2006.

J.-J. Moreau. Proximité et dualité dans un espace hilbertien. *Bull. Soc. Math. France*, 93: 273–299, 1965.

MOSEK. *Rmosek: The R to MOSEK Optimization Interface*, 2014. URL `http://rmosek.r-forge.r-project.org/,http://www.mosek.com/`. R package version 7.0.5.

A. Pajor and L. Pastur. On the limiting empirical measure of eigenvalues of the sum of rank one matrices with log-concave distribution. *Studia Math.*, 195(1):11–29, 2009.

M. I. Parzen, L. J. Wei, and Z. Ying. A resampling method based on pivotal estimating functions. *Biometrika*, 81(2):341–350, 1994.

D. N. Politis, J. P. Romano, and M. Wolf. *Subsampling*. Springer Series in Statistics. Springer-Verlag, New York, 1999.

S. Portnoy. Asymptotic behavior of M-estimators of p regression parameters when p^2/n is large. I. Consistency. *Ann. Statist.*, 12(4):1298–1309, 1984.

S. Portnoy. Asymptotic behavior of M estimators of p regression parameters when p^2/n is large. II. Normal approximation. *Ann. Statist.*, 13(4):1403–1417, 1985.

S. Portnoy. Asymptotic behavior of the empiric distribution of M-estimated residuals from a regression model with many parameters. *Ann. Statist.*, 14(3):1152–1170, 1986.

S. Portnoy. A central limit theorem applicable to robust regression estimators. *J. Multivariate Anal.*, 22(1):24–50, 1987.

G. R. Shorack. Bootstrapping robust regression. *Comm. Statist. A—Theory Methods*, 11 (9):961–972, 1982.

J. W. Silverstein. Strong convergence of the empirical distribution of eigenvalues of large-dimensional random matrices. *J. Multivariate Anal.*, 55(2):331–339, 1995.

D. W. Stroock. *Probability theory, an analytic view*. Cambridge University Press, Cambridge, 1993.

A. W. van der Vaart. *Asymptotic statistics*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, Cambridge, 1998.

W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Springer, New York, fourth edition, 2002. ISBN 0-387-95457-0.

K. W. Wachter. The strong limits of random matrix spectra for sample matrices of independent elements. *Annals of Probability*, 6(1):1–18, 1978.

X. Wang and B. Wang. Deconvolution estimation in measurement error models: The r package decon. *Journal of Statistical Software*, 39(10):1–24, 2011.

S. Weisberg. *Applied linear regression*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, fourth edition, 2014.

C.-F. J. Wu. Jackknife, bootstrap and other resampling methods in regression analysis. *Ann. Statist.*, 14(4):1261–1350, 1986. With discussion and a rejoinder by the author.

S. Zheng, D. Jiang, Z. Bai, and X. He. Inference on multiple correlation coefficients with moderately high dimensional data. *Biometrika*, 101:748–754, 2014.

Journal of Machine Learning Research 19 (2018) 1-33 Submitted 1/17; Revised 4/18; Published 8/18

RSG: Beating Subgradient Method without Smoothness and Strong Convexity

Tianbao Yang* TIANBAO-YANG@UIOWA.EDU
Department of Computer Science
The University of Iowa, Iowa City, IA 52242, USA

Qihang Lin QIHANG-LIN@UIOWA.EDU
Department of Management Sciences
The University of Iowa, Iowa City, IA 52242, USA

Editor: Inderjit Dhillon

Abstract

In this paper, we study the efficiency of a **R**estarted **S**ub**G**radient (RSG) method that periodically restarts the standard subgradient method (SG). We show that, when applied to a broad class of convex optimization problems, RSG method can find an ϵ-optimal solution with a lower complexity than the SG method. In particular, we first show that RSG can reduce the dependence of SG's iteration complexity on the distance between the initial solution and the optimal set to that between the ϵ-level set and the optimal set multiplied by a logarithmic factor. Moreover, we show the advantages of RSG over SG in solving a broad family of problems that satisfy a local error bound condition, and also demonstrate its advantages for three specific families of convex optimization problems with different power constants in the local error bound condition. (a) For the problems whose epigraph is a polyhedron, RSG is shown to converge linearly. (b) For the problems with local quadratic growth property in the ϵ-sublevel set, RSG has an $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ iteration complexity. (c) For the problems that admit a local Kurdyka-Łojasiewicz property with a power constant of $\beta \in [0,1)$, RSG has an $O(\frac{1}{\epsilon^{2\beta}}\log(\frac{1}{\epsilon}))$ iteration complexity. The novelty of our analysis lies at exploiting the lower bound of the first-order optimality residual at the ϵ-level set. It is this novelty that allows us to explore the local properties of functions (e.g., local quadratic growth property, local Kurdyka-Łojasiewicz property, more generally local error bound conditions) to develop the improved convergence of RSG. We also develop a practical variant of RSG enjoying faster convergence than the SG method, which can be run without knowing the involved parameters in the local error bound condition. We demonstrate the effectiveness of the proposed algorithms on several machine learning tasks including regression, classification and matrix completion.

Keywords: subgradient method, improved convergence, local error bound, machine learning

1. Introduction

We consider the following generic optimization problem

$$f_* := \min_{\mathbf{w}\in\Omega} f(\mathbf{w}), \tag{1}$$

where $f : \mathbb{R}^d \to (-\infty, +\infty]$ is an extended-valued, lower semicontinuous and convex function, and $\Omega \subseteq \mathbb{R}^d$ is a closed convex set such that $\Omega \subseteq \mathrm{dom}(f)$. Here, we do not assume the smoothness of f on $\mathrm{dom}(f)$. During the past several decades, many fast (especially linearly convergent) optimization algorithms have been developed for (1) when f is smooth and/or strongly convex. On the contrary, there are relatively fewer techniques for solving generic non-smooth and non-strongly convex optimization problems, which have many applications in machine learning, statistics, computer vision, and etc. To solve (1) with f being potentially non-smooth and non-strongly convex, one of the simplest algorithms to use is the subgradient (SG) [1] method. When f is Lipschitz-continuous, it is known that SG method requires $O(1/\epsilon^2)$ iterations for obtaining an ϵ-optimal solution (Rockafellar, 1970; Nesterov, 2004). It has been shown that this iteration complexity is unimprovable for general non-smooth and non-strongly convex problems in a black-box first-order oracle model of computation (Nemirovsky A.S. and Yudin, 1983). However, better iteration complexity can be achieved by other first-order algorithms for certain classes of f where additional structural information is available (Nesterov, 2005; Gilpin et al., 2012; Freund and Lu, 2017; Renegar, 2014, 2015, 2016).

In this paper, we present a generic restarted subgradient (RSG) method for solving (1) which runs in multiple stages with each stage warm-started by the solution from the previous stage. Within each stage, the standard projected subgradient update is performed for a fixed number of iterations with a constant step size. This step size is reduced geometrically from stage to stage. With these schemes, we show that RSG can achieve a lower iteration complexity than the classical SG method when f belongs to some classes of functions. In particular, we summarize the main results and properties of RSG below:

- For the general problem (1), under mild assumptions (see Assumption 1 and 2), RSG has an iteration complexity of $O(\frac{1}{\epsilon^2}\log(\frac{\epsilon_0}{\epsilon}))$ which has an additional $\log(\frac{\epsilon_0}{\epsilon})$ [2] term but has significantly smaller constant in $O(\cdot)$ compared to SG. In particular, compared with SG whose iteration complexity quadratically depends on the distance from the initial solution to the optimal set, RSG's iteration complexity has a quadratic dependence on the distance from the ϵ-level set to the optimal set, which is much smaller than the distance from the initial solution to the optimal set. Its dependence on the initial solution is through ϵ_0 - a known upper bound of the initial optimality gap, which only scales logarithmically.
- When the epigraph of f over Ω is a polyhedron, RSG can achieve linear convergence, i.e., an $O(\log(\frac{1}{\epsilon}))$ iteration complexity.
- When f is locally quadratically growing (see Definition 10), which is a weaker condition than strong convexity, RSG can achieve an $O(\frac{1}{\epsilon}\log(\frac{1}{\epsilon}))$ iteration complexity.
- When f admits a local Kurdyka-Łojasiewicz property (see Definition 13) with a power desingularizing function of degree $1-\beta$ where $\beta \in [0,1)$, RSG can achieve an $O(\frac{1}{\epsilon^{2\beta}}\log(\frac{1}{\epsilon}))$ complexity.

∗. Correspondence

1. In this paper, we use SG to refer deterministic subgradient method, though it is used in literature for stochastic gradient methods.
2. ϵ_0 is a known upper bound of the initial optimality gap in terms of the objective value.

These results, except for the first one, are derived from a generic complexity of RSG for the problem satisfying a *local error condition* (15), which has a close connection to the existing error bound conditions and growth conditions in the literature (Pang, 1997, 1987; Luo and Tseng, 1993; Necoara et al., 2015; Bolte et al., 2006). In spite of its simplicity, the analysis of RSG provides additional insight on improving first-order methods' iteration complexity via restarting. It is known that restarting can improve the theoretical complexity of (stochastic) SG method for non-smooth problems when strongly convexity is assumed (Ghadimi and Lan, 2013; Chen et al., 2012; Hazan and Kale, 2011) but we show that restarting can be still helpful for SG methods under other (weaker) assumptions. We would like to remark that the key lemma (Lemma 4) developed in this work can be leveraged to develop faster algorithms in different contexts. For example, built on the groundwork laid in this paper, Xu et al. (2016) have developed new smoothing algorithms to improve the convergence of Nesterov's smoothing algorithm (Nesterov, 2005) for non-smooth optimization with a special structure, and Xu et al. (2017) have developed new stochastic subgradient methods to improve the convergence of standard stochastic subgradient method.

We organize the reminder of the paper as follows. Section 2 reviews some related work. Section 3 presents some preliminaries and notations. Section 4 presents the algorithm of RSG and the general theory of convergence. Section 5 considers several classes of non-smooth and non-strongly convex problems and shows the improved iteration complexities of RSG. Section 6 presents parameter-free variants of RSG. Section 8 presents some experimental results. Finally, we conclude in Section 9.

2. Related Work

Smoothness and strong convexity are two key properties of a convex optimization problem that affect the iteration complexity of finding an ϵ-optimal solution by first-order methods. In general, a lower iteration complexity is expected when the problem is either smooth or strongly convex. Recently there has emerged a surge of interest in further accelerating first-order methods for non-strongly convex or non-smooth problems that satisfy some particular conditions (Bach and Moulines, 2013; Wang and Lin, 2014; So and Zhou, 2017; Hou et al., 2013; Zhou et al., 2015; Gong and Ye, 2014; Gilpin et al., 2012; Freund and Lu, 2017). The key condition for us to develop an improved complexity is a local error bound condition (15) which is closely related to the error bound conditions in the literature (Pang, 1987, 1997; Luo and Tseng, 1993; Necoara et al., 2015; Bolte et al., 2006; Zhang, 2016).

Various error bound conditions have been exploited in many studies to analyze the convergence of optimization algorithms. For example, Luo and Tseng (1992a,b, 1993) established the asymptotic linear convergence of a class of feasible descent algorithms for smooth optimization, including coordinate descent method and projected gradient method, based on a local error bound condition. Their results on coordinate descent method were further extended to a more general class of objective functions and constraints by Tseng and Yun (2009a,b). Wang and Lin (2014) showed that a global error bound holds for a family of non-strongly convex and smooth objective functions for which feasible descent methods can achieve a global linear convergence rate. Recently, these error bounds have been generalized and leveraged to show faster convergence for structured convex optimization that consists of a smooth function and a simple non-smooth function (Hou et al., 2013; Zhou and So, 2017; Zhou et al., 2015). Recently, Necoara and Clipici (2016) considered a generalized error bound condition, and established linear convergence of a parallel version of a randomized (block) coordinate descent method for minimizing the sum of a partially separable smooth convex function and a fully separable non-smooth convex function.

We would like to emphasize that the aforementioned error bounds are different from the local error bound explored in this paper. In particular, they bound the distance of a point to the optimal set by using the norm of the projected gradient or proximal gradient at the point, thus requiring the (partial) smoothness of the objective function. In contrast, we bound the distance of a point to the optimal set by its objective residual with respect to the optimal value, covering a much broader family of functions. More recently, there have appeared many studies that consider smooth optimization or composite smooth optimization problems whose objective functions satisfy different error bound conditions, growth conditions or other non-degeneracy conditions and established the linear convergence rates of several first-order methods including proximal-gradient method, accelerated gradient method, prox-linear method and so on (Gong and Ye, 2014; Necoara et al., 2015; Zhang and Cheng, 2015; Zhang, 2016; Karimi et al., 2016; Drusvyatskiy and Lewis, 2018; Drusvyatskiy and Kempton, 2016; Hou et al., 2013; Zhou et al., 2015). The relative strength and relationships between some of those conditions are studied by Necoara et al. (2015) and Zhang (2016). For example, Necoara et al. (2015) showed that under the smoothness assumption the second-order growth condition (i.e., the considered error bound condition in the present work with $\theta = 1/2$) is equivalent to the error bound condition considered by Wang and Lin (2014). It was brought to our attention that the local error bound condition in the present paper is closely related to metric subregularity of subdifferentials (Artacho and Geoffroy, 2008; Kruger, 2015; Drusvyatskiy et al., 2014; Mordukhovich and Ouyang, 2015).

Gilpin et al. (2012) established a polyhedral error bound condition for problems whose epigraph is polyhedral and domain is a bounded polytope. Using this polyhedral error bound condition, they studied a two-person zero-sum game and proposed a restarted first-order method based on Nesterov's smoothing technique (Nesterov, 2005) that can find the Nash equilibrium and has linear convergence rate. The differences between Gilpin et al. (2012)'s work and this work are: (i) we study subgradient methods instead of Nesterov's smoothing technique, where the former have broader applicability than Nesterov's smoothing technique; (ii) our linear convergence can be derived for a slightly general problem where the domain is allowed to be an unbounded polyhedron as long as the polyhedral error bound condition in Lemma 8 holds, which is the case for many important applications; (iii) we consider a general condition that subsumes the polyhedral error bound condition as a special case and we try to solve the general problem (1) rather than the bilinear saddle-point problem considered by Gilpin et al. (2012).

The error bound condition that allows us to derive a linear convergence of RSG is the same to the weak sharp minimum condition, which was first coined in 1970s (Polyak, 1979). However, it was used even earlier for studying the convergence of subgradient method (Eremin, 1965; Polyak, 1969). Later, it was studied in many subsequent works (Polyak, 1987; Burke and Ferris., 1993; Studniarski and Ward, 1999; Ferris, 1991; Burke and Deng, 2002, 2005, 2009). Finite or linear convergence of several algorithms has been established under the weak sharp minimum condition, including gradient projection method (Polyak, 1987), the proximal point algorithm (PPA) (Ferris, 1991), and subgradient method with a

particular choice of step size (see below) (Polyak, 1969). We would like to emphasize the differences between the results in these works and the results in the present work that make our results novel: (i) the gradient projection method and its finite convergence established in (Polyak, 1987) requires the gradient of the objective function to be Lipschitz continuous, i.e., the objective function is smooth (see Polyak, 1987, Chap. 7, pp 207, Theorem 1), in contrast we do not assume smoothness of the objective function; (ii) the PPA studied in (Ferris, 1991) requires solving a proximal sub-problem consisting of the original objective function and a strongly convex function at every iteration, and therefore its finite convergence does not mean that only a finite number of subgradient evaluations is needed. In contrast, the linear convergence in this paper was in terms of the number of subgradient evaluations; (iii) linear convergence of a subgradient method studied by Polyak (1969) requires knowing the optimal objective value for setting its step size, and its convergence is in terms of the distance of the iterates to the optimal set, which is weaker than our linear convergence in terms of objective gap. In addition, our method does not require knowing the optimal objective value. Instead the basic variant of RSG that has a linear convergence only needs to know the value of the multiplicative constant parameter in the local error bound condition. For problems without knowing this parameter, we also develop a practical variant of RSG that can achieve a convergence rate close to linear convergence.

In his recent work (Renegar, 2014, 2015, 2016), Renegar presented a framework of applying first-order methods to general conic optimization problems by transforming the original problem into an equivalent convex optimization problem with only linear equality constraints and a Lipschitz-continuous objective function. This framework greatly extends the applicability of first-order methods to the problems with general linear inequality constraints and leads to new algorithms and new iteration complexity. One of his results related to this work implies (Renegar, 2015, Corollary 3.4), if the objective function has a polyhedral epigraph and the optimal objective value is known beforehand, a subgradient method can have a linear convergence rate. Compared to this result of his, our method does not need to know the optimal objective value. Note that Renegar's method can be applied in a general setting where the objective function is not necessarily polyhedral while our method obtains improved iteration complexities under the local error bound conditions.

More recently, Freund and Lu (2017) proposed a new SG method by assuming that a strict lower bound of f_*, denoted by f_{slb}, is known and f satisfies a growth condition, $\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \mathcal{G} \cdot (f(\mathbf{w}) - f_{slb})$, where $\mathbf{w}^*$ is the optimal solution closest to $\mathbf{w}$ and $\mathcal{G}$ is a growth rate constant depending on f_{slb}. Using a novel step size that incorporates f_{slb}, for non-smooth optimization, their SG method achieves an iteration complexity of $O(\mathcal{G}^2(\frac{\log H}{\epsilon'} + \frac{1}{\epsilon'^2}))$ for finding a solution $\hat{\mathbf{w}}$ such that $f(\hat{\mathbf{w}}) - f_* \leq \epsilon'(f_* - f_{slb})$, where $H = \frac{f(\mathbf{w}_0) - f_{slb}}{f_* - f_{slb}}$ and $\mathbf{w}_0$ is the initial solution. We note that there are several key differences in the theoretical properties and implementations between our work and that by Freund and Lu (2017): (i) Their growth condition has a similar form to the inequality (7) proved for a general function but there are still noticeable differences in the both sides and the growth constants. (ii) The convergence results established by Freund and Lu (2017) are based on finding an solution $\hat{\mathbf{w}}$ with a relative error of ϵ' while we consider absolute error. (iii) By rewriting the convergence results of Freund and Lu (2017) in terms of absolute accuracy ϵ with $\epsilon = \epsilon'(f_* - f_{slb})$, their algorithm's complexity depends on $f_* - f_{slb}$ and may be higher than ours if $f_* - f_{slb}$ is large. However, Freund and Lu's new SG method is still attractive due to that it is a parameter free algorithm without requiring the value of the growth constant $\mathcal{G}$. We will compared our RSG method with the method of Freund and Lu (2017) with more details in Section 7.

Restarting and multi-stage strategies have been employed to achieve the (uniformly) optimal theoretical complexity of (stochastic) SG methods when f is strongly convex (Ghadimi and Lan, 2013; Chen et al., 2012; Hazan and Kale, 2011) or uniformly convex (Juditsky and Nesterov, 2014). Here, we show that restarting can be still helpful even without uniform or strong convexity. Furthermore, in all the algorithms proposed in existing works (Ghadimi and Lan, 2013; Chen et al., 2012; Hazan and Kale, 2011; Juditsky and Nesterov, 2014), the number of iterations per stage increases between stages while our algorithm uses the same number of iterations in all stages. This provides a different possibility of designing restarted algorithms for a better complexity only under a local error bound condition.

3. Preliminaries

In this section, we define some notations used in this paper and present the main assumptions needed to establish our results. We use $\partial f(\mathbf{w})$ to denote the set of subgradients (the subdifferential) of f at $\mathbf{w}$. Since the objective function is not necessarily strongly convex, the optimal solution is not necessarily unique. We denote by Ω_* the optimal solution set and by f_* the unique optimal objective value. We denote by $\|\cdot\|_2$ the Euclidean norm in $\mathbb{R}^d$.

Throughout the paper, we make the following assumption.

Assumption 1 *For the convex minimization problem (1), we assume*

- **a.** *For any $\mathbf{w}_0 \in \Omega$, we know a constant $\epsilon_0 \geq 0$ such that $f(\mathbf{w}_0) - f_* \leq \epsilon_0$.*
- **b.** *There exists a constant G such that $\max_{\mathbf{v} \in \partial f(\mathbf{w})} \|\mathbf{v}\|_2 \leq G$ for any $\mathbf{w} \in \Omega$.*

We make several remarks about the above assumptions: (i) Assumption 1.a is equivalent to assuming we know a lower bound of f_* which is one of the assumptions made by Freund and Lu (2017). In machine learning applications, f_* is usually bounded below by zero, i.e., $f_* \geq 0$, so that $\epsilon_0 = f(\mathbf{w}_0)$ for any $\mathbf{w}_0 \in \mathbb{R}^d$ will satisfy the condition; (ii) Assumption 1.b is a standard assumption also made in many previous subgradient-based methods.

Let $\mathbf{w}^*$ denote the closest optimal solution in Ω_* to $\mathbf{w}$ measured in terms of norm $\|\cdot\|_2$, i.e.,

$$\mathbf{w}^* := \arg\min_{\mathbf{u} \in \Omega_*} \|\mathbf{u} - \mathbf{w}\|_2^2.$$

Note that $\mathbf{w}^*$ is uniquely defined for any $\mathbf{w}$ due to the convexity of Ω_* and that $\|\cdot\|_2^2$ is strongly convex. We denote by $\mathcal{L}_\epsilon$ the ϵ-level set of $f(\mathbf{w})$ and by $\mathcal{S}_\epsilon$ the ϵ-sublevel set of $f(\mathbf{w})$, respectively, i.e.,

$$\mathcal{L}_\epsilon := \{\mathbf{w} \in \Omega : f(\mathbf{w}) = f_* + \epsilon\} \quad \text{and} \quad \mathcal{S}_\epsilon := \{\mathbf{w} \in \Omega : f(\mathbf{w}) \leq f_* + \epsilon\}. \tag{2}$$

Let B_ϵ be the maximum distance between the points in the ϵ-level set $\mathcal{L}_\epsilon$ and the optimal set Ω_*, i.e.,

$$B_\epsilon := \max_{\mathbf{w} \in \mathcal{L}_\epsilon} \min_{\mathbf{u} \in \Omega_*} \|\mathbf{w} - \mathbf{u}\|_2 = \max_{\mathbf{w} \in \mathcal{L}_\epsilon} \|\mathbf{w} - \mathbf{w}^*\|_2. \tag{3}$$

In the sequel, we also make the following assumption.

Assumption 2 *For the convex minimization problem (1), we assume that B_ϵ is finite.*

Remark: B_ϵ is finite when the optimal set Ω_* is bounded (e.g., when the objective function is a proper lower-semicontinuous convex and coercive function). This is because that the sublevel set $\mathcal{S}_\epsilon$ must be bounded for any $\epsilon \geq 0$ (Rockafellar, 1970, Corollary 8.7.1). Nevertheless, the bounded optimal set is not a necessary condition for a finite B_ϵ. For example, $f(x) = \max(0, x)$. Although its optimal set is not bounded, $B_\epsilon = \epsilon$. In Section 5, we will consider a broad family of problems with a local error bound condition, which will satisfy the above assumption.

Let $\mathbf{w}_\epsilon^\dagger$ denote the closest point in the ϵ-sublevel set to $\mathbf{w}$, i.e.,

$$\mathbf{w}_\epsilon^\dagger := \arg\min_{\mathbf{u}\in\mathcal{S}_\epsilon} \|\mathbf{u} - \mathbf{w}\|_2^2. \tag{4}$$

Denote by $\Omega\backslash\mathcal{S} = \{\mathbf{w} \in \Omega : \mathbf{w} \notin \mathcal{S}\}$. It is easy to show that $\mathbf{w}_\epsilon^\dagger \in \mathcal{L}_\epsilon$ when $\mathbf{w} \in \Omega\backslash\mathcal{S}_\epsilon$ (using the optimality condition of 4).

Given $\mathbf{w} \in \Omega$, we denote the normal cone of Ω at $\mathbf{w}$ by $\mathcal{N}_\Omega(\mathbf{w})$. Formally, $\mathcal{N}_\Omega(\mathbf{w}) = \{\mathbf{v} \in \mathbb{R}^d : \mathbf{v}^\top(\mathbf{u} - \mathbf{w}) \leq 0, \forall \mathbf{u} \in \Omega\}$. Define $\text{dist}(0, f(\mathbf{w}) + \mathcal{N}_\Omega(\mathbf{w}))$ as

$$\text{dist}(0, f(\mathbf{w}) + \mathcal{N}_\Omega(\mathbf{w})) := \min_{\mathbf{g}\in\partial f(\mathbf{w}), \mathbf{v}\in\mathcal{N}_\Omega(\mathbf{w})} \|\mathbf{g} + \mathbf{v}\|_2. \tag{5}$$

Note that $\mathbf{w} \in \Omega_*$ if and only if $\text{dist}(0, f(\mathbf{w}) + \mathcal{N}_\Omega(\mathbf{w})) = 0$. Therefore, we call $\text{dist}(0, f(\mathbf{w}) + \mathcal{N}_\Omega(\mathbf{w}))$ the *first-order optimality residual* of (1) at $\mathbf{w} \in \Omega$. Given any $\epsilon > 0$ such that $\mathcal{L}_\epsilon \neq \emptyset$, we define a constant ρ_ϵ as

$$\rho_\epsilon := \min_{\mathbf{w}\in\mathcal{L}_\epsilon} \text{dist}(0, f(\mathbf{w}) + \mathcal{N}_\Omega(\mathbf{w})). \tag{6}$$

Given the notations above, we provide the following lemma which is the key to our analysis.

Lemma 1 *For any $\epsilon > 0$ such that $\mathcal{L}_\epsilon \neq \emptyset$ and any $\mathbf{w} \in \Omega$, we have*

$$\|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2 \leq \frac{1}{\rho_\epsilon}(f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger)). \tag{7}$$

Proof Since the conclusion holds trivially if $\mathbf{w} \in \mathcal{S}_\epsilon$ (so that $\mathbf{w}_\epsilon^\dagger = \mathbf{w}$), we assume $\mathbf{w} \in \Omega\backslash\mathcal{S}_\epsilon$. According to the first-order optimality conditions of (4), there exist a scalar $\zeta \geq 0$ (the Lagrangian multiplier of the constraint $f(\mathbf{u}) \leq f_* + \epsilon$ in 4), a subgradient $\mathbf{g} \in \partial f(\mathbf{w}_\epsilon^\dagger)$ and a vector $\mathbf{v} \in \mathcal{N}_\Omega(\mathbf{w}_\epsilon^\dagger)$ such that

$$\mathbf{w}_\epsilon^\dagger - \mathbf{w} + \zeta\mathbf{g} + \mathbf{v} = 0. \tag{8}$$

The definition of normal cone leads to $(\mathbf{w}_\epsilon^\dagger - \mathbf{w})^\top\mathbf{v} \geq 0$. This inequality and the convexity of $f(\cdot)$ imply

$$\zeta\left(f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger)\right) \geq \zeta(\mathbf{w} - \mathbf{w}_\epsilon^\dagger)^\top\mathbf{g} \geq (\mathbf{w} - \mathbf{w}_\epsilon^\dagger)^\top(\zeta\mathbf{g} + \mathbf{v}) = \|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2^2,$$

where the equality is due to (8). Since $\mathbf{w} \in \Omega\backslash\mathcal{S}_\epsilon$, we must have $\|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2 > 0$ so that $\zeta > 0$. Therefore, $\mathbf{w}_\epsilon^\dagger \in \mathcal{L}_\epsilon$ by complementary slackness. Dividing the inequality above by ζ gives

$$f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger) \geq \frac{\|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2^2}{\zeta} = \|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2\|\mathbf{g} + \mathbf{v}/\zeta\|_2 \geq \rho_\epsilon\|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2, \tag{9}$$

where the equality is due to (8) and the last inequality is due to the definition of ρ_ϵ in (6). The lemma is then proved. ■

The inequality in (7) is the key to achieve improved convergence by RSG, which hinges on the condition that the first-order optimality residual on the ϵ-level set is lower bounded. It is important to note that (i) the above result depends on f rather than the optimization algorithm applied; and (ii) the above result can be generalized to using other norms such as the p-norm $\|\mathbf{w}\|_p$ ($p \in (1, 2]$) to measure the distance between $\mathbf{w}$ and $\mathbf{w}_\epsilon^\dagger$ and using the corresponding dual norm to define the lower bound of the residual in (5) and (6). This generalization allows one to design mirror decent (Nemirovski et al., 2009) variant of RSG. To our best knowledge, this is the first work that leverages the lower bound of the optimal residual to improve the convergence for non-smooth convex optimization.

In the next several sections, we will exhibit the value of ρ_ϵ for different classes of problems and discuss its impact on the convergence. In the sequel, we abuse the Big O notation $T = O(h(\epsilon))$ to mean that there exists a constant $C > 0$ independent of ϵ such that $T \leq Ch(\epsilon)$.

4. Restarted SubGradient (RSG) Method and Its Generic Complexity for General Problem

In this section, we present a framework of restarted subgradient (RSG) method and prove its general convergence result using Lemma 1. It will be noticed that the algorithmic results developed in this section is less interesting from the viewpoint of practice. However, it will exhibit the insights for the improvements and provide the template for the developments in next several sections, where we will present improved convergence of RSG for problems of different classes.

The steps of RSG are presented in Algorithm 2 where SG is a subroutine of projected subgradient method given in Algorithm 1 and $\Pi_\Omega[\mathbf{w}]$ is defined as

$$\Pi_\Omega[\mathbf{w}] = \arg\min_{\mathbf{u}\in\Omega} \|\mathbf{u} - \mathbf{w}\|_2^2.$$

The values of K and t in RSG will be revealed later for proving the convergence of RSG to an 2ϵ-optimal solution. The number of iterations t is the only varying parameter in RSG that depends on the classes of problems. The parameter α could be any value larger than 1 (e.g., 2) and it only has a small influence on the iteration complexity.

We emphasize that (i) RSG is a generic algorithm that is applicable to a broad family of non-smooth and/or non-strongly convex problems without changing updating schemes except for one tuning parameter, the number of iterations per stage, whose best value varies with problems; (ii) RSG has different variants with different subroutines in stages. In

Algorithm 1 SG: $\widehat{\mathbf{w}}_T = \mathrm{SG}(\mathbf{w}_1, \eta, T)$

1: **Input**: a step size η, the number of iterations T, and the initial solution $\mathbf{w}_1 \in \Omega$
2: **for** $\tau = 1, \ldots, T$ **do**
3: Query the subgradient oracle to obtain $\mathcal{G}(\mathbf{w}_\tau) \in \partial f(\mathbf{w}_\tau)$
4: Update $\mathbf{w}_{\tau+1} = \Pi_\Omega[\mathbf{w}_\tau - \eta \mathcal{G}(\mathbf{w}_\tau)]$
5: **end for**
6: **Output**: $\widehat{\mathbf{w}}_T = \sum_{\tau=1}^T \frac{\mathbf{w}_\tau}{T}$

Algorithm 2 RSG: $\mathbf{w}_K = \mathrm{RSG}(\mathbf{w}_0, K, t, \alpha)$

1: **Input**: the number of stages K and the number of iterations t per-stage, $\mathbf{w}_0 \in \Omega$, and $\alpha > 1$.
2: Set $\eta_1 = \epsilon_0/(\alpha G^2)$, where ϵ_0 is from Assumption 1.a
3: **for** $k = 1, \ldots, K$ **do**
4: Call subroutine SG to obtain $\mathbf{w}_k = \mathrm{SG}(\mathbf{w}_{k-1}, \eta_k, t)$
5: Set $\eta_{k+1} = \eta_k/\alpha$
6: **end for**
7: **Output**: $\mathbf{w}_K$

fact, we can use other optimization algorithms than SG as the subroutine in Algorithm 2, as long as a similar convergence result to Lemma 2 is guaranteed. Examples include dual averaging (Nesterov, 2009) and the regularized dual averaging (Chen et al., 2012) in the non-Euclidean space. In the following discussions, we will focus on using SG as the subroutine.

Next, we establish the convergence of RSG. It relies on the convergence result of the SG subroutine which is given in the lemma below.

Lemma 2 *(Zinkevich, 2003; Nesterov, 2004) If Algorithm 1 runs for T iterations, we have, for any $\mathbf{w} \in \Omega$,*

$$f(\widehat{\mathbf{w}}_T) - f(\mathbf{w}) \leq \frac{G^2\eta}{2} + \frac{\|\mathbf{w}_1 - \mathbf{w}\|_2^2}{2\eta T}.$$

We omit the proof because it follows a standard analysis and can be found in cited papers. With the above lemma, we can prove the following convergence of RSG.

Theorem 3 *Suppose Assumption 1 and 2 holds. If $t \geq \frac{\alpha^2 G^2}{\rho_\epsilon^2}$ and $K = \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil$ in Algorithm 2, with at most K stages, Algorithm 2 returns a solution $\mathbf{w}_K$ such that $f(\mathbf{w}_K) - f_* \leq 2\epsilon$. The total number of iterations for Algorithm 2 to find an 2ϵ-optimal solution is at most $T = t\lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil$ where $t \geq \frac{\alpha^2 G^2}{\rho_\epsilon^2}$.*

Remark: If t also satisfies $t = O\left(\frac{\alpha^2 G^2}{\rho_\epsilon^2}\right)$, then the iteration complexity of Algorithm 2 for finding an ϵ-optimal solution is $O\left(\frac{\alpha^2 G^2}{\rho_\epsilon^2} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil\right)$.

Proof

Let $\mathbf{w}_{k,\epsilon}^\dagger$ denote the closest point to $\mathbf{w}_k$ in the ϵ-sublevel set. Let $\epsilon_k := \frac{\epsilon_0}{\alpha^k}$ so that $\eta_k = \epsilon_k/G^2$ because $\eta_1 = \epsilon_0/(\alpha G^2)$ and $\eta_{k+1} = \eta_k/\alpha$. We will show by induction that

$$f(\mathbf{w}_k) - f_* \leq \epsilon_k + \epsilon, \tag{10}$$

for $k = 0, 1, \ldots, K$ which leads to our conclusion if we let $k = K$.

Note that (10) holds obviously for $k = 0$. Suppose it holds for $k-1$, namely, $f(\mathbf{w}_{k-1}) - f_* \leq \epsilon_{k-1} + \epsilon$. We want to prove (10) for k. We apply Lemma 2 to the k-th stage of Algorithm 2 and get

$$f(\mathbf{w}_k) - f(\mathbf{w}_{k-1,\epsilon}^\dagger) \leq \frac{G^2\eta_k}{2} + \frac{\|\mathbf{w}_{k-1} - \mathbf{w}_{k-1,\epsilon}^\dagger\|_2^2}{2\eta_k t}. \tag{11}$$

We now consider two cases for $\mathbf{w}_{k-1}$. First, assume $f(\mathbf{w}_{k-1}) - f_* \leq \epsilon$, i.e., $\mathbf{w}_{k-1} \in \mathcal{S}_\epsilon$. Then $\mathbf{w}_{k-1,\epsilon}^\dagger = \mathbf{w}_{k-1}$ and $f(\mathbf{w}_k) - f(\mathbf{w}_{k-1,\epsilon}^\dagger) \leq \frac{G^2\eta_k}{2} = \frac{\epsilon_k}{2}$. As a result,

$$f(\mathbf{w}_k) - f_* \leq f(\mathbf{w}_{k-1,\epsilon}^\dagger) - f_* + \frac{\epsilon_k}{2} \leq \epsilon + \epsilon_k.$$

Next, we consider the case that $f(\mathbf{w}_{k-1}) - f_* > \epsilon$, i.e., $\mathbf{w}_{k-1} \notin \mathcal{S}_\epsilon$. Then we have $f(\mathbf{w}_{k-1,\epsilon}^\dagger) = f_* + \epsilon$. By Lemma 1, we have

$$\begin{aligned} \|\mathbf{w}_{k-1} - \mathbf{w}_{k-1,\epsilon}^\dagger\|_2 &\leq \frac{1}{\rho_\epsilon}(f(\mathbf{w}_{k-1}) - f(\mathbf{w}_{k-1,\epsilon}^\dagger)) = \frac{f(\mathbf{w}_{k-1}) - f_* + (f_* - f(\mathbf{w}_{k-1,\epsilon}^\dagger))}{\rho_\epsilon} \\ &\leq \frac{\epsilon_{k-1} + \epsilon - \epsilon}{\rho_\epsilon}. \end{aligned} \tag{12}$$

Combining (11) and (12) and using the facts that $\eta_k = \frac{\epsilon_k}{G^2}$ and $t \geq \frac{\alpha^2 G^2}{\rho_\epsilon^2}$, we have

$$f(\mathbf{w}_k) - f(\mathbf{w}_{k-1,\epsilon}^\dagger) \leq \frac{\epsilon_k}{2} + \frac{\epsilon_{k-1}^2}{2\epsilon_k \alpha^2} = \epsilon_k,$$

which, together with the fact that $f(\mathbf{w}_{k-1,\epsilon}^\dagger) = f_* + \epsilon$, implies (10) for k. Therefore, by induction, we have (10) holds for $k = 1, 2, \ldots, K$ so that

$$f(\mathbf{w}_K) - f_* \leq \epsilon_K + \epsilon = \frac{\epsilon_0}{\alpha^K} + \epsilon \leq 2\epsilon,$$

where the last inequality is due to the definition of K. ■

In Theorem 3, the iteration complexity of RSG for the general problem (1) is given in terms of ρ_ϵ. Next, we show that $\rho_\epsilon \geq \frac{\epsilon}{B_\epsilon}$, which allows us to leverage the local error bound condition in next sections to upper bound B_ϵ to obtain specialized and more practical algorithms for different classes of problems.

Lemma 4 *For any $\epsilon > 0$ such that $\mathcal{L}_\epsilon \neq \emptyset$, we have $\rho_\epsilon \geq \frac{\epsilon}{B_\epsilon}$, where B_ϵ is defined in (3), and for any $\mathbf{w} \in \Omega$*

$$\|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2 \leq \frac{\|\mathbf{w}_\epsilon^\dagger - \mathbf{w}_\epsilon^*\|_2}{\epsilon}(f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger)) \leq \frac{B_\epsilon}{\epsilon}(f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger)), \tag{13}$$

where $\mathbf{w}_\epsilon^*$ is the closest point in Ω_* to $\mathbf{w}_\epsilon^\dagger$.

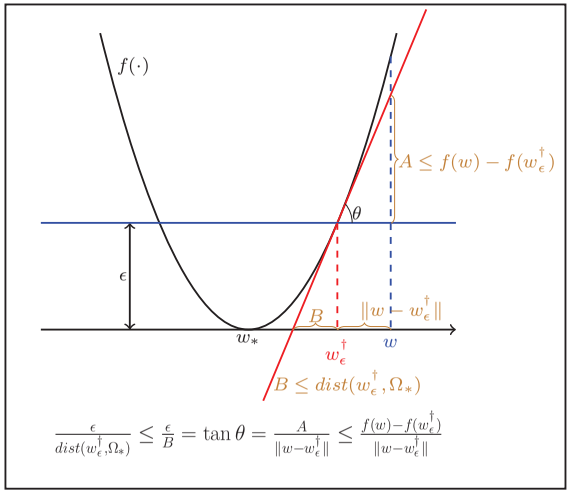


Figure 1: A geometric illustration of the inequality (13), where $dist(w_\epsilon^\dagger, \Omega_*) = |w_\epsilon^\dagger - w_\epsilon^*|$.

Proof Given any $\mathbf{u} \in \mathcal{L}_\epsilon$, let $\mathbf{g}_\mathbf{u}$ be any subgradient in $\partial f(\mathbf{u})$ and $\mathbf{v}_\mathbf{u}$ be any vector in $\mathcal{N}_\Omega(\mathbf{u})$. By the convexity of $f(\cdot)$ and the definition of normal cone, we have

$$f(\mathbf{u}^*) - f(\mathbf{u}) \ge (\mathbf{u}^* - \mathbf{u})^\top \mathbf{g}_\mathbf{u} \ge (\mathbf{u}^* - \mathbf{u})^\top (\mathbf{g}_\mathbf{u} + \mathbf{v}_\mathbf{u}),$$

where $\mathbf{u}^*$ is the closest point in Ω_* to $\mathbf{u}$. This inequality further implies

$$\|\mathbf{u}^* - \mathbf{u}\|_2 \|\mathbf{g}_\mathbf{u} + \mathbf{v}_\mathbf{u}\|_2 \ge f(\mathbf{u}) - f(\mathbf{u}^*) = \epsilon, \quad \forall \mathbf{g}_\mathbf{u} \in \partial f(\mathbf{u}) \text{ and } \mathbf{v}_\mathbf{u} \in \mathcal{N}_\Omega(\mathbf{u}), \tag{14}$$

where the equality is because $\mathbf{u} \in \mathcal{L}_\epsilon$. By (14) and the definition of B_ϵ, we obtain

$$B_\epsilon \|\mathbf{g}_\mathbf{u} + \mathbf{v}_\mathbf{u}\|_2 \ge \epsilon \Longrightarrow \|\mathbf{g}_\mathbf{u} + \mathbf{v}_\mathbf{u}\|_2 \ge \epsilon / B_\epsilon.$$

Since $\mathbf{g}_\mathbf{u} + \mathbf{v}_\mathbf{u}$ can be any element in $\partial f(\mathbf{u}) + \mathcal{N}_\Omega(\mathbf{u})$, we have $\rho_\epsilon \ge \frac{\epsilon}{B_\epsilon}$ by the definition (6).

To prove (13), we assume $\mathbf{w} \in \Omega \backslash \mathcal{S}_\epsilon$ and thus $\mathbf{w}_\epsilon^\dagger \in \mathcal{L}_\epsilon$; otherwise it is trivial. In the proof of Lemma 1, we have shown that (see (9)) there exists $\mathbf{g} \in \partial f(\mathbf{w}_\epsilon^\dagger)$ and $\mathbf{v} \in \mathcal{N}_\Omega(\mathbf{w}_\epsilon^\dagger)$ such that $f(\mathbf{w}) - f(\mathbf{w}_\epsilon^\dagger) \ge \|\mathbf{w} - \mathbf{w}_\epsilon^\dagger\|_2 \|\mathbf{g} + \mathbf{v}/\zeta\|_2$, which, according to (14) with $\mathbf{u} = \mathbf{w}_\epsilon^\dagger$, $\mathbf{g}_\mathbf{u} = \mathbf{g}$ and $\mathbf{v}_\mathbf{u} = \mathbf{v}/\zeta$, leads to (13). ■

A geometric explanation of the inequality (13) in one dimension is shown in Figure 1. With Lemma 4, the iteration complexity of RSG can be stated in terms of B_ϵ in the following corollary of Theorem 3.

Corollary 5 *Suppose Assumption 1 holds. The iteration complexity of RSG for obtaining an 2ϵ-optimal solution is $O(\frac{\alpha^2 G^2 B_\epsilon^2}{\epsilon^2} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil)$ provided $t = \frac{\alpha^2 G^2 B_\epsilon^2}{\epsilon^2}$ and $K = \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil$.*

We will compare this result with SG in Section 7. Compared to the standard SG, the above improved result of RSG does require knowing strong knolwedge about f. In particular, one issue is that the above improved complexity is obtained by choosing $t = \frac{\alpha^2 G^2 B_\epsilon^2}{\epsilon^2}$, which requires knowing the order of magnitude of B_ϵ, if not its exact value. To address the issue of unknown B_ϵ for general problems, in the next section, we consider different families of problems that admit a local error bound condition and show that the requirement of knowing B_ϵ is relaxed to knowing some particular parameters related to the local error bound.

5. RSG for Some Classes of Non-smooth Non-strongly Convex Optimization

In this section, we consider a particular family of problems that admit local error bounds and show the improved iteration complexities of RSG compared to standard SG method.

5.1 Complexity for the Problems with Local Error Bounds

We first define a local error bound condition of the objective function.

Definition 6 *We say $f(\cdot)$ admits a **local error bound** on the ϵ-sublevel set $\mathcal{S}_\epsilon$ if*

$$\|\mathbf{w} - \mathbf{w}^*\|_2 \le c(f(\mathbf{w}) - f_*)^\theta, \quad \forall \mathbf{w} \in \mathcal{S}_\epsilon, \tag{15}$$

where $\mathbf{w}^*$ is the closet point in Ω_* to $\mathbf{w}$, $\theta \in (0, 1]$ and $0 < c < \infty$ are constants.

Because $\mathcal{S}_{\epsilon_2} \subset \mathcal{S}_{\epsilon_1}$ for $\epsilon_2 \le \epsilon_1$, if (15) holds for some ϵ, it will always hold when ϵ decreases to zero with the same θ and c. Indeed, a smaller ϵ may induce a smaller value of c. It is notable that the local error bound condition has been extensively studied in the community of optimization, mathematical programming and variational analysis (Yang, 2009; Li, 2010, 2013; Artacho and Geoffroy, 2008; Kruger, 2015; Drusvyatskiy et al., 2014; Li and Mordukhovich, 2012; Hou et al., 2013; Zhou and So, 2017; Zhou et al., 2015), to name just a few of them. The value of θ has been exhibited for many problems. For certain problems, the value of c is also computable (see Bolte et al., 2017).

If the problem admits a local error bound like (15), RSG can achieve a better iteration complexity than $O(1/\epsilon^2)$. In particular, the property (15) implies

$$B_\epsilon \le c\epsilon^\theta. \tag{16}$$

Replacing B_ϵ in Corollary 5 by this upper bound and choosing $t = \frac{\alpha^2 G^2 c^2}{\epsilon^{2(1-\theta)}}$ in RSG if c and θ are known, we obtain the following complexity of RSG.

Corollary 7 *Suppose Assumption 1 holds and $f(\cdot)$ admits a local error bound on $\mathcal{S}_\epsilon$. The iteration complexity of RSG for obtaining an 2ϵ-optimal solution is $O\left(\frac{\alpha^2 G^2 c^2}{\epsilon^{2(1-\theta)}} \log_\alpha\left(\frac{\epsilon_0}{\epsilon}\right)\right)$ provided $t = \frac{\alpha^2 G^2 c^2}{\epsilon^{2(1-\theta)}}$ and $K = \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil$.*

Remark: If $t = \Theta(\frac{\alpha^2 G^2}{\epsilon^{2(1-\theta)}}) > \frac{\alpha^2 G^2 c^2}{\epsilon^{2(1-\theta)}}$, then the same order of iteration complexity remains. If one aims to find a point $\mathbf{w}$ such that $\|\mathbf{w} - \mathbf{w}^*\|_2 \le \epsilon$, we can apply RSG to find a solution $\mathbf{w}$

such that $f(\mathbf{w}) - f_* \leq (\epsilon/c)^{1/\theta} \leq \epsilon$ (where the last inequality is due to $\theta \leq 1$ and assuming $c \geq 1$ without loss of generality). Then under the local error bound condition, we have $\|\mathbf{w}-\mathbf{w}^*\|_2 \leq c(f(\mathbf{w})-f_*)^\theta \leq \epsilon$. For finding a solution $\mathbf{w}$ such that $f(\mathbf{w})-f_* \leq (\epsilon/c)^{1/\theta} \leq \epsilon$, RSG requires an iteration complexity of $\widetilde{O}(\frac{1}{\epsilon^{2(1-\theta)/\theta}})$. Therefore, in order to find a solution $\mathbf{w}$ such that $\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \epsilon$, the iteration complexity of RSG is $\widetilde{O}(\frac{1}{\epsilon^{2(1-\theta)/\theta}})$.

Next, we will consider different convex optimization problems that admit a local error bound on $\mathcal{S}_\epsilon$ with different θ and show the faster convergence of RSG when applied to these problems.

5.2 Linear Convergence for Polyhedral Convex Optimization

In this subsection, we consider a special family of non-smooth and non-strongly convex problems where the epigraph of $f(\cdot)$ over Ω is a polyhedron. In this case, we call (1) a **polyhedral convex minimization** problem. We show that, in polyhedral convex minimization problem, $f(\cdot)$ has a linear growth property and admits a local error bound with $\theta = 1$ so that $B_\epsilon \leq c\epsilon$ for a constant $c < \infty$.

Lemma 8 (Polyhedral Error Bound Condition) *Suppose Ω is a polyhedron and the epigraph of $f(\cdot)$ is also polyhedron. There exists a constant $\kappa > 0$ such that*

$$\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \frac{f(\mathbf{w}) - f_*}{\kappa}, \quad \forall \mathbf{w} \in \Omega.$$

Thus, $f(\cdot)$ admits a local error bound on $\mathcal{S}_\epsilon$ with $\theta = 1$ and $c = \frac{1}{\kappa}$ [3] (so $B_\epsilon \leq \frac{\epsilon}{\kappa}$) for any $\epsilon > 0$.

Remark: The above inequality is also known as weak sharp minimum condition in literature (Burke and Ferris., 1993; Studniarski and Ward, 1999; Ferris, 1991; Burke and Deng, 2002, 2005, 2009). A proof of Lemma 8 is given by Burke and Ferris. (1993). We also provide a proof (see Yang and Lin, 2016). We remark that the above result can be extended to any valid norm to measure the distance between $\mathbf{w}$ and $\mathbf{w}_*$. Lemma 8 generalizes Lemma 4 of Gilpin et al. (2012), which requires Ω to be a bounded polyhedron, to a similar result where Ω can be an unbounded polyhedron. This generalization is simple but useful because it helps the development of efficient algorithms based on this error bound for unconstrained problems without artificially including a box constraint.

Lemma 8 provides the basis for RSG to achieve a linear convergence for the polyhedral convex minimization problems. In fact, the following linear convergence of RSG can be obtained if we plugin the values of $\theta = 1$ and $c = \frac{1}{\kappa}$ into Corollary 7.

Corollary 9 *Suppose Assumption 1 holds and (1) is a polyhedral convex minimization problem. The iteration complexity of RSG for obtaining an ϵ-optimal solution is $O(\frac{\alpha^2 G^2}{\kappa^2}\lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil)$ provided $t = \frac{\alpha^2 G^2}{\kappa^2}$ and $K = \lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil$.*

We want to point out that Corollary 9 can be proved directly by replacing $\mathbf{w}^\dagger_{k-1,\epsilon}$ by $\mathbf{w}^*_{k-1}$ and replacing ρ_ϵ by κ in the proof of Theorem 3. Here, we derive it as a corollary of a more general result. We also want to mention that, as shown by Renegar (2015), the linear convergence rate in Corollary 9 can be also obtained by a SG method for the historically best solution, provided f_* is known.

3. In fact, this property of $f(\cdot)$ is a global error bound on Ω.

5.2.1 Examples

Many non-smooth and non-strongly convex machine learning problems satisfy the assumptions of Corollary 9, for example, ℓ_1 **or** ℓ_∞ **constrained or regularized piecewise linear loss minimization**. In many machine learning tasks (e.g., classification and regression), there exists a set of data $\{(\mathbf{x}_i, y_i)\}_{i=1,2,\ldots,n}$ and one often needs to solve the following empirical risk minimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d} f(\mathbf{w}) \triangleq \frac{1}{n}\sum_{i=1}^n \ell(\mathbf{w}^\top\mathbf{x}_i, y_i) + R(\mathbf{w}),$$

where $R(\mathbf{w})$ is a regularization term and $\ell(z, y)$ denotes a loss function. We consider a special case where (a) $R(\mathbf{w})$ is a ℓ_1 regularizer, ℓ_∞ regularizer or an indicator function of a ℓ_1/ℓ_∞ ball centered at zero; and (b) $\ell(z, y)$ is any piecewise linear loss function, including hinge loss $\ell(z, y) = \max(0, 1 - yz)$, absolute loss $\ell(z, y) = |z - y|$, ϵ-insensitive loss $\ell(z, y) = \max(|z - y| - \epsilon, 0)$, and etc (Yang et al., 2014). It is easy to show that the epigraph of $f(\mathbf{w})$ is a polyhedron if $f(\mathbf{w})$ is defined as a sum of any of these regularization terms and any of these loss functions. In fact, a piecewise linear loss functions can be generally written as

$$\ell(\mathbf{w}^\top\mathbf{x}, y) = \max_{1\leq j\leq m} a_j\mathbf{w}^\top\mathbf{x} + b_j, \tag{17}$$

where (a_j, b_j) for $j = 1, 2, \ldots, m$ are finitely many pairs of scalars. The formulation (17) indicates that $\ell(\mathbf{w}^\top\mathbf{x}, y)$ is a piecewise affine function so that its epigraph is a polyhedron. In addition, the ℓ_1 or ℓ_∞ norm is also a polyhedral function because we can represent them as

$$\|\mathbf{w}\|_1 = \sum_{i=1}^d \max(w_i, -w_i), \quad \|\mathbf{w}\|_\infty = \max_{1\leq i\leq d}|w_i| = \max_{1\leq i\leq d}\max(w_i, -w_i).$$

Since the sum of finitely many polyhedral functions is also a polyhedral function, the epigraph of $f(\mathbf{w})$ is a polyhedron.

Another important family of problems whose objective function has a polyhedral epigraph is **submodular function minimization**. Let $V = \{1, \ldots, d\}$ be a set and 2^V denote its power set. A submodular function $F(A) : 2^V \to \mathbb{R}$ is a set function such that $F(A)+F(B) \geq F(A\cup B)+F(A\cap B)$ for all subsets $A, B \subseteq V$ and $F(\emptyset) = 0$. A submodular function minimization can be cast into a non-smooth convex optimization using the Lovász extension (Bach, 2013). In particular, let the base polyhedron $B(F)$ be defined as

$$B(F) = \{\mathbf{s} \in \mathbb{R}^d, \mathbf{s}(V) = F(V), \forall A \subseteq V, \mathbf{s}(A) \leq F(A)\},$$

where $\mathbf{s}(A) = \sum_{i\in A} s_i$. Then the Lovász extension of $F(A)$ is $f(\mathbf{w}) = \max_{\mathbf{s}\in B(F)} \mathbf{w}^\top\mathbf{s}$, and $\min_{A\subseteq V} F(A) = \min_{\mathbf{w}\in[0,1]^d} f(\mathbf{w})$. As a result, a submodular function minimization is essentially a non-smooth and non-strongly convex optimization with a polyhedral epigraph.

5.3 Improved Convergence for Locally Semi-Strongly Convex Problems

First, we give a definition of local semi-strong convexity.

Definition 10 *A function $f(\mathbf{w})$ is semi-strongly convex on the ϵ-sublevel set $\mathcal{S}_\epsilon$ if there exists $\lambda > 0$ such that*

$$\frac{\lambda}{2}\|\mathbf{w} - \mathbf{w}^*\|_2^2 \leq f(\mathbf{w}) - f(\mathbf{w}^*), \quad \forall \mathbf{w} \in \mathcal{S}_\epsilon, \tag{18}$$

where $\mathbf{w}^*$ is the closest point to $\mathbf{w}$ in the optimal set.

We refer to the property (18) as *local* semi-strong convexity when $\mathcal{S}_\epsilon \neq \Omega$. The two papers (Gong and Ye, 2014; Necoara et al., 2015) have explored the semi-strong convexity on the whole domain Ω to prove linear convergence of smooth optimization problems. In some literature (Necoara et al., 2015), the inequality (18) is also called **second-order growth property**. Necoara et al. (2015) have also shown that a class of problems satisfy (18) (see examples given below). The inequality (18) indicates that $f(\cdot)$ admits a local error bound on $\mathcal{S}_\epsilon$ with $\theta = \frac{1}{2}$ and $c = \sqrt{\frac{2}{\lambda}}$, which leads to the following the corollary about the iteration complexity of RSG for locally semi-strongly convex problems.

Corollary 11 *Suppose Assumption 1 holds and $f(\mathbf{w})$ is semi-strongly convex on $\mathcal{S}_\epsilon$. Then $B_\epsilon \leq \sqrt{\frac{2\epsilon}{\lambda}}$ [4] and the iteration complexity of RSG for obtaining an 2ϵ-optimal solution is $O(\frac{2\alpha^2G^2}{\lambda\epsilon}\lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil)$ provided $t = \frac{2\alpha^2G^2}{\lambda\epsilon}$ and $K = \lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil$.*

Remark: Here, we obtain an $\widetilde{O}(1/\epsilon)$ iteration complexity ($\widetilde{O}(\cdot)$ suppresses constants and logarithmic terms) only with local semi-strong convexity. It is obvious that strong convexity implies local semi-strong convexity (Hazan and Kale, 2011) but not vice versa.

For examples, let us consider a family of functions in the form of $f(\mathbf{w}) = h(X\mathbf{w}) + r(\mathbf{w})$, where $X \in \mathbb{R}^{n\times d}$, $h(\cdot)$ is *strongly convex* on any *compact set* and $r(\cdot)$ has a polyhedral epigraph. According to (Gong and Ye, 2014; Necoara et al., 2015), such a function $f(\mathbf{w})$ satisfies (18) for any $\epsilon \leq \epsilon_0$ with a constant value for λ. Although smoothness is assumed for $h(\cdot)$ in (Gong and Ye, 2014; Necoara et al., 2015), we find that it is not necessary for proving (18). We state this result as the lemma below.

Lemma 12 *Suppose Assumption 1 holds, $\Omega = \{\mathbf{w} \in \mathbb{R}^d | C\mathbf{w} \leq \mathbf{b}\}$ with $C \in \mathbb{R}^{k\times d}$ and $\mathbf{b} \in \mathbb{R}^k$, and $f(\mathbf{w}) = h(X\mathbf{w}) + r(\mathbf{w})$ where $h : \mathbb{R}^n \to \mathbb{R}$ satisfies $dom(h) = \mathbb{R}^k$ and is a strongly convex function on any compact set in $\mathbb{R}^n$, and $r(\mathbf{w})$ has a polyhedral epigraph. Then, $f(\mathbf{w})$ satisfies (18) for any $\epsilon \leq \epsilon_0$.*

The proof of this lemma can be duplicated following analysis in some existing works (Gong and Ye, 2014; Necoara et al., 2015; Necoara and Clipici, 2016). For example, it is almost identical to the proof of Lemma 1 by Gong and Ye (2014) which assumes $h(\cdot)$ is smooth. However, a similar result holds without the smoothness of $h(\cdot)$.

4. Recall (16).

The function of this type covers some commonly used loss functions and regularization terms in machine learning and statistics. For example, we can consider **robust regression with/without ℓ_1 regularizer** (Xu et al., 2010; Bertsimas and Copenhaver, 2014):

$$\min_{\mathbf{w}\in\Omega} \frac{1}{n}\sum_{i=1}^n |\mathbf{x}_i^\top\mathbf{w} - y_i|^p + \lambda\|\mathbf{w}\|_1, \tag{19}$$

where $p \in (1, 2)$, $\mathbf{x}_i \in \mathbb{R}^d$ denotes the feature vector and y_i is the target output. The objective function is in the form of $h(X\mathbf{w}) + r(\mathbf{w})$ where X is a $n \times d$ matrix with $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$ being its rows and $h(\mathbf{u}) := \sum_{i=1}^n |u_i - y_i|^p$. According to Goebel and Rockafellar (2007), $h(\mathbf{u})$ is a strongly convex function on any compact set so that the objective function above is semi-strongly convex on $\mathcal{S}_\epsilon$ for any $\epsilon \leq \epsilon_0$.

5.4 Improved Convergence for Convex Problems with KL property

Lastly, we consider a family of non-smooth functions with a local Kurdyka-Łojasiewicz (KL) property. The definition of KL property is given below.

Definition 13 *The function $f(\mathbf{w})$ has the Kurdyka - Łojasiewicz (KL) property at $\bar{\mathbf{w}}$ if there exist $\eta \in (0, \infty]$, a neighborhood $U_{\bar{\mathbf{w}}}$ of $\bar{\mathbf{w}}$ and a continuous concave function $\varphi : [0, \eta) \to \mathbb{R}_+$ such that (i) $\varphi(0) = 0$; (ii) φ is continuous on $(0, \eta)$; (iii) for all $s \in (0, \eta)$, $\varphi'(s) > 0$; (iv) and for all $\mathbf{w} \in U_{\bar{\mathbf{w}}} \cap \{\mathbf{w} : f(\bar{\mathbf{w}}) < f(\mathbf{w}) < f(\bar{\mathbf{w}}) + \eta\}$, the Kurdyka - Łojasiewicz (KL) inequality holds*

$$\varphi'(f(\mathbf{w}) - f(\bar{\mathbf{w}}))\|\partial f(\mathbf{w})\|_2 \geq 1, \tag{20}$$

where $\|\partial f(\mathbf{w})\|_2 := \min_{\mathbf{g}\in\partial f(\mathbf{w})} \|\mathbf{g}\|_2$.

The function φ is called the **desingularizing function** of f at $\bar{\mathbf{w}}$, which sharpens the function $f(\mathbf{w})$ by reparameterization. An important desingularizing function is in the form of $\varphi(s) = cs^{1-\beta}$ for some $c > 0$ and $\beta \in [0, 1)$, by which, (20) gives the KL inequality

$$\|\partial f(\mathbf{w})\|_2 \geq \frac{1}{c(1-\beta)}(f(\mathbf{w}) - f(\bar{\mathbf{w}}))^\beta.$$

Note that all semi-algebraic functions satisfy the KL property at any point (Bolte et al., 2014). Indeed, all the concrete examples given before satisfy the Kurdyka - Łojasiewicz property. For more discussions about the KL property, we refer readers to some previous works (Bolte et al., 2014, 2007; Schneider and Uschmajew, 2015; Attouch et al., 2013; Bolte et al., 2006). The following corollary states the iteration complexity of RSG for unconstrained problems that have the KL property at each $\bar{\mathbf{w}} \in \Omega_*$.

Corollary 14 *Suppose Assumption 1 holds, $f(\mathbf{w})$ satisfies a (uniform) Kurdyka - Łojasiewicz property at any $\bar{\mathbf{w}} \in \Omega_*$ with the same desingularizing function φ and constant η, and*

$$\mathcal{S}_\epsilon \subset \cup_{\bar{\mathbf{w}}\in\Omega_*} \left[U_{\bar{\mathbf{w}}} \cap \{\mathbf{w} : f(\bar{\mathbf{w}}) < f(\mathbf{w}) < f(\bar{\mathbf{w}}) + \eta\}\right]. \tag{21}$$

RSG has an iteration complexity of $O\left(\alpha^2G^2(\frac{\varphi(\epsilon)}{\epsilon})^2\lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil\right)$ for obtaining an 2ϵ-optimal solution provided $t = \alpha^2G^2(\varphi(\epsilon)/\epsilon)^2$. In addition, if $\varphi(s) = cs^{1-\beta}$ for some $c > 0$ and $\beta \in [0, 1)$, the iteration complexity of RSG is $O(\frac{\alpha^2G^2c^2(1-\beta)^2}{\epsilon^{2\beta}}\lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil)$ provided $t = \frac{\alpha^2G^2c^2}{\epsilon^{2\beta}}$ and $K = \lceil\log_\alpha(\frac{\epsilon_0}{\epsilon})\rceil$.

Proof We can prove the above corollary following a result by Bolte et al. (2017) as presented in Proposition 1 in the Appendix. According to Proposition 1, if $f(\cdot)$ satisfies the KL property at $\bar{\mathbf{w}}$, then for all $\mathbf{w} \in U_{\bar{\mathbf{w}}} \cap \{\mathbf{w} : f(\bar{\mathbf{w}}) < f(\mathbf{w}) < f(\bar{\mathbf{w}}) + \eta\}$ it holds that $\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \varphi(f(\mathbf{w}) - f(\bar{\mathbf{w}}))$. It then, under the uniform condition in (21), implies that, for any $\mathbf{w} \in \mathcal{S}_\epsilon$

$$\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \varphi(f(\mathbf{w}) - f_*) \leq \varphi(\epsilon),$$

where we use the monotonic property of φ. Then the first conclusion follows similarly as Corollary 5 by noting $B_\epsilon \leq \varphi(\epsilon)$. The second conclusion immediately follows by setting $\varphi(s) = cs^{1-\beta}$ in the first conclusion. Please note that the above inequality implies the local error bound condition with $\theta = 1 - \beta$ for $\varphi(s) = cs^{1-\beta}$. ■

While the conclusion in Corollary 14 hinges on a condition in (21) for certain $U_{\bar{\mathbf{w}}}$ and η, in practice many convex functions (e.g., continuous semi-algebraic or subanalytic functions) satisfy the KL property with $U = \mathbb{R}^d$ and any finite $\eta < \infty$ (Attouch et al., 2010; Bolte et al., 2017; Li, 2010).

It is worth mentioning that to our best knowledge, the present work is the first to leverage the KL property for developing improved subgradient methods, though it has been explored in non-convex and convex optimization for deterministic descent methods for smooth optimization (Bolte et al., 2017, 2014; Attouch et al., 2010; Karimi et al., 2016). For example, Bolte et al. (2017) studied the convergence of **subgradient descent sequence** for minimizing a convex function under an error bound condition. A sequence $\{\mathbf{x}_k\}$ is called a subgradient descent sequence if there exist $a > 0, b > 0$ it satisfies two conditions, namely sufficient decrease condition $f(\mathbf{x}_k) + a\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2^2 \leq f(\mathbf{x}_{k-1})$, and relative error condition, i.e., there exists $\omega_k \in \partial f(\mathbf{x}_k)$ such that $\|\omega_k\|_2 \leq b\|\mathbf{x}_k - \mathbf{x}_{k-1}\|_2$. However, for a general non-smooth function $f(\mathbf{x})$, the sequence generated by subgradient method, i.e., $\mathbf{x}_k = \mathbf{x}_{k-1} - \eta_k \partial \nabla f(\mathbf{x}_{k-1})$ do not necessarily satisfy the above two conditions. Instead, Bolte et al. (2017) considered proximal gradient method that only applies to a smaller family of functions consisting of a smooth component and a non-smooth component by assuming the proximal mapping for the non-smooth component can be efficiently computed. In contrast, our algorithm and analysis are developed for much general non-smooth functions.

6. Variants of RSG without knowing the constant c and the exponent θ in the local error bound

In Section 5, we have discussed the local error bound and presented several classes of problems to reveal the magnitude of B_ϵ, i.e., $B_\epsilon = c\epsilon^\theta$. For some problems, the value of θ is exhibited. However, the value of the constant c could be still difficult to estimate, which renders it challenging to set the appropriate value $t = \frac{\alpha^2 c^2 G^2}{\epsilon^{2(1-\theta)}}$ for inner iterations of RSG. In practice, one might use a sufficiently large c to set up the value of t. However, such an approach might be vulnerable to both over-estimation and under-estimation of t. Over-estimating the value of t leads to a waste of iterations while under-estimation leads to an less accurate solution that might not reach to the target accuracy level. In addition, for some problems the value of θ is still an open problem. One interesting family of objective functions in machine learning is the sum of piecewise linear loss over training data and a nuclear norm regularizer or an overlapped or non-overlapped group lasso regularizer. In this section, we present variants of RSG that can be implemented without knowing the value of c in the local error bound condition and even the value of exponent θ, and prove their improved convergence over the SG method.

6.1 RSG without knowing c

The key idea is to use an increasing sequence of t and another level of restarting for RSG. The detailed steps are presented in Algorithm 3, to which we refer as R^{2}SG. With large enough t_1 in R^{2}SG, the complexity of R^{2}SG for finding an ϵ solution is given by the theorem below.

Theorem 15 *Suppose $\epsilon \leq \epsilon_0/4$ and $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil$. Let t_1 in Algorithm 3 be large enough so that there exists $\hat{\epsilon}_1 \in (\epsilon, \epsilon_0/2)$, with which $f(\cdot)$ satisfies a local error bound condition on $\mathcal{S}_{\hat{\epsilon}_1}$ with $\theta \in (0,1)$ and the constant $\hat{c}$, and $t_1 = \frac{\alpha^2 \hat{c}^2 G^2}{\hat{\epsilon}_1^{2(1-\theta)}}$. Then, with at most $S = \lceil \log_2(\hat{\epsilon}_1/\epsilon) \rceil + 1$ calls of RSG in Algorithm 3, we find a solution $\mathbf{w}^S$ such that $f(\mathbf{w}^S) - f_* \leq 2\epsilon$. The total number of iterations of R^{2}SG for obtaining 2ϵ-optimal solution is upper bounded by $T_S = O\left(\frac{\hat{c}^2 G^2}{\epsilon^{2(1-\theta)}} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil\right)$.*

Proof Since $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil \geq \lceil \log_\alpha(\epsilon_0/\hat{\epsilon}_1) \rceil$ and $t_1 = \frac{\alpha^2 \hat{c}^2 G^2}{\hat{\epsilon}_1^{2(1-\theta)}}$, we can apply Corollary 7 with $\epsilon = \hat{\epsilon}_1$ to the first call of RSG in Algorithm 3 so that the output $\mathbf{w}^1$ satisfies

$$f(\mathbf{w}^1) - f_* \leq 2\hat{\epsilon}_1. \tag{22}$$

Then, we consider the second call of RSG with the initial solution $\mathbf{w}^1$ satisfying (22). By the setup $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil \geq \lceil \log_\alpha(2\hat{\epsilon}_1/(\hat{\epsilon}_1/2)) \rceil$ and $t_2 = t_1 2^{2(1-\theta)} = \frac{\hat{c}^2 G^2}{(\hat{\epsilon}_1/2)^{2(1-\theta)}}$, we can apply Corollary 7 with $\epsilon = \hat{\epsilon}_1/2$ and $\epsilon_0 = 2\hat{\epsilon}_1$ so that the output $\mathbf{w}^2$ of the second call satisfies $f(\mathbf{w}^2) - f_* \leq \hat{\epsilon}_1$. By repeating this argument for all the subsequent calls of RSG, with at most $S = \lceil \log_2(\hat{\epsilon}_1/\epsilon) \rceil + 1$ calls, Algorithm 3 ensures that

$$f(\mathbf{w}^S) - f_* \leq 2\hat{\epsilon}_1 / 2^{S-1} \leq 2\epsilon.$$

The total number of iterations during the S calls of RSG is bounded by

$$\begin{aligned} T_S &= K \sum_{s=1}^{S} t_s = K \sum_{s=1}^{S} t_1 2^{2(s-1)(1-\theta)} = K t_1 2^{2(S-1)(1-\theta)} \sum_{s=1}^{S} \left(\frac{1}{2^{2(1-\theta)}}\right)^{S-s} \\ &\leq \frac{K t_1 2^{2(S-1)(1-\theta)}}{1 - 1/2^{2(1-\theta)}} \leq O\left(K t_1 \left(\frac{\hat{\epsilon}_1}{\epsilon}\right)^{2(1-\theta)}\right) = O\left(\frac{\hat{c}^2 G^2}{\epsilon^{2(1-\theta)}} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil\right). \end{aligned}$$

■

Remark: We make several remarks about Algorithm 3 and Theorem 15: (i) Theorem 15 applies only when $\theta \in (0,1)$. If $\theta = 1$, in order to have an increasing sequence of t_s, we can set θ in Algorithm 3 to a little smaller value than 1 in practical implementation, and the

Algorithm 3 RSG with restarting: R^{2}SG

1: **Input**: the number of iterations t_1 in each stage of the first call of RSG and the number of stages K in each call of RSG
2: **Initialization:** $\mathbf{w}^0 \in \Omega$;
3: **for** $s = 1, 2 \ldots, S$ **do**
4: Let $\mathbf{w}^s = \text{RSG}(\mathbf{w}^{s-1}, K, t_s, \alpha)$
5: Let $t_{s+1} = t_s 2^{2(1-\theta)}$
6: **end for**

iteration complexity in Theorem 15 implies that R^{2}SG can enjoy a convergence rate close to linear convergence for problems satisfying the weak sharp minimum condition. (ii) the ϵ_0 in the implementation of RSG (Algorithm 2) can be re-calibrated for $s \geq 2$ to improve the performance (e.g., one can use the relationship $f(\mathbf{w}_{s-1}) - f_* = f(\mathbf{w}_{s-2}) - f_* + f(\mathbf{w}_{s-1}) - f(\mathbf{w}_{s-2})$ to do re-calibration); (iii) as a tradeoff, the exiting criterion of R^{2}SG is not as automatic as RSG. In fact, the total number of calls S of RSG for obtaining an 2ϵ-optimal solution depends on an unknown parameter (namely $\hat{\epsilon}_1$). In practice, one could use other stopping criteria to terminate the algorithm. For example, in machine learning applications one can monitor the performance on the validation data set to terminate the algorithm. (vi) The quantities $\hat{\epsilon}_1$, S in the proof above are implicitly determined by t_1 and one does not need to compute $\hat{\epsilon}_1$ and S in order to apply Algorithm 3. Finally, we note that when a local strong convexity condition holds on $\mathcal{S}_{\hat{\epsilon}_1}$ with $\hat{\epsilon}_1 \geq \epsilon$ one might derive an iteration complexity of $O(1/\epsilon)$ for SG by first showing that SG converges to $\mathcal{S}_{\hat{\epsilon}_1}$ with a number of iterations independent of ϵ, then showing that the iterates stay within $\mathcal{S}_{\hat{\epsilon}_1}$ and converge to an ϵ-level set with an iteration complexity of $O(1/\epsilon)$ following existing analysis of SG for strongly convex functions, e.g., (Lacoste-Julien et al., 2012). However, it still needs to know the value of the local strong convexity parameter unlike our result in Theorem 15 that does not need to known the local strong convexity parameter.

6.2 RSG for unknown θ and c

Without knowing $\theta \in (0, 1]$ and c to get a sharper local error bound, we can simply let $\theta = 0$ and $c = B_{\epsilon'}$ with $\epsilon' \geq \epsilon$, which still render the inequaity (15) hold (c.f. Definition 6). Then we can employ the same trick to increase the values of t. In particular, we start with a sufficiently large value of t and run RSG with $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil$ stages, and then increase the value of t by a factor of 4 and repeat the process.

Theorem 16 *Let $\theta = 0$ in Algorithm 3 and suppose $\epsilon \leq \epsilon_0/4$ and $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil$. Assume t_1 in Algorithm 3 is large enough so that there exists $\hat{\epsilon}_1 \in (\epsilon, \epsilon_0/2]$ giving $t_1 = \frac{\alpha^2 B^2_{\hat{\epsilon}_1} G^2}{\hat{\epsilon}_1^2}$. Then, with at most $S = \lceil \log_2(\hat{\epsilon}_1/\epsilon) \rceil + 1$ calls of RSG in Algorithm 3, we find a solution $\mathbf{w}^S$ such that $f(\mathbf{w}^S) - f_* \leq 2\epsilon$. The total number of iterations of R^{2}SG for obtaining 2ϵ-optimal solution is upper bounded by $T_S = O\left(\frac{B^2_{\hat{\epsilon}_1} G^2}{\epsilon^2} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil\right)$.*

Remark: Since B_ϵ/ϵ is a monotonically decreasing function in ϵ (Xu et al., 2017, Lemma 7), such a t_1 in Theorem 16 exists. Note that if the problem satisfies a KL property as in Corollary 14 and the value of β is unknown, the above theorem still holds.

Proof The proof is similar to that of Theorem 15 except that we let $c = B_{\hat{\epsilon}_1}$ and $\theta = 0$. Since $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil \geq \lceil \log_\alpha(\epsilon_0/\hat{\epsilon}_1) \rceil$ and $t_1 = \frac{\alpha^2 B^2_{\hat{\epsilon}_1} G^2}{\hat{\epsilon}_1^2}$, we can apply Corollary 5 with $\epsilon = \hat{\epsilon}_1$ to the first call of RSG in Algorithm 3 so that the output $\mathbf{w}^1$ satisfies

$$f(\mathbf{w}^1) - f_* \leq 2\hat{\epsilon}_1. \tag{23}$$

Then, we consider the second call of RSG with the initial solution $\mathbf{w}^1$ satisfying (23). By the setup $K = \lceil \log_\alpha(\epsilon_0/\epsilon) \rceil \geq \lceil \log_\alpha(2\hat{\epsilon}_1/(\hat{\epsilon}_1/2)) \rceil$ and $t_2 = t_1 2^2 = \frac{B^2_{\hat{\epsilon}_1} G^2}{(\hat{\epsilon}_1/2)^2}$, we can apply Corollary 5 with $\epsilon = \hat{\epsilon}_1/2$ and $\epsilon_0 = 2\hat{\epsilon}_1$ (noting that $B_{\hat{\epsilon}_1} > B_{\hat{\epsilon}_1/2}$) so that the output $\mathbf{w}^2$ of the second call satisfies $f(\mathbf{w}^2) - f_* \leq \hat{\epsilon}_1$. By repeating this argument for all the subsequent calls of RSG, with at most $S = \lceil \log_2(\hat{\epsilon}_1/\epsilon) \rceil + 1$ calls, Algorithm 3 ensures that

$$f(\mathbf{w}^S) - f_* \leq 2\hat{\epsilon}_1/2^{S-1} \leq 2\epsilon.$$

The total number of iterations during the S calls of RSG is bounded by

$$\begin{aligned} T_S = K\sum_{s=1}^{S} t_s = K\sum_{s=1}^{S} t_1 2^{2(s-1)} = K t_1 2^{2(S-1)} \sum_{s=1}^{S} \left(\frac{1}{2^2}\right)^{S-s} \\ \leq \frac{K t_1 2^{2(S-1)}}{1 - 1/2^2} \leq O\left(K t_1 \left(\frac{\hat{\epsilon}_1}{\epsilon}\right)^2\right) = O\left(\frac{B^2_{\hat{\epsilon}_1} G^2}{\epsilon^2} \lceil \log_\alpha(\frac{\epsilon_0}{\epsilon}) \rceil\right). \end{aligned}$$

■

7. Discussions and Comparisons

In this section, we further discuss the obtained results and compare them with existing results.

7.1 Comparison with the standard SG

The standard SG's iteration complexity is known as $O(\frac{G^2 \|\mathbf{w}_0 - \mathbf{w}_0^*\|_2^2}{\epsilon^2})$ for achieving an 2ϵ-optimal solution. By assuming t is appropriately set in RSG according to Corollary 5, its iteration complexity is $O(\frac{G^2 B_\epsilon^2}{\epsilon^2} \log(\epsilon_0/\epsilon))$, which depends on B_ϵ^2 instead of $\|\mathbf{w}_0 - \mathbf{w}_0^*\|_2^2$ and only has a logarithmic dependence on ϵ_0, the upper bound of $f(\mathbf{w}_0) - f_*$. When the initial solution is far from the optimal set so that $B_\epsilon^2 \ll \|\mathbf{w}_0 - \mathbf{w}_0^*\|_2^2$, RSG could have a lower worst-case complexity. Even if t is not appropriately set up to be larger than $\alpha^2 G^2 B_\epsilon^2/\epsilon^2$, Theorem 16 guarantees that the proposed R^{2}SG could still has a lower iteration complexity than that of SG as long as t_1 is sufficiently large. In some special cases, e.g., when f satisfies the local error bound condition (15) with $\theta \in (0, 1]$, RSG only needs $O\left(\frac{1}{\epsilon^{2(1-\theta)}} \log\left(\frac{1}{\epsilon}\right)\right)$ iterations (see Corollary 7 and Theorem 15), which has a better dependency on ϵ than the complexity of standard SG method.

7.2 Comparison with the SG method by Freund and Lu (2017)

Freund and Lu (2017) introduced a similar but different growth condition:

$$\|\mathbf{w} - \mathbf{w}^*\|_2 \leq \mathcal{G} \cdot (f(\mathbf{w}) - f_{slb}), \quad \forall \mathbf{w} \in \Omega, \tag{24}$$

where f_{slb} is a strict lower bound of f_*. The main differences from our key condition (7) are: the left-hand side is the distance of $\mathbf{w}$ to the optimal set in (24) while it is the distance of $\mathbf{w}$ to the ϵ-sublevel set in (7); the right-hand side is the objective gap with respect to f_{slb} in (24) and it is the objective gap with respect to f_* in (7); the growth constant $\mathcal{G}$ in (24) varies with f_{slb} and ρ_ϵ in (7) may depend on ϵ in general.

Freund and Lu's SG method has an iteration complexity of $O(G^2\mathcal{G}^2(\frac{\log H}{\epsilon'} + \frac{1}{\epsilon'^2}))$ for finding a solution $\hat{\mathbf{w}}$ such that $f(\hat{\mathbf{w}}) - f_* \leq \epsilon'(f_* - f_{slb})$, where f_{slb} and $\mathcal{G}$ are defined in (24) and $H = \frac{f(\mathbf{w}_0) - f_{slb}}{f_* - f_{slb}}$. In comparison, our RSG can be better if $f_* - f_{slb}$ is large. To see this, we represent the complexity of the method by Freund and Lu (2017) in terms of the absolute error ϵ with $\epsilon = \epsilon'(f_* - f_{slb})$ and obtain $O(G^2\mathcal{G}^2(\frac{(f_* - f_{slb})\log H}{\epsilon} + \frac{(f_* - f_{slb})^2}{\epsilon^2}))$. If the gap $f_* - f_{slb}$ is large, e.g., $O(f(\mathbf{w}_0) - f_{slb})$, the second term is dominating, which is at least $\Omega(\frac{G^2\|\mathbf{w}_0 - \mathbf{w}_0^*\|_2^2}{\epsilon^2})$ due to the definition of $\mathcal{G}$ in (24). This complexity has the same order of magnitude as the standard SG method so that RSG can be better due to the reasoning in last paragraph. More generally, the iteration complexity of Freund and Lu's SG method can be reduced to $O(\frac{G^2 B_{f_* - f_{slb}}^2}{\epsilon^2})$ by choosing the best $\mathcal{G}$ in the proof of Theorem 1.1 in Freund and Lu (2017)'s paper, which depends on $f_* - f_{slb}$. In comparison, RSG could have a lower complexity if $f_* - f_{slb}$ is larger than ϵ as in Corollary 5 or $\hat{\epsilon}_1$ as in Theorem 15. Our experiments in subsection 8.4 also corroborate this point. In addition, RSG can leverage the local error bound condition to enjoy a lower iteration complexity than $O(1/\epsilon^2)$.

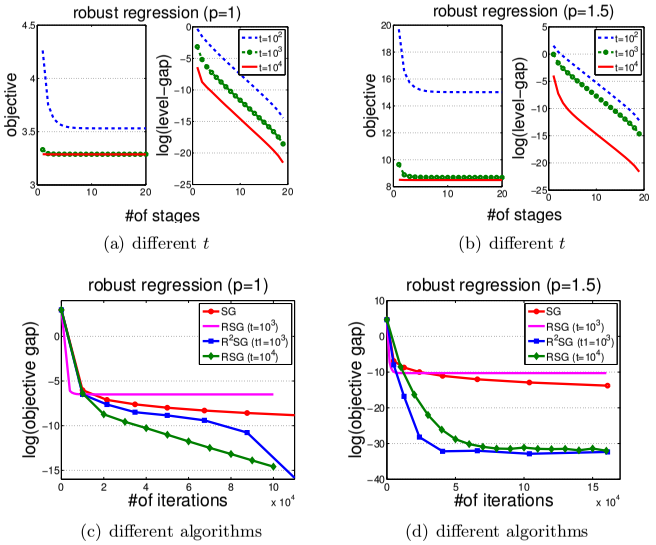
7.3 Comparison with the method by Juditsky and Nesterov (2014)

Juditsky and Nesterov (2014) considered primal-dual subgradient methods for solving the problem (1) with f being *uniformly convex*, namely,

$$f(\alpha\mathbf{w} + (1-\alpha)\mathbf{v}) \leq \alpha f(\mathbf{w}) + (1-\alpha)f(\mathbf{v}) - \frac{1}{2}\mu\alpha(1-\alpha)[\alpha^{\rho-1} + (1-\alpha)^{\rho-1}]\|\mathbf{w} - \mathbf{v}\|_2^\rho,$$

for any $\mathbf{w}$ and $\mathbf{v}$ in Ω and any $\alpha \in [0,1]$ [5], where $\rho \in [2, +\infty]$ and $\mu \geq 0$. In this case, the method by (Juditsky and Nesterov, 2014) has an iteration complexity of $O\left(\frac{G^2}{\mu^{2/\rho}\epsilon^{2(\rho-1)/\rho}}\right)$. The uniform convexity of f further implies $f(\mathbf{w}) - f_* \geq \frac{1}{2}\mu\|\mathbf{w} - \mathbf{w}^*\|_2^\rho$ for any $\mathbf{w} \in \Omega$ so that $f(\cdot)$ admits a local error bound on the ϵ-sublevel set $\mathcal{S}_\epsilon$ with $c = \left(\frac{2}{\mu}\right)^{\frac{1}{\rho}}$ and $\theta = \frac{1}{\rho}$. Therefore, our RSG has a complexity of $O\left(\frac{G^2}{\mu^{2/\rho}\epsilon^{2(\rho-1)/\rho}}\log(\frac{\epsilon_0}{\epsilon})\right)$ according to Corollary 7. Compared to the result of Juditsky and Nesterov (2014), our complexity is higher by a logarithmic factor. However, we only require the local error bound property of f that is weaker than uniform convexity and also covers much broader family of functions. Note that the above comparison is fair, since for achieving a target ϵ-optimal solution the algorithms

5. The Euclidean norm in the definition here can be replaced by a general norm as in (Juditsky and Nesterov, 2014).



(a) different t (b) different t

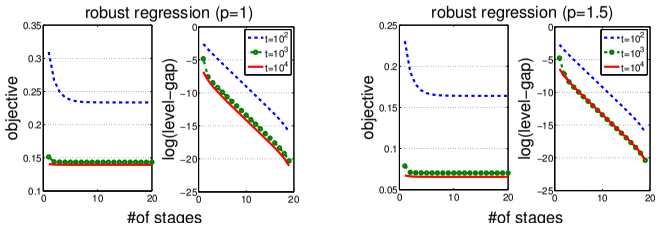
(c) different algorithms (d) different algorithms

Figure 2: Comparison of RSG with different t and of different algorithms on the housing data. One iteration means one subgradient update in all algorithms. (t1 for R^{2}SG represents the initial value of t in the first call of RSG.)

proposed by Juditsky and Nesterov (2014) do need the knowledge of uniform convexity parameter ρ and the parameter μ. It is worth mentioning that Juditsky and Nesterov (2014) also presented algorithms with a fixed number of iterations T as input that achieve adaptive rates without knowledge of ρ and μ. However, they only considered the case when $\rho >= 2$, which corresponds to $\theta \leq 1/2$ in our notations, while our methods can be applied also when $\theta > 1/2$.

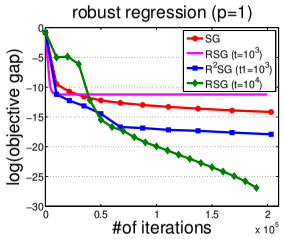
8. Experiments

In this section, we present some experiments to demonstrate the effectiveness of RSG. We first consider several applications in machine learning, in particular regression, classification and matrix completion, and focus on the comparison between RSG and SG. Then we make comparison between RSG with Freund & Lu's SG variant for solving regression problems. In experiments, all compared algorithms use the same initial solution unless otherwise specified.

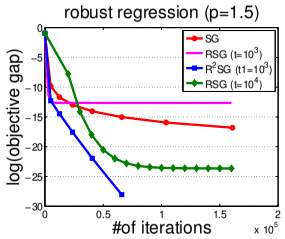


(a) different t

(b) different t



(c) different algorithms



(d) different algorithms

Figure 3: Comparison of RSG with different t and of different algorithms on the space-ga data. One iteration means one subgradient update in all algorithms.

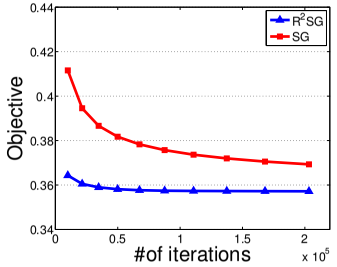
8.1 Robust Regression

The regression problem is to predict an output y based on a feature vector $\mathbf{x} \in \mathbb{R}^d$. Given a set of training examples $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, a linear regression model can be found by solving the optimization problem in (19).

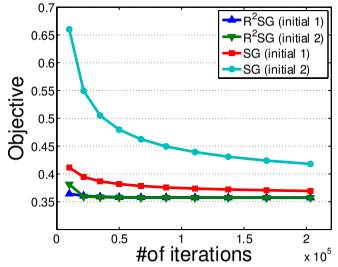
We solve two instances of the problem with $p = 1$ and $p = 1.5$ and $\lambda = 0$. We conduct experiments on two data sets from libsvm website [6], namely housing ($n = 506$ and $d = 13$) and space-ga ($n = 3107$ and $d = 6$). We first examine the convergence behavior of RSG with different values for the number of iterations per-stage $t = 10^2$, 10^3, and 10^4. The value of α is set to 2 in all experiments. The initial step size of RSG is set to be proportional to $\epsilon_0/2$ with the same scaling parameter for different variants. We plot the results on housing data in Figure 2 (a,b) and on space-ga data in Figure 3 (a,b). In each figure, we plot the objective value vs number of stages and the log difference between the objective value and the converged value (to which we refer as level gap). We can clearly see that with different values of t RSG converges to an ϵ-level set and the convergence rate is linear in terms of the number of stages, which is consistent with our theory.

Secondly, we compare with SG to verify the effectiveness of RSG. The baseline SG is implemented with a decreasing step size proportional to $1/\sqrt{\tau}$, where τ is the iteration

6. https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/



(a) comparison between different algorithms



(b) sensitivity to initial solutions

Figure 4: Results for solving SVM classification with GFlassso regularizer. In (b), the objective values of the two initial solutions are 1 and 70.35. One iteration means one subgradient update in all algorithms.

index. The initial step size of SG is tuned in a wide range to give the fastest convergence. The initial step size of RSG is also tuned around the best initial step size of SG. The results are shown in Figure 2(c,d) and Figure 3(c,d), where we show RSG with two different values of t and also R^{2}SG with an increasing sequence of t. In implementing R^{2}SG, we restart RSG for every 5 stages, and increase the number of iterations by a certain factor. In particular,we increase t by a factor of 1.15 and 1.5 respectively for $p = 1$ and $p = 1.5$. From the results, we can see that (i) RSG with a smaller value of $t = 10^3$ can quickly converge to an ϵ-level, which is less accurate than SG after running a sufficiently large number of iterations; (ii) RSG with a relatively large value $t = 10^4$ can converge to a much more accurate solution; (iv) R^{2}SG converges much faster than SG and can bridge the gap between RSG-$t = 10^3$ and RSG-$t = 10^4$.

8.2 SVM Classification with a graph-guided fused lasso

The classification problem is to predict a binary class label $y \in \{1, -1\}$ based on a feature vector $\mathbf{x} \in \mathbb{R}^d$. Given a set of training examples $(\mathbf{x}_i, y_i), i = 1, \ldots, n$, the problem of training a linear classification model can be cast into

$$\min_{\mathbf{w} \in \mathbb{R}^d} F(\mathbf{w}) := \frac{1}{n} \sum_{i=1}^{n} \ell(\mathbf{w}^\top \mathbf{x}_i, y_i) + R(\mathbf{w}).$$

Here we consider the hinge loss $\ell(z, y) = \max(0, 1 - yz)$ as in support vector machine (SVM) and a graph-guided fused lasso (GFlasso) regularizer $R(\mathbf{w}) = \lambda \|F\mathbf{w}\|_1$ (Kim et al., 2009), where $F = [F_{ij}]_{m \times d} \in \mathbb{R}^{m \times d}$ encodes the edge information between variables. Suppose there is a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ where nodes $\mathcal{V}$ are the attributes and each edge is assigned a weight s_{ij} that represents some kind of similarity between attribute i and attribute j. Let $\mathcal{E} = \{e_1, \ldots, e_m\}$ denote a set of m edges, where an edge $e_\tau = (i_\tau, j_\tau)$ consists of a tuple of two attributes. Then the τ-th row of F matrix can be formed by setting $F_{\tau, i_\tau} = s_{i_\tau, j_\tau}$ and $F_{\tau, j_\tau} = -s_{i_\tau, j_\tau}$ for $(i_\tau, j_\tau) \in \mathcal{E}$, and zeros for other entries. Then the GFlasso becomes

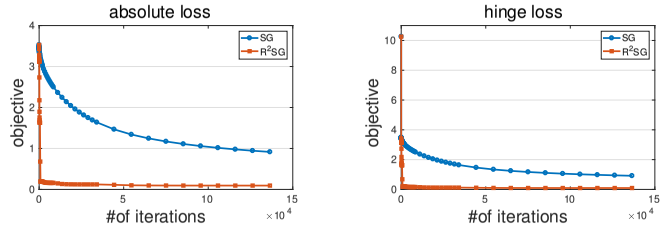


Figure 5: Results for solving low rank matrix completion with different loss functions.

$R(\mathbf{w}) = \lambda \sum_{(i,j)\in\mathcal{E}} s_{ij}|w_i - w_j|$. Previous studies have found that a carefully designed GFlasso regularization helps in reducing the risk of over-fitting. In this experiment, we follow (Ouyang et al., 2013) to generate a dependency graph by sparse inverse covariance selection (Friedman et al., 2008). To this end, we first generate a sparse inverse covariance matrix using the method in (Friedman et al., 2008) and then assign an equal weight $s_{ij} = 1$ to all edges that have non-zero entries in the resulting inverse covariance matrix. We conduct the experiment on the dna data ($n = 2000$ and $d = 180$) from the libsvm website, which has three class labels. We solve the above problem to classify class 3 versus the rest. The comparison between different algorithms starting from an initial solution with all zero entries for solving the above problem with $\lambda = 0.1$ is presented in Figure 4(a). For R^{2}SG, we start from $t_1 = 10^3$ and restart it every 10 stages with t increased by a factor of 1.15. The initial step sizes for all algorithms are tuned.

We also compare the dependence of R^{2}SG's convergence on the initial solution with that of SG. We use two different initial solutions (the first initial solution $\mathbf{w}_0 = 0$ and the second initial solution $\mathbf{w}_0$ is generated once from a normal Gaussian distribution). The convergence curves of the two algorithms from the two different initial solutions are plotted in Figure 4(b). Note that the initial step sizes of SG and R^{2}SG are separately tuned for each initial solution. We can see that R^{2}SG is much less sensitive to a bad initial solution than SG consistent with our theory.

8.3 Matrix Completion for Collaborative Filtering

In this subsection, we consider low rank matrix completion problems to demonstrate the effectiveness of R^{2}SG without having the knowledge of c and θ in the local error bound condition. We consider a movie recommendation data set, namely MovieLens 100k data [7], which contains 100,000 ratings from $m = 943$ users on $n = 1682$ movies. We formulate the problem as a task of recovering a full user-movie rating matrix X from the partially observed matrix Y. The objective is composed of a loss function measuring the difference between X and Y on the observed entries and a nuclear norm regularizer on X for enforcing

7. https://grouplens.org/datasets/movielens/

a low rank, i.e.,

$$\min_{X\in\mathbb{R}^{m\times n}} \frac{1}{N}\sum_{(i,j)\in\Sigma} \ell(X_{ij}, Y_{ij}) + \lambda\|X\|_*, \tag{25}$$

where Σ is a set of user-movie pairs that denote the observed entries, $\ell(\cdot,\cdot)$ denote a loss function, $\|X\|_*$ denotes the nuclear norm, $N = |\Sigma|$ and $\lambda > 0$ is a regularization parameter. We consider two loss functions, i.e, the hinge loss and the absolute loss. For absolute loss, we set $\ell(a, b) = |a - b|$. For hinge loss, we follow Rennie and Srebro (2005) by introducing four thresholds $\theta_{1,2,3,4}$ due to there are five distinct ratings in $\{1, 2, 3, 4, 5\}$ that can be assigned to each movie, and defining $\ell(a, b) = \sum_{r=1}^{4} \max(0, 1 - T^r_{i,j}(\theta_r - X_{ij}))$, where $T^r_{ij} = \begin{cases} 1 & \text{if } r \geq Y_{ij} \\ 0 & \text{otherwise} \end{cases}$. In our experiment, we set $\theta_{1,2,3,4} = (0, 3, 6, 9)$ and $\lambda = 10^{-5}$ following (Yang et al., 2014). Since the loss function and the nuclear norm are both semi-algebraic functions (Yang et al., 2016; Bolte et al., 2014), then the problem (25) satisfies an error bound condition on any compact set (Bolte et al., 2017). However, it remains an open problem what are the proper values of c and θ to make local error bound condition hold. Hence, we run R^{2}SG by setting $\theta = 0$. To compare with SG, we simply set $t_1 = 10$ - the number of iterations of each stage of the first call of RSG. The baseline SG is implemented in the same way as before. The results of the objective values vs the number of iterations are plotted in Figure 5. We can see that R^{2}SG converges much faster than SG, verifying the effectiveness of R^{2}SG predicted by Theorem 16.

8.4 Comparison with Freund & Lu's SG

In this subsection, we compare the proposed RSG with Freund & Lu' SG algorithm empirically. The later algorithm is designed with a fixed relative accuracy ϵ' such that $\frac{f(\mathbf{x}_t)-f_*}{f_*-f_{slb}} \leq \epsilon'$, where f_{slb} is a strict lower bound of f_*, and requires to maintain the best solution in terms of the objective value during the optimization. For fair comparison, we run RSG with a fixed t and then vary ϵ' for Freund & Lu's SG algorithm that is an input parameter, and then plot the objective values versus the running time and the number of iterations for both algorithms. The experiments are conducted on the two classification data sets as used in subsection 8.1, namely the housing data and the space-ga data, for solving robust regression problems (19) with $p = 1$ and $p = 1.5$. The strict lower bound f_{slb} in Freund & Lu's algorithm is set to 0. The results are shown in Figure 6 and Figure 7, where SGR refers to Freund & Lu's SG algorithm with a specified relative accuracy. For each problem instance (a data set and a particular value of p), we report two results comparing the objective values vs. running time and the number of iterations. We can see that RSG is very competitive in performance in terms of running time and converge faster than Freund & Lu's algorithm with a small $\epsilon' = 10^{-4}$ for achieving the same accurate solution (e.g., with objective gap less than 10^{-10}).

9. Conclusion

In this work, we have proposed a novel restarted subgradient method for non-smooth and/or non-strongly convex optimization for obtaining an ϵ-optimal solution. By leveraging the

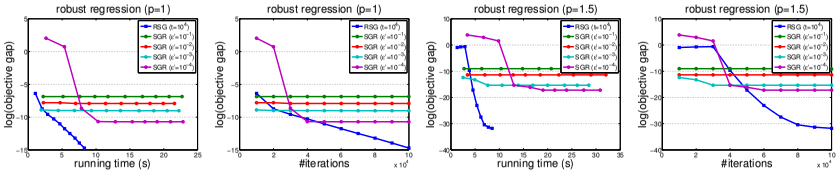


Figure 6: Comparison of RSG with Freund & Lu's SG algorithm (SGR) on the housing data.

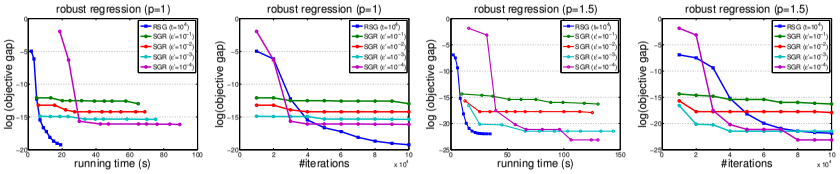


Figure 7: Comparison of RSG with Freund & Lu's SG algorithm (SGR) on the space-ga data.

lower bound of the first-order optimality residual, we establish a generic complexity of RSG that improves over standard subgradient method. We have also considered several classes of non-smooth and non-strongly convex problems that admit a local error bound condition and derived the improved order of iteration complexities for RSG. Several extensions have been made to design a parameter-free variant of RSG without requiring the knowledge of the constants in the local error bound condition. Experimental results on several machine learning tasks have demonstrated the effectiveness of the proposed algorithms in comparison to the subgradient method.

Acknowledgments

We would like to sincerely thank anonymous reviewers for their very helpful comments. We thank James Renegar for pointing out the connection to his work and for his valuable comments on the difference between the two work. We also thank to Nghia T.A. Tran for pointing out the connection between the local error bound and metric subregularity of subdifferentials. Thanks to Mingrui Liu for spotting an error in the formulation of the F matrix for GFlasso in earlier versions. T. Yang was supported by NSF (1463988, 1545995).

Appendix A. A proposition needed to prove Corollary 14

The proof of Corollary 14 leverages the following result (Bolte et al., 2017).

Proposition 1 *(Bolte et al., 2017, Theorem 5) Let $f(x)$ be an extended-valued, proper, convex and lower semicontinuous function that satisfies the KL inequality* (20) *at $x_* \in \arg\min f(\cdot)$ for all $x \in U \cap \{x : f(x_*) < f(x) < f(x_*)+\eta\}$, where U is a neighborhood of x_*, then $dist(x, \arg\min f(\cdot)) \leq \varphi(f(x) - f(x_*))$ for all $x \in U \cap \{x : f(x_*) < f(x) < f(x_*)+\eta\}$.*

References

Francisco J Aragon Artacho and Michel H Geoffroy. Characterization of metric regularity of subdifferentials. *Journal of Convex Analysis*, 15:365–380, 2008.

Hédy Attouch, Jérôme Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the kurdyka-lojasiewicz inequality. *Math. Oper. Res.*, 35:438–457, 2010.

Hedy Attouch, Jérôme Bolte, and Benar Fux Svaiter. Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized gauss-seidel methods. *Math. Program.*, 137(1-2):91–129, 2013.

Francis R. Bach. Learning with submodular functions: A convex optimization perspective. *Foundations and Trends in Machine Learning*, 6(2-3):145–373, 2013.

Francis R. Bach and Eric Moulines. Non-strongly-convex smooth stochastic approximation with convergence rate o(1/n). In *Advances in Neural Information Processing Systems (NIPS)*, pages 773–781, 2013.

Dimitris Bertsimas and Martin S. Copenhaver. Characterization of the equivalence of robustification and regularization in linear, median, and matrix regression. *arXiv*, 2014.

Jérôme Bolte, Aris Daniilidis, and Adrian Lewis. The łojasiewicz inequality for nonsmooth subanalytic functions with applications to subgradient dynamical systems. *SIAM J. on Optimization*, 17:1205–1223, 2006.

Jérôme Bolte, Aris Daniilidis, Adrian S. Lewis, and Masahiro Shiota. Clarke subgradients of stratifiable functions. *SIAM Journal on Optimization*, 18:556–572, 2007.

Jérôme Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146: 459–494, 2014.

Jérôme Bolte, Trong Phong Nguyen, Juan Peypouquet, and Bruce W. Suter. From error bounds to the complexity of first-order descent methods for convex functions. *Mathematical Programming*, 165(2):471–507, Oct 2017. doi: 10.1007/s10107-016-1091-6.

James V. Burke and Sien Deng. Weak sharp minima revisited part i: Basic theory. *Control and Cybernetics*, 31:439?469, 2002.

James V. Burke and Sien Deng. Weak sharp minima revisited, part II: application to linear regularity and error bounds. *Math. Program.*, 104(2-3):235–261, 2005.

James V. Burke and Sien Deng. Weak sharp minima revisited, part III: error bounds for differentiable convex inclusions. *Math. Program.*, 116(1-2):37–56, 2009.

James V. Burke and Michael C. Ferris. Weak sharp minima in mathematical programming. *SIAM Journal on Control and Optimization*, 31(5):1340–1359, 1993. doi: 10.1137/0331063.

Xi Chen, Qihang Lin, and Javier Pena. Optimal regularized dual averaging methods for stochastic optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 395–403. 2012.

Dmitriy Drusvyatskiy and Courtney Kempton. An accelerated algorithm for minimizing convex compositions. *arXiv:1605.00125*, 2016.

Dmitriy Drusvyatskiy and Adrian S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods. *Mathematics of Operations Research*, 2018.

Dmitriy Drusvyatskiy, Boris Mordukhovich, and Nghia T.A. Tran. Second-order growth, tilt stability, and metric regularity of the subdifferential. *Journal of Convex Analysis*, 21: 1165–1192, 2014.

I. I. Eremin. The relaxation method of solving systems of inequalities with convex functions on the left-hand side. *Dokl. Akad. Nauk SSSR*, 160:994 – 996, 1965.

Michael C. Ferris. Finite termination of the proximal point algorithm. *Mathematical Programming*, 50(1):359–366, Mar 1991.

Robert M. Freund and Haihao Lu. New computational guarantees for solving convex optimization problems with first order methods, via a function growth condition measure. *Mathematical Programming*, 2017.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9, 2008.

Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013.

Andrew Gilpin, Javier Peña, and Tuomas Sandholm. First-order algorithm with log(1/epsilon) convergence for epsilon-equilibrium in two-person zero-sum games. *Math. Program.*, 133(1-2):279–298, 2012.

R. Goebel and R. T. Rockafellar. Local strong convexity and local lipschitz continuity of the gradient of convex functions. *Journal of Convex Analysis*, 2007.

Pinghua Gong and Jieping Ye. Linear convergence of variance-reduced projected stochastic gradient without strong convexity. *CoRR*, abs/1406.1102, 2014.

Elad Hazan and Satyen Kale. Beyond the regret minimization barrier: an optimal algorithm for stochastic strongly-convex optimization. In *Proceedings of the 24th Annual Conference on Learning Theory (COLT)*, pages 421–436, 2011.

Ke Hou, Zirui Zhou, Anthony Man-Cho So, and Zhi-Quan Luo. On the linear convergence of the proximal gradient method for trace norm regularization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 710–718, 2013.

Anatoli Juditsky and Yuri Nesterov. Deterministic and stochastic primal-dual subgradient algorithms for uniformly convex minimization. *Stochastic Systems*, 4(1):44–80, 2014.

Hamed Karimi, Julie Nutini, and Mark W. Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases - European Conference (ECML-PKDD)*, pages 795–811, 2016.

Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12), 2009.

Alexander Y. Kruger. Error bounds and hölder metric subregularity. *Set-Valued and Variational Analysis*, 23:705–736, 2015.

Simon Lacoste-Julien, Mark W. Schmidt, and Francis R. Bach. A simpler approach to obtaining an o(1/t) convergence rate for the projected stochastic subgradient method. *CoRR*, abs/1212.2002, 2012. URL `http://arxiv.org/abs/1212.2002`.

Guoyin Li. On the asymptotically well behaved functions and global error bound for convex polynomials. *SIAM Journal on Optimization*, 20(4):1923–1943, 2010.

Guoyin Li. Global error bounds for piecewise convex polynomials. *Math. Program.*, 137 (1-2):37–64, 2013.

Guoyin Li and Boris S. Mordukhovich. Hölder metric subregularity with applications to proximal point method. *SIAM Journal on Optimization*, 22(4):1655–1684, 2012.

Zhi-Quan Luo and Paul Tseng. On the convergence of coordinate descent method for convex differentiable minization. *Journal of Optimization Theory and Applications*, 72(1):7–35, 1992a.

Zhi-Quan Luo and Paul Tseng. On the linear convergence of descent methods for convex essenially smooth minization. *SIAM Journal on Control and Optimization*, 30(2):408–425, 1992b.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: a general approach. *Annals of Operations Research*, 46:157–178, 1993.

Boris Mordukhovich and Wei Ouyang. Higher-order metric subregularity and its applications. *Journal of Global Optimization- An International Journal Dealing with Theoretical and Computational Aspects of Seeking Global Optima and Their Applications in Science, Management and Engineering*, 63(4):777–795, 2015.

Ion Necoara and Dragos Clipici. Parallel random coordinate descent method for composite minimization: Convergence analysis and error bounds. *SIAM Journal on Optimization*, 26(1):197–226, 2016. doi: 10.1137/130950288.

Ion Necoara, Yurri Nesterov, and Francois Glineur. Linear convergence of first order methods for non-strongly convex optimization. *CoRR*, abs/1504.06298, 2015.

Arkadi Nemirovski, Anatoli Juditsky, Guanghui Lan, and Alexander Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19:1574–1609, 2009.

Arkadii Semenovich. Nemirovsky A.S. and D. B Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience series in discrete mathematics. Wiley, Chichester, New York, 1983. ISBN 0-471-10345-4. A Wiley-Interscience publication.

Yurii Nesterov. *Introductory lectures on convex optimization : a basic course*. Applied optimization. Kluwer Academic Publ., 2004. ISBN 1-4020-7553-7.

Yurii Nesterov. Smooth minimization of non-smooth functions. *Mathematical Programming*, 103(1):127–152, 2005.

Yurii Nesterov. Primal-dual subgradient methods for convex problems. *Mathematical Programming*, 120:221–259, 2009.

Hua Ouyang, Niao He, Long Tran, and Alexander G. Gray. Stochastic alternating direction method of multipliers. In *Proceedings of the 30th International Conference on Machine Learning (ICML)*, pages 80–88, 2013.

Jong-Shi Pang. A posteriori error bounds for the linearly-constrained variational inequality problem. *Mathematics of Operations Research*, 12(3):474–484, August 1987. ISSN 0364-765X.

Jong-Shi Pang. Error bounds in mathematical programming. *Mathematical Programming*, 79(1):299–332, October 1997.

B. T. Polyak. Sharp minima. In *Proceedings of the IIASA Workshop on Generalized Lagrangians and Their Applications*, Institute of Control Sciences Lecture Notes, Moscow. 1979.

B. T. Polyak. *Introduction to Optimization*. Optimization Software Inc, New York, 1987.

B.T. Polyak. Minimization of unsmooth functionals. *USSR Computational Mathematics and Mathematical Physics*, 9:509 – 521, 1969.

James Renegar. Efficient first-order methods for linear programming and semidefinite programming. *ArXiv e-prints*, 2014.

James Renegar. A framework for applying subgradient methods to conic optimization problems. *ArXiv e-prints*, 2015.

James Renegar. "efficient" subgradient methods for general convex optimization. *SIAM Journal on Optimization*, 26(4):2649–2676, 2016.

Jasson D. M. Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22Nd International Conference on Machine Learning*, pages 713–719, New York, NY, USA, 2005. ACM. ISBN 1-59593-180-5. doi: 10.1145/1102351.1102441. URL `http://doi.acm.org/10.1145/1102351.1102441`.

R.T. Rockafellar. *Convex Analysis*. Princeton mathematical series. Princeton University Press, 1970.

Reinhold Schneider and André Uschmajew. Convergence results for projected line-search methods on varieties of low-rank matrices via łojasiewicz inequality. *SIAM Journal on Optimization*, 25(1):622–646, 2015.

Anthony Man-Cho So and Zirui Zhou. Non-asymptotic convergence analysis of inexact gradient methods for machine learning without strong convexity. *Optimization Methods and Software*, 32:963 – 992, 2017.

Marcin Studniarski and Doug E. Ward. Weak sharp minima: Characterizations and sufficient conditions. *SIAM Journal on Control and Optimization*, 38(1):219–236, 1999. doi: 10.1137/S0363012996301269.

P. Tseng and S. Yun. A block-coordinate gradient descent method for linearly constrained nonsmooth separable optimization. *Journal of Optimization Theory Application*, 140: 513–535, 2009a.

P. Tseng and S. Yun. A coordinate gradient descent method for nonsmooth separable minimization. *Mathematical Programming*, 117:387–423, 2009b.

Po-Wei Wang and Chih-Jen Lin. Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, 15(1):1523–1548, 2014.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and lasso. *IEEE Trans. Information Theory*, 56(7):3561–3574, 2010.

Yi Xu, Yan Yan, Qihang Lin, and Tianbao Yang. Homotopy smoothing for non-smooth problems with lower complexity than 1/epsilon. In *Advances in Neural Information Processing Systems*, 2016.

Yi Xu, Qihang Lin, and Tianbao Yang. Stochastic convex optimization: Faster local growth implies faster global convergence. In *Proceedings of the 34th International Conference on Machine Learning, (ICML)*, pages 3821–3830, 2017.

Tianbao Yang and Qihang Lin. Rsg: Beating sgd without smoothness and/or strong convexity. *CoRR*, abs/1512.03107, 2016.

Tianbao Yang, Mehrdad Mahdavi, Rong Jin, and Shenghuo Zhu. An efficient primal-dual prox method for non-smooth optimization. *Machine Learning*, 2014.

W. H. Yang. Error bounds for convex polynomials. *SIAM Journal on Optimization*, 19(4): 1633–1647, 2009.

Yuning Yang, Yunlong Feng, and Johan A. K. Suykens. Robust low-rank tensor recovery with regularized redescending m-estimator. *IEEE Trans. Neural Netw. Learning Syst.*, 27(9):1933–1946, 2016.

Hui Zhang. Characterization of linear convergence of gradient descent. *arXiv:1606.00269*, 2016.

Hui Zhang and Lizhi Cheng. Restricted strong convexity and its applications to convergence analysis of gradient-type methods in convex optimization. *Optimization Letter*, 9:961–979, 2015.

Zirui Zhou and Anthony Man-Cho So. A unified approach to error bounds for structured convex optimization problems. *Mathematical Programming*, 165(2):689–728, Oct 2017.

Zirui Zhou, Qi Zhang, and Anthony Man-Cho So. L1p-norm regularization: Error bounds and convergence rate analysis of first-order methods. In *Proceedings of the 32nd International Conference on Machine Learning, (ICML)*, pages 1501–1510, 2015.

Martin Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 928–936, 2003.

Journal of Machine Learning Research 19 (2018) 1-43 Submitted 1/17; Revised 7/18; Published 7/18

Patchwork Kriging for Large-scale Gaussian Process Regression

Chiwoo Park cpark5@fsu.edu
Department of Industrial and Manufacturing Engineering
Florida State University
2525 Pottsdamer St., Tallahassee, FL 32310-6046, USA.

Daniel Apley apley@northwestern.edu
Dept. of Industrial Engineering and Management Sciences
Northwestern University
2145 Sheridan Rd., Evanston, IL 60208-3119, USA.

Editor: Neil Lawrence

Abstract

This paper presents a new approach for Gaussian process (GP) regression for large datasets. The approach involves partitioning the regression input domain into multiple local regions with a different local GP model fitted in each region. Unlike existing local partitioned GP approaches, we introduce a technique for patching together the local GP models nearly seamlessly to ensure that the local GP models for two neighboring regions produce nearly the same response prediction and prediction error variance on the boundary between the two regions. This largely mitigates the well-known discontinuity problem that degrades the prediction accuracy of existing local partitioned GP methods over regional boundaries. Our main innovation is to represent the continuity conditions as additional pseudo-observations that the differences between neighboring GP responses are identically zero at an appropriately chosen set of boundary input locations. To predict the response at any input location, we simply augment the actual response observations with the pseudo-observations and apply standard GP prediction methods to the augmented data. In contrast to heuristic continuity adjustments, this has an advantage of working within a formal GP framework, so that the GP-based predictive uncertainty quantification remains valid. Our approach also inherits a sparse block-like structure for the sample covariance matrix, which results in computationally efficient closed-form expressions for the predictive mean and variance. In addition, we provide a new spatial partitioning scheme based on a recursive space partitioning along local principal component directions, which makes the proposed approach applicable for regression domains having more than two dimensions. Using three spatial datasets and three higher dimensional datasets, we investigate the numerical performance of the approach and compare it to several state-of-the-art approaches.

Keywords: Local Kriging, Model Split and Merge, Pseudo Observations, Spatial Partition

1. Introduction

Gaussian process (GP) regression is a popular Bayesian nonparametric approach for nonlinear regression (Rasmussen and Williams, 2006). A GP prior is assumed for the unknown regression function, and the posterior estimate of the function is from this prior, combined with noisy (or noiseless, for deterministic simulation response surfaces) response observations. The posterior estimate can be easily derived in a simple closed form using the properties induced by the GP prior, and the estimator has several desirable properties, e.g., it is the best linear unbiased estimator under the assumed model and offers convenient quantification of the prediction error uncertainty. Its conceptual simplicity and attractive properties are major reasons for its popularity. On the other hand, the computational expense for evaluating the closed form solution is proportional to N^3, where N denotes the number of observations, which can be prohibitively expensive for large N. Broadly speaking, this paper concerns fast computation of the GP regression estimate for large N.

The major computational bottleneck for GP regression is the inversion of a $N \times N$ sample covariance matrix, which is also often poorly numerically conditioned. Different approaches for representing or approximating the sample covariance matrix with a more efficiently invertible form have been proposed. The approaches can be roughly categorized as sparse approximations, low-rank approximations, or local approximations. Sparse methods represent the sample covariance with a sparse version, e.g. by applying a covariance tapering technique (Furrer et al., 2006; Kaufman et al., 2008), using a compactly supported covariance function (Gneiting, 2002), or using a Gaussian Markov approximation of a GP (Lindgren et al., 2011). The inversion of a sparse positive definite matrix is less computationally expensive than the inversion of a non-sparse matrix of the same size, because its Cholesky decomposition is sparse and can be achieved more quickly.

Low-rank approximations of the sample covariance matrix can be performed in multiple ways. The most popular approach for the low-rank approximation introduces latent variables and assume a certain independence conditioned on the latent variables (Seeger et al., 2003; Snelson and Ghahramani, 2006), so that the resulting sample covariance matrix has reduced rank. The (pseudo)inversion of a $N \times N$ matrix of rank M can be computed with reduced $O(NM^2)$ expense. Titsias (2009) introduced a variational formulation to infer the latent variables along with covariance parameters, and a variant of the idea was proposed using the stochastic variational inference technique (Hensman et al., 2013). The latent variable model approaches are exploited to develop parallel computing algorithms for GP regression (Chen et al., 2013). Another way for low rank approximation is to approximate the sample covariance matrix with a product of a block diagonal matrix and multiple blocked low-rank matrices (Ambikasaran et al., 2016).

Local approximation approaches partition the input domain into a set of local regions and assume an independent GP regression model within each region (Das and Srivastava, 2010). The resulting sample covariance matrix is a block diagonal matrix of local sample covariance matrices, and inverting the block diagonal matrix is much cheaper computationally. Such local approximation approaches have many advantages. By their local nature, they adapt better to local and nonstationary data features, and independent local approximation models can be computed in parallel to reduce total computation time. Their major weakness is that two local models for two neighboring local regions produce different predictions at the boundary between the regions, creating discontinuity of the regression function over the boundary. This boundary discontinuity is not just an aesthetic problem, as it was empirically shown that greater discontinuity implies greater degradation in prediction accuracy, particularly around the boundaries of the local regions (Park and Huang, 2016). This discontinuity issue has been addressed in different ways. Perhaps the most popular

approach is to smooth out some of the discontinuity by using some weighted average across the local models or across multiple sets of local models via a Dirichlet mixture (Rasmussen and Ghahramani, 2002), a treed mixture (Gramacy and Lee, 2008), Bayesian model averaging (Tresp, 2000; Chen and Ren, 2009; Deisenroth and Ng, 2015), or locally weighted projections (Nguyen-Tuong et al., 2009). Other, related approaches use additive covariance functions consisting of a global covariance and a local covariance (Snelson and Ghahramani, 2007; Vanhatalo and Vehtari, 2008), construct a local model for each testing point (Gramacy and Apley, 2015), or use a local partition but constrain the local models for continuity (Park et al., 2011; Park and Huang, 2016).

In this work we use a partitioned input domain like Park et al. (2011) and Park and Huang (2016), but we introduce a different form of continuity constraints that are more easily and more naturally integrated into the GP modeling framework. Both Park et al. (2011) and Park and Huang (2016) basically reformulated local GP regression as an optimization problem, and the local GP models for neighboring regions were constrained to have the same predictive means on the boundaries of the local regions by adding some linear constraints to the optimization problems that infer the local GP models. Park et al. (2011) used a constrained quadratic optimization that constrains the predictive means for a finite number of boundary locations, and Park and Huang (2016) introduced a functional optimization formulation to enforce the same constraints for all boundary locations. The optimization-based formulations make it infeasible to derive the marginal likelihood and the predictive variances in closed forms, which were roughly approximated. In contrast, this paper presents a simple and natural way to enforce continuity. We consider a set of GPs that are defined as the differences between the responses for the local GPs in neighboring regions. Continuity implies that these differenced GPs are identically zero along the boundary between neighboring regions. Hence, we impose continuity constraints by treating the values of the differenced GPs at a specified set of boundary points as all having been "observed to be zero", and we refer to these zero-valued differences as pseudo-observations. We can then conveniently incorporate continuity constraints by simply augmenting the actual set of response observations with the set of pseudo-observations, and then using standard GP modeling to calculate the posterior predictive distribution given the augmented set of observations. We note that *observing* the differenced GPs to be zero at a set of boundary points is essentially equivalent to *assuming* continuity at these points without imposing any further assumptions on the nature of the GPs.

The new modeling approach creates several major benefits over the previous domain partitioning approaches. The new modeling is simpler than the previous approaches, so the marginal likelihood function can be explicitly derived for tuning hyperparameters, which was not possible for the previous approaches. In the previous approaches, the values of the predictive means on the boundaries of local regions must be explicitly specified, which involves solving a large linear system of equations. Unlike the previous approaches, observing the pseudo-observations of the differenced GPs to be zero does not require specifying the actual values of the predictive means and variances on the boundaries. Furthermore, the proposed approach enforces that the local models for neighboring regions produce the same values for both the predictive means and variances at the boundary points between the local regions, while both of the previous approaches are only able to enforce the same predictive means but not the same predictive variances. Last, the previous approaches are

only applicable for one- or two-dimensional problems, while our new approach is applicable for higher dimensional regression problems. We view our approach as "patching" together a collection of local GP regression models using the boundary points as "stitches", and, hence, we refer to it as *patchwork kriging*.

The remainder of the paper is organized as follows. Section 2 briefly reviews the general GP regression problem and notational convention. Section 3 presents the core methodology of the patchwork kriging approach, including the prior model assumptions, the pseudo-observation definition, the resulting posterior predictive mean and variance equations, and the detailed computation steps along with choice of tuning parameters. Section 4 shows how the patchwork kriging performs with a toy example for illustrative purpose. Section 5 investigates the numerical performance of the proposed method for different simulated cases and compares it with the exact GP regression (i.e., the GP regression without partitions, using the entire dataset to predict each point) and a global GP approximation method. Section 6 presents the numerical performance of the proposed approach for five real datasets and compares it with Park and Huang (2016) and other state-of-the-art methods. Finally, Section 7 concludes the paper with a discussion.

2. Gaussian Process Regression

Consider the general regression problem of estimating an unknown predictive function f that relates a d dimensional predictor $x \in \mathbb{R}^d$ to a real response y, using noisy observations $\mathcal{D} = \{(x_i, y_i), i = 1, \ldots, N\}$,

$$y_i = \mu + f(x_i) + \epsilon_i, \qquad i = 1, \ldots, N,$$

where $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ is white noise, independent of $f(x_i)$. We assume that $\mu = 0$. Otherwise, one can normalize y_i by subtracting the sample mean of the y_i's from y_i. Notice that we do not use bold font for the multivariate predictor x_i and reserve bold font for the collection of observed predictor locations, $\mathbf{x} = [x_1, x_2, \ldots, x_N]^T$.

In a GP regression for this problem, one assumes that f is a realization of a zero-mean Gaussian process having covariance function $c(\cdot, \cdot)$ and then uses the observations $\mathcal{D}$ to obtain the posterior predictive distribution of f at an arbitrary x_*, denoted by $f_* = f(x_*)$. Denote $\boldsymbol{y} = [y_1, y_2, \ldots, y_N]^T$. The joint distribution of $(f_*, \boldsymbol{y})$ is

$$P(f_*, \boldsymbol{y}) = \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} c_{**} & \boldsymbol{c}_{\mathbf{x}*}^T \\ \boldsymbol{c}_{\mathbf{x}*} & \sigma^2 \boldsymbol{I} + \boldsymbol{C}_{\mathbf{xx}} \end{bmatrix}\right),$$

where $c_{**} = c(x_*, x_*)$, $\boldsymbol{c}_{\mathbf{x}*} = (c(x_1, x_*), \ldots, c(x_N, x_*))^T$ and $\boldsymbol{C}_{\mathbf{xx}}$ is an $N \times N$ matrix with $(i, j)^{th}$ entry $c(x_i, x_j)$. The subscripts on c_{**}, $\boldsymbol{c}_{\mathbf{x}*}$, and $\boldsymbol{C}_{\mathbf{xx}}$ indicate the two sets of locations between which the covariance is computed, and we have abbreviated the subscript x_* as $*$. Applying the Gaussian conditioning formula to the joint distribution gives the predictive distribution of f_* given $\boldsymbol{y}$ (Rasmussen and Williams, 2006),

$$P(f_*|\boldsymbol{y}) = \mathcal{N}(\boldsymbol{c}_{\mathbf{x}*}^T(\sigma^2 \boldsymbol{I} + \boldsymbol{C}_{\mathbf{xx}})^{-1}\boldsymbol{y}, c_{**} - \boldsymbol{c}_{\mathbf{x}*}^T(\sigma^2 \boldsymbol{I} + \boldsymbol{C}_{\mathbf{xx}})^{-1}\boldsymbol{c}_{\mathbf{x}*}). \tag{1}$$

The predictive mean $\boldsymbol{c}_{\mathbf{x}*}^T(\sigma^2 \boldsymbol{I} + \boldsymbol{C}_{\mathbf{xx}})^{-1}\boldsymbol{y}$ is taken to be the point prediction of $f(x)$ at location x_*, and its uncertainty is measured by the predictive variance $c_{**} - \boldsymbol{c}_{\mathbf{x}*}^T(\sigma^2 \boldsymbol{I} + \boldsymbol{C}_{\mathbf{xx}})^{-1}\boldsymbol{c}_{\mathbf{x}*}$. Efficient calculation of the predictive mean and variance for large datasets has been the focus of much research.

3. Patchwork Kriging

As mentioned in the introduction, for efficient computation we replace the GP regression by a set of local GP models on some partition of the input domain, in a manner that enforces a level of continuity in the local GP model responses over the boundaries separating their respective regions. Section 3.1 conveys the main idea of the proposed approach,

3.1 Inference with Boundary Continuity Constraints

To specify the idea more precisely, consider a spatial partition of the input domain of $f(x)$ into K local regions $\{\Omega_k : k = 1, 2, ..., K\}$, and define $f_k(x)$ as the local GP approximation of $f(x)$ at $x \in \overline{\Omega}_k$, where $\overline{\Omega}_k$ is the closure of Ω_k. Temporarily ignoring the continuity requirements, the local models are assumed to follow independent GP priors:

Assumption 1 *Each $f_k(x)$ follows a GP prior distribution with zero mean and covariance function $c_k(\cdot,\cdot)$, and the $f_k(x)$'s are mutually independent a priori (prior to enforcing the continuity conditions). The choice of the local covariance function(s) can differ depending on the specifics of the problem. If $f(x)$ is expected to be a stationary process, then one could use the same $c_k(\cdot,\cdot) = c(\cdot,\cdot)$ for all k. In this case, the purpose of this local GP approximation would be to approximate $f(x)$ computationally efficiently. On the other hand, if one expects non-stationary behavior of the data, then different covariance functions should be used for each region.*

It is important to note that the independence of the GPs in Assumption 1 is prior to enforcing the continuity conditions via the pseudo-observations, as described below. After enforcing the continuity conditions, the GPs will no longer be independent *a priori*, since the assumed continuity at the boundaries imposes a very strong prior dependence of the surfaces. Since the pseudo-observations should also be viewed as additional prior information, the independence condition in Assumption 1 might be more appropriately viewed as a *hyperprior* condition. In fact, we view the incorporation of the boundary pseudo-observations as an extremely tractable and straightforward way of imposing some reasonable form of dependency of the $f_k(x)$ across regions (which is the ultimate goal), while still allowing us to begin with an independent GP *hyperprior* (which results in the tractability of the analyses).

Now partition the training set $\mathcal{D}$ into $\mathcal{D}_k = \{(x_i, y_i) : x_i \in \Omega_k\}$ $(k = 1, 2, ..., K)$, and denote $\mathbf{x}_k = \{x_i : x_i \in \Omega_k\}$ and $\boldsymbol{y}_k = \{y_i : x_i \in \Omega_k\}$. By the independence part of Assumption 1, the predictive distribution of $f_k(x)$ given $\mathcal{D}$ is equivalent to the predictive distribution of $f_k(x)$ given $\mathcal{D}_k$, which gives the standard local GP solution with no continuity requirements. The primary problem with this formulation is that the predictive distributions of $f_k(x)$ and $f_l(x)$ are not equal on the boundary of their neighboring regions Ω_k and Ω_l.

Our objective is to improve the local kriging prediction by enforcing $f_k(x) = f_l(x)$ on their shared boundary. The key idea is illustrated in Figure 1 and described as follows. For two neighboring regions Ω_k and Ω_l, let $\Gamma_{k,l} = \overline{\Omega}_k \cap \overline{\Omega}_l$ denote their shared boundary. For each pair of neighboring regions Ω_k and Ω_l, we define the auxiliary process $\delta_{k,l}(x)$ to be the difference between the two local GP models,

$$\delta_{k,l}(x) = f_k(x) - f_l(x) \text{ for } x \in \Gamma_{k,l}, \tag{2}$$

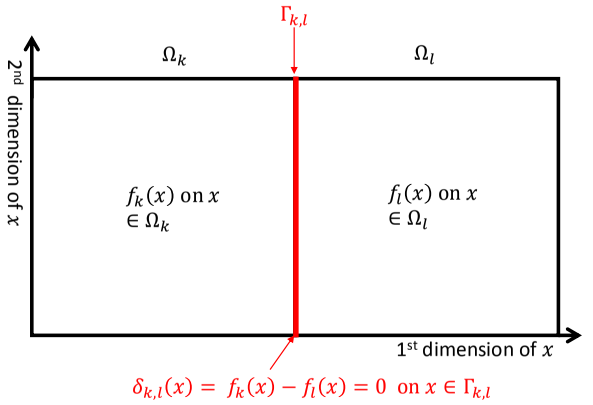


Figure 1: Illustration of the notation and concepts for defining the local models $f_k(x)$ and $f_l(x)$. Ω_k and Ω_l represent two local regions resulting from some appropriate spatial partition (discussed later) of the regression input domain. The (posterior distributions for the) GP functions $f_k(x)$ and $f_l(x)$ represent the local approximations of the regression function on Ω_k and Ω_l, respectively. The subset $\Gamma_{k,l}$ represents the interfacial boundary between the two regions Ω_k and Ω_l, and $\delta_{k,l}(x)$ is defined as the difference between $f_k(x)$ and $f_l(x)$, which is identically zero on $\Gamma_{k,l}$ by the continuity assumption.

and it is only defined for $k < l$ to avoid any duplicated definition of the auxiliary process. By the definition and under Assumption 1, $\delta_{k,l}(x)$ is a Gaussian process with zero mean and covariance function $c_k(\cdot,\cdot) + c_l(\cdot,\cdot)$, and its covariance with $f_j(x)$ is

$$\begin{aligned}\mathbb{C}\text{ov}(\delta_{k,l}(x_1), f_j(x_2)) &= \mathbb{C}\text{ov}(f_k(x_1) - f_l(x_1), f_j(x_2)) \\ &= \mathbb{C}\text{ov}(f_k(x_1), f_j(x_2)) - \mathbb{C}\text{ov}(f_l(x_1), f_j(x_2)).\end{aligned}$$

Since $\mathbb{C}\text{ov}(f_k(x_1), f_l(x_2)) = c_k(x_1, x_2)$ for $k = l$ and zero otherwise under Assumption 1,

$$\mathbb{C}\text{ov}(\delta_{k,l}(x_1), f_j(x_2)) = \begin{cases} c_k(x_1, x_2) & \text{if } k = j \\ -c_l(x_1, x_2) & \text{if } l = j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Likewise, $\delta_{k,l}(x)$ and $\delta_{u,v}(x)$ are correlated with covariance

$$\mathbb{C}\text{ov}(\delta_{k,l}(x_1), \delta_{u,v}(x_2)) = \begin{cases} c_k(x_1, x_2) & \text{if } k = u, l \neq v \\ c_l(x_1, x_2) & \text{if } l = v, k \neq u \\ -c_k(x_1, x_2) & \text{if } k = v, l \neq u \\ -c_l(x_1, x_2) & \text{if } l = u, k \neq v \\ c_k(x_1, x_2) + c_l(x_1, x_2) & \text{if } k = u, l = v \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Boundary continuity between $f_k(x)$ and $f_l(x)$ can be achieved by enforcing the condition $\delta_{k,l}(x) = 0$ at $\Gamma_{j,k}$. We reiterate that $f_k(x)$ and $f_l(x)$ are no longer independent after conditioning on the additional information $\delta_{k,l}(x) = 0$. In fact, they are strongly dependent, as they must be in order to achieve continuity *a priori*. Hence, the independence condition in Assumption 1 is really a *hyperprior* independence.

Deriving the exact prior distribution of the surface conditioned on $\delta_{k,l}(x) = 0$ everywhere on the boundaries appears to be computationally intractable, because there are uncountably infinitely many x's in $\Gamma_{j,k}$. Instead, we propose a much simpler continuity correction that begins with the Assumption 1 prior (including the independence *hyperprior*) and augments the observed data $\mathcal{D}$ with the pseudo observations $\delta_{k,l}(x) = 0$ for a finite number of input locations $x \in \Gamma_{k,l}$. As the number of boundary pseudo-observations grows, we can better approximate the theoretical ideal condition that $\delta_{k,l}(x) = 0$ everywhere on the boundary. The choice of the boundary input locations will be discussed later.

Notice that observing "$\delta_{k,l}(x) = 0$" is equivalent to observing that $f_k(x) = f_l(x)$ without observing the actual values of $f_k(x)$ and $f_l(x)$. Thus, if we augment $\mathcal{D}$ to include these pseudo observations when calculating the posterior predictive distributions of $f_k(x)$ and $f_l(x)$, it will force the posterior distributions of $f_k(x)$ and $f_l(x)$ to be the same at each boundary input location x, because observing $\delta_{k,l}(x) = 0$ means that we have observed $f_k(x)$ and $f_l(x)$ to be the same (see (19) and (20), for a formal proof). This implies that their posterior means (which are the GP regression predictive functions) and their posterior variances (which quantify the uncertainty in the predictions) will both be equal.

Suppose that we place B pseudo observations on each $\Gamma_{k,l}$. Let $\boldsymbol{x}_{k,l}$ denote the set of B input boundary locations chosen in $\Gamma_{k,l}$, let $\boldsymbol{\delta}_{k,l}$ denote a collection of the noiseless observations of $\delta_{k,l}(x)$ at the selected boundary locations, and let $\boldsymbol{\delta}$ denote the collection of all $\boldsymbol{\delta}_{k,l}$'s in the following order,

$$\boldsymbol{\delta}^T = (\boldsymbol{\delta}_{1,1}^T, \boldsymbol{\delta}_{1,2}^T, \ldots, \boldsymbol{\delta}_{1,K}^T, \boldsymbol{\delta}_{2,1}^T, \ldots, \boldsymbol{\delta}_{2,K}^T, \ldots, \boldsymbol{\delta}_{KK}^T).$$

Note that the observed pseudo value of $\boldsymbol{\delta}$ will be a vector of zeros, but its prior distribution (prior to observing the pseudo values or any response observations) is represented by the above covariance expressions. Additionally, let $f_*^{(k)} = f_k(x_*)$ denote the value of the response $f_k(x)$ at any $x_* \in \Omega_k$, and let $\boldsymbol{y}^T = (\boldsymbol{y}_1^T, \boldsymbol{y}_2^T, \ldots, \boldsymbol{y}_K^T)$. The prior joint distribution of $f_*^{(k)}$, $\boldsymbol{y}$ and $\boldsymbol{\delta}$ is

$$\begin{bmatrix} f_*^{(k)} \\ \boldsymbol{y} \\ \boldsymbol{\delta} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} 0 \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \begin{bmatrix} c_{**} & \boldsymbol{c}_{*\mathcal{D}}^{(k)} & \boldsymbol{c}_{*,\delta}^{(k)} \\ \boldsymbol{c}_{\mathcal{D}*}^{(k)} & \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{c}_{\delta,*}^{(k)} & \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta} \end{bmatrix} \right), \tag{5}$$

where the expressions of the covariance blocks are given by $c_{**} = \mathbb{C}\mathrm{ov}(f_*^{(k)}, f_*^{(k)})$,

$$\boldsymbol{c}_{*\mathcal{D}}^{(k)} = (\mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{y}_1), \mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{y}_2), \ldots, \mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{y}_K)),$$

$$\boldsymbol{c}_{*\delta}^{(k)} = (\mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{\delta}_{1,1}), \mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{\delta}_{1,2}), \ldots, \mathbb{C}\mathrm{ov}(f_*^{(k)}, \boldsymbol{\delta}_{K,K})),$$

$$\boldsymbol{c}_{\mathcal{D}\mathcal{D}} = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{y}_1) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{y}_2) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{y}_K) \\ \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{y}_1) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{y}_2) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{y}_K) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{y}_1) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{y}_2) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{y}_K) \end{bmatrix},$$

$$\boldsymbol{C}_{\mathcal{D},\delta} = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{\delta}_{1,2}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_1, \boldsymbol{\delta}_{K,K}) \\ \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{\delta}_{1,1}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_2, \boldsymbol{\delta}_{K,K}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{\delta}_{1,2}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{y}_K, \boldsymbol{\delta}_{K,K}) \end{bmatrix}, \text{ and}$$

$$\boldsymbol{C}_{\delta,\delta} = \begin{bmatrix} \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,1}, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,1}, \boldsymbol{\delta}_{1,2}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,1}, \boldsymbol{\delta}_{K,K}) \\ \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,2}, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,2}, \boldsymbol{\delta}_{1,1}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{1,2}, \boldsymbol{\delta}_{K,K}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{K,K}, \boldsymbol{\delta}_{1,1}) & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{K,K}, \boldsymbol{\delta}_{1,2}) & \ldots & \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{K,K}, \boldsymbol{\delta}_{K,K}) \end{bmatrix}.$$

Note that joint covariance matrix is very sparse, because of the many zero values in (3) and (4).

From the standard GP modeling results applied to the augmented data, the posterior predictive distribution of $f_*^{(k)}$ given $\boldsymbol{y}$ and the pseudo observations $\boldsymbol{\delta} = \boldsymbol{0}$ is Gaussian with mean

$$\mathbb{E}[f_*^{(k)} | \boldsymbol{y}, \boldsymbol{\delta} = \boldsymbol{0}] = (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\delta}^{(k)} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1} \boldsymbol{y}. \tag{6}$$

and variance

$$\begin{aligned} \mathbb{V}\mathrm{ar}[f_*^{(k)} | \boldsymbol{y}, \boldsymbol{\delta}] = {} & c_{**} - \boldsymbol{c}_{*,\delta}^{(k)} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{c}_{\delta*}^{(k)} \\ & - (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\delta}^{(k)} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1} (\boldsymbol{c}_{\mathcal{D}*}^{(k)} - \boldsymbol{C}_{\mathcal{D}\delta} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{c}_{\delta*}^{(k)}). \end{aligned} \tag{7}$$

The derivation of the predictive mean and variance can be found in Appendix A.

One implication of the continuity imposed by including the pseudo observations $\boldsymbol{\delta} = \boldsymbol{0}$ is that the posterior predictive means and variances of $f_*^{(k)}$ and $f_*^{(l)}$ for two neighboring regions Ω_k and Ω_l are equal at the specified input boundary locations $\mathbf{x}_{k,l}$; see Appendix B for details. The continuity imposed certainly does not guarantee that the posterior means and variances of $f_*^{(k)}$ and $f_*^{(l)}$ are equal for every $x^* \in \Gamma_{k,l}$, including those not in the locations of pseudo observations $\mathbf{x}_{k,l}$. Our numerical experiments in Section 5 demonstrate that as we place more pseudo inputs, the posterior means and variances of $f_*^{(k)}$ and $f_*^{(l)}$ converge to each other.

From the preceding, our proposed approach enforces that the two local GP models for two neighboring local regions have the same posterior predictive means and variances (and they satisfy an even stronger condition, that the responses themselves are identical) at the chosen set of boundary points corresponding to the pseudo observations. We view this as patching together the independent local models in a nearly continuous way. The chosen sets of boundary points serve as the stitches when patching together the pieces, and the more boundary points are chosen, the closer the models are to being continuous over the entire boundary. In light of this, we refer to the approach as *patchwork kriging*.

3.2 Hyperparameter Learning and Prediction

The hyperparameters of the covariance function(s) $c_k(\cdot, \cdot)$ determine the correlation among the values of $f(x)$, which has significant effect on the accuracy of a Gaussian process regression. We jointly estimate all correlation parameters (multiple sets of parameters if

different $c_k(\cdot,\cdot)$ are assumed for each region) using maximum likelihood estimation (MLE) by minimizing the negative log marginal likelihood,

$$\begin{aligned} NL(\theta) =& -\log p(\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}|\theta) \\ =& \frac{N}{2}\log(2\pi) + \frac{1}{2}\log\left|\begin{array}{cc} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta}\end{array}\right| + \frac{1}{2}[\boldsymbol{y}^T \mathbf{0}^T]\left[\begin{array}{cc} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta}\end{array}\right]^{-1}\left[\begin{array}{c}\boldsymbol{y}\\ \mathbf{0}\end{array}\right] \end{aligned} \tag{8}$$

Note that we have augmented the data to included the pseudo observations $\boldsymbol{\delta}=\mathbf{0}$ in the likelihood expression, which results in a better behaved likelihood by imposing some continuity across regions. This essentially allows data to be shared across regions when estimating the covariance parameters. Using the properties of a determinant for a partitioned matrix, the determinant part in the marginal likelihood becomes

$$\left|\begin{array}{cc} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta}\end{array}\right| = |\boldsymbol{C}_{\mathcal{D}\mathcal{D}}||\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\delta,\mathcal{D}}\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D},\delta}|,$$

which can be used to compute the log determinant term in $NL(\theta)$ as follows,

$$\log\left|\begin{array}{cc} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta}\end{array}\right| = \log|\boldsymbol{C}_{\mathcal{D}\mathcal{D}}| + \log\left|\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\delta,\mathcal{D}}\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D},\delta}\right|.$$

Note that the log determinant of the block diagonal matrix $\boldsymbol{C}_{\mathcal{D}\mathcal{D}}$ is equal to the sum of the log determinants of its diagonal blocks, and $\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\delta,\mathcal{D}}\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D},\delta}$ is very sparse, so the cholesky decomposition of the sparse matrix can be taken to evaluate their determinants; we will detail the sparsity discussion in the next section. Evaluating the quadratic term of the negative log marginal likelihood function involves the inversion of $(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)$. The inversion can be effectively evaluated using

$$(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1} = \boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1} + \boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}(\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta})^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}. \tag{9}$$

After the hyperparameters were chosen by the MLE criterion, evaluating the predictive mean and variance for the patchwork kriging model can be performed as follows. Let $\boldsymbol{Q}$ denote the inversion result of (9), and let $\boldsymbol{L}$ denote the cholesky decomposition of $\boldsymbol{C}_{\delta,\delta}$ such that $\boldsymbol{C}_{\delta,\delta} = \boldsymbol{L}\boldsymbol{L}^T$. After the pre-computation of the two matrices and $\boldsymbol{v} = \boldsymbol{L}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T$, the predictive mean (6) and the predictive variance (7) can be evaluated for each $x_* \in \Omega_k$,

$$\begin{aligned} \mathbb{E}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}] &= (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})\boldsymbol{Q}\boldsymbol{y} \\ \mathbb{V}\mathrm{ar}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}] &= c_{**} - \boldsymbol{w}_*^T\boldsymbol{w}_* - (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})\boldsymbol{Q}(\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})^T, \end{aligned} \tag{10}$$

where $\boldsymbol{w}_* = \boldsymbol{L}^{-1}(\boldsymbol{c}_{*\delta}^{(k)})^T$. The computation steps of patchwork kriging were described in Algorithm 1.

3.3 Sparsity and Complexity Analysis

The computational expense of patchwork kriging is much less than that of the global GP regression. The computational expense of the patchwork kriging model is dominated by evaluating the inversion (9). The inversion comes in two parts. The first part is to invert

Algorithm 1 Computation Steps for Patchwork Kriging

Require:
1: Decomposition of domain $\{\Omega_k; k = 1, \ldots, K\}$; see Section 3.4 for a choice.
2: Locations of pseudo data $\{\mathbf{x}_{k,l}; k, l = 1, \ldots, K\}$; see Section 3.4 for a choice.
3: Hyperparameters of covariance function $c_k(\cdot,\cdot)$; use the MLE criterion (8) for a choice.
Input: Data $\mathcal{D}$ and test location x_*
Output: $\mathbb{E}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}]$ and $\mathbb{V}\mathrm{ar}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}]$
4: **Evaluate** $\mathbf{Q} = (\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}$ using (9).
5: **Take** the Cholesky Decomposition of $\boldsymbol{C}_{\delta,\delta} = \boldsymbol{L}\boldsymbol{L}^T$.
6: **Evaluate** $\boldsymbol{v} = \boldsymbol{L}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T$.
7: **for** $x_* \in \Omega_k$ **do**
8: **Evaluate** $\boldsymbol{w}_* = \boldsymbol{L}^{-1}(\boldsymbol{c}_{*\delta}^{(k)})^T$.
9: $\mathbb{E}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}] = (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})\boldsymbol{Q}\boldsymbol{y}$.
10: $\mathbb{V}\mathrm{ar}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}=\mathbf{0}] = c_{**} - \boldsymbol{w}_*^T\boldsymbol{w}_* - (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})\boldsymbol{Q}(\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{w}_*^T\boldsymbol{v})^T$.
11: **end for**

$\boldsymbol{C}_{\mathcal{D}\mathcal{D}}$. Note that $\boldsymbol{C}_{\mathcal{D}\mathcal{D}}$ is a block diagonal matrix with the kth block size equal to the size of $\mathcal{D}_k$. If the size of each $\mathcal{D}_k$ is M, evaluating $\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}$ requires only inverting K matrices of size $M \times M$, and its expense is $O(KM^3)$. Given $\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}$, evaluating $\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}$ adds $O(KBM^2)$ to the computational expense.

The second part of the inversion (9) is to invert $\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}$. The matrix is very sparse, because $\mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{k,l}, \boldsymbol{\delta}_{u,v}) - \sum_{m=1}^K \mathbb{C}\mathrm{ov}(\boldsymbol{\delta}_{k,l}, \boldsymbol{y}_m)\,\mathbb{C}\mathrm{ov}(\boldsymbol{y}_m, \boldsymbol{y}_m)^{-1}\,\mathbb{C}\mathrm{ov}(\boldsymbol{y}_m, \boldsymbol{\delta}_{u,v})$ is a zero matrix unless the tuple (k, l, u, v) satisfies the non-zero conditions listed in (4). The symmetric sparse matrix can be converted into a symmetric sparse banded matrix by the reverse Cuthill-McKee algorithm (Chan and George, 1980), and the computational complexity of the conversion algorithm is linearly proportional to the number of non-zero elements in the original sparse matrix. Let d_f denote the number of neighboring local regions of each local region, and B denote the number of pseudo observations placed per boundary. The number of non-zero elements in the sparse matrix is $O(d_f BK)$, so the time complexity of the reverse Cuthill-McKee algorithm is $O(d_f BK)$. The bandwidth of the resulting sparse matrix is linearly proportional to $d_f B$, and the size of the matrix is proportional to $d_f BK$. The complexity of taking the inverse of a symmetric banded matrix with size r and bandwidth p through Cholesky decomposition is $O(rp^2)$ (Golub and Van Loan, 2012, pp. 154). Therefore, the complexity of inverting $\boldsymbol{C}_{\delta,\delta} - \boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}$ is $O(d_f^3 B^3 K)$. Note that $d_f \propto d$ if data are more densely distributed over the entire input dimensions, and $d_f \propto d'$ if data are a d'-dimensional embedding in d dimensional space. The complexity becomes $O(d^3 B^3 K)$ for the worst case scenario.

Therefore, the total computational expense of the inversion (9) is $O(KM^3 + KBM^2 + d_f^3 B^3 K)$. Typically, $B \ll M$, in which case the complexity is $O(KM^3 + d_f^3 B^3 K)$. Note that $M \approx N/K$, where the approximation error is due to rounding N/K to an integer value. Therefore, the complexity can be written as $O(N^3/K^2 + d_f^3 B^3 K)$. Given a fixed data size N, more splits of the regression domain will reduce the computation time due to the first term in the complexity, but too much increase would increase the second term, which will

be shown later in Section 6. When data are dense in the input space, $d_f \propto d$, for which the complexity would increase in d^3. For $d < 10$, the effect is pretty ignorable unless B is very large, but it becomes significant when $d > 10$. This computation issue related to data dimensions basically suggests to limit the practical use of this method to d less than 100. We will later discuss more on this issue and how to choose K and B to balance off the computation and prediction accuracy in Section 5.2.

In addition to the computational expense of the big inverse, additional incremental computations are needed per each test location x_*. The first part is to evaluate for the predictive mean and variances,

$$(\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\delta}^{(k)} \boldsymbol{C}_{\delta,\delta}^{-1} \boldsymbol{C}_{\mathcal{D}\delta}^{T}). \tag{11}$$

Note that the elements in $\boldsymbol{c}_{*\mathcal{D}}^{(k)}$ are mostly zero except for the columns that correspond to $\mathcal{D}_k$ (size M), and similarly most elements of $\boldsymbol{c}_{*\delta}^{(k)}$ are zero except for the columns that correspond to $\boldsymbol{\delta}_{k,l}$'s (size $d_f B$). The cost of evaluating (11) is $O(M + d_f B)$. Therefore, the cost of the predictive mean prediction per a test location is $O(M + d_f B)$, and the cost for the predictive variance is $O((M + d_f B)^2)$. When data are dense in the input dimensions, the costs become $O(M + dB)$ and $O((M + dB)^2)$.

3.4 Tuning Parameter Selection

The performance of the proposed patchwork kriging method depends on the choice of tuning parameters, including the number of partitions (K) and the number (B) and locations ($\mathbf{x}_{k,l}$) of the pseudo observations. This section presents guidelines for these choices. Choosing the locations of pseudo observations is related to the choice of domain partitioning. In this section, we discuss the choices of the locations and partitioning together.

There are many existing methods to partition a large set of data into smaller pieces. The simplest spatial partitioning is a uniform grid partitioning that divides a domain into uniform grids and splits data accordingly (Park et al., 2011; Park and Huang, 2016). This is simple and effective if the data are uniformly distributed over a low dimensional space. However, if the input dimension is high, it would either generate too many regions or it would produce many sparse regions that contain very few or no observations, and the latter also happens when the data are non-uniformly distributed; see examples in Figure 2-(c) and Figure 2-(e). Shen et al. (2006) used a kd-tree for spatial partitioning of unevenly distributed data points in a high dimensional space. A kd-tree is a recursive partitioning scheme that recursively bisects the subspaces along one chosen data dimension at a time. Later, McFee and Lanckriet (2011) generalized it to the spatial tree. Starting with a level 0 space $\Omega_1^{(0)}$ equal to the entire regression domain, the spatial tree recursively bisects each of level s spaces into two level $s+1$ spaces. Let $\Omega_j^{(s)} \in \mathbb{R}^d$ denote the jth region in the level s space. It is bisected into two level $s+1$ spaces as

$$\Omega_{2j-1}^{(s+1)} = \{\boldsymbol{x} \in \Omega_j^{(s)} : \boldsymbol{v}_{j,s}^T \boldsymbol{x} \le \nu\} \text{ and } \Omega_{2j}^{(s+1)} = \{\boldsymbol{x} \in \Omega_j^{(s)} : \boldsymbol{v}_{j,s}^T \boldsymbol{x} > \nu\}. \tag{12}$$

Each of $\Omega_{2j-1}^{(s+1)}$ and $\Omega_{2j}^{(s+1)}$ will be further partitioned in the next level using the same procedure. The choice of the linear projection vector $\boldsymbol{v}_{j,s}$ depends on the distribution of the

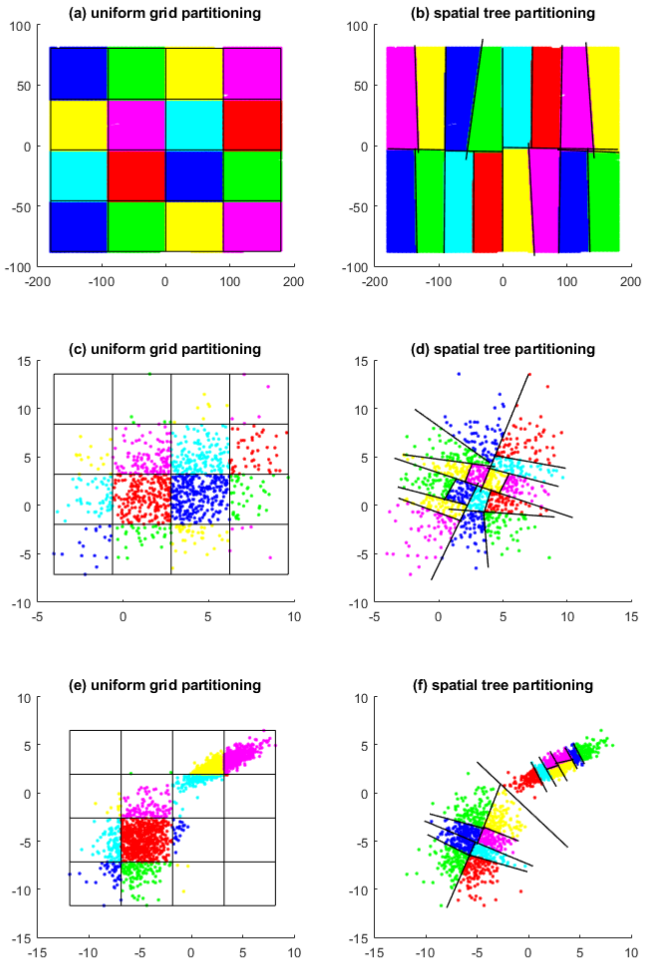


Figure 2: Comparison of two spatial partitioning schemes: a uniform grid (left panel) and a spatial tree (right panel). The spatial tree generates data partitioning of uniform sizes when data is unevenly distributed, but the uniform grid does not.

local data belonging to $\Omega_j^{(s)}$. For example, it can be the first principal component direction of the local data. The value for ν is chosen so that $\Omega_{2j-1}^{(s+1)}$ and $\Omega_{2j}^{(s+1)}$ have an equal number of observations. In this sense, the subregions at the same level are equally sized, which helps to level off the computation times of the local models. When the spatial tree is applied on data uniformly distributed over a rectangular domain, it produces a uniform grid partitioning; see examples in Figure 2-(b). The spatial tree is more effective than the grid partitioning when data is unevenly distributed; see examples in Figure 2-(d) and Figure 2-(f).

In this work, we use a spatial tree with the principal component (PC) direction for $\boldsymbol{v}_{j,s}$. Bisecting a space along the PC direction has effects of minimizing the area of the interfacial boundaries in between the two bisected regions, so the number of the pseudo observations necessary for connecting the two regions can be minimized. The maximum level of the recursive partitioning depends on the choice of K via $s_{max} = \lfloor \log_2 K \rfloor$. The choices of K and B will be discussed in the next section. Given B, the pseudo observations $\mathbf{x}_{k,l}$ are randomly generated from an uniform distribution over the intersection of the hyper-plane $\boldsymbol{v}_{j,s}^T \boldsymbol{x} = \nu$ and the level s region $\Omega_j^{(s)}$.

4. Illustrative Example

To illustrate how patchwork kriging changes the model predictions (relative to a set of independent GP models over each region, with no continuity conditions imposed), we designed the following simple simulation study; we will present more comprehensive simulation comparisons and analyses in Section 5. We generated a dataset of 6,000 noisy observations

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \ldots, 6000,$$

from a zero-mean Gaussian process with an exponential covariance function of $c(x_i, x_j) = 10 \exp(-||x_i - x_j||_2)$, where $x_i \sim \text{Uniform}(0, 10)$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ are independently sampled, and $f(x_i)$ is simulated by the R package `RandomField`. Three hundred of the 6,000 observations were randomly selected as the training data $\mathcal{D}$, while the remaining 5,700 were reserved for test data. Figure 3 illustrates how the patchwork kriging predictor changes for different K, relative to the global GP predictor and the regular local GP predictor with no continuity conditions across regions. As the number of regions (K) increases, the regular local GP predictor deviates more from the global GP predictor. The test prediction mean square error (MSE) for the regular local GP predictor at the 5,700 test locations is 0.0137 for $K = 4$, 0.0269 for $K = 8$, 0.0594 for $K = 16$, and 0.1268 for $K = 32$. In comparison, patchwork kriging substantially improves the test MSE to 0.0072 for $K = 4$, 0.0123 for $K = 8$, 0.0141 for $K = 16$, and 0.0301 for $K = 32$.

We also generated a synthetic dataset in 2-d using the R package `RandomField`, and we denote this dataset by `synthetic-2d`. `synthetic-2d` consists of 8,000 noisy observations from a zero-mean Gaussian process with the exponential covariance function of $c(x_i, x_j) = 10 \exp(-||x_i - x_j||_2)$,

$$y_i = f(x_i) + \epsilon_i \quad \text{for } i = 1, \ldots, 8000,$$

where $x_i \sim \text{Uniform}([0, 6] \times [0, 6])$ and $\epsilon_i \sim \mathcal{N}(0, 1)$ were independently sampled. We used the dataset to illustrate how two local GP models for two neighboring local regions change

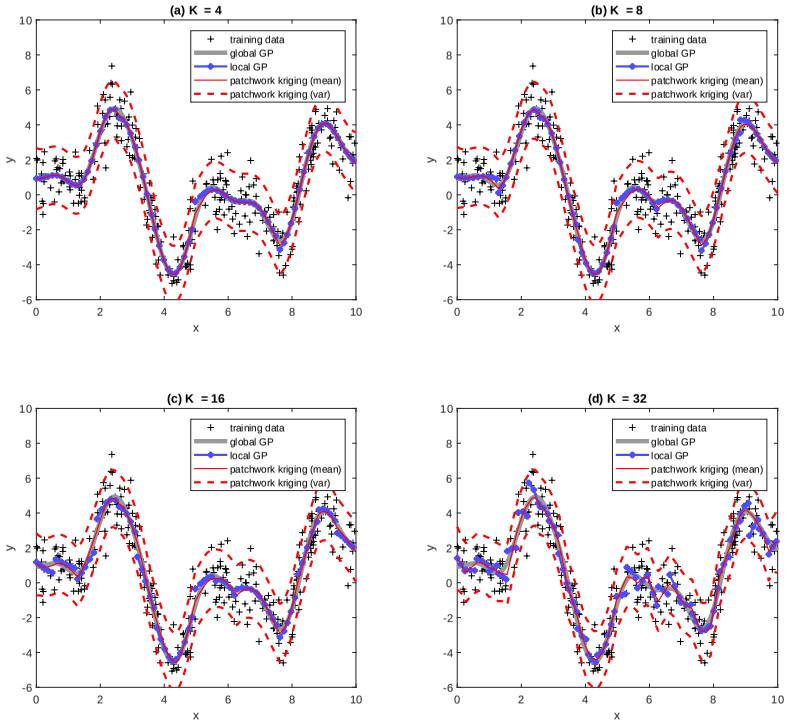


Figure 3: Example illustrating the patchwork kriging predictor, together with the global GP predictor and the regular local GP predictor with no continuity constraints. K is the number of local regions.

as B changes. We first partitioned the dataset into 128 local regions as shown in Figure 4-(a). For evaluation purposes, we considered test points that fell on the boundary cutting the entire regression domain into two (indicated by the black solid line in Figure 4-(a)), and sampled 201 test points uniformly over this boundary; the test locations do not coincide with the locations that pseudo observations placed. For each point, we get two mean predictions from the two local patchwork kriging models that straddle the boundary at that point. We compared the two mean predictions to each other for different choices of B and also compared them with the optimal global GP prediction, i.e., the prediction using the true GP covariance function and the entire dataset globally without spatial partitioning. Figure 4 shows the comparison. When $B = 0$, the two local models exhibited significant differences in their mean predictions. The differences decreased as B increased, and became negligible

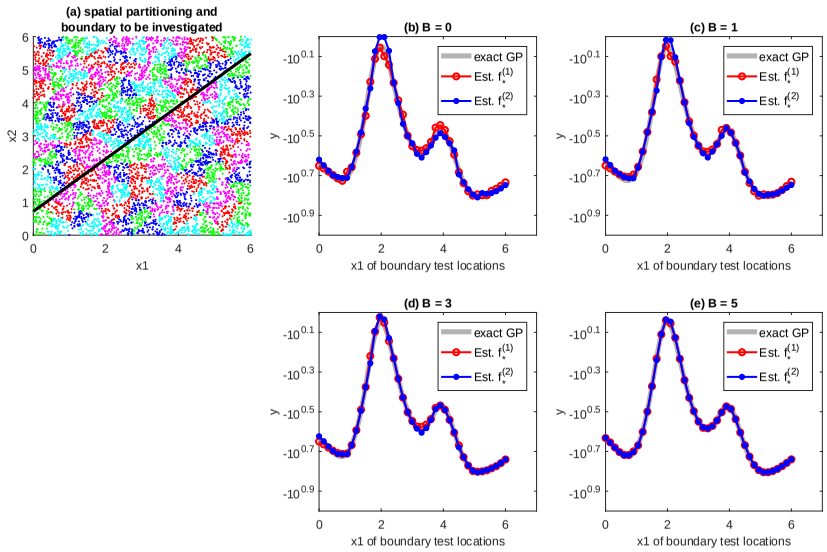


Figure 4: Comparison of the patchwork kriging mean predictions of two local models over interfacial boundaries. Panel (a) shows how the entire regression domain was spatially partitioned into 128 local regions, which are distinguished by their colors. The black solid line cutting through the entire space is the interfacial boundary that we selected to study the behavior of the patchwork kriging at interfacial boundaries. Panels (b), (c), (d) and (e) compare the patchwork kriging mean predictions of the neighboring local models when $B = 0$, 1, 3, and 5. In the panels, the horizontal axes represent the x1 coordinates of test locations on the solid interfacial boundary line shown in panel (a). $f_*^{(1)}$ and $f_*^{(2)}$ denote the mean predictions of the two local models on each side of the solid boundary line. As B increases, the two local predictions converge to each other, and the converged values are very close to the benchmark predictor achieved using the true GP model globally.

when $B \geq 5$. The mean predictions were also very close to the exact GP predictions. The similar results were observed in different simulated examples, which will be discussed in Section 5.

5. Evaluation with Simulated Examples

In this section, we use simulation datasets to understand how the patchwork kriging behaves under different input dimensions, covariance ranges and choices of K and B.

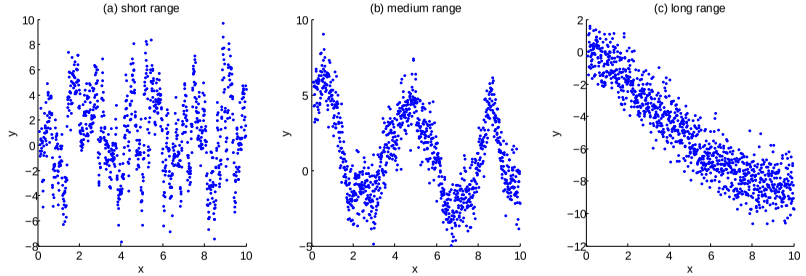


Figure 5: Illustrative Data with Short-range, Med-range and Long-range Covariances

5.1 Datasets and Evaluation Criteria

Simulation datasets are sampled from a Gaussian process with the squared exponential covariance function,

$$c(x_i, x_j) = \tau \exp\left(-\frac{(x_i - x_j)^T(x_i - x_j)}{2\rho^2}\right) \text{ for } x_i, x_j \in \mathbb{R}^d, \tag{13}$$

where $\tau > 0$ is the scale parameter, and $\rho > 0$ determines the range of covariance. We randomly sampled 10,000 pairs of x_i and y_i. Each x_i is uniformly from $[0, 10]^d$, and then evaluate the sample covariance matrix with τ and ρ for the 10,000 sampled inputs, $\boldsymbol{C}_{\tau,\rho}$. All y_i's are jointly sampled from $\mathcal{N}(\boldsymbol{0}, \sigma^2\boldsymbol{I} + \boldsymbol{C}_{\tau,\rho})$. We fixed $\tau = 10$ and $\sigma^2 = 1$, but chose ρ to 0.1 (short range), 1 (med range) or 10 (long range) to simulate datasets having different covariance ranges; see Figure 5 for illustrating simulated datasets for one dimensional input. In addition, the input dimension d was varied over $\{2, 5, 10, 100\}$. In total, we considered 12 different combinations of different ρ and d values. For each combination, we drew 50 datasets, so there are 600 datasets in total.

For each of the datasets, we ran the patchwork kriging with different choices of $K \in \{16, 32, 64, 128, 256\}$ and $B \in \{0, 1, 2, 3, 5, 7, 10, 15, 20, 25\}$. For each run, we evaluate the computation time and prediction accuracy of patchwork kriging. For the prediction accuracy, we first computed the predictive mean of the optimal GP predictor (i.e., using the true exponential covariance function) at test locations, and we used these optimal prediction values as a benchmark against which to judge the accuracy of the patchwork kriging predictions. One thousand test locations are uniformly sampled from the interior of local regions, denoted by $\{x_t; t = 1, ..., T_I\}$, and 200 additional test locations were uniformly sampled from the boundaries between local regions, which are denoted by $\{x_t; t = T_I + 1, ..., T_I + T_B\}$. Let μ_t denote the estimated posterior predictive mean at location x_t, and let $\tilde{\mu}_t$ denote the benchmark predictive mean at the same location. We measure three performance metrics for the mean predictions. The first two measures are the interior mean squared error (I-MSE) and the boundary mean squared error (B-MSE)

$$\text{I-MSE} = \frac{1}{T_I}\sum_{t=1}^{T_I}(\tilde{\mu}_t - \mu_t)^2, \text{ and B-MSE} = \frac{1}{T_B}\sum_{t=T_I+1}^{T_I+T_B}(\tilde{\mu}_t - \mu_t)^2, \tag{14}$$

which measure the average accuracy of the mean prediction inside local regions and on the boundary of local regions. For each boundary point in $\{x_t; t = T_I + 1, ..., T_I + T_B\}$, we get two mean predictions from the two local patchwork kriging models that straddle the boundary at that point. In the B-MSE calculation, we took one of the two predictions following the rule: when $x_* \in \Gamma_{kl}$, choose the prediction for $f_*^{(k)}$ if $k < l$. Please note that when a test location is at a corner where three or more local regions meet, we do have more than two predictions, which did not happen in all of our testing scenarios. We also evaluated the squared difference of the two mean predictions for each of 200 boundary points, and the mean squared mismatch (MSM) was defined as the average of the squared differences. We also measured the three performance metrics for the variance predictions, which were named 'I-MSE(σ^2)', 'B-MSE(σ^2)' and 'MSM(σ^2)' respectively.

5.2 Analysis of the Outcomes and Choices of K and B

Figure 6 shows the I-MSE, B-MSE and MSM performance of the patchwork kriging for different covariance ranges and different choices of K and B when $d = 100$, and Appendix C contains the plots of all six performance metrics for all simulation configurations. All of the performance metrics have shown the similar patterns:

- **Covariance Ranges:** All of the performance metrics became negligibly small for medium-range and long-range covariances with large B. This implies that the patchwork kriging approximates the full GP very well for medium-range and longer-range covariances; please see Appendix for detailed plots. This result is opposite to our initial expectation that local-based approaches would have some deviations from the full GP for long-range covariances. As long as the underlying covariance is stationary, the proposed approach works well for long-range covariance cases.

- **Effect of B:** All of the metrics decrease in B but does not change much for B above 8 for medium-range and long-range covariances. However, when the covariance range is short, the improvement of the three metrics goes slower. This implies larger B is required to achieve good accuracy for short-range covariances.

- **Effect of K:** All of the metrics increase in K when the other conditions are kept same. This is understandable, because the simulated data came from a stationary process. However, the effect of K on the three metrics was relatively small when $B > 7$ and covariance ranges are medium or long. Since the computational complexity of the proposed method decreases with increase of K, choosing a large K with $B > 7$ could be a computationally economic option with good prediction accuracy. See our computation time analysis below for an additional discussion on the choices of K and B.

- **Boundary Consistency:** Both of the MSM and MSM(σ^2) goes to zero as B increases for medium and long-range covariances. This implies that if data change smoothly, the patchwork idea does not only guarantee the same predictions on the locations pseudo data placed but also gives the same predictions over the entire inter-domain boundaries.

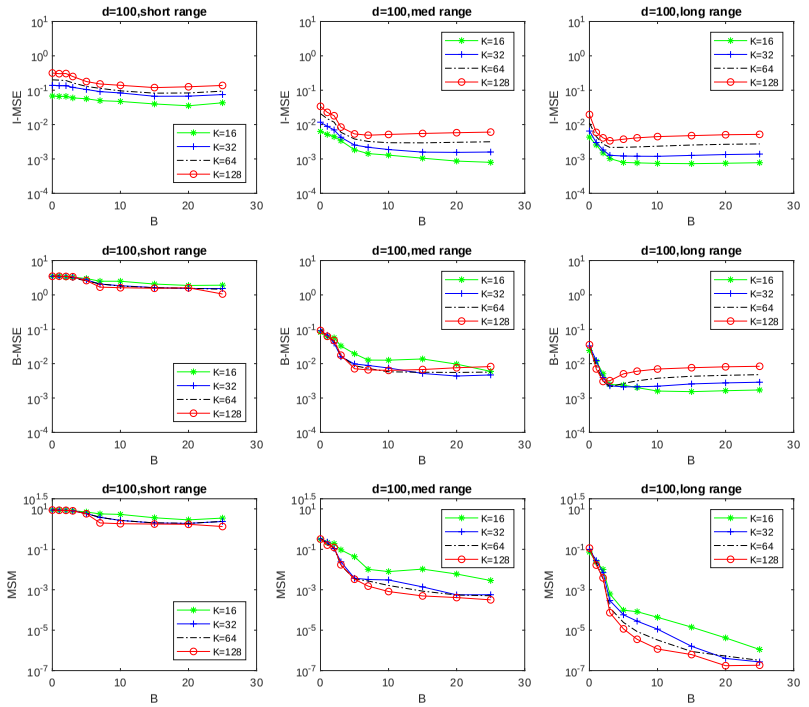


Figure 6: Performance of Patchwork Kriging for Simulated Cases with 100 Input Dimensions.

Figure 7 summarizes the total computation time of the patchwork kriging for different configuration.

- It appears that the input dimension is not a determining factor of the time if the input dimension $d \leq 10$, but for $d > 10$, it became a major factor to affect the time. As we discussed in Section 3.3, the computational complexity of the patchwork kriging is $O(N^3/K^2 + d_f^3 B^3 K)$. When data are uniformly located over the regression domain, $d_f \propto d$ and the computation of the patchwork kriging is scaling proportionally to d^3.

- When $d \leq 10$, the deciding factor for the computation time was K. In general, larger K gave shorter computation times.

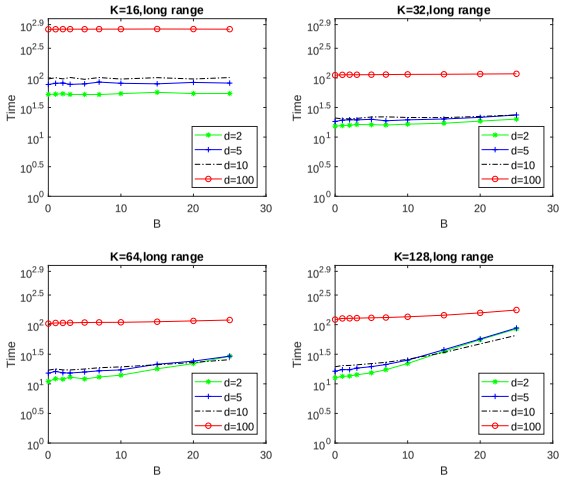


Figure 7: Summary of Total Computation Times for Simulated Cases.

- With larger K, B becomes more influential to the total computation time. This is due to the increase of the second term in the overall computational complexity, $O(N^3/K^2 + d_f^3 B^3 K)$.
- To keep the total computation time lower, both of N/K and $d_f B$ should be kept lower. On the other hand, N/K and $d_f B$ cannot be too small due to degradation of prediction accuracy with a large K and a small B.

Our numerical studies suggest to choose K so that N/K be in between 200 and 600 and then choose B so that $d_f B$ is in between 15 and 400 to balance off the computation and prediction accuracy; these were based on all of the simulation cases presented in this section as well as the six real data studies that will be presented in the next section. In order to keep $d_f B \leq 400$ for efficient computation and $B \geq 7$ for prediction accuracy, $d_f \leq 400/7$. Therefore, the proposed approach would benefit more for $d_f \leq 55$. However, the proposed approach still worked better than some existing approaches for the simulated cases with $d = 100$; see the numerical results in Sections 5.3 and 5.4.

5.3 Comparison to a Global Approximation Method

We also used the simulated cases to compare the patchwork kriging to a global GP approximation method, the Fully Independent Training Conditional (FITC) algorithm (Snelson and Ghahramani, 2006, FITC). We decided to compare ours with the global GP approximation method because we thought that the global GP approximation would work better when stationary covariances are used. For the patchwork kriging, we fixed $B = 7$ and varied $K \in \{32, 64, 128, 256\}$. For the FITC, the total number of pseudo inputs was varied over

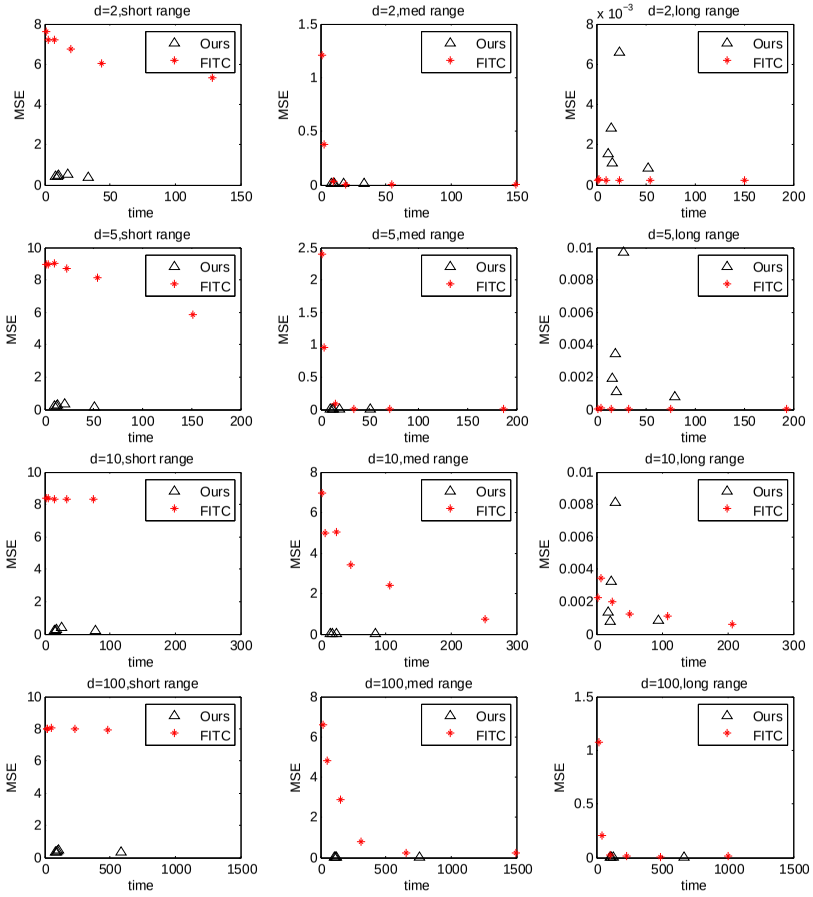


Figure 8: Comparison of Total Computation Times vs. MSE for Simulated Cases; triangles and stars represent the results of the patchwork kriging and FITC respectively.

$\{16, 32, 64, 128, 256, 512\}$. The computation times versus MSE of the mean prediction were compared for different input dimensions and covariance ranges. Figure 8 summarizes the outcome. The patchwork kriging outperformed the FITC with significant performance gaps for all short covariance and medium range covariance cases. For long range covariance cases,

the FITC performed better when $d < 10$, but the patchwork kriging performed comparably when $d \geq 10$.

The performance gap in between the FITC and the patchwork kriging can be explained by a more efficient computation of the patchwork kriging. Both of the FITC and patchwork kriging use pseudo inputs. Their accuracies depend on the total number of pseudo inputs used. When Q pseudo inputs were applied for both of FITC and the patchwork kriging, the computation of the FITC involves the inversion of a $Q \times Q$ dense matrix, while the computation of the patchwork kriging involves the inversion of the sparse matrix of the same size that corresponds to equation (9). Therefore, when comparable computation times were invested, the patchwork kriging could place more pseudo inputs than the FITC, so it can give better accuracy. In addition, the locations of the pseudo inputs in the FITC need to be learned together with covariance hyperparameters, and the increase in the number of pseudo inputs would increase the computation time for hyperparameter learning.

5.4 Comparison to Local Approximation Methods

We also used the simulated cases to compare the patchwork kriging to two local GP approximation methods, a robust Bayesian committee machine (Deisenroth and Ng, 2015, RBCM), and a partially independent conditional approach (Snelson and Ghahramani, 2007, PIC). For the patchwork kriging, we fixed $B = 7$ and varied $K \in \{32, 64, 128\}$. For RBCM, we used $K \in \{32, 64, 128\}$. For PIC, we used $K \in \{32, 64, 128\}$ with the total number of pseudo inputs fixed to 128. Figure 9 summarizes the comparison of MSE performance. The patchwork kriging performed very competitively for all simulated cases. The significant increase of computation time for the input dimension more than 10 was observed for all of the compared methods.

6. Evaluation with Real Data

In this section, we use five real datasets to evaluate the patchwork kriging and compare it with the state-of-the-art, including (Park and Huang, 2016, PGP), a Gaussian Markov random field approximation (Lindgren et al., 2011, GMRF), a robust Bayesian committee machine (Deisenroth and Ng, 2015, RBCM), and a partially independent conditional approach (Snelson and Ghahramani, 2007, PIC). The comparison with one additional dataset is presented in Appendix D.

6.1 Datasets and Evaluation Criteria

We considered five real datasets: two spatial datasets in 2-d with different spatial distributions of observations, one additional spatial dataset with a very large data size, and three higher dimensional datasets, one with 9-dimension, another with 21-dimension and the other with 8-dimension.

The first spatial dataset, `TCO.L2`, 182,591 observations collected by the NIMBUS-7/TOMS satellite, which measures the total column of ozone over the globe on Oct 1 1988. Two predictors represent the longitude and latitude of a measurement location, while the corresponding independent variable represents the measurement at the location. The observations are uniformly spread over the range of the two predictors. The main analysis objective with this dataset is to predict the total column of ozone at unobserved locations.

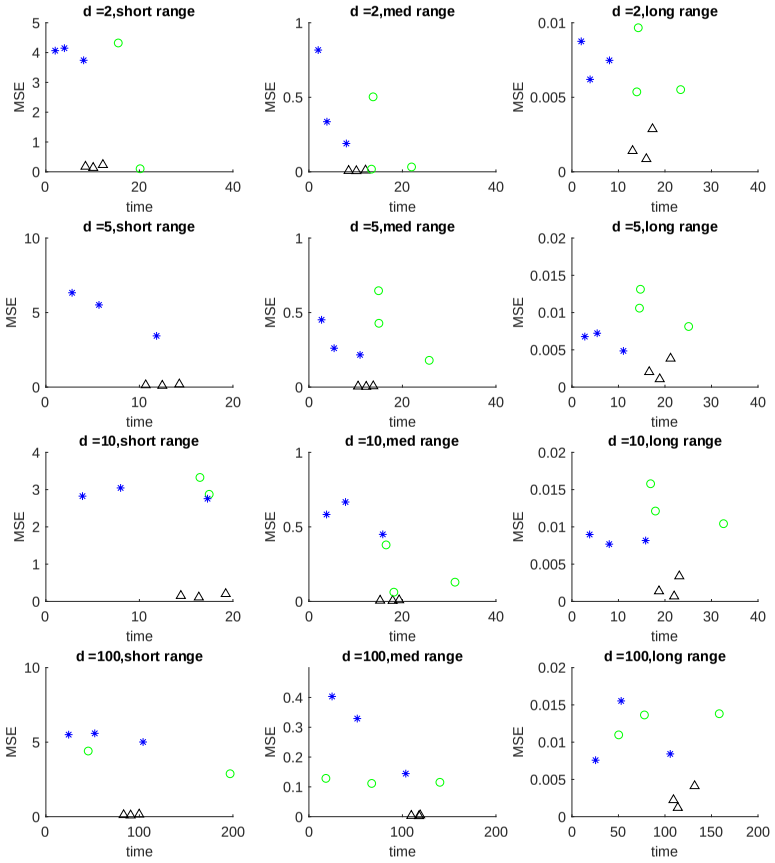


Figure 9: Comparison of Total Computation Times vs. MSE for Simulated Cases; triangles, stars and circles represent data for the patchwork kriging, PIC and RBCM respectively. The number of circles are not always same because PIC could not produce outcomes for some simulated cases due to singularity in numerical inversion.

The second dataset, `ICETHICK`, contains ice thickness measurements at 32,481 locations on the western Antarctic ice sheet and is available at `http://nsidc.org/`. It has two predictors that represent the longitude and latitude of a measurement location, and the

corresponding independent variable is the ice thickness measurement. The dataset has many sparse regions where there are very few observations. Regression analysis with this dataset would give the prediction of ice thickness at unobserved locations.

The third dataset, `PROTEIN`, has nine input variables that describe the tertiary structure of proteins and one independent variable that describes the physiochemical property of proteins. These data, which are available at `https://archive.ics.uci.edu/ml/datasets`, consist of 45,730 observations. Like typical high dimensional datasets, the measurements are embedded on a low dimensional subspace of the entire domain. This dataset can be studied to relate the structure of a protein with the physiochemical property of the protein for predicting the property from the structure.

The fourth dataset, `SARCOS`, contains measurements from a seven degrees-of-freedom SARCOS anthropomorphic robot arm. There are 21 predictors that describe the positions, moving velocities and accelerations of seven joints of the robot arm, and the seven response variables are the corresponding torques at the seven joints. We only use the first response variable for this numerical study. The dataset, which is available at `http://www.gaussianprocess.org/gpml/data/`, contains 44,484 training observations and 4,449 test observations. The main objective of the regression analysis is to predict one of the joint torques in a robot arm when the values of the predictors are available.

The last dataset, `FLIGHT`, consists of 800,000 flight records randomly selected from the database available at `http://stat-computing.org/dataexpo/2009/`. The same size subset of the database was used as a benchmark dataset in literature (Hensman et al., 2013). Following the use in the literature, we used 8 predictors that include the age of the aircraft, distance that needs to be covered, airtime, departure time, arrival time, day of the week, day of the month and month, and the response variable is the arrival time delay. This dataset was studied to predict the flight delay time when the predictors are given.

Using the five datasets, we compare the computation time and prediction accuracy of patchwork kriging with other methods. We randomly split each dataset into a training set containing 90% of the total observations and a test set containing the remaining 10% of the observations. To compare the computational efficiency of methods, we measure total computation times. For comparison of prediction accuracy, we measure two performance metrics on the test data, denoted by $\{(x_t, y_t) : t = 1, \ldots, T\}$, where T is the size of the test set. Let μ_t and σ_t^2 denote the estimated posterior predictive mean and variance at location x_t; when the testing location x_t is in the domain boundary Γ_{kl}, we may have two predictions, one for $f^{(k)}(x_t)$ and the other for $f^{(l)}(x_t)$, for which we choose one for $f^{(k)}(x_t)$ if $k < l$. Please note that when a test location is at a corner where three or more local regions meet, we do have more than two predictions, which did not happen in all of our testing scenarios. We also evaluated the squared The first measure is the mean squared error (MSE)

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \mu_t)^2, \tag{15}$$

which measures the accuracy of the mean prediction μ_t at location x_t. The second measure is the negative log predictive density (NLPD)

$$\text{NLPD} = \frac{1}{T}\sum_{t=1}^{T}\left[\frac{(y_t - \mu_t)^2}{2\sigma_t^2} + \frac{1}{2}\log(2\pi\sigma_t^2)\right]. \tag{16}$$

The NLPD quantifies the degree of fitness of the estimated predictive distribution $\mathcal{N}(\mu_t, \sigma_t^2)$ for the test data. These two criteria are used broadly in the GP regression literature. A smaller value of MSE or NLPD indicates better performance. All numerical experiments were performed on a desktop computer with Intel Xeon Processor W3520 and 6GB RAM.

The comparison was made in between our method and the state-of-the-art previously listed. Note that the PGP and the GMRF approaches cannot be applied for more than two input dimensions, and so were only compared for the three spatial datasets. We tried two covariance functions, a squared exponential covariance function and an exponential covariance function. Note that the PIC method does not work with an exponential covariance function because learning the pseudo inputs for the PIC method requires the derivative of a covariance function but an exponential covariance function is not differentiable. On the other hand, when an squared exponential covariance function is applied to the GMRF, the precision matrix construction is not straightforward. Therefore, we used a squared exponential covariance function for comparing the proposed approach with the PIC, RBCM, and PGP, while using an exponential covariance function for comparing it with the GMRF. For both of the cases, we assumed the same hyperparameters for local regions, and we used the entire training dataset to estimate the hyperparameters.

We chose and applied different partitioning methods for the compared methods. The choice of the partitioning schemes for the patchwork kriging and PGP is restrictive because every local region needs to be simply connected to minimize the area of the boundaries between local regions, so we used the spatial tree. The GMRF comes with a mesh generation scheme instead of a partitioning scheme, and following the suggestion by the GMRF's authors, we used the voronoi-tessellation of training points for the mesh generation. We tested the k-means clustering and the spatial tree for PIC and RBCM, but the choice did not make much difference in their performance. The results reported in this paper were the ones with the k-means clustering.

We tried different numbers of the local regions that partition an input domain, and the numbers of the local regions were ranged so that the numbers of observations per local region would be approximately in between 80 and 600 for the proposed approach. The numbers were similarly ranged for the other compared methods with some variations to have the computation times of all the compared methods comparable; note that we like to compare the prediction accuracies of the methods when the computation times spent are comparable. For patchwork kriging, the locations of pseudo observation were selected using the rule described in Section 3.4. For PIC, the locations were regarded as hyperparameters and were optimized using marginal likelihood maximization.

6.2 Example 1: `TCO.L2` Dataset

This dataset has two input dimensions, and the inputs of the data are densely distributed over a rectangular domain. For patchwork kriging, we varied $B \in \{3, 5\}$ and $K \in \{256, 512, 1024\}$.

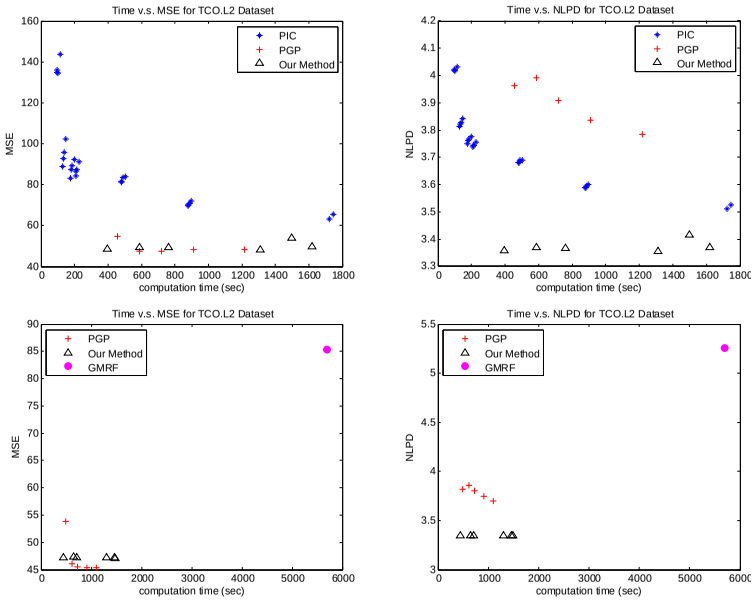


Figure 10: Prediction accuracy versus total computation time for the `TCO.L2` data.

The prediction accuracy of the PGP did not depend on the number of local regions K, so we fixed $K = 623$, while the number of finite element meshes per local region was varied from 5 to 25 with step size 5. For RBCM, we varied the number of local experts $K \in \{100, 150, 200, 250, 300, 600\}$. For PIC, K was varied over $\{100, 200, 300, 400, 600\}$, and the total number of pseudo inputs was also varied over $\{50, 70, 80, 100, 150, 200, 300\}$.

Figure 10 shows the main results. The shortest computation time of RBCM (2319 seconds) was much longer than the longest time of the other compared methods, while its MSE was not competitive as well. Therefore we did not plot its results in the figure. For both of the square exponential and the exponential covariance functions, our approach and the PGP approach had comparable MSE. However, our approach significantly outperformed the PGP and PIC approaches in terms of the NLPD. This implies that our approach provides more accurate variance estimations.

6.3 Example 2: `ICETHICK` Dataset

One characteristic of this dataset is the presence of many spatial voids where there are no or very little data points. For patchwork kriging, we varied $B \in \{3, 5, 7\}$ and $K \in \{64, 128, 256, 512, 1024\}$. For the PGP, we used $K = 47$, while the number of finite element meshes per local region was varied from 5 to 40 with step size 5. For RBCM, we

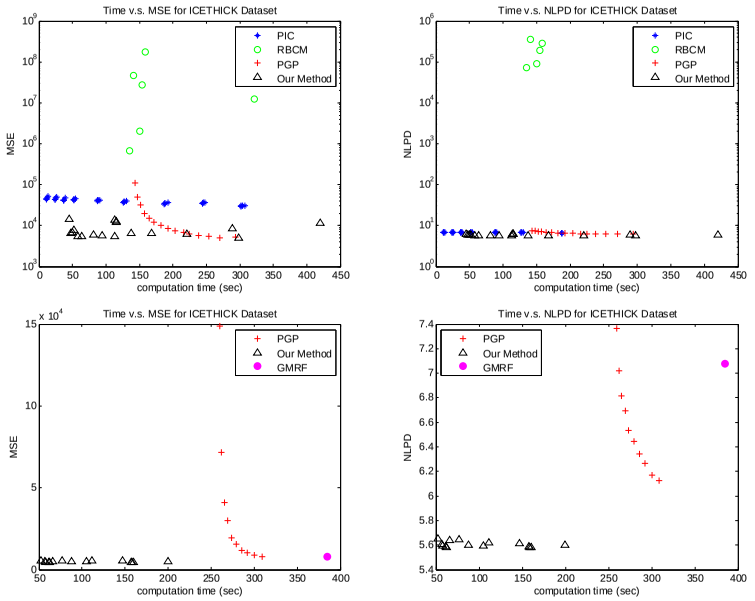


Figure 11: Prediction accuracy versus total computation time for the `ICETHICK` data. A squared exponential covariance function was used for the results in the top panel, while an exponential covariance function was used for the results in the bottom panel.

varied the number of local experts $K \in \{50, 100, 150, 200, 250, 300\}$. For PIC, M was varied over $\{50, 100, 150, 200\}$, and the total number of pseudo inputs was also varied over $\{50, 100, 150, 200, 300, 400, 500, 600, 700\}$.

Figure 11 compares the MSE and NLPD performance of the methods. Again, the PGP approach and the proposed approach outperformed the other methods, and the proposed approach achieved the best accuracy with much less computation time than the PGP approach. In addition, the proposed approach uniformly outperformed the other methods in terms of the NLPD. In other words, the proposed approach gives a predictive distribution that better fits the test data.

6.4 Example 3: `PROTEIN` Dataset

Different from the previous datasets, this dataset features nine input variables. We will use this dataset to see how the proposed approach works for input dimension more than two. For the patchwork kriging, we varied $B \in \{2, 3, 4\}$ and varied $K \in \{64, 128, 256\}$; we have not included the results for larger B because a larger B increased the computation times of the patchwork kriging to a range incomparable to those of the other algorithms. The

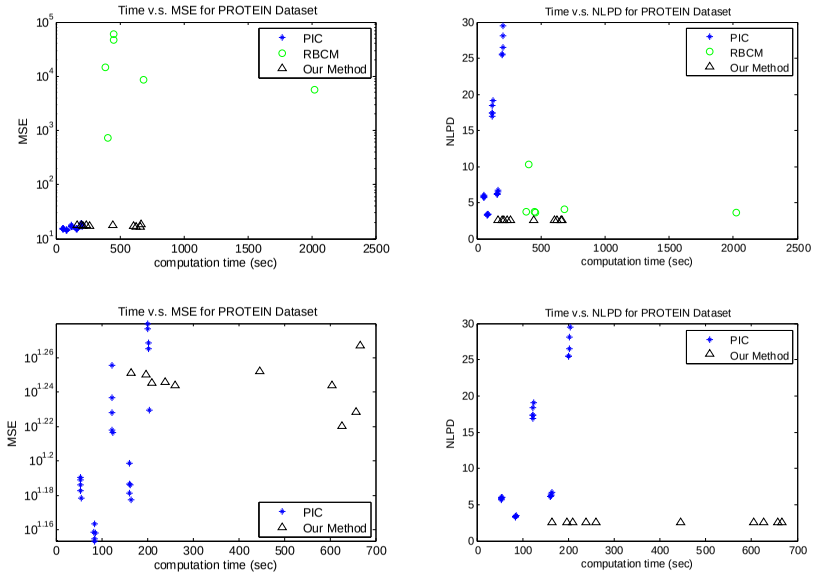


Figure 12: Prediction accuracy versus total computation time for the `PROTEIN` data. The upper panel compares all three methods. Since the performance of the PIC and our method was very close, the bottom panel provides a closer look of the PIC and our method.

PGP and GMRF approaches do not work with input dimensions more than two, and so were not included in this comparison. For RBCM, we varied the number of local experts $K \in \{100, 150, 200, 250, 300\}$. For PIC, K was varied over $\{100, 150, 200, 250, 300\}$, and the total number of pseudo inputs (M) was also varied over $\{100, 150, 200, 250, 300\}$. In this comparison, we used a squared exponential covariance function for all three methods.

Figure 12 shows the main results. For this dataset, the PIC approach outperformed our method in terms of the MSE performance, providing more accurate mean predictions. On the other hand, our method provided better NLPD performance, which implies that the predictive distribution estimated by our method was better fit to test data than that of the PIC. Figure 13 compares the predictive distributions estimated by the two methods. In the figure, the predicted mean ± 1.5 predicted standard deviations was plotted for 100 randomly chosen test observations. The interval for the PIC was overly narrow and excluded many of the test response observations, while the interval for our method more appropriately covered the majority of data, which is reflected in the better NLPD performance for our method. The percentages of the 4,573 test observations falling within the intervals was 50.47% for the PIC and 86.53% for our method. Note that the probability of a standard

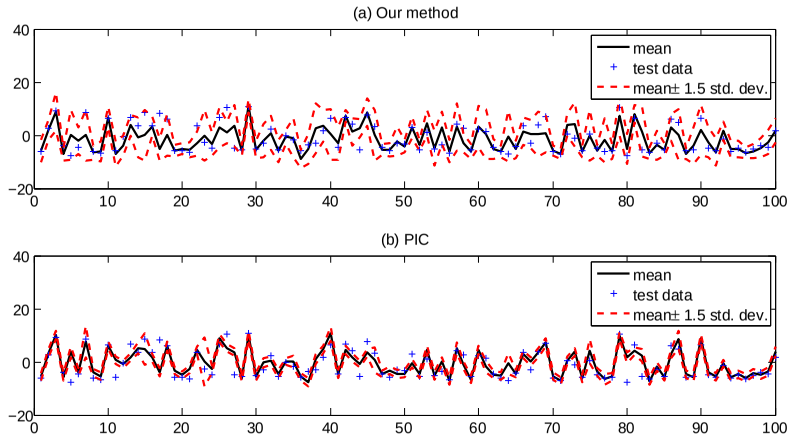


Figure 13: Comparison of the predictive distributions estimated by our method and by the PIC method for the `PROTEIN` data.

normal random variable within $\pm 1.5\sigma$ is 86.64%. Clearly, our method provides a better fit to the test data.

6.5 Example 4: `SARCOS` Dataset

This dataset has 21 input variables. For patchwork kriging, we varied $B \in \{3, 5, 7\}$, and we varied $K \in \{128, 256\}$. Again, the PGP and GMRF approaches do not work with high dimensional inputs, and so were not included in this comparison. For the RBCM approach, we varied the number of local experts $K \in \{100, 150, 200, 250, 300\}$. For PIC, K was varied over $\{100, 150, 200, 250, 300\}$, and the total number of pseudo inputs (M) was also varied over $\{100, 150, 200, 250, 300\}$. In this comparison, we used a squared exponential covariance function for all three methods.

Figure 14 summarizes the comparison of the MSEs and the NLPDs for the three methods. The MSE performances were comparable for all of the methods, while our approach provided a better fit to test data, which was evidenced by the smaller NLPD values of our approach. The PIC produced negative predictive variances for this dataset, so its NLPD values could not be calculated. In theory, the PIC approach should provide non-negative predictive variances with infinite precision. It evidently produced negative variances because of numerical errors. To be more specific, the numerical errors are mostly related to the inversion of covariance matrix of pseudo inputs. The condition number of the covariance matrix was very large, which incurred some round-off errors. Our patchwork kriging approach did not experience any such numerical issues in any of the examples, and it appears to be more numerically stable than the PIC approach.

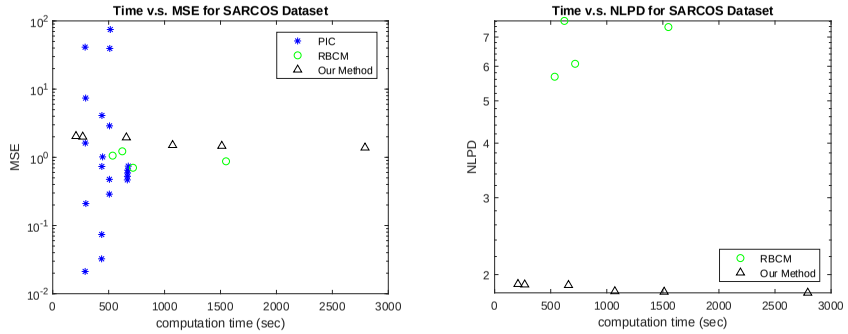


Figure 14: Prediction accuracy versus total computation time for the `SARCOS` data. A squared exponential covariance function was used. The PIC approach produced negative predictive variances, so its NLPD could not be computed. In the MSE plot, four triangles are supposed to show up. However, two of the four triangles are very closely located, so it looks like that there are only 3 triangles.

Methods	MSE	NLPD	Computation Time (seconds)
Ours	1188.4	4.8917	11729
RBCM	9790.2	10.5630	13218
PIC ($M = 1000$)	1494.8	5.0742	1624
PIC ($M = 1500$)	1492.7	5.0731	2094

Table 1: Comparison of MSE and NLPD performance for `Flight Delays` Dataset.

6.6 Example 5: `Flight Delays` Dataset

This dataset has 800,000 records and eight input variables. For this dataset, due to memory limitation of our testing environment, we could not try various cases with different choices of tuning parameters. For patchwork kriging, we fixed $K = 1024$ and $B = 5$. For the RBCM, we set the number of local experts $K = 1024$. For PIC, we set $K = 1024$, and the total number of pseudo inputs (M) was chosen to 1000 or 1500, and the further increase of M gave an out-of-memory error in our testing environment. Note that PIC requires to precompute K dense covariance matrices of size $N/K \times M$, which could be very large for this scale of N and M.

Table 1 summarizes the MSE and NLPD performance of our method, RBCM and PIC. The proposed approach gave a better MSE than the RBCM with less use of computation time. PIC showed very competitive computation performance, but its MSE was not as good as the MSE of our method. The increase of M could improve the PIC's MSE performance, but the improvement was not big in between $M = 1000$ and $M = 1500$.

6.7 Discussion

Based on the numerical comparison results for the two spatial datasets and the three higher dimensional datasets, we can summarize the benefits of the proposed approach as three-fold. First, it provides competitive or better mean prediction accuracy for both the spatial datasets and the higher dimensional datasets. As evidenced by the MSE comparisons, for the three spatial datasets, the mean predictions of the proposed approach were either more accurate than or at least comparable to those of the state-of-the-art approaches. For the first two higher dimensional datasets, its mean prediction accuracy was comparable to those of the state-of-the-art approaches. For the last dataset, it gave a better MSE performance than the state-of-the-art approaches.

Second, as evidenced by the NLPD comparisons, the predictive distribution estimated by the proposed approach provides a better fit to test data. This implies that the proposed approach provides better predictive mean and variance estimates when taken together. We designed a simple experiment to investigate this. We used the `TCO` dataset to compare the degrees of fitness of the predictive distributions from our approach and the PIC, RBCM, and PGP approaches when applied to the test data. In the comparison, the tuning parameters of each compared method were chosen so that the computation time was around 300 seconds, which resulted in approximately the best MSE value for each of the methods. Using these parameters, we calculated the predictive means $\hat{\mu}(x)$ and predictive standard deviations $\hat{\sigma}(x)$ for x in 4,833 test inputs, and we counted the fractions of 4,833 test observations that fell within $\hat{\mu}(x) \pm c\hat{\sigma}(x)$ for different c values, and the fractions were compared with ground truth $P(|X| \leq c)$ where $X \sim \mathcal{N}(0, 1)$. The fractional numbers closer to the ground truth are better. Table 2 shows the fractional numbers for different c values. The fractions for our method were very close to the ground truth for all choices c. The PGP method has much higher fractions than the ground truth, which implies that the PGP tends to overestimate the predictive standard deviation. Both the PIC and the RBCM methods have much lower fractional numbers than the ground truth, which implies that these two local methods significantly underestimate the predictive standard deviation.

Last but not least, the proposed patchwork kriging advances the PGP method by broadening the applicability to higher dimensional datasets, while the PGP method is practically limited to spatial regression with only two input dimensions.

c	0.5	1.0	1.5	2.0	2.5	3.0
$P(\|X\| \leq c)$, $X \sim \mathcal{N}(0,1)$	0.3829	0.6827	0.8664	0.9545	0.9876	0.9973
Our method	0.4471	0.7366	0.8814	0.9487	0.9766	0.9888
PGP	0.7118	0.8850	0.9549	0.9810	0.9905	0.9944
PIC	0.0126	0.0223	0.0362	0.0474	0.0604	0.0741
RBCM	0.1057	0.2090	0.3095	0.3987	0.4809	0.5535

Table 2: Percentages of test data ranging in between the estimated predictive mean $\pm c$ the estimated predictive standard deviation. The percentages were compared with $P(|X| \leq c)$ where X is a standard normal random variable. The percentage numbers closer to $P(|X| \leq c)$ are better.

7. Conclusion

We presented a new approach to efficiently solve the Gaussian process regression problem for large data. The approach first performs a spatial partitioning of a regression domain into multiple local regions and then assumes a local GP model for each local region. The local GP models are assumed *a priori* independent. However, *a posterior* dependence and related continuity constraints between the local GP models in neighboring regions are achieved by defining an auxiliary process that represents the difference between the local GP models on the boundary of the neighboring regions. By defining zero-valued pseudo observations of the auxiliary process and augmenting the actual data with the pseudo observations, we in essence force the two local GP models to have the same posterior predictive distributions at the collection of boundary points. The proposed idea of enforcing the local models to have the same boundary predictions via pseudo observations is entirely different from that of Park and Huang (2016), creating an entirely new framework for patching local GP models. The new approach provides significantly better prediction variance accuracy than the approach of Park and Huang (2016), while providing computation efficiency and mean prediction accuracy that are at least comparable and sometimes better. In addition, the spatial partitioning scheme proposed as a part of the new approach makes the new approach applicable for high dimensional regression settings, while the approach of Park and Huang (2016) is only applicable for one or two dimensional problems. Another advantage of the proposed approach is that its prediction accuracy does not depend strongly on the choice of tuning parameters, so one can simply fine-tune the tuning parameters to minimize the total computation time. Those advantages were numerically demonstrated with six well designed numerical experiments using six real datasets featuring different patterns and dimensions. The new approach has shown better trade-offs between total computation times and prediction accuracy than the approach of Park and Huang (2016) and other local-based approaches for GP regression. We believe that the proposed patchwork kriging approach is an attractive alternative for large-scale GP regression problems.

Acknowledgments

The authors are thankful for generous support of this work. Park was supported in part by the grants from National Science Foundation (CMMI-1334012) and Air Force Office of Scientific Research (FA9550-18-1-0144). D. Apley was supported in part by National Science Foundation (CMMI-1537641).

Appendix A. Derivation of the predictive mean and variance of the patchwork kriging

This appendix provides the detailed derivation of the predictive mean and variance in (6) and (7). From the standard GP modeling result (5), the posterior predictive distribution of $f_*^{(k)}$ given $\boldsymbol{y}$ and the pseudo observations $\boldsymbol{\delta}$ is Gaussian with mean

$$\mathbb{E}[f_*^{(k)}|\boldsymbol{y},\boldsymbol{\delta}] = [\boldsymbol{c}_{*\mathcal{D}}^{(k)}, \boldsymbol{c}_{*,\delta}^{(k)}] \begin{bmatrix} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{\delta} \end{bmatrix}$$

and variance given below. Using the partitioned matrix inversion formula

$$\begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{B}^T & \boldsymbol{D} \end{bmatrix}^{-1} = \begin{bmatrix} (\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{B}^T)^{-1} & -(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{B}^T)^{-1}\boldsymbol{B}\boldsymbol{D}^{-1} \\ -\boldsymbol{D}^{-1}\boldsymbol{B}^T(\boldsymbol{A}-\boldsymbol{B}\boldsymbol{D}^{-1}\boldsymbol{B}^T)^{-1} & (\boldsymbol{D}-\boldsymbol{B}^T\boldsymbol{A}^{-1}\boldsymbol{B})^{-1} \end{bmatrix}, \quad (17)$$

we have

$$\begin{aligned}\mathbb{E}[f_*^{(k)}|\boldsymbol{y},\boldsymbol{\delta}] =& \boldsymbol{c}_{*\mathcal{D}}^{(k)}(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{y} - \boldsymbol{c}_{*,\delta}^{(k)}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{y} \\ & - \boldsymbol{c}_{*\mathcal{D}}^{(k)}(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{\delta} + \boldsymbol{c}_{*,\delta}^{(k)}(\boldsymbol{C}_{\delta,\delta}-\boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta})^{-1}\boldsymbol{\delta}.\end{aligned}$$

In particular, for $\boldsymbol{\delta}=\boldsymbol{0}$, which is the only value for the pseudo observations that we need, we have

$$\mathbb{E}[f_*^{(k)}|\boldsymbol{y},\boldsymbol{\delta}=\boldsymbol{0}] = (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\delta}^{(k)}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{y}.$$

Similarly, the posterior predictive variance of $f_*^{(k)}$ given $\boldsymbol{y}$ and $\boldsymbol{\delta}$ is

$$\mathbb{V}\mathrm{ar}[f_*^{(k)}|\boldsymbol{y},\boldsymbol{\delta}] = c_{**} - [\boldsymbol{c}_{*\mathcal{D}}^{(k)}, \boldsymbol{c}_{*,\delta}^{(k)}] \begin{bmatrix} \boldsymbol{C}_{\mathcal{D}\mathcal{D}} & \boldsymbol{C}_{\mathcal{D},\delta} \\ \boldsymbol{C}_{\delta,\mathcal{D}} & \boldsymbol{C}_{\delta,\delta} \end{bmatrix}^{-1} \begin{bmatrix} \boldsymbol{c}_{\mathcal{D}*}^{(k)} \\ \boldsymbol{c}_{\delta*}^{(k)} \end{bmatrix}.$$

Applying the matrix inversion result (17) to this variance expression, we have

$$\begin{aligned}\mathbb{V}\mathrm{ar}[f_*^{(k)}|\boldsymbol{y},\boldsymbol{\delta}] =& c_{**} - \boldsymbol{c}_{*\mathcal{D}}^{(k)}(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{c}_{\mathcal{D}*}^{(k)} \\ & + \boldsymbol{c}_{*,\delta}^{(k)}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{c}_{\mathcal{D}*}^{(k)} \\ & + \boldsymbol{c}_{*\mathcal{D}}^{(k)}(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{c}_{\delta*}^{(k)} \\ & - \boldsymbol{c}_{*,\delta}^{(k)}(\boldsymbol{C}_{\delta,\delta}-\boldsymbol{C}_{\mathcal{D}\delta}^T\boldsymbol{C}_{\mathcal{D}\mathcal{D}}^{-1}\boldsymbol{C}_{\mathcal{D}\delta})^{-1}\boldsymbol{c}_{\delta*}^{(k)} \\ =& c_{**} - \boldsymbol{c}_{*,\delta}^{(k)}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{c}_{\delta*}^{(k)} \\ & - (\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\delta}^{(k)}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}}-\boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}(\boldsymbol{c}_{\mathcal{D}*}^{(k)} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{c}_{\delta*}^{(k)}).\end{aligned}$$

Appendix B. Equality of the posterior predictive means and variances of $f_*^{(k)}$ and $f_*^{(l)}$

Suppose that $\Gamma_{k,l} \neq \emptyset$, i.e., Ω_k and Ω_l are neighboring. This appendix shows

$$\begin{aligned} &\mathbb{E}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta} = \mathbf{0}] = \mathbb{E}[f_*^{(l)}|\boldsymbol{y}, \boldsymbol{\delta} = \mathbf{0}] \text{ for } x_* \in \mathbf{x}_{k,l}, \text{ and} \\ &\mathbb{V}\text{ar}[f_*^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}] = \mathbb{V}\text{ar}[f_*^{(l)}|\boldsymbol{y}, \boldsymbol{\delta}] \text{ for } x_* \in \mathbf{x}_{k,l}. \end{aligned} \tag{18}$$

Let $\boldsymbol{f}^{(k)}$ denote a column vector of the $f_*^{(k)}$ values for $x_* \in \mathbf{x}_{k,l}$ and $\boldsymbol{f}^{(l)}$ denote a column vector of the $f_*^{(l)}$ values for $x_* \in \mathbf{x}_{k,l}$.

$$\mathbb{E}[\boldsymbol{f}^{(k)} - \boldsymbol{f}^{(l)}|\boldsymbol{y}, \boldsymbol{\delta} = \mathbf{0}] = (\boldsymbol{V} - \boldsymbol{W}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{y}.$$

where $\boldsymbol{V}$ denote the matrix with its ith row $\boldsymbol{c}_{*\mathcal{D}}^{(k)} - \boldsymbol{c}_{*\mathcal{D}}^{(l)}$ evaluated for x_* equal to the ith entry of $\mathbf{x}_{k,l}$, and let $\boldsymbol{W}$ denote the matrix with its ith row $\boldsymbol{c}_{*\delta}^{(k)} - \boldsymbol{c}_{*\delta}^{(l)}$ evaluated for x_* equal to the ith entry of $\mathbf{x}_{k,l}$. Note that we can write $\boldsymbol{V} = \boldsymbol{\Theta}\boldsymbol{C}_{\mathcal{D}\delta}^T$ and $\boldsymbol{W} = \boldsymbol{\Theta}\boldsymbol{C}_{\delta,\delta}$ for some non-random matrix $\boldsymbol{\Theta}$. Therefore,

$$\begin{aligned} \mathbb{E}[\boldsymbol{f}^{(k)} - \boldsymbol{f}^{(l)}|\boldsymbol{y}, \boldsymbol{\delta} = \mathbf{0}] &= \boldsymbol{\Theta}(\boldsymbol{C}_{\mathcal{D}\delta}^T - \boldsymbol{C}_{\delta,\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{y} \\ &= \mathbf{0}. \end{aligned} \tag{19}$$

Similarly, we have

$$\begin{aligned} \mathbb{V}\text{ar}[\boldsymbol{f}^{(k)}|\boldsymbol{y}, \boldsymbol{\delta}] - \mathbb{V}\text{ar}[\boldsymbol{f}^{(l)}|\boldsymbol{y}, \boldsymbol{\delta}] &= \boldsymbol{\Theta}(\boldsymbol{C}_{\mathcal{D}\delta}^T - \boldsymbol{C}_{\delta,\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)(\boldsymbol{C}_{\mathcal{D}\mathcal{D}} - \boldsymbol{C}_{\mathcal{D}\delta}\boldsymbol{C}_{\delta,\delta}^{-1}\boldsymbol{C}_{\mathcal{D}\delta}^T)^{-1}\boldsymbol{Z} \\ &= \mathbf{0}, \end{aligned} \tag{20}$$

where $\boldsymbol{Z}$ denote the matrix with its jth column $\boldsymbol{c}_{\mathcal{D}*}^{(k)} + \boldsymbol{c}_{\mathcal{D}*}^{(l)}$ evaluated for x_* equal to the jth entry of $\mathbf{x}_{k,l}$.

Appendix C. Comparison of the patchwork kriging and the full GP for the simulated cases in Section 5

This appendix presents the full comparison data that are summarized in Section 5. Figures 15, 16 and 17 shows the comparison in a posterior mean prediction, and Figures 18, 19 and 20 shows the comparison in a posterior variance prediction

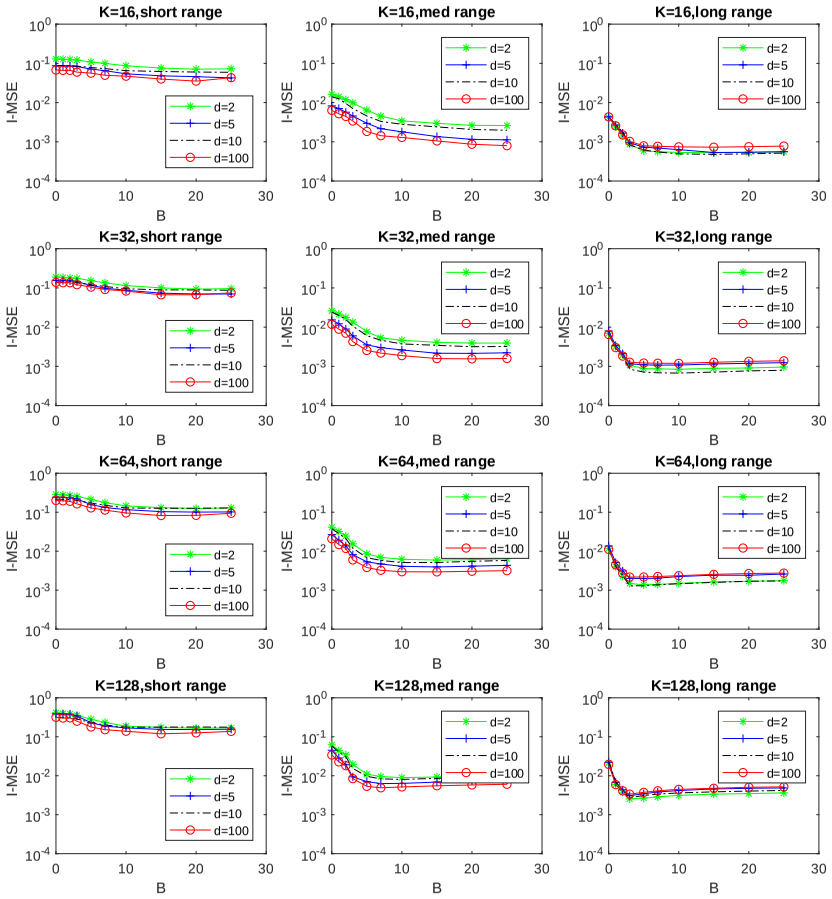


Figure 15: Summary of Interior Mean Squared Error for Predictive Mean (I-MSE)

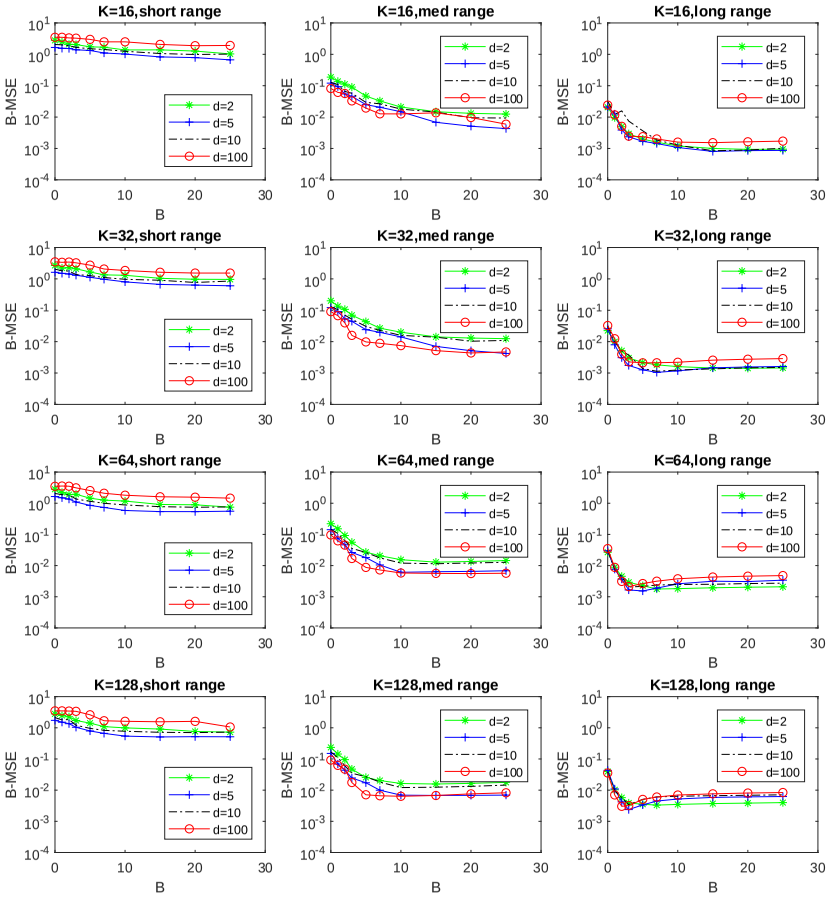


Figure 16: Summary of Boundary Mean Squared Error for Predictive Mean (B-MSE)

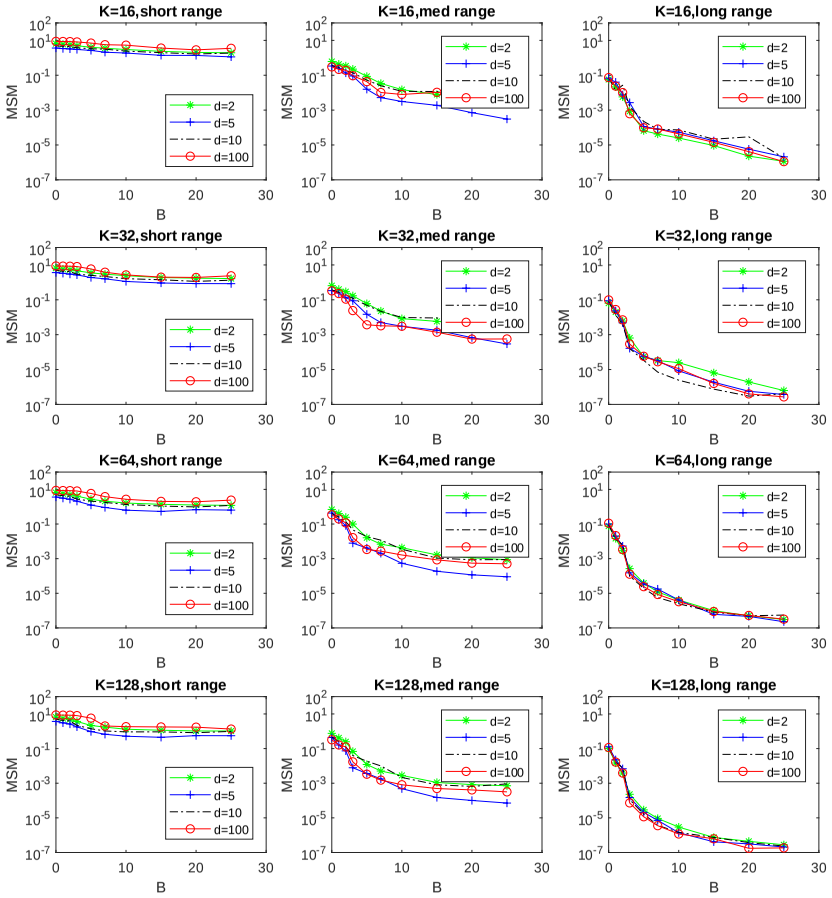


Figure 17: Summary of Mean Squared Mismatch for Predictive Mean (MSM)

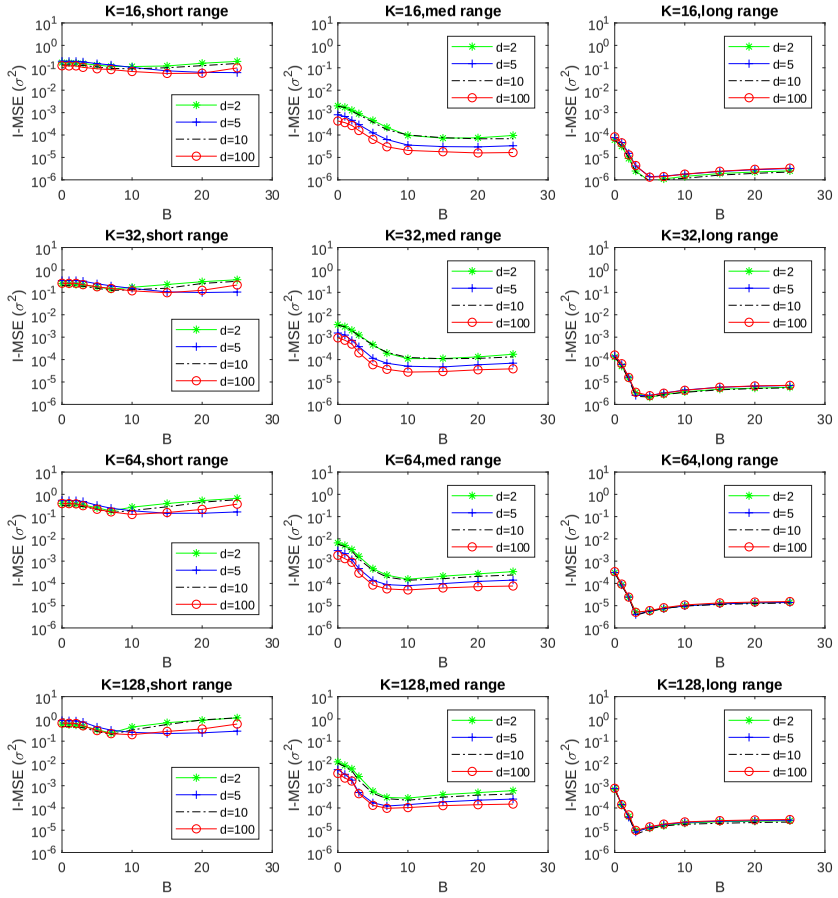


Figure 18: Summary of Interior Mean Squared Error for Predictive Variance (I-MSE(σ^2))

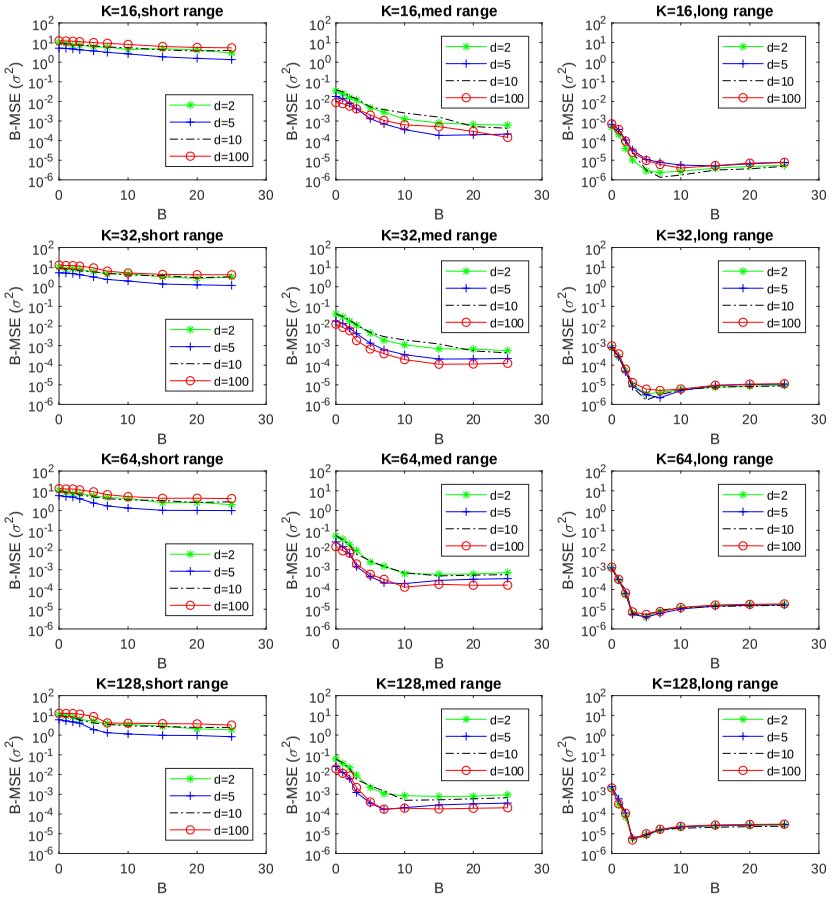


Figure 19: Summary of Boundary Mean Squared Error for Predictive Variance (B-MSE(σ^2))

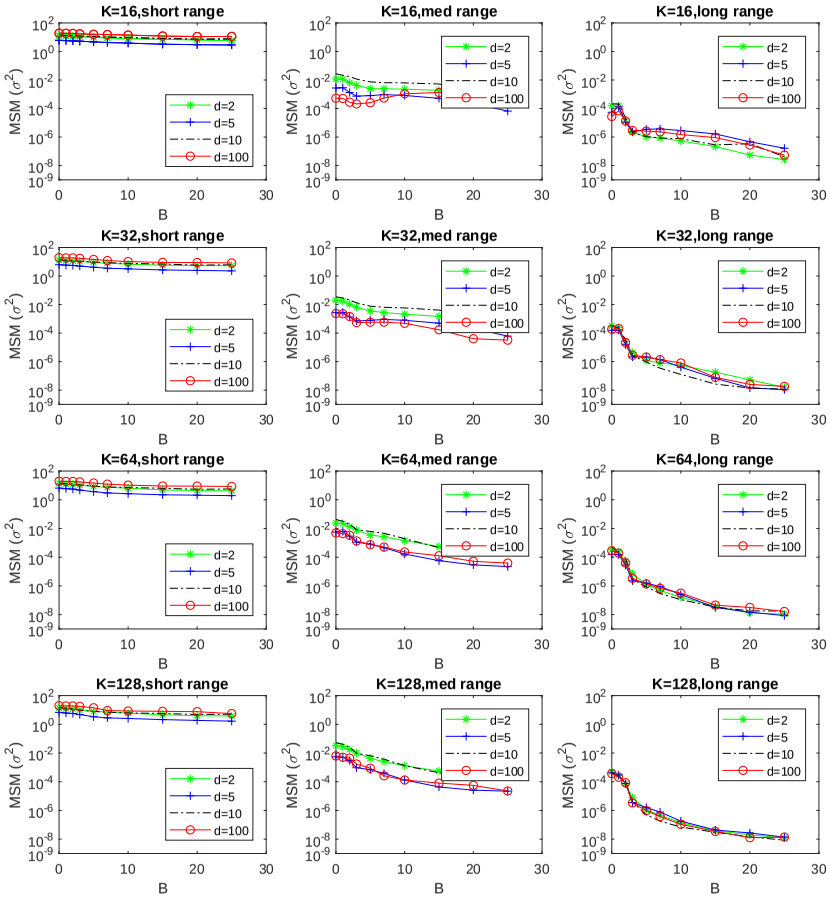


Figure 20: Summary of Mean Squared Mismatch for Predictive Variance (MSM(σ^2))

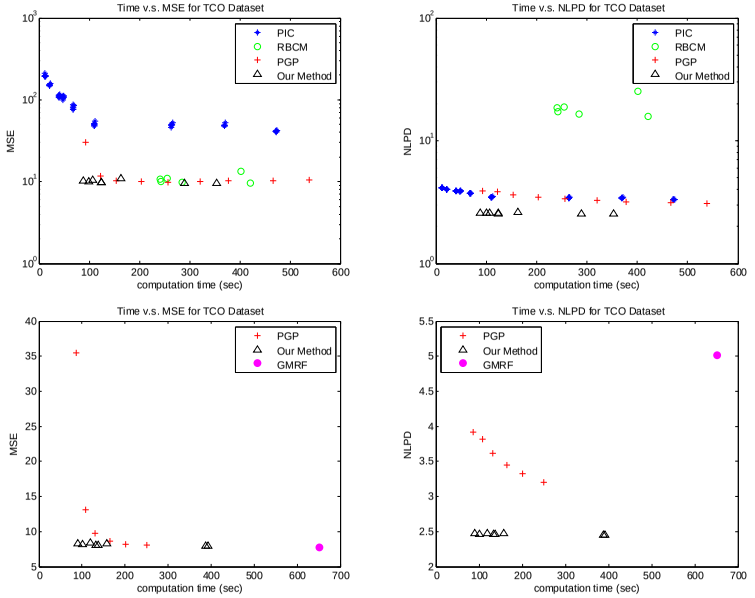


Figure 21: Prediction accuracy versus total computation time for the `TCO` data. Top panel: a squared exponential covariance function was used. Bottom panel: an exponential covariance function was used. In the top panel, eight triangles are supposed to show up. However, two of the eight triangles are very closely located, so it looks like that there are only 7 triangles.

Appendix D. Extra real data study

This appendix includes the numerical comparison with an extra real dataset, which were not in the main text. The dataset, `TCO`, contains 48,331 observations collected by the NIMBUS-7/TOMS satellite, which measures the total column of ozone over the globe on Oct 1 1988. This dataset has two input dimensions that represent a spatial location of the measurement, and the inputs of the data are densely distributed over a rectangular domain. For patchwork kriging, we varied $B \in \{5, 7\}$ and $K \in \{64, 128, 256, 512\}$. The prediction accuracy of the PGP did not depend on the number of local regions K, so we fixed $K = 145$, while the number of finite element meshes per local region was varied from 5 to 40 with step size 5. For RBCM, we varied the number of local experts $K \in \{100, 150, 200, 250, 300, 600\}$. For PIC, K was varied over $\{100, 150, 200, 250, 300, 400\}$, and the total number of pseudo inputs was also varied over $\{30, 50, 70, 80, 100, 150, 200, 250, 300\}$. For the GMRF, following the suggestion by the GMRF's authors, we used the voronoi-tessellation of training points for mesh generation.

Figure 21 summarizes the performance results. The two panels in the top row compare the proposed approach to the PGP, RBCM and PIC approaches when a square exponential covariance function is used. The MSE plot shows the mean square error of the mean predictions for the test data. The proposed approach and the PGP approach had better MSE than the RBCM and PIC approaches, and the proposed approach also had better MSE than the PGP approach at the 100 second computation time. The NLPD plot shows the degree of fitness of the estimated predictive distribution to the test data, which is affected by both of the mean prediction and the variance prediction. The proposed approach uniformly outperforms the other methods, including the PGP. The two panels in the bottom row compare the proposed approach to the PGP and the GMRF approaches when an exponential covariance function is used. The PGP and the proposed approaches had comparable MSEs at larger computation times, whereas the proposed approach had much smaller MSE at the lower computation times, and the proposed approach outperformed the PGP and the GMRF in terms of NLPD. The GMRF had longer computation time than the PGP and the proposed method.

References

S. Ambikasaran, D. Foreman-Mackey, L. Greengard, D. W. Hogg, and M. O'Neil. Fast direct methods for Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):252–265, 2016.

W. M. Chan and Alan George. A linear time implementation of the reverse Cuthill-McKee algorithm. *BIT Numerical Mathematics*, 20(1):8–14, 1980.

Jie Chen, Nannan Cao, Kian Hsiang Low, Ruofei Ouyang, Colin Keng-Yan Tan, and Patrick Jaillet. Parallel Gaussian process regression with low-rank covariance matrix approximations. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 152–161, Bellevue, WA, 2013.

Tao Chen and Jianghong Ren. Bagging for Gaussian process regression. *Neurocomputing*, 72(7):1605–1610, 2009.

K. Das and A. N. Srivastava. Block-GP: Scalable Gaussian process regression for multimodal data. In *Proceedings of the Tenth IEEE International Conference on Data Mining*, pages 791–796, Sydney, Australia, 2010.

Marc Peter Deisenroth and Jun Wei Ng. Distributed Gaussian processes. In *Proceedings of the Thirty-second International Conference on Machine Learning*, pages 1–10, Lille, France, 2015.

Reinhard Furrer, Marc G Genton, and Douglas Nychka. Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics*, 15(3), 2006.

Tilmann Gneiting. Compactly supported correlation functions. *Journal of Multivariate Analysis*, 83(2):493–508, 2002.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Robert B Gramacy and Daniel W Apley. Local Gaussian process approximation for large computer experiments. *Journal of Computational and Graphical Statistics*, 24(2):561–578, 2015.

Robert B Gramacy and Herbert KH Lee. Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association*, 103 (483), 2008.

James Hensman, Nicolò Fusi, and Neil D. Lawrence. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290, Bellevue, WA, 2013.

Cari G Kaufman, Mark J Schervish, and Douglas W Nychka. Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association*, 103(484):1545–1555, 2008.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73 (4):423–498, 2011.

Brian McFee and Gert RG Lanckriet. Large-scale music similarity search with spatial trees. In *Proceedings of 12th International Society for Music Information Retrieval Conference*, pages 55–60, Miami,Florida, 2011.

Duy Nguyen-Tuong, Jan R Peters, and Matthias Seeger. Local Gaussian process regression for real time online model learning. In *Proceedings of the Neural Information Processing Systems Conference 21*, pages 1193–1200, Vancouver, Canada, 2009.

Chiwoo Park and Jianhua Z. Huang. Efficient computation of Gaussian process regression for large spatial data sets by patching local Gaussian processes. *Journal of Machine Learning Research*, 17(174):1–29, 2016.

Chiwoo Park, Jianhua Z. Huang, and Yu Ding. Domain decomposition approach for fast Gaussian process regression of large spatial datasets. *Journal of Machine Learning Research*, 12:1697–1728, 2011.

C. E. Rasmussen and Z. Ghahramani. Infinite mixtures of Gaussian process experts. In *Proceedings of the Neural Information Processing Systems Conference 14*, pages 881–888, Granada, Spain, 2002.

C.E. Rasmussen and C.K.I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

Matthias Seeger, Christopher K. I. Williams, and Neil D. Lawrence. Fast forward selection to speed up sparse Gaussian process regression. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, Key West, Florida, 2003.

Yirong Shen, Andrew Ng, and Matthias Seeger. Fast Gaussian process regression using kd-trees. In *Proceedings of the Neural Information Processing Systems Conference 18*, pages 1225–1232, British Columbia, Canada, 2006. MIT Press.

Edward Snelson and Zoubin Ghahramani. Sparse Gaussian processes using pseudo-inputs. In *Proceedings of the Neural Information Processing Systems Conference 18*, pages 1257–1264, British Columbia, Canada, 2006.

Edward Snelson and Zoubin Ghahramani. Local and global sparse Gaussian process approximations. In *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, pages 524–531, San Juan, Puerto Rico, 2007.

Michalis K Titsias. Variational learning of inducing variables in sparse Gaussian processes. In *Proceedings of the Twelth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 567–574, Clearwater Beach, Florida, 2009.

Volker Tresp. A bayesian committee machine. *Neural Computation*, 12(11):2719–2741, 2000.

Jarno Vanhatalo and Aki Vehtari. Modelling local and global phenomena with sparse Gaussian processes. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, pages 571—-578, Helsinki, Finland, 2008.

Journal of Machine Learning Research 19 (2018) 1-35 Submitted 02/17; Revised 03/18; Published 08/18

Scalable Bayes via Barycenter in Wasserstein Space

Sanvesh Srivastava SANVESH-SRIVASTAVA@UIOWA.EDU
Department of Statistics and Actuarial Science
University of Iowa
Iowa City, Iowa 52242, USA

Cheng Li STALIC@NUS.EDU.SG
Department of Statistics and Applied Probability
National University of Singapore
Singapore 117546, Singapore

David B. Dunson DUNSON@DUKE.EDU
Departments of Statistical Science, Mathematics, and ECE
Duke University
Durham, North Carolina 27708, USA

Editor: David Blei

Abstract

Divide-and-conquer based methods for Bayesian inference provide a general approach for tractable posterior inference when the sample size is large. These methods divide the data into smaller subsets, sample from the posterior distribution of parameters in parallel on all the subsets, and combine posterior samples from all the subsets to approximate the full data posterior distribution. The smaller size of any subset compared to the full data implies that posterior sampling on any subset is computationally more efficient than sampling from the true posterior distribution. Since the combination step takes negligible time relative to sampling, posterior computations can be scaled to massive data by dividing the full data into sufficiently large number of data subsets. One such approach relies on the geometry of posterior distributions estimated across different subsets and combines them through their barycenter in a Wasserstein space of probability measures. We provide theoretical guarantees on the accuracy of approximation that are valid in many applications. We show that the geometric method approximates the full data posterior distribution better than its competitors across diverse simulations and reproduces known results when applied to a movie ratings database.

Keywords: barycenter; big data; distributed Bayesian computations; empirical measures; linear programming; optimal transportation; Wasserstein distance; Wasserstein space.

1. Introduction

Developing efficient sampling algorithms is an active area of research motivated by tractable Bayesian inference in large sample settings. Sampling remains a primary tool for inference in Bayesian models, with Markov chain Monte Carlo (MCMC) and sequential Monte Carlo (SMC) providing two broad classes of algorithms that are routinely used. Most MCMC and SMC algorithms face problems in scaling up to massive data settings due to memory and computational bottlenecks that arise; this has motivated a rich literature in recent years proposing a variety of strategies to enable better performance in such settings. Our focus is on proposing a very general divide-and-conquer technique, which is designed to combine results from any posterior sampling algorithm applied in parallel using subsets of the data.

Massive data pose major problems for existing sampling algorithms. First, if full data require multiple machines for storage, then a sampler has access to only a small fraction of the full data stored on the machine where it runs. Posterior sampling given the full data is expensive due to network latency and extensive communication among machines. Second, with sample size n, sampling in hierarchical Bayesian models requires generation of $O(n)$ latent variables, which becomes inefficient as n increases. Finally, even if full data are available to the sampler, sampling can be infeasible in practice because computation of Hessians and acceptance ratios can scale as $O(n^3)$ in some nonparametric models based on Gaussian process priors (Rasmussen and Williams, 2006). A variety of methods exist to address these issues using optimization and sampling.

Optimization-based methods for Bayesian inference obtain an analytic approximation of the full data posterior distribution. The two most common techniques are polynomial approximation (Rue et al., 2009) and projection of the full data posterior distribution on a class of distributions with analytically tractable posterior densities, which includes variational Bayes and expectation propagation (Wainwright and Jordan, 2008; Gelman et al., 2014). Both techniques estimate parameters of the approximate distribution using a variety of optimization algorithms (Tan and Nott, 2013; Kucukelbir et al., 2015; Rezende and Mohamed, 2015; Ranganath et al., 2016). Stochastic approximation significantly improves the efficiency of estimation by accessing the data in small batches and updating the parameter estimates sequentially (Broderick et al., 2013; Hoffman et al., 2013); however, optimization can be nontrivial for complex likelihoods frequently used in hierarchical models. Furthermore, variational Bayes and expectation propagation often have excellent predictive performance but can be highly biased in estimation of posterior uncertainty and dependence (Giordano et al., 2017).

There is extensive work in sampling-based methods for Bayesian inference. The three main techniques used are as follows. First, subsampling-based methods obtain posterior samples conditioned on a small fraction of the data (Maclaurin and Adams, 2015). Coupling of subsampling with modified Hamiltonian or Langevin dynamics improves posterior exploration and convergence to the stationary distribution (Welling and Teh, 2011; Ahn et al., 2012; Chen et al., 2014; Korattikara et al., 2014; Lan et al., 2014; Shahbaba et al., 2014); see Bardenet et al. (2017) for a review. Second, the exact transition kernel in posterior sampling is replaced by an approximation that significantly reduces the time required to finish an iteration of the sampler (Johndrow et al., 2015; Alquier et al., 2016). Finally, divide-and-conquer approaches first divide the data into smaller subsets and sample in parallel across subsets, and then combine the posterior samples from all the subsets. Our focus is on scalable Bayesian methods based on the divide-and-conquer technique. These methods have two subgroups that differ mainly in their sampling scheme for every subset and their method for combining posterior samples obtained from all the subsets.

The first subgroup modifies the prior to sample from the posterior distribution of the parameter conditioned on a data subset. Let k be the number of subsets, $\pi(\theta)$ be the prior density of parameter θ, and $l_i(\theta)$ be the likelihood for subset i $(i = 1, \ldots, k)$. Samples from subset posterior distribution i are obtained using $l_i(\theta)$ and $\pi(\theta)^{1/k}$ as the likelihood

and prior. Consensus Monte Carlo combines subset posterior samples by averaging, which has been generalized in many ways (Rabinovich et al., 2015; Scott et al., 2016). This relies heavily on the normality assumption, which is relaxed using a combination based on kernel density estimation (Neiswanger et al., 2014). Both methods perform poorly if the supports of subset posteriors are different, which motivates the combination using the Weierstrass transform and random partition trees (Wang and Dunson, 2013; Wang et al., 2015). These methods offer simple approaches for combining samples from subset posterior distributions but have a major limitation that the sampling algorithm depends on the model parameterization.

The second subgroup modifies the subset likelihood to sample from a subset posterior distribution and combines samples from subset posterior distributions through their geometric center. These methods modify the likelihood to $l_i(\theta)^k$ and use prior $\pi(\theta)$ to sample from subset posterior distribution i ($i = 1, \ldots, k$). M-Posterior combines subset posterior distributions through their median in the Wasserstein space of order 1 (Minsker et al., 2014, 2017). The robustness of the median implies that it could ignore valuable information in some subset posterior distributions, which motivates combination through the mean in the Wasserstein space of order 2 called Wasserstein Posterior (WASP) (Srivastava et al., 2015). The WASP approach strikes a balance between the generality of sampling and the efficiency of optimization. While WASP can be applied to any data or Bayesian model, its computations are developed for independent identically distributed (*iid*) data and its theoretical properties are unknown.

Our main goal is to study the theoretical properties of WASP and apply WASP in a variety of practical problems. The *iid* assumption of WASP rules out many important practical problems, including regression and classification, where the data are independent and non-identically distributed (*inid*). We relax this assumption and our theoretical results are applicable to *inid* data. Second, we show that if the number of subsets are chosen appropriately, then the WASP achieves almost the same rate of convergence as that of the full data posterior distribution. For linear models with error distribution in the location-scale family, we strengthen this result and show that the WASP and the full data posterior distribution have the same asymptotic mean and asymptotic variance. This implies that WASP can be used as an efficient alternative to the full data posterior distribution in massive data settings. Third, we show that the method for estimating WASP is independent of the form of the model, which implies that WASP is very general and can be easily used for estimating posterior summaries for any function of the model parameters. We emphasize that WASP is not a new sampling algorithm but a general approach to easily extend any existing sampling algorithms for massive data applications.

2. Preliminaries

2.1 Wasserstein Space, Wasserstein Distance, and Wasserstein Barycenter

We recall elementary properties and definitions related to the Wasserstein space of probability measures. Let (Θ, ρ) be a complete separable metric space and $\mathcal{P}(\Theta)$ be the space of

all probability measures on Θ. The Wasserstein space of order 2 is defined as

$$\mathcal{P}_2(\Theta) := \left\{ \mu \in \mathcal{P}(\Theta) : \int_\Theta \rho^2(\theta_0, \theta)\mu(d\theta) < \infty \right\}, \tag{1}$$

where $\theta_0 \in \Theta$ is arbitrary and $\mathcal{P}_2(\Theta)$ does not depend on the choice of θ_0. The space $\mathcal{P}_2(\Theta)$ is equipped with a natural distance between its elements. Let $\mu, \nu \in \mathcal{P}_2(\Theta)$ and $\Pi(\mu, \nu)$ be the set of all probability measures on $\Theta \times \Theta$ with marginals μ and ν, then the Wasserstein distance of order 2 between μ and ν is defined as

$$W_2(\mu, \nu) = \left(\inf_{\pi \in \Pi(\mu,\nu)} \int_{\Theta\times\Theta} \rho^2(x, y)\, d\pi(x, y) \right)^{\frac{1}{2}}. \tag{2}$$

In our applications ρ is the Euclidean metric and we refer to $\mathcal{P}_2(\Theta)$ and W_2 as the Wasserstein space and the Wasserstein distance without explicitly mentioning their order. If $\Pi_1, \ldots, \Pi_k$ are a collection of probability measures in $\mathcal{P}_2(\Theta)$, then their barycenter in $\mathcal{P}_2(\Theta)$ is defined as

$$\overline{\Pi} = \operatorname*{argmin}_{\Pi \in \mathcal{P}_2(\Theta)} \sum_{j=1}^{k} \frac{1}{k} W_2^2(\Pi, \Pi_j). \tag{3}$$

This generalizes the Euclidean barycenter, which is the sample mean, to $\mathcal{P}_2(\Theta)$ (Agueh and Carlier, 2011). The barycenter $\overline{\Pi}$ is analytically intractable, except in few special cases. Let $\delta_a(x) = 1$ if $a = x$ and 0 otherwise. If $X_{j1}, \ldots, X_{jm}$ are samples from Π_j ($j = 1, \ldots, k$), then $\widehat{\Pi}_j(\cdot) = \sum_{i=1}^m \delta_{X_{ji}}(\cdot)/m$ is an empirical measure that approximates Π_j ($j = 1, \ldots, k$). If $\overline{\Pi}$ is assumed to be an empirical measure, then the optimization problem in (3) reduces to a linear program; see Cuturi and Doucet (2014), Carlier et al. (2015), and Srivastava et al. (2015) for different algorithms to solve this linear program.

2.2 Stochastic Approximation and Subset Posterior Density

Consider a general set-up for *inid* data. Let $Y^{(n)} = (Y_1, \ldots, Y_n)$ be n observations and the distribution of Y_i is $P_{\theta,i}$, $i = 1, \ldots, n$, where θ lies in the parameter space $\Theta \subset \mathbb{R}^p$. Assume that $P_{\theta,i}$ has density $p_i(\cdot|\theta)$ with respect to the Lebesgue measure, so $dP_{\theta,i}(y_i) = p_i(y_i|\theta)dy_i$ and the likelihood given $Y^{(n)}$ is $l(\theta) = \prod_{i=1}^n p_i(y_i|\theta)$. Given a prior distribution Π on Θ that has density π with respect to the Lebesgue measure, the posterior density of θ given $Y^{(n)}$ using Bayes theorem is

$$\pi(\theta \mid Y^{(n)}) = \frac{\prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)}{\int_\Theta \prod_{i=1}^n p_i(y_i \mid \theta)\pi(\theta)d\theta} = \frac{l(\theta)\pi(\theta)}{\int_\Theta l(\theta)\pi(\theta)d\theta}. \tag{4}$$

In most cases $\pi(\theta \mid Y^{(n)})$ is analytically intractable, and accurate approximations of $\pi(\theta \mid Y^{(n)})$ are obtained using Monte Carlo methods, such as importance sampling and MCMC, and deterministic approximations, such as Laplace's method and variational Bayes. For example, in the context of logistic regression, $P_{\theta,i}$ is the Bernoulli distribution with mean $1/\left\{1 + \exp(-x_i^T\theta)\right\}$, where x_i^T is the ith row of the design matrix $X \in \mathbb{R}^{n\times p}$ and $\Theta = \mathbb{R}^p$. The posterior density of θ is analytically intractable, and it is typical to rely on Gibbs

samplers based on data augmentation (Bishop, 2006). These samplers introduce latent variables $\{z_i, i = 1, \ldots, n\}$ and alternately sample the latent variables and the parameters from their full conditional distributions. Related algorithms are very common and are computationally prohibitive for large n because they require repeated passes through the whole data.

Divide-and-conquer-type methods resolve this problem by partitioning the data into smaller subsets. Let k be the number of subsets. The default strategy is to randomly allocate samples to subsets. Let $Y_{[j]} \equiv Y_j^{(m_j)} = (Y_{j1}, \ldots, Y_{jm_j})$ denote data on the jth subset, where m_j is the size of the jth subset and $\sum_{j=1}^{k} m_j = n$. We assume that $m_j = m$ $(j = 1, \ldots, k)$ for ease of presentation, so $n = km$, the likelihood given $Y_{[j]}$ is $l_j(\theta) = \prod_{i=1}^{m} p_{ji}(y_{ji}|\theta)$, and $l(\theta)$ in (4) equals $\prod_{j=1}^{k} l_j(\theta)$. Define subset posterior density j given $Y_{[j]}$ as

$$\pi_m(\theta \mid Y_{[j]}) = \frac{\{\prod_{i=1}^{m} p_{ji}(y_{ji}|\theta)\}^{\gamma}\pi(\theta)}{\int_{\Theta}\{\prod_{i=1}^{m} p_{ji}(y_{ji}|\theta)\}^{\gamma}\pi(\theta)d\theta} = \frac{l_j(\theta)^{\gamma}\pi(\theta)}{\int_{\Theta} l_j(\theta)^{\gamma}\pi(\theta)d\theta}, \tag{5}$$

where γ is a positive real number such that $g_1\gamma m \leq n \leq g_2\gamma m$ for some $g_1, g_2 > 0$. In the present context, we assume that $\gamma = k$ with $g_1 = g_2 = 1$ following Minsker et al. (2014); more general conditions on γ are defined later in Section 3.2. This modified form of subset posterior compensates for the fact that jth subset has access to only (m/n)-fraction of the full data and ensures that $\pi_m(\theta \mid Y_{[j]})$ and $\pi_n(\theta \mid Y^{(n)})$ in (4) have variances of the same order. Minsker et al. (2014) refer to this as *stochastic approximation* because raising $l_j(\theta)$ $(j = 1, \ldots, k)$ to the power γ is equivalent to replicating every X_{ji} $(i = 1, \ldots, m)$ γ-times so that $\pi_m(\theta \mid Y_{[j]})$ $(j = 1, \ldots, k)$ are noisy approximations of $\pi(\theta \mid Y^{(n)})$.

One advantage of using stochastic approximation to define $\pi_m(\theta \mid Y_{[j]})$ in (5) is that off-the-shelf sampling algorithms can be used directly even when the prior density is the form of a discrete mixture. Consider a simple example of univariate density estimation using Dirichlet process (DP) mixtures of Gaussians. Let X_i $(i = 1, \ldots, n)$ be *iid* samples from a distribution P_0 with density p_0. The data are randomly split into k subsets of equal size m. The truncated stick-breaking representation of DP implies that the prior distribution Π on $\mathcal{P}$ has a finite mixture representation, where $\mathcal{P}$ is the set of probability distributions that have a density. We show in the Appendix that modification of the likelihood using stochastic approximation leads to nearly identical subset and full data posterior computations.

Stochastic approximation does not add any extra burden to the computations required for sampling from the subset posterior distribution of θ conditioned on m observations. We raise the likelihood in every subset to the power γ. This is equivalent to replicating observations γ-times, which seems to offset the benefits of partitioning. However, the replication of observation is not required in implementation of the sampler; we simply modify the likelihood in the full data sampler by raising it to the power γ. For example, stochastic approximation is easily implemented using the `increment_log_prob` function in Stan (Stan Development Team, 2014). We provide more examples for a variety of models in Section 4.

A simple logistic regression example demonstrates that $\pi_m(\theta \mid Y_{[j]})$ in (5) is a noisy approximation of $\pi(\theta \mid Y^{(n)})$ in (4). We simulated data for logistic regression with $n = 10^5$, $p = 2$, $\theta = (-1, 1)^T$, and entries of X randomly set to ± 1 (Figure 1). We set $\gamma = k = 40$ and obtained samples of θ from $\pi(\theta \mid Y^{(n)})$ and from $\pi_m(\theta \mid Y_{[j]})$ $(j = 1, \ldots, k)$ using the Stan's HMC sampling algorithm. The contours for the subset and full data posterior densities

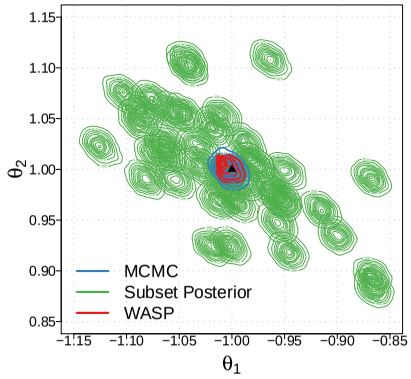


Figure 1: Binned kernel density estimates of full data posterior distribution, subset posterior distributions, and WASP for coefficients (θ_1, θ_2) in logistic regression. The x and y axes represent posterior samples for θ_1 and θ_2. The true values of θ_1 and θ_2 are -1 and 1 (black triangle).

are very similar, indicating all densities have similar spreads. We also notice that subset posteriors are noisy approximations of the full data posterior in that most of them have a bias and do not concentrate at the true θ.

3. Wasserstein Posterior (WASP): The General Framework

3.1 Definition and Estimation of the WASP

The WASP approach combines subset posterior distributions $\Pi_m(\cdot \mid Y_{[j]})$ $(j = 1, \ldots, k)$ through their barycenter in $\mathcal{P}_2(\Theta)$, where the density of $\Pi_m(\cdot \mid Y_{[j]})$ is $\pi_m(\cdot \mid Y_{[j]})$ in (5). The barycenter represents a geometric center of a collection of probability distributions that can be efficiently computed using a linear program. Motivated by this, Srivastava et al. (2015) proposed to combine a collection of subset posterior distributions through their barycenter in the Wasserstein space called *WASP*. Assuming that subset posterior distributions $\Pi_m(\cdot \mid Y_{[j]})$ $(j = 1, \ldots, k)$ have finite second moments, the WASP is defined using (3) as

$$\overline{\Pi}_n(\cdot \mid Y^{(n)}) = \underset{\Pi \in \mathcal{P}_2(\Theta)}{\operatorname{argmin}} \sum_{j=1}^{k} \frac{1}{k} W_2^2\{\Pi, \Pi_m(\cdot \mid Y_{[j]})\}. \tag{6}$$

Consider the following Gaussian example where the WASP is analytically tractable. Assume that the subset posterior distributions, $\Pi_1, \ldots, \Pi_k$, are Gaussian with means $\mu_1, \ldots, \mu_k$ and covariance matrices $\Sigma_1, \ldots, \Sigma_k$. If we fix ρ to be the Euclidean metric and $\Theta = \mathbb{R}^d$ in

Algorithm 1 Estimation of the WASP for $f(\theta)$ given samples of θ from k subset posteriors

Input: Samples from k subset posteriors, $\{\theta_{ji} : \theta_{ji} \sim \Pi_m(\cdot \mid Y_{[j]}),\ i = 1, \ldots, s_j,\ j = 1, \ldots, k\}$; mesh size $\epsilon > 0$.

Do:

1. Define $\phi_i^j = (\phi_{i1}^j, \ldots, \phi_{iq}^j) = f(\theta_{ji})$ $(i = 1, \ldots, s_j;\ j = 1, \ldots, k)$, the matrix of atoms of subset posterior j, $\Phi_j \in \mathbb{R}^{s_j \times q}$, with ϕ_i^j as row i $(i = 1, \ldots, s_j)$. For $r = 1, \ldots, q$, let $\phi_{\min} = (\phi_{\min\ 1}, \ldots, \phi_{\min\ q})$ with $\phi_{\min\ r} = \min_{j\ i} \phi_{ir}^j$, and $\phi_{\max} = (\phi_{\max\ 1}, \ldots, \phi_{\max\ q})$ with $\phi_{\max\ r} = \max_{j\ i} \phi_{ir}^j$.
2. Set the number of atoms in the empirical approximation for the WASP $g = g_1 \times \ldots \times g_q$, where $g_r = \left\lceil \frac{\phi_{\max\ r} - \phi_{\min\ r}}{\epsilon} \right\rceil$ $(r = 1, \ldots, q)$.
3. Define the matrix of WASP atoms $\overline{\Phi} \in \mathbb{R}^{g \times q}$ with rows formed by stacking vectors
$$\left\{ \phi_{\min\ 1} + \frac{i_1}{g_1} (\phi_{\max\ 1} - \phi_{\min\ 1}), \ldots, \phi_{\min\ q} + \frac{i_q}{g_q} (\phi_{\max\ q} - \phi_{\min\ q}) \right\}, \quad (i_r = 1, \ldots, g_r;\ r = 1, \ldots, q).$$
4. Set the distance matrix between the atoms of WASP and the jth subset posterior, $D_j \in \mathbb{R}_+^{g \times s_j}$, as
$$(D_j)_{uv} = \sum_{r=1}^{q} (\overline{\phi}_{ur} - \phi_{vr}^j)^2, \quad (u = 1, \ldots, g;\ v = 1, \ldots, s_j;\ j = 1, \ldots, k),$$
where $\overline{\phi}_{ur}$ is the (u, r)-entry of $\overline{\Phi}$.
5. Estimate $\hat{a}_1, \ldots, \hat{a}_g$ by solving the linear program (42) in Appendix C.

Return: $\widehat{f \sharp \overline{\Pi}}(\cdot \mid Y^{(n)}) = \sum_{i=1}^{g} \hat{a}_i \delta_{\overline{\phi}_i}(\cdot)$, the atomic approximation of $\overline{f \sharp \Pi}_n(\cdot \mid Y^{(n)})$.

(2), then (3) implies that $\overline{\Pi}_n$ is Gaussian with mean $\overline{\mu}$ and covariance matrix $\overline{\Sigma}$, where

$$\overline{\mu} = \frac{1}{k} \sum_{j=1}^{k} \mu_j \text{ and } \overline{\Sigma} \text{ is such that } \frac{1}{k} \sum_{j=1}^{k} \left(\overline{\Sigma}^{1/2} \Sigma_j \overline{\Sigma}^{1/2} \right)^{1/2} = \overline{\Sigma}, \tag{7}$$

where $A^{1/2}$ is the symmetric square root of A (Agueh and Carlier, 2011). If θ is one dimensional, then (7) says that the standard deviation of WASP is the average of standard deviations of subset posteriors; therefore, the variance of WASP is typically about the same order as that of any subset posterior distribution. A similar relation also holds in higher dimensions and for a large class of posterior distributions, including elliptical distributions (Álvarez-Esteban et al., 2016).

The WASP is analytically tractable only in special cases, but it can be estimated using a linear program if the subset posterior distributions have an atomic form. Let $\{\theta_{j1}, \ldots, \theta_{jS}\}$ be the θ samples obtained from subset posterior density j in (6) using a sampling algorithm, including HMC, MCMC, SMC, or importance sampling. Approximate jth subset posterior distribution $\Pi_m(\cdot \mid Y_{[j]})$ using the empirical measure

$$\hat{\Pi}_m(\cdot \mid Y_{[j]}) = \sum_{i=1}^{S} \frac{1}{S} \delta_{\theta_{ji}}(\cdot) \quad (j = 1, \ldots, k). \tag{8}$$

Srivastava et al. (2015) approximate the WASP as

$$\hat{\overline{\Pi}}_n(\cdot \mid Y^{(n)}) = \sum_{j=1}^{k} \sum_{i=1}^{S} a_{ji} \delta_{\theta_{ji}}(\cdot), \quad 0 \le a_{ji} \le 1, \quad \sum_{j=1}^{k} \sum_{i=1}^{S} a_{ji} = 1, \tag{9}$$

where a_{ji} $(j = 1, \ldots, k;\ i = 1, \ldots, S)$ are unknown weights of the atoms. There are many specialized algorithms to estimate the WASP that exploit the structure of the linear program in (6) when $\Pi_m(\cdot \mid Y_{[j]})$ and $\overline{\Pi}_n(\cdot \mid Y^{(n)})$ are restricted to have atomic forms in (8) and (9), respectively; for example, Cuturi and Doucet (2014) extend the Sinkhorn algorithm using entropy-smoothed sub-gradient methods, Carlier et al. (2015) develop a non-smooth optimization algorithm, and Srivastava et al. (2015) propose an efficient linear program that exploits the sparsity of constraints to solve (6). A simple and efficient algorithm to find the WASP of a given function of parameters is summarized in Algorithm 1.

3.2 Theoretical Properties of the WASP

The WASP, denoted as $\overline{\Pi}_n$, replaces the full data posterior distribution, denoted as Π_n, for inference and prediction in massive data applications where n is large. In motivating applications, computation of Π_n is inefficient, and dividing the data into smaller subsets and performing posterior computations in parallel leads to massive speed-ups. A formal asymptotic justification for using $\overline{\Pi}_n$ to approximate Π_n would ideally show that the distance between $\overline{\Pi}_n$ and Π_n tends to 0 as the full data size n increases to infinity. We will illustrate this using a linear model example in Section 3.2.1, where we show that $n^{1/2} W_2(\overline{\Pi}_n, \Pi_n) \to 0$ as $n \to \infty$. Since both $\overline{\Pi}_n$ and Π_n have variances of order n^{-1}, our result implies that the mean and the variance of WASP match those of the full data posterior distribution.

A general theoretical justification for using $\overline{\Pi}_n$ in the place of Π_n for a multivariate θ given *inid* data is technically much more challenging. If the data are *iid* and θ is one-dimensional, then Li et al. (2017) proves that $n^{1/2} W_2(\overline{\Pi}_n, \Pi_n) \to 0$ as $n \to \infty$ for regular parametric models. The proof in Li et al. (2017) relies heavily on the Bernstein-von Mises theorem (BvM) for *iid* data and the one-dimensional quantile representation of Wasserstein distance. Unlike the *iid* case, a BvM-type theorem is generally unavailable if the data are *inid* or the model is non-regular (Ibragimov and Has' Minskii, 2013). In Section 3.2.2, we show that the WASP $\overline{\Pi}_n$ converges to the true parameter value at almost the same rate as Π_n when the number of subset k increases slowly with n. The previous theoretical justification of WASP in Srivastava et al. (2015) only includes posterior consistency under the stronger *iid* assumption without characterizing the convergence rate. Relaxing these limitations, we provide the convergence rate for the WASP in the *inid* case, including the convergence rate for WASP of general functionals of the original parameters.

3.2.1 Approximation Error of WASP for inid Data: Weighted Linear Model Example

We use a weighted linear model example to illustrate the theoretical approximation accuracy of WASP to the true posterior under the *inid* setup. For $i = 1, \ldots, n$, let y_i be a scalar response, x_i be a $p \times 1$ vector of predictors, and ϵ_i be the idiosyncratic error in y_i. Let $\theta = (\theta_1, \ldots, \theta_p)^T$ be the $p \times 1$ regression coefficients vector. Let $y = (y_1, \ldots, y_n)^T$, $X = [x_1, \ldots, x_n]^T$, and $\epsilon = (\epsilon_1, \ldots, \epsilon_n)^T$ be the $n \times 1$ response vector, the $n \times p$ design matrix, and the $n \times 1$ error vector, respectively. If Σ is a known diagonal matrix with positive elements and $\text{cov}(\epsilon) = \Sigma$, then the weighted linear regression model of y on X with a flat prior on θ assumes that

$$y = X\theta + \epsilon, \quad \epsilon \sim N_n(0, \Sigma), \quad \Sigma = \text{diag}(\sigma_1^2, \ldots, \sigma_n^2), \quad \pi(\theta) \propto 1, \tag{10}$$

where $\pi(\theta)$ is the flat prior on θ and $N_n(0,\Sigma)$ is a n-variate Gaussian distribution with $n\times 1$ mean 0 and covariance Σ. In this case, the data are *inid* since the distribution of y_i depends on the value of x_i. Since Σ is assumed to be known, the posterior distribution of θ is normal with mean $\mu=(X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}y$ and covariance matrix $V=(X^T\Sigma^{-1}X)^{-1}$. Although the posterior of θ has a closed form in this example, the computational complexity of finding μ and V is $O(n^2)$, which becomes inefficient as the size of the data n increases.

The WASP of θ in (10) is analytically tractable. The computation of WASP has three steps. First, the training data are randomly split into k subsets. Let y_j, X_j, and Σ_j be the response vector, design matrix, and error covariance matrix specific to subset j ($j=1,\ldots,k$). Second, we compute the subset posterior distributions after stochastic approximation on the k subsets in parallel as in (5) with $\gamma=k$. The jth subset posterior distribution of θ is $N_p(\mu_j,V_j)$, where $\mu_j=(X_j^T\Sigma_j^{-1}X_j)^{-1}X_j^T\Sigma_j^{-1}y_j$ and $V_j=k^{-1}(X_j^T\Sigma_j^{-1}X_j)^{-1}$. Third, (7) implies that the WASP of θ is also Gaussian with mean vector $\overline{\mu}$ and covariance matrix $\overline{V}$, where $\overline{\mu}=k^{-1}\sum_{j=1}^k\mu_j$ and $\overline{V}$ satisfies $\overline{V}=k^{-1}\sum_{j=1}^k(\overline{V}^{1/2}V_j\overline{V}^{1/2})^{1/2}$.

The WASP and full data posterior distributions lead to the same posterior inference on θ up to $o(n^{-1})$ terms. Let $\overline{\Pi}_n=N_p(\overline{\mu},\overline{V})$ and $\Pi_n=N_p(\mu,V)$ be the WASP and full data posterior distributions for θ. Based on the divide-and-conquer technique, the computational complexity of $\overline{\Pi}_n$ is $O(km^2)$, which is smaller than that of Π_n by a factor of k. The true distribution of y, denoted as $P_{\theta_0}^{(n)}$, in (10) is $N_n(X\theta_0,\Sigma)$. If uncertainty quantification using $\overline{\Pi}_n$ and Π_n is the same, then it suffices to show that the difference in the second moments of $\overline{\Pi}_n$ and Π_n is $o(n^{-1})$ in $P_{\theta_0}^{(n)}$-probability because the variances $\overline{V}$ and V are both of order n^{-1}. This is equivalent to showing that the W_2 distance between $\overline{\Pi}_n$ and Π_n is $o(n^{-1})$ in $P_{\theta_0}^{(n)}$-probability, which is proved in the next theorem. In the statement of the theorem, we denote $A\prec B$ for positive definite matrices A and B if $B-A$ is also positive definite.

Theorem 1 *Assume that there exist $a_n=o(1)$, $b_m=o(1)$ such that $\Omega_0-a_nI_p\prec\frac{1}{n}X^T\Sigma^{-1}X\prec\Omega_0+a_nI_p$ and $\Omega_0-b_mI_p\prec\frac{1}{m}X_j^T\Sigma_j^{-1}X_j\prec\Omega_0+b_mI_p$ for all $j=1,\ldots,k$, where I_p, Ω_0 are $p\times p$ identity and constant positive definite matrices. Then,*

$$E_{P_{\theta_0}^{(n)}}\|\overline{\mu}-\mu\|_2^2=o\left(n^{-1}\right),\quad \mathrm{tr}\left(\overline{V}-V\right)=o\left(n^{-1}\right),\quad E_{P_{\theta_0}^{(n)}}W_2^2(\overline{\Pi}_n,\Pi_n)=o\left(n^{-1}\right).$$

The proof of this theorem is in the appendix along with other proofs.

Theorem 1 shows that the uncertainty quantification of Π_n and $\overline{\Pi}_n$ are the same in $P_{\theta_0}^{(n)}$-probability for the data following the model in (10). Essentially, the WASP and the true posterior have the same posterior mean and posterior variance, and their differences are only in high order of the full data size n. Furthermore, Theorem 1 is valid for any block diagonal Σ as long as the data that belong to a particular diagonal block of Σ also belong to the same partition. In other words, Theorem 1 even holds for dependent data in which the dependence can be expressed as a block diagonal Σ in (10). Finally, Theorem 1 is in fact true for any error distribution satisfying $E(\epsilon)=0$ and $\mathrm{cov}(\epsilon)=\Sigma$, which includes the Gaussian distribution; see Definition 2.1 and Theorem 2.3 in Álvarez-Esteban et al. (2016).

3.2.2 General Convergence Rates of the WASP for inid Data

For general non-iid data, the standard Bayesian asymptotic theory for posterior convergence rates has been established in Ghosal and van der Vaart (2007), which also includes our *inid* setup. We follow the theoretical framework of Ghosal and van der Vaart (2007) and develop the corresponding theory for divide-and-conquer Bayesian inference using the WASP.

We start with two definitions required to state the assumptions of our theoretical setup.

Definition 2 (Pseudo Hellinger distance) *The pseudo Hellinger distance between probability measures $P_{\theta_1}^{(m)},P_{\theta_2}^{(m)}\in\{\otimes_{i=1}^m P_{\theta,j,i}:\theta\in\Theta,\ dP_{\theta,j,i}(y)=p_{ji}(y\mid\theta)dy\}$ is $h_{mj}^2(\theta_1,\theta_2)=\frac{1}{m}\sum_{i=1}^m h^2\left\{p_{ji}(\cdot\mid\theta_1),p_{ji}(\cdot\mid\theta_2)\right\}$, where $h(p_1,p_2)=[\int\{\sqrt{p_1(y)}-\sqrt{p_2(y)}\}^2dy]^{1/2}$ is the Hellinger distance between two generic densities p_1,p_2.*

This definition generalizes the usual Hellinger distance to account for the *inid* data generating mechanism. The space $(\{\otimes_{i=1}^m P_{\theta,j,i}:\theta\in\Theta\},h_{mj})$ is a metric space.

Definition 3 (Generalized bracketing entropy) *Let Ξ be a fixed subset of Θ. For an m-dimensional random vector $Z=(Z_1,\ldots,Z_m)^T$, denote its L_q norm as $|Z|_q=\left[\frac{1}{m}\sum_{i=1}^m E\left(|Z_i|^q\right)\right]^{1/q}$ and use $\|Z\|$ to represent $|Z|_2$. For a fixed $j\in\{1,\ldots,k\}$, let*

$$\mathcal{P}_j(\Xi)=\left\{\boldsymbol{p}_j(\boldsymbol{y}|\theta)=(p_{j1}(y_1|\theta),\ldots,p_{jm}(y_m|\theta))^T:\boldsymbol{y}=(y_1,\ldots,y_m)^T\in\otimes_{i=1}^m\mathcal{Y}_{ji},\theta\in\Xi\right\}$$

be the class of m-dimensional functions indexed by θ. For a given $\delta>0$, let

$$\begin{aligned}\mathcal{B}\left(\delta,\mathcal{P}_j(\Xi)\right)=&\Big\{[\boldsymbol{l}_s,\boldsymbol{u}_s]:\boldsymbol{l}_s(\boldsymbol{y})=(l_{s1}(y_1),\ldots,l_{sm}(y_m))^T,\boldsymbol{u}_s(\boldsymbol{y})=(u_{s1}(y_1),\ldots,u_{sm}(y_m))^T,\\&\boldsymbol{y}=(y_1,\ldots,y_m)^T\in\otimes_{i=1}^m\mathcal{Y}_{ji},s=1,\ldots,N\Big\}\end{aligned}$$

be the generalized bracketing set of $\mathcal{P}_j(\Xi)$ with cardinality N, such that for any $\boldsymbol{p}_j(\boldsymbol{y}|\theta)\in\mathcal{P}_j(\Xi)$, there exists a pair of functions $[\boldsymbol{l}_s,\boldsymbol{u}_s]\in\mathcal{B}\left(\delta,\mathcal{P}_j(\Xi)\right)$, such that

$$\begin{aligned}&l_{si}(y_i)\le p_{ji}(y_i)\le u_{si}(y_i),\ \text{for all } \boldsymbol{y}\in\otimes_{i=1}^m\mathcal{Y}_{ji},\ \text{and all } i=1,\ldots,m\\&\text{and } \|\sqrt{\boldsymbol{u}_s}-\sqrt{\boldsymbol{l}_s}\|\le\delta.\end{aligned}$$

The h_{mj}-bracketing number of $\mathcal{P}_j(\Xi)$, $N_{[]}\left(\delta,\mathcal{P}_j(\Xi),h_{mj}\right)$, is defined as the smallest cardinality of the generalized bracketing set $\mathcal{B}\left(\delta,\mathcal{P}_j(\Xi)\right)$. The h_{mj}-bracketing entropy of $\mathcal{P}_j(\Xi)$ is defined as $H_{[]}\left(\delta,\mathcal{P}_j(\Xi),h_{mj}\right)=\log\left(1+N_{[]}\left(\delta,\mathcal{P}_j(\Xi),h_{mj}\right)\right)$.

Again, this definition generalizes the usual bracketing entropy to the *inid* cases. If the data are indeed *iid*, then Definition 3 coincides with that of the usual bracketing entropy.

Our theory for the convergence rate of WASP is built on the following assumptions.

(A1) Θ is a compact space in ρ metric, θ_0 is an interior point of Θ, and $g_1\gamma m\le n\le g_2\gamma m$ for some constants $g_1,g_2>0$.

(A2) For any $\theta,\theta'\in\Theta$ and $j=1,\ldots,m$, there exist positive constants α and C_L such that $h_{mj}^2(\theta,\theta')\ge C_L\rho^{2\alpha}(\theta,\theta')$, where h_{mj}^2 is the pseudo Hellinger distance in Definition 2.

(A3) (Entropy Condition) There exist constants $D_1 > 0$, $0 < D_2 < D_1^2/2^{12}$, a function $\Psi(u, r) \geq 0$ that is nonincreasing in $u \in \mathbb{R}^+$ and nondecreasing in $r \in \mathbb{R}^+$, such that for all $j = 1, \ldots, k$, for any $u, r > 0$ and for all sufficiently large m,

$$H_{[]}\left(u, \{\boldsymbol{p}_j(\boldsymbol{y}|\theta) : \theta \in \Theta, h_{mj}(\theta, \theta_0) \leq r\}, h_{mj}\right) \leq \Psi(u, r) \text{ for all } j = 1, \ldots, k;$$
$$\text{and } \int_{D_1 r^2/2^{12}}^{D_1 r} \sqrt{\Psi(u, r)} du < D_2 \sqrt{m} r^2,$$

where $\boldsymbol{p}_j(\boldsymbol{y}|\theta) = \{p_{j1}(y_{j1} \mid \theta), \ldots, p_{jm}(y_{jm} \mid \theta)\}^T$ and $H_{[]}$ is the h_{mj}-bracketing entropy of the set $\{\boldsymbol{p}_j(\boldsymbol{y}|\theta) : \theta \in \Theta, h_{mj}(\theta, \theta_0) \leq r\}$ in Definition 3.

(A4) (Prior Thickness) There exist positive constants κ and c_π, such that uniformly over all $j = 1, \ldots, k$,

$$\Pi\left(\theta \in \Theta : \frac{1}{m}\sum_{i=1}^{m} E_{P_{\theta_0}} \exp\left(\kappa \log_+ \frac{p_{ji}(Y_{ji}|\theta_0)}{p_{ji}(Y_{ji}|\theta)}\right) - 1 \leq \frac{\log^2 m}{m}\right) \geq \exp(-c_\pi k \log^2 m)$$

where $\log_+ x = \max(\log x, 0)$ for $x > 0$.

(A5) The metric ρ satisfies $\rho(\sum_{i=1}^N w_i \theta_i, \theta') \leq \sum_{i=1}^N w_i \rho(\theta_i, \theta')$ for any $N \in \{1, 2, \ldots\}$, $\theta_1, \ldots, \theta_N, \theta' \in \Theta$ and non-negative weights $\sum_{i=1}^N w_i = 1$.

Our assumptions above are based on the standard assumptions in Bayesian asymptotic theory. Similar to Theorem 10 in Ghosal and van der Vaart (2007), we have assumed a compact support in (A1) and lower bounded pseudo Hellinger distance in (A2). Typically, $\alpha = 1$ for most regular models, such as generalized linear models. If the model is non-regular, then α can be less than 1; for example, the densities may have discontinuities depending on the parameter (Ibragimov and Has' Minskii, 2013, Chapters V, VI). Assumption (A3) parallels the entropy condition used in Theorem 1 of Wong and Shen (1995), which has been adapted here for the *inid* setup using the generalized bracketing entropy, and will simplify to a similar entropy condition to that in Theorem 1 of Wong and Shen (1995) if the data are *iid*. Assumption (A4) is crucial in providing a stronger control over the tail probability as the posterior probability mass moves away from the true parameter θ_0, typically with an exponentially decaying rate. The convexity property of ρ in (A5) is mainly used to establish an averaging inequality under W_2 distance and is satisfied by, for example, the Euclidean metric and L_q metric with $q \geq 1$.

The posterior risks of Π_n and $\overline{\Pi}_n$ in the ρ metric is directly related to the W_2 distance based on the ρ metric. If θ_0 denotes the true parameter value from which the data are generated, then the posterior risk of Π_n in the estimation of θ_0 is

$$\int_{\mathcal{Y}^{(n)}} \int_\Theta \rho^2(\theta, \theta_0) d\Pi_n(\theta \mid Y^{(n)}) dP_{\theta_0}^{(n)}(y_1, \ldots, y_n) = E_{P_{\theta_0}^{(n)}}\left[W_2^2\left\{\Pi_n(\cdot \mid Y^{(n)}), \delta_{\theta_0}(\cdot)\right\}\right]. \quad (11)$$

The classical result says that the posterior risk (11) in regular parametric models converges to zero at the n^{-1} rate under assumptions similar to (A2)–(A4), with m replaced by n (van der Vaart, 2000). The next theorem shows that the same posterior risk of the WASP converges at a similar rate to that of the true posterior Π_n, which mainly depends on the size of subsets m, and can be made close to the standard n^{-1} rate up to some logarithmic factors for regular parametric models.

Theorem 4 *If Assumptions (A1)-(A4) hold for the jth subset posterior $\Pi_m(\cdot \mid Y_{[j]})$ $(j = 1, \ldots, k)$, then there exists a constants universal $C_1 > 0$ independent of j, such that as $m \to \infty$,*

$$E_{P_{\theta_0}^{(m)}}\left[W_2^2\left\{\Pi_m(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot)\right\}\right] \leq C_1 \left(\frac{\log^2 m}{m}\right)^{\frac{1}{\alpha}}, \quad j = 1, \ldots, k. \quad (12)$$

Additionally, if Assumption (A5) holds, then as $m \to \infty$,

$$E_{P_{\theta_0}^{(n)}}\left[W_2^2\left\{\overline{\Pi}_n(\cdot \mid Y^{(n)}), \delta_{\theta_0}(\cdot)\right\}\right] \leq C_1 \left(\frac{\log^2 m}{m}\right)^{\frac{1}{\alpha}}. \quad (13)$$

Theorem 4 proves posterior convergence in expectation, which is stronger than the commonly studied posterior convergence in probability. We present our results using the W_2 distance in order to account for the fact that the k subset posteriors sit on a common parameter space. Alternatively, from (11), the convergence rates in (12) and (13) are also the rates of posterior risks for the subset posterior distributions and the WASP. For regular models with $\alpha = 1$, if the number of subsets k increases slowly with n (e.g., $k = O(\log^c n)$ for some constant $c > 0$), then Theorem 4 implies that the WASP converges in W_2 distance at a near optimal convergence rate $O_p(n^{-1/2} \log^{c/2+1} n)$ to δ_{θ_0}. In this case, the standard parametric convergence rate of Π_n is $O_p(n^{-1/2})$, so the WASP attains the optimal convergence rate up to the $\log^{c/2+1} n$ factor. Equivalently, using (11), the posterior risk of the WASP converges to zero at the near optimal rate $O_p(n^{-1} \log^{c+2} n)$, compared to the $O_p(n^{-1})$ posterior risk of the true posterior Π_n.

In most applications, the interest also lies in functions of θ. Suppose $f : \Theta \mapsto \mathbb{R}^q$ is a function that maps θ to $\{f_1(\theta), \ldots, f_q(\theta)\}$, where $q \geq 1$ is a positive integer. A direct application of Lemma 8.5 in Bickel and Freedman (1981) gives the following corollary about the WASP of a function of θ. As long as the function is bounded almost linearly by the ρ metric in (1), its WASP possesses the same posterior convergence rate as in Theorem 4.

Corollary 5 *Suppose $f(\cdot) = \{f_1(\cdot), \ldots, f_q(\cdot)\}$ is a function that maps $\Theta \mapsto \mathbb{R}^q$ such that $|f(\theta)|^2 = \sum_{i=1}^q \{f_i(\theta)\}^2 \leq C_f\{1 + \rho^2(\theta, \theta_0)\}$, where $C_f > 0$ is a fixed constant. If the conditions in Theorem 4 hold and $\overline{f\sharp\Pi}_n(\cdot \mid Y^{(n)})$ represents the WASP of the subset posterior distributions for $f(\theta)$, then as $m \to \infty$,*

$$W_2\left\{\overline{f\sharp\Pi}_n(\cdot \mid Y^{(n)}), \delta_{f(\theta_0)}(\cdot)\right\} = O_{P_{\theta_0}^{(n)}}\left(\sqrt{\frac{\log^{2/\alpha} m}{m^{1/\alpha}}}\right).$$

Corollary 5 is very useful in applications because it says that the combination step in the WASP is independent of the model parametrization. Let $f\sharp\Pi_m(\cdot \mid Y_{[j]})$ be the jth subset posterior distribution for $f(\theta)$ $(j = 1, \ldots, k)$, then the WASP of k subset posterior distributions converges to $f(\theta_0)$ at the rate obtained in Theorem 4. In practice, we have S_j posterior samples of θ obtained from subset posterior j denoted as θ_{ji} $(i = 1, \ldots, s_j;$ $j = 1, \ldots, k)$. Algorithm 1 estimates an atomic approximation of $\overline{f\sharp\Pi}_n(\cdot \mid Y^{(n)})$, denoted as $\widehat{\overline{f\sharp\Pi}}_n(\cdot \mid Y^{(n)})$, based on the subset posterior samples $f(\theta_{ji})$ $(i = 1, \ldots, s_j;\ j = 1, \ldots, k)$.

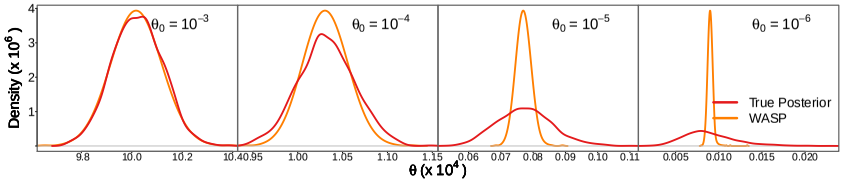


Figure 2: Kernel density estimates of the posterior densities of θ in the rare events example where assumption (A1) fails to hold for $\theta_0 = 10^{-5}, 10^{-6}$.

The atomic form of the WASP is supported on a grid with mesh-size ϵ estimated from the subset posterior samples of $f(\theta)$. Algorithm 1 estimates the weights of the atoms located on the grid by solving a discrete version of (6). The theoretical properties of discrete barycenters imply that $\widehat{f\sharp\Pi}_n(\cdot \mid Y^{(n)})$ is supported only on $O(k)$ elements of the grid; see Theorem 2 in Anderes et al. (2016). We exploit this sparsity by adapting the algorithm in Srivastava et al. (2015) and by using Gurobi (Gurobi Optimization Inc., 2014).

A key assumption in Theorem 4 and Corollary 5 is that the subset posterior distributions provide a noisy approximation of the full data posterior distribution. This is stated precisely in (12), which shows that the convergence rate of a subset posterior distribution in W_2 distance is obtained by using m as the sample size instead of n. If any of the assumptions (A1)–(A4) fail, then the subset posterior distributions may approximate the full data posterior distribution poorly, which could possibly lead to poor approximation quality for the WASP.

A simple example based on rare events demonstrates this phenomenon. Let $Y_1, \ldots, Y_n$ be *iid* Bernoulli random variables with unknown success probability $\theta \in (0, 1)$. The assumption (A1) is violated if the true parameter θ_0 is very close to 0; that is, observing 1 is a rare event. In our simulation example, we set $n = 10^7$ and $\theta_0 = 10^{-a}$ for $a = 3, 4, 5, 6$ so that as a increases, $s = \sum_{i=1}^{n} Y_i$ decreases and θ_0 gets closer and closer to the boundary of the parameter space. The standard Bayesian approach is to put Jefferys' prior Beta(0.5, 0.5) on θ and perform inference on θ using Beta($s + 0.5$, $n - s + 0.5$), which leads to a full data posterior that concentrates around the correct value of θ_0 even if θ_0 is small (Figure 2). However, if the data are randomly divided into $k = 100$ subsets, then a majority of the subsets contain only 0s as θ_0 decreases. As a result, a majority of the subset posterior distributions differ significantly in shape from the full data posterior distribution, leading to a failure of the WASP in approximating the full data posterior distribution because the assumption (A1) is severely violated for $\theta_0 = 10^{-5}, 10^{-6}$ (Figure 2).

4. Experiments

4.1 Setup

We compared WASP with consensus Monte Carlo (CMC) (Scott et al., 2016), semiparametric density product (SDP) (Neiswanger et al., 2014), and variational Bayes (VB). The sample sizes and the number of parameters in our experiments were chosen such that sampling from the full data posterior distribution was computationally feasible. Every sampling algorithm ran for 10,000 iterations. We discarded the first 5,000 samples as burn-in and thinned the chain by collecting every fifth sample. Convergence of the chains to their stationary distributions was confirmed using trace plots. All experiments ran on an Oracle Grid Engine cluster with 2.6GHz 16 core compute nodes. Full data posterior computations were allotted memory resources of 64GB, and all other methods were allotted memory resources of 16GB.

The sampling algorithm for the full data posterior was modified to obtain samples from the subset posteriors in CMC, SDP, and WASP. The sampling algorithms for subset posteriors in CMC and SDP were the same and were based on Equation (2) in Scott et al. (2016). The sampling algorithm for subset posteriors in WASP was based on (5). Samples from the approximate posterior distributions of θ in CMC, SDP, and WASP were obtained in two steps. First, samples from subset posteriors of θ were obtained in parallel across k subsets. Second, the samples of θ from all the subsets were combined using implementations of CMC and SDP in `parallelMCMC` package (Miroshnikov and Conlon, 2014) and using Algorithm 1 for the WASP.

The full data posterior distribution obtained using MCMC served as the benchmark in all our comparisons. Let $\pi(\theta \mid Y^{(n)})$ be the density of the full data posterior distribution for θ estimated using sampling and $\hat{\pi}(\theta \mid Y^{(n)})$ be the density of an approximate posterior distribution for θ estimated using the WASP or its competitors. We used the following metric based on the total variation distance to compare the accuracy $\hat{\pi}(\theta \mid Y^{(n)})$ in approximating $\pi(\theta \mid Y^{(n)})$

$$\text{accuracy}\left\{\hat{\pi}(\theta \mid Y^{(n)})\right\} = 1 - \frac{1}{2}\int_{\Theta} \left|\hat{\pi}(\theta \mid Y^{(n)}) - \pi(\theta \mid Y^{(n)})\right| \mathrm{d}\,\theta. \tag{14}$$

The accuracy metric lies in $[0, 1]$ (Faes et al., 2012). The approximation of full data posterior density by $\hat{\pi}$ is poor or excellent if the accuracy metric is close to 0 or 1, respectively. In our experiments, we computed the kernel density estimates of $\hat{\pi}$ and π from the posterior samples of θ using R package `KernSmooth` (Wand, 2015) and calculated the integral in (14) using numerical approximation.

4.2 Simulated Data: Finite Mixture of Gaussians

Finite mixture of Gaussians are widely used for model-based classification, clustering, and density estimation (Fraley and Raftery, 2002). Let n, p, and L be the sample size, the dimension of observations, and the number of mixture components. If $\mathbf{y}_i \in \mathbb{R}^p$ is the ith observation ($i = 1, \ldots, n$), then the mixture of L Gaussians assumes that any $\mathbf{y} \in \{\mathbf{y}_1, \ldots, \mathbf{y}_n\}$ is generated from the density

$$f_{\text{mix}}(\mathbf{y} \mid \theta) = \sum_{l=1}^{L} \pi_l \, \mathcal{N}_p(\mathbf{y} \mid \boldsymbol{\mu}_l, \Sigma_l), \tag{15}$$

where $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_L)$ lies in a $(L-1)$-simplex, μ_l and Σ_l ($l = 1, \ldots, L$) are the mean and covariance parameters of a p-variate Gaussian distribution, and $\theta = \{\boldsymbol{\pi}, \boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_L, \Sigma_1, \ldots, \Sigma_L\}$. We set $L = 2$ and $p = 2$ and simulated data from (15) using $\boldsymbol{\pi} = (0.3, 0.7)$, $\mu_1 = (1, 2)^T$, $\mu_2 = (7, 8)^T$, and $\Sigma_l = \Sigma$ ($l = 1, 2$), where $\Sigma_{12} = 0.5$, $\Sigma_{11} = 1$, and $\Sigma_{22} = 2$. We performed 10 simulation replications.

Table 1: Accuracies of the approximate posteriors for ρ_1, ρ_2, and $g_{0.05}(x)$ and $g_{0.95}(x)$ for $x \in \mathbb{R}$. The accuracies are averaged over 10 simulation replications. Monte Carlo errors are in parenthesis. CMC, consensus Monte Carlo; SDP, semiparametric density product; VB, variational Bayes; WASP, Wasserstein posterior

	ρ_1		ρ_2		$g_{0.05}$		$g_{0.95}$	
VB	0.77 (0.31)		0.76 (0.29)		0.99 (0.00)		0.99 (0.00)	
	$k = 5$	$k = 10$	$k = 5$	$k = 10$	$k = 5$	$k = 10$	$k = 5$	$k = 10$
CMC	0.97 (0.01)	0.96 (0.01)	0.96 (0.01)	0.96 (0.01)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)
SDP	0.97 (0.01)	0.96 (0.01)	0.95 (0.01)	0.96 (0.01)	-	-	-	-
WASP	0.97 (0.01)	0.95 (0.01)	0.97 (0.01)	0.96 (0.01)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)

This simple example demonstrated the generality of WASP in estimating the posterior distribution of functions of θ as described in Corollary 5. We defined two nonlinear functions of θ as

$$\rho_l = (\Sigma_l)_{12} / \{(\Sigma_l)_{11}(\Sigma_l)_{22}\}^{1/2} \quad l = 1, 2, \quad g(x) = f_{\text{mix}}\{(x, x)^T\} \quad x \in \mathbb{R}, \tag{16}$$

where ρ_l is the correlation of lth mixture component and $g(x)$ is the value of density f_{mix} in (15) when $\mathbf{y} = (x, x)^T$. Our simulation setup implied that $\rho_1 = \rho_2$ and $g(x)$ was bimodal for $x \in \mathbb{R}$. We completed the hierarchical model in (15) by specifying independent conjugate priors on $\boldsymbol{\pi}$ and (μ_l, Σ_l) $(l = 1, 2)$ as

$$\boldsymbol{\pi} \sim \text{Dirichlet}(1/2, 1/2), \quad \boldsymbol{\mu}_l \mid \Sigma_l \sim \mathcal{N}_2(\mathbf{0}, 100\Sigma_l), \quad \Sigma_l \sim \text{Inverse-Wishart}(2, 4I_2), \tag{17}$$

where 2 is the prior degrees of freedom and $4I_p$ is the scale matrix of the Inverse-Wishart distribution. The posterior samples of θ were obtained using Gibbs sampling (Bishop, 2006), which were used to obtain posterior samples for ρ_1, ρ_2, and g.

We compared WASP with the posterior distributions estimated using CMC, Gibbs sampling, SDP, and VB. We used the VB algorithm developed in Bishop (2006). Two values of $k \in \{5, 10\}$ were used for CMC, SDP, and WASP and full data were partitioned into k subsets such that the mixture proportions were preserved in every subset. The approximate posterior distributions of ρ_1, ρ_2, and $g(x), x \in \mathbb{R}$, under each method were estimated using the subset posterior samples obtained after modifying the original Gibbs sampler. The sampling algorithm for WASP is described in the Supplementary Material.

We compared the accuracy (14) of CMC, SDP, VB, and WASP in approximating the full data posterior distributions of ρ_1, ρ_2, and point-wise 90% credible bands of $g(x)$ for $x \in \mathbb{R}$, denoted as $g_{0.05}(x)$ and $g_{0.95}(x)$. CMC, SDP, and WASP accurately approximated the full data posterior distributions of ρ_1 and ρ_2 for both ks, but VB underestimated the posterior uncertainty in ρ_1 and ρ_2. CMC, VB, and WASP were very accurate in estimating $g_{0.05}(x)$ and $g_{0.95}(x)$ for $x \in \mathbb{R}$, whereas the application of SDP failed due to a numerical error in matrix inversion (Table 1). This provides an empirical verification of Corollary 5, showing that the accuracy of the WASP was unaffected by the form of the parameters in the combination step. Theoretical guarantees similar to Corollary 5 were unavailable for CMC or SDP, but our numerical results illustrated that a similar result might also hold for these methods in mixture models.

4.3 Simulated Data: Linear Mixed Effects Model

Linear mixed effects models are extensively used in extending linear regression to account for longitudinal and nested dependence structures. Let n, s, and s_i be the sample size, total number of observations, and total number of observations for sample i $(i = 1, \ldots, n)$ so that $s = \sum_{i=1}^{n} s_i$. Suppose $X_i \in \mathbb{R}^{s_i \times p}$ and $Z_i \in \mathbb{R}^{s_i \times r}$ include predictors in the fixed and random effects components, respectively. Letting $\mathbf{y}_i \in \mathbb{R}^{s_i}$ be the response for sample i, the linear mixed effects model assumes that

$$\mathbf{y}_i \mid \boldsymbol{\beta}, \mathbf{u}_i, \tau^2 \sim \mathcal{N}_{s_i}(X_i\boldsymbol{\beta} + Z_i\mathbf{u}_i, \tau^2 I_{n_i}), \quad \mathbf{u}_i \sim \mathcal{N}_r(\mathbf{0}, \Sigma), \quad (i = 1, \ldots, n), \tag{18}$$

where $\mathbf{u}_i \in \mathbb{R}^r$ is the random effect for sample i with mean $\mathbf{0}$ and $r \times r$ covariance Σ, $\boldsymbol{\beta} \in \mathbb{R}^p$ denotes the fixed effects, and τ^2 is the error variance. The model parameters are $\theta = \{\boldsymbol{\beta}, \Sigma, \tau^2\}$.

We simulated data for a fixed n and s and varying p and r. We chose two values of $(p, r) \in \{(4, 3), (80, 6)\}$, fixed n and s to be 6000 and 100,000, and randomly assigned the s observations to n samples. The two choices of (p, r) ensured that the number of unknown parameters in $\boldsymbol{\beta}$ and Σ was 10 and 100 in the former and latter cases. The entries of X_i and Z_i were set to 1 or -1 with equal probability for every i. We fixed $\boldsymbol{\beta}$ entries as -2 and 2 alternately and $\tau^2 = 1$. The random effects covariance matrix $\Sigma = \text{diag}(\sqrt{1}, \ldots, \sqrt{r})R\,\text{diag}(\sqrt{1}, \ldots, \sqrt{r})$, where diag($\mathbf{a}$) is a diagonal matrix with $\mathbf{a}$ along the diagonal and R is a correlation matrix with 1 along the diagonal. We set $R = R_1$ if $r = 3$ and $R = \text{bdiag}(R_1, R_1)$ if $r = 6$, where bdiag(A, B) is a block-diagonal matrix with A, B along the diagonal, $(R_1)_{ii} = 1$ $(i = 1, 2, 3)$, $R_{12} = -0.40$, $R_{13} = 0.30$, and $R_{23} = 0.001$. The matrix R_1 included negative, positive, and small to moderate strength correlations (Kim et al., 2013). We used this setup to simulate data from (18) and performed 10 replications.

We used the HMC algorithm in Stan for sampling from the full data and subset posterior distributions. The full data posterior computations were feasible for the chosen values of n and s and posterior samples were obtained after completing the hierarchical model in (18) by using the default weakly informative priors for $\boldsymbol{\beta}$, Σ, and τ^2 in Stan. Two values of $k \in \{10, 20\}$ were used for CMC, SDP, and WASP, and the n samples were randomly partitioned into k subsets. The sampling algorithms for subset posterior distributions for the three methods were implemented in Stan and posterior samples of θ were obtained in parallel across k subsets. This was followed by a combination step to estimate the approximate posterior distributions for the three methods. The sampling algorithm for WASP is described in the Supplementary Material. Stochastic gradient Langevin dynamics (SGLD; Welling and Teh 2011) has proven to be a successful stochastic version of MCMC in mixture and regression models but has not been extensively tested on linear mixed effects models in which multiple observations are available on a subject. We compared Stan's HMC and SGLD with batch sizes 2000, 4000, step sizes $10^{-4}, 10^{-5}$ and 10^4 iterations.

We compared the accuracy (14) of CMC, SDP, SGLD, VB, and WASP in approximating the marginal posterior distributions of fixed effects, variances and covariances of random effects, and the joint posterior distributions of three pairs of covariances of random effects. We used the streamlined algorithm (SA; Lee and Wand 2016) and automatic differentiation variational inference in Stan (ADVI; Kucukelbir et al. 2015) for estimating the VB posteriors for $\boldsymbol{\beta}$ and Σ . All methods except SGLD were significantly faster than the full data posterior

Table 2: Accuracies of the approximate posteriors for variances in (18). The accuracies are averaged over 10 simulation replications and across all diagonal elements of Σ. Monte Carlo errors are in parenthesis. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$r=3$		$r=6$	
ADVI	0.48 (0.31)		0.09 (0.23)	
SA	0.26 (0.19)		0.34 (0.22)	
SGLD (2000)	0.68 (0.08)		0.73 (0.08)	
SGLD (4000)	0.69 (0.09)		0.72 (0.08)	
	$k=10$	$k=20$	$k=10$	$k=20$
CMC	0.93 (0.03)	0.91 (0.05)	0.89 (0.05)	0.80 (0.08)
SDP	0.92 (0.06)	0.86 (0.07)	0.84 (0.10)	0.77 (0.14)
WASP	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)

distribution, with SA being the fastest. CMC, SA, SDP, and WASP provided accurate approximations of the marginal posterior distributions of fixed effects and covariances of random effects. Unlike Stan's HMC, SGLD's performance was sensitive to the choices of step size and batch size. SGLD failed to converge for all batch sizes when the step size was 10^{-4}, and its accuracy increased with batch size. The performance of ADVI and SGLD deteriorated quickly as r increased from 3 to 6. The accuracy of CMC and SDP in approximating the marginal posterior distributions of variances of random effects depended on k and r. ADVI and SA provided a poor approximation for the posterior variances of random effects. In all these cases, the accuracy of WASP was stable for every k and r (Tables 2 and 3). All methods except SGLD showed similar accuracies in approximating the true joint posterior distributions of three pairs of covariances of random effects. The differences in accuracies of CMC, SA, SDP, and WASP for different values of k and r were due to the differences in numerical approximation of (14) (Tables 4 and 5 and Figures 3 and 4); see Table 1 in the Supplementary Material.

The accuracy of CMC, SDP, and WASP decreased when k increased from 10 to 20 because subset posterior distributions conditioned on a smaller fraction of the data. This provided an empirical verification of Theorem 4 for the WASP. Our numerical results illustrated that a similar result might also hold for CMC and SDP. The stable performance of WASP compared to that of CMC and SDP in the approximation of the posterior distributions of variances of random effects showed that the validity of the normal approximation for subset posterior distributions was crucial in obtaining accurate approximations of full data posterior using CMC and SDP. On the other hand, WASP results were free of any such assumptions and were valid for any nonlinear function of μ and Σ; see Corollary 5.

4.4 Simulated Data: Probablistic Parafac Model

We use probabilistic parafac model as a representative example for nonparametric density estimation using WASP. Probabilistic parafac is an approach for nonparametric Bayes modeling of joint dependence in multivariate categorical data (Dunson and Xing, 2009). Let $\mathbf{x}_i = (x_{i1}, \ldots, x_{ij}, \ldots, x_{ip})$ be the data from sample i, where x_{ij} has d_j possible categorical values in $\{1, \ldots, d_j\}$ $(j = 1, \ldots, p)$. The hierarchical model for x_{ij} $(i = 1, \ldots, n;$

Table 3: Accuracies of the approximate posteriors for covariances in (18). The accuracies are averaged over 10 simulation replications and across all off-diagonal elements of Σ. Monte Carlo errors are in parenthesis. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$r=3$		$r=6$	
ADVI	0.69 (0.23)		0.49 (0.29)	
SA	0.94 (0.02)		0.94 (0.02)	
SGLD (2000)	0.07 (0.11)		0.13 (0.09)	
SGLD (4000)	0.07 (0.11)		0.12 (0.09)	
	$k=10$	$k=20$	$k=10$	$k=20$
CMC	0.94 (0.03)	0.91 (0.05)	0.94 (0.03)	0.92 (0.05)
SDP	0.92 (0.04)	0.89 (0.06)	0.89 (0.07)	0.87 (0.10)
WASP	0.97 (0.01)	0.97 (0.01)	0.97 (0.01)	0.96 (0.01)

Table 4: Accuracies of the approximate two-dimensional joint posteriors for the covariances of random effects when $r = 3$ in (18). The accuracies are averaged over 10 simulation replications. Monte Carlo errors are in parenthesis. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$(\sigma_{12}, \sigma_{13})$		$(\sigma_{12}, \sigma_{23})$		$(\sigma_{13}, \sigma_{32})$	
ADVI	0.53 (0.28)		0.62 (0.14)		0.49 (0.25)	
SA	0.91 (0.01)		0.91 (0.01)		0.92 (0.01)	
SGLD (2000)	0.03 (0.01)		0.01 (0.00)		0.02 (0.01)	
SGLD (4000)	0.03 (0.01)		0.01 (0.00)		0.02 (0.01)	
	$k=10$	$k=20$	$k=10$	$k=20$	$k=10$	$k=20$
CMC	0.88 (0.05)	0.79 (0.06)	0.88 (0.04)	0.82 (0.07)	0.91 (0.02)	0.85 (0.04)
SDP	0.90 (0.03)	0.89 (0.03)	0.90 (0.03)	0.87 (0.05)	0.92 (0.02)	0.89 (0.04)
WASP	0.93 (0.01)	0.94 (0.01)	0.93 (0.01)	0.94 (0.01)	0.94 (0.01)	0.94 (0.01)

$j = 1, \ldots, p)$ is

$$
\begin{aligned}
x_{ij} \mid \left(\psi_{h1}^{(j)}\right)_{h=1}^{\infty}, \ldots, \left(\psi_{hd_j}^{(j)}\right)_{h=1}^{\infty}, z_i &\sim \text{Multinomial}(\{1, \ldots, d_j\}, \psi_{z_i 1}^{(j)}, \ldots, \psi_{z_i d_j}^{(j)}), \\
z_i &\sim \sum_{h=1}^{\infty} V_h \prod_{l<h} (1 - V_l)\delta_h \equiv \sum_{h=1}^{\infty} \nu_h \delta_h, \quad V_h \sim \text{Beta}(1, \alpha), \\
\psi_h^{(j)} &\sim \text{Dirichlet}(a_{j1}, \ldots, a_{jd_j}), \quad \alpha \sim \text{Gamma}(a_\alpha, b_\alpha), \quad (19)
\end{aligned}
$$

where α has prior mean a_α / b_α. The hierarchical model for probabilistic parafac implies that

$$
\text{pr}(x_{i1} = c_1, \ldots, x_{ij} = c_j, \ldots, x_{ip} = c_p) = \pi_{c_1, \ldots, c_p} = \sum_{h=1}^{\infty} \nu_h \prod_{j=1}^{p} \psi_{hc_j}^{(j)}. \quad (20)
$$

The x_{ij}s are sampled independently given the latent class z_i and probability vectors $\boldsymbol{\psi}_h^{(j)}$ $(h = 1, \ldots, \infty)$. The latent class for every sample is generated using the stick breaking representation of Dirichlet processes. The Gibbs sampling algorithm developed in Dunson and Xing (2009) is very slow even for moderate sample sizes. This example demonstrates that WASP can easily scale existing sampling algorithms to massive data, even when efficient VB alternatives are unavailable.

Table 5: Accuracies of the approximate two-dimensional joint posteriors for the covariances of random effects when $r = 6$ in (18). The accuracies are averaged over 10 simulation replications. Monte Carlo errors are in parenthesis. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$(\sigma_{12}, \sigma_{13})$		$(\sigma_{12}, \sigma_{23})$		$(\sigma_{13}, \sigma_{32})$	
ADVI	0.06 (0.16)		0.08 (0.22)		0.08 (0.17)	
SA	0.89 (0.02)		0.90 (0.02)		0.91 (0.02)	
SGLD (2000)	0.02 (0.01)		0.01 (0.01)		0.01 (0.01)	
SGLD (4000)	0.02 (0.01)		0.01 (0.01)		0.01 (0.01)	
	$k = 10$	$k = 20$	$k = 10$	$k = 20$	$k = 10$	$k = 20$
CMC	0.88 (0.05)	0.76 (0.10)	0.88 (0.04)	0.78 (0.07)	0.90 (0.04)	0.83 (0.07)
SDP	0.90 (0.03)	0.86 (0.05)	0.90 (0.04)	0.86 (0.04)	0.90 (0.03)	0.87 (0.04)
WASP	0.93 (0.02)	0.94 (0.01)	0.94 (0.01)	0.94 (0.01)	0.94 (0.01)	0.94 (0.02)

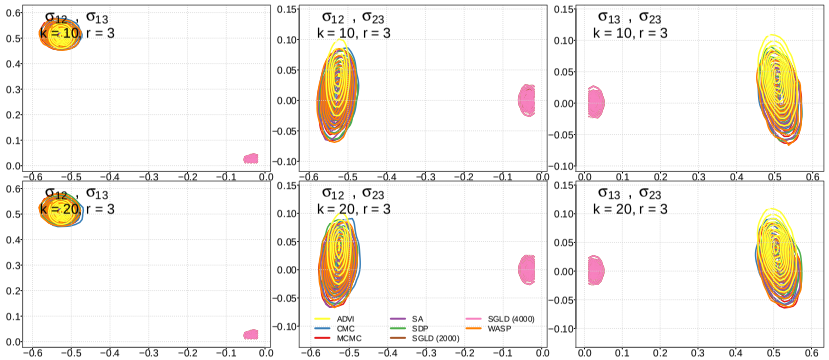


Figure 3: Kernel density estimates of the posterior densities of three covariance pairs when $r = 3$ in (18), where σ_{ab}, σ_{cd} on every panel represents the two-dimensional posterior density of $(\sigma_{ab}, \sigma_{cd})$. ADVI, automatic differentiation variational inference; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; MCMC, Markov chain Monte Carlo; SA, streamlined algorithm; SDP, semiparametric density product; WASP, Wasserstein posterior.

We followed the simulation setup in Dunson and Xing (2009), except with a much larger sample size. We fixed the sample size, number of dimensions, and number of categories in each dimension at $n = 10^5$, $p = 20$, and $d_j = 2$ $(j = 1, \ldots, p)$, respectively. These choices of n, p, and d_js ensured that computations for sampling from the full data posterior were tractable. Data were simulated as a mixture of two populations such that any sample belonged to the two populations with equal probability. The two categories in every dimension excluding $2, 4, 12$, and 14 were simulated from a discrete uniform in both populations. The dependence across dimensions $2, 4, 12$, and 14 was induced as follows. The probabilities π_2, π_4, π_{12}, and π_{14} were set to (0.20, 0.80), (0.25, 0.75), (0.80, 0.20), and (0.75, 0.25) in the first population and to (0.80, 0.20), (0.75, 0.25), (0.20, 0.80), and (0.25, 0.75) in the second population. The simulation setup was replicated 10 times.

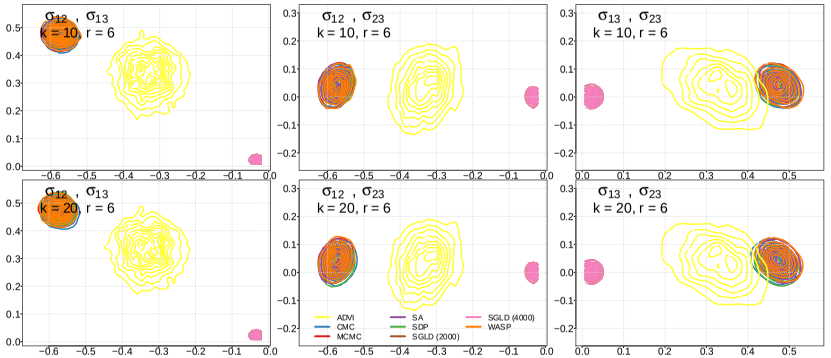


Figure 4: Kernel density estimates of the posterior densities of three covariance pairs when $r = 6$ in (18), where σ_{ab}, σ_{cd} on every panel represents the two-dimensional posterior density of $(\sigma_{ab}, \sigma_{cd})$. ADVI, automatic differentiation variational inference; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; MCMC, Markov chain Monte Carlo; SA, streamlined algorithm; SDP, semiparametric density product; WASP, Wasserstein posterior.

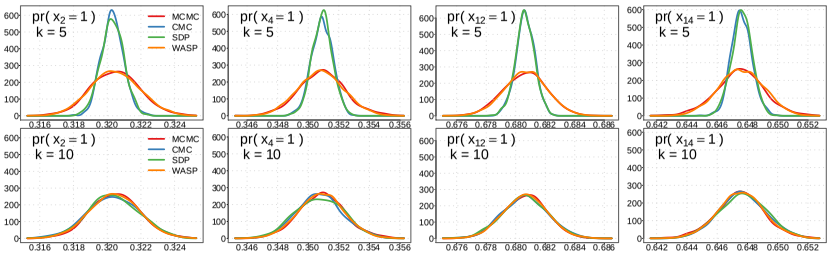


Figure 5: Kernel density estimates of the marginal posterior densities for dimensions 2, 4, 12, and 14. MCMC, Gibbs sampling algorithm of Dunson and Xing (2009); CMC, consensus Monte Carlo; SDP, semiparametric density product; VB, variational Bayes; WASP, Wasserstein posterior

We used CMC, SDP, and WASP to approximate the full data posterior distributions for $\text{pr}(x_i = 1)$, where $i \in \{2, 4, 12, 14\}$. Two values of $k \in \{5, 10\}$ were used for CMC, SDP, and WASP. The full data were randomly partitioned into k subsets and subset posterior samples for WASP were obtained after modifying the Gibbs sampling algorithm in Dunson and Xing (2009) using (5). Examples for the application of CMC and SDP were unavailable for Dirchlet process mixtures, and it was unclear how to raise the prior density to the power $1/k$ when the prior distribution has an atomic form similar to that in (19); therefore, we did not raise the prior to a power of $1/k$ for sampling from the subset posterior distributions

Table 6: Accuracies of the approximate marginal posterior distributions for dimensions 2, 4, 12, and 14 in (19). The accuracies are averaged over 10 simulation replications. Monte Carlo errors are in parenthesis. CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$k = 5$			$k = 10$		
	CMC	SDP	WASP	CMC	SDP	WASP
$\text{pr}(x_2 = 1)$	0.63 (0.02)	0.62 (0.02)	0.97 (0.01)	0.96 (0.02)	0.95 (0.01)	0.97 (0.01)
$\text{pr}(x_4 = 1)$	0.63 (0.02)	0.62 (0.02)	0.97 (0.01)	0.96 (0.01)	0.95 (0.02)	0.97 (0.01)
$\text{pr}(x_{12} = 1)$	0.62 (0.02)	0.62 (0.02)	0.97 (0.01)	0.95 (0.01)	0.96 (0.02)	0.97 (0.01)
$\text{pr}(x_{14} = 1)$	0.64 (0.01)	0.63 (0.01)	0.97 (0.01)	0.96 (0.02)	0.95 (0.02)	0.97 (0.01)

in CMC and SDP. The sampling algorithm for WASP based on stochastic approximation is summarized in the Supplementary Material. Subset posterior samples for $\text{pr}(x_2 = 1)$, $\text{pr}(x_4 = 1)$, $\text{pr}(x_{12} = 1)$, and $\text{pr}(x_{14} = 1)$ were combined to obtain their approximate posterior distributions using CMC, SDP, and WASP.

The accuracy (14) of CMC and SDP in approximating the full data marginal posterior distribution depended on k, with WASP outperforming CMC and SDP when $k = 5$ (Table 6). The approximate and full data posterior distributions were centered at the same value across all dimensions and replications, but the posterior densities for CMC and SDP were highly concentrated compared to the full data posterior density when $k = 5$ (Figure 5). The accuracy of WASP remained stable with varying k, providing an empirical verification of Theorem 4 in cases where our theory is not applicable. The time spent in combining subset posterior samples was negligible compared to the time spent in sampling; therefore, WASP could be used for data with much larger sample size by choosing k large enough such that sampling was efficient across all the data subsets.

4.5 Real Data: MovieLens Ratings Data

We used MovieLens data to illustrate the application of WASP to large-scale ratings data. MovieLens data are one of the largest publicly available ratings data with about 10 million ratings from about 72 thousand users of the MovieLens recommender system. Each observation in the database consists of a user, movie, rating of the movie from 0.5 to 5 in increments of 0.5, and the time of rating. Every movie is also classified into at least one of the 19 genres. We fit a linear mixed effects model (18) using movie- and user-specific information as predictors and the ratings as responses.

We generated three new predictors for accurate modeling of ratings following Perry (2017). First, movie genres were grouped into *movie categories* to reduce the number of genres from 19 to four: *Action* category included Action, Adventure, Fantasy, Horror, Sci-Fi, and Thriller genres; *Children* category included Animation and Children genres; *Comedy* category included Comedy genre; and *Drama* category included Crime, Documentary, Drama, Film-Noir, Musical, Mystery, Romance, War, and Western genres. If a movie belonged to multiple genres, then movie category scores were fractions proportional to the number of genres in the respective categories. Second, *popularity* predictor was defined as $\text{logit}\{(l+0.5)/(n+1.0)\}$, where l and n respectively were the number of users who liked and rated the movie in 30 most recent observations for the movie and $\text{logit}(x) = \log \frac{x}{1-x}$. Third, *previous* predictor was defined to be 1 if the user liked the previous movie and 0 otherwise.

Table 7: Accuracies of the approximate posteriors for variances in (18). The accuracies are averaged over 10 replications. Monte Carlo errors are in parenthesis. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	σ^2_{Action}	$\sigma^2_{\text{Children}-\text{Action}}$	$\sigma^2_{\text{Comedy}-\text{Action}}$	$\sigma^2_{\text{Drama}-\text{Action}}$	$\sigma^2_{\text{Popularity}}$	$\sigma^2_{\text{Previous}}$
ADVI	0.06 (0.14)	0.33 (0.30)	0.16 (0.23)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
SA	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
SGLD (2000)	0.10 (0.06)	0.06 (0.03)	0.05 (0.05)	0.08 (0.04)	0.10 (0.00)	0.10 (0.07)
SGLD (4000)	0.07 (0.06)	0.06 (0.03)	0.02 (0.06)	0.08 (0.04)	0.10 (0.00)	0.08 (0.07)
CMC	0.28 (0.13)	0.01 (0.01)	0.01 (0.01)	0.14 (0.09)	0.74 (0.10)	0.22 (0.10)
SDP	0.05 (0.03)	0.00 (0.00)	0.00 (0.00)	0.01 (0.01)	0.35 (0.10)	0.03 (0.03)
WASP	0.92 (0.04)	0.93 (0.02)	0.87 (0.06)	0.85 (0.08)	0.92 (0.03)	0.93 (0.05)

We used *Action*, *Children* − *Action*, *Comedy* − *Action*, *Drama* − *Action*, *popularity*, and *previous* as the fixed and random effects in (18).

Following the setup in Section 4.3, we compared the performance of WASP with ADVI, CMC, SA, SGLD with batch sizes 2000, 4000, step size 10^{-5} and 10^4 iterations, and SDP using the full data posterior distribution as the benchmark. Sampling using the HMC algorithm in Stan was prohibitively slow for the full data posterior distribution, so we first randomly selected 5000 users and then randomly selected 20 ratings for every user. This resulted in a data set with 100,000 ratings. We randomly split the users into 10 training data sets such that ratings for any user belonged to the same training data set. To compute the approximate posteriors using CMC, SDP, and WASP, we set $k = 10$ and randomly partitioned the users into k subsets such that each subset contained all the ratings for a user. This setup was replicated for every training data.

WASP performed better than its competitors in approximating the full data posterior distributions for variances and covariances of the random effects. Similar to the simulation results in Section 4.3, ADVI, CMC, SA, SDP, and WASP were significantly faster than the full data posterior distribution, with SA being the fastest, and SGLD was the slowest. CMC, SDP, and WASP showed excellent performed in approximating the full data posterior distributions for the fixed effects. WASP outperformed its competitors in approximating the full data posterior distributions for variances, covariances, and pairs of covariances of the random effects (Tables 7, 8, and 9). ADVI, SA, and SGLD significantly under-performed in the estimation of the posterior distribution for the fixed effects and covariance matrix of the random effects. The accuracy of marginals in CMC and SDP depended on the magnitude of covariances, with both methods showing excellent accuracy for covariances with low magnitude. The accuracies of the two-dimensional joint distributions in CMC and SDP were poor because the full data posteriors concentrated at different locations (Figure 6). Except for the poor performance of CMC, SA, and SDP in approximating the posterior distribution of variances and covariances of the random effects, our real data results agreed with our simulation results. We concluded that WASP performed better than its competitors in MovieLens data analysis.

Table 8: Accuracies of the approximate posteriors for covariances in (18). The accuracies are averaged over 10 replications. Monte Carlo errors are in parenthesis. The subscripts $1, \ldots, 6$ are used for predictors *Action*, *Children* − *Action*, *Comedy* − *Action*, *Drama* − *Action*, *popularity*, and *previous*. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	σ_{12}	σ_{13}	σ_{14}	σ_{15}	σ_{16}	σ_{23}	σ_{24}	σ_{25}
ADVI	0.15 (0.30)	0.25 (0.26)	0.14 (0.16)	0.32 (0.12)	0.06 (0.09)	0.00 (0.00)	0.18 (0.20)	0.66 (0.15)
SA	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
SGLD (2000)	0.08 (0.03)	0.19 (0.10)	0.18 (0.08)	0.18 (0.11)	0.23 (0.09)	0.14 (0.00)	0.14 (0.01)	0.14 (0.10)
SGLD (4000)	0.08 (0.02)	0.16 (0.10)	0.14 (0.08)	0.12 (0.08)	0.20 (0.08)	0.14 (0.00)	0.13 (0.01)	0.11 (0.10)
CMC	0.06 (0.03)	0.16 (0.04)	0.18 (0.04)	0.83 (0.07)	0.33 (0.13)	0.01 (0.01)	0.07 (0.02)	0.80 (0.04)
SDP	0.01 (0.01)	0.08 (0.03)	0.07 (0.02)	0.75 (0.06)	0.14 (0.09)	0.00 (0.00)	0.02 (0.01)	0.73 (0.08)
WASP	0.95 (0.02)	0.91 (0.04)	0.91 (0.05)	0.94 (0.03)	0.90 (0.07)	0.89 (0.07)	0.85 (0.08)	0.93 (0.03)
	σ_{26}	σ_{34}	σ_{35}	σ_{36}	σ_{45}	σ_{46}	σ_{56}	
ADVI	0.47 (0.22)	0.50 (0.22)	0.64 (0.11)	0.62 (0.23)	0.64 (0.18)	0.49 (0.29)	0.42 (0.11)	
SA	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.01 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	
SGLD (2000)	0.11 (0.10)	0.14 (0.10)	0.16 (0.07)	0.10 (0.11)	0.14 (0.12)	0.12 (0.10)	0.15 (0.09)	
SGLD (4000)	0.07 (0.10)	0.11 (0.09)	0.14 (0.07)	0.03 (0.11)	0.10 (0.11)	0.08 (0.10)	0.14 (0.08)	
CMC	0.66 (0.09)	0.65 (0.07)	0.76 (0.08)	0.71 (0.05)	0.82 (0.04)	0.61 (0.11)	0.55 (0.09)	
SDP	0.59 (0.11)	0.62 (0.06)	0.64 (0.09)	0.66 (0.08)	0.66 (0.09)	0.56 (0.14)	0.55 (0.13)	
WASP	0.91 (0.05)	0.94 (0.05)	0.93 (0.03)	0.91 (0.04)	0.93 (0.04)	0.93 (0.04)	0.94 (0.04)	

Table 9: Accuracies of the approximate two-dimensional joint posteriors for the covariances of random effects. The accuracies are averaged over 10 replications. Monte Carlo errors are in parenthesis. The subscripts $1, \ldots, 6$ are used for predictors *Action*, *Children* − *Action*, *Comedy* − *Action*, *Drama* − *Action*, *popularity*, and *previous*. ADVI, automatic differentiation variational inference; SA, streamlined algorithm; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; SDP, semiparametric density product; WASP, Wasserstein posterior

	$(\sigma_{12}, \sigma_{13})$	$(\sigma_{12}, \sigma_{14})$	$(\sigma_{12}, \sigma_{15})$	$(\sigma_{12}, \sigma_{16})$
ADVI	0.03 (0.06)	0.03 (0.07)	0.02 (0.06)	0.05 (0.11)
SA	0.18 (0.04)	0.22 (0.07)	0.31 (0.03)	0.31 (0.02)
SGLD (2000)	0.01 (0.02)	0.01 (0.02)	0.01 (0.01)	0.01 (0.01)
SGLD (4000)	0.01 (0.02)	0.01 (0.02)	0.01 (0.01)	0.01 (0.01)
CMC	0.05 (0.02)	0.04 (0.02)	0.06 (0.03)	0.05 (0.02)
SDP	0.05 (0.02)	0.04 (0.02)	0.06 (0.03)	0.05 (0.02)
WASP	0.88 (0.03)	0.88 (0.03)	0.88 (0.02)	0.86 (0.06)

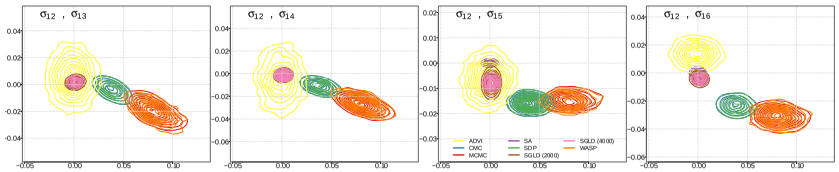


Figure 6: Kernel density estimates of the posterior densities of four covariance pairs, where σ_{ab}, σ_{cd} on every panel represents the two-dimensional posterior density of $(\sigma_{ab}, \sigma_{cd})$. ADVI, automatic differentiation variational inference; SGLD, stochastic gradient Langevin dynamics with batch size in parenthesis; CMC, consensus Monte Carlo; MCMC, Markov chain Monte Carlo; SA, streamlined algorithm; SDP, semiparametric density product; WASP, Wasserstein posterior.

5. Discussion

We have presented WASP as an approach for computationally efficient approximation of the posterior distributions of parameters and their functions when the sample size is large. WASP allows extensions of existing samplers to massive data with minimal modifications and is easily implemented using probabilistic programming languages, such as Stan. Theoretically, we have showed that the rate of convergence of WASP to the Dirac measure centered at the true parameter value in W_2 distance matches the optimal parametric rate up to a logarithmic factor if the number of subsets increases slowly with the size of the full data set. Empirically, we demonstrated that results from WASP and MCMC agree closely in several widely different examples, while WASP enables massive speed-ups in computational time.

We plan to explore several extensions of WASP in the future. First, the combination of subset posterior distributions using WASP and the proof of the convergence rate for the WASP in Theorem 4 are valid even if the data in different subsets are dependent; however, independence assumption within each subset is required in the proof of (12) in Theorem 4 and in our justification of stochastic approximation. Currently, it is unclear how to extend stochastic approximation to cases where the likelihood is unavailable in a product form. This extension in crucial for proper uncertainty quantification outside of settings in which the observations are conditionally independent given latent variables. Second, it is unclear how to optimally choose k in practice; larger k improves computational time when abundant processors are available but choosing k too large may lead to increasing statistical errors (refer to Theorem 4). Our numerical experiments show that the accuracy of WASP is robust to the choice of k if all the subset sizes are moderately large relative to the number of parameters. In addition, it is of interest to study more deeply the impact of the partitioning schemes and attempt to develop approaches that deal with not only large sample sizes but also high-dimensional data. A possibility in this regard is to combine WASP with approximate MCMC (Johndrow et al., 2015).

Acknowledgments

Volkan Cevher and Quoc Tran-Dinh proposed and implemented the linear program for calculating Wasserstein barycenter described in Srivastava et al. (2015). All experiments were based on a modified version of Tran-Dinh's Matlab and Gurobi code for estimating Wasserstein barycenter. Jack Baker provided extensive help in implementing the SGLD algorithm. The code used in the experiments is available at `https://github.com/blayes/WASP`. We thank the Associate Editor and two anonymous referees for their helpful comments that improved our paper. Cheng Li's work was partially supported by National University of Singapore start-up grant R155000172133.

Appendix A. Proofs of Theorems

A.1 Proof of Theorem 1

If $E_{P_{\theta_0}^{(n)}}$ represents the expectation with respect to $P_{\theta_0}^{(n)}$, then

$$E_{P_{\theta_0}^{(n)}}\left[W_2^2\{N_p(\mu,V),N_p(\overline{\mu},\overline{V})\}\right] = E_{P_{\theta_0}^{(n)}}\|\mu-\overline{\mu}\|_2^2 + \operatorname{tr}\left\{V+\overline{V}-2(\overline{V}^{1/2}V\overline{V}^{1/2})^{1/2}\right\}. \tag{21}$$

First, we find the asymptotic order of $E_{P_{\theta_0}^{(n)}}\|\mu_1-\mu_2\|_2^2$ in (21). Define

$$A=(X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}, \quad B=k^{-1}\left[(X_1^T\Sigma_1^{-1}X_1)^{-1}X_1^T\Sigma_1^{-1},\cdots,(X_k^T\Sigma_k^{-1}X_k)^{-1}X_k^T\Sigma_k^{-1}\right],$$

and $C=A-B$. After some algebra, we have that $AX=I_p$, $BX=I_p$, where I_p is a $p\times p$ identity matrix, and

$$\|\mu-\overline{\mu}\|_2^2=\|Cy\|_2^2, \quad E_{P_{\theta_0}^{(n)}}\|\mu-\overline{\mu}\|_2^2 = E_{P_{\theta_0}^{(n)}}(y^T)C^TCE_{P_{\theta_0}^{(n)}}(y)+\operatorname{tr}(C\Sigma C^T).$$

Since $E_{P_{\theta_0}^{(n)}}(y)=X\theta_0$ and $CX=AX-BX=I_p-I_p=0$, $E_{P_{\theta_0}^{(n)}}\|\mu-\overline{\mu}\|_2^2=\operatorname{tr}(C\Sigma C^T)$. Expanding $C\Sigma C^T$, we get

$$C=(X^T\Sigma^{-1}X)^{-1}X^T\Sigma^{-1}-k^{-1}\left[(X_1^T\Sigma_1^{-1}X_1)^{-1}X_1^T\Sigma_1^{-1},\cdots,(X_k^T\Sigma_k^{-1}X_k)^{-1}X_k^T\Sigma_k^{-1}\right],$$

$$C^T=\Sigma^{-1}X(X^T\Sigma^{-1}X)^{-1}-k^{-1}\begin{bmatrix}\Sigma_1^{-1}X_1(X_1^T\Sigma_1^{-1}X_1)^{-1}\\ \vdots \\ \Sigma_k^{-1}X_k(X_k^T\Sigma_k^{-1}X_k)^{-1}\end{bmatrix},$$

$$\operatorname{tr}(C\Sigma C^T)=\operatorname{tr}\{(X^T\Sigma X)^{-1}\}+k^{-2}\sum_{j=1}^k\operatorname{tr}\left\{(X_j^T\Sigma_jX_j)^{-1}\right\}-2\operatorname{tr}(D),$$

where

$$\begin{aligned}D&=k^{-1}\left[(X_1^T\Sigma_1X_1)^{-1}X_1^T,\cdots,(X_k^T\Sigma_kX_k)^{-1}X_k^T\right]\Sigma^{-1}X(X^T\Sigma^{-1}X)^{-1}\\&=\left\{k^{-1}\sum_{j=1}^k(X_j^T\Sigma_j^{-1}X_j)^{-1}X_j^T\Sigma_j^{-1}X_j\right\}(X^T\Sigma^{-1}X)^{-1}=(X^T\Sigma^{-1}X)^{-1}\end{aligned}$$

because Σ is diagonal. We use the above display to obtain that

$$\begin{aligned}E_{P_{\theta_0}^{(n)}}\|\mu-\overline{\mu}\|_2^2=\operatorname{tr}(C\Sigma C^T)&=\frac{1}{k^2}\sum_{j=1}^k\operatorname{tr}\left\{(X_j^T\Sigma_j^{-1}X_j)^{-1}\right\}-\operatorname{tr}\left\{(X^T\Sigma^{-1}X)^{-1}\right\},\\&=\frac{1}{km}\operatorname{tr}\left\{\tfrac{1}{k}\sum_{j=1}^k\left(\tfrac{1}{m}X_j^T\Sigma_j^{-1}X_j\right)^{-1}\right\}-\frac{1}{n}\operatorname{tr}\left\{\left(\tfrac{1}{n}X^T\Sigma^{-1}X\right)^{-1}\right\}.\end{aligned}$$

Our assumptions and continuity of the matrix inverse for positive definite matrices imply that there are exist positive $a_n'=o(1)$, $b_m'=o(1)$, such that

$$\begin{aligned}\Omega_0^{-1}-a_n'I_p\prec\left(\tfrac{1}{n}X^T\Sigma^{-1}X\right)^{-1}\prec\Omega_0^{-1}+a_n'I_p,\\ \Omega_0^{-1}-b_m'I_p\prec\left(\tfrac{1}{m}X_j^T\Sigma_j^{-1}X_j\right)^{-1}\prec\Omega_0^{-1}+b_m'I_p.\end{aligned}$$

This implies that the previous display reduces to

$$E_{P_{\theta_0}^{(n)}}\|\mu-\overline{\mu}\|_2^2\le p(b_m'+a_n')/n=o(n^{-1}), \tag{22}$$

where the equality follow because p is fixed.

We now find the asymptotic order of the traces of the covariance matrices in (21). Following the same arguments used to derive (22), the full data and jth subset posterior covariance matrices satisfy

$$\begin{aligned}\frac{1}{n}\left(\Omega_0^{-1}-a_n'I_p\right)\prec V=\frac{1}{n}\left(\frac{1}{n}X^T\Sigma^{-1}X\right)^{-1}\prec\frac{1}{n}\left(\Omega_0^{-1}+a_n'I_p\right),\\ \frac{1}{n}\left(\Omega_0^{-1}-b_m'I_p\right)\prec V_j=\frac{1}{km}\left(\frac{1}{m}X_j^T\Sigma_j^{-1}X_j\right)^{-1}\prec\frac{1}{n}\left(\Omega_0^{-1}+b_m'I_p\right).\end{aligned} \tag{23}$$

Let $M_j=\left\{\overline{V}^{1/2}\frac{1}{km}\left(\frac{1}{m}X_j^T\Sigma_j^{-1}X_j\right)^{-1}\overline{V}^{1/2}\right\}^{1/2}$. Then (23) implies that

$$-b_m'\overline{V}\prec nM_j^2-\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}=n\overline{V}^{1/2}\left(V_j-n^{-1}\Omega_0^{-1}\right)\overline{V}^{1/2}\prec b_m'\overline{V}. \tag{24}$$

From the first inequality in (24), we have

$$\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\prec\left(nM_j^2+b_m'\overline{V}\right)^{1/2}\prec n^{1/2}M_j+b_m'^{1/2}\overline{V}^{1/2}.$$

And similarly the second inequality in (24) implies that

$$n^{1/2}M_j\prec\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}+b_m'\overline{V}\right)^{1/2}\prec\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}+b_m'^{1/2}\overline{V}^{1/2}.$$

Therefore

$$\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}-b_m'^{1/2}\overline{V}^{1/2}\prec n^{1/2}M_j\prec\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}+b_m'^{1/2}\overline{V}^{1/2}.$$

Using this relation and the definition of $\overline{V}$, we have that

$$\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}-b_m'^{1/2}\overline{V}^{1/2}\prec n^{1/2}\overline{V}=\frac{1}{k}\sum_{j=1}^k n^{1/2}M_j\prec\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}+b_m'^{1/2}\overline{V}^{1/2}. \tag{25}$$

In (25), we take the square of $n^{1/2}\overline{V}$, apply the inequality $(A_1+A_2)^2 \prec 2(A_1^2+A_2^2)$ for two generic positive definite matrices A_1, A_2, and obtain that

$$n\overline{V}^2 \prec 2\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2} + 2b'_m\overline{V},$$
$$n\overline{V}^2 \succ \frac{1}{2}\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2} - b'_m\overline{V}.$$

Multiplying by $\overline{V}^{-1/2}$ on the left and right hand sides yields,

$$n\overline{V} \prec 2\Omega_0^{-1} + 2b'_m I_p,$$
$$n\overline{V} \succ \frac{1}{2}\Omega_0^{-1} - b'_m I_p. \tag{26}$$

Notice that $b'_m = o(1)$, Ω_0 is a constant positive definite matrix, and $\overline{V}$ is a positive definite matrix. Clearly, (26) forces $n\overline{V}$ to be an order-1 matrix. Now we take the square of $n^{1/2}\overline{V}$ in (25) again and obtain that

$$n\overline{V}^2 \prec \overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2} + b'_m\overline{V} + b'^{1/2}_m\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\overline{V}^{1/2} + b'^{1/2}_m\overline{V}^{1/2}\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2},$$
$$n\overline{V}^2 \succ \overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2} + b'_m\overline{V} - b'^{1/2}_m\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\overline{V}^{1/2} - b'^{1/2}_m\overline{V}^{1/2}\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}.$$

Multiplying by $\overline{V}^{-1/2}$ on the left and right hand sides yields,

$$n\overline{V} \prec \Omega_0^{-1} + b'_m I_p + b'^{1/2}_m\overline{V}^{-1/2}\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2} + b'^{1/2}_m\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\overline{V}^{-1/2},$$
$$n\overline{V} \succ \Omega_0^{-1} + b'_m I_p - b'^{1/2}_m\overline{V}^{-1/2}\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2} - b'^{1/2}_m\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\overline{V}^{-1/2}. \tag{27}$$

Since $n\overline{V}$ is an order-1 matrix, we have that $b'_m\overline{V}^{-1/2}\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2} = o(1)$, $b'_m\left(\overline{V}^{1/2}\Omega_0^{-1}\overline{V}^{1/2}\right)^{1/2}\overline{V}^{-1/2} = o(1)$. Hence (23) and (27) reduce to

$$\frac{1}{n}\{\Omega_0^{-1} - o(1)I_p\} \prec V_j \prec \frac{1}{n}\{\Omega_0^{-1} + o(1)I_p\}, \quad \frac{1}{n}(\Omega_0^{-1} - o(1)I_p) \prec \overline{V} \prec \frac{1}{n}\{\Omega_0^{-1} + o(1)I_p\}.$$

This implies that

$$\mathrm{tr}(\overline{V} - V) = o(n^{-1}), \tag{28}$$

where the last equality follows because p is fixed.

Finally, we find the asymptotic order of the variance term in (21). The display before (28) implies that for some positive $c_n = o(1)$,

$$\overline{V}^{1/2}V\overline{V}^{1/2} \prec \frac{1}{n^2}\{\Omega_0^{-1/2} + o(1)I_p\}\{\Omega_0^{-1} + o(1)I_p\}\{\Omega_0^{-1/2} + o(1)I_p\}$$
$$\prec \frac{1}{n^2}[\Omega_0^{-2} + c_n I_p],$$
$$\overline{V}^{1/2}V\overline{V}^{1/2} \succ \frac{1}{n^2}\{\Omega_0^{-1/2} - o(1)I_p\}\{\Omega_0^{-1} - o(1)I_p\}\{\Omega_0^{-1/2} - o(1)I_p\}$$

$$\succ \frac{1}{n^2}[\Omega_0^{-2} - c_n I_p].$$

Therefore, $\mathrm{tr}\{(\overline{V}^{1/2}V\overline{V}^{1/2})^{1/2}\} = n^{-1}\,\mathrm{tr}(\Omega_0^{-1}) + o(n^{-1})$ since p is fixed. Using this and (23) for the variance term in (21) gives

$$\begin{aligned}&\mathrm{tr}\left\{V + \overline{V} - 2\left(\overline{V}^{1/2}V\overline{V}^{1/2}\right)^{1/2}\right\}\\ &= \{n^{-1}\,\mathrm{tr}(\Omega_0^{-1}) + o(n^{-1})\} + \{n^{-1}\,\mathrm{tr}(\Omega_0^{-1}) + o(n^{-1})\} - \{2n^{-1}\,\mathrm{tr}(\Omega_0^{-1}) + 2o(n^{-1})\}\\ &= o\left(n^{-1}\right).\end{aligned} \tag{29}$$

Combining the asymptotic expressions for the mean and variance terms in (22) and (29), (21) reduces to

$$E_{P_{\theta_0}^{(n)}}\left[W_2^2\left\{N(\overline{\mu},\overline{V}), N(\mu,V)\right\}\right] = o\left(n^{-1}\right),$$

which completes the proof. □

A.2 Proof of Theorem 4

Let $\epsilon_m = \left(\frac{m}{\log^2 m}\right)^{-1/(2\alpha)}$. For ease of notation, in all the following proofs, we will sometimes write $p(y_{ji} \mid \theta) \equiv p_{ji}(y_{ji} \mid \theta)$.

Due to the compactness of Θ in (A1), we assume that $\rho(\theta,\theta_0) \le M_0$ for a large finite constant M_0. We start with a decomposition of the W_2 distance from the jth subset posterior $\Pi_m(\cdot \mid Y_{[j]})$ to the Dirac measure at the true parameter θ_0:

$$\begin{aligned}&E_{P_{\theta_0}}W_2^2\left(\Pi_m(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot)\right) = E_{P_{\theta_0}}\int_\Theta \rho^2(\theta,\theta_0)\Pi_m(d\theta \mid Y_{[j]})\\ \le& E_{P_{\theta_0}}\int_{\{\theta:\rho(\theta,\theta_0)\le c_1\epsilon_m\}}\rho^2(\theta,\theta_0)\Pi_m(d\theta \mid Y_{[j]}) + E_{P_{\theta_0}}\int_{\{\theta:\rho(\theta,\theta_0)>c_1\epsilon_m\}}\rho^2(\theta,\theta_0)\Pi_m(d\theta \mid Y_{[j]})\\ \le&(c_1\epsilon_m)^2 + M_0^2 E_{P_{\theta_0}}\Pi_m\left(\rho(\theta,\theta_0) > c_1\epsilon_m \mid Y_{[j]}\right).\end{aligned} \tag{30}$$

We will choose the constant c_1 as $c_1 = \left(\frac{2r_1g_2}{q_1C_L}\right)^{1/(2\alpha)}$, where g_1, C_L, q_1, r_1 are the constants in (A1), (A2), and Lemma 5 and Lemma 6 in the Supplementary Material.

The following proofs are similar to the proofs of Theorem 1, 4, and 10 in Ghosal and van der Vaart (2007). The main difference is that our likelihood has been raised to the power γ. Using condition (A2), we can further replace the ρ metric by the pseudo Hellinger distance:

$$\begin{aligned}&\Pi_m\left(\theta\in\Theta : \rho(\theta,\theta_0) > c_1\epsilon_m \mid Y_{[j]}\right)\\ \le&\,\Pi_m\left(\theta\in\Theta : h_{mj}(P_{\theta,j}, P_{\theta_0,j}) > \sqrt{C_L}(c_1\epsilon_m)^\alpha \mid Y_{[j]}\right)\\ =&\int_{\left\{\theta\in\Theta:h_{mj}(\theta,\theta_0)>\sqrt{\frac{2r_1g_2}{q_1}}\epsilon_m^\alpha\right\}}\frac{\prod_{i=1}^m\left[\frac{p(Y_{ji}|\theta)}{p(Y_{ji}|\theta_0)}\right]^\gamma\Pi(d\theta)}{\int_\Theta\prod_{i=1}^m\left[\frac{p(Y_{ji}|\theta)}{p(Y_{ji}|\theta_0)}\right]^\gamma\Pi(d\theta)}.\end{aligned} \tag{31}$$

For the denominator in (31), by Condition (A4) and Lemma 6, for m sufficiently large, with probability at least $1-\exp(-r_2 m\epsilon_m^{2\alpha})$

$$\int_\Theta \prod_{i=1}^m \frac{p(Y_{ji}|\theta)^\gamma}{p(Y_{ji}|\theta_0)^\gamma}\Pi(d\theta) > \exp(-r_1 n\epsilon_m^{2\alpha}). \tag{32}$$

For the numerator in (31), by Condition (A3) and Lemma 5, we set $\delta = \sqrt{2r_1 g_2/q_1}\epsilon_m^\alpha$ and obtain that with probability at least $1-4\exp\left(-\frac{2r_1 g_2 q_2}{q_1} m\epsilon_m^{2\alpha}\right) \geq 1-4\exp\left(-\frac{2r_1 q_2}{q_1} n\epsilon_m^{2\alpha}\right)$,

$$\sup_{\left\{\theta\in\Theta: h_{mj}(\theta,\theta_0)\geq\sqrt{2r_1 g_2/q_1}\epsilon_m^\alpha\right\}} \prod_{i=1}^m \left[\frac{p(Y_{ji}|\theta)}{p(Y_{ji}|\theta_0)}\right]^\gamma \leq \exp\left(-2r_1 g_2 m\epsilon_m^{2\alpha}\right) \leq \exp\left(-2r_1 n\epsilon_m^{2\alpha}\right) \tag{33}$$

Therefore, based on (31), (32), and (33), we obtain that with probability at least $1-4\exp\left(-\frac{2r_1 q_2}{q_1} n\epsilon_m^{2\alpha}\right) - \exp(-r_2 m\epsilon_m^{2\alpha})$,

$$\Pi_m\left(\theta\in\Theta : \rho(\theta,\theta_0) > c_1\epsilon_m \;\middle|\; Y_{[j]}\right) \leq \exp\left(-2r_1 n\epsilon_m^{2\alpha} + r_1 n\epsilon_m^{2\alpha}\right) \leq \exp\left(-r_1 n\epsilon_m^{2\alpha}\right).$$

Let A_{ϵ_m} be the event $\left\{\theta\in\Theta : \Pi\left(\theta\in\Theta : \rho(\theta,\theta_0) > c_1\epsilon_m \;\middle|\; Y_{[j]}\right) \leq \exp\left(-r_1 n\epsilon_m^{2\alpha}\right)\right\}$. Then we can bound the second term in (30) as

$$\begin{aligned}
&E_{P_{\theta_0}}\Pi_m\left(\rho(\theta,\theta_0) > c_1\epsilon_m \;\middle|\; Y_{[j]}\right)\\
&\leq E_{P_{\theta_0}}\left[I(A_{\epsilon_m})\Pi_m\left(\rho(\theta,\theta_0) > c_1\epsilon_m \;\middle|\; Y_{[j]}\right)\right] + E_{P_{\theta_0}}\left[I(A_{\epsilon_m}^c)\Pi_m\left(\rho(\theta,\theta_0) > c_1\epsilon_m \;\middle|\; Y_{[j]}\right)\right]\\
&\leq \exp\left(-r_1 n\epsilon_m^{2\alpha}\right) + P_{\theta_0}^{(n)}(A_{\epsilon_m}^c)\cdot 1\\
&\leq \exp\left(-r_1 n\epsilon_m^{2\alpha}\right) + 4\exp\left(-\frac{2r_1 q_2}{q_1} n\epsilon_m^{2\alpha}\right) + \exp(-r_2 m\epsilon_m^{2\alpha})\\
&\leq 6\exp\left(-c_2 m\epsilon_m^{2\alpha}\right),
\end{aligned}$$

for $c_2 = \min(r_1, r_2, 2r_1 q_2/q_1)$, as clearly the second term is dominating the other two given $m \lesssim n$.

Therefore, for (30), since $\epsilon_m = (m/\log^2 m)^{-1/(2\alpha)}$, as $m\to\infty$, an explicit bound will be

$$\begin{aligned}
&E_{P_{\theta_0}} W_2^2\left(\Pi_m(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot)\right) \leq c_1^2\frac{\log^{2/\alpha} m}{m^{1/\alpha}} + 6M_0^2\exp\left(-c_2\log^2 m\right)\\
&\leq c_1^2\frac{\log^{2/\alpha} m}{m^{1/\alpha}} + \frac{1}{m^{1+\frac{1}{\alpha}}} \leq C_1\frac{\log^{2/\alpha} m}{m^{1/\alpha}}
\end{aligned}$$

as m becomes sufficiently large, where the constant C_1 depends on α, c_1, c_2, which further depends on $g_1, g_2, q_1, q_2, r_1, r_2, C_L$. Since q_1, q_2 in Lemma 5 and r_1, r_2 in Lemma 6 depend on $g_1, g_2, D_1, D_2, \kappa, c_\pi$, it follows that C_1 depends on $g_1, g_2, C_L, D_1, D_2, \kappa, c_\pi$. □

Based on Lemma 7 in the Supplementary Material, if the assumption (A5) holds, then we have

$$E_{P_{\theta_0}^{(n)}}\left[W_2^2\left\{\overline{\Pi}_n(\cdot \mid Y^{(n)}), \delta_{\theta_0}(\cdot)\right\}\right] \leq E_{P_{\theta_0}^{(n)}}\left[\frac{1}{k}\sum_{j=1}^k W_2\left\{\Pi_m(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot)\right\}\right]^2$$

$$\leq \frac{1}{k}\sum_{j=1}^k E_{P_{\theta_0}^{(n)}} W_2^2\left\{\Pi_m(\cdot \mid Y_{[j]}), \delta_{\theta_0}(\cdot)\right\} \leq C_1\frac{\log^{2/\alpha} m}{m^{1/\alpha}},$$

where the first inequality follows from Lemma 7 in the Supplementary Material, the second inequality follows from the Cauchy-Schwarz inequality, and the third inequality follows from the subset bound (12). □

Appendix B. Univariate Density Estimation

Let $X_1, \ldots, X_n$ be n copies of a scalar random variable X that follows probability distribution P_0 with density p_0. The full data are randomly split into k subsets and $X_{j1}, \ldots, X_{jm}$ represent the data on subset j ($j = 1, \ldots, k$). The hierarchical model for density estimation using the stick-breaking representation of Dirichlet process mixtures is

$$\begin{aligned}
&X_{ji} \mid z_{ji}, \{\mu_h\}_{h=1}^\infty, \{\sigma_h^2\}_{h=1}^\infty \sim \mathcal{N}(\mu_{z_{ji}}, \sigma_{z_{ji}}^2), \quad z_{ji} \sim \sum_{h=1}^\infty \nu_h\delta_h, \quad \nu_h = V_h\prod_{l<h}(1-V_l), \quad V_h \mid \alpha \sim \text{Beta}(1,\alpha),\\
&\alpha \sim \text{Gamma}(a_\alpha, b_\alpha), \quad \mu_h \mid \sigma_h^2 \sim \mathcal{N}(0, \sigma_h^2), \quad \sigma_h^2 \sim \text{Inverse-Gamma}(a_\sigma, b_\sigma),
\end{aligned} \tag{34}$$

where $a_\sigma > 2$ and Beta, Gamma, and Inverse-Gamma random variables have means $\frac{1}{1+\alpha}$, $\frac{a_\alpha}{b_\alpha}$, and $\frac{b_\sigma}{a_\sigma - 1}$ and variances $\frac{\alpha}{(1+\alpha)^2(2+\alpha)}$, $\frac{a_\alpha}{b_\alpha^2}$, and $\frac{b_\sigma^2}{(a_\sigma-1)^2(a_\sigma-2)}$. If l^* is the maximum number of atoms in the stick-breaking representation, then the prior density π is in the form a discrete mixture. We cannot use existing sampling algorithms directly if π is raised to a power of $1/k$, so it is unclear how to sample from the subset posterior density of competing approaches in Section 2.2.

We show that it is still possible to sample from the subset posterior density in (5) using data augmentation. Let L_j be the likelihood given $X_{j1}, \ldots, X_{jm}$ and latent variables $z_{j1}, \ldots, z_{jm}$ in (34), then

$$L_j(\{\mu_h\}_{h=1}^{l^*}, \{\sigma_h^2\}_{h=1}^{l^*}, \{\nu_h\}_{h=1}^{l^*}) = \prod_{h=1}^{l^*}(2\pi\sigma_h^2)^{-\frac{\sharp h_j}{2}} e^{-\frac{1}{2\sigma_h^2}\sum_{i=1}^m 1(z_{ji}=h)(x_{ji}-\mu_h)^2}\nu_h^{\sharp h_j}, \tag{35}$$

where $1(z_{ji}=h)$ is 1 if $z_{ji}=h$ and 0 otherwise and $\sharp h_j = \sum_{i=1}^m 1(z_{ji}=h)$. For stochastic approximation, we raise L_j in (35) to the power γ and obtain

$$L_j^\gamma(\{\mu_h\}_{h=1}^{l^*}, \{\sigma_h^2\}_{h=1}^{l^*}, \{\nu_h\}_{h=1}^{l^*}) = \prod_{h=1}^{l^*}(2\pi\sigma_h^2)^{-\frac{\gamma\sharp h_j}{2}} e^{-\frac{\gamma}{2\sigma_h^2}\sum_{i=1}^m 1(z_{ji}=h)(x_{ji}-\mu_h)^2}\nu_h^{\gamma\sharp h_j}. \tag{36}$$

Standard arguments imply that the analytic form of full conditional densities of parameters are

$$\begin{aligned}
\mu_h \mid \text{rest} &\propto e^{-\frac{\gamma\sharp h_j+1}{2\sigma_h^2}\left(\mu_h^2 - 2\mu_h\gamma\frac{\sum_{i=1}^m 1(z_{ji}=h)x_{ji}}{\gamma\sharp h_j+1}\right)},\\
\sigma_h^2 \mid \text{rest} &\propto \sigma_h^{2-\frac{\gamma\sharp h_j}{2}} e^{-\frac{\gamma}{2\sigma_h^2}\sum_{i=1}^m 1(z_{ji}=h)(x_{ji}-\mu_h)^2}\sigma_h^{2-\frac{1}{2}} e^{-\frac{\mu_h^2}{2\sigma_h^2}}\sigma_h^{2-a_\sigma-1} e^{-\frac{b_\sigma}{\sigma_h^2}},\\
V_h \mid \text{rest} &\propto V_h^{\gamma\sum_{i=1}^m 1(z_i=h)}(1-V_h)^{\gamma\sum_{i=1}^m 1(z_{ji}>h)}(1-V_h)^{\alpha-1},
\end{aligned}$$

$$\alpha \mid \text{rest} \propto \alpha^{a_\alpha - 1} e^{-b_\alpha \alpha} \alpha^{l^*} \prod_{h=1}^{l^*} (1 - V_d)^{\alpha - 1} \tag{37}$$

for $h = 1, \ldots, l^*$. Let

$$m_{jh} = \frac{\gamma \sum_{i=1}^m 1(z_{ji} = h) x_{ji}}{\gamma \sharp h_j + 1}, \quad v_{jh} = \frac{\sigma_h^2}{\gamma \sharp h_j + 1}, \tag{38}$$

$$a_{jh} = \frac{\gamma \sharp h_j + 1}{2} + a_\sigma, \quad b_{jh} = \frac{\gamma}{2} \sum_{i=1}^m 1(z_{ji} = h) \, (x_{ji} - \mu_h)^2 + \frac{\mu_h^2}{2} + b_\sigma \tag{39}$$

for $h = 1, \ldots, l^*$, then all full conditional densities are tractable in terms of standard distributions:

$$\begin{aligned}
&\mu_{jh} \mid \text{rest} \sim N(m_{jh}, v_{jh}), \quad \sigma_{jh}^2 \mid \text{rest} \sim \text{Inverse-Gamma}(a_{jh}, b_{jh}), \\
&V_{jh} \mid \text{rest} \sim \text{Beta}(1 + \gamma \sum_{i=1}^m 1(z_{ji} = h), \alpha + \gamma \sum_{i=1}^m 1(z_{ji} > h)), \\
&\alpha_{jh} \mid \text{rest} \sim \text{Gamma}(a_\alpha + l^*, b_\alpha - \sum_{h=1}^{l^*} \log(1 - V_{jh})).
\end{aligned} \tag{40}$$

Finally, posterior distribution of the latent variables is

$$z_{ji} \mid \text{rest} \sim \sum_{h=1}^{l^*} p_{jh} \delta_h, \quad p_{jh} = \frac{\nu_{jh} \mathcal{N}(\mu_{jh}, \sigma_{jh}^2)}{\sum_{\tilde{h}=1}^{l^*} \nu_{j\tilde{h}} \mathcal{N}(\mu_{j\tilde{h}}, \sigma_{j\tilde{h}}^2)}, \quad (i = 1, \ldots, m), \tag{41}$$

where $\nu_{jh} = V_{jh} \prod_{l<h} (1 - V_{jl})$ and $\mathcal{N}(m, v)$ is the Gaussian density with mean m and variance v.

Appendix C. Linear Program

$$\begin{aligned}
\underset{\mathbf{a}, T_1, \ldots, T_k}{\text{minimize}} \quad & \sum_{j=1}^k \text{trace}(T_j^T D_j) \\
\text{subject to} \quad & \\
& 0 \le a_i \le 1, \quad i = 1, \ldots, g, \\
& 0 \le (T_j)_{uv} \le 1, \quad u = 1, \ldots, g, \quad v = 1, \ldots, s_j, \quad j = 1, \ldots, k, \\
& \mathbf{1}^T \mathbf{a} = 1, \\
& T_j \, \mathbf{1}_{s_j} = \mathbf{a}, \quad j = 1, \ldots, k, \\
& T_j^T \, \mathbf{1}_s = \frac{\mathbf{1}_{s_j}}{s_j}, \quad j = 1, \ldots, k.
\end{aligned} \tag{42}$$

This linear program can be solved using a variety of linear programming solvers in `Matlab` or `R`, including the algorithms of Cuturi and Doucet (2014) and Srivastava et al. (2015).

References

Martial Agueh and Guillaume Carlier. Barycenters in the Wasserstein space. *SIAM Journal on Mathematical Analysis*, 43(2):904–924, 2011.

Sungjin Ahn, Anoop Korattikara, and Max Welling. Bayesian posterior sampling via stochastic gradient Fisher scoring. *Proceedings of the 29th International Conference on Machine Learning*, 2012.

Pierre Alquier, Nial Friel, Richard Everitt, and Aidan Boland. Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47, 2016.

Pedro C Álvarez-Esteban, E del Barrio, JA Cuesta-Albertos, and C Matrán. A fixed-point approach to barycenters in Wasserstein space. *Journal of Mathematical Analysis and Applications*, 441(2):744–762, 2016.

Ethan Anderes, Steffen Borgwardt, and Jacob Miller. Discrete Wasserstein barycenters: optimal transport for discrete data. *Mathematical Methods of Operations Research*, 84 (2):389409, 2016. ISSN 1432-5217. doi: 10.1007/s00186-016-0549-x. URL `http://dx.doi.org/10.1007/s00186-016-0549-x`.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *Journal of Machine Learning Research*, 18(47):1–43, 2017.

Peter J Bickel and David A Freedman. Some asymptotic theory for the bootstrap. *The Annals of Statistics*, 9(6):1196–1217, 1981.

C. M. Bishop. *Pattern recognition and machine learning*, volume 4. Springer New York, 2006.

Tamara Broderick, Nicholas Boyd, Andre Wibisono, Ashia C Wilson, and Michael Jordan. Streaming variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1727–1735, 2013.

Guillaume Carlier, Adam Oberman, and Edouard Oudet. Numerical methods for matching for teams and Wasserstein barycenters. *ESAIM: Mathematical Modelling and Numerical Analysis*, 49(6):1621–1642, 2015.

Tianqi Chen, Emily Fox, and Carlos Guestrin. Stochastic gradient Hamiltonian Monte Carlo. In *International Conference on Machine Learning*, pages 1683–1691, 2014.

Marco Cuturi and Arnaud Doucet. Fast computation of Wasserstein barycenters. In *Proceedings of the 31st International Conference on Machine Learning, JMLR W&CP*, pages 685–693, 2014.

David B Dunson and Chuanhua Xing. Nonparametric Bayes modeling of multivariate categorical data. *Journal of the American Statistical Association*, 104(487):1042–1051, 2009.

Christel Faes, John T Ormerod, and Matt P Wand. Variational Bayesian inference for parametric and nonparametric regression with missing data. *Journal of the American Statistical Association*, 106(495):959–971, 2012.

Chris Fraley and Adrian E Raftery. Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association*, 97(458):611–631, 2002.

Andrew Gelman, Aki Vehtari, Pasi Jylänki, Christian Robert, Nicolas Chopin, and John P Cunningham. Expectation propagation as a way of life. *arXiv preprint arXiv:1412.4869*, 2014.

Subhashis Ghosal and Aad van der Vaart. Convergence rates of posterior distributions for noniid observations. *The Annals of Statistics*, 35(1):192–223, 2007.

Ryan Giordano, Tamara Broderick, and Michael I Jordan. Covariances, robustness, and variational bayes. *arXiv preprint arXiv:1709.02536*, 2017.

Gurobi Optimization Inc. *Gurobi Optimizer Reference Manual Version 6.0.0*, 2014.

Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. Stochastic variational inference. *Journal of Machine Learning Research*, 14:1303–1347, 2013.

Ildar Abdulovič Ibragimov and Rafail Zalmanovich Has' Minskii. *Statistical Estimation: Asymptotic Theory*, volume 16. Springer Science & Business Media, 2013.

James E. Johndrow, Jonathan C. Mattingly, Sayan Mukherjee, and David B. Dunson. Approximations of Markov chains and High-Dimensional Bayesian Inference. *arXiv preprint arXiv:1508.03387v1*, 2015.

Yoonsang Kim, Young-Ku Choi, and Sherry Emery. Logistic regression with multiple random effects: a simulation study of estimation methods and statistical packages. *The American Statistician*, 67(3):171–182, 2013.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *Proceedings of the 31st International Conference on Machine Learning*, page 181189, 2014.

Alp Kucukelbir, Rajesh Ranganath, Andrew Gelman, and David Blei. Automatic variational inference in Stan. In *Advances in Neural Information Processing Systems*, pages 568–576, 2015.

Shiwei Lan, Bo Zhou, and Babak Shahbaba. Spherical Hamiltonian Monte Carlo for constrained target distributions. In *JMLR workshop and conference proceedings*, volume 32, page 629. NIH Public Access, 2014.

Cathy Yuen Yi Lee and Matt P. Wand. Streamlined mean field variational Bayes for longitudinal and multilevel data analysis. *Biometrical Journal*, 58(4):868–895, 2016. ISSN 1521-4036. doi: 10.1002/bimj.201500007. URL `http://dx.doi.org/10.1002/bimj.201500007`.

Cheng Li, Sanvesh Srivastava, and David B Dunson. Simple, scalable and accurate posterior interval estimation. *Biometrika*, 104:665–680, 2017.

Dougal Maclaurin and Ryan Prescott Adams. Firefly Monte Carlo: Exact MCMC with Subsets of Data. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David Dunson. Scalable and robust Bayesian inference via the median posterior. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1656–1664, 2014.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David B Dunson. Robust and scalable Bayes via a median of subset posterior measures. *The Journal of Machine Learning Research*, 18(1):4488–4527, 2017.

Alexey Miroshnikov and Erin Conlon. *parallelMCMCcombine: Methods for combining independent subset Markov chain Monte Carlo posterior samples to estimate a posterior density given the full data set*, 2014. URL `https://CRAN.R-project.org/package=parallelMCMCcombine`. R package version 1.0.

Willie Neiswanger, Chong Wang, and Eric Xing. Asymptotically exact, embarrassingly parallel MCMC. In *Proceedings of the 30th International Conference on Uncertainty in Artificial Intelligence*, pages 623–632, 2014.

Patrick O Perry. Fast moment-based estimation for hierarchical models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(1):267–291, 2017.

Maxim Rabinovich, Elaine Angelino, and Michael I Jordan. Variational consensus Monte Carlo. In *Advances in Neural Information Processing Systems*, pages 1207–1215, 2015.

Rajesh Ranganath, Dustin Tran, and David Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

Carl Edward Rasmussen and Christopher KI Williams. Gaussian processes for machine learning. *MIT Press*, 2006.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 1530–1538, 2015.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested laplace approximations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 71(2):319–392, 2009.

Steven L. Scott, Alexander W. Blocker, Fernando V. Bonassi, Hugh A. Chipman, Edward I. George, and Robert E. McCulloch. Bayes and big data: the consensus Monte Carlo algorithm. *International Journal of Management Science and Engineering Management*, 11(2):78–88, 2016.

Babak Shahbaba, Shiwei Lan, Wesley O Johnson, and Radford M Neal. Split Hamiltonian Monte Carlo. *Statistics and Computing*, 24(3):339–349, 2014.

Sanvesh Srivastava, Volkan Cevher, Quoc Dinh, and David Dunson. WASP: Scalable Bayes via barycenters of subset posteriors. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, pages 912–920, 2015.

Stan Development Team. Stan: A C++ library for probability and sampling, version 2.5.0, 2014. URL `http://mc-stan.org/`.

Linda SL Tan and David J Nott. Variational inference for generalized linear mixed models using partially noncentered parametrizations. *Statistical Science*, 28(2):168–188, 2013.

Aad W van der Vaart. *Asymptotic Statistics*, volume 3. Cambridge University Press, 2000.

Martin J. Wainwright and Michael I. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learn.*, 1:1–305, January 2008. doi: 10.1561/2200000001. URL `http://portal.acm.org/citation.cfm?id=1498840.1498841`.

Matt Wand. *KernSmooth: Functions for Kernel Smoothing Supporting Wand & Jones (1995)*, 2015. URL `http://CRAN.R-project.org/package=KernSmooth`. R package version 2.23-14.

Xiangyu Wang and David B Dunson. Parallel MCMC via Weierstrass sampler. *arXiv preprint arXiv:1312.4605*, 2013.

Xiangyu Wang, Fangjian Guo, Katherine A Heller, and David B Dunson. Parallelizing MCMC with random partition trees. In *Advances in Neural Information Processing Systems*, pages 451–459, 2015.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient Langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, pages 681–688, 2011.

Wing Hung Wong and Xiaotong Shen. Probability inequalities for likelihood ratios and convergence rates of sieve MLEs. *The Annals of Statistics*, 23(2):339–362, 1995.

Journal of Machine Learning Research 19 (2018) 1-56 Submitted 3/17; Revised 07/18; Published 08/18

Experience Selection in Deep Reinforcement Learning for Control

Tim de Bruin T.D.DEBRUIN@TUDELFT.NL
Jens Kober J.KOBER@TUDELFT.NL
Cognitive Robotics Department
Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands

Karl Tuyls KARLTUYLS@GOOGLE.COM
Deepmind
14 Rue de Londres, 75009 Paris, France

Department of Computer Science
University of Liverpool
Ashton Street, Liverpool L69 3BX, United Kingdom

Robert Babuška R.BABUSKA@TUDELFT.NL
Cognitive Robotics Department
Delft University of Technology
Mekelweg 2, 2628 CD Delft, The Netherlands

Editor: George Konidaris

Abstract

Experience replay is a technique that allows off-policy reinforcement-learning methods to reuse past experiences. The stability and speed of convergence of reinforcement learning, as well as the eventual performance of the learned policy, are strongly dependent on the experiences being replayed. Which experiences are replayed depends on two important choices. The first is which and how many experiences to retain in the experience replay buffer. The second choice is how to sample the experiences that are to be replayed from that buffer. We propose new methods for the combined problem of experience *retention* and experience *sampling*. We refer to the combination as experience *selection*. We focus our investigation specifically on the control of physical systems, such as robots, where exploration is costly. To determine which experiences to keep and which to replay, we investigate different proxies for their immediate and long-term utility. These proxies include age, temporal difference error and the strength of the applied exploration noise. Since no currently available method works in all situations, we propose guidelines for using prior knowledge about the characteristics of the control problem at hand to choose the appropriate experience replay strategy.

Keywords: reinforcement learning, deep learning, experience replay, control, robotics

1. Introduction

Reinforcement learning is a powerful framework that makes it possible to learn complex nonlinear policies for sequential decision making processes while requiring very little prior knowledge. Especially the subfield of deep reinforcement learning, where neural networks are used as function approximators, has recently yielded some impressive results. Among these results are learning to play Atari games (Mnih et al., 2015) and to control robots (Levine et al., 2016) straight from raw images, as well as beating the top human player in the game of Go (Silver et al., 2016).

Reinforcement learning methods can be divided into on-policy and off-policy methods. On-policy methods directly optimize the policy that is used to make decisions, while off-policy methods can learn about an optimal policy from data generated by another policy. Neither approach is without its problems, which has motivated work on methods that combine on and off-policy updates (Wang et al., 2017; Gu et al., 2017; O'Donoghue et al., 2017).

When a reinforcement learning method is either partially or entirely off-policy, past experiences can be stored in a buffer and reused for learning. Doing so not only reduces the sample complexity of the learning algorithm, but can also be crucial for the stability of reinforcement-learning algorithms that use deep neural networks as function approximators (Mnih et al., 2015; Lillicrap et al., 2016; Schaul et al., 2016; Wang et al., 2017).

If we have access to a buffer with past experiences, an interesting question arises: how should we *sample* the experiences to be replayed from this buffer? It has been shown by Schaul et al. (2016) that a good answer to this question can significantly improve the performance of the reinforcement-learning algorithm.

However, even if we know how to sample from the experience buffer, two additional questions arise: what should the buffer capacity be and, once it is full, how do we decide which experiences should be *retained* in the buffer and which ones can be overwritten with new experiences? These questions are especially relevant when learning on systems with a limited storage capacity, for instance when dealing with high-dimensional inputs such as images. Finding a good answer to the question of which experiences to retain in the buffer becomes even more important when exploration is costly. This can be the case for physical systems such as robots, where exploratory actions cause wear or damage and risks need to be minimized (Kober et al., 2013; Garcıa and Fernández, 2015; Tamar et al., 2016; Koryakovskiy et al., 2017). It is also the case for tasks where a minimum level of performance needs to be achieved at all times (Banerjee and Peng, 2004) or when the policy that generates the experiences is out of our control (Seo and Zhang, 2000; Schaal, 1999).

We will refer to the combined problem of experience *retention* and experience *sampling* as experience *selection*. The questions of which experiences to sample and which experiences to retain in the buffer are related, since they both require a judgment on the utility of the experiences. The difference between them is that determining which experiences to sample requires a judgment on the instantaneous utility: from which experiences can the agent learn the most at the moment of sampling? In contrast, a decision on experience retention should be based on the expected long term utility of experiences. Experiences need to be retained in a way that prevents insufficient coverage of the state action space in the future, as experiences cannot be recovered once they have been discarded.

To know the true utility of an experience, it would be necessary to foresee the effects of having the reinforcement-learning agent learn from the experience at any given time. Since this is not possible, we instead investigate *proxies* for the experience utility that are cheap to obtain.

In this work, we investigate age, surprise (in the form of the temporal difference error), and the amplitude of the exploration noise as proxies for the utility of experiences. To motivate the need for *multiple* proxies, we will start by showing the performance of different experience selection methods on control benchmarks that, at first sight, seem very closely related. As a motivating example we show how the current state-of-the-art experience selection method of Schaul et al. (2016), based on retaining a large number of experiences and sampling them according to their temporal difference error, compares on these benchmarks to sampling uniformly at random from the experiences of the most recent episodes. We show that the state-of-the-art method significantly outperforms the standard method on one benchmark while significantly *under*-performing on the other, seemingly similar benchmark.

The focus of this paper is on the control of physical systems such as robots. The hardware limitations of these systems can impose constraints on the exploration policy and the number of experiences that can be stored in the buffer. These factors make the correct choice of experience sampling strategy especially important. As we show on additional, more complex benchmarks, even when sustained exploration *is* possible, it can be beneficial to be selective about which and how many experiences to retain in the buffer. The costs involved in operating a robot mean that it is generally infeasible to rely on an extensive hyper-parameter search to determine which experience selection strategy to use. We therefore want to understand how this choice can be made based on prior knowledge of the control task.

With this in mind, the contributions of this work are twofold:

1. We investigate how the utility of different experiences is influenced by the aspects of the control problem. These aspects include properties of the system dynamics such as the sampling frequency and noise, as well as constraints on the exploration.

2. We describe how to perform experience retention and experience sampling based on experience utility proxies. We show how these two parts of experience selection work together under a range of conditions. Based on this we provide guidelines on how to use prior knowledge about the control problem at hand to choose an experience selection strategy.

Note that for many of the experiments in this work most of the hyper-parameters of the deep reinforcement-learning algorithms are kept fixed. While it would be possible to improve the performance through a more extensive hyper-parameter search, our focus is on showing the relationships between the performance of the different methods and the properties of the control problems. While we do introduce new methods to address specific problems, the intended outcome of this work is to be able to make more informed choices regarding experience selection, rather than to promote any single method.

The rest of this paper is organized as follows. Section 2 gives an overview of related work. In Section 3, the basics of reinforcement learning, as well as the deep reinforcement learning and experience replay methods used as a starting point are discussed. Section 4 gives a high-level overview of the simple benchmarks used in most of this work, with the mathematical details presented in Appendix 9.3. The notation we use to distinguish between different methods, as well as the performance criteria that we use, are discussed in Section 5. In

Section 6, we investigate what spread over the state-action the experiences ideally should have, based on the characteristics of the control problem to be solved. The proposed methods to select experiences are detailed in Section 7, with the results of applying these methods to the different scenarios in simple and more complex benchmarks are presented in Section 8. The conclusions, as well as our recommended guidelines for choosing the buffer size, retention proxy and sampling strategy are given in Section 9.

2. Related Work

When a learning system needs to learn a task from a set of examples, the order in which the examples are presented to the learner can be very important. One method to improve the learning performance on complex tasks is to gradually increase the difficulty of the examples that are presented. This concept is known as shaping (Skinner, 1958) in animal training and curriculum learning (Bengio et al., 2009) in machine learning. Sometimes it is possible to generate training examples of just the right difficulty on-line. Recent machine learning examples of this include generative adversarial networks (Goodfellow et al., 2014) and self play in reinforcement learning (see for example the work by Silver et al. 2017). When the training examples are fixed, learning can be sped up by repeating those examples that the learning system is struggling with more often than those that it finds easy, as was shown for supervised learning by, among others, Hinton (2007) and Loshchilov and Hutter (2015). Additionally, the eventual performance of supervised-learning methods can be improved by re-sampling the training data proportionally to the difficulty of the examples, as done in the boosting technique (Valiant, 1984; Freund et al., 1999)

In on-line reinforcement learning, a set of examples is generally not available to start with. Instead, an agent interacts with its environment and observes a stream of experiences as a result. The experience replay technique was introduced to save those experiences in a buffer and replay them from that buffer to the learning system (Lin, 1992). The introduction of an experience buffer makes it possible to choose which examples should be presented to the learning system again. As in supervised learning, we can replay those experiences that induced the largest error (Schaul et al., 2016). Another option that has been investigated in the literature is to replay more often those experiences that are associated with large immediate rewards (Narasimhan et al., 2015).

In off-policy reinforcement learning the question of which experiences to learn from extends beyond choosing how to sample from a buffer. It begins with determining which experiences should be in the buffer. Lipton et al. (2016) fill the buffer with successful experiences from a pre-existing policy before learning starts. Other authors have investigated criteria to determine which experiences should be retained in a buffer of limited capacity when new experiences are observed. In this context, Pieters and Wiering (2016) have investigated keeping only experiences with the highest immediate rewards in the buffer, while our previous work has focused on ensuring sufficient diversity in the state-action space (de Bruin et al., 2016a,b).

Experience replay techniques, including those in this work, often take the stream of experiences that the agent observes as given and attempt to learn from this stream in an optimal way. Other authors have investigated ways to instill the desire to seek out information that is useful for the learning process directly into the agent's behavior (Schmidhuber,

1991; Chentanez et al., 2004; Houthooft et al., 2016; Bellemare et al., 2016; Osband et al., 2016). Due to the classical exploration-exploitation dilemma, changing the agents behavior to obtain more informative experiences comes at the price of the agent acting less optimally according to the original reward function.

A safer alternative to actively seeking out *real* informative but potentially dangerous experiences is to learn, at least in part, from *synthetic* experiences. This can be done by using an a priori available environment model such as a physics simulator (Barrett et al., 2010; Rusu et al., 2016), or by learning a model from the stream of experiences itself and using that to generate experiences (Sutton, 1991; Kuvayev and Sutton, 1996; Gu et al., 2016; Caarls and Schuitema, 2016). The availability of a generative model still leaves the question of *which* experiences to generate. Prioritized sweeping bases updates again on surprise, as measured by the size of the change to the learned functions (Moore and Atkeson, 1993; Andre et al., 1997). Ciosek and Whiteson (2017) dynamically adjusted the distribution of experiences generated by a simulator to reduce the variance of learning updates.

Learning a model can reduce the sample complexity of a learning algorithm when learning the dynamics and reward functions is easy compared to learning the value function or policy. However, it is not straightforward to get improved performance in general. In contrast, the introduction of an experience replay buffer has shown to be both simple and very beneficial for many deep reinforcement learning techniques (Mnih et al., 2015; Lillicrap et al., 2016; Wang et al., 2017; Gu et al., 2017). When a buffer is used, we can decide which experiences to have in the buffer and which experiences to sample from the buffer. In contrast to previous work on this topic we investigate the combined problem of experience retention and sampling. We also look at several different proxies for the usefulness of experiences and how prior knowledge about the specific reinforcement learning problem at hand can be used to choose between them, rather than attempting to find a single universal experience-utility proxy.

3. Preliminaries

We consider a standard reinforcement learning setting (Section 3.1) in which an agent learns to act optimally in an environment, using the implementation by Lillicrap et al. (2016) of the off-policy actor-critic algorithm by Silver et al. (2014) (Section 3.2). Actor-critic algorithms make it possible to deal with the continuous action spaces that are often found in control applications. The off-policy nature of the algorithm enables the use of experience replay (Section 3.3), which helps to reduce the number of environment steps needed by the algorithm to learn a successful policy and improves the algorithms stability. Here, we summarize the deep reinforcement learning (Lillicrap et al., 2016) and experience replay (Schaul et al., 2016) methods that we use as a starting point.

3.1 Reinforcement Learning

In reinforcement learning, an agent interacts with an environment $\mathcal{E}$ with (normalized) state $s_\mathcal{E}$ by choosing (normalized) actions a according to its policy π: $a = \pi(s)$, where s is the agent's perception of the environment state.

To simplify the analysis in Section 6 and 7, and to aid learning, we normalize the state and action spaces in our benchmarks such that $s_\mathcal{E} \in [-1, 1]^n$ and $a_\mathcal{E} \in [-1, 1]^m$, where n

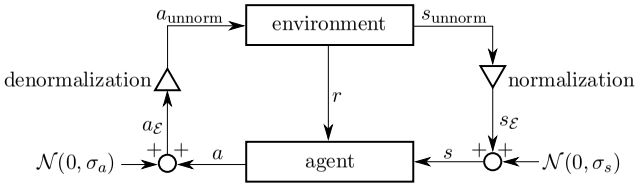


Figure 1: Reinforcement learning scheme and symbols used.

and m are the dimensions of the state and action spaces. We perform the (de)normalization on the connections between the agent and the environment, so the agent only deals with normalized states and actions.

We consider the dynamics of the environment to be deterministic: $s'_\mathcal{E} = f(s_\mathcal{E}, a_\mathcal{E})$. Here, $s'_\mathcal{E}$ is the state of the environment at the next time step after applying action $a_\mathcal{E}$ in state $s_\mathcal{E}$. Although the environment dynamics are deterministic, in some of our experiments we do consider sensor and actuator noise. In these cases, the state s that the agent perceives is perturbed from the actual environment state $s_\mathcal{E}$ by additive Gaussian noise

$$s = s_\mathcal{E} + \mathcal{N}(0, \sigma_s). \tag{1}$$

Similarly, actuator noise changes the actions sent to the environment according to:

$$a_\mathcal{E} = a + \mathcal{N}(0, \sigma_a). \tag{2}$$

A reward function ρ describes the desirability of being in an unnormalized state s_{unnorm} and taking an unnormalized action a_{unnorm}: $r_k = \rho(s^k_{\text{unnorm}}, a^k_{\text{unnorm}}, s^{k+1}_{\text{unnorm}})$, where k indicates the time step. An overview of the different reinforcement learning signals and symbols used is given in Figure 1.

The goal of the agent is to choose the actions that maximize the expected return from the current state, where the return is the discounted sum of future rewards: $\sum_{k=0}^{\infty} \gamma^k r_k$. The discount factor $0 \leq \gamma < 1$ keeps this sum finite and allows trading off short-term and long-term rewards.

Although we will come back to the effect of the sensor and actuator noise later on, in the remainder of this section we will look at the reinforcement learning problem from the perspective of the agent and consider the noise to be part of the environment. This makes the transition dynamics and reward functions stochastic: $\mathcal{F}(s'|s, a)$ and $\mathcal{P}(r|s, a, s')$.

3.2 Off-Policy Deep Actor-Critic Learning

In this paper we use the Deep Deterministic Policy Gradient (DDPG) reinforcement-learning method of Lillicrap et al. (2016), with the exception of Section 6.3, where we compare it to DQN (Mnih et al., 2015). In the DDPG method, based on the work of Silver et al. (2014), a neural network with parameters θ_π implements the policy: $a = \pi(s; \theta_\pi)$. A second neural network with parameters θ_Q, the critic, is used to approximate the Q function. The $Q^\pi(s, a)$ function gives the expected return when taking action a in state s and following

the policy π from next time-step onwards

$$Q^{\pi}(s,a) = \mathbb{E}_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} r_k | s^0 = s, a^0 = a\right]. \tag{3}$$

The critic function $Q(s,a;\theta_Q)$ is trained to approximate the true $Q^{\pi}(s,a)$ function by minimizing the squared temporal difference error δ for experience $\langle s_i, a_i, s'_i, r_i\rangle$

$$\delta_i = \left[r_i + \gamma Q\left(s'_i, \pi(s'_i;\theta^-_{\pi});\theta^-_Q\right)\right] - Q\left(s_i, a_i;\theta_Q\right), \tag{4}$$

$$L_i(\theta_Q) = \delta_i^2,$$

$$\Delta_{i\theta_Q} \sim -\nabla_{\theta_Q} L_i(\theta_Q). \tag{5}$$

The index i is a generic index for experiences that we will in the following use to indicate the index of an experience in a buffer. The parameter vectors θ^-_{π} and θ^-_Q are copies of θ_{π} and θ_Q that are updated with a low-pass filter to slowly track θ_{π} and θ_Q

$$\theta^-_{\pi} \leftarrow (1-\tau)\theta^-_{\pi} + \tau\theta_{\pi},$$
$$\theta^-_Q \leftarrow (1-\tau)\theta^-_Q + \tau\theta_Q,$$

with $\tau \in (0,1), \tau \ll 1$. This was found to be important for ensuring stability when using deep neural networks as function approximators in reinforcement learning (Mnih et al., 2015; Lillicrap et al., 2016).

The parameters θ_{π} of the policy neural network $\pi(s;\theta_{\pi})$ are updated in the direction that changes the action $a = \pi(s;\theta_{\pi})$ in the direction for which the critic predicts the steepest ascent in the expected sum of discounted rewards

$$\Delta_{\theta_{\pi}} \sim \nabla_a Q\left(s_i, \pi(s_i;\theta_{\pi});\theta_Q\right) \nabla_{\theta_{\pi}} \pi(s_i;\theta_{\pi}). \tag{6}$$

3.3 Experience Replay

The actor and critic neural networks are trained by using sample-based estimates of the gradients ∇_{θ_Q} and $\nabla_{\theta_{\pi}}$ in a stochastic gradient optimization algorithm such as ADAM (Kingma and Ba, 2015). These algorithms are based on the assumption of independent and identically distributed (i.i.d.) data. This assumption is violated when the experiences $\langle s_i, a_i, s'_i, r_i\rangle$ in (5) and (6) are used in the same order during the optimization of the networks as they were observed by the agent. This is because the subsequent samples are strongly correlated, since the world only changes slowly over time. To solve this problem, an experience replay (Lin, 1992) buffer $\mathcal{B}$ with some finite capacity $\mathcal{C}$ can be introduced.

Most commonly, experiences are written to this buffer in a First In First Out (FIFO) manner. When experiences are needed to train the neural networks, they are sampled uniformly at random from the buffer. This breaks the temporal correlations of the updates and restores the i.i.d. assumption of the optimization algorithms, which improves their performance (Mnih et al., 2015; Montavon et al., 2012). The increased stability comes in addition to the main advantage of experience replay, which is that experiences can be used multiple times for updates, increasing the sample efficiency of the algorithm.

3.3.1 Prioritized Experience Replay

Although sampling experiences uniformly at random from the experience buffer is an easy default, the performance of reinforcement-learning algorithms can be improved by choosing the experience samples used for training in a smarter way. Here, we summarize one of the variants of Prioritized Experience Replay (PER) that was introduced by Schaul et al. (2016). Our enhancements to experience replay are given in Section 7.

The PER technique is based on the idea that the temporal difference error (4) provides a good proxy for the instantaneous utility of an experience. Schaul et al. (2016) argue that, when the critic made a large error on an experience the last time it was used in an update, there is more to be learned from the experience. Therefore, its probability of being sampled again should be higher than that of an experience associated with a low temporal difference error.

In this work we consider the rank-based stochastic PER variant. In this method, the probability of sampling an experience i from the buffer is approximately given by:

$$P(i) \approx \frac{\left(\frac{1}{\text{rank}(i)}\right)^{\alpha}}{\sum_j \left(\frac{1}{\text{rank}(j)}\right)^{\alpha}}. \tag{7}$$

Here, $\text{rank}(i)$ is the rank of sample i according to the absolute value of the temporal difference error $|\delta|$ according to (4), calculated when the experience was last used to train the critic. All experiences that have not yet been used for training have $\delta = \infty$, resulting in a large probability of being sampled. The parameter α determines how strongly the probability of sampling an experience depends on δ. We use $\alpha = 0.7$ as proposed by Schaul et al. (2016) and have included a sensitivity analysis for different buffer sizes in Appendix 9.3. Note that the relation is only approximate as sampling from this probability distribution directly is inefficient. For efficient sampling, (7) is used to divide the buffer $\mathcal{B}$ into S segments of equal cumulative probability, where S is taken as the number of experiences per training mini batch. During training, one experience is sampled uniformly at random from each of the segments.

3.3.2 Importance Sampling

The estimation of an expected value with stochastic updates relies on those updates corresponding to the same distribution as its expectation. Schaul et al. (2016) proposed to compensate for the fact that the changed sampling procedure can affect the value of the expectation in (3) by multiplying the gradients (5) with an Importance Sampling (IS) weight

$$\omega_i = \left(\frac{1}{\mathcal{C}}\frac{1}{P(i)}\right)^{\beta}. \tag{8}$$

Here, β allows scaling between not compensating at all ($\beta = 0$) to fully compensating for the changes in the sample distribution caused by the sampling strategy ($\beta = 1$). In our experiments, when IS is used, we follow Schaul et al. (2016) in scaling β linearly per episode from 0.5 at the start of a learning run to $\beta = 1$ at the end of the learning run. $\mathcal{C}$ indicates the capacity of the buffer.

Not all changes to the sampling distribution need to be compensated for. Since we use a deterministic policy gradient algorithm with a Q-learning critic, we do not need to compensate for the fact that the samples are obtained by a different policy than the one we are optimizing for (Silver et al., 2014). We can change the sampling distribution from the buffer, without compensating for the change, so long as these samples accurately represent the transition and reward functions.

Sampling based on the TD error can cause issues here, as infrequently occurring transitions or rewards will tend to be surprising. Replaying these samples more often will introduce a bias, which should be corrected through importance sampling.

However, the temporal difference error will also be partly caused by the function approximation error. These errors will be present even for a stationary sample distribution after learning has converged. The errors will vary over the state-action space and their magnitude will be related to the sample density. Sampling based on *this* part of the temporal difference error will make the function approximation accuracy more consistent over the state-space. This effect might be unwanted when the learned controller will be tested on the same initial state distribution as it was trained on. In that case, it is preferable to have the function approximation accuracy be highest where the sample density is highest. However, when the aim is to train a controller that generalizes to a larger part of the state space, we might *not* want to use importance sampling to correct this effect. Note that importance sampling based on the sample distribution over the state space is heuristically motivated and based on function approximation considerations. The motivation does not stem from the reinforcement learning theory, where most methods assume that the Markov decision process is ergodic and that the initial state distribution does not factor into the optimal policy (Aslanides et al., 2017). In practice however, deep reinforcement-learning methods can be rather sensitive to the initial state distribution (Rajeswaran et al., 2017).

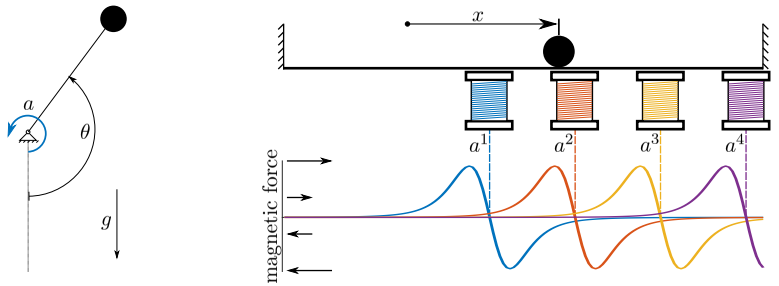
Unfortunately, we do not know to what extent the temporal difference error is caused by the stochasticity of the environment dynamics and to what extent it is caused by function approximation errors. We will empirically investigate the use of importance sampling in Section 8.4.

4. Experimental Benchmarks

In this section, we discuss two relatively simple control tasks that are considered in this paper, so that an understanding of their properties can be used in the following sections. The relative simplicity of these tasks enables a thorough analysis. We test our findings on more challenging benchmarks in Section 8.5.

We perform our tests on two simulated control benchmarks: a pendulum swing-up task and a magnetic manipulation problem. Both were previously discussed by Alibekov et al. (2018). Although both represent dynamical systems with a two dimensional state-space, it will be shown in Section 6 that they are quite different when it comes to the optimal experience selection strategy. Here, a high level description of these benchmarks is presented, with the full mathematical description given in Appendix 9.3.

The first task is the classic under-actuated pendulum swing-up problem, shown in Figure 2a. The pendulum starts out hanging down under gravity. The goal is to balance the pendulum in the upright position. The motor is torque limited such that a swing to one



(a) Pendulum task (b) Magnetic manipulation task

Figure 2: The two benchmark problems considered in this paper. In the pendulum task, an underactuated pendulum needs to be swung up and balanced in the upright position by controlling the torque applied by a motor. In the magnetic manipulation (magman) task, a steel ball (top) needs to be positioned by controlling the currents through four electromagnets. The magnetic forces exerted on the ball are shown at the bottom of the figure and can be seen to be a nonlinear function of the position. The forces scale linearly with the actions $a^1, ..., a^4$, which represent the squared currents through the magnets.

side is needed to build up momentum before swinging towards the upright position in the opposite direction. Once the pendulum is upright it needs to stabilize around this unstable equilibrium point. The state of the problem $s_\mathcal{E}$ consists of *normalized* versions of the angle θ and angular velocity $\dot{\theta}$ of the pendulum. The action space is a normalized version of the voltage applied to the motor that applies a torque to the pendulum. A reward is given at every time-step, based on the absolute distance of the state from the reference state of being upright with no rotational velocity.

The second benchmark is a magnetic manipulation (*magman*) task, in which the goal is to accurately position a steel ball on a 1-D track by dynamically changing a magnetic field. The relative magnitude and direction of the force that each magnet exerts on the ball is shown in Figure 2b. This force is linearly dependent on the actions, which represent the squared currents through the electromagnet coils. Normalized versions of the position x and velocity $\dot{x}$ form the state-space of the problem. A reward is given at every time-step, based on the absolute distance of the state from the reference state of having the ball at the fixed desired position.

In experiments where the buffer capacity $\mathcal{C}$ is limited, we take $\mathcal{C} = 10^4$ experiences, unless stated otherwise. All our experiments have episodes which last four seconds. Unless stated otherwise, a sampling frequency of 50 Hz is used, which means the buffer can store 50 episodes of experience tuples $\langle s_i, a_i, s'_i, r_i \rangle$.

Since we are especially interested in physical control problems where sustained exhaustive exploration is infeasible, the amount of exploration is reduced over time from its max-

imum at episode 1, to a minimum level from episode 500 onwards in all our experiments. At the minimum level, the amplitude of the exploration noise we add to the neural network policy is 10% of the amplitude at episode 1. Details of the exploration strategies used are given in Appendix 9.3.

5. Performance Measures and Experience Selection Notation

This section introduces the performance measures used and the notation used to distinguish between the experience selection strategies.

5.1 Performance Measures

When we investigate the performance of the learning methods in Sections 6 and 8, we are interested in the effect that these methods might have on *three* aspects of the learning performance: *the learning stability, the maximum controller performance* and *the learning speed*. We define performance metrics for these aspects, related to the normalized mean reward per episode μ_r. The normalization is performed such that $\mu_r = 0$ is the performance achieved by a random controller, while $\mu_r = 1$ is the performance of the off-line dynamic programming method described in Appendix 9.3. This baseline method is, at least for the noise-free tests, proven to be close to optimal.

The first learning performance aspect we consider is the *stability* of the learning process. As we have discussed in previous work (de Bruin et al., 2015, 2016a), even when a good policy has already been learned, the learning process can become unstable and the performance can drop significantly when the properties of the training data change. We investigate to what extent different experience replay methods can help prevent this instability. We use the mean of μ_r over the last 100 episodes of each learning run, where the learning runs should have converged to good behavior already, as a measure of learning stability. We denote this measure by μ_r^{final}.

Although changing the data distribution might help stability, it could at the same time prevent us from accurately approximating the true optimal policy. Therefore we also report the *maximum performance* achieved per learning trial $\mu_r^{\max}$.

Finally, we want to know the effects of the experience selection methods on the *learning speed*. We therefore report the number of episodes before the learning method achieves a normalized mean reward per episode of $\mu_r = 0.8$ and denote this by Rise-time 0.8.

For these performance metrics we report the means and the 95% confidence bounds of those means over 50 trials for each experiment. The confidence bounds are based on bootstrapping (Efron, 1992).

5.2 Experience Selection Strategy Notation

We consider the problem of *experience selection*, which we have defined as the combination of *experience retention* and *experience sampling*. The experience retention strategy determines which experiences are discarded when new experiences are available to a full buffer. The sampling strategy determines which experiences are used in the updates of the reinforcement-learning algorithm. We use the following notation for the complete experience selection strategy: *retention strategy[sampling strategy]*. Our abbreviations for the retention

Notation	Proxy	Explanation
FIFO	age	The oldest experiences are overwritten with new ones.
FULL DB	-	The buffer capacity $\mathcal{C}$ is chosen to be large enough to retain all experiences.

Table 1: Commonly used experience retention strategies for deep reinforcement learning.

Notation	Proxy	Explanation
Uniform	-	Experiences are sampled uniformly at random.
PER	surprise	Experiences are sampled using rank-based stochastic *prioritized experience replay* based on the temporal difference error. See Section 3.3.1.
PER+IS	surprise	Sampling as above, but with weighted *importance sampling* to compensate for the distribution changes caused by the sampling procedure. See Section 3.3.2.

Table 2: Experience sampling strategies from the literature.

and sampling strategies commonly used in deep RL that were introduced in Section 3.3 are given in Tables 1 and 2 respectively. The abbreviations used for the new or uncommonly used methods introduced in Section 7 are given there, in Tables 4 and 5.

6. Analysis of Experience Utility

As previously noted by Schaul et al. (2016); Narasimhan et al. (2015); Pieters and Wiering (2016) and de Bruin et al. (2016a, 2015), when using experience replay, the criterion that determines which experiences are used to train the reinforcement learning agent can have a large impact on the performance of the method. The aim of this section is to investigate what makes an experience useful and how this usefulness depends on several identifiable characteristics of the control problem at hand.

In the following sections, we mention only some relevant aspects of our implementation of the deep reinforcement-learning methods, with more details given in Appendix 9.3.

6.1 The Limitations of a Single Proxy

To motivate the need for understanding how the properties of a control problem influence the applicability of different experience selection strategies, and the need for multiple proxies for the utility of experiences rather than one universal proxy, we compare the performance of the two strategies from the literature that were presented in Section 3.3 on the benchmarks described in Section 4.

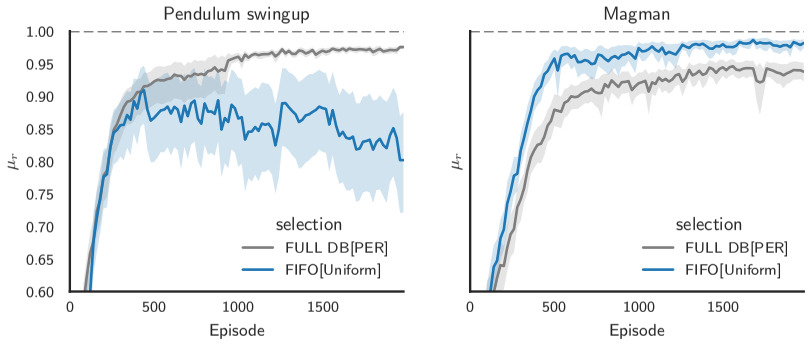


Figure 3: Comparison of the state-of-the-art (FULL DB[PER]) and the default method (FIFO[Uniform]) for experience selection on our two benchmark problems.

The first experience selection strategy tested is FIFO[Uniform]: overwriting the oldest experiences when the buffer is full and sampling uniformly at random from the buffer. We compare this strategy to the state-of-the-art prioritized experience replay method FULL DB[PER] by Schaul et al. (2016). Here, the buffer capacity $\mathcal{C}$ is chosen such that all experiences are retained during the entire learning run ($\mathcal{C} = N = 4 \times 10^5$ for this test).[1] The sampling strategy is the rank-based stochastic prioritized experience replay strategy as described in Section 3.3. The results of the experiments are shown in Figure 3.

Figure 3 shows that FULL DB[PER] method, which samples training batches based on the temporal difference error from a buffer that is large enough to contain all previous experiences, works well for the pendulum swing-up task. The method very reliably finds a near optimal policy. The FIFO[Uniform] method, which keeps only the experiences from the last 50 episodes in memory, performs much worse. As we reported previously (de Bruin et al., 2016a), the performance degrades over time as the amount of exploration is reduced and the experiences in the buffer fail to cover the state-action space sufficiently.

If we look at the result on the magman benchmark in Figure 3, the situation is reversed. Compared to simply sampling uniformly from the most recent experiences, sampling from all previous experiences according to their temporal difference error limits the final performance significantly. As shown in Appendix 9.3, this is not simply a matter of the function approximator capacity, as even much larger networks trained on all available data are outperformed by small networks trained on only recent data. When choosing an experience selection strategy for a reinforcement learning task, it seems therefore important to have some insights into how the characteristics of the task determine the need for specific kinds of experiences during training. We will investigate some of these characteristics below.

1. Schaul et al. (2016) use a FIFO database with a capacity of 10^6 experiences. We here denote this as FULL DB since all our experiments use a smaller number of time-steps in total.

6.2 Generalizability and Sample Diversity

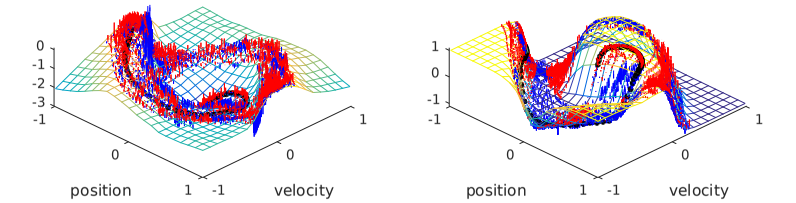
One important aspect of the problem, which at least partly explains the differences in performance for the two methods on the two benchmarks in Figure 3, is the complexity of generalizing the value function and policy across the state and action spaces.

For the pendulum task, learning actor and critic functions that generalize across the entire state and action spaces will be relatively simple as a sufficiently deep neural network can efficiently exploit the symmetry in the value and policy functions (Montufar et al., 2014). Figure 4b shows the learned policy after 100 episodes for a learning run with FIFO[Uniform] experience selection. Due to the thorough initial exploration, the experiences in the buffer cover much of the state-action space. As a result, a policy has been learned that is capable of swinging the pendulum up and stabilizing it in both the clockwise and anticlockwise directions, although the current policy favors one direction over the other.

For the next 300 episodes this favored direction does not change and as the amount of exploration is decayed, the experiences in the buffer become less diverse and more centered around this favored trajectory through the state-action space. Even though the information on how to further improve the policy becomes increasingly local, the updates to the network parameters can cause the policy to be changed over the whole state space, as neural networks are global function approximators. This can be seen from Figure 4d, where the updates that further refine the policy for swinging up in the currently preferred direction have removed the previously obtained skill of swinging up in the opposite direction. The policy has suffered from *catastrophic forgetting* (Goodfellow et al., 2013) and has over-fitted to the currently preferred swing up direction.

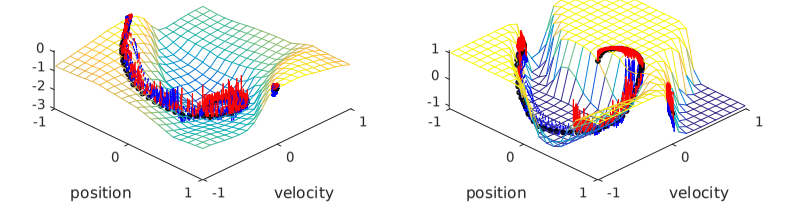
For the pendulum swing up task, this over-fitting is particularly risky since the preferred swing up direction can and does change during learning, since both directions are equivalent with respect to the reward function. When this happens, the FIFO experience retention method can cause the data distribution in the buffer to change rapidly, which by itself can cause instability. In addition, the updates (4) and (6) now use the critic $Q(s, a; \theta_Q)$ function in regions of the state-action space that it has not been trained on in a while, resulting in potentially bad gradients. Both of these factors might destabilize the learning process. This can be seen in Figure 4f where, after the preferred swing up direction has rapidly changed a few times, the learning process is destabilized and the policy has deteriorated to the point that it no longer accomplishes the balancing task. By keeping all experiences in memory and ensuring the critic error δ stays low over the entire state-action space, the FULL DB[PER] method largely avoids these learning stability issues. We believe that this accounts for the much better performance for this benchmark shown in Figure 3.

For the magman task, a policy that generalizes over the whole state-space might be harder to find. This is because the effects of the actions, shown as the colored lines in Figure 2b, are strongly nonlinear functions of the (position)-state. The actor and critic functions must therefore be very accurate for the states that are visited under the policy. Requiring the critic to explain all of the experiences that have been collected so far might limit the ability of the function approximators to achieve sufficient accuracy for the relevant states.



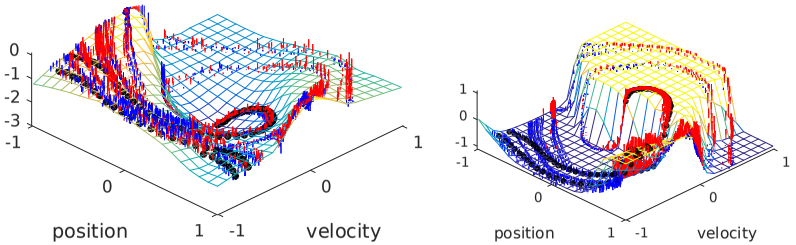
(a) critic, episode 100

(b) actor, episode 100



(c) critic, episode 390

(d) actor, episode 390



(e) critic, episode 507

(f) actor, episode 507

Figure 4: The critic $Q(s, \pi(s;\theta_\pi);\theta_Q)$ and actor $\pi(s;\theta_\pi)$ functions trained on the pendulum swing up task using FIFO[Uniform] experience selection. The surfaces represent the functions. The black dots show the trajectories through the state-action space resulting from deterministically following the current policy. The red and blue lines show respectively the positive and negative 'forces' that shape the surfaces caused by the experiences in the buffer: for the critic these are $\delta(s,a)$ (note $a \neq \pi(s;\theta_\pi)$). For the actor these forces represent $\partial Q(s, \pi(s;\theta_\pi);\theta_Q)/\partial a$. Animations of these graphs for different experience selection strategies are available at `https://youtu.be/Hli1kyObgT4`. The episodes are chosen to illustrate the effect of reduced sample diversity described in Section 6.2.

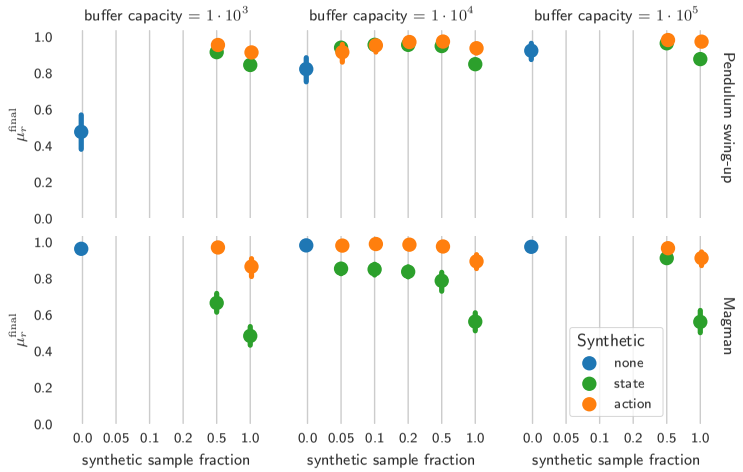
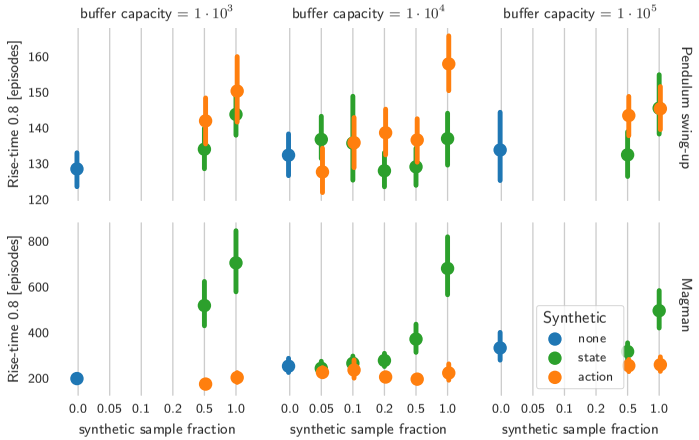


Figure 5: The effect on the mean performance during the last 100 episodes of the learning runs μ_r^{final} of the FIFO[Uniform] method when changing a fraction of the observed experiences with synthetic experiences, for different buffer sizes.

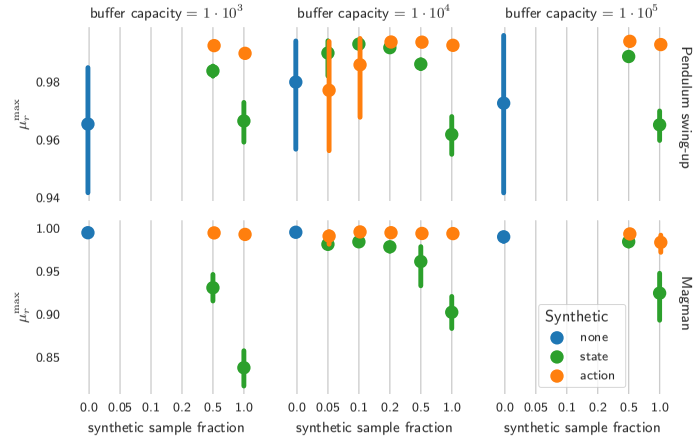
6.2.1 Buffer Size and Synthetic Sample Fraction

To test the hypothesis that the differences in performance observed in Figure 3 revolve around sample diversity, we will artificially alter the sample diversity and investigate how this affects the reinforcement learning performance. We will do so by performing the following experiment. We use the plain FIFO[Uniform] method as a baseline. However, with a certain probability we make a change to an experience $\langle s_i, a_i, s_i', r_i \rangle$ before it is written to the buffer. We change either the state s_i or the action a_i. The changed states and actions are sampled uniformly at random from the state and action spaces. When the state is re-sampled the action is recalculated as the policy action for the new state including exploration. In both cases, the next state and reward are recalculated to complete the altered experience. To calculate the next state and reward, we use the real system model. This is not possible for most practical problems; it serves here merely to gain a better understanding of the need for sample diversity.

The results of performing this experiment for different probabilities and buffer sizes are given in Figures 5 and 6. Interestingly, for the pendulum swing up task, changing some fraction of the experiences to be more diverse improves the stability of the learning method dramatically, regardless of whether the diversity is in the states or in the actions. The effect is especially noticeable for smaller experience buffers.



(a) Effect on the number of episodes needed to reach $\mu_r = 0.8$.



(b) Effect on the number of maximum controller performance obtained per learning run.

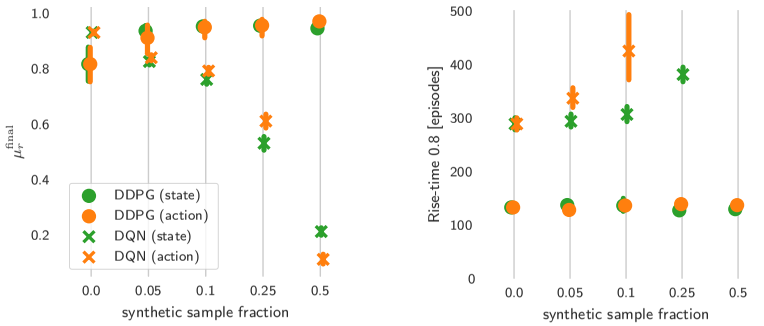
Figure 6: The effects on the learning performance of the FIFO[Uniform] method when replacing a fraction of the observed experiences with synthetic experiences, for different buffer sizes.

For the magman benchmark, as expected, having more diverse states reduces the performance significantly. Having a carefully chosen fraction of more diverse actions in the original states can however improve the stability and learning speed slightly. This can be explained from the fact that even though the effects of the actions are strongly nonlinear in the state-space, they are linear in the action space. Generalizing across the action space might thus be more straightforward and it is helped by having the training data spread out over this domain.

6.3 Reinforcement-Learning Algorithm

The need for experience diversity also depends on the algorithm that is used to learn from those experiences. In the rest of this work we exclusively consider the DDPG actor-critic algorithm, as the explicitly parameterized policy enables continuous actions, which makes it especially suitable for control. An alternative to using continuous actions is to discretize the action space. In this subsection, we compare the need for diverse data of the actor-critic DDPG algorithm (Lillicrap et al., 2016; Silver et al., 2014) to that of the closely related critic-only DQN algorithm (Mnih et al., 2015). The experiments are performed on the pendulum benchmark, where the one dimensional action is divided uniformly into 15 discrete actions. Results for the magman benchmark are omitted as the four dimensional action space makes discretization impractical.

For the actor-critic scheme to work, the critic needs to learn a general dependency of the Q-values on the states and actions. For the DQN critic, this is not the case as the Q-values for different actions are separate. Although the processing of the inputs is shared, the algorithm can learn at least partially independent value predictions for the different



(a) Effect on the mean performance during the last 100 episodes of the learning runs μ_r^{final}.

(b) Effect on the number of episodes needed to reach $\mu_r = 0.8$.

Figure 7: RL algorithm dependent effect of adding synthetic experiences to the FIFO[Uniform] method. Experiments on the pendulum benchmark. The effect on $\mu_r^{\max}$ is given in Figure 22 in Appendix 9.3.

actions. These functions additionally do not need to be correct, as long as the optimal action in a state has a higher value than the sub-optimal actions.

These effects can be seen in Figure 7. The DDPG algorithm can make more efficient use of the state-action space samples by learning a single value prediction, resulting in significantly faster learning than the DQN algorithm. The DDPG algorithm additionally benefits from more diverse samples, with the performance improving for higher fractions of randomly sampled states or actions. The DQN algorithm conversely seems to suffer from a more uniform sampling of the state-action space. This could be because it is now tasked with learning accurate mappings from the states to the state-action values for all actions. While doing so might not help to improve the predictions in the relevant parts of the state-action space, it could increase the time required to learn the function and limit the function approximation capacity available for those parts of the state-space where the values need to be accurate. Note again that learning precise Q-values for all actions over the whole state-space is not needed, as long as the optimal action has the largest Q-value.

Due to the better scalability of policy-gradient methods in continuous control settings, we exclusively consider the DDPG algorithm in the remainder of this work.

6.3.1 Sample Age

In the model-free setting it is not possible to add synthetic experiences to the buffer. Instead, in Section 7 we will introduce ways to select real experiences that have desirable properties and should be remembered for a longer time and replayed more often. This will inevitably mean that some experiences are used more often than others, which could have detrimental effects such as that the learning agent could over-fit to those particular experiences.

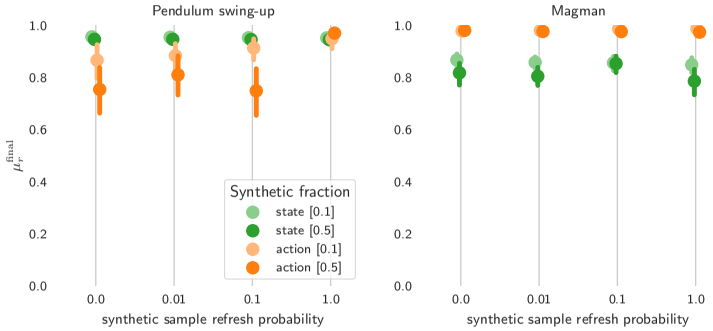


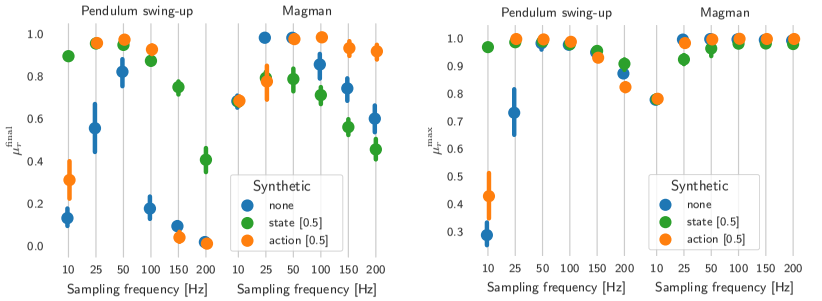
Figure 8: The effects on μ_r^{final} of the FIFO[Uniform] method when changing a fraction of the observed experiences with synthetic experiences, when the synthetic experiences are updated only with a certain probability each time they are overwritten. The effects on μ_r^{max} and the rise-time are given in Figure 24 in Appendix 9.3.

To investigate the effects of adding older experiences for diversity, we perform the following experiment. As before, a FIFO buffer is used with a certain fraction of synthetic experiences. However, when a synthetic experience is about to be over-written, we only sample a new synthetic experience with a certain probability. Otherwise, the experience is left unchanged. The result of this experiment is shown in Figure 8. For the pendulum benchmark, old experiences only hurt when they were added to provide diversity in the action space in states that were visited by an older policy. For the magman benchmark the age of the synthetic experiences is not seen to affect the learning performance.

6.4 Sampling Frequency

An important property of control problems that can influence the need for experience diversity is the frequency at which the agent needs to produce control decisions. The sampling frequency of a task is something that is often considered a given property of the environment in reinforcement learning. For control tasks however, a sufficiently high sampling frequency can be crucial for the performance of the controller and for disturbance rejection (Franklin et al., 1998). At the same time, higher sampling frequencies can make reinforcement learning more difficult as the effect of taking an action for a single time-step diminishes for increasing sampling frequencies (Baird, 1994). Since the sampling rate can be an important hyperparameter to choose, we investigate whether changing it changes the diversity demands for the experiences to be replayed.

In Figure 9, the performance of the FIFO[Uniform] method is shown for different sampling frequencies, with and without synthetic samples. The first thing to note is that, as expected, low sampling frequencies limit the controller performance. Interestingly, much of the performance loss on the pendulum at low frequencies can be prevented through in-



(a) Effect on the mean performance during the last 100 episodes of the learning runs μ_r^{final}.

(b) Effect on maximum controller performance per episode μ_r^{max}.

Figure 9: Sampling frequency dependent effect of adding synthetic experiences to the FIFO[Uniform] method. The effect on the rise time is given in Figure 23 in Appendix 9.3.

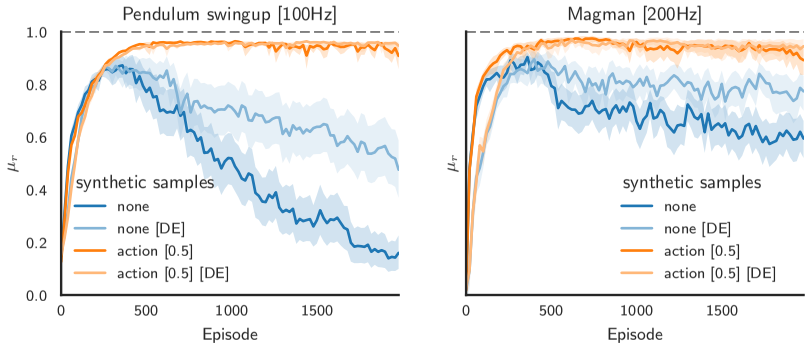


Figure 10: The effect of synthetic actions and stochastically preventing experiences from being written to the buffer [DE] for the FIFO[Uniform] method on the benchmarks with increased sampling frequencies.

creased sample diversity. This indicates that on this benchmark most of the performance loss at the tested control frequencies stems from the learning process rather than the fundamental control limitations. When increasing the sampling frequencies beyond our baseline frequency of 50Hz, sample diversity becomes more important for both stability and performance. For the pendulum swing-up it can be seen that as sampling frequency increases further, increased diversity in the *state-space* becomes more important. For the magman, adding synthetic *action* samples has clear benefits. This is very likely related to the idea that the effects of actions become harder to distinguish for higher sampling frequencies (Baird, 1994; de Bruin et al., 2016b).

There are several possible additional causes for the performance decrease at higher frequencies. The first is that by increasing the sampling frequency, we have increased the number of data points that are obtained and learned from per episode. Yet the amount of information that the data contains has not increased by the same amount. Since the buffer capacity is kept equal, the amount of information that the buffer contains has decreased and the learning rate has effectively increased. To compensate for these specific effects, experiments are performed in which samples are stochastically prevented from being written to the buffer with a probability proportional to the increase in sampling frequency. The results of these experiments are indicated with [DE] (dropped experiences) in Figure 10 and are indeed better, but still worse than the performance for lower sampling frequencies.

The second potential reason for the drop in performance is that we have changed the problem definition by changing the sampling frequency. This is because the forgetting factor γ determines how far into the future we consider the effects of our actions according to:

$$\gamma = e^{-\frac{T_s}{\tau_\gamma}},$$

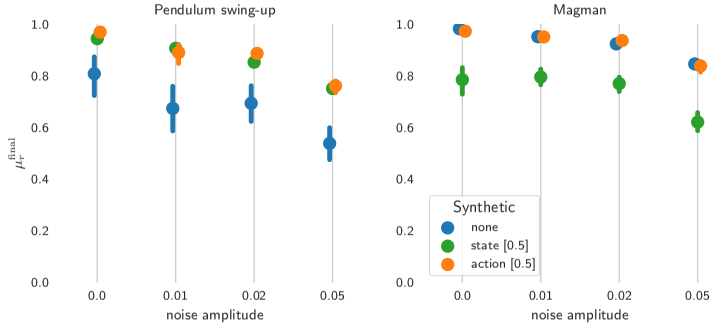


Figure 11: Experiments with altered experiences and sensor and actuator noise. Results are from the last 100 episodes of 50 learning runs. A description of the performance measures is given in Section 5.1.

where T_s is the sampling period in seconds and τ_γ is the lookahead horizon in seconds. To keep the same lookahead horizon, we recalculate γ, which is 0.95 in our other experiments ($T_s = 0.02$), to be $\gamma_{\text{pendulum}} = 0.9747$ ($T_s = 0.01$) and $\gamma_{\text{magman}} = 0.9873$ ($T_s = 0.005$). To keep the scale of the Q functions the same, which prevents larger gradients, the rewards are scaled down. Correcting the lookahead horizon was found to hurt performance on both benchmarks. The likely cause of this is that higher values of γ increase the dependence on the biased estimation of Q over the unbiased immediate reward signal (see Equation (4)). This can cause instability (François-Lavet et al., 2015).

6.5 Noise

The final environment property that we consider is the presence of sensor and actuator noise. So far, the agent has perceived the (normalized) environment state exactly and its (de-normalized) chosen actions have been implemented without change. Now we consider Equations (1) and (2) with $\sigma_s = \sigma_a \in \{0, 0.01, 0.02, 0.05\}$. The results of performing these experiments are shown in Figure 11. The results indicate that the need for data diversity is not dependent on the presence of noise. However, in Section 8.3 it will be shown that the methods used to determine which experiences are useful *can* be affected by noise.

6.6 Summary

This section has presented an investigation into how different aspects of the reinforcement learning problem at hand influence the need for experience diversity. In Table 3 a summary is given of the investigated aspects and the strength of their effect on the need for experience diversity. While this section has used the true environment model to examine the potential

Property	Effect	Explanation
Benchmark	Very high	The need for diverse states and actions largely depends on the ease and importance of generalizing across the state-actions space, which is benchmark dependent.
RL algorithm	Very high	Generalizing across the action space is fundamental to actor-critic algorithms, but not to critic-only algorithms with discrete action spaces.
Sampling frequency	High	The stability of RL algorithms depends heavily on the sampling frequency. Experience diversity can help learning stability. Having diverse actions at higher frequencies might be crucial as the size of their effect on the observed returns diminishes.
Buffer size	Medium	Small buffers can lead to rapidly changing data distributions, which causes unstable learning. Large buffers have more inherent diversity.
Sample age	Low	Although retaining old samples could theoretically be problematic, these problems were not clearly observable in practice.
Noise	None	The presence of noise was not observed to influence the need for experience diversity, although it can influence experience selection strategies, as will be shown in Section 8.3.

Table 3: The dependence of the need for diverse experiences on the investigated environment and reinforcement learning properties.

benefits of diversity, the next section will propose strategies to obtain diverse experiences in ways that are feasible on real problems.

7. New Experience-Selection Strategies

For the reasons discussed in Section 2, we do not consider changing the stream of experiences that an agent observes by either changing the exploration or by generating synthetic experiences. Instead, to be able to replay experiences with desired properties, valuable experiences need to be identified, so that they can be retained in the buffer and replayed from it. In this section we look at how several proxies for the utility of experiences can be used in experience selection methods.

7.1 Experience Retention

Although we showed in Section 6.4 that high sampling rates might warrant dropping experiences, in general we assume that each new experience has at least some utility. Therefore, unless stated otherwise, we will always write newly obtained experiences to the buffer. When the buffer is full, this means that we need some metric that can be used to decide which experiences should be overwritten.

7.1.1 Experience Utility Proxies

A criterion used to manage the contents of an experience replay buffer should be cheap enough to calculate,[2] should be a good proxy for the usefulness of the experiences and should not depend on the learning process in a way that would cause a feedback loop and possibly might destabilize that learning process. We consider three criteria for overwriting experiences.

Age: The default and simplest criterion is age. Since the policy is constantly changing and we are trying to learn its current effects, recent experiences might be more relevant than older ones. This (FIFO) criterion is computationally as cheap as it gets, since determining which experience to overwrite involves simply incrementing a buffer index. For smaller buffers, this does however make the buffer contents quite sensitive to the learning process, as a changing policy can quickly change the distribution of the experiences in the buffer. As seen in Figure 4, this can lead to instability.

Besides FIFO, we also consider reservoir sampling (Vitter, 1985). When the buffer is full, new experiences are added to it with a probability $\mathcal{C}/i$ where i is the index of the current experience. If the experience is written to the buffer, the experience it replaces is chosen uniformly at random. Note that this is the only retention strategy we consider that does not write all new experiences to the buffer. Reservoir sampling ensures that at every stage of learning, each experience observed so far has an equal probability of being in the buffer. As such, initial exploratory samples are kept in memory and the data distribution converges over time. These properties are shared with the FULL DB strategy, without needing the same amount of memory. The method might in some cases even improve the learning stability compared to using a full buffer, as the data distribution converges faster. However, when the buffer is too small this convergence can be premature, resulting in a buffer that does not adequately reflect the policy distribution. This can seriously compromise the learning performance.

Surprise: Another possible criterion is the unexpectedness of the experience, as measured by the temporal difference error δ from (4). The success of the Prioritized Experience Replay (PER) method of Schaul et al. (2016) shows that this can be a good proxy for the utility of experiences. Since the values have to be calculated to update the critic, the computational cost is very small if we accept that the utility values might not be

2. We have discussed the need for experience diversity in Section 6 and we have previously proposed overwriting a buffer in a way that directly optimized for diversity (de Bruin et al., 2016a). However, calculating the experience density in the state-action space is very expensive and therefore prohibits using the method on anything but small-scale problems.

current since they are only updated for experiences that are sampled. The criterion is however strongly linked with the learning process, as we are actively trying to minimize δ. This means that, when the critic is able to accurately predict the long term rewards of the policy in a certain region of the state-action space, these samples can be overwritten. If the predictions of the critic later become worse in this region, there is no way of getting these samples back. An additional problem might be that the error according to (4) will be caused partially by state and actuator noise. Keeping experiences for which the temporal difference error is high might therefore cause the samples saved in the buffer to be more noisy than necessary.

Exploration: We introduce a new criterion based on the observation that problems can occur when the amount of exploration is reduced. On physical systems that are susceptible to damage or wear, or for tasks where adequate performance is required even during training, exploration can be costly. This means that preventing the problems caused by insufficiently diverse experiences observed in Section 6 simply by sustained thorough exploration might not be an option. We therefore view the amount of exploration performed during an experience as a proxy for its usefulness. We take the 1-norm of the deviation from the policy action to be the usefulness metric. In our experiments on the small scale benchmarks we follow the original DDPG paper (Lillicrap et al., 2016) in using an Ornstein-Uhlenbeck noise process added to the output of the policy network. The details of the implementation are given in Appendix 9.3. In the experiments in Section 8.5 a copy of the policy network with noise added to the parameters is used to calculate the exploratory actions (Plappert et al., 2018).

For discrete actions, the cost of taking exploratory actions could be used as a measure of experience utility as well. The inverse of the probability of taking an action could be seen as a measure of the cost of the action. It could also be worth investigating the use of a low-pass filter, as a series of (semi)consecutive exploratory actions would be more likely to result in states that differ from the policy distribution in a meaningful way. These ideas are not tested here, as we only consider continuous actions in the remainder of this work.

Note that the size of the exploration signal is the deviation of the chosen action in a certain state from the policy action for that state. Since the policy evolves over time we could recalculate this measure of deviation from the policy actions per experience at a later time. Although we have investigated using this policy deviation proxy previously (de Bruin et al., 2016b), we found empirically that using the strength of the initial exploration yields better results. This can partly be explained by the fact that recalculating the policy deviation makes the proxy dependent on the learning process and partly by the fact that sequences with more exploration also result in different states being visited.

7.1.2 Stochastic Experience Retention Implementation

For the temporal difference error and exploration-based experience retention methods, keeping some experiences in the buffer indefinitely might lead to over-fitting to these samples.

Notation	Proxy	Explanation
Expl(α)	Exploration	Experiences with the least exploration are stochastically overwritten with new ones.
TDE(α)	Surprise	Experiences with the smallest temporal difference error are stochastically overwritten with new ones.
Resv	Age	The buffer is overwritten such that each experience observed so far has an equal probability of being in the buffer.

Table 4: New and uncommon experience retention strategies considered in this work.

Notation	Proxy	Explanation
Uniform + FIS	-	Experiences are sampled uniformly at random, FIS (Section 7.2) is used to account for the distribution changes caused by the *retention* policy.
PER+FIS	Surprise	Experiences are sampled using rank based stochastic prioritized experience replay based on the temporal difference error. Full importance sampling is used to account for the distribution changes caused by both the retention and sampling policies.

Table 5: New experience sampling strategies considered in this work.

Additionally, although the overwrite metric we choose might provide a decent proxy for the usefulness of experiences, we might still want to be able to scale the extent to which we base the contents of the buffer on this proxy. We therefore use the same stochastic rank-based selection criterion of (7) suggested by Schaul et al. (2016), but now to determine which experience in the buffer is overwritten by a new experience. We denote this as TDE(α) for the temporal difference-based retention strategy and Expl(α) for the exploration-based policy. Here, α is the parameter in (7) which determines how strongly the buffer contents will be based on the chosen utility proxy. A sensitivity analysis of α for both Expl and PER is given in Appendix 9.3. The notation used for the new experience retention strategies is given in Table 4.

7.2 Experience Sampling

For the choice of proxy when *sampling* experiences from the buffer, we consider the available methods from the literature: sampling either uniformly at random [Uniform], using stochastic rank-based prioritized experience replay [PER] and combining this with weighted importance sampling [PER+IS]. Given a buffer that contains useful experiences, these methods have shown to work well. We therefore focus on investigating how the experience reten-

tion and experience sampling strategies interact. In this context we introduce a weighted importance sampling method that accounts for the full experience selection strategy.

Importance sampling according to (8) can be used when performing prioritized experience replay from a buffer that contains samples with a distribution that is unbiased with respect to the environment dynamics. When this is not the case, we might need to compensate for the effects of changing the contents of the buffer, potentially in addition to the current change in the sampling probability. The contents of the buffer might be the result of many subsequent retention probability distributions. Instead of keeping track of all of these, we compensate for both the retention and sampling probabilities by using the number of times an experience in the buffer has actually been replayed. When replaying an experience i for the K-th time, we relate the importance-weight to the probability under uniform sampling from a FIFO buffer of sampling an experience X times, where X is at least K: $\Pr(X \geq K|_{\text{FIFO[Uniform]}})$. We refer to this method as Full Importance Sampling (FIS) and calculate the weights according to :

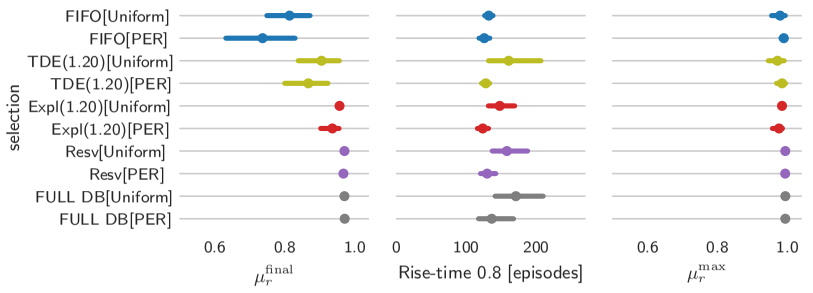
$$\omega_i^{\text{FIS}} = \left(\frac{\Pr(X \geq K|_{\text{FIFO[Uniform]}})}{\left[\sum_{j=1}^{\lceil np \rceil} \Pr(X \geq j|_{\text{FIFO[Uniform]}})\right]/np} \right)^{\beta}.$$

Here, n is the lifetime of an experience for a FIFO retention strategy in the number of batch updates, which is the number of batch updates performed so far when the buffer is not yet full. The probability of sampling an experience during a batch update when sampling uniformly at random is denoted by p. Note that np is the expected number of replays per experience, which following Schaul et al. (2016) we take as 8 by choosing the number of batch updates per episode accordingly. As in Section 3.3.2 we use β to scale between not correcting for the changes and correcting fully. Since the probability of being sampled at least K times is always smaller than one for $K > 0$, we scale the weights such that the sum of the importance weights for the expected np replays under FIFO[Uniform] sampling is the same as when not using the importance weights ($n \cdot p \cdot 1$). The probability of sampling an experience at least K times under FIFO[Uniform] sampling is calculated using the binomial distribution:

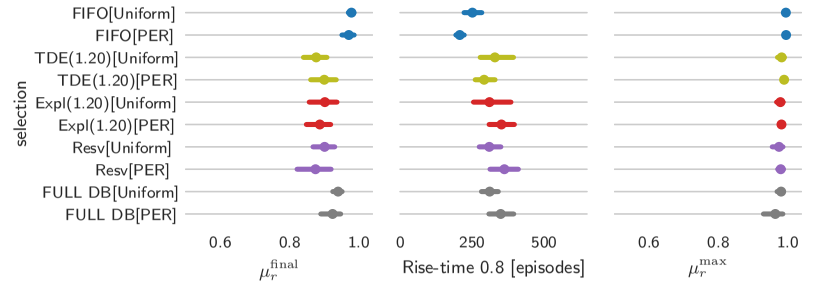
$$\Pr(X \geq K|_{\text{FIFO[Uniform]}}) = 1 - \sum_{k=0}^{K} \binom{n}{k} p^k (1-p)^{n-k}.$$

Correcting fully ($\beta = 1$) for the changed distributions would make the updates as unbiased as those from the unbiased FIFO uniform distribution (Needell et al., 2016). However, since the importance weights of experiences that are repeatedly sampled for stability will quickly go to zero, it might also undo the stabilizing effects that were the intended outcome of changing the distribution in the first place. Additionally, as discussed in Section 3.3.2, the FIFO Uniform distribution is not the only valid distribution. As demonstrated in Section 8.4, it is therefore important to determine whether compensating for the retention strategy is necessary before doing so.

The notation for the selection strategies with this form of importance sampling is given in Table 5.



(a) Swing-up



(b) Magman

Figure 12: Performance of the experience selection methods under the default conditions of moderate sampling frequencies and no state or actuator noise. A description of the performance measures is given in Section 5.1.

8. Experience Selection Results

Using the experience retention and sampling methods discussed in Section 7, we revisit the scenarios discussed in Section 6. We first focus on the methods without importance sampling, which we discuss separately in Section 8.4. Besides the tests on the benchmarks of Section 4, we also show results on six additional benchmarks in Section 8.5. There we also discuss how to choose the size of the experience buffer.

8.1 Basic Configuration

We start by investigating how these methods perform on the benchmarks in their basic configuration, with a sampling rate of 50 Hz and no sensor or actuator noise. The results are given in Figure 12 and show that it is primarily the combination of *retention* method

and buffer size that determines the performance. It is again clear that this choice here depends on the benchmark. On the pendulum benchmark, where storing all experiences works well, the Resv method works equally well while storing only 10^4 experiences, which equals 50 of the 3000 episodes. On the magman benchmark, using a small buffer with only recent experiences works better than any other method.

Sampling according to the temporal difference error can be seen to benefit primarily the learning speed on the pendulum. On the magman, PER only speeds up the learning process when sampling from recent experiences. When sampling from diverse experiences, PER will attempt to make the function approximation errors more even across the state-action space, which as discussed before, hurts performance on this benchmark.

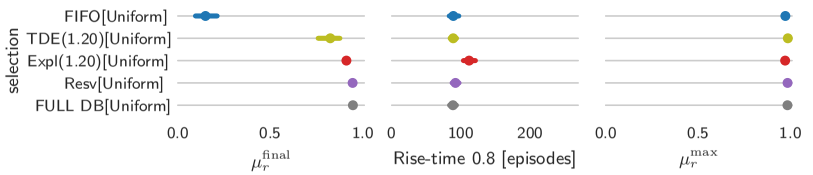
8.2 Effect of the Sampling Frequency

For higher sampling frequencies, the performance of the different experience selection methods is shown in Figure 13. We again see that higher sampling frequencies place different demands on the training data distribution. With the decreasing exploration, retaining the right experiences becomes important. This is most visible on the Magman benchmark where FIFO retention, which resulted in the best performance at the end of training for the base sampling frequency, now performs worst. Retaining all experiences works well on both benchmarks. When not all experiences can be retained, the reservoir retention method is still a good option here, with the exploration-based method a close second.

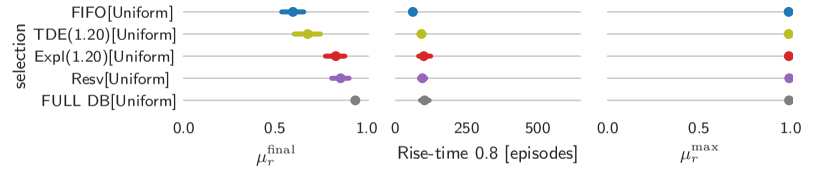
8.3 Sensor and Actuator Noise

We also test the performance of the methods in the presence of noise, similarly to Section 6.5. The main question here is how the noise might affect the methods that use the temporal difference error δ as the usefulness proxy. The concern is that these methods might favor noisy samples, since these samples might cause bigger errors. To test this we perform learning runs on the pendulum task while collecting statistics on all of the experiences in the mini-batches that are sampled for training. The mean absolute values of the noise in the experiences that are sampled are given in Table 6. It can be seen that the temporal difference error-based methods indeed promote noisy samples. The noise is highest for those dimensions that have the largest influence on the value of Q.

In Figure 14 the performance of the different methods on the two benchmarks with noise is given. The tendency to seek out noisy samples in the buffer is now clearly hurting the performance of PER sampling, as the performance with PER is consistently worse than with uniform sampling. For our chosen buffer size the retention strategy is still more influential and interestingly the TDE-based retention method does not seem to suffer as much here. The relative rankings of the retention strategies are similar to those without noise.



(a) Swing-up [100 Hz]

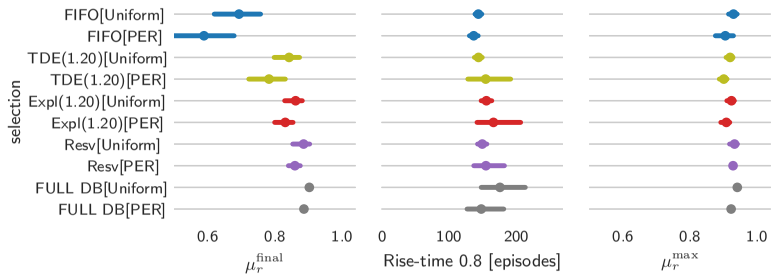


(b) Magman [200 Hz]

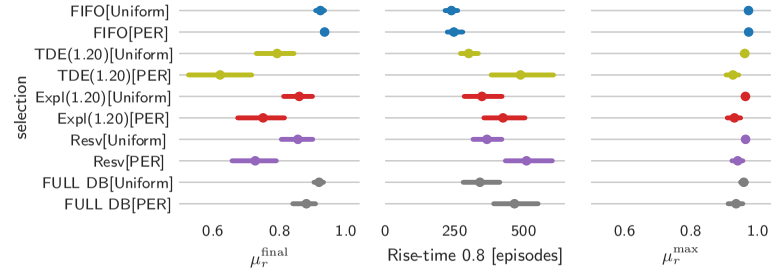
Figure 13: Performance of the experience selection methods with increased sampling frequencies. Results are from 50 learning runs. A description of the performance measures is given in Section 5.1.

	position	velocity	action
Expl(1.0)[Uniform]	$1.584 \cdot 10^{-2}$	$1.582 \cdot 10^{-2}$	$1.594 \cdot 10^{-2}$
Expl(1.0)[PER]	$1.654 \cdot 10^{-2}$	$1.630 \cdot 10^{-2}$	$1.595 \cdot 10^{-2}$
TDE(1.0)[Uniform]	$1.713 \cdot 10^{-2}$	$1.627 \cdot 10^{-2}$	$1.598 \cdot 10^{-2}$
TDE(1.0)[PER]	$1.846 \cdot 10^{-2}$	$1.743 \cdot 10^{-2}$	$1.594 \cdot 10^{-2}$

Table 6: Mean absolute magnitude of the noise per state-action dimension in the mini batches as a function of the experience selection procedure.



(a) Swing-up [$\sigma_s = \sigma_a = 0.02$]



(b) Magman [$\sigma_s = \sigma_a = 0.02$]

Figure 14: Performance of the experience selection methods with with sensor and actuator noise. Results are from 50 learning runs. A description of the performance measures is given in Section 5.1.

8.4 Importance Sampling

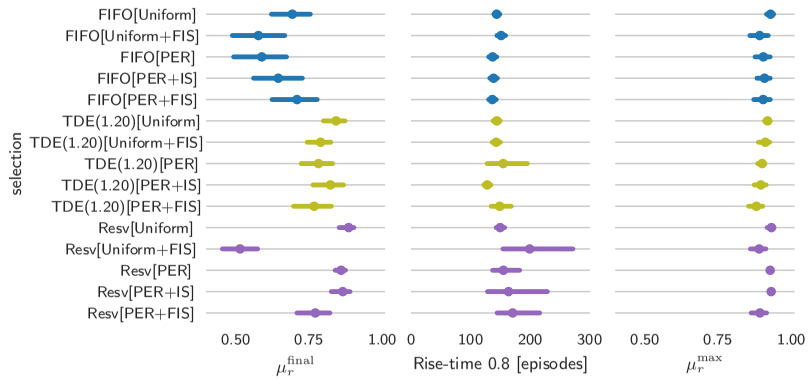
Finally, we investigate the different importance sampling strategies that were discussed in Sections 3.3.2 and 7.2. We do this by using the FIFO, TDE and Resv retention strategies as representative examples. We consider the benchmarks with noise, since as we discussed in Section 3.3.2, the stochasticity in the environment can make importance sampling more relevant. The results are shown in Figure 15. We discuss per retention strategy how the sample distribution is changed and whether the change introduces a bias that should be compensated for through importance sampling.

FIFO: This retention method results in an unbiased sample distribution. When combined with uniform sampling, there is no reason to compensate for the selection method. Doing so anyway (FIFO[Uniform + FIS]) results in downscaling the updates from experiences that happen to have been sampled more often than expected, effectively reducing the batch-size while not improving the distribution. The variance of the updates is therefore increased without reducing bias. This can be seen to hurt performance in Figure 15, especially on the swing-up task where sample diversity is most important. Using PER also hurts performance in the noisy setting as this sampling procedure *does* bias the sample distribution. Using importance sampling to compensate for just the sampling procedure (FIFO[PER+IS]) helps, but the resulting method is not clearly better than uniform sampling.

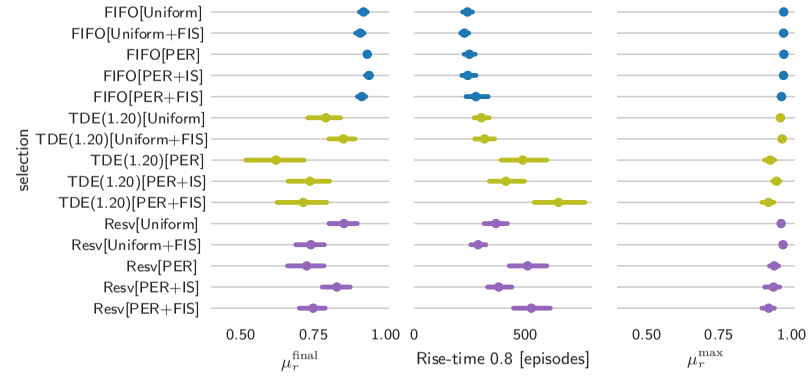
TDE: When the retention strategy is based on the temporal difference error, there is a reason to compensate for the bias in the sample distribution. It can be seen from Figure 15 however, that the full importance sampling scheme improves performance on the magman benchmark, but not on the swing-up task. The likely reason is again that importance sampling indiscriminately compensates for both the unwanted re-sampling of the environment dynamics and reward distributions as well as the beneficial re-sampling of the state-action space distribution. The detrimental effects of compensating for the latter seen to outweigh the beneficial effects of compensating for the former on this benchmark where state-action space diversity has been shown to be so crucial.

Resv: The reservoir retention method is not biased with respect to the reward function or the environment dynamics. Although the resulting distribution is strongly off-policy (assuming the policy has changed during learning), this does not present a problem for a deterministic policy gradient algorithm with Q-learning updates, other than that it might be harder to learn a function that generalizes to a larger part of the state space. When sampling uniformly, we do sample certain experiences, from early in the learning process, far more often than would be expected under a FIFO[Uniform] selection strategy. The FIS method compensates for this by weighing these experiences down, effectively reducing the size of both the buffer and the mini-batches. In Figure 15, this can be seen to severely hurt the performance on the swing-up problem, as well as the learning stability on the magman benchmark.

Interesting to note is that on these two benchmarks, for all three considered retention strategies, using importance sampling to compensate for the changes introduces by PER



(a) Swing-up [$\sigma_s = \sigma_a = 0.02$]



(b) Magman [$\sigma_s = \sigma_a = 0.02$]

Figure 15: Performance of representative experience selection methods with and without importance sampling on the benchmarks with sensor and actuator noise. A description of the performance measures is given in Section 5.1.

only improved the performance significantly when using PER resulted in poorer performance than not using PER. Similarly, using FIS to compensate for the changes introduced in the buffer distribution only improved the performance when those changes should not have been introduced to begin with.

8.5 Additional Benchmarks

The computational and conceptual simplicity of the two benchmarks used so far allowed for comprehensive tests and a good understanding of the characteristics of the benchmarks. However, we also saw that the right experience selection strategy is benchmark dependent. Furthermore, deep reinforcement learning yields most of its advantages over reinforcement learning without simpler function approximation on problems with higher dimensional state and action spaces. To obtain a more complete picture we therefore perform additional tests on 6 benchmarks of varying complexity.

8.5.1 Benchmarks

In the interest of reproducibility, we use the open source RoboSchool (Klimov, 2017) benchmarks together with the openAI baselines (Dhariwal et al., 2017) implementation of DDPG. We have adapted the baselines code to include the experience selection methods considered in this section. Our adapted code is available online.[3]

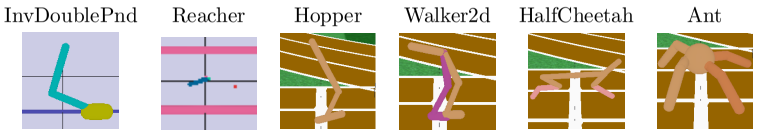
The baselines version of DDPG uses Gaussian noise added to the parameters of the policy network for exploration (Plappert et al., 2018). In contrast to the other experiments in this work, the strength of the exploration is kept constant during the entire learning run. For the Expl method we still consider the 1-norm of the distance between the exploration policy action and the unperturbed policy action as the utility of the sample.

For the benchmarks listed in Table 7, we compare the default FULL DB[Uniform] selection strategy in the baselines code to the alternative retention strategies considered in this work with uniform sampling. We show the maximum performance for these different retention strategies as a function of the buffer size in Figure 16.

8.5.2 Results

As shown in Figure 16, on these noise-free benchmarks with constant exploration and moderate sampling frequencies, the gains obtained by using the considered non-standard experience selection strategies are limited. However, in spite of the limited number of trials performed due to the computational complexity, trends do emerge on most of the bench-

3. The code is available at https://github.com/timdebruin/baselines-experience-selection.



$\lvert S\rvert$	9	9	15	22	26	28
$\lvert A\rvert$	1	2	3	6	6	8

Table 7: The RoboSchool benchmarks considered in this section with the dimensionalities of their state and action spaces.

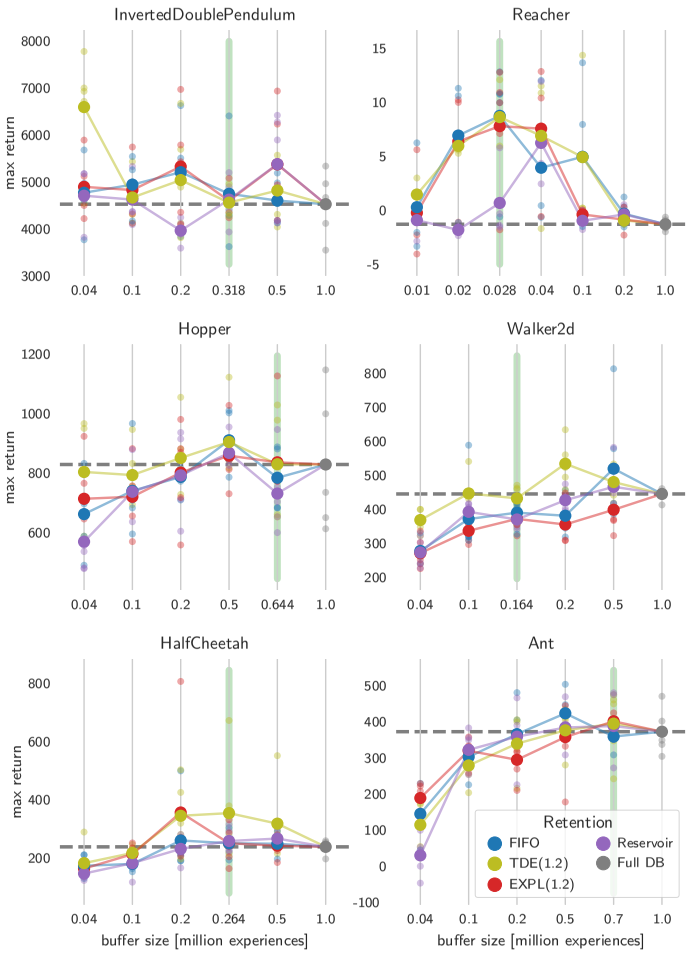


Figure 16: Maximum performance during a training run on the Roboschool benchmarks as a function of the retention strategy and buffer size. Results for the individual runs and their means are shown. In Appendix 9.3, we additionally show the mean (Figure 26) and final (Figure 25) performance. Green lines indicate the rule of thumb buffer sizes of Figure 17.

marks. On all benchmarks, the best performance is seen not when retaining all experiences, but rather when learning from a smaller number of experiences. This is most visible on the reacher task, which involves learning a policy for a 2-DOF arm to move from one random location in its workspace to another random location. For this task, the best performance for all retention strategies is observed when retaining less than a tenth of all experiences.

For these noise-free benchmarks, the temporal difference error is an effective proxy for the utility of the experiences, resulting in the highest or close to the highest maximum performance on all benchmarks.

The exploration-based retention strategy was introduced to prevent problems when reducing exploration and for high sampling frequencies. Since the exploration is not decayed and the sampling frequencies are modest, there is no real benefit when applying this strategy to these benchmarks. However, it also does not seem to hurt performance compared to the age-based retention strategies. The constant exploration on these benchmarks additionally means that the performance of FIFO and Reservoir retention are rather close, although due to premature convergence of the data distribution, reservoir retention does suffer the most when the buffer capacity is too low.

Figures 16, 25 and 26 show that when the right proxy for the utility of experiences is chosen, performance equal to and often exceeding that of retaining all experiences can be obtained while using only a fraction of the memory. This begs the question of how to choose the buffer size.

As it tends to result in more stable learning, retaining as many experiences as possible seems a sensible first choice for the buffer size. We therefore base our suggestion for subsequently tuning the buffer size on the learning curves of the FULL DB[Uniform] method. The complexity of the control task at hand determines the minimal number of environment steps required to learn a good policy, as well as the number of experiences that need to be retained in a buffer for decent learning performance. We propose to use the number of experiences needed to get to 90% of the final performance as a *rough* empirical estimate of the optimal buffer size. We show this rule of thumb in Figure 17 and have indicated the experiments with the proposed buffer sizes in Figure 16 with vertical green lines.

Instead of iteratively optimizing the buffer size over several reinforcement learning trials, extrapolation of the learning curve (Domhan et al., 2015) could also be used to limit the buffer capacity when the remaining learning performance increase is expected to be small. This would allow the method to work immediately for novel tasks.

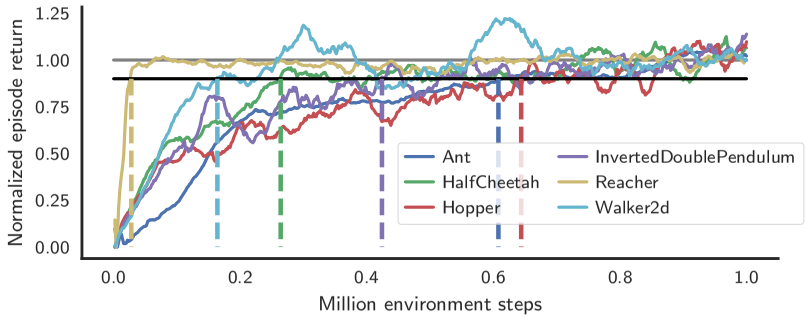


Figure 17: Learning curves of the FULL DB method on the different benchmarks, averaged over 5 trials. The curves are normalized by the final performance (the mean performance over the last $2 \cdot 10^5$ steps). Indicated are the number of steps needed to get to 90% of the final performance.

9. Conclusions and Recommendations

In this work, we have investigated how the characteristics of a control problem determine what experiences should be replayed when using experience replay in deep reinforcement learning.

We first investigated how factors such as the generalizability over the state-action space and the sampling frequency of the control problem influenced the balance between the need for on-policy experiences versus a broader coverage of the state-action space.

We then investigated a number of proxies for the utility of experiences which we used to both decide which experiences to retain in a buffer and how to sample from that buffer. We performed experiments that showed how these methods were affected by noise, increased sampling frequencies and how their performance varied across benchmarks and experience buffer sizes.

Based on these investigations we present a series of recommendations below for the three choices concerning experience selection: how to choose the capacity of the experience replay buffer, which experiences to retain in a buffer that has reached its capacity and how to sample from that buffer. These choices together should ensure that the experiences that are replayed to the reinforcement learning agent facilitate quick and stable learning of a policy with good performance. An example of applying the procedure outlined below on the Magman benchmark is given in Figure 18. Note the proposed methods are especially relevant when faced by issues that might occur in a physical-control setting, such as a need for constrained exploration, high or low sampling frequencies, the presence of noise and hardware limitations that limit the experience buffer size. Section 8.5 shows that the potential gains might be limited for processes where these problems do not occur.

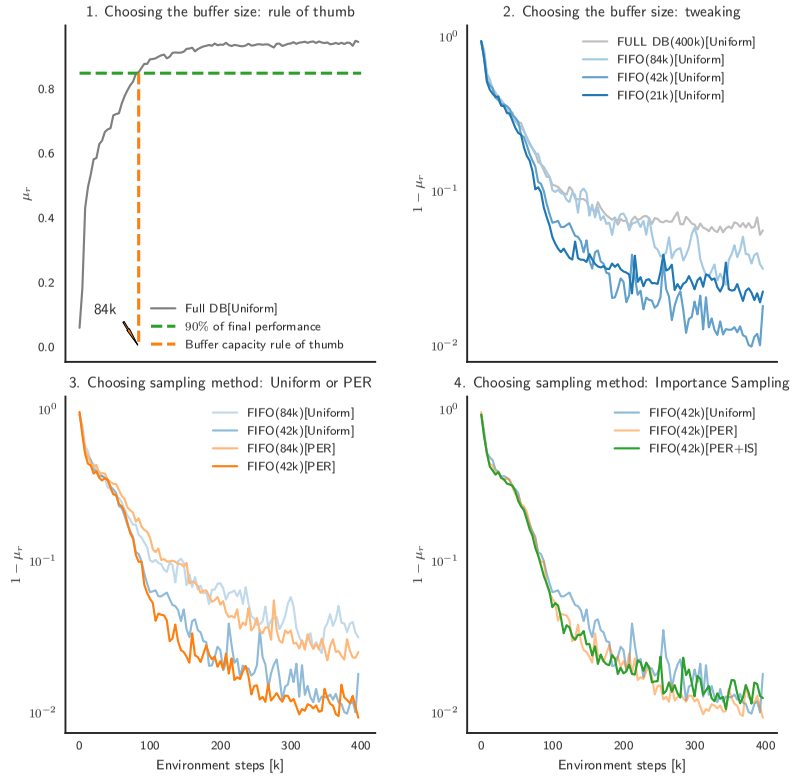


Figure 18: Demonstration of the proposed process for the magman benchmark. 1: Based on the performance of the Full DB[Uniform] method, the rule of thumb indicates a buffer capacity of 84×10^3 experiences. As there are no special circumstances such as high sampling frequencies and the magman requires a very precise policy that does not easily generalize due to the highly nonlinear behavior of the magnets, FIFO retention is used. 2: By exploring around the proposed buffer size, a buffer capacity of 42×10^3 experiences is chosen. 3: Sampling from the buffer based on the temporal difference error can help speed up and stabilize learning, but is very dependent on the experiences that are in the buffer to begin with. 4: Since the benchmark fully deterministic (noise free), importance sampling is not needed in this case and can be seen to undo some of the benefits of PER.

9.1 Choosing the Buffer Capacity

Although it is not the best retention strategy in most of the benchmarks we have considered, retaining as many experiences as possible is a good place to start. This tends to result in more stable learning, even if the eventual performance is not always optimal.

If the learning curve for the FULL DB experiments reaches a level of performance close to the performance after convergence in significantly fewer environment steps than there are experience samples in the buffer, it might be worthwhile reducing the size of the buffer. Our proposed rule of thumb is to to make the buffer size roughly equal to 90% the number of environment steps needed to reach the final performance level.

9.2 Choosing the Experience Utility Proxy

When not all experiences are retained in the buffer, a proxy for the utility of the individual experiences is needed to determine which experiences to retain and which to discard. In this work, we have discussed strategies based on several proxies and shown that the right strategy is problem dependent. Although finding the right one will likely require some experimentation, we discuss here what properties of the control problem at hand make certain strategies more likely or less likely to succeed.

FIFO: Although off-policy reinforcement-learning algorithms can learn from samples obtained by a different policy than the optimal policy that is being learned, the reality of deep reinforcement learning is that a finite amount of shared function approximation capacity is available to explain all of the training data. While simply using larger networks might help, we show in Appendix 9.3 that learning only from more recent data (which corresponds more closely to the policy being learned) can work better. A large potential downfall presents itself when the policy suddenly changes in a way that changes the distribution of the states that are visited. As shown in Section 6.2, this can quickly destabilize the learning process. Extra care should be taken when using FIFO retention in combination with decaying exploration. This is especially true for problems where multiple policies are possible that give similar returns but distinct state-space trajectories, such as swinging up a pendulum either clockwise or anti-clockwise.

TDE: The idea behind selecting certain experiences over others is that more can be learned from these samples. The temporal difference error is therefore an interesting proxy, especially during the early stages of the learning process when the error is mostly caused by the fact that the value function has not been accurately learned yet. In the experiments of Schaul et al. (2016) as well as in our own experiments, prioritizing experiences with larger TD errors was observed to improve both the speed of learning as well as the eventual performance in many cases. The downside of using the TD error as an experience utility proxy is that the error can also be caused by sensor and actuator noise, environment stochasticity or function approximation accuracy differences as a result of differences in the state space coverage. We have shown in Section 8.3 how noise can hurt the performance of the algorithm when using this proxy and argued in Section 3.3.2 how this proxy introduces a harmful bias in the presence of environment stochasticity.

Exploration: We introduced an additional proxy based on the observation that, on physical systems, exploration can be costly. By using the strength of the exploration signal as a proxy for the utility of the experience, some of the problems mentioned for the FIFO strategy when reducing exploration can be ameliorated. As shown in Section 6.4 and Section 8.2, sufficient diversity in the action space is most important when the dependency of the value function on the action is relatively small, such as for increased sampling frequencies. The downside of this strategy is that since it focuses on early experiences that are more off-policy, it can take longer for the true value function to be learned. Besides the impact on training speed, the focus on off-policy data can also limit the maximum controller performance.

Reservoir sampling: By using reservoir sampling as a retention strategy, the buffer contains, at all times, samples from all stages of learning. As with the exploration-based policy, this ensures that initial exploratory samples are retained which can significantly improve learning stability on domains where FIFO retention does not work. However, of the methods mentioned here, reservoir sampling is the one most severely impacted by a too small experience buffer, as the data distribution in the buffer will converge prematurely and will not cover the state-action space distribution of the optimal policy well enough.

9.3 Experience Sampling and Importance Sampling

The experiences that are used to learn from are not just determined by the buffer retention strategy, but also by the method of sampling experiences from the buffer. While the retention strategy needs to ensure that a good coverage of the state-action space is maintained in the buffer throughout learning, the sampling strategy can seek out those experiences that can result in the largest immediate improvement to the value function and policy. It can therefore be beneficial to *sample* based on the temporal difference error (as suggested by Schaul et al. 2016), which can improve learning speed and performance, while basing the *retention* strategy on a more stable criterion that either promotes stability or ensures that only samples from the relevant parts of the state-action space are considered by the sampling procedure.

As discussed in Sections 3.3.2 and 8.4, selecting experiences based on the temporal difference error in stochastic environments introduces a bias that should be compensated for through weighted importance sampling in order to make the learning updates valid. While the other experience selection methods in this work change the distribution of the samples, these changed distributions are still valid for an off-policy deterministic gradient algorithm.

Acknowledgments

This work is part of the research programme Deep Learning for Robust Robot Control (DL-Force) with project number 656.000.003, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

Appendix A. Simple Benchmarks

Here, a more detailed mathematical description is given of the pendulum swing-up and magnetic manipulation benchmarks. A high level description of these benchmarks was given in Section 4.

The benchmarks will be described based on their true (unnormalized) physical environment states s_{unnorm} and actions a_{unnorm}. In the main body of this work the components of these states and actions are normalized: $s, s_{\mathcal{E}} \in [-1, 1]^n, a, a_{\mathcal{E}} \in [-1, 1]^m$. See Figure 1 for a description of the symbols used.

The dynamics of both problems are defined as differential equations, which we use to calculate the next environment state s'_{unnorm} as a function of the current state s_{unnorm} and action a_{unnorm} using the (fourth order) Runge-Kutta method. The reward is in both cases given by:

$$r = -(W_1|s'_{\text{unnorm}} - s_{\text{unnorm}_{\text{ref}}}| + W_2|a_{\text{unnorm}}|). \tag{9}$$

In both cases a fixed reference state $s_{\text{unnorm}_{\text{ref}}}$ is used.

A.1 Pendulum Swing-Up

For the pendulum swing-up task, the state s_{unnorm} is given by the angle $\theta \in [-\pi, \pi]$ and angular velocity $\dot{\theta}$ of a pendulum, which starts out hanging down under gravity $s^0_{\text{unnorm}} = [\theta\ \dot{\theta}]^T = [0\ 0]^T$. For the normalization of the velocities, $\theta_{\min} = -30\,\text{rad}\,\text{s}^{-1}$ and $\theta_{\max} = 30\,\text{rad}\,\text{s}^{-1}$ are used. The action space is one dimensional: it is the voltage applied to a motor that exerts torque on the pendulum $a_{\text{unnorm}} \in [-3, 3]$ V. The angular acceleration of the pendulum is given by:

$$\ddot{\theta} = \frac{-Mgl\sin(\theta) - (b + K^2/R)\dot{\theta} + (K/R)a_{\text{unnorm}}}{J}.$$

Where $J = 9.41 \times 10^{-4}\,\text{kg}\,\text{m}^2$, $M = 5.5 \times 10^{-2}\,\text{kg}$, $g = 9.81\,\text{m}\,\text{s}^{-2}$, $l = 4.2 \times 10^{-2}\,\text{m}$, $b = 3 \times 10^{-6}\,\text{kg}\,\text{m}^2\,\text{s}^{-1}$, $K = 5.36 \times 10^{-2}\,\text{kg}\,\text{m}^2\,\text{s}^{-2}\,\text{A}^{-1}$ and $R = 9.5\,\text{V}\,\text{A}^{-1}$ are respectively the pendulum inertia, the pendulum mass, the acceleration due to gravity, pendulum length, viscous damping coefficient, the torque constant and the rotor resistance (Alibekov et al., 2018). For this task $W_1 = [50\ 1]$ and $W_2 = 10$ and $s_{\text{unnorm}_{\text{ref}}} = [-\pi\ 0]^T = [\pi\ 0]^T$. The absolute value of the state is used in (9).

A.2 Magnetic Manipulation

In the magnetic manipulation problem, the action space represents the squared currents through four electromagnets under the track; $a^j_{\text{unnorm}} \in [0,\ 0.6]A^2$ for $j = 1, 2, 3, 4$. The state of the problem is defined as the position $x \in [-0.035,\ 0.105]$ m of the ball relative to the center of the first magnet and the velocity $\dot{x}$ $\text{m}\,\text{s}^{-1}$ of the ball: $s_{\text{unnorm}} = [x\ \dot{x}]^T$. For the normalization of the velocities, $\dot{x}_{\min} = -0.4\,\text{m}\,\text{s}^{-1}$ and $\dot{x}_{\max} = 0.4\,\text{m}\,\text{s}^{-1}$ are used. When the position of the ball exceeds the bounds, the position is set to the bound and the velocity is set to $0.01\,\text{m}\,\text{s}^{-1}$ away from the wall. An additional reward of -1 is given for the time-step at which the collision occurred. The acceleration of the ball is given by:

$$\ddot{x} = -\frac{b}{m}\dot{x} + \frac{1}{m}\sum_{j=1}^{4} g\,(x, j)\,a^j_{\text{unnorm}},$$

with

$$g(x, j) = \frac{-c_1\ (x - 0.025j)}{\left((x - 0.025j)^2 + c_2\right)^3}.$$

Here, $g(x, j)$ is the nonlinear magnetic force equation, $m = 3.200 \times 10^{-2}\,\text{kg}$ the ball mass, and $b = 1.613 \times 10^{-2}\,\text{N}\,\text{s}\,\text{m}^{-1}$ the viscous friction of the ball on the rail. The parameters c_1 and c_2 were empirically determined from experiments on a physical setup to be $c_1 = 5.520 \times 10^{-10}\,\text{N}\,\text{m}^5\,\text{A}^{-1}$ and $c_2 = 1.750 \times 10^{-4}\,\text{m}^2$ (Alibekov et al., 2018).

For the magnetic manipulation problem we take $W_1 = [100\ 5]$, $W_2 = [0\ 0\ 0\ 0]$, $s^0_{\text{unnorm}} = [0\ 0]^T$ and $s_{\text{unnorm}_{\text{ref}}} = [0.035\ 0]^T$ in (9).

Appendix B. Implementation Details

This appendix discusses the chosen hyperparameters of the methods discussed in Section 3, that were used to obtain the results in this paper. Only those hyperparameters that were not explicitly mentioned in the earlier sections of this work are mentioned here.

B.1 Neural Networks

This appendix describes the architecture and training procedure of the used neural networks.

Swing-up and Magman

To perform the experiments in this work, the DDPG method of Lillicrap et al. (2016) was reimplemented in Torch (Collobert et al., 2011). For all experiments except for the control experiment in Appendix 9.3, the actor and critic networks had the following configuration:

The actor is a fully connected network with two hidden layers, each with 50 units. The hidden layers have rectified linear activation functions. The output layer has hyperbolic tangent nonlinearities to map to the normalized action space.

The critic is a fully connected network with three hidden layers. The layers have rectified linear activation functions and 50, 50 and 20 units respectively. The state is the input to the first hidden layer, while the action is concatenated with the output of the first hidden layer and used as input to the second hidden layer. The output layer is linear.

To train the networks, the ADAM optimization algorithm is used (Kingma and Ba, 2015). We use a batch size of 16 to calculate the gradients. For all experiments we use 0.9 and 0.999 as the exponential decay rates of the first and second order moment estimates respectively. The step-sizes used are 10^{-4} for the actor and 10^{-3} for the critic. We additionally use $\mathcal{L}_2$ regularization on the critic weights of 5×10^{-3}.

For the DQN experiments, a critic network similar to the DDPG critic was used. The critic-only differs in the fact that instead of having actions as an input, the output size is increased to the number of discrete actions considered. The parameters θ^- of the target critic are updated to equal the online parameters θ every 200 batch updates.

Roboschool Benchmarks

For the experiments on the Roboschool benchmarks, we use a slightly modified version of the DDPG implementation in the openAI baselines (Dhariwal et al., 2017) repository. We have adapted the baselines code to include the experience selection methods considered in this section. Our adapted code is available online.[4] We here summarize the relevant differences from the implementation used on the simple benchmarks.

The actor and critic networks have two hidden layers with 64 units each. Layer normalization (Ba et al., 2016) is used in both networks after both hidden layers. The multiplier of the $\mathcal{L}_2$ regularization on the weights of the critic with is 1×10^{-2}. A batch size of 64 is used, with a sample reuse of 32. Training is performed every 100 environment steps, rather than after completed episodes.

B.2 Exploration

Swing-Up and Magman

We use an Ornstein-Uhlenbeck noise process (Uhlenbeck and Ornstein, 1930) as advocated by Lillicrap et al. (2016). The dynamics of the noise process are given by

$$u(k+1) = u(k) + \theta\mathcal{N}(0,1) - \sigma u(k).$$

The equation models the velocity of a Brownian particle with friction. We use $\theta = 5.14$, $\sigma = 0.3$. Using this temporally correlated noise allows for more effective exploration in domains such as the pendulum swing-up. It also reduces the amount of damage on physical systems relative to uncorrelated noise (Koryakovskiy et al., 2017). For high frequencies, uncorrelated noise is unlikely to result in more than some small oscillations around the downward equilibrium position.

The noise signal is clipped between -1 and 1 after which it is added to the policy action and clipped again to get the normalized version of the control action a.

For the DQN experiments, epsilon greedy exploration was used with the probability of taking an action uniformly at random decaying linearly from $\epsilon = 0.7$ to $\epsilon = 0.01$ over the first 500 episodes.

Roboschool Benchmarks

For easy comparison to other work, we use the exploration strategy included in the baselines code. This means that for the Roboschool benchmarks we do not decay the strength of the exploration signal over time. Compared to our other benchmarks, the second difference is that the noise is added in the parameter space of the policy rather than directly in the action space (Plappert et al., 2018). The amplitude of the noise on the parameters is scaled such that the standard deviation of the exploration signal in action-space is 0.2.

Appendix C. Baseline Controller

In this work we use the fuzzy Q-iteration algorithm of Buşoniu et al. (2010) as a baseline. This algorithm uses full knowledge of the system dynamics and reward function to compute a controller that has a proven bound on its sub-optimality for the deterministic (noise-free) case.

4. The code is available at https://github.com/timdebruin/baselines-experience-selection.

For the tests with sensor and actuator noise, the same controller as in the noise-free setting is used. To make the performance normalization (Section 5.1) fair, the performance of the controller is taken as the mean of 50 repetitions of taking the maximum obtained mean reward per episode over 1000 episodes with different realizations of the noise:

$$r_{\text{baseline with noise}} = \frac{1}{50}\sum_{i=1}^{50} \max(r^{\text{mean}}_{\text{episode i,1}}, \ldots, r^{\text{mean}}_{\text{episode i,1000}}).$$

Note that although this equalizes the chances of getting a favorable realization of the sensor and actuator noise sequences, it does not compensate for the fact that the fuzzy Q-iteration algorithm is unsuitable for noisy environments. Since the DDPG method used in this work can adjust the learned policy to the presence of noise in the environment, it outperforms the baseline in some situations. This is not an issue since we are interested in the relative performance of different experience selection strategies and only use the baseline as a reference point.

Appendix D. Additional Sensitivity Analyses and Figures

This section contains additional analyses and figures that were left out of the main body of the paper for brevity.

D.1 Performance on the Magman Benchmark as a Function of Network Size

In the main body of the paper, a number of experiments are shown in which the performance of the magman benchmark is better with a small FIFO experience buffer than it is when retaining all experiences. As we use relatively small neural networks on the magman benchmark, it could be expected that at least part of the reason that training on all experiences results in poorer performance is that the function approximator simply does not have enough capacity to accurately cover the state-action space. We therefore compare the performance of the networks used on the magman benchmark in the main body of this work to that of the original DDPG architecture, which has more than 40 times as many parameters. Table 8 compares the network architectures and the number of parameters of

Architecture	hidden layer units	parameters swing-up	parameters magman
Small-critic	[50, 50, 20]	3791	3941
Small-actor	[50, 50]	2751	2904
DDPG original-critic	[400, 300]	122101	123001
DDPG original-actor	[400, 300]	121801	122704

Table 8: The architectures of the networks compared in this section, with the number of parameters.

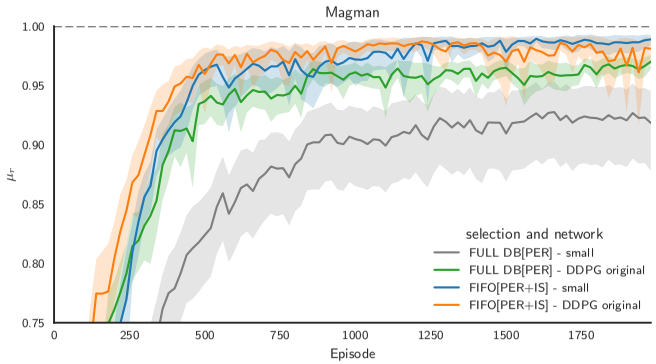


Figure 19: Influence of network size on the performance of the magman benchmark, when retaining all 4×10^5 experiences (FULL DB) versus retaining only the last 10^4 experiences (FIFO). The small policy network used for most of the experiments on the magman has 2904 parameters, while the original DDPG network has 122704 parameters on the magman benchmark. In both cases the critic networks had slightly more parameters.

these architectures. It can be seen from Figure 19 that, while the larger network is able to learn more successfully from the FULL DB buffer, it is outperformed by both the small and the large network using the FIFO buffer. The eventual performance is best for our smaller network trained on a small buffer, although learning is somewhat faster with the larger network.

D.2 Sensitivity Analysis α

In both the PER sampling as well as the TDE and Expl retention methods, the parameter α (7) determines how strongly the used experience utility proxy influences the selection method. Here, we show the sensitivity of both PER (Figure 21) and Expl (Figure 20) with respect to this parameter.

In Figure 20 it can be seen that on the Pendulum benchmark, where Expl retention has already been shown to aid stability, increasing α helps to improve the final performance more. This increased stability comes at the cost of somewhat reduced maximum performance. With PER sampling it does not seem to hurt the learning speed. On the Magman benchmark, where FIFO retention works better than Expl retention, increasing α (and thus relying more on the wrong proxy for the benchmark) hurts performance. Interesting to see is that compared to uniform sampling, PER speeds up the learning for low values of α, while it hurts for large values of α. This demonstrates again the need to choose both parts of experience selection with care.

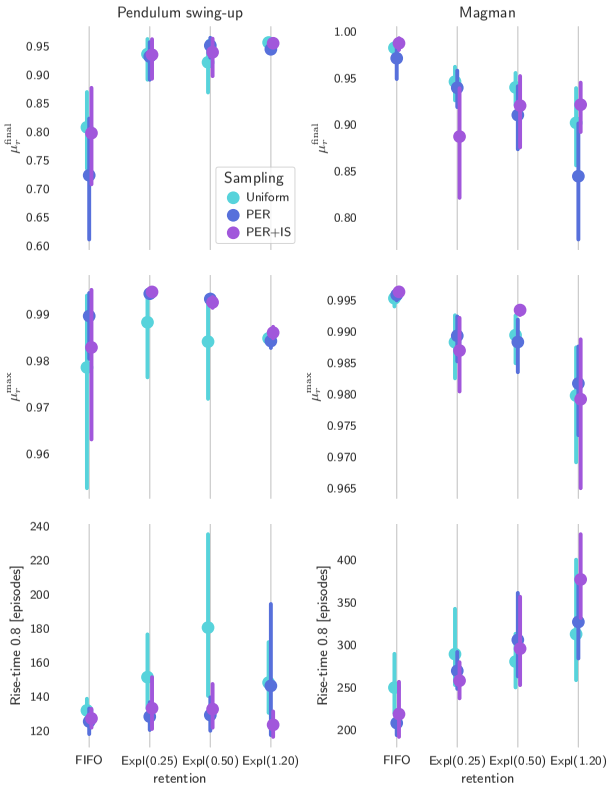


Figure 20: Influence of α in the Expl algorithm for different sampling strategies.

In Figure 21 it can again be seen that the benefits of PER are mostly to the speed of learning. Improvements to the maximum and final performance are possible when α is chosen correctly, but depend mostly on the contents of the buffer that PER is sampling from.

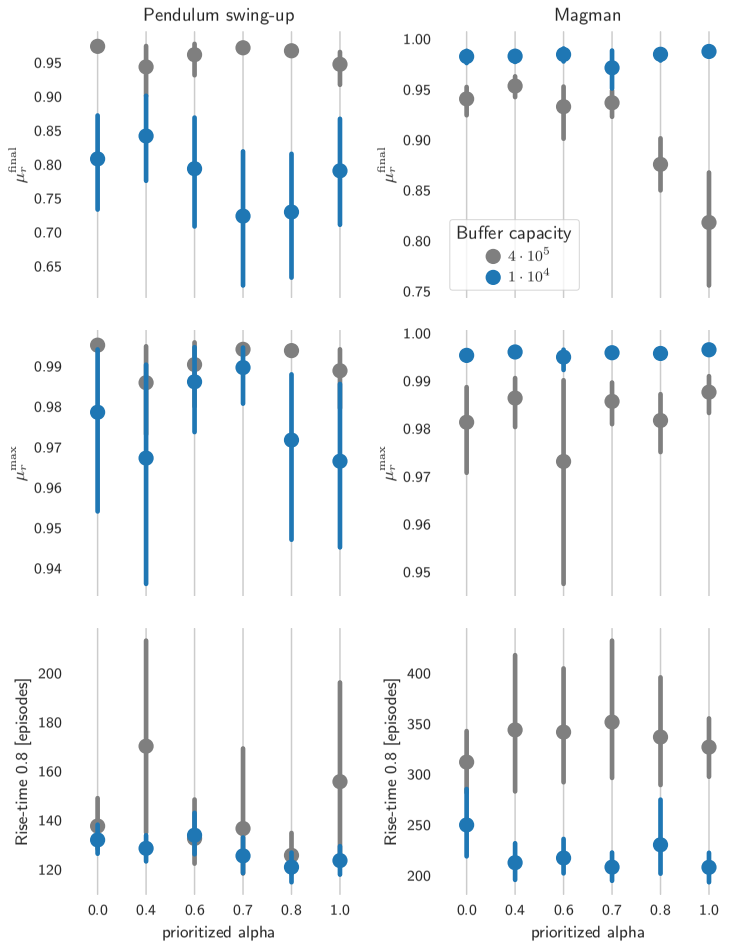


Figure 21: Influence of α in the PER algorithm for the Full DB strategy (buffer capacity $= 4 \times 10^5$) and FIFO retention (buffer capacity 1×10^{-4}).

D.3 Additional Figures Related to the Main Body

This subsection contains several figures that were left out of the main text of this work for brevity. They show the same experiments as Figures 6, 8, 16, according to the remaining performance criteria.

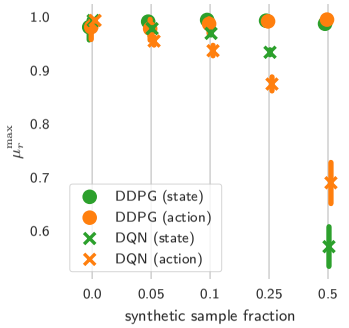


Figure 22: RL algorithm dependent effect of adding synthetic experiences to the FIFO[Uniform] method on the maximum performance per episode μ_r^{max} on the pendulum swing-up benchmark. The effect on the final performance and the rise-time is given in Figure 7.

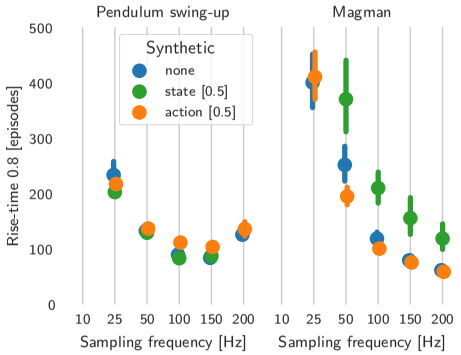
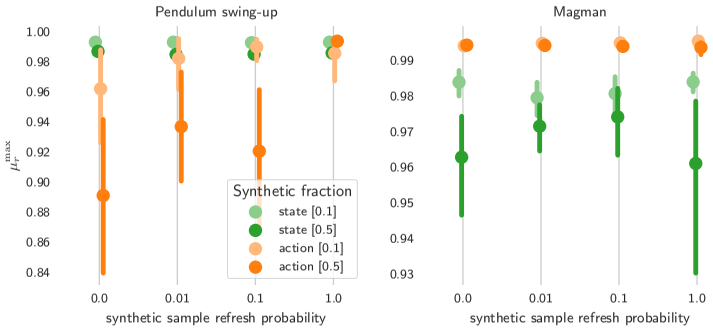
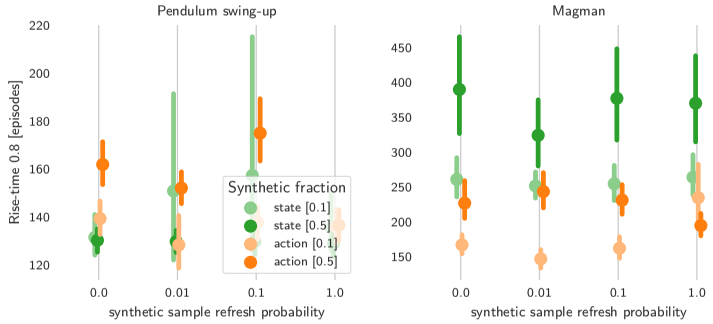


Figure 23: Sampling frequency dependent effect on the learning speed of adding synthetic experiences to the FIFO[Uniform] method. The effect on the final and maximum performance is given in Figure 9.



(a) Effect on $\mu_r^{\max}$.



(b) Effect on Rise − time0.8.

Figure 24: The effects on the performance of the FIFO[Uniform] method when changing a fraction of the observed experiences with synthetic experiences, when the synthetic experiences are updated only with a certain probability each time they are overwritten. The effects on μ_r^{final} is shown in Figure 8.

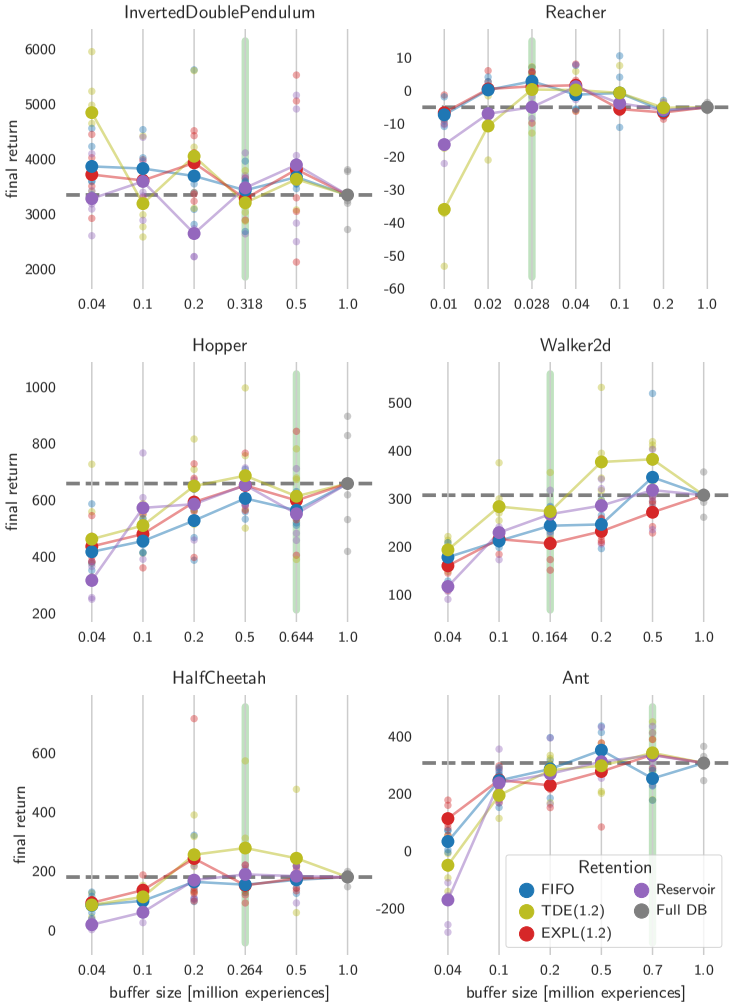


Figure 25: Mean performance during the last 2×10^5 training steps of a 1×10^6 step training run on the Roboschool benchmarks as a function of the retention strategy and buffer size. Results for the individual runs and their means are shown.

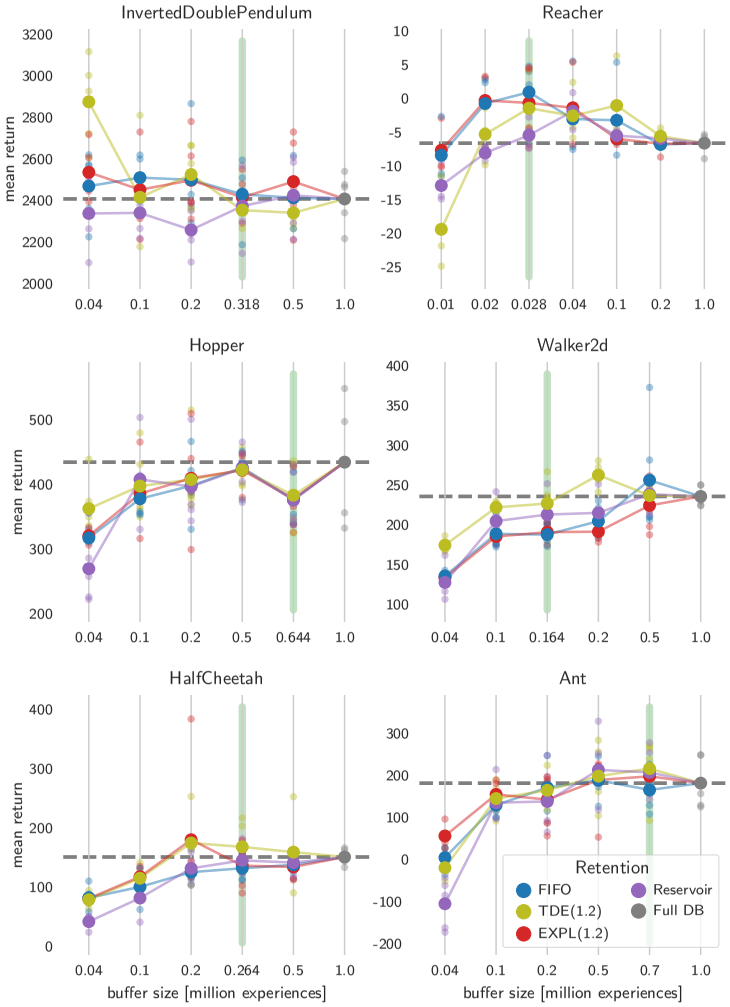


Figure 26: Mean performance during the whole training run on the Roboschool benchmarks as a function of the retention strategy and buffer size. Results for the individual runs and their means are shown.

References

Eduard Alibekov, Jiri Kubalík, and Robert Babuška. Policy derivation methods for critic-only reinforcement learning in continuous action spaces. *Engineering Applications of Artificial Intelligence*, 69:178–187, 2018.

David Andre, Nir Friedman, and Ronald Parr. Generalized prioritized sweeping. In *Advances In Neural Information Processing Systems (NIPS)*, pages 1001–1007. MIT Press, 1997.

John Aslanides, Jan Leike, and Marcus Hutter. Universal reinforcement learning algorithms: Survey and experiments. In *International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1403–1410, 2017.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

Leemon C Baird. Reinforcement learning in continuous time: Advantage updating. In *World Congress on Computational Intelligence (WCCI)*, volume 4, pages 2448–2453, 1994.

Bikramjit Banerjee and Jing Peng. Performance bounded reinforcement learning in strategic interactions. In *AAAI National Conference on Artificial Intelligence (AAAI)*, volume 4, pages 2–7, 2004.

Samuel Barrett, Matt Taylor, and Peter Stone. Transfer learning for reinforcement learning on a physical robot. Adaptive Learning Agents Workshop, International Conference on Autonomous Agents and Multiagent Systems (AAMAS - ALA), 2010.

Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1471–1479, 2016.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *International Conference on Machine Learning (ICML)*, pages 41–48, 2009.

Lucian Buşoniu, Damien Ernst, Robert Babuška, and Bart De Schutter. Approximate dynamic programming with a fuzzy parameterization. *Automatica*, 46(5):804–814, 2010.

Wouter Caarls and Erik Schuitema. Parallel online temporal difference learning for motor control. *IEEE Transactions on Neural Networks and Learning Systems*, 27(7):1457–1468, 2016.

Nuttapong Chentanez, Andrew G Barto, and Satinder P Singh. Intrinsically motivated reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1281–1288, 2004.

Kamil Ciosek and Shimon Whiteson. OFFER: off-environment reinforcement learning. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2017.

Ronan Collobert, Koray Kavukcuoglu, and Clément Farabet. Torch7: A Matlab-like environment for machine learning. BigLearn Workshop, Advances in Neural Information Processing Systems (NIPS - BLWS), 2011.

Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. The importance of experience replay database composition in deep reinforcement learning. Deep Reinforcement Learning Workshop, Advances in Neural Information Processing Systems (NIPS - DRLWS), 2015.

Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Improved deep reinforcement learning for robotics through distribution-based experience retention. In *International Conference on Intelligent Robots and Systems (IROS)*, 2016a.

Tim de Bruin, Jens Kober, Karl Tuyls, and Robert Babuška. Off policy experience retention for deep actor critic learning. Deep Reinforcement Learning Workshop, Advances in Neural Information Processing Systems (NIPS - DRLWS), 2016b.

Prafulla Dhariwal, Christopher Hesse, Oleg Klimov, Alex Nichol, Matthias Plappert, Alec Radford, John Schulman, Szymon Sidor, and Yuhuai Wu. OpenAI Baselines. `https://github.com/openai/baselines`, 2017.

Tobias Domhan, Jost Tobias Springenberg, and Frank Hutter. Speeding up automatic hyperparameter optimization of deep neural networks by extrapolation of learning curves. In *International Joint Conference on Artificial Intelligence (IJCAI)*, volume 15, pages 3460–3468, 2015.

Bradley Efron. Bootstrap methods: Another look at the jackknife. In *Breakthroughs in Statistics*, pages 569–593. Springer, 1992.

Vincent François-Lavet, Raphael Fonteneau, and Damien Ernst. How to discount deep reinforcement learning: Towards new dynamic strategies. arXiv preprint arXiv:1512.02011, 2015.

Gene F Franklin, David J Powell, and Michael L Workman. *Digital Control of Dynamic Systems*, volume 3. Addison-Wesley Menlo Park, 1998.

Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, 14(771-780):1612, 1999.

Javier Garcıa and Fernando Fernández. A comprehensive survey on safe reinforcement learning. *Journal of Machine Learning Research*, 16(1):1437–1480, 2015.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014.

Ian J Goodfellow, Mehdi Mirza, Xiao Da, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgeting in gradient-based neural networks. arXiv preprint arXiv:1312.6211, 2013.

Shixiang Gu, Timothy Lillicrap, Ilya Sutskever, and Sergey Levine. Continuous deep Q-learning with model-based acceleration. arXiv preprint arXiv:1603.00748, 2016.

Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine. Q-prop: sample-efficient policy gradient with an off-policy critic. In *International Conference on Learning Representations (ICLR)*, 2017.

Geoffrey E Hinton. To recognize shapes, first learn to generate images. *Progress in Brain Research*, 165:535–547, 2007.

Rein Houthooft, Xi Chen, Yan Duan, John Schulman, Filip De Turck, and Pieter Abbeel. VIME: Variational information maximizing exploration. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1109–1117, 2016.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations (ICLR)*, 2015.

Oleg Klimov. OpenAI Roboschool. `https://github.com/openai/roboschool`, 2017.

Jens Kober, J. Andrew Bagnell, and Jan Peters. Reinforcement learning in robotics: a survey. *International Journal of Robotics Research (IJRR)*, 32(11):1238–1274, 2013.

Ivan Koryakovskiy, Heike Vallery, Robert Babuška, and Wouter Caarls. Evaluation of physical damage associated with action selection strategies in reinforcement learning. In *IFAC World Congress*, 2017.

Leonid Kuvayev and Richard S Sutton. Model-based reinforcement learning with an approximate, learned model. Yale Workshop on Adaptive Learning Systems, 1996.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(39):1–40, 2016.

Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. Continuous control with deep reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2016.

Long-Ji Lin. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Machine Learning*, 8(3-4):293–321, 1992.

Zachary C Lipton, Jianfeng Gao, Lihong Li, Xiujun Li, Faisal Ahmed, and Li Deng. Efficient exploration for dialogue policy learning with BBQ networks & replay buffer spiking. arXiv preprint arXiv:1608.05081, 2016.

Ilya Loshchilov and Frank Hutter. Online batch selection for faster training of neural networks. arXiv preprint arXiv:1511.06343, 2015.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.

Grégoire Montavon, Geneviève B. Orr, and Klaus-Robert Müller, editors. *Neural Networks: Tricks of the Trade*. Lecture Notes in Computer Science (LNCS). Springer, 2nd edition, 2012.

Guido F Montufar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2924–2932, 2014.

Andrew W Moore and Christopher G Atkeson. Prioritized sweeping: reinforcement learning with less data and less time. *Machine Learning*, 13(1):103–130, 1993.

Karthik Narasimhan, Tejas Kulkarni, and Regina Barzilay. Language understanding for text-based games using deep reinforcement learning. In *Empirical Methods in Natural Language Processing (EMNLP)*, 2015.

Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Mathematical Programming*, 155 (1-2):549–573, 2016.

Brendan O'Donoghue, Remi Munos, Koray Kavukcuoglu, and Volodymyr Mnih. Combining policy gradient and Q-learning. In *International Conference on Learning Representations (ICLR)*, 2017.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped DQN. In *Advances In Neural Information Processing Systems (NIPS)*, pages 4026–4034, 2016.

Mathijs Pieters and Marco A Wiering. Q-learning with experience replay in a dynamic environment. In *Symposium Series on Computational Intelligence (SSCI)*, pages 1–8. IEEE, 2016.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y. Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. In *International Conference on Learning Representations (ICLR)*, 2018.

Aravind Rajeswaran, Kendall Lowrey, Emanuel V Todorov, and Sham M Kakade. Towards generalization and simplicity in continuous control. In *Advances In Neural Information Processing Systems (NIPS)*, pages 6550–6561, 2017.

Andrei A Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. arXiv preprint arXiv:1610.04286, 2016.

Stefan Schaal. Is imitation learning the route to humanoid robots? *Trends in Cognitive Sciences*, 3(6):233–242, 1999.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. Prioritized experience replay. In *International Conference on Learning Representations (ICLR)*, 2016.

Jürgen Schmidhuber. A possibility for implementing curiosity and boredom in model-building neural controllers. In *From Animals to Animats: International Conference on Simulation of Adaptive Behavior (SAB)*, 1991.

Young-Woo Seo and Byoung-Tak Zhang. Learning user's preferences by analyzing web-browsing behaviors. In *International Conference on Autonomous Agents (ICAA)*, pages 381–387, 2000.

David Silver, Guy Lever, Nicolas Heess, Thomas Degris, Daan Wierstra, and Martin Riedmiller. Deterministic policy gradient algorithms. In *International Conference on Machine Learning (ICML)*, 2014.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al. Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Burrhus F Skinner. Reinforcement today. *American Psychologist*, 13(3):94, 1958.

Richard S Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM SIGART Bulletin*, 2(4):160–163, 1991.

Aviv Tamar, Yinlam Chow, Mohammad Ghavamzadeh, and Shie Mannor. Sequential decision making with coherent risk. *IEEE Transactions on Automatic Control*, 2016.

George E Uhlenbeck and Leonard S Ornstein. On the theory of the Brownian motion. *Physical Review*, 36(5):823, 1930.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Jeffrey S Vitter. Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57, 1985.

Ziyu Wang, Victor Bapst, Nicolas Heess, Volodymyr Mnih, Remi Munos, Koray Kavukcuoglu, and Nando de Freitas. Sample efficient actor-critic with experience replay. In *International Conference on Learning Representations (ICLR)*, 2017.

Journal of Machine Learning Research 19 (2018) 1-37 Submitted 4/17; Revised 6/18; Published 8/18

A Constructive Approach to L_0 Penalized Regression

Jian Huang J.HUANG@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hung Hom, Kowloon
Hong Kong, China

Yuling Jiao* YULINGJIAOMATH@WHU.EDU.CN
School of Statistics and Mathematics
Zhongnan University of Economics and Law
Wuhan, 430063, China

Yanyan Liu LIUYY@WHU.EDU.CN
School of Mathematics and Statistics
Wuhan University
Wuhan, 430072, China

Xiliang Lu† XLLV.MATH@WHU.EDU.CN
School of Mathematics and Statistics
Wuhan University
Wuhan, 430072, China

Editor: Tong Zhang

Abstract

We propose a constructive approach to estimating sparse, high-dimensional linear regression models. The approach is a computational algorithm motivated from the KKT conditions for the ℓ_0-penalized least squares solutions. It generates a sequence of solutions iteratively, based on support detection using primal and dual information and root finding. We refer to the algorithm as SDAR for brevity. Under a sparse Riesz condition on the design matrix and certain other conditions, we show that with high probability, the ℓ_2 estimation error of the solution sequence decays exponentially to the minimax error bound in $O(\log(R\sqrt{J}))$ iterations, where J is the number of important predictors and R is the relative magnitude of the nonzero target coefficients; and under a mutual coherence condition and certain other conditions, the ℓ_∞ estimation error decays to the optimal error bound in $O(\log(R))$ iterations. Moreover the SDAR solution recovers the oracle least squares estimator within a finite number of iterations with high probability if the sparsity level is known. Computational complexity analysis shows that the cost of SDAR is $O(np)$ per iteration. We also consider an adaptive version of SDAR for use in practical applications where the true sparsity level is unknown. Simulation studies demonstrate that SDAR outperforms Lasso, MCP and two greedy methods in accuracy and efficiency.

Keywords: Geometrical convergence, KKT conditions, nonasymptotic error bounds, oracle property, root finding, support detection

∗. Also in the Institute of Big Data of Zhongnan University of Economics and Law
†. Also in the Hubei Key Laboratory of Computational Science

1. Introduction

Consider the linear regression model

$$y = X\beta^* + \eta \tag{1}$$

where $y \in \mathbb{R}^n$ is a response vector, $X \in \mathbb{R}^{n\times p}$ is the design matrix with $\sqrt{n}$-normalized columns, $\beta^* = (\beta_1^*, \ldots, \beta_p^*)' \in \mathbb{R}^p$ is the vector of the underlying regression coefficients and $\eta \in \mathbb{R}^n$ is a vector of random noises. We focus on the case where $p \gg n$ and the model is sparse in the sense that only a relatively small number of predictors are important. Without any constraints on β^* there exist infinitely many least squares solutions for (1) since it is a highly undetermined linear system when $p \gg n$. These solutions usually over-fit the data. Under the assumption that β^* is sparse in the sense that the number of important nonzero elements of β^* is small relative to n, we can estimate β^* by the solution of the ℓ_0 minimization problem

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\|X\beta - y\|_2^2, \quad \text{subject to} \quad \|\beta\|_0 \le s, \tag{2}$$

where $s > 0$ controls the sparsity level. However, (2) is generally NP hard (Natarajan, 1995; Chen et al., 2014), hence it is not tractable to design a stable and fast algorithm to solve it, especially in high-dimensional settings.

In this paper we propose a constructive approach to approximating the ℓ_0-penalized solution to (1). The approach is a computational algorithm motivated from the necessary KKT conditions for the Lagrangian form of (2). It finds an approximate sequence of solutions to the KKT equations iteratively based on support detection and root finding until convergence is achieved. For brevity, we refer to the proposed approach as SDAR.

1.1 Literature review

Several approaches have been proposed to approximate (2). Among them the Lasso (Tibshirani, 1996; Chen et al., 1998), which uses the ℓ_1 norm of β in the constraint instead of the ℓ_0 norm in (2), is a popular method. Under the irrepresentable condition on the design matrix X and a sparsity assumption on β^*, Lasso is model selection (and sign) consistent (Meinshausen and Bühlmann, 2006; Zhao and Yu, 2006; Wainwright, 2009). Lasso is a convex minimization problem. Several fast algorithms have been proposed, including LARS (Homotopy) (Osborne et al., 2000; Efron et al., 2004; Donoho and Tsaig, 2008), coordinate descent (Fu, 1998; Friedman et al., 2007; Wu and Lange, 2008), and proximal gradient descent (Agarwal et al., 2012; Xiao and Zhang, 2013; Nesterov, 2013).

However, Lasso tends to overshrink large coefficients, which leads to biased estimates (Fan and Li, 2001; Fan and Peng, 2004). The adaptive Lasso proposed by Zou (2006) and analyzed by Huang et al. (2008b) in high-dimensions can achieve the oracle property under certain conditions, but its requirements on the minimum value of the nonzero coefficients are not optimal. Nonconvex penalties such as the smoothly clipped absolute deviation (SCAD) penalty (Fan and Li, 2001), the minimax concave penalty (MCP) (Zhang, 2010a) and the capped ℓ_1 penalty (Zhang, 2010b) were proposed to remedy these problems (but these methods still require a minimum signal strength in order to achieve support recovery).

Although the global minimizers (also certain local minimizers) of these nonconvex regularized models can eliminate the estimation bias and enjoy the oracle property (Zhang and Zhang, 2012), computing the global or local minimizers with the desired statistical properties is challenging, since the optimization problem is nonconvex, nonsmooth and large scale in general.

There are several numerical algorithms for nonconvex regularized problems. The first kind of such methods can be considered a special case (or variant) of minimization maximization algorithm (Lange et al., 2000; Hunter and Li, 2005) or of multi-stage convex relaxation (Zhang, 2010b). Examples include local quadratic approximation (LQA) (Fan and Li, 2001), local linear approximation (LLA) (Zou and Li, 2008), decomposing the penalty into a difference of two convex terms (CCCP) (Kim et al., 2008; Gasso et al., 2009). The second type of methods is the coordinate descent algorithms, including coordinate descent of the Gauss-Seidel version (Breheny and Huang, 2011; Mazumder et al., 2011) and coordinate descent of the Jacobian version, i.e., the iterative thresholding method (Blumensath and Davies, 2008; She, 2009). These algorithms generate a sequence of solutions at which the objective functions are nonincreasing, but the convergence of the sequence itself is generally unknown. Moreover, if the sequence generated from multi-stage convex relaxation (starts from a Lasso solution) converges, it converges to some stationary point which may enjoy certain oracle statistical properties with the cost of a Lasso solver per iteration (Zhang, 2010b; Fan et al., 2014). Huang et al. (2018) proposed a globally convergent primal dual active set algorithm for a class of nonconvex regularized problems. Recently, there has been much effort to show that CCCP, LLA and the path following proximal-gradient method can track the local minimizers with the desired statistical properties (Wang et al., 2013; Fan et al., 2014; Wang et al., 2014; Loh and Wainwright, 2015).

Another line of research concerns the greedy methods such as the orthogonal matching pursuit (OMP) (Mallat and Zhang, 1993) for solving (2) approximately. The main idea is to iteratively select one variable with the strongest correlation with the current residual at a time. Roughly speaking, the performance of OMP can be guaranteed if the small submatrices of X are well conditioned like orthogonal matrices (Tropp, 2004; Donoho et al., 2006; Cai and Wang, 2011; Zhang, 2011b). Fan and Lv (2008) proposed a marginal correlation learning method called sure independence screening (SIS), see also Huang et al. (2008a) for an equivalent formulation that uses penalized univariate regression for screening. Fan and Lv (2008) recommended an iterative SIS to improve the finite-sample performance. As they discussed the iterative SIS also uses the core idea of OMP but it can select more features at each iteration. There are several more recently developed greedy methods aimed at selecting several variables a time or removing variables adaptively, such as iterative hard thresholding (IHT) (Blumensath and Davies, 2009; Jain et al., 2014) or hard thresholding gradient descent (GraDes) (Garg and Khandekar, 2009), adaptive forward-backward selection (FoBa) (Zhang, 2011a).

Liu and Wu (2007) proposed a Mixed Integer Optimization (MIO) approach for solving penalized classification and regression problems with a penalty that is a combination of ℓ_0 and ℓ_1 penalties. However, they only considered low-dimensional problems with p in the 10s and n in the 100s. Bertsimas et al. (2016) also considered an MIO approach for solving the best subset selection problem in linear regression with a possible side constraint. Their approach can solve problems with moderate sample sizes and moderate dimensions in minutes, for example, for $(n, p) \approx (100, 1000)$ or $(n, p) \approx (1000, 100)$. For the $p > n$ examples, the authors carried out all the computations on Columbia University's high performance computing facility using a commercial MIO solver GUROBI (Gurobi Optimization, 2015). In comparison, our proposed approach can deal with high-dimensional models. For the examples we consider in our simulation studies with $(n, p) = (5000, 50000)$, it can find the solution in seconds on a personal laptop computer.

1.2 Contributions

SDAR is a new approach for fitting sparse, high-dimensional regression models. Compared with the penalized methods, SDAR generates a sequence of solutions $\{\beta^k, k \geq 1\}$ to the KKT system of the ℓ_0 penalized criterion, which can be viewed as a primal-dual active set method for solving the ℓ_0 regularized least squares problem with a changing regularization parameter λ in each iteration (this will be explained in detail in Section 2).

We show that SDAR achieves sharp estimation error bounds within a finite number of iterations. Specifically, we show that: (a) under a sparse Riesz condition on X and a sparsity assumption on β^*, $\|\beta^k - \beta^*\|_2$ achieves the minimax error bound up to a constant factor with high probability in $O(\sqrt{J}\log(R))$ iterations, where J is the number of important predictors and R is the relative magnitude of the nonzero target coefficients (the exact definitions of J and R are given in Section 3); (b) under a mutual coherence condition on X and a sparsity assumption on β^*, the $\|\beta^k - \beta^*\|_\infty$ achieves the optimal error bound $O(\sigma\sqrt{\log(p)/n})$ in $O(\log(R))$ iterations; (c) under the conditions in (a) and (b), with high probability, β^k coincides with the oracle least squares estimator in $O(\sqrt{J}\log(R))$ and $O(\log(R))$ iterations, respectively, if J is available and the minimum magnitude of the nonzero elements of β^* is of the order $O(\sigma\sqrt{2\log(p)/n})$, which is the optimal magnitude of detectable signal.

An interesting aspect of the result in (b) is that the number of iterations for SDAR to achieve the optimal error bound is $O(\log(R))$, which does not depend on the underlying sparsity level. This is an appealing feature for the problems with a large triple (n, p, J). We also analyze the computational cost of SDAR and show that it is $O(np)$ per iteration, comparable to the existing penalized methods and the greedy methods.

In summary, the main contributions of this paper are as follows.

- We propose a new approach to fitting sparse, high-dimensional regression models. The approach seeks to directly approximate the solutions to the KKT equations for the ℓ_0 penalized problem.

- We show that the sequence of solutions $\{\beta^k, k \geq 1\}$ generated by the SDAR achieves sharp error bounds within a finite number of iterations.

- We also consider an adaptive version of SDAR, or simply ASDAR, by tuning the size of the fitted model based on a data driven procedure such as the BIC. Our simulation studies demonstrate that SDAR/ASDAR outperforms the Lasso, MCP and several greedy methods in terms of accuracy and efficiency in the generating models we considered.

1.3 Notation

For a column vector $\beta = (\beta_1, \ldots, \beta_p)' \in \mathbb{R}^p$, denote its q-norm by $\|\beta\|_q = (\sum_{i=1}^p |\beta_i|^q)^{1/q}, q \in [1, \infty]$, and its number of nonzero elements by $\|\beta\|_0$. Let $\mathbf{0}$ denote a column vector in $\mathbb{R}^p$ or a matrix whose elements are all 0. Let $S = \{1, 2, ..., p\}$. For any A and $B \subseteq S$ with length $|A|$ and $|B|$, let $\beta_A = (\beta_i, i \in A) \in \mathbb{R}^{|A|}$, $X_A = (X_i, i \in A) \in \mathbb{R}^{n\times|A|}$, and let $X_{AB} \in \mathbb{R}^{|A|\times|B|}$ be a submatrix of X whose rows and columns are listed in A and B, respectively. Let $\beta|_A \in \mathbb{R}^p$ be a vector with its i-th element $(\beta|_A)_i = \beta_i \mathbf{1}(i \in A)$, where $\mathbf{1}(\cdot)$ is the indicator function. Denote the support of β by $\text{supp}(\beta)$. Denote $A^* = \text{supp}(\beta^*)$ and $K = \|\beta^*\|_0$. Let $\|\beta\|_{k,\infty}$ and $|\beta|_{\min}$ be the kth largest elements (in absolute value) and the minimum absolute value of β, respectively. Denote the operator norm of X induced by the vector 2-norm by $\|X\|$. Let $\mathbb{I}$ be an identity matrix.

1.4 Organization

In Section 2 we develop the SDAR algorithm based on the necessary conditions for the ℓ_0 penalized solutions. In Section 3 we establish the nonasymptotic error bounds of the SDAR solutions. In Section 4 we describe the adaptive SDAR, or ASDAR. In Section 5 we analyze the computational complexity of SDAR and ASDAR. In Section 6 we compare SDAR with several greedy methods and a screening method. In Section 7 we conduct simulation studies to evaluate the performance of SDAR/ASDAR and compare it with Lasso, MCP, FoBa and DesGras. We conclude in Section 8 with some final remarks. The proofs are given in the Appendix.

2. Derivation of SDAR

Consider the Lagrangian form of the ℓ_0 regularized minimization problem (2),

$$\min_{\beta\in\mathbb{R}^p} \frac{1}{2n}\|X\beta - y\|_2^2 + \lambda\|\beta\|_0. \tag{3}$$

Lemma 1 *Let $\beta^\diamond$ be a coordinate-wise minimizer of (3). Then $\beta^\diamond$ satisfies:*

$$\begin{cases} d^\diamond = X'(y - X\beta^\diamond)/n, \\ \beta^\diamond = H_\lambda(\beta^\diamond + d^\diamond), \end{cases} \tag{4}$$

where $H_\lambda(\cdot)$ is the hard thresholding operator defined by

$$(H_\lambda(\beta))_i = \begin{cases} 0, & \text{if } |\beta_i| < \sqrt{2\lambda}, \\ \beta_i, & \text{if } |\beta_i| \geq \sqrt{2\lambda}. \end{cases} \tag{5}$$

Conversely, if $\beta^\diamond$ and $d^\diamond$ satisfy (4)*, then $\beta^\diamond$ is a local minimizer of* (3)*.*

Remark 2 *Lemma 1 gives the KKT condition of the ℓ_0 regularized minimization problem (3), which is also derived in Jiao et al. (2015). Similar results for SCAD, MCP and capped-ℓ_1 regularized least squares models can be derived by replacing the hard thresholding operator in* (4) *with their corresponding thresholding operators, see Huang et al. (2018) for details.*

Let $A^\diamond = \text{supp}(\beta^\diamond)$ and $I^\diamond = (A^\diamond)^c$. Suppose that the rank of $X_{A^\diamond}$ is $|A^\diamond|$. From the definition of $H_\lambda(\cdot)$ and (4) it follows that

$$A^\diamond = \left\{i \in S \middle| \, |\beta_i^\diamond + d_i^\diamond| \geq \sqrt{2\lambda}\right\}, \quad I^\diamond = \left\{i \in S \middle| \, |\beta_i^\diamond + d_i^\diamond| < \sqrt{2\lambda}\right\},$$

and

$$\begin{cases} \beta_{I^\diamond}^\diamond = \mathbf{0}, \\ d_{A^\diamond}^\diamond = \mathbf{0}, \\ \beta_{A^\diamond}^\diamond = (X'_{A^\diamond}X_{A^\diamond})^{-1}X'_{A^\diamond}y, \\ d_{I^\diamond}^\diamond = X'_{I^\diamond}(y - X_{A^\diamond}\beta_{A^\diamond}^\diamond)/n. \end{cases}$$

We solve this system of equations iteratively. Let $\{\beta^k, d^k\}$ be the solution at the kth iteration. We approximate $\{A^\diamond, I^\diamond\}$ by

$$A^k = \left\{i \in S \middle| |\beta_i^k + d_i^k| \geq \sqrt{2\lambda}\right\}, \quad I^k = (A^k)^c. \tag{6}$$

Then we can obtain an updated approximation pair $\{\beta^{k+1}, d^{k+1}\}$ by

$$\begin{cases} \beta_{I^k}^{k+1} = \mathbf{0}, \\ d_{A^k}^{k+1} = \mathbf{0}, \\ \beta_{A^k}^{k+1} = (X'_{A^k}X_{A^k})^{-1}X'_{A^k}y, \\ d_{I^k}^{k+1} = X'_{I^k}(y - X_{A^k}\beta_{A^k}^{k+1})/n. \end{cases} \tag{7}$$

Now suppose we want the support of the solutions to have the size T, where $T \geq 1$ is a given integer. We can choose

$$\sqrt{2\lambda^k} \triangleq \|\beta^k + d^k\|_{T,\infty} \tag{8}$$

in (6). With this choice of λ, we have $|A^k| = T, k \geq 1$. Then with an initial β^0 and using (6) and (7) with the λ^k in (8), we obtain a sequence of solutions $\{\beta^k, k \geq 1\}$.

There are two key aspects of SDAR. In (6) we detect the support of the solution based on the sum of the primal (β^k) and dual (d^k) approximations and, in (7) we calculate the least squares solution on the detected support. Therefore, SDAR can be considered an iterative method for solving the KKT equations (4) with an important modification: a different λ value given in (8) in each step of the iteration is used. Thus we can also view SDAR as a method that combines adaptive thresholding using primal and dual information and least-squares fitting. We summarize SDAR in Algorithm 1.

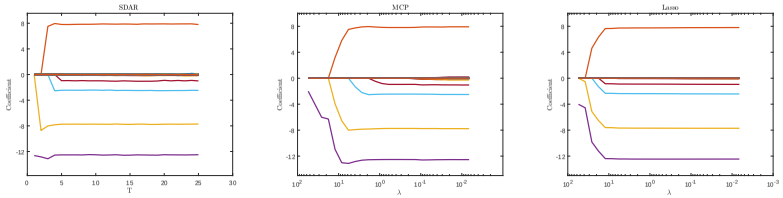


Figure 1: The solution paths of SDAR, MCP and Lasso. We see that large components were selected in by SDAR gradually when T increases. This is similar to Lasso and MCP as λ decreases.

As an example, Figure 1 shows the solution path of SDAR with $T = 1, 2, \ldots, 5K$ along with the MCP and the Lasso paths on $5K$ different λ values for a data set generated from a model with $(n = 50, p = 100, K = 5, \sigma = 0.3, \rho = 0.5, R = 10)$, which will be described in Section 7. The Lasso path is computed using LARS (Efron et al., 2004). Note that the SDAR path is a function of the fitted model size $T = 1, \ldots, L$, where L is the size of the largest fitted model. In comparison, the paths of MCP and Lasso are functions of the penalty parameter λ in a prespecified interval. In this example, when $T \leq K$, SDAR selects the first T largest components of β^* correctly. When $T > K$, there will be spurious elements included in the estimated support, the exact number of such elements is $T - K$. In Figure 1, the estimated coefficients of the spurious elements are close to zero.

Algorithm 1 Support detection and root finding (SDAR)

Require: β^0, $d^0 = X'(y - X\beta^0)/n$, T; set $k = 0$.
1: **for** $k = 0, 1, 2, \cdots$ **do**
2: $A^k = \{i \in S \big| |\beta_i^k + d_i^k| \geq \|\beta^k + d^k\|_{T,\infty}\}$, $I^k = (A^k)^c$
3: $\beta_{I^k}^{k+1} = \mathbf{0}$
4: $d_{A^k}^{k+1} = \mathbf{0}$
5: $\beta_{A^k}^{k+1} = (X'_{A^k} X_{A^k})^{-1} X'_{A^k} y$
6: $d_{I^k}^{k+1} = X'_{I^k}(y - X_{A^k}\beta_{A^k}^{k+1})/n$
7: **if** $A^{k+1} = A^k$, **then**
8: Stop and denote the last iteration by $\beta_{\hat{A}}, \beta_{\hat{I}}, d_{\hat{A}}, d_{\hat{I}}$
9: **else**
10: $k = k + 1$
11: **end if**
12: **end for**
Ensure: $\hat{\beta} = (\beta'_{\hat{A}}, \beta'_{\hat{I}})'$ as the estimate of β^*.

Remark 3 *If $A^{k+1} = A^k$ for some k we stop SDAR since the sequences generated by SDAR will not change. Under certain conditions, we will show that $A^{k+1} = A^k = supp(\beta^*)$ if k is large enough, i.e., the stop condition in SDAR will be active and the output is the oracle estimator when it stops.*

3. Nonasymptotic error bounds

In this section we present the nonasymptotic ℓ_2 and ℓ_∞ error bounds for the solution sequence generated by SDAR as given in Algorithm 1.

We say that X satisfies the sparse Rieze condition (SRC) (Zhang and Huang, 2008; Zhang, 2010a) with order s and spectrum bounds $\{c_-(s), c_+(s)\}$ if

$$0 < c_-(s) \leq \frac{\|X_A u\|_2^2}{n\|u\|_2^2} \leq c_+(s) < \infty, \forall\, 0 \neq u \in \mathbb{R}^{|A|} \text{ with } A \subset S \text{ and } |A| \leq s.$$

We denote this condition by $X \sim \text{SRC}\{s, c_-(s), c_+(s)\}$. The SRC gives the range of the spectrum of the diagonal sub-matrices of the Gram matrix $G = X'X/n$. The spectrum of the off diagonal sub-matrices of G can be bounded by the sparse orthogonality constant $\theta_{a,b}$ defined as the smallest number such that

$$\theta_{a,b} \geq \frac{\|X'_A X_B u\|_2}{n\|u\|_2}, \forall\, 0 \neq u \in \mathbb{R}^{|B|} \text{ with } A, B \subset S, |A| \leq a, |B| \leq b, \text{ and } A \cap B = \emptyset.$$

Another useful quantity is the mutual coherence μ defined as $\mu = \max_{i \neq j} |G_{i,j}|$, which characterizes the minimum angle between different columns of $X/\sqrt{n}$. Some useful properties of these quantities are summarized in Lemma 20 in the Appendix.

In addition to the regularity conditions on the design matrix, another key condition is the sparsity of the regression parameter β^*. The usual sparsity condition is to assume that the regression parameter β_i^* is either nonzero or zero and that the number of nonzero coefficients is relatively small. This strict sparsity condition is not realistic in many problems. Here we allow that β^* may not be strictly sparse but most of its elements are small. Let $A_J^* = \{i \in S : |\beta_i^*| \geq \|\beta^*\|_{J,\infty}\}$ be the set of the indices of the first J largest components of β^*. Typically, we have $J \ll n$. Let

$$R = \frac{\bar{M}}{\bar{m}}, \tag{9}$$

where $\bar{m} = \min\{|\beta_i^*|, i \in A_J^*\}$ and $\bar{M} = \max\{|\beta_i^*|, i \in A_J^*\}$. Since $\beta^* = \beta^*|_{A_J^*} + \beta^*|_{(A_J^*)^c}$, we can transform the non-exactly sparse model (1) to the following exactly sparse model by including the small components of β^* in the noise,

$$y = X\bar{\beta}^* + \bar{\eta}, \tag{10}$$

where

$$\bar{\beta}^* = \beta^*|_{A_J^*} \text{ and } \bar{\eta} = X\beta^*|_{(A_J^*)^c} + \eta. \tag{11}$$

Let $R_J = \|\beta^*|_{(A_J^*)^c}\|_2 + \|\beta^*|_{(A_J^*)^c}\|_1/\sqrt{J}$, which is a measure of the magnitude of the small components of β^* outside A_J^*. Of course, $R_J = 0$ if β^* is exactly K-sparse with $K \leq J$. Without loss of generality, we let $J = K$, $m = \bar{m}$ and $M = \bar{M}$ for simplicity if β^* is exactly K-sparse.

Let $\beta^{J,o}$ be the oracle estimator defined as $\beta^{J,o} = \arg\min_\beta\{\frac{1}{2n}\|y - X\beta\|_2^2, \beta_j = 0, j \notin A_J^*\}$, that is, $\beta_{A_J^*}^{J,o} = X_{A_J^*}^\dagger y$ and $\beta_{(A_J^*)^c}^{J,o} = \mathbf{0}$, where $X_{A_J^*}^\dagger$ is the generalized inverse of $X_{A_J^*}$ and equals to $(X'_{A_J^*} X_{A_J^*})^{-1} X'_{A_J^*}$ if $X_{A_J^*}$ is of full column rank. So $\beta^{J,o}$ is obtained by keeping the predictors corresponding to the J largest components of β^* in the model and dropping the other predictors. Obviously, $\beta^{J,o} = \beta^o$ if β^* is exactly K-sparse, where $\beta_{A^*}^o = X_{A^*}^\dagger y, \beta_{(A^*)^c}^o = \mathbf{0}$.

3.1 ℓ_2 error bounds

Let $1 \le T \le p$ be a given integer used in Algorithm 1. We require the following basic assumptions on the design matrix X and the error vector η.

(A1) The input integer T used in Algorithm 1 satisfies $T \ge J$.

(A2) For the input integer T used in Algorithm 1, $X \sim \mathrm{SRC}\{2T, c_-(2T), c_+(2T)\}$.

(A3) The random errors $\eta_1, \ldots, \eta_n$ are independent and identically distributed with mean zero and sub-Gaussian tails, that is, there exists a $\sigma \ge 0$ such that $E[\exp(t\eta_i)] \le \exp(\sigma^2 t^2/2)$ for $t \in \mathbb{R}^1$, $i = 1, \ldots, n$.

Let

$$\gamma = \frac{2\theta_{T,T} + (1+\sqrt{2})\theta_{T,T}^2}{c_-(T)^2} + \frac{(1+\sqrt{2})\theta_{T,T}}{c_-(T)}.$$

Define

$$h_2(T) = \max_{A \subseteq S: |A| \le T} \|X_A' \bar{\eta}\|_2 / n, \tag{12}$$

where $\bar{\eta}$ is defined in (11).

Theorem 4 *Let T be the input integer used in Algorithm 1, where $1 \le T \le p$. Suppose $\gamma < 1$.*

(i) Assume (A1) and (A2) hold. We have

$$\|\bar{\beta}^*|_{A_J^* \setminus A^{k+1}}\|_2 \le \gamma^{k+1}\|\bar{\beta}^*\|_2 + \frac{\gamma}{(1-\gamma)\theta_{T,T}} h_2(T), \tag{13}$$

$$\|\beta^{k+1} - \bar{\beta}^*\|_2 \le b_1 \gamma^k \|\bar{\beta}^*\|_2 + b_2 h_2(T), \tag{14}$$

where

$$b_1 = 1 + \frac{\theta_{T,T}}{c_-(T)} \quad \text{and} \quad b_2 = \frac{\gamma}{(1-\gamma)\theta_{T,T}} b_1 + \frac{1}{c_-(T)}. \tag{15}$$

(ii) Assume (A1)-(A3) hold. Then for any $\alpha \in (0, 1/2)$, with probability at least $1 - 2\alpha$,

$$\|\bar{\beta}^*|_{A_J^* \setminus A^{k+1}}\|_2 \le \gamma^{k+1}\|\bar{\beta}^*\|_2 + \frac{\gamma}{(1-\gamma)\theta_{T,T}} \varepsilon_1, \tag{16}$$

$$\|\beta^{k+1} - \bar{\beta}^*\|_2 \le b_1 \gamma^k \|\bar{\beta}^*\|_2 + b_2 \varepsilon_1, \tag{17}$$

where

$$\varepsilon_1 = c_+(J) R_J + \sigma\sqrt{T}\sqrt{2\log(p/\alpha)/n}. \tag{18}$$

Remark 5 *Part (i) of Theorem 4 establishes the ℓ_2 bounds for the approximation errors of the solution sequence generated by the SDAR algorithm at the $(k+1)$th iteration for a general noise vector η. In particular, (13) gives the ℓ_2 bound of the elements in A_J^* not included in the active set in the $(k+1)$th iteration, and (14) provides an upper bound for the ℓ_2 estimation error of β^{k+1}. These error bounds decay geometrically to the model error measured by $h_2(T)$ up to a constant factor. Part (ii) specializes these results to the case where the noise terms are sub-Gaussian.*

Remark 6 *Assumption (A1) is necessary for SDAR to select at least J nonzero features. The SRC in (A2) has been used in the analysis of the Lasso and MCP (Zhang and Huang, 2008; Zhang, 2010a). Sufficient conditions are provided for a design matrix to satisfy the SRC in Propositions 4.1 and 4.2 in Zhang and Huang (2008). For example, the SRC would follow from a mutual coherence condition. Let $c(T) = (1 - c_-(2T)) \vee (c_+(2T) - 1)$, which is closely related to the the RIP (restricted isometry property) constant δ_{2T} for X (Candes and Tao, 2005). By (43) in the Appendix, it can be verified that a sufficient condition for $\gamma < 1$ is $c(T) \le 0.1599$, i.e., $c_+(2T) \le 1.1599$, $c_-(2T) \ge 0.8401$. The sub-Gaussian condition (A3) is often assumed in the literature on sparse estimation and slightly weaker than the standard normality assumption. It is used to calculate the tail probabilities of certain maximal functions of the noise vector η.*

Remark 7 *Several greedy algorithms have also been studied under the assumptions related to the sparse Riesz condition. For example, Zhang (2011b) studied OMP under the condition $c_+(T)/c_-(31T) \le 2$. Zhang (2011a) analyzed the forward-backward greedy algorithm (FoBa) under the condition $8(T+1) \le (s-2)Tc_-^2(sT)$, where $s > 0$ is a properly chosen parameter. GraDes has been analyzed under the RIP condition $\delta_{2T} \le 1/3$ (Garg and Khandekar, 2009). These conditions and (A2) are related but do not imply each other. The order of ℓ_2-norm estimation error of SDAR is at least as good as that of the above mentioned greedy methods since it achieves the minimax error bound, see, Remark 10 below. A high level comparison between SDAR and the greedy algorithms will be given in Section 6.*

Corollary 8 *(i) Suppose (A1) and (A2) hold. Then*

$$\|\beta^k - \bar{\beta}^*\|_2 \le c h_2(T) \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}} \frac{\sqrt{J}\bar{M}}{h_2(T)}, \tag{19}$$

where $c = b_1 + b_2$ with b_1 and b_2 defined in (15).

Furthermore, assume $\bar{m} \ge \frac{\gamma h_2(T)}{(1-\gamma)\theta_{T,T}\xi}$ for some $0 < \xi < 1$, then we have

$$A^k \supseteq A_J^* \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}} \frac{\sqrt{J}R}{1-\xi}. \tag{20}$$

(ii) Suppose (A1)-(A3) hold. Then, for any $\alpha \in (0, 1/2)$, with probability at least $1 - 2\alpha$, we have

$$\|\beta^k - \bar{\beta}^*\|_2 \le c\varepsilon_1 \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}} \frac{\sqrt{J}\bar{M}}{\varepsilon_1}, \tag{21}$$

where ε_1 is defined in (18). Furthermore, assume $\bar{m} \ge \frac{\varepsilon_1 \gamma}{(1-\gamma)\theta_{T,T}\xi}$ for some $0 < \xi < 1$, then with probability at least $1 - 2\alpha$, we have

$$A^k \supseteq A_J^* \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}} \frac{\sqrt{J}R}{1-\xi}. \tag{22}$$

(iii) Suppose β^* is exactly K-sparse. Let $T = K$ in SDAR. Suppose (A1)-(A3) hold and $m \ge \frac{\gamma}{(1-\gamma)\theta_{T,T}\xi}\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n}$ for some $0 < \xi < 1$, we have with probability

at least $1-2\alpha$, $A^k = A^{k+1} = A^*$ *if* $k \geq \log_{\frac{1}{\gamma}}(\sqrt{K}R/(1-\xi))$, *i.e., with at most* $O(\log\sqrt{K}R)$ *iterations, SDAR stops and the output is the oracle least squares estimator* β^o.

Remark 9 *Parts (i) and (ii) in Corollary 8 show that the SDAR solution sequence achieves the minimax* ℓ_2 *error bound up to a constant factor and its support covers* A^*_J *within a finite number of iterations. In particular, the number of iterations required is* $O(\log(\sqrt{J}R))$, *depending on the sparsity level* J *and the relative magnitude* R *of the coefficients of the important predictors. In the case of exact sparsity with* K *nonzero coefficients in the model, part (iii) provides conditions under which the SDAR solution is the same as the oracle least squares estimator in* $O(\log(\sqrt{K}R))$ *iterations with high probability.*

Remark 10 *Suppose* β^* *is exactly* K*-sparse. In the event* $\|\eta\|_2 \leq \varepsilon$, *part (i) of Corollary 8 implies* $\|\beta^k - \beta^*\|_2 = O(\varepsilon/\sqrt{n})$ *if* k *is sufficiently large. Under certain conditions on the RIP constant of* X, *Candes et al. (2006) showed that* $\|\hat{\beta} - \beta^*\|_2 = O(\varepsilon/\sqrt{n})$, *where* $\hat{\beta}$ *solves*

$$\min_{\beta\in\mathbb{R}^p} \|\beta\|_1 \text{ subject to } \|X\beta - y\|_2 \leq \varepsilon. \tag{23}$$

So the result here is similar to that of Candes et al. (2006) (they assumed the columns of X *are unit-length normalized, here the result is stated for the case where the columns of* X *are* $\sqrt{n}$*-length normalized). However, it is a nontrivial task to solve* (23) *in high-dimensional settings. In comparison, SDAR only involves simple computational steps.*

Remark 11 *If* β^* *is exactly* K*-sparse and* $T = K$, *part (ii) of Corollary 8 implies that SDAR achieves the minimax error bound (Raskutti et al., 2011), that is,*

$$\|\beta^k - \beta^*\|_2 \leq c\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n}$$

with high probability if $k \geq \log_{\frac{1}{\gamma}} \frac{\sqrt{K}M}{\sigma\sqrt{T}\sqrt{2\log(p/\alpha)/n}}$.

3.2 ℓ_∞ error bounds

We now consider the ℓ_∞ error bounds of SDAR. We replace condition (A2) by

(A2*) The mutual coherence μ of X satisfies $T\mu \leq 1/4$.

Let

$$\gamma_\mu = \frac{(1+2T\mu)T\mu}{1-(T-1)\mu} + 2T\mu \text{ and } c_\mu = \frac{16}{3(1-\gamma_\mu)} + \frac{5}{3}.$$

Define

$$h_\infty(T) = \max_{A\subseteq S:|A|\leq T} \|X'_A\bar{\eta}\|_\infty/n, \tag{24}$$

where $\bar{\eta}$ is defined in (11).

Theorem 12 *Let* T *be the input integer used in Algorithm 1, where* $1 \leq T \leq p$.

(i) Assume (A1) and (A2) hold. We have*

$$\|\bar{\beta}^*|_{A^*_J\setminus A^{k+1}}\|_\infty < \gamma_\mu^{k+1}\|\bar{\beta}^*\|_\infty + \frac{4}{1-\gamma_\mu}h_\infty(T), \tag{25}$$

$$\|\beta^{k+1} - \bar{\beta}^*\|_\infty < \frac{4}{3}\gamma_\mu^k\|\bar{\beta}^*\|_\infty + \frac{4}{3}(\frac{4}{1-\gamma_\mu}+1)h_\infty(T), \tag{26}$$

(ii) Assume (A1), (A2) and (A3) hold. For any* $\alpha \in (0,1/2)$, *with probability at least* $1-2\alpha$,

$$\|\bar{\beta}^*|_{A^*_J\setminus A^{k+1}}\|_\infty < \gamma_\mu^{k+1}\|\bar{\beta}^*\|_\infty + \frac{4}{1-\gamma_\mu}\varepsilon_2, \tag{27}$$

$$\|\beta^{k+1} - \bar{\beta}^*\|_\infty < \frac{4}{3}\gamma_\mu^k\|\bar{\beta}^*\|_\infty + \frac{4}{3}(\frac{4}{1-\gamma_\mu}+1)\varepsilon_2, \tag{28}$$

where

$$\varepsilon_2 = (1+(T-1)\mu)R_J + \sigma\sqrt{2\log(p/\alpha)/n}. \tag{29}$$

Remark 13 *Part (i) of Theorem 12 establishes the* ℓ_∞ *bounds for the approximation errors of the solution sequence at the* $(k+1)$*th iteration for a general noise vector* η. *In particular, (25) gives the* ℓ_∞ *bound of the elements in* A^*_J *not selected at the* $(k+1)$*th iteration, and (26) provides an upper bound for the* ℓ_∞ *estimation error of* β^{k+1}. *These errors bounds decay geometrically to the model error measured by* $h_\infty(T)$ *up to a constant factor. Part (ii) specializes these to the case where the noise terms are sub-Gaussian.*

Corollary 14 *(i) Suppose (A1) and (A2*) hold. Then*

$$\|\beta^k - \bar{\beta}^*\|_\infty \leq c_\mu h_\infty(T) \quad \text{if} \quad k \geq \log_{\frac{1}{\gamma_\mu}} \frac{4\bar{M}}{h_\infty(T)}. \tag{30}$$

Furthermore, assume $\bar{m} \geq \frac{4h_\infty(T)}{(1-\gamma_\mu)\xi}$ *with* $\xi < 1$, *then we have*

$$A^k \supseteq A^*_J \quad \text{if} \quad k \geq \log_{\frac{1}{\gamma_\mu}} \frac{R}{1-\xi}. \tag{31}$$

(ii) Suppose (A1), (A2) and (A3) hold. Then for any* $\alpha \in (0,1/2)$, *with probability at least* $1-2\alpha$,

$$\|\beta^k - \bar{\beta}^*\|_\infty \leq c_\mu\varepsilon_2 \quad \text{if} \quad k \geq \log_{\frac{1}{\gamma_\mu}} \frac{4\bar{M}}{\varepsilon_2}, \tag{32}$$

where ε_2 *is given in (29).*

Furthermore, assume $\bar{m} \geq \frac{4\varepsilon_2}{\xi(1-\gamma_\mu)}$ *for some* $0 < \xi < 1$, *then*

$$A^k \supseteq A^*_J \quad \text{if} \quad k \geq \log_{\frac{1}{\gamma_\mu}} \frac{R}{1-\xi}. \tag{33}$$

(iii) Suppose β^* is exactly K-sparse. Let $T = K$ in SDAR. Suppose (A1), (A2) and (A3) hold and $m \geq \frac{4}{\xi(1-\gamma_\mu)}\sigma\sqrt{2\log(p/\alpha)/n}$ for some $0 < \xi < 1$. We have with probability at least $1-2\alpha$, $A^k = A^{k+1} = A^*$ if $k \geq \log_{\frac{1}{\gamma_\mu}} \frac{R}{1-\xi}$, i.e., with at most $O(\log R)$ iterations, SDAR stops and the output is the oracle least squares estimator β^o.*

Remark 15 *Theorem 4 and Corollary 8 can be derived from Theorem 12 and Corollary 14, respectively, by using the relationship between the ℓ_∞ norm and the ℓ_2 norm. Here we present them separately because (A2) is weaker than (A2*). The stronger assumption (A2*) brings us some new insights into the SDAR, i.e., the sharp ℓ_∞ error bound, based on which we can show that the worst case iteration complexity of SDAR does not depend on the underlying sparsity level, as stated in parts (ii) and (iii) of Corollary 14.*

Remark 16 *The mutual coherence condition $s\mu \leq 1$ with $s \geq 2K-1$ is used in the study of OMP and Lasso under the assumption that β^* is exactly K-sparse. In the noiseless case with $\eta = 0$, Tropp (2004); Donoho and Tsaig (2008) showed that under the condition $(2K-1)\mu < 1$, OMP can recover β^* exactly in K steps. In the noisy case with $\|\eta\|_2 \leq \varepsilon$, Donoho et al. (2006) proved that OMP can recover the true support if $(2K-1)\mu \leq 1-(2\varepsilon/m)$. Cai and Wang (2011) gave a sharp analysis of OMP under the condition $(2K-1)\mu < 1$. The mutual coherence condition $T\mu \leq 1/4$ in (A2*) is a little stronger than those used in the analysis of the OMP. However, under (A2*) we obtain a sharp ℓ_∞ error bound, which is not available for OMP in the literature. Furthermore, Corollary 14 implies that the number of iterations of SDAR does not depend on the sparsity level, which is a surprising result and does not appear in the literature on greedy methods, see Remark 18 below. Lounici (2008); Zhang (2009) derived an ℓ_∞ estimation error bound for the Lasso under the conditions $K\mu < 1/7$ and $K\mu \leq 1/4$, respectively. However, they needed a nontrivial Lasso solver for computing an approximate solution while SDAR only involves simple computational steps.*

Remark 17 *Suppose β^* is exactly K-sparse. Part (ii) of Corollary 14 implies that the sharp error bound*

$$\|\beta^k - \beta^*\|_\infty \leq c_\mu \sigma\sqrt{2\log(p/\alpha)/n} \tag{34}$$

is achieved with high probability if $k \geq \log_{\frac{1}{\gamma_\mu}} \frac{M}{\sigma\sqrt{2\log(p/\alpha)/n}}$.

Remark 18 *Suppose β^* is exactly K-sparse. Part (iii) of Corollary 14 implies that with high probability, the oracle estimator can be recovered in no more than $O(\log R)$ steps if we set $T = K$ in SDAR and the minimum magnitude of the nonzero elements of β^* is $O(\sigma\sqrt{2\log(p)/n})$, which is the optimal magnitude of detectable signals.*

Remark 19 *The number of iterations in Corollary 14 depends on the relative magnitude R, but not the sparsity level K, see Figure 2 for the numerical results supporting this. This improves the result in part (iii) of Corollary 8. This is a surprising result since as far as we know the number of iterations for the greedy methods to recover A^* depends on K, see for example, Garg and Khandekar (2009).*

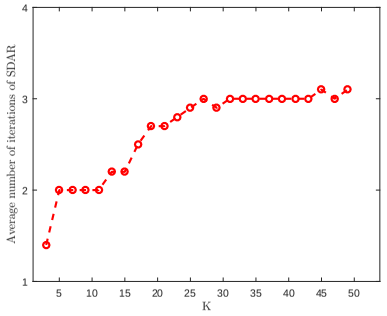


Figure 2: The average number of iterations of SDAR as K increases.

Figure 2 shows the average number of iterations of SDAR with $T = K$ based on 100 independent replications on data sets generated from a model with $(n = 500, p = 1000, K = 3:2:50, \sigma = 0.01, \rho = 0.1, R = 1)$, which will be described in Section 7.4. We can see that as the sparsity level increases from 3 to 50 the average number of iterations of SDAR remains stable, ranging from 1 to 3, which supports the assertion in Corollary 14. More numerical comparison on number of iterations with greedy methods are shown in Section 7.2.

3.3 A brief high-level description of the proofs

The detailed proofs of Theorems 4 and 12 and their corollaries are given in the Appendix. Here we describe the main ideas behind the proofs and point out the places where the SRC and the mutual coherence condition are used.

SDAR iteratively detects the support of the solution and then solves a least squares problem on the support. Therefore, to study the convergence properties of the sequence generated by SDAR, the key is to show that the sequence of active sets A^k can approximate A^*_J more and more accurately as k increases. Let

$$D(A^k) = \|\bar{\beta}^*|_{A^*_J \setminus A^k}\|, \tag{35}$$

where $\|\cdot\|$ can be either the ℓ_2 norm or the ℓ_∞ norm. This is a measure of the difference between A^*_J and A^k at the kth iteration in terms of the norm of the coefficients in A^*_J but not in A^k. A crucial step is to show that $D(A^k)$ decays geometrically to a value bounded by $h(T)$ up to a constant factor, where $h(T)$ is $h_2(T)$ defined in (12) or $h_\infty(T)$ in (24). Here $h(T)$ is a measure of the intrinsic error due to the noise η and the approximate error in (10). Specifically, much effort is spent on establishing the inequality (Lemma 27 in the Appendix)

$$D(A^{k+1}) \leq \gamma^* D(A^k) + c^* h(T), k = 0, 1, 2, \ldots, \tag{36}$$

where $\gamma^* = \gamma$ for the ℓ_2 results in Theorem 4 and $\gamma^* = \gamma_\mu$ for the ℓ_∞ results in Theorem 12, and $c^* > 0$ is a constant depending on the design matrix. The SRC (A2) and the

mutual coherence condition (A2*) play a critical role in establishing (36). Clearly, for this inequality to be useful, we need $0 < \gamma^* < 1$.

Another useful inequality is

$$\|\beta^{k+1} - \bar{\beta}^*\| \le c_1 D(A^k) + c_2 h(T), \tag{37}$$

where c_1 and c_2 are positive constants depending on the design matrix, see Lemma 23 in the Appendix. The SRC and the mutual coherence condition are needed to establish this inequality for the ℓ_2 norm and the ℓ_∞ norm, respectively. Then combining (36) and (37), we can show part (i) of Theorem 4 and part (i) of Theorem 12.

The inequalities (36) and (37) hold for any noise vector η. Under the sub-Gaussian assumption for η, $h(T)$ can be controlled by the sum of unrecoverable approximation error R_J and the universal noise level $O(\sigma\sqrt{2\log(p)/n})$ with high probability. This leads to the results in the remaining parts of Theorems 4 and 12, as well as Corollaries 8 and 14.

4. Adaptive SDAR

In practice, because the sparsity level of the model is usually unknown, we can use a data driven procedure to determine an upper bound, T, for the number of important variables, J, used in SDAR (Algorithm 1). The idea is to take T as a tuning parameter, so T plays a role similar to the penalty parameter λ in a penalized method. We can run SDAR from $T = 1$ to a large $T = L$. For example, we can take $L = O(n/\log(n))$ as suggested by Fan and Lv (2008), which is an upper bound of the largest possible model that can be consistently estimated with sample size n. By doing so we obtain a solution path $\{\hat{\beta}(T) : T = 0, 1, \ldots, L\}$, where $\hat{\beta}(0) = 0$, that is, $T = 0$ corresponds to the null model. Then we use a data driven criterion, such as HBIC (Wang et al., 2013), to select a $T = \hat{T}$ and use $\hat{\beta}(\hat{T})$ as the final estimate. The overall computational complexity of this process is $O(Lnp\log(R))$, see Section 5.

We can also compute the path by increasing T along a subsequence of the integers in $[1, L]$, for example, by taking a geometrically increasing subsequence. This will reduce the computational cost, but here we consider the worst-case scenario.

We note that tuning T is no more difficult than tuning a continuous penalty parameter λ in a penalized method. Indeed, we can simply increase T one by one from $T = 0$ to $T = L$ (or along a subsequence). In comparison, in tuning the value of λ based on a pathwise solution over an interval $[\lambda_{\min}, \lambda_{\max}]$, where $\lambda_{\max}$ corresponds to the null model and $\lambda_{\min} > 0$ is a small value, we need to determine the grid of λ values on $[\lambda_{\min}, \lambda_{\max}]$ as well as $\lambda_{\min}$. Here $\lambda_{\min}$ corresponds to the largest model on the solution path. In the numerical implementation of the coordinate descent algorithms for the Lasso (Friedman et al., 2007), MCP and SCAD (Breheny and Huang, 2011), $\lambda_{\min} = \alpha\lambda_{\max}$ for a small α, for example, $\alpha = 0.0001$. Determining the value of L is somewhat similar to determining $\lambda_{\min}$. However, L has the meaning of the model size, but the meaning of $\lambda_{\min}$ is less explicit.

We also have the option to stop the iteration early according to other criterions. For example, we can run SDAR by gradually increasing T until the change in the consecutive solutions is smaller than a given value. Candes et al. (2006) proposed to recover β^* based on (23) by finding the most sparse solution whose residual sum of squares is smaller than a

prespecified noise level ε. Inspired by this, we can also run SDAR by increasing T gradually until the residual sum of squares is smaller than a prespecified value ε.

We summarize these ideas in Algorithm 2 below.

Algorithm 2 Adaptive SDAR (ASDAR)

Require: Initial guess β^0, d^0, an integer τ, an integer L, and an early stopping criterion (optional). Set $k = 1$.

1: **for** $k = 1, 2, \cdots$ **do**
2: Run Algorithm 1 with $T = \tau k$ and with initial value (β^{k-1}, d^{k-1}). Denote the output by (β^k, d^k).
3: **if** the early stopping criterion is satisfied or $T > L$ **then**
4: stop
5: **else**
6: $k = k + 1$.
7: **end if**
8: **end for**

Ensure: $\hat{\beta}(\hat{T})$ as estimations of β^*.

5. Computational complexity

We look at the number of floating point operations line by line in Algorithm 1. Clearly it takes $O(p)$ flops to finish step 2-4. In step 5, we use conjugate gradient (CG) method (Golub and Van Loan, 2012) to solve the linear equation iteratively. During the CG iterations the main operation include two matrix-vector multiplications, which cost $2n|A_{k+1}|$ flops (the term $X'y$ on the right-hand side can be precomputed and stored). Therefore the number of CG iterations is smaller than $p/(2|A_{k+1}|)$, this ensures that the number of flops in step 5 is $O(np)$. In step 6, calculating the matrix-vector product costs np flops. In step 7, checking the stopping condition needs $O(p)$ flops. So the the overall cost per iteration of Algorithm 1 is $O(np)$. By Corollary 14 it needs no more than $O(\log(R))$ iterations to get a good solution for Algorithm 1 under the certain conditions. Therefore the overall cost of Algorithm 1 is $O(np\log(R))$ for exactly sparse and approximately sparse case under proper conditions.

Now we consider the cost of ASDAR (Algorithm 2). Assume ASDAR is stopped when $k = L$. Then the above discussion shows the the overall cost of Algorithm 2 is bounded by $O(Lnp\log(R))$ which is very efficient for large scale high dimension problem since the cost increases linearly in the ambient dimension p.

6. Comparison with greedy and screening methods

We give a high level comparison between SDAR and several greedy and screening methods, including OMP (Mallat and Zhang, 1993; Tropp, 2004; Donoho et al., 2006; Cai and Wang, 2011; Zhang, 2011b), FoBa (Zhang, 2011a), IHT (Blumensath and Davies, 2009; Jain et al., 2014) or GraDes (Garg and Khandekar, 2009), and SIS (Fan and Lv, 2008). These greedy methods iteratively select/remove one or more variables and project the response vector onto the linear subspace spanned by the variables that have already been selected. From

this point of view, they and SDAR share a similar characteristic. However, OMP and FoBa, select one variable per iteration based on the current correlation, i.e., the dual variable d^k in our notation, while SDAR selects T variables at a time based on the sum of primal (β^k) and dual (d^k) information. The following interpretation in a low-dimensional setting with a small noise term may clarify the differences between these two approaches. If $X'X/n \approx \mathbb{I}$ and $\eta \approx 0$, we have

$$d^k = X'(y - X\beta^k)/n = X'(X\beta^* + \eta - X\beta^k)/n \approx \beta^* - \beta^k + X'\eta/n \approx \beta^* - \beta^k,$$

and

$$\beta^k + d^k \approx \beta^*.$$

Hence, SDAR can approximate the underlying support A^* more accurately than OMP and Foba. This is supported by the simulation results given in Section 7.

IHT (Blumensath and Davies, 2009; Jain et al., 2014) or GraDes (Garg and Khandekar, 2009), can be formulated as

$$\beta^{k+1} = H_K(\beta^k + s_k d^k), \tag{38}$$

where $H_K(\cdot)$ is the hard thresholding operator by keeping the first K largest elements and setting others to 0. The step size s_k is chosen as $s_k = 1$ and $s_k = 1/(1 + \delta_{2K})$ (where δ_{2K} is the RIP constant) for IHT and GraDes, respectively. IHT and GraDes use both primal and dual information to detect the support of the solution, which is similar to SDAR. But when the approximate active set is given, SDAR uses least squares fitting, which is more accurate than just keeping the largest elements by hard thresholding. This is supported by the simulation results given in Section 7. Jain et al. (2014) proposed an iterative hard thresholding algorithm for general high-dimensional sparse regressions. In the linear regression setting, the algorithm proposed in Jain et al. (2014) is the same as GraDes. Jain et al. (2014) also considered a two-stage IHT, which involves a refit step on the detected support. Yuan et al. (2018) extended gradient hard thresholding for least squares loss to a general class of convex losses and analyzed the estimation and sparsity recovery performance of their proposed method. Under restricted strongly convexity (RSS) and restricted strongly smoothness conditions (RSC), Jain et al. (2014) derived an error estimate between the approximate solutions and the oracle solution in ℓ_2 norm, which has the same order as our result in Section 3.1. There are some differences between SDAR and the two-stage IHT proposed in Jain et al. (2014). First, SDAR solves an $n \times K$ least squares problem at each iteration while the two-stage IHT involves two least-squares problems with larger sizes. The regularity conditions on X for SDAR concerns $2K \times 2K$ submatrices of X, while the regularity conditions for the two-stage IHT involves larger submatries of X. Second, our results are applicable to approximately sparse models. Jain et al. (2014) only considered exact sparse case. Third, we showed in (iii) of Corollary 3.1 that the iteration complexity of SDAR is $\mathcal{O}(\log K)$. In comparison, the iteration complexity of the two-stage IHT is $\mathcal{O}(K)$. We also established an ℓ_∞ norm estimation result and showed that the number of iterations of SDAR is independent of the sparsity level, see (iii) of Corollary 3.2. Last, we showed that the stopping criterion for SDAR can be archived in finitely many steps (Corollary 3.1 (iii) and Corollary 3.2. (iii)). However, Jain et al. (2014) did not discuss this issue.

Fan and Lv (2008) proposed SIS for dimension reduction in ultrahigh dimensional liner regression problems. This method selects variables with the T largest absolute values of $X'y$. To improve the performance of SIS, Fan and Lv (2008) also considered an iterative SIS, which iteratively selects more than one feature at a time until a desired number of variables are selected. They reported that the iterative SIS outperforms SIS numerically. However, the iterative SIS lacks a theoretical analysis. Interestingly, the first step in SDAR initialized with $\mathbf{0}$ is exactly the same as the SIS. But again the process of SDAR is different from the iterative SIS in that the active set of SDAR is determined based on the sum of primal and dual approximations while the iterative SIS uses dual only.

7. Simulation Studies

7.1 Implementation

We implemented SDAR/ASDAR, FoBa, GraDes and MCP in MatLab. For FoBa, our MatLab implementation follows the R package developed by Zhang (2011a). We optimize it by keeping track of rank-one updates after each greedy step. Our implementation of MCP uses the iterative threshholding algorithm (She, 2009) with warm starts. Publicly available Matlab packages for LARS (included in the SparseLab package) are used. Since LARS and FoBa add one variable at a time, we stop them when K variables are selected in addition to their default stopping conditions. Of course, doing so will reduce the computation time for these algorithms as well as improve accuracy by preventing overfitting.

In GraDes, the optimal gradient step length s_k depends on the RIP constant of X, which is NP hard to compute (Tillmann and Pfetsch, 2014). Here, we set $s_k = 1/3$ following Garg and Khandekar (2009). We stop GraDes when the residual norm is smaller than $\varepsilon = \sqrt{n}\sigma$, or the maximum number of iterations is greater than $n/2$. We compute the MCP solution path and select an optimal solution using the HBIC (Wang et al., 2013). We stop the iteration when the residual norm is smaller than $\varepsilon = \|\eta\|_2$, or the estimated support size is greater than $L = n/\log(n)$. In ASDAR (Algorithm 2), we set $\tau = 50$ and we stop the iteration if the residual $\|y - X\beta^k\|$ is smaller than $\varepsilon = \sqrt{n}\sigma$ or $k \geq L = n/\log(n)$.

7.2 Accuracy and efficiency

We compare the accuracy and efficiency of SDAR/ASDAR with Lasso (LARS), MCP, GraDes and FoBa.

We consider a moderately large scale setting with $n = 5000$ and $p = 50000$. The number of nonzero coefficients is set to be $K = 400$. So the sample size n is about $O(K\log(p-K))$. The dimension of the model is nearly at the limit where β^* can be reasonably well estimated by the Lasso (Wainwright, 2009).

To generate the design matrix X, we first generate an $n \times p$ random Gaussian matrix $\bar{X}$ whose entries are i.i.d. $\mathcal{N}(0,1)$ and then normalize its columns to the $\sqrt{n}$ length. Then X is generated with $X_1 = \bar{X}_1$, $X_j = \bar{X}_j + \rho(\bar{X}_{j+1} + \bar{X}_{j-1})$, $j = 2, \ldots, p-1$ and $X_p = \bar{X}_p$. The underlying regression coefficient β^* is generated with the nonzero coefficients uniformly distributed in $[m, M]$, where $m = \sigma\sqrt{2\log(p)/n}$ and $M = 100m$. Then the observation vector $y = X\beta^* + \eta$ with $\eta_1, \ldots, \eta_n$ generated independently from $\mathcal{N}(0, \sigma^2)$. We set $R = 100, \sigma = 1$ and $\rho = 0.2, 0.4$ and 0.6.

Table 1 shows the results based on 100 independent replications. The first column gives the correlation value ρ and the second column shows the methods in the comparison. The third and the fourth columns give the averaged relative error, defined as ReErr $= \sum \|\hat{\beta} - \beta^*\|/\|\beta^*\|$, and the averaged CPU time (in seconds), The standard deviations of the CPU times and the relative errors are shown in the parentheses. In each column of Table 1, the numbers in boldface indicate the best performers.

Table 1: Numerical results (relative errors, CPU times) on data sets with $n = 5000, p = 50000, K = 400, R = 100, \sigma = 1, \rho = 0.2 : 0.2 : 0.6$.

ρ	Method	ReErr	time(s)
0.2	LARS	1.1e-1 (2.5e-2)	4.8e+1 (9.8e-1)
	MCP	**7.5e-4** (**3.6e-5**)	9.3e+2 (2.4e+3)
	GraDes	1.1e-3 (7.0e-5)	2.3e+1 (9.0e-1)
	FoBa	**7.5e-4** (7.0e-5)	4.9e+1 (3.9e-1)
	ASDAR	**7.5e-4** (4.0e-5)	8.4e+0 (4.5e-1)
	SDAR	**7.5e-4** (4.0e-5)	**1.4e+0** (**5.1e-2**)
0.4	LARS	1.8e-1 (1.2e-2)	4.8e+1 (1.8e-1)
	MCP	6.2e-4 (3.6e-5)	2.2e+2 (1.6e+1)
	GraDes	8.8e-4 (5.7e-5)	8.7e+2 (2.6e+3)
	FoBa	1.0e-2 (1.4e-2)	5.0e+1 (4.2e-1)
	ASDAR	**6.0e-4** (**2.6e-5**)	8.8e+0 (**3.2e-1**)
	SDAR	**6.0e-4** (**2.6e-5**)	**2.3e+0** (1.7e+0)
0.6	LARS	3.0e-1 (2.5e-2)	4.8e+1 (3.5e-1)
	MCP	4.5e-4 (**2.5e-5**)	4.6e+2 (5.1e+2)
	GraDes	7.8e-4 (1.1e-4)	1.5e+2 (2.3e+2)
	FoBa	8.3e-3 (1.3e-2)	5.1e+1 (1.1e+0)
	ASDAR	**4.3e-4** (3.0e-5)	1.1e+1 (5.1e-1)
	SDAR	**4.3e-4** (3.0e-5)	**2.1e+0** (**8.6e-2**)

We see that when the correlation ρ is low, i.e., $\rho = 0.2$, MCP, FoBa, SDAR and ASDAR are on the top of the list in average error (ReErr). In terms of speed, SDAR/ASDAR is about 3 to 100 times faster than the other methods. As the correlation ρ increases to $\rho = 0.4$ and $\rho = 0.6$, FoBa becomes less accurate than SDAR/ASDAR. MCP is similar to SDAR/ASDAR in terms of accuracy, but it is 20 to 100 times slower than SDAR/ASDAR. The standard deviations of the CPU times and the relative errors of MCP and SDAR/ASDAR are similar and smaller than those of the other methods in all the three settings.

7.3 Influence of the model parameters

We now consider the effects of each of the model parameters on the performance of ASDAR, LARS, MCP, GraDes and FoBa more closely.

In this set of simulations, the rows of the design matrix X are drawn independently from $\mathcal{N}(0, \Sigma)$ with $\Sigma_{jk} = \rho^{|j-k|}, 1 \le j, k \le p$. The elements of the error vector η are generated independently with $\eta_i \sim \mathcal{N}(0, \sigma^2)$, $i = 1, \ldots, n$. Let $R = M/m$, where, $M = \max\{|\beta^*_{A^*}|\}, m = \min\{|\beta^*_{A^*}|\} = 1$. The underlying regression coefficient vector $\beta^* \in \mathbb{R}^p$ is generated in such a way that A^* is a randomly chosen subset of $\{1, 2, ..., p\}$ with $|A^*| = K < n$ and $R \in [1, 10^3]$. Then the observation vector $y = X\beta^* + \eta$. We use $\{n, p, K, \sigma, \rho, R\}$ to indicate the parameters used in the data generating model described above. We run ASDAR with $\tau = 5, L = n/\log(n)$ (if not specified). We use the HBIC (Wang et al., 2013) to select the tuning parameter T. The simulation results given in Figure 3 are based on 100 independent replications.

7.3.1 Influence of the sparsity level K

The top left panel of Figure 3 shows the results of the influence of sparsity level K on the probability of exact recovery of A^* of ASDAR, LARS, MCP, GraDes and FoBa. Data are generated from the model with $(n = 500, p = 1000, K = 10 : 50 : 360, \sigma = 0.5, \rho = 0.1, R = 10^3)$. Here $K = 10 : 50 : 360$ means the sample size starts from 10 to 360 with an increment of 50. We use $L = 0.8n$ for both ASDAR and MCP to eliminate the effect of stopping rule since the maximum $K = 360$. When the sparsity level $K = 10$, all the solvers performed well in recovering the true support. As K increases, LARS was the first one that failed to recover the support and vanished when $K = 60$ (this phenomenon had also been observed in Garg and Khandekar (2009), MCP began to fail when $K > 110$, GraDes and FoBa began to fail when $K > 160$. In comparison, ASDAR was still able to do well even when $K = 260$.

7.3.2 Influence of the sample size n

The top right panel of Figure 3 shows the influence of the sample size n on the probability of correctly estimating A^*. Data are generated from the model with $(n = 30 : 20 : 200, p = 500, K = 10, \sigma = 0.1, \rho = 0.1, R = 10)$. We see that the performance of all the five methods becomes better as n increases. However, ASDAR performs better than the others when $n = 30$ and 50. These simulation results indicate that ASDAR is more capable of handling high-dimensional data when p/n is large in the generating models considered here

7.3.3 Influence of the ambient dimension p

The bottom left panel of Figure 3 shows the influence of ambient dimension p on the performance of ASDAR, LARS, MCP, GraDes and FoBa. Data are generated from the model with $(n = 100, p = 200 : 200 : 1000, K = 20, \sigma = 1, \rho = 0.3, R = 10)$. We see that the probabilities of exactly recovering the support of the underlying coefficients of ASDAR and MCP are higher than those of the other solvers as p increasing, which indicate that ASDAR and MCP are more robust to the ambient dimension.

7.3.4 Influence of correlation ρ

The bottom right panel of Figure 3 shows the influence of correlation ρ on the performance of ASDAR, LARS, MCP, GraDes and FoBa. Data are generated from the model with $(n = 150, p = 500, K = 25, \sigma = 0.1, \rho = 0.05 : 0.1 : 0.95, R = 10^2)$. The performance of

all the solvers becomes worse when the correlation ρ increases. However, ASDAR generally performed better than the other methods as ρ increases.

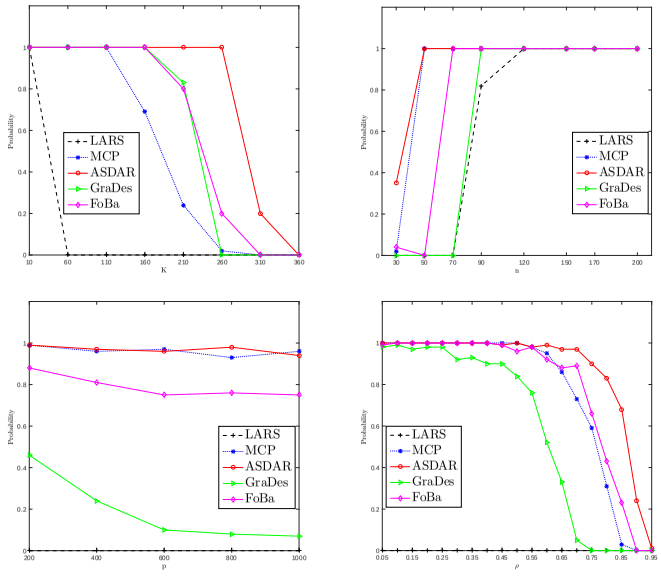


Figure 3: Numerical results of the influence of sparsity level K (top left panel), sample size n (top right panel), ambient dimension p (bottom left panel) and correlation ρ (bottom right panel) on the probability of exact recovery of the true support of all the solvers considered here.

In summary, our simulation studies demonstrate that SDAR/ASDAR is generally more accurate, more efficient and more stable than Lasso, MCP, FoBa and GraDes.

7.4 Number of iterations

In this subsection we compare SDAR with GraDes (IHT) in terms of the number of iterations. We run 100 independent replications on data sets generated from the models with $(n = 500, p = 1000, K = 5 : 5 : 55, \sigma = 0.05, \rho = 0, R = 1)$ and $(n = 2000, p = 5000, K = 10 : 20 : 250, \sigma = 0.05, \rho = 0, R = 1)$ described in Section 7.3. The average number of iteration (left column) and average absolute error in ℓ_∞ norm (right column) are displayed in Figure 4. We can see that the number of iterations of GraDes increases almost sublinearly as the sparsity level K increases while that of SDAR almost varies little. And in terms of the average error, SDAR is serval times more accurate than GraDes. This provides empirical support for our theoretical results in Corollary 30.

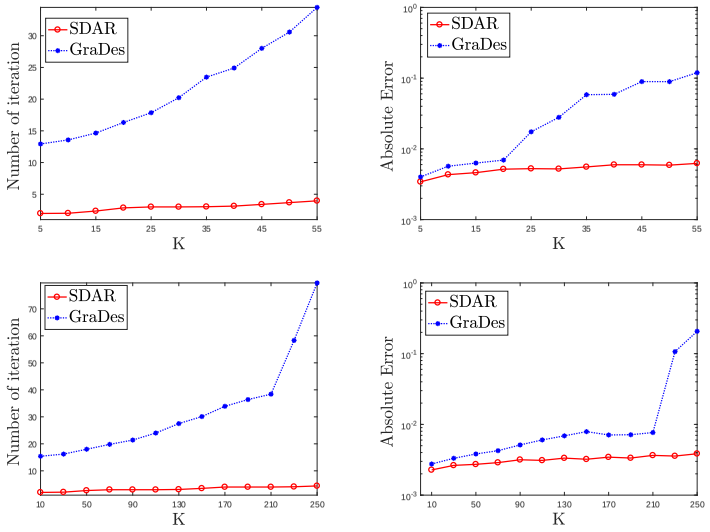


Figure 4: Comparisons the dependence of number of iterations (left panels) and accuracy (right panels) on sparsity level K with data set $(n = 500, p = 1000, K = 5 : 5 : 55, \sigma = 0.05, \rho = 0.3, R = 1)$ and $(n = 2000, p = 5000, K = 10 : 20 : 250, \sigma = 0.05, \rho = 0.3, R = 1)$.

8. Concluding remarks

SDAR is a constructive approach for fitting sparse, high-dimensional linear regression models. Under appropriate conditions, we established the nonasymptotic minimax ℓ_2 error bound and optimal ℓ_∞ error bound of the solution sequence generated by SDAR. We also calculated the number of iterations required to achieve these bounds. In particular, an interesting and surprising aspect of our results is that, under a mutual coherence condition on the design matrix, the number of iterations required for the SDAR to achieve the optimal ℓ_∞ bound does not depend on the underlying sparsity level. In addition, SDAR has the same computational complexity per iteration as LARS, coordinate descent and greedy methods. Our simulation studies demonstrate that SDAR/ASDAR is accurate, fast, stable and easy to implement, and it is competitive with or outperforms the Lasso, MCP and two greedy methods in efficiency and accuracy in the generating models we considered. These theoretical and numerical results suggest that SDAR/ASDAR is a useful addition to the literature on sparse modeling.

We have only considered the linear regression model. It would be interesting to generalize SDAR to models with more general loss functions or with other types of sparsity structures.

It would also be interesting to develop parallel or distributed versions of SDAR that can run on multiple cores for data sets with big n and large p or for data that are distributively stored.

We have implemented SDAR in a Matlab package *sdar*, which is available at `http://homepage.stat.uiowa.edu/~jian/`.

Acknowledgments

We are grateful to the action editor and the anonymous reviewers for their detailed and constructive comments which led to considerable improvements in the paper. We also thank Patrick Breheny for his critical reading of the paper and providing helpful comments. This research is supported in part by the National Science Foundation of China (NSFC) grants 11501579, 11571263, 11471253, 91630313, 11871385, 11871474 and 11801531.

Appendix A

Proof of Lemma 1.

Proof Let $L_\lambda(\beta) = \frac{1}{2n}\|X\beta - y\|_2^2 + \lambda\|\beta\|_0$. Suppose $\beta^\diamond$ is a coordinate-wise minimizer of L_λ. Then

$$\begin{aligned}
& \beta_i^\diamond \in \operatorname*{argmin}_{t\in\mathbb{R}} L_\lambda(\beta_1^\diamond, ..., \beta_{i-1}^\diamond, t, \beta_{i+1}^\diamond, ..., \beta_p^\diamond) \\
\Rightarrow \quad & \beta_i^\diamond \in \operatorname*{argmin}_{t\in\mathbb{R}} \tfrac{1}{2n}\|X\beta^\diamond - y + (t-\beta_i^\diamond)X_i\|_2^2 + \lambda|t|_0 \\
\Rightarrow \quad & \beta_i^\diamond \in \operatorname*{argmin}_{t\in\mathbb{R}} \tfrac{1}{2}(t-\beta_i^\diamond)^2 + \tfrac{1}{n}(t-\beta_i^\diamond)X_i'(X\beta^\diamond - y) + \lambda|t|_0 \\
\Rightarrow \quad & \beta_i^\diamond \in \operatorname*{argmin}_{t\in\mathbb{R}} \tfrac{1}{2}(t-(\beta_i^\diamond + X_i'(y-X\beta^\diamond)/n))^2 + \lambda|t|_0.
\end{aligned}$$

Let $d^\diamond = X'(y - X\beta^\diamond)/n$. By the definition of the hard thresholding operator $H_\lambda(\cdot)$ in (5), we have

$$\beta_i^\diamond = H_\lambda(\beta_i^\diamond + d_i^\diamond) \quad \text{for } i = 1, ..., p,$$

which shows (4) holds.

Conversely, suppose (4) holds. Let

$$A^\diamond = \Big\{ i \in S \big| \, |\beta_i^\diamond + d_i^\diamond| \ge \sqrt{2\lambda} \Big\}.$$

By (4) and the definition of $H_\lambda(\cdot)$ in (5), we deduce that for $i \in A^\diamond$, $|\beta_i^\diamond| \ge \sqrt{2\lambda}$. Furthermore, $\mathbf{0} = d_{A^\diamond}^\diamond = X_{A^\diamond}'(y - X_{A^\diamond}\beta_{A^\diamond}^\diamond)/n$, which is equivalent to

$$\beta_{A^\diamond}^\diamond \in \operatorname{argmin} \tfrac{1}{2n}\|X_{A^\diamond}\beta_{A^\diamond} - y\|_2^2. \tag{39}$$

Next we show $L_\lambda(\beta^\diamond + h) \ge L_\lambda(\beta^\diamond)$ if h is small enough with $\|h\|_\infty < \sqrt{2\lambda}$. We consider two cases. If $h_{(A^\diamond)^c} \ne 0$, then

$$L_\lambda(\beta^\diamond + h) - L_\lambda(\beta^\diamond) \ge \tfrac{1}{2n}\|X\beta^\diamond - y + Xh\|_2^2 - \tfrac{1}{2n}\|X\beta^\diamond - y\|_2^2 + \lambda \ge \lambda - |\langle h, d^\diamond\rangle|,$$

which is positive for sufficiently small h. If $h_{(A^\diamond)^c} = 0$, by the minimizing property of $\beta_{A^\diamond}^\diamond$ in (39) we deduce that $L_\lambda(\beta^\diamond + h) \ge L_\lambda(\beta^\diamond)$. This completes the proof of Lemma 1. ■

Lemma 20 *Let A and B be disjoint subsets of S, with $|A| = a$ and $|B| = b$. Assume $X \sim SRC(a+b, c_-(a+b), c_+(a+b))$. Let $\theta_{a,b}$ be the sparse orthogonality constant and let μ be the mutual coherence of X. Then we have*

$$nc_-(a) \le \left\|X_A^T X_A\right\| \le nc_+(a), \tag{40}$$

$$\frac{1}{nc_+(a)} \le \left\|(X_A^T X_A)^{-1}\right\| \le \frac{1}{nc_-(a)}, \tag{41}$$

$$\left\|X_A'\right\| \le \sqrt{nc_+(a)} \tag{42}$$

$$\theta_{a,b} \le (c_+(a+b) - 1) \vee (1 - c_-(a+b)) \tag{43}$$

$$\|X_B' X_A u\|_\infty \le na\mu\|u\|_\infty, \quad \forall u \in \mathbb{R}^{|A|}, \tag{44}$$

$$\|X_A\| = \left\|X_A'\right\| \le \sqrt{n(1 + (a-1)\mu)}. \tag{45}$$

Furthermore, if $\mu < 1/(a-1)$, then

$$\|(X_A' X_A)^{-1} u\|_\infty \le \frac{\|u\|_\infty}{n(1-(a-1)\mu)}, \quad \forall u \in \mathbb{R}^{|A|}. \tag{46}$$

Moreover, $c_+(s)$ is an increasing function of s, $c_-(s)$ a decreasing function of s and $\theta_{a,b}$ an increasing function of a and b.

Proof The assumption $X \sim \mathrm{SRC}(a, c_-(a), c_+(a))$ implies the spectrum of $X_A'X_A/n$ is contained in $[c_-(a), c_+(a)]$. So (40) - (42) hold. Let $\mathbb{I}$ be an $(a+b)\times(a+b)$ identity matrix. (43) follows from the fact that $X_A'X_B/n$ is a submatrix of $X_{A\cup B}'X_{A\cup B}/n - \mathbb{I}$ whose spectrum norm is less than $(1 - c_-(a+b)) \vee (c_+(a+b) - 1)$. Let $G = X'X/n$. Then, $|\sum_{j=1}^a G_{i,j}u_j| \le \mu a\|u\|_\infty$, for all $i \in B$, which implies (44). By Gerschgorin's disk theorem,

$$|\,\|G_{A,A}\| - G_{i,i}| \le \sum_{i\ne j=1}^{a} |G_{i,j}| \le (a-1)\mu \quad \forall i \in A,$$

thus (45) holds. For (46), it suffices to show $\|G_{A,A}u\|_\infty \ge (1-(a-1)\mu)\|u\|_\infty$ if $\mu < 1/(a-1)$. In fact, let $i \in A$ such that $\|u\|_\infty = |u_i|$, then

$$\|G_{A,A}u\|_\infty \ge |\sum_{j=1}^{a} G_{i,j}u_j| \ge |u_i| - \sum_{i\ne j=1}^{a} |G_{i,j}|\,|u_j| \ge \|u\|_\infty - \mu(a-1)\|u\|_\infty.$$

The last assertion follows from their definitions. This completes the proof of Lemma 20. ■

Lemma 21 *Suppose (A3) holds. We have for any $\alpha \in (0, 1/2)$,*

$$\mathbf{P}\Big(\|X'\eta\|_\infty \le \sigma\sqrt{2\log(p/\alpha)n}\Big) \ge 1 - 2\alpha, \tag{47}$$

$$\mathbf{P}\Big(\max_{|A|\le T} \|X_A'\eta\|_2 \le \sigma\sqrt{T}\sqrt{2\log(p/\alpha)n}\Big) \ge 1 - 2\alpha. \tag{48}$$

Proof This lemma follows from the sub-Gaussian assumption (A3) and standard probability calculation, see Zhang and Huang (2008); Wainwright (2009) for details. ■

We now define some notation that will be useful in proving Theorems 4 and 12. For any given integers T and J with $T \geq J$ and $F \subseteq S$ with $|F| = T - J$, let $A^\circ = A^*_J \cup F$ and $I^\circ = (A^\circ)^c$. Let $\{A^k\}_k$ be the sequence of active sets generated by SDAR (Algorithm 1).

Define

$$D_2(A^k) = \|\bar{\beta}^*|_{A^*_J \setminus A^k}\|_2 \quad \text{and} \quad D_\infty(A^k) = \|\bar{\beta}^*|_{A^* \setminus A^k}\|_\infty.$$

These quantities measure the differences between A_k and A^*_J in terms of the ℓ_2 and ℓ_∞ norms of the coefficients in A^*_J but not in A^k. A crucial step in our proofs is to control the sizes of these measures.

Let

$$A^k_1 = A^k \cap A^\circ, A^k_2 = A^\circ \setminus A^k_1, I^k_3 = A^k \cap I^\circ, I^k_4 = I^\circ \setminus I^k_3.$$

Denote the cardinality of I^k_3 by $l_k = |I^k_3|$. Let

$$A^k_{11} = A^k_1 \setminus (A^{k+1} \cap A^k_1), A^k_{22} = A^k_2 \setminus (A^{k+1} \cap A^k_2), I^k_{33} = A^{k+1} \cap I^k_3, I^k_{44} = A^{k+1} \cap I^k_4,$$

and

$$\triangle^k = \beta^{k+1} - \bar{\beta}^*|_{A^k}.$$

These notation can be easily understood in the case $T = J$. For example, $D_2(A^k)$ and $D_\infty(A^k)$ are measures of the difference between the active set A^k and the target support A^*_J. A^k_1 and I^k_3 contain the correct indices and incorrect indices in A^k, respectively. A^k_{11} and A^k_{22} include the indices in A° that will be lost from the kth iteration to the $(k+1)$th iteration. I^k_{33} and I^k_{44} contain the indices included in I° that will be gained from the kth iteration to the $(k+1)$th iteration. By Algorithm 1, we have $|A^k| = |A^{k+1}| = T$, $A^k = A^k_1 \cup I^k_3$, $|A^k_2| = |A^\circ| - |A^k_1| = |A^\circ| - |I^k_3| = T - (T - l_k) = l_k \leq T$, and

$$|A^k_{11}| + |A^k_{22}| = |I^k_{33}| + |I^k_{44}|, \tag{49}$$

$$D_2(A^k) = \|\bar{\beta}^*|_{A^\circ \setminus A^k}\|_2 = \|\bar{\beta}^*|_{A^k_2}\|_2, \tag{50}$$

$$D_\infty(A^k) = \|\bar{\beta}^*|_{A^\circ \setminus A^k}\|_\infty = \|\bar{\beta}^*|_{A^k_2}\|_\infty. \tag{51}$$

In Subsection 3.3, we described the overall approach for proving Theorems 4 and 12. Before proceeding to the proofs, we break down the argument into the following steps.

1. In Lemma 22 we show that the effects of the noise and the approximation model (10) measured by $h_2(T)$ and $h_\infty(T)$ can be controlled by the sum of unrecoverable approximation error R_J and the universal noise level $O(\sigma\sqrt{2\log(p)/n})$ with high probability, provided that η is sub-Gaussian.

2. In Lemma 23 we show that the ℓ_2 norms (ℓ_∞ norms) of $\triangle^k$ and $\beta^k - \bar{\beta}^*$ are controlled in terms of $D_2(A^k)$ and $h_2(T)$ ($D_\infty(A^k)$ and $h_\infty(T)$).

3. In Lemma 24 we show that $D_2(A^{k+1})$ ($D_\infty(A^{k+1})$) can be bounded by the norm of $\bar{\beta}^*$ on the lost indices, which in turn can be controlled in terms of $D_2(A^k)$ and $h_2(T)$ ($D_\infty(A^k)$ and $h_\infty(T)$) and the norms of $\triangle^k$, β^{k+1} and d^{k+1} on the lost indices.

4. In Lemma 25 we make use of the orthogonality between β^k and d^k to show that the norms of β^{k+1} and d^{k+1} on the lost indices can be bounded by the norm on the gained indices. Lemma 26 gives the upper bound of the norms of β^{k+1} and d^{k+1} on the gained indices by the sum of $D_2(A^k)$, $h_2(T)$ ($D_\infty(A^k)$, $h_\infty(T)$), and the norm of $\triangle^k$.

5. We combine Lemmas 22-26 and get the desired relations between $D_2(A^{k+1})$ and $D_2(A^k)$ ($D_\infty(A^{k+1})$ and $D_\infty(A^k)$) in Lemma 27.

Then we prove Theorems 4 and 12 based on Lemma 27, (56) and (58).

Lemma 22 *Let $A \subset S$ with $|A| \leq T$. Suppose (A1) and (A3) holds. Then for $\alpha \in (0, 1/2)$ with probability at least $1 - 2\alpha$, we have*

(i) If $X \sim SRC(T, c_-(T), c_+(T))$, then

$$h_2(T) \leq \varepsilon_1, \tag{52}$$

where ε_1 is defined in (18).

(ii) We have

$$h_\infty(T) \leq \varepsilon_2, \tag{53}$$

where ε_2 is defined in (29).

Proof We first show

$$\|X\beta^*|_{(A^*_J)^c}\|_2 \leq \sqrt{nc_+(J)}R_J, \tag{54}$$

under the assumption of $X \sim SRC(c_-(T), c_+(T), T)$ and (A1). In fact, let β be an arbitrary vector in $\mathbb{R}^p$ and A_1 be the first J largest positions of β, A_2 be the next and so forth. Then

$$\begin{aligned}
\|X\beta\|_2 &\leq \|X\beta_{A_1}\|_2 + \sum_{i\geq 2}\|X\beta_{A_i}\|_2 \\
&\leq \sqrt{nc_+(J)}\|\beta_{A_1}\|_2 + \sqrt{nc_+(J)}\sum_{i\geq 2}\|\beta_{A_i}\|_2 \\
&\leq \sqrt{nc_+(J)}\|\beta\|_2 + \sqrt{nc_+(J)}\sum_{i\geq 1}\sqrt{\frac{1}{J}}\|\beta_{A_{i-1}}\|_1 \\
&\leq \sqrt{nc_+(J)}(\|\beta\|_2 + \sqrt{\frac{1}{J}}\|\beta\|_1),
\end{aligned}$$

where the first inequality follows from the triangle inequality, the second inequality follows from (42), and the third and fourth ones follows from simple algebra. This implies (54) holds by observing the definition of R_J. By the triangle inequality, (42), (54) and (48), we have with probability at least $1 - 2\alpha$,

$$\begin{aligned}
\|X'_A\bar{\eta}\|_2/n &\leq \|X'_A X\beta^*|_{(A^*_J)^c}\|_2/n + \|X'_A\eta\|_2/n \\
&\leq c_+(J)R_J + \sigma\sqrt{T}\sqrt{2\log(p/\alpha)/n}.
\end{aligned}$$

Therefore, (52) follows by noticing the monotone increasing property of $c_+(\cdot)$, the definition of ε_1 in (18) and the arbitrariness of A.

By a similar argument for (54) and replacing $\sqrt{nc_+(J)}$ with $\sqrt{n(1+(J-1)\mu)}$ based on (45), we get

$$\|X\beta^*|_{(A^*_J)^c}\|_2 \le \sqrt{n(1+(K-1)\mu)}R_J. \tag{55}$$

Therefore, by (45), (55) and (47), we have with probability at least $1-2\alpha$,

$$\begin{aligned}\|X'_A\bar{\eta}\|_\infty/n &\le \|X'_AX\beta^*|_{(A^*_J)^c}\|_\infty/n + \|X'_A\eta\|_2/n\\ &\le \|X'_AX\beta^*|_{(A^*_J)^c}\|_2/n + \|X'_A\eta\|_2/n\\ &\le (1+(J-1)\mu)R_J + \sigma\sqrt{2\log(p/\alpha)/n}.\end{aligned}$$

This implies part (ii) of Lemma 22 by noticing the definition of ε_2 in (29) and the arbitrariness of A. This completes the proof of Lemma 22. ■

Lemma 23 *Let $A \subset S$ with $|A| \le T$. Suppose (A1) holds.*

(i) If $X \sim SRC(T, c_-(T), c_+(T))$,

$$\|\beta^{k+1} - \bar{\beta}^*\|_2 \le \Big(1 + \frac{\theta_{T,T}}{c_-(T)}\Big)D_2(A^k) + \frac{h_2(T)}{c_-(T)}, \tag{56}$$

and

$$\|\triangle^k\|_2 \le \frac{\theta_{T,T}}{c_-(T)}\|\bar{\beta}^*|_{A^k_2}\|_2 + \frac{h_2(T)}{c_-(T)}. \tag{57}$$

(ii) If $(T-1)\mu < 1$, then

$$\|\beta^{k+1} - \bar{\beta}^*\|_\infty \le \frac{1+\mu}{1-(T-1)\mu}D_\infty(A^k) + \frac{h_\infty(T)}{1-(T-1)\mu}, \tag{58}$$

and

$$\|\triangle^k\|_\infty \le \frac{T\mu}{1-(T-1)\mu}\|\bar{\beta}^*|_{A^k_2}\|_\infty + \frac{h_\infty(T)}{(1-(T-1)\mu)}. \tag{59}$$

Proof We have

$$\begin{aligned}\beta^{k+1}_{A^k} &= (X'_{A^k}X_{A^k})^{-1}X'_{A^k}y\\ &= (X'_{A^k}X_{A^k})^{-1}X'_{A^k}(X_{A^k_1}\bar{\beta}^*_{A^k_1} + X_{A^k_2}\bar{\beta}^*_{A^k_2} + \bar{\eta}),\end{aligned} \tag{60}$$

$$\begin{aligned}(\bar{\beta}^*|_{A^k})_{A^k} &= (X'_{A^k}X_{A^k})^{-1}X'_{A^k}X_{A^k}(\bar{\beta}^*|_{A^k})_{A^k}\\ &= (X'_{A^k}X_{A^k})^{-1}X'_{A^k}(X_{A^k_1}\bar{\beta}^*_{A^k_1}),\end{aligned} \tag{61}$$

where the first equality uses the definition of β^{k+1} in Algorithm 1, the second equality follows from $y = X\bar{\beta}^* + \bar{\eta} = X_{A^k_1}\bar{\beta}^*_{A^k_1} + X_{A^k_2}\bar{\beta}^*_{A^k_2} + \bar{\eta}$, the third equality is simple algebra, and the last one uses the definition of A^k_1. Therefore,

$$\begin{aligned}\|\triangle^k\|_2 &= \|\beta^{k+1}_{A^k} - (\bar{\beta}^*|_{A^k})_{A^k}\|_2\\ &= \|(X'_{A^k}X_{A^k})^{-1}X'_{A^k}(X_{A^k_2}\bar{\beta}^*_{A^k_2} + \bar{\eta})\|_2\\ &\le \frac{1}{nc_-(T)}(\|X'_{A^k}X_{A^k_2}\bar{\beta}^*_{A^k_2}\|_2 + \|X'_{A^k}\bar{\eta}\|_2)\\ &\le \frac{\theta_{T,T}}{c_-(T)}\|\bar{\beta}^*|_{A^k_2}\|_2 + \frac{h_2(T)}{c_-(T)},\end{aligned}$$

where the first equality uses $\mathrm{supp}(\beta^{k+1}) = A^k$, the second equality follows from (61) and (60), the first inequality follows from (41) and the triangle inequality, and the second inequality follows from (50), the definition of $\theta_{a,b}$ and the definition of $h_2(T)$. This proves (57). Then the triangle inequality $\|\beta^{k+1} - \bar{\beta}^*\|_2 \le \|\beta^{k+1} - \bar{\beta}^*|_{A^k}\|_2 + \|\bar{\beta}^*|_{A^\circ\setminus A^k}\|_2$ and (57) imply (56).

Using an argument similar to the proof of (57) and by (46), (44) and (51), we can show (59). Thus (58) follows from the triangle inequality and (59). This completes the proof of Lemma 23. ■

Lemma 24

$$D_2(A^{k+1}) \le \|\bar{\beta}^*_{A^k_{11}}\|_2 + \|\bar{\beta}^*_{A^k_{22}}\|_2, \tag{62}$$

$$D_\infty(A^{k+1}) \le \|\bar{\beta}^*_{A^k_{11}}\|_\infty + \|\bar{\beta}^*_{A^k_{22}}\|_\infty. \tag{63}$$

$$\|\bar{\beta}^*_{A^k_{11}}\|_2 \le \|\triangle^k_{A^k_{11}}\|_2 + \|\beta^{k+1}_{A^k_{11}}\|_2, \tag{64}$$

$$\|\bar{\beta}^*_{A^k_{11}}\|_\infty \le \|\triangle^k_{A^k_{11}}\|_\infty + \|\beta^{k+1}_{A^k_{11}}\|_\infty. \tag{65}$$

Furthermore, assume (A1) holds. We have

$$\|\bar{\beta}^*_{A^k_{22}}\|_\infty \le \|d^{k+1}_{A^k_{22}}\|_\infty + T\mu\|\triangle^k_{A^k}\|_\infty + T\mu D_\infty(A^k) + h_\infty(T), \tag{66}$$

$$\|\bar{\beta}^*_{A^k_{22}}\|_2 \le \frac{\|d^{k+1}_{A^k_{22}}\|_2 + \theta_{T,T}\|\triangle^k_{A^k}\|_2 + \theta_{T,T}D_2(A^k) + h_2(T)}{c_-(T)} \text{ if } X \sim SRC(T, c_-(T), c_+(T)). \tag{67}$$

Proof By the definitions of $D_2(A^{k+1})$, A^k_{11}, A^k_{11} and A^k_{22}, we have

$$D_2(A^{k+1}) = \|\bar{\beta}^*|_{A^\circ\setminus A^{k+1}}\|_2 = \|\bar{\beta}^*|_{A^k_{11}\cup A^k_{22}}\|_2 \le \|\bar{\beta}^*_{A^k_{11}}\|_2 + \|\bar{\beta}^*_{A^k_{22}}\|_2.$$

This proves (62). (63) can be proved similarly. To show (64), we note that $\triangle^k = \beta^{k+1} - \bar{\beta}^*|_{A^k}$. Thus

$$\|\beta^{k+1}_{A^k_{11}}\|_2 = \|\Big(\bar{\beta}^*\big|_{A^k}\Big)_{A^k_{11}} + \triangle^k_{A^k_{11}}\|_2 \ge \|\bar{\beta}^*_{A^k_{11}}\|_2 - \|\triangle^k_{A^k_{11}}\|_2.$$

This proves (64). (65) can be proved similarly.

Now consider (67). We have

$$\begin{aligned}
\|d^{k+1}_{A^k_{22}}\|_2 &= \|X'_{A^k_{22}}\Big(X_{A^k}\beta^{k+1}_{A^k} - y\Big)/n\|_2 \\
&= \|X'_{A^k_{22}}\Big(X_{A^k}\triangle^k_{A^k} + X_{A^k}\bar\beta^*_{A^k} - X_{A^\circ}\bar\beta^*_{A^\circ} - \bar\eta\Big)/n\|_2 \\
&= \|X'_{A^k_{22}}\Big(X_{A^k}\triangle^k_{A^k} - X_{A^k_{22}}\bar\beta^*_{A^k_{22}} - X_{A^k_2\setminus A^k_{22}}\bar\beta^*_{A^k_2\setminus A^k_{22}} - \bar\eta\Big)/n\|_2 \\
&\geq c_-(|A^k_{22}|)\|\bar\beta^*_{A^k_{22}}\|_2 - \theta_{|A^k_{22}|,T}\|\triangle^k_{A^k}\|_2 - \theta_{l_k,l_k-|A^k_{22}|}\|\bar\beta^*_{A^k_2\setminus A^k_{22}}\|_2 - \|X_{A^k_{22}}\bar\eta/n\|_2 \\
&\geq c_-(T)\|\bar\beta^*_{A^k_{22}}\|_2 - \theta_{T,T}\|\triangle^k_{A^k}\|_2 - \theta_{T,T}D_2(A^k) - h_2(T),
\end{aligned}$$

where the first equality uses the definition of d^{k+1}, the second equality uses the the definition of $\triangle^k$ and y, the third equality is simple algebra, the first inequality uses the triangle inequality, (40) and the definition of $\theta_{a,b}$, and the last inequality follows from the monotonicity property of $c_-(\cdot)$, $\theta_{a,b}$ and the definition of $h_2(T)$. This proves (67).

Finally, we show (66). Let $i_k \in A^k_{22}$ be an index satisfying $\big|\bar\beta^*_{i_k}\big| = \|\bar\beta^*_{A^k_{22}}\|_\infty$. Then

$$\begin{aligned}
\Big|d^{k+1}_{i_k}\Big| &= \|X'_{i_k}(X_{A^k}\triangle^k_{A^k} - X_{i_k}\bar\beta^*_{i_k} - X_{A^k_2\setminus\, i_k}\bar\beta^*_{A^k_2\setminus\, i_k} - \bar\eta)/n\|_\infty \\
&\geq \big|\bar\beta^*_{i_k}\big| - T\mu\|\triangle^k_{A^k}\|_\infty - l_k\mu\|\bar\beta^*_{A^k_2\setminus\, i_k}\|_\infty - \|X'_{i_k}\bar\eta\|_\infty \\
&\geq \|\bar\beta^*_{A^k_{22}}\|_\infty - T\mu\|\triangle^k_{A^k}\|_\infty - T\mu D_\infty(A^k) - h_\infty(T),
\end{aligned}$$

where the first equality is derived from the first three equalities in the proof of (67) by replacing A^k_{22} with i_k, the first inequality follows from the triangle inequality and (44), and the last inequality follows from the definition of $h_\infty(T)$. Then (66) follows by rearranging the terms in the above inequality. This completes the proof of Lemma 24. ■

Lemma 25

$$\|\beta^k\|_\infty \vee \|d^k\|_\infty = \max\{|\beta^k_i| + |d^k_i| | i \in S\}, \forall k \geq 1. \tag{68}$$

$$\|\beta^{k+1}_{A^k_{11}}\|_\infty + \|d^{k+1}_{A^k_{22}}\|_\infty \leq \Big|\beta^{k+1}_{I^k_{33}}\Big|_{min} \wedge \Big|d^{k+1}_{I^k_{44}}\Big|_{min}. \tag{69}$$

$$\|\beta^{k+1}_{A^k_{11}}\|_2 + \|d^{k+1}_{A^k_{22}}\|_2 \leq \sqrt{2}\Big(\|\beta^{k+1}_{I^k_{33}}\|_2 + \|d^{k+1}_{I^k_{44}}\|_2\Big). \tag{70}$$

Proof By the definition of Algorithm 1 we have $\beta^k_i d^k_i = 0$, $\forall i \in S$, $\forall k \geq 1$, thus (68) holds. (69) follows from the definition of A^k_{11}, A^k_{22}, I^k_{33}, I^k_{44} and (68). Now

$$\begin{aligned}
\frac{1}{2}(\|\beta^{k+1}_{A^k_{11}}\|_2 + \|d^{k+1}_{A^k_{22}}\|_2)^2 &\leq \|\beta^{k+1}_{A^k_{11}}\|^2_2 + \|d^{k+1}_{A^k_{22}}\|^2_2 \\
&\leq (\|\beta^{k+1}_{I^k_{33}}\|_2 + \|d^{k+1}_{I^k_{44}}\|_2)^2,
\end{aligned}$$

where the first inequality follows from simple algebra, and the second inequality follows from (49) and (69). Thus (70) follows. This completes the proof of Lemma 25. ■

Lemma 26

$$\|\beta^{k+1}_{I^k_{33}}\|_2 \leq \|\triangle^k_{I^k_{33}}\|_2. \tag{71}$$

Furthermore, suppose (A1) holds. We have

$$\|d^{k+1}_{I^k_{44}}\|_\infty \leq T\mu\|\triangle^k_{A^k}\|_\infty + T\mu D_\infty(A^k) + h_\infty(T) \quad \textit{under the mutual coherence condition } (A^*), \tag{72}$$

$$\|d^{k+1}_{I^k_{44}}\|_2 \leq \theta_{T,T}\Big\|\triangle^k_{A^k}\Big\| + \theta_{T,T}D_2(A^k) + h_2(T) \quad \textit{if} \quad X \sim SRC(T, c_-(T), c_+(T)). \tag{73}$$

Proof By the definition of $\triangle^k$, the triangle inequality and the fact that $\bar\beta^*$ vanishes on $A^k \cap I^k_{33}$, we have

$$\|\beta^{k+1}_{I^k_{33}}\|_2 = \|\triangle^k_{I^k_{33}} + \bar\beta^*_{I^k_{33}}\|_2 \leq \|\triangle^k_{I^k_{33}}\|_2 + \|\bar\beta^*_{A^k\cap I^k_{33}}\|_2 = \|\triangle^k_{I^k_{33}}\|_2.$$

So (71) follows. Now

$$\begin{aligned}
\|d^{k+1}_{I^k_{44}}\|_2 &= \|X'_{I^k_{44}}\Big(X_{A^k}\triangle^k_{A^k} - X_{A^k_2}\bar\beta^*_{A^k_2} - \bar\eta\Big)/n\|_2 \\
&\leq \theta_{|I^k_{44}|,T}\|\triangle^k_{A^k}\|_2 + \theta_{|I^k_{44}|,l_k}\|\bar\beta^*_{A^k_2}\|_2 + \|X'_{I^k_{44}}\bar\eta\|_2 \\
&\leq \theta_{T,T}\|\triangle^k_{A^k}\|_2 + \theta_{T,T}D_2(A^k) + h_2(T),
\end{aligned}$$

where the first equality is derived from the first three equalities in the proof of (67) by replacing A^k_{22} with I^k_{44}, the first inequality follows from the triangle inequality and the definition of $\theta_{a,b}$, and the last inequality follows from the monotonicity property of $\theta_{a,b}$ and $h_2(T)$. This implies (73). Finally, (72) can be proved similarly by using (44) and (53). This completes the proof of Lemma 26. ■

Lemma 27 *Suppose (A1) holds.*

(i) If $X \sim SRC(T, c_-(T), c_+(T))$, then

$$D_2(A^{k+1}) \leq \gamma D_2(A^k) + \frac{\gamma}{\theta_{T,T}}h_2(T), \tag{74}$$

(ii) If $(T-1)\mu < 1$, then

$$D_\infty(A^{k+1}) \leq \gamma_\mu D_2(A^k) + \frac{3+2\mu}{1-(T-1)\mu}h_\infty(T). \tag{75}$$

Proof We have

$$\begin{aligned}
D_2(A^{k+1}) &\le \|\bar{\beta}^*_{A^k_{11}}\|_2 + \|\bar{\beta}^*_{A^k_{22}}\|_2 \\
&\le (\|\beta^{k+1}_{A^k_{11}}\|_2 + \|d^{k+1}_{A^k_{22}}\|_2 + \|\triangle^k_{A^k_{11}}\|_2 + \theta_{T,T}\|\triangle^k_{A^k}\|_2 + \theta_{T,T}D_2(A^k) + h_2(T))/c_-(T) \\
&\le (\sqrt{2}(\|\beta^{k+1}_{I^k_{33}}\|_2 + \|d^{k+1}_{I^k_{44}}\|_2) + \|\triangle^k_{A^k_{11}}\|_2 + \theta_{T,T}\left\|\triangle^k_{A^k}\right\| + \theta_{T,T}D_2(A^k) + h_2(T))/c_-(T) \\
&\le ((2+(1+\sqrt{2})\theta_{T,T})\|\triangle^k\|_2 + (1+\sqrt{2})\theta_{T,T}D_2(A^k) + (1+\sqrt{2})h_2(T))/c_-(T) \\
&\le (\frac{2\theta_{T,T}+(1+\sqrt{2})\theta^2_{T,T}}{c_-(T)^2} + \frac{(1+\sqrt{2})\theta_{T,T}}{c_-(T)})D_2(A^k) \\
&+ (\frac{2+(1+\sqrt{2})\theta_{T,T}}{c_-(T)^2} + \frac{1+\sqrt{2}}{c_-(T)})h_2(T),
\end{aligned}$$

where the first inequality is (62), the second inequality follows from (64) and (67), the third inequality follows from (70), the fourth inequality uses the sum of (71) and (73), and the last inequality follows from (57). This implies (74) by noticing the definitions of γ.

Now

$$\begin{aligned}
D_\infty(A^{k+1}) &\le \|\bar{\beta}^*_{A^k_{11}}\|_\infty + \|\bar{\beta}^*_{A^k_{22}}\|_2 \\
&\le \|\beta^{k+1}_{A^k_{11}}\|_\infty + \|d^{k+1}_{A^k_{22}}\|_\infty + \|\triangle^k_{A^k_{11}}\|_\infty + T\mu\|\triangle^k_{A^k}\|_\infty + T\mu D_\infty(A^k) + h_\infty(T). \\
&\le \|d^{k+1}_{I^k_{44}}\|_\infty + \|\triangle^k_{A^k_{11}}\|_\infty + T\mu\|\triangle^k_{A^k}\|_\infty + T\mu D_\infty(A^k) + h_\infty(T) \\
&\le \|\triangle^k_{A^k_{11}}\|_\infty + 2T\mu\|\triangle^k_{A^k}\|_\infty + 2T\mu D_\infty(A^k) + 2h_\infty(T) \\
&\le (\frac{(1+2T\mu)T\mu}{1-(T-1)\mu} + 2T\mu)D_\infty(A^k) + \frac{3+2\mu}{1-(T-1)\mu}h_\infty(T),
\end{aligned}$$

where the first inequality is (63), the second inequality follows from (65) and (66), the third inequality follows from (69), the fourth inequality follows from (72), and the last inequality follows from (59). Thus part (ii) of Lemma 27 follows by noticing the definition of γ_μ. This completes the proof of Lemma 27. ■

Proof of Theorem 4.

Proof Suppose $\gamma < 1$. By using (74) repeatedly,

$$\begin{aligned}
D_2(A^{k+1}) &\le \gamma D_2(A^k) + \frac{\gamma}{\theta_{T,T}}h_2(T) \\
&\le \gamma(\gamma D_2(A^{k-1}) + \frac{\gamma}{\theta_{T,T}}h_2(T)) + \gamma h_2(T) \\
&\le \cdots \\
&\le \gamma^{k+1}D_2(A^0) + \frac{\gamma}{\theta_{T,T}}(1+\gamma+\cdots+\gamma^k)h_2(T) \\
&< \gamma^{k+1}\|\bar{\beta}^*\|_2 + \frac{\gamma}{(1-\gamma)\theta_{T,T}}h_2(T),
\end{aligned}$$

i.e., (13) holds. Now

$$\begin{aligned}
\|\beta^{k+1} - \bar{\beta}^*\|_2 &\le (1+\frac{\theta_{T,T}}{c_-(T)})D_2(A^k) + \frac{h_2(T)}{c_-(T)} \\
&\le (1+\frac{\theta_{T,T}}{c_-(T)})\Big[\gamma^k\|\bar{\beta}^*\|_2 + \frac{\gamma\theta_{T,T}}{1-\gamma}h_2(T)\Big] \\
&= (1+\frac{\theta_{T,T}}{c_-(T)})\gamma^k\|\bar{\beta}^*\|_2 + \Big[\frac{\gamma\theta_{T,T}}{(1-\gamma)}(1+\frac{\theta_{T,T}}{c_-(T)}) + \frac{1}{c_-(T)}\Big]h_2(T),
\end{aligned}$$

where the first inequality follows from (56), the second inequality follows from (13), and the third line follows after some algebra. Thus (14) follows by noticing the definitions of b_1 and b_2. This completes the proof of part (i) of Theorem 4.

For part (ii), (16) follows from (13) and (52), (17) follows from (14) and (52). This completes the proof of Theorem 4. ■

Proof of Corollary 8.

Proof By (14),

$$\begin{aligned}
\|\beta^{k+1} - \bar{\beta}^*\|_2 &\le b_1\gamma_1^k\|\bar{\beta}^*\|_2 + b_2h_2(T) \\
&\le b_1h_2(T) + b_2h_2(T) \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}}\frac{\sqrt{J}\bar{M}}{h_2(T)}
\end{aligned}$$

where the second inequality follows after some algebra. By (13),

$$\begin{aligned}
\|\bar{\beta}^*|_{A^*_J\setminus A^k}\|_2 &\le \gamma^k\|\bar{\beta}^*\|_2 + \frac{\gamma\theta_{T,T}}{1-\gamma}h_2(T) \\
&\le \gamma^k\sqrt{J}\bar{M} + \xi\bar{m} \\
&< \bar{m} \quad \text{if} \quad k \ge \log_{\frac{1}{\gamma}}\frac{\sqrt{J}R}{1-\xi},
\end{aligned}$$

where the second inequality follows from the assumption $\bar{m} \ge \frac{\gamma h_2(T)}{(1-\gamma)\theta_{T,T}\xi}$ with $0 < \xi < 1$, and the last inequality follows after some simple algebra. This implies $A^*_J \subset A^k$ if $k \ge \log_{\frac{1}{\gamma}}\frac{\sqrt{J}R}{1-\xi}$. This proves part (i). The proof of part (ii) is similar to that of part (i) by using (52), we omit it here. For part (iii), suppose β^* is exactly K-sparse and $T = K$ in the SDAR algorithm (Algorithm 1). It follows from part (ii) that with probability at least $1-2\alpha$, $A^* = A^k$ if $k \ge \log_{\frac{1}{\gamma}}\frac{\sqrt{K}R}{1-\xi}$. Then part (iii) holds by showing that $A^{k+1} = A^*$. Indeed, by (74) and (52) we have

$$\begin{aligned}
\|\bar{\beta}^*|_{A^*\setminus A^{k+1}}\|_2 &\le \gamma\|\bar{\beta}^*|_{A^*\setminus A^k}\|_2 + \frac{\gamma}{\theta_{T,T}}\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n} \\
&= \frac{\gamma}{\theta_{T,T}}\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n}.
\end{aligned}$$

Then $A^{k+1} = A^*$ follows from the assumption that $m \ge \frac{\gamma}{(1-\gamma)\theta_{T,T}\xi}\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n} > \frac{\gamma}{\theta_{T,T}}\sigma\sqrt{K}\sqrt{2\log(p/\alpha)/n}$. This completes the proof of Corollary 8. ■

Proof of Theorem 12.

Proof For μ satisfying $T\mu \leq 1/4$, some algebra shows $\gamma_\mu < 1$ and $\frac{1+\mu}{1-(T-1)\mu} < \frac{3+2\mu}{1-(T-1)\mu} < 4$. Now Theorem 12 can be proved similarly to Theorem 4 by using (75), (53) and (58). We omit it here. This completes the proof of Theorem 12. ■

Proof of Corollary 14.

Proof The proofs of part (i) and part (ii) are similar to those of Corollary 8, we omit them here. Suppose β^* is exactly K-sparse and $T = K$ in SDAR. It follows from part (ii) that with probability at least $1-2\alpha$, $A^* = A^k$ if $k \geq \log_{\frac{1}{\gamma_\mu}} \frac{R}{1-\xi}$. Then part (iii) holds by showing that $A^{k+1} = A^*$. By (75), (53) and $\frac{3+2\mu}{1-(T-1)\mu} < 4$ we have

$$\begin{aligned}\|\bar{\beta}^*|_{A^*\setminus A^{k+1}}\|_\infty &\leq \gamma_\mu \|\bar{\beta}^*|_{A^*\setminus A^k}\|_\infty + 4\sigma\sqrt{2\log(p/\alpha)/n} \\ &= 4\sigma\sqrt{2\log(p/\alpha)/n}.\end{aligned}$$

Then $A^{k+1} = A^*$ by the assumption that

$$m \geq \frac{4}{\xi(1-\gamma_\mu)}\sigma\sqrt{2\log(p/\alpha)/n} > 4\sigma\sqrt{2\log(p/\alpha)/n}.$$

This completes the proof of Corollary 14. ■

References

Alekh Agarwal, Sahand Negahban, and Martin J Wainwright. Fast global convergence of gradient methods for high-dimensional statistical recovery. *The Annals of Statistics*, 40 (5):2452–2482, 2012.

Dimitris Bertsimas, Angela King, and Rahul Mazumder. Best subset selection via a modern optimization lens. *The annals of statistics*, 44(2):813–852, 2016.

Thomas Blumensath and Mike E Davies. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications*, 14(5-6):629–654, 2008.

Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

Patrick Breheny and Jian Huang. Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The annals of applied statistics*, 5(1):232, 2011.

T Tony Cai and Lie Wang. Orthogonal matching pursuit for sparse signal recovery with noise. *IEEE Transactions on Information theory*, 57(7):4680–4688, 2011.

Emmanuel J Candes and Terence Tao. Decoding by linear programming. *IEEE transactions on information theory*, 51(12):4203–4215, 2005.

Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on pure and applied mathematics*, 59(8):1207–1223, 2006.

Scott Shaobing Chen, David L Donoho, and Michael A Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1):33, 1998.

Xiaojun Chen, Dongdong Ge, Zizhuo Wang, and Yinyu Ye. Complexity of unconstrained l2-lp minimization. *Mathematical Programming*, 143(1-2):371–383, 2014.

David L Donoho and Yaakov Tsaig. Fast solution of l1 norm minimization problems when the solution may be sparse. *IEEE Transactions on Information Theory*, 54(11):4789–4812, 2008.

David L Donoho, Michael Elad, and Vladimir N Temlyakov. Stable recovery of sparse overcomplete representations in the presence of noise. *IEEE Transactions on information theory*, 52(1):6–18, 2006.

Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association*, 96(456):1348–1360, 2001.

Jianqing Fan and Jinchi Lv. Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5): 849–911, 2008.

Jianqing Fan and Heng Peng. Nonconcave penalized likelihood with a diverging number of parameters. *The Annals of Statistics*, 32(3):928–961, 2004.

Jianqing Fan, Lingzhou Xue, and Hui Zou. Strong oracle optimality of folded concave penalized estimation. *Annals of statistics*, 42(3):819, 2014.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *The Annals of Applied Statistics*, 1(2):302–332, 2007.

Wenjiang J Fu. Penalized regressions: the bridge versus the lasso. *Journal of computational and graphical statistics*, 7(3):397–416, 1998.

Rahul Garg and Rohit Khandekar. Gradient descent with sparsification: an iterative algorithm for sparse recovery with restricted isometry property. pages 337–344, 2009.

Gilles Gasso, Alain Rakotomamonjy, and Stéphane Canu. Recovering sparse signals with a certain family of nonconvex penalties and dc programming. *IEEE Transactions on Signal Processing*, 57(12):4686–4698, 2009.

Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU Press, 2012.

Inc Gurobi Optimization. Gurobi optimizer reference manual. *URL http://www. gurobi. com*, 2015.

Jian Huang, Joel L Horowitz, and Shuangge Ma. Asymptotic properties of bridge estimators in sparse high-dimensional regression models. *The Annals of Statistics*, 36(2):587–613, 2008a.

Jian Huang, Shuangge Ma, and Cun-Hui Zhang. Adaptive lasso for sparse high-dimensional regression models. *Statistica Sinica*, pages 1603–1618, 2008b.

Jian Huang, Yuling Jiao, Bangti Jin, Jin Liu, Xiliang Lu, and Can Yang. A unified primal dual active set algorithm for nonconvex sparse recovery. *arXiv:1310.1147v4*, 2018.

David R Hunter and Runze Li. Variable selection using mm algorithms. *Annals of statistics*, 33(4):1617, 2005.

Prateek Jain, Ambuj Tewari, and Purushottam Kar. On iterative hard thresholding methods for high-dimensional m-estimation. *Advances in Neural Information Processing Systems*, pages 685–693, 2014.

Yuling Jiao, Bangti Jin, and Xiliang Lu. A primal dual active set with continuation algorithm for the l0-regularized optimization problem. *Applied and Computational Harmonic Analysis*, 39(3):400–426, 2015.

Yongdai Kim, Hosik Choi, and Hee-Seok Oh. Smoothly clipped absolute deviation on high dimensions. *Journal of the American Statistical Association*, 103(484):1665–1673, 2008.

Kenneth Lange, David R Hunter, and Ilsoon Yang. Optimization transfer using surrogate objective functions. *Journal of computational and graphical statistics*, 9(1):1–20, 2000.

Yufeng Liu and Yichao Wu. Variable selection via a combination of the l0 and l1 penalties. *Journal of Computational and Graphical Statistics*, 16(4):782–798, 2007.

Po-Ling Loh and Martin J Wainwright. Regularized m-estimators with nonconvexity: statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16(1):559–616, 2015.

Karim Lounici. Sup-norm convergence rate and sign concentration property of lasso and dantzig estimators. *Electronic Journal of statistics*, 2:90–102, 2008.

Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

Rahul Mazumder, Jerome H Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495): 1125–1138, 2011.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the lasso. *The annals of statistics*, 34:1436–1462, 2006.

Balas Kausik Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

Yu Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming*, 140(1):125–161, 2013.

Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Minimax rates of estimation for high-dimensional linear regression over lq balls. *IEEE transactions on information theory*, 57 (10):6976–6994, 2011.

Yiyuan She. Thresholding-based iterative selection procedures for model selection and shrinkage. *Electronic Journal of statistics*, 3:384–415, 2009.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 58(1):267–288, 1996.

Andreas M Tillmann and Marc E Pfetsch. The computational complexity of the restricted isometry property, the nullspace property, and related concepts in compressed sensing. *IEEE Transactions on Information Theory*, 60(2):1248–1259, 2014.

Joel A Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information theory*, 50(10):2231–2242, 2004.

Martin J Wainwright. Sharp thresholds for high-dimensional and noisy sparsity recovery using l1-constrained quadratic programming (lasso). *IEEE transactions on information theory*, 55(5):2183–2202, 2009.

Lan Wang, Yongdai Kim, and Runze Li. Calibrating non-convex penalized regression in ultra-high dimension. *Annals of statistics*, 41(5):2505, 2013.

Zhaoran Wang, Han Liu, and Tong Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *Annals of statistics*, 42(6):2164, 2014.

Tong Tong Wu and Kenneth Lange. Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics*, 2(1):224–244, 2008.

Lin Xiao and Tong Zhang. A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*, 23(2):1062–1091, 2013.

Xiao-Tong Yuan, Ping Li, and Tong Zhang. Gradient hard thresholding pursuit. *Journal of Machine Learning Research*, 18:1–43, 2018.

Cun-Hui Zhang. Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics*, 38(2):894–942, 2010a.

Cun-Hui Zhang and Jian Huang. The sparsity and bias of the lasso selection in high-dimensional linear regression. *The Annals of Statistics*, 36(4):1567–1594, 2008.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, pages 576–593, 2012.

Tong Zhang. Some sharp performance bounds for least squares regression with l1 regularization. *The Annals of Statistics*, 37(5):2109–2144, 2009.

Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *Journal of Machine Learning Research*, 11:1081–1107, 2010b.

Tong Zhang. Adaptive forward-backward greedy algorithm for learning sparse representations. *IEEE transactions on information theory*, 57(7):4689–4708, 2011a.

Tong Zhang. Sparse recovery with orthogonal matching pursuit under rip. *IEEE Transactions on Information Theory*, 57(9):6215–6221, 2011b.

Peng Zhao and Bin Yu. On model selection consistency of lasso. *Journal of Machine learning research*, 7:2541–2563, 2006.

Hui Zou. The adaptive lasso and its oracle properties. *Journal of the American statistical association*, 101(476):1418–1429, 2006.

Hui Zou and Runze Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509, 2008.

Journal of Machine Learning Research 19 (2018) 1-38 Submitted 4/17; Revised 3/18; Published 8/18

Change-Point Computation for Large Graphical Models: A Scalable Algorithm for Gaussian Graphical Models with Change-Points

Leland Bybee lelandb@umich.edu
Yves Atchadé yvesa@umich.edu
Department of Statistics
University of Michigan,
1085 South University, Ann Arbor,
48109, MI, United States.

Editor: Mohammad Emtiyaz Khan

Abstract

Graphical models with change-points are computationally challenging to fit, particularly in cases where the number of observation points and the number of nodes in the graph are large. Focusing on Gaussian graphical models, we introduce an approximate majorize-minimize (MM) algorithm that can be useful for computing change-points in large graphical models. The proposed algorithm is an order of magnitude faster than a brute force search. Under some regularity conditions on the data generating process, we show that with high probability, the algorithm converges to a value that is within statistical error of the true change-point. A fast implementation of the algorithm using Markov Chain Monte Carlo is also introduced. The performances of the proposed algorithms are evaluated on synthetic data sets and the algorithm is also used to analyze structural changes in the S&P 500 over the period 2000-2016.

Keywords: change-points, Gaussian graphical models, proximal gradient, simulated annealing, stochastic optimization

1. Introduction

Networks are fundamental structures that are commonly used to describe interactions between sets of actors or nodes. In many applications, the behaviors of the actors are observed over time and one is interested in recovering the underlying network connecting these actors. High-dimensional versions of this problem where the number of actors is large (compared to the number of time points) is of special interest. In the statistics and machine learning literature, this problem is typically framed as fitting large graphical models with sparse parameters, and significant progress has been made recently, both in terms of the statistical theory (Meinshausen and Buhlmann, 2006; Yuan and Lin, 2007; Banerjee et al., 2008; Ravikumar et al., 2011; Hastie et al., 2015), and practical algorithms (Friedman et al., 2007; Höfling and Tibshirani, 2009; Atchade et al., 2017).

In many problems arising in areas such as biology, finance, and political sciences, it is well-accepted that the underlying networks of interest are not static, but can undergo changes over time. Graphical models with change-points (or piecewise constant graphical models) are simple, yet powerful models that are particularly well-suited for such problems, and different versions have been explored in the literature. In this work, similarly to Zhou et al. (2009); Kolar et al. (2010); Roy et al. (2017), we focus on settings where the change occurring at a given change-point is global in the sense that it affects the joint distribution of all nodes. This differs from the approach of Kolar and Xing (2012) where at a given change-point only the conditional distribution of a single node sees a change. Which framework is more appropriate depends in general on the application. For instance in biological applications where interests are often on single biomolecules, nodewise change-point analysis might be preferred, whereas in many social science problems global structural changes in the network is often of interest. We also mention the alternative approach of Liu et al. (2013) which has an original parametrization that focuses directly on the occurring change. Although we work within the joint-change framework, we stress that our proposed algorithms can be easily adapted to work with other alternative models.

Despite their conceptual simplicity, graphical models with change-points are computationally challenging to fit. For instance a full grid search approach to locate a single change-point in a Gaussian graphical model with a lasso penalty (glasso) requires solving $O(T)$ glasso sub-problems, where T is the number of time points. Most algorithms for the glasso problem scale like $O(p^3)$ or worst[1], where p is the number of nodes. Hence when p and T are large, fitting a high-dimensional Gaussian graphical model with a single change-point has a taxing computational cost of $O(Tp^3)$ per iteration.

The literature addressing the computational aspects of model-based change-point models is rather sparse. A large portion of change-point detection procedures are based on cumulative sums (CUSUM) or similar statistic-monitoring approaches (Lévy-Leduc and Roueff, 2009; Aue et al., 2009; Fryzlewicz, 2014; Chen and Zhang, 2015; Cho and Fryzlewicz, 2015, and the references therein). By and large, these change-point detection procedures can be efficiently implemented, and the computational difficulty aforementioned can be avoided. However in problems where one wishes to detect structural changes in large networks, a CUSUM-based or a statistic-based approach can be difficult to employ, since it requires knowledge of the pertinent statistics to monitor. Furthermore the estimation of the parameters in a model-based change-point models can provide new insight in the underlying phenomenon driving the changes. Hence CUSUM-based approaches may not be appropriate in applications where the main driving forces of the network changes are poorly understood, and/or are of prime interest.

Specific works addressing computational issues in model-based change-point estimation include Roy et al. (2017); Leonardi and Bühlmann (2016). In Roy et al. (2017) the authors considered a discrete graphical model with change-point and proposed a two-steps algorithm for computation. However the success of their algorithm depends crucially on the choice of the coarse and refined grids, and there is limited insight on how to choose these. A related work is Leonardi and Bühlmann (2016) where the authors considered a high-dimensional linear regression model with change-points and proposed a dynamic programming approach to compute the change points. In the case of a single change-point their algorithm corresponds to the brute force (full-grid search) approach mentioned above.

In this work we propose an approximate majorize-minimize (MM) algorithm for fitting piecewise constant high-dimensional models. The algorithm can be applied more broadly. However to focus the idea we limit our discuss to Gaussian graphical models with an elastic net penalty. In this specific setting, the algorithm takes the form of a block update algorithm that alternates between a proximal gradient update of the graphical model parameters followed by a line search of the change-point. The proposed algorithm only solves for a single change-point. We extend it to multiple change-points by binary segmentation. We study the convergence of the algorithm and show under some regularity conditions on the data generating mechanism that the algorithm is stable, and produces values in the

1. Furthermore the constant in the big-O is typically problem dependent and can be large

vicinity of the true change-point (under the assumption that one such true change-point exists).

Each iteration of the proposed algorithm has a computational cost of $O(Tp^2 + p^3)$. Although this cost is one order of magnitude smaller than the $O(Tp^3)$ cost of the brute force approach, it can still be large when p and T are both large. As a solution we propose a stochastic version of the algorithm where the line search performed to update the change-point is replaced by a Markov Chain Monte Carlo (MCMC)-based simulated annealing. The simulated annealing update is cheap (its computational cost per iteration is $O(p^2)$) and is used as a stochastic approximation of the full line search. We show by simulation that the stochastic algorithm behaves remarkably well, and as expected outperforms the deterministic algorithm is terms of computing time.

The paper is organized as follows. Section 2 contains a presentation of the Gaussian graphical model with change-points, followed by a detailed presentation of the proposed algorithms. We performed extensive numerical experiments to investigate the behavior of the proposed algorithms. We also use the algorithm to analyze structural changes in the Standard & Poors (S&P) 500 over the period 2000-2016. The results are reported in Section 3. We gather some of the technical proofs in Section 4.

We end this introduction with some notation that we shall use throughout the paper. We denote $\mathcal{M}_p$ the set of all symmetric elements of $\mathbb{R}^{p\times p}$ equipped with its Frobenius norm $\|\cdot\|_{\mathsf{F}}$ and associated inner product

$$\langle A, B\rangle_{\mathsf{F}} \stackrel{\text{def}}{=} \sum_{1\le i\le j\le p} A_{ij}B_{ij}.$$

We denote $\mathcal{M}_p^+$ the subset of $\mathcal{M}_p$ of positive definite elements. For $0 < a < A \le +\infty$, let $\mathcal{M}_p^+(a, A)$ denote the subset of $\mathcal{M}_p^+$ of matrices θ such that $\lambda_{\mathsf{min}}(\theta) \ge a$, and $\lambda_{\mathsf{max}}(\theta) \le A$, where $\lambda_{\mathsf{min}}(M)$ (resp. $\lambda_{\mathsf{max}}(M)$) denotes the smallest eigenvalue (resp. the largest eigenvalue) of M.

If $u \in \mathbb{R}^p$, and $q \in [1,\infty]$, we define $\|u\|_q \stackrel{\text{def}}{=} (\sum_{j=1}^p |u_j|^q)^{1/q}$ ($\|u\|_\infty \stackrel{\text{def}}{=} \max_{1\le j\le p}|u_j|$). For a matrix $\theta \in \mathbb{R}^{p\times p}$ and $q \in [1,\infty]\setminus\{2\}$, we define $\|\theta\|_q$ similarly by viewing θ as a $\mathbb{R}^{p^2}$ vector. For $q = 2$, $\|\theta\|_2$ denotes the spectral norm (operator norm) of θ.

2. Fitting Gaussian Graphical Models with a Single Change-Point

Let $\{X^{(t)},\ 1 \le t \le T\}$ be a sequence of p-dimensional random vectors. The grid over which the change-points are searched is denoted $\mathcal{T} \stackrel{\text{def}}{=} \{n_0, \ldots, T - n_0\}$, for some integer $1 \le n_0 < T$. We define

$$S_1(\tau) \stackrel{\text{def}}{=} \frac{1}{\tau}\sum_{t=1}^{\tau} X^{(t)}X^{(t)'},\quad S_2(\tau) \stackrel{\text{def}}{=} \frac{1}{T-\tau}\sum_{t=\tau+1}^{T} X^{(t)}X^{(t)'},\ \ \tau \in \mathcal{T}.$$

We define the regularization function as

$$\wp(\theta) \stackrel{\text{def}}{=} \alpha\|\theta\|_1 + \frac{1-\alpha}{2}\|\theta\|_{\mathsf{F}}^2,\quad \theta \in \mathcal{M}_p, \tag{1}$$

where $\alpha \in [0, 1)$ is a given constant, and $\|\theta\|_1 \stackrel{\text{def}}{=} \sum_{i\le j}^p |\theta_{ij}|$. Then we define

$$g_{1,\tau}(\theta) = \begin{cases} \frac{1}{2}\frac{\tau}{T}\left[-\log\det(\theta) + \mathsf{Tr}(\theta S_1(\tau))\right] & \text{if } \theta \in \mathcal{M}_p^+, \\ +\infty & \text{otherwise,}\end{cases},\quad \tau \in \mathcal{T},$$

where $\mathsf{Tr}(A)$ (resp. $\det(A)$) denotes the trace (resp. the determinant) of A, and

$$g_{2,\tau}(\theta) = \begin{cases} \frac{1}{2}\left(1-\frac{\tau}{T}\right)\left[-\log\det(\theta) + \mathsf{Tr}(\theta S_2(\tau))\right] & \text{if } \theta \in \mathcal{M}_p^+, \\ +\infty & \text{otherwise,}\end{cases},\quad \tau \in \mathcal{T}.$$

For $j \in \{1, 2\}$, we set

$$\hat\theta_{j,\tau} \stackrel{\text{def}}{=} \mathsf{Argmin}_{\ \vartheta\in\mathcal{M}_p^+}\left[g_{j,\tau}(\vartheta) + \lambda_{j,\tau}\wp(\vartheta)\right], \tag{2}$$

for regularization parameters $\lambda_{1,\tau} > 0, \lambda_{2,\tau} > 0$, that we assume fixed throughout. Note that due to the quadratic term in the elastic-net regularization (1), each of these minimization problems (2) is strongly convex. Hence for each $\tau \in \mathcal{T}$, and $j \in \{1, 2\}$, $\hat\theta_{j,\tau}$ is well-defined. We consider the problem of computing the change point estimate $\hat\tau$ defined as

$$\hat\tau = \mathsf{Argmin}_{\ \tau\in\mathcal{T}}\left[g_{1,\tau}(\hat\theta_{1,\tau}) + \lambda_{1,\tau}\wp(\hat\theta_{1,\tau}) + g_{2,\tau}(\hat\theta_{2,\tau}) + \lambda_{2,\tau}\wp(\hat\theta_{2,\tau})\right]. \tag{3}$$

If the minimization problem in (3) has more than one solution, then $\hat\tau$ denotes any one of these solutions. The quantity $\hat\tau$ is the maximum likelihood estimate of a change point τ in the model which assumes that $X^{(1)}, \ldots, X^{(\tau)}$ are independent with common distribution $\mathbf{N}(0, \theta_1^{-1})$, and $X^{(\tau+1)}, \ldots, X^{(T)}$ are independent with common distribution $\mathbf{N}(0, \theta_2^{-1})$, for an unknown change-point τ, and unknown precision matrices $\theta_1 \ne \theta_2$.

The problem of computing the graphical lasso (glasso) estimators $\hat\theta_{j,\tau}$ in (2) has received a lot of attention in the literature, and several efficient algorithms have been developed for this purpose (see for instance Atchadé et al., 2015, and the references therein). Hence in principle, using any of these available glasso algorithms, the change-point problem in (3) can be solved by solving $T - 2n_0 + 1 = O(T)$ glasso sub-problems. A similar algorithm is advocated in Leonardi and Bühlmann (2016) for fitting a high-dimensional linear regression model with change-points. However this brute force approach can be very time-consuming in cases where p and T are large. For instance, one of the most cost-efficient algorithm for solving the glasso problem in high-dimensional cases is the standard proximal gradient algorithm (Rolfs et al., 2012; Atchadé et al., 2015), which has a computational cost of $O(p^3\mathsf{cond}(\hat\theta)^2\log(1/\delta))$ to deliver a δ-accurate solution (that is $\|\theta-\hat\theta\|_{\mathsf{F}} \le \delta$), where $\mathsf{cond}(A)$ denotes the condition number of A, that is the ratio of the largest eigenvalue over the smallest eigenvalue of A. Hence when p and T are large the computational cost of the brute force approach for computing (3) is of order $O\left(Tp^3\mathsf{cond}(\hat\theta_{j,\tau})^2\log(1/\delta)\right)$, which can become prohibitively large.

We propose an algorithm that we show has a better computational complexity. To motivate the algorithm we first introduce a majorize-minimize (MM) algorithm for solving (3). We refer the reader to Wu and Lange (2010) for a general introduction to MM algorithms. Let

$$G(t) \stackrel{\text{def}}{=} g_{1,t}(\hat\theta_{1,t}) + \lambda_{1,t}\wp(\hat\theta_{1,t}) + g_{2,t}(\hat\theta_{2,t}) + \lambda_{2,\tau}\wp(\hat\theta_{2,t}),\ \ t \in \mathcal{T}$$

denote the objective function of the minimization problem in (3). For $\theta_1, \theta_2 \in \mathcal{M}_p$, we also define

$$\mathcal{H}(\tau|\theta_1, \theta_2) \stackrel{\text{def}}{=} g_{1,\tau}(\theta_1) + \lambda_{1,\tau}\wp(\theta_1) + g_{2,\tau}(\theta_2) + \lambda_{2,\tau}\wp(\theta_2),\ \ \tau \in \mathcal{T}. \tag{4}$$

Instead of the brute force approach that requires solving (2) for each value $\tau \in \mathcal{T}$, consider the following algorithm.

Algorithm 1 (MM algorithm) *Pick $\tau^{(0)} \in \mathcal{T}$, and for $k = 1, \ldots, K$, repeat the following steps.*

1. *Given $\tau^{(k-1)} \in \mathcal{T}$, compute $\hat{\theta}_{1,\tau^{(k-1)}}$ and $\hat{\theta}_{2,\tau^{(k-1)}}$, and minimize the function $\mathcal{H}(t|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}})$ to get $\tau^{(k)}$:*

$$\tau^{(k)} = \mathsf{Argmin}_{t\in\mathcal{T}}\, \mathcal{H}(t|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}}).$$

□

By definition of $\hat{\theta}_{j,\tau}$ in (2), we have $G(t) \leq \mathcal{H}(t|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}})$ for all $t \in \mathcal{T}$. Furthermore $G(\tau^{(k-1)}) = \mathcal{H}(\tau^{(k-1)}|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}})$. Therefore, for all $k \geq 1$,

$$G(\tau^{(k)}) \leq \mathcal{H}(\tau^{(k)}|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}}) \leq \mathcal{H}(\tau^{(k-1)}|\hat{\theta}_{1,\tau^{(k-1)}}, \hat{\theta}_{2,\tau^{(k-1)}}) = G(\tau^{(k-1)}).$$

Hence the objective function G is non-increasing along the iterates of Algorithm 1. Note that this algorithm is already potentially faster than the brute force approach, particular when T is large, since we compute the graphical-lasso solutions $\hat{\theta}_{j,\tau^{(k)}}$ only for time points visited along the iterations. We propose to further reduce the computational cost by computing the solutions $\hat{\theta}_{j,\tau^{(k)}}$ only approximately, by simple gradient updates.

Given $\gamma > 0$, and a matrix $\theta \in \mathbb{R}^{p\times p}$, define $\mathrm{Prox}_\gamma(\theta)$ (the proximal map with respect to the penalty function $\wp(\theta) = \alpha\|\theta\|_1 + (1-\alpha)\,\|\theta\|_{\mathsf{F}}^2/2)$ as the symmetric $\mathbb{R}^{p\times p}$ matrix such that for $1 \leq i, j \leq p$,

$$(\mathrm{Prox}_\gamma(\theta))_{ij} = \begin{cases} 0 & \text{if } |\theta_{ij}| < \alpha\gamma \\ \frac{\theta_{ij}-\alpha\gamma}{1+(1-\alpha)\gamma} & \text{if } \theta_{ij} \geq \alpha\gamma \\ \frac{\theta_{ij}+\alpha\gamma}{1+(1-\alpha)\gamma} & \text{if } \theta_{ij} \leq -\alpha\gamma\,. \end{cases}$$

We consider the following algorithm.

Algorithm 2 *[Approximate MM algorithm] Fix a step-size $\gamma > 0$. Pick some initial value $\tau^{(0)} \in \mathcal{T}$, $\theta_1^{(0)}, \theta_2^{(0)} \in \mathcal{M}_p^+$. Repeat for $k = 1, \ldots, K$. Given ($\tau^{(k-1)}$, $\theta_1^{(k-1)}$, $\theta_2^{(k-1)}$), do the following:*

1. *Compute*

$$\theta_1^{(k)} = \mathrm{Prox}_{\gamma\lambda_{1,\tau^{(k-1)}}}\left(\theta_1^{(k-1)} - \gamma\left(S_1(\tau^{(k-1)}) - (\theta_1^{(k-1)})^{-1}\right)\right),$$

2. *compute*

$$\theta_2^{(k)} = \mathrm{Prox}_{\gamma\lambda_{2,\tau^{(k-1)}}}\left(\theta_2^{(k-1)} - \gamma\left(S_2(\tau^{(k-1)}) - (\theta_2^{(k-1)})^{-1}\right)\right),$$

3. *compute*

$$\tau^{(k)} \stackrel{\text{def}}{=} \mathsf{Argmin}_{t\in\mathcal{T}}\, \mathcal{H}\left(t|\theta_1^{(k)}, \theta_2^{(k)}\right).$$

□

Note that, if instead of a single proximal gradient update in Step (1)-(2), we do a large number proximal gradient updates (an infinite number for the sake of the argument), we recover exactly Algorithm 1. Hence Algorithm 2 is an approximate version of Algorithm 1.

Remark 1 *1. Notice that one can easily compute $\mathcal{H}(\tau+1|\theta_1, \theta_2)$ from $\mathcal{H}(\tau|\theta_1, \theta_2)$ by a rank-one update in $O(p^2)$ number of operations. Hence the computational cost of Step (3) is $O(Tp^2)$. And the total computational cost of one iteration of Algorithm 2 is $O(p^3 + Tp^2)$.*

2. *In practice, and as with any gradient descent algorithm, one needs to exercise some care in choosing the step-size γ. Clearly, too small values of γ lead to slow convergence. However, choosing γ too large might cause the algorithm to diverge. Another (related) issue is how to guarantee that the matrices $\theta_1^{(k)}$ and $\theta_2^{(k)}$ maintain positive definiteness throughout the iterations. What we show below is that positive definiteness is automatically guaranteed if the step-size γ is taken small enough. A nice trade-off that works well from the software engineering viewpoint is to start with a large value of γ and to re-initialize the algorithm with a smaller γ if at some point positive definiteness is lost. This issue is discussed more extensively in Atchadé et al. (2015).*

As suggested in the remark above, Algorithm 2 raises two basic questions. The first question is whether the algorithm is stable, where here by stability we mean whether the algorithm runs without $\theta_1^{(k-1)}$ or $\theta_2^{(k-1)}$ losing positive definiteness. Indeed we notice that Steps (1 and 2) involve taking the inverse of the matrices $\theta_1^{(k-1)}$, and $\theta_2^{(k-1)}$, but there is no guarantee a priori that these matrices are non-singular. Using results established in Atchadé et al. (2015), we answer this question by showing below that if the step-size γ is small enough then the algorithm is actually stable. The second basic question is whether the algorithm converges to the optimal value. We address this question below.

For $j \in \{1, 2\}$, we set

$$\underline{\lambda}_j \stackrel{\text{def}}{=} \min_{\tau\in\mathcal{T}} \lambda_{j,\tau},\quad \bar{\lambda}_j \stackrel{\text{def}}{=} \max_{\tau\in\mathcal{T}} \lambda_{j,\tau},\quad \mu_j \stackrel{\text{def}}{=} \max_{\tau\in\mathcal{T}} \left[\frac{1}{2}\|S_j(\tau)\|_2 + \alpha p\lambda_{j,\tau}\right],$$

$$\mathsf{b}_j \stackrel{\text{def}}{=} \frac{-\mu_j + \sqrt{\mu_j^2 + 2\bar{\lambda}_j(1-\alpha)\frac{n_0}{T}}}{2(1-\alpha)\bar{\lambda}_j},\quad \mathsf{B}_j \stackrel{\text{def}}{=} \frac{\mu_j + \sqrt{\mu_j^2 + 2\underline{\lambda}_j(1-\alpha)}}{2(1-\alpha)\underline{\lambda}_j}.$$

Lemma 2 *Fix $j \in \{1, 2\}$. For all $\tau \in \mathcal{T}$, $\hat{\theta}_{j,\tau} \in \mathcal{M}_p^+(\mathsf{b}_j, +\infty)$. Let $\{(\theta_1^{(k)}, \theta_2^{(k)}),\ k \geq 0\}$ be the output of Algorithm 2. If the step-size γ satisfies $\gamma \in (0, \mathsf{b}_j^2]$, and $\theta_j^{(0)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$, then $\theta_j^{(k)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$, for all $k \geq 0$.*

Proof We present the proof for $j = 1$, the case $j = 2$ being similar. Note that $\hat{\theta}_{1,\tau}$ is the graphical elastic-net estimate based on data $X^{(1)}, \ldots, X^{(\tau)}$. The fact that $\hat{\theta}_{1,\tau}$ exists (and is unique) and satisfies the spectral bound $\lambda_{\mathsf{min}}(\hat{\theta}_{1,\tau}) \geq \mathsf{b}_1$ then follows from known results on the graphical elastic-net (see for instance Lemma 1 of Atchadé et al., 2015).

The second part of the lemma is similar to Lemma 2 of Atchadé et al. (2015). The idea is to show that if $\theta_1^{(k)} \in \mathcal{M}_p^+(\mathsf{b}_1, \mathsf{B}_1)$ then $\theta_1^{(k+1)} \in \mathcal{M}_p^+(\mathsf{b}_1, \mathsf{B}_1)$. This is proved as follows. Suppose that $\theta_1^{(k)} \in \mathcal{M}_p^+(\mathsf{b}_1, \mathsf{B}_1)$. Hence $\theta_1^{(k)}$ is non-singular. It is well-known (see for instance Parikh and Boyd, 2013, Section 4.2) that we can write $\theta_1^{(k+1)}$ as

$$\theta_1^{(k+1)} \quad = \quad \mathsf{Argmin}_{u\in\mathcal{M}_p}\left[\left\langle \nabla g_{1,\tau^{(k)}}(\theta_1^{(k)}), u - \theta_1^{(k)}\right\rangle + \frac{1}{2\gamma}\left\|u - \theta_1^{(k)}\right\|_{\mathsf{F}}^2 + \lambda_{1,\tau^{(k)}}\wp(u)\right].$$

The optimality conditions of this problem implies that there exists $Z \in \mathbb{R}^{p\times p}$, where $Z_{ij} \in [-1, 1]$ for all i, j such that

$$\nabla g_{1,\tau^{(k)}}(\theta_1^{(k)}) + \frac{1}{\gamma}\left(\theta_1^{(k+1)} - \theta_1^{(k)}\right) + \lambda_{1,\tau^{(k)}}\left(\alpha Z + (1-\alpha)\theta_1^{(k+1)}\right) = 0.$$

Since $\nabla g_{1,\tau}(\theta) = \frac{\tau}{2T}(S_1(\tau) - \theta^{-1})$, we re-arrange this optimality condition into:

$$\left(1 + (1-\alpha)\lambda_{1,\tau^{(k)}}\gamma\right)\theta_1^{(k+1)} \quad = \quad \theta_1^{(k)} \; + \; \frac{\gamma\tau^{(k)}}{2T}\left(\theta_1^{(k)}\right)^{-1} \; - \; \gamma\left(\frac{\tau^{(k)}}{2T}S_1(\tau^{(k)}) + \alpha\lambda_{1,\tau^{(k)}}Z\right).$$

Hence, if $\lambda_{\mathsf{min}}(\theta_1^{(k)}) \geq \mathsf{b}_1$, and $\mathsf{b}_1^2 \geq \gamma\tau/(2T)$ (which holds true if $\gamma \leq 2\mathsf{b}_1^2$), and using the fact that $\lambda_{\mathsf{min}}(A+B) \geq \lambda_{\mathsf{min}}(A) + \lambda_{\mathsf{min}}(B)$, we get

$$\lambda_{\mathsf{min}}(\theta_1^{(k+1)}) \geq \frac{1}{1 + (1-\alpha)\bar{\lambda}_1\gamma}\left(\mathsf{b}_1 + \frac{\gamma n_0}{2T}\frac{1}{\mathsf{b}_1} - \gamma\mu_1\right), \tag{5}$$

where $\mu_1 = \max_{\tau\in\mathcal{T}}\left[\frac{1}{2}\|S_1(\tau)\|_2 + \alpha p\lambda_{1,\tau}\right]$, using the fact that $\|Z\|_2 \leq p$. We note that as chosen, b_1 satisfies

$$(1-\alpha)\bar{\lambda}_1\mathsf{b}_1^2 + \mu_1\mathsf{b}_1 - \frac{n_0}{2T} = 0,$$

and this (with some easy algebra) implies that the right hand side of (5) is equal to b_1. Hence $\lambda_{\mathsf{min}}(\theta_1^{(k+1)}) \geq \mathsf{b}_1$. Similarly, if $\lambda_{\mathsf{max}}(\theta_1^{(k)}) \leq \mathsf{B}_1$, then

$$\lambda_{\mathsf{max}}(\theta_1^{(k+1)}) \leq \frac{1}{1 + (1-\alpha)\underline{\lambda}_1\gamma}\left(\mathsf{B}_1 + \frac{\gamma}{2}\frac{1}{\mathsf{B}_1} + \gamma\mu_1\right) = \mathsf{B}_1,$$

where the last equality follows from the fact that we have chosen B_1 such that

$$(1-\alpha)\underline{\lambda}_1\mathsf{B}_1^2 - \mu_1\mathsf{B}_1 - \frac{1}{2} = 0.$$

This completes the proof. ■

Remark 3 *The first statement of Lemma 2 implies that the change-point problem (3) has at least one solution. The second part shows that when the step-size γ is small enough, all the iterates of the algorithm remains positive definite. We note that the fact that $\alpha < 1$ is crucial in the arguments. The result remains true where $\alpha = 1$, however the arguments is slightly more involved (see Atchadé et al., 2015, Lemma 2). For simplicity we focus in this paper on the case $\alpha \in [0,1)$.*

We now address the issue of convergence. Clearly the function $t \mapsto \mathcal{H}(t|\theta_1,\theta_2)$ is not smooth, nor convex. This implies that Algorithm 2 cannot be analyzed using standard optimization tools. And indeed, we will not be able to establish that the output of Algorithm 2 converges to the minimizer $\hat{\tau}$. Rather, we introduce a containment assumption (Assumption H1) and we show that when it holds, then the output of Algorithm 2 converges to some neighborhood of the true change-point (the existence of this true change-point is part of the assumption).

H1 *There exist $\epsilon > 0$, $c \geq 0$, $\kappa \in [0,1)$, and $\tau_\star \in \mathcal{T}$ such that the following holds. For any $\tau \in \mathcal{T}$, and for any $\theta_1, \theta_2 \in \mathcal{M}_p^+$ such that $\left\|\theta_1 - \hat{\theta}_{1,\tau}\right\|_F + \left\|\theta_2 - \hat{\theta}_{2,\tau}\right\|_F \leq \epsilon$ we have*

$$|\mathsf{Argmin}_{\,t\in\mathcal{T}}\mathcal{H}(t|\theta_1,\theta_2) - \tau_\star| \leq \kappa|\tau - \tau_\star| + c. \tag{6}$$

Remark 4 *Plainly, what is imposed in H1 is the existence of a time point $\tau_\star \in \mathcal{T}$ (that we can view as the true change-point), such that anytime we take $\tau \in \mathcal{T}$ that is far from $\tau_\star$ in the sense that $|\tau - \tau_\star| > c/(1-\kappa)$, if θ_1, θ_2 are sufficiently close to the solutions $\hat{\theta}_{1,\tau}$ and $\hat{\theta}_{2,\tau}$ respectively, then computing $\mathsf{Argmin}_{\,t\in\mathcal{T}}\mathcal{H}(t|\theta_1,\theta_2)$ brings us closer to $\tau_\star$:*

$$|\mathsf{Argmin}_{\,t\in\mathcal{T}}\mathcal{H}(t|\theta_1,\theta_2) - \tau_\star| \leq \kappa|\tau - \tau_\star| + c < |\tau - \tau_\star|.$$

This containment assumption is akin to a curvature assumption on the function $t \mapsto \mathcal{H}(t|\theta_1,\theta_2)$ when θ_1 and θ_2 are reasonably close to $\hat{\theta}_{1,\tau}$, $\hat{\theta}_{2,\tau}$, respectively. The assumption seems realistic in settings where the data $X^{(1:T)}$ is indeed drawn from a Gaussian graphical model with true change-point $\tau_\star$, and parameters $\theta_{\star,1}$, $\theta_{\star,2}$. Indeed in this case, and if T is large enough, for any τ that is not too close to the boundaries, one expects $\hat{\theta}_{1,\tau}$ and $\hat{\theta}_{2,\tau}$ to be good estimates of $\theta_{\star,1}$ and $\theta_{\star,2}$, respectively. Therefore if $\left\|\theta_1 - \hat{\theta}_{1,\tau}\right\|_F + \left\|\theta_2 - \hat{\theta}_{2,\tau}\right\|_F \leq \epsilon$ for ϵ small enough, one expect as well θ_1 and θ_2 to be close to $\theta_{\star,1}$ and $\theta_{\star,2}$ respectively. Hence $\mathsf{Argmin}_{\,t\in\mathcal{T}}\mathcal{H}(t|\theta_1,\theta_2)$ should be close to $\mathsf{Argmin}_{\,t\in\mathcal{T}}\mathcal{H}(t|\theta_{\star,1},\theta_{\star,2})$, which in turn should be close to $\tau_\star$. Theorem 9 below will make this intuition precise.

□

In the next result we will see that in fact the iterates $\theta_1^{(k)}$ and $\theta_2^{(k)}$ closely track $\theta_{1,\tau^{(k)}}$ and $\theta_{2,\tau^{(k)}}$ respectively. Hence, when H1 holds Equation (6) guarantees that the sequence $\tau^{(k)}$ remains close to $\tau_\star$.

Theorem 5 *Suppose that $\gamma \in (0, b_1^2 \wedge b_2^2]$, and $\theta_j^{(0)} \in \mathcal{M}_p^+(b_j, B_j)$, for $j = 1,2$. Then*

$$\lim_k \left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_F = 0, \quad \lim_k \left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_F = 0.$$

Furthermore, if H1 holds then

$$\limsup_{k\to\infty}\left|\tau^{(k)} - \tau_\star\right| \leq \frac{c}{1-\kappa}.$$

Proof See Section 4.1 ■

Remark 6 *Note that the theorem does not guarantee that $\tau^{(k)}$ converges to $\tau_\star$, but rather its conclusion is that for k large $\tau^{(k)}$ stays within $c/(1-\kappa)$ of $\tau_\star$.*

We now address the question whether H1 is a realistic assumption. More precisely we will show that the argument highlighted in Remark 4 holds true under some regularity conditions. Suppose that $X^{(1:T)} \stackrel{\text{def}}{=} (X^{(1)}, \ldots, X^{(T)})$ are p-dimensional independent random variables such that

$$X^{(1)}, \ldots, X^{(\tau_\star)} \overset{i.i.d.}{\sim} \mathbf{N}(0, \theta_{\star,1}^{-1}), \quad \text{and} \quad X^{(\tau_\star+1)}, \ldots, X^{(T)} \overset{i.i.d.}{\sim} \mathbf{N}(0, \theta_{\star,2}^{-1}), \tag{7}$$

for some unknown change-point $\tau_\star$, and unknown symmetric positive definite precision matrices $\theta_{\star,1} \neq \theta_{\star,2}$. We set $\Sigma_{\star,j} \stackrel{\text{def}}{=} \theta_{\star,j}^{-1}$, and we let s_j denote the number of non-zero entries of $\theta_{\star,j}$, $j = 1,2$. For an integer $\iota \in \{1, \ldots, p\}$, we define the ι-th restricted eigenvalues of $\Sigma_{\star,j}$ as

$$\underline{\kappa}_j(\iota) \stackrel{\text{def}}{=} \inf\left\{u'(\Sigma_{\star,j})u,\ \|u\|_2 = 1,\ \|u\|_0 \leq \iota\right\},$$

$$\bar{\kappa}_j(\iota) \stackrel{\text{def}}{=} \sup\left\{u'(\Sigma_{\star,j})u,\ \|u\|_2 = 1,\ \|u\|_0 \leq \iota\right\}.$$

We set $s \stackrel{\text{def}}{=} \max(s_1, s_2)$, $\bar{\kappa} \stackrel{\text{def}}{=} \max(\bar{\kappa}_1(2), \bar{\kappa}_2(2))$, $\underline{\kappa} \stackrel{\text{def}}{=} \min(\underline{\kappa}_1(2), \underline{\kappa}_2(2))$, and we set the regularization parameter $\lambda_{j,\tau}$ as

$$\lambda_{1,\tau} \stackrel{\text{def}}{=} \frac{\bar{\kappa}}{\alpha T}\sqrt{48\tau\log(pT)}, \quad \lambda_{2,\tau} \stackrel{\text{def}}{=} \frac{\bar{\kappa}}{\alpha T}\sqrt{48(T-\tau)\log(pT)}, \quad \tau \in \mathcal{T}. \tag{8}$$

We need to assume that the parameter $\alpha \in [0,1)$ in the regularization term is large enough to produce approximately sparse solutions in (2). To that end, we assume that

$$\frac{\alpha}{1-\alpha} \geq \max\left(\|\theta_{\star,1}\|_\infty, \|\theta_{\star,2}\|_\infty\right). \tag{9}$$

Finally, we assume that the search domain $\mathcal{T}$ is such that for all $\tau \in \mathcal{T}$,

$$\min(\tau, T-\tau) \geq A_1^2 \log(pT), \tag{10}$$

where

$$A_1 \stackrel{\text{def}}{=} \max\left(2\left(\frac{\bar{\kappa}}{\underline{\kappa}}\right)^2, (1280)s^{1/2}\bar{\kappa}(\|\theta_{\star,1}\|_2 \vee \|\theta_{\star,2}\|_2)\right),$$

and

$$\bar{\kappa}\sqrt{\tau\log(pT)} \geq \frac{1}{2\sqrt{3}}(\tau-\tau_\star)_+\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty,$$
$$\text{and} \quad \bar{\kappa}\sqrt{(T-\tau)\log(pT)} \geq \frac{1}{2\sqrt{3}}(\tau_\star-\tau)_+\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty, \tag{11}$$

where $x_+ \stackrel{\text{def}}{=} \max(x, 0)$.

Remark 7 *Assumption (10) is a minimum sample size requirement. See for instance Ravikumar et al. (2011) Theorem 1, and 2 for similar conditions in standard Gaussian graphical model estimation. Here we require to have $\mathcal{T}$ such that $\min(\tau, T-\tau) = O(s\log(pT))$ for all $\tau \in \mathcal{T}$. This obviously implies that we need T to be at least $O(s\log(p))$. It is unclear whether the large constant 1280 in (10) is tight or simply an artifact of our proof techniques.*

To understand Assumption (11), note that for $\tau > \tau_\star$, the estimator $\hat{\theta}_{1,\tau}$ in (2) is based on misspecified data $X^{(\tau_\star+1)}, \ldots, X^{(\tau)}$. Hence if $\tau > \tau_\star$ is too far away from $\tau_\star$, the estimators $\hat{\theta}_{1,\tau}$ may behave poorly, particularly if $\theta_{\star,1}$ are $\theta_{\star,2}$ are very different. Assumption (11) rules out such settings, by requiring the search domains $\mathcal{T}$ to be roughly a $\sqrt{T}$ neighborhood of $\tau_\star$. Indeed, suppose that $\tau_\star = \rho_\star T$, for some $\rho_\star \in (0,1)$. Then it can be easily checked that any search domain of the form $(\tau_\star - r_1 T^{1/2}, \tau_\star + r_2 T^{1/2})$, satisfies (10) and (11) for T large enough, provided that

$$0 < r_1 \leq \frac{2\sqrt{3}\bar{\kappa}\sqrt{\rho_\star\log(pT)}}{\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty}, \quad \textit{and} \quad 0 < r_2 \leq \frac{2\sqrt{3}\bar{\kappa}\sqrt{(1-\rho_\star)\log(pT)}}{\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty}.$$

Of course, this search domain is difficult to use in practice since it depends on $\tau_\star$. In practice, we have found that taking $\mathcal{T}$ of the form $(rT, (1-r)T)$ for $r \leq 0.1$ works well, even though it is much wider than what is prescribed by our theory.

□

For $\tau \in \mathcal{T}$, let

$$r_{1,\tau} \stackrel{\text{def}}{=} A_2\bar{\kappa}\|\theta_{\star,1}\|_2^2\sqrt{\frac{s_1\log(pT)}{\tau}}, \quad r_{2,\tau} \stackrel{\text{def}}{=} A_2\bar{\kappa}\|\theta_{\star,2}\|_2^2\sqrt{\frac{s_2\log(pT)}{T-\tau}},$$

where A_2 is an absolute constant that can be taken as $16 \times 20 \times \sqrt{48}$. We set $b \stackrel{\text{def}}{=} \min(\lambda_{\mathsf{min}}(\theta_{\star,1}), \lambda_{\mathsf{min}}(\theta_{\star,2}))$, and $B \stackrel{\text{def}}{=} \max(\lambda_{\mathsf{max}}(\theta_{\star,1}), \lambda_{\mathsf{max}}(\theta_{\star,2}))$. We assume that for $j = 1,2$, and for $\tau \in \mathcal{T}$,

$$r_{j,\tau} \leq \min\left(\frac{\lambda_{\mathsf{min}}(\theta_{\star,j})}{4}, \frac{\|\theta_{\star,j}\|_\infty}{2}, \frac{\|\theta_{\star,j}\|_1}{1+8s_j^{1/2}}\right), \quad r_{j,\tau} \leq \frac{\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}}{2(1+8s^{1/2})}$$
$$\text{and} \quad r_{j,\tau} \leq A_2\left(\frac{b}{B}\right)^4\frac{\|\theta_{\star,j}\|_1}{s_j^{1/2}}. \tag{12}$$

Remark 8 *Condition (12) is mostly technical. As we will see below in Lemma 16, the term $r_{j,\tau}$ is the convergence rate toward $\theta_{\star,j}$ of the estimator $\hat{\theta}_{j,\tau}$, and is expected to converge to 0 with p, T (which implies that the sample size T cannot be too small compared to $\|\theta_{\star,j}\|_2^4 s_j\log(pT)$). Hence according to (12) the matrices $\theta_{\star,1}$ and $\theta_{\star,2}$ need to be such that the terms on the right-hand sides do no vanish faster than the rate $r_{j,\tau}$. In particular $\theta_{\star,1}$ and $\theta_{\star,2}$ should be well-conditioned so that $\lambda_{min}(\theta_{\star,j})$ and the ratio b/B do not decay too fast.*

Theorem 9 *Consider the output $\{(\theta_1^{(k)}, \theta_2^{(k)}),\ k \geq 0\}$ of Algorithm 2. Suppose that $\gamma \in (0, \mathsf{b}_1^2 \wedge \mathsf{b}_2^2]$, and $\theta_j^{(0)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$, for $j = 1,2$. Suppose that the statistical model underlying the data $X^{(1:T)}$ is as in (7), and that (8)-(12) hold. Suppose also that*

$$\|\theta_{\star,2} - \theta_{\star,1}\|_F \geq 8A_2\max\left[\left(\frac{\lambda_{min}(\theta_{\star,1})}{\lambda_{max}(\theta_{\star,1})}\right)^2\frac{\|\theta_{\star,1}\|_1}{s_1^{1/2}}, \left(\frac{\lambda_{min}(\theta_{\star,2})}{\lambda_{max}(\theta_{\star,2})}\right)^2\frac{\|\theta_{\star,2}\|_1}{s_2^{1/2}}\right]. \tag{13}$$

Then with probability at least $1 - \frac{8}{pT} - \frac{4}{p^2(1-e^{-C_0})}$, H1 holds with $\epsilon = (1/\sqrt{p})\min_{\tau\in\mathcal{T}}(r_{1,\tau} \wedge r_{2,\tau})$, $\kappa = 0$, and $c = 4\log(p)/C_0$, where

$$C_0 \stackrel{\text{def}}{=} \min\left[\frac{\|\theta_{\star,2}-\theta_{\star,1}\|_F^4}{128B^4\|\theta_{\star,2}-\theta_{\star,1}\|_1^2}, \left(\frac{\underline{\kappa}}{\bar{\kappa}}\right)^4\right].$$

In particular, with probability at least $1 - \frac{8}{pT} - \frac{4}{p^2(1-e^{-C_0})}$ we have

$$\limsup_{k\to\infty}\left|\tau^{(k)} - \tau_\star\right| \leq \frac{4}{C_0}\log(p), \tag{14}$$

Proof See Section 4.2. ■

Remark 10 *The main point of the theorem is that under the assumptions and data generation mechanism described above, the containment assumption H1 holds with probability as least $1 - \frac{8}{pT} - \frac{4}{p^2(1-e^{-C_0})}$, and where ϵ can be taken as $\min_\tau r_{1,\tau} \wedge r_{2,\tau}/\sqrt{p}$, $\kappa = 0$, and $c = 4\log(p)/C_0$. Conclusion (14) is then simply a consequence of Theorem 5. One should view (14) as saying that for k large, the output of Algorithm 2 fluctuates around $\tau_\star$, and the size of the fluctuation is $O(\log(p))$, under the assumed data generating mechanism. And we should stress that Algorithm 2 is not stochastic. Hence the randomness expressed in the theorem is with respect to the data generating mechanism.*

Remark 11 *We note that the bound in (14) grows with p. In classical change-point problems where p is fixed, and $T \to \infty$, it is known (see e.g. Bai, 1997) that with a fixed-magnitude change, the*

best one can achieve in estimating τ is $O(1)$. The rate in Theorem 9 suggests that in the high-dimensional setting where p grows the estimation rate for τ if of order $O(\log(p))$ (see also Roy et al., 2017). We believe that it is not possible to remove the additional $\log(p)$ factor, although to the best of our knowledge this question is still open. Note that it is customary in the change-point literature to take a re-scaled viewpoint and to define the change point as $a_\star \in (0,1)$ such that $\tau_\star = a_\star T$. In that setting the estimation rate for $a_\star$ is $O(1/T)$ in the classical fixed-dimensional fixed-magnitude change setting, and $O(\log(p)/T)$ in our setting.

2.1 A Stochastic Version

When T is much larger than p, Step 3 of Algorithm 2 becomes costly. In such cases, one can gain in efficiency by replacing Step 3 by a Monte Carlo approximation. We explore the use of simulated annealing to approximately solve Step 3 of Algorithm 2. Given $\theta_1, \theta_2 \in \mathcal{M}_p$, and $\beta > 0$, let $\pi_{\beta,\theta_1,\theta_2}$ denote the probability distribution on $\mathcal{T}$ defined as

$$\pi_{\beta,\theta_1,\theta_2}(\tau) = \frac{1}{Z_{\beta,\theta_1,\theta_2}} \exp\left(-\frac{\mathcal{H}(\tau|\theta_1,\theta_2)}{\beta}\right), \quad \tau \in \mathcal{T}.$$

Here, $Z_{\beta,\theta_1,\theta_2}$ is the normalizing constant, and $\beta > 0$ is the cooling parameter, that we shall drive down to zero with the iteration to increase the accuracy of the Monte Carlo approximation. Direct sampling from $\pi_{\beta,\theta_1,\theta_2}$ is typically possible, but this has the same computational cost as Step 3 of Algorithm 2. We will use a Markov Chain Monte Carlo approach which will allow us to make only a small number of calls of the function $\mathcal{H}$, per iteration. Let $\mathcal{K}_{\beta,\theta_1,\theta_2}$ denote a Markov kernel on $\mathcal{T}$ with invariant distribution $\pi_{\beta,\theta_1,\theta_2}$. Typically we will choose $\mathcal{K}_{\beta,\theta_1,\theta_2}$ as a Metropolis-Hastings Markov kernel (we give examples below).

We consider the following algorithm. As in Algorithm 2, γ is a given step-size. We choose a decrease sequence of temperature $\beta^{(k)}$ that we use along the iterations.

Algorithm 3 *Fix a step-size $\gamma > 0$, and a cooling sequence $\{\beta^{(k)}\}$. Pick some initial value $\tau^{(0)} \in \mathcal{T}$, $\theta_1^{(0)}, \theta_2^{(0)} \in \mathcal{M}_p^+$. Repeat for $k = 1, \ldots, K$. Given ($\tau^{(k-1)}$, $\theta_1^{(k-1)}$, $\theta_2^{(k-1)}$), do the following:*

1. *Compute*
$$\theta_1^{(k)} = \mathrm{Prox}_{\gamma\lambda_{1,\tau^{(k-1)}}}\left(\theta_1^{(k-1)} - \gamma\left(S_1(\tau^{(k-1)}) - (\theta_1^{(k-1)})^{-1}\right)\right),$$
2. *compute*
$$\theta_2^{(k)} = \mathrm{Prox}_{\gamma\lambda_{2,\tau^{(k-1)}}}\left(\theta_2^{(k-1)} - \gamma\left(S_2(\tau^{(k-1)}) - (\theta_2^{(k-1)})^{-1}\right)\right),$$
3. *draw*
$$\tau^{(k)} \sim \mathcal{K}_{\beta^{(k)},\theta_1^{(k)},\theta_2^{(k)}}(\tau^{(k-1)}, \cdot).$$

□

For most commonly used MCMC kernels, each iteration of Algorithm 3 has a computational cost of $O(p^3)$, which is better than $O(p^3 + Tp^2)$ needed by Algorithm 2, when $T \geq p$. However Algorithm 3 travels along the change-point space $\mathcal{T}$ more slowly. Hence overall, a larger number of iterations would typically be needed for Algorithm 3 to converge. Even after accounting for this slow convergence, Algorithm 3 is still substantially faster than Algorithm 2, as shown in Table 1 and 2. A rigorous analysis of the convergence of Algorithm 3 is beyond the scope of this work, and it left as a possible future research.

2.2 Extension to Multiple Change-Points

We extend the method to multiple change-points by binary segmentation. Binary segmentation is a standard method for detecting multiple change-points. The method proceeds by first searching for a single change-point. When a change-point is found the data is split into the two parts defined by the detected change-point. A similar search is then performed on each segment which can result in further splits. This recursive procedure continues until a certain stopping criterion is satisfied. Here we stop the recursion if

$$\ell_\tau + Cp \geq \ell_F,$$

where ℓ_τ is the penalized negative log-likelihood obtained with the additional change-point τ, and ℓ_F is the penalized negative log-likelihood without the change-point. The term Cp is a penalty term for model complexity, where C is a user-defined regularization parameter that controls the sparsity of the change-point model (the number of change-points). To the best of our knowledge there is no easy and principled approach for choosing C. We identify this as an important issue where more research is needed. Since C controls the number of change-points, in practice one ad-hoc approach is to set C such that the number of detected change-points is reasonable. This is the approach that we use in the real data analysis. Here we rely on simulation. We explore various scenarios by simulation and found that values of C between $(0, 4)$ produce the best results in our setting.

The binary segmentation algorithm can be defined more precisely as follows. Let us call $\mathcal{J}(X, t_0, t_1)$ the (single) change-point output either by Algorithm 3 or Algorithm 4 when applied to data set X using sample $X_{t_0}, \ldots, X_{t_1}$, for some $t_0, t_1 \in \mathcal{T}$, $t_0 < t_1$. Let $\mathcal{L}(X, t_0, t_1)$ denote the (penalized) minimum negative log-likelihood achieved on data $X_{t_0}, \ldots, X_{t_1}$. That is,

$$\mathcal{L}(X, t_0, t_1) = \min_{\theta \succ 0}\left[-\log\det(\theta) + \mathsf{Tr}\left(\theta\left(\frac{1}{t_1 - t_0 + 1}\sum_{t=t_0}^{t_1} X^{(t)}X^{(t)'}\right)\right) + \lambda\wp(\theta)\right].$$

Then the binary-segmentation algorithm $\mathcal{B}(X, t_0, t_1)$ can be written recursively as follows:

Algorithm 4 *Binary Segmentation*

1: ***function*** $\mathcal{B}(X, t_0, t_1)$
2: $\quad\tau = \mathcal{J}(X, t_0, t_1)$ *(apply either algorithm 3 or 4 to data $X_{t_0}, \ldots, X_{t_1}$)*
3: $\quad\ell_\tau = \mathcal{L}(X, t_0, \tau) + \mathcal{L}(X, \tau + 1, t_1)$
4: $\quad\ell_F = \mathcal{L}(X, t_0, t_1)$
5: $\quad$***if*** $\ell_\tau + Cp \geq \ell_F$ ***then***
6: $\quad\quad$***return*** *Null*
7: $\quad$***else***
8: $\quad\quad$***return*** $\{\tau, \mathcal{B}(X, t_0, \tau), \mathcal{B}(X, \tau + 1, t_1)\}$
9: $\quad$***end if***
10: ***end function***

We end this section with some words of caution. Binary segmentation is well-known to be a sub-optimal procedure and can perform poorly in some settings (see for instance Fryzlewicz, 2014). The issue is that at each step, binary segmentation is actually fitting a possibly misspecified model—one with a single change-point—to data with possibly multiple change-points. One approach is overcoming this limitation is to extend our proposed algorithms so as to handle directly multiple change-points. We leave this as an important future work.

3. Numerical Experiments

We investigate the different algorithms presented here in a variety of settings. For all the algorithms investigated the choice of the step-size γ and the regularizing parameter λ are important. For all experiments, and as suggested by (8), we found that setting $\lambda_{1,\tau} = \lambda\sqrt{\frac{\log\{p\}}{\tau}}$ and $\lambda_{2,\tau} = \lambda\sqrt{\frac{\log\{p\}}{T-\tau}}$ worked well. For the time-comparison in Section 3.1 we used $\lambda = 0.1$ and $\gamma = 3.5$ when $T = 1000$, and we used $\lambda = 0.01$ and $\gamma = 3.5$ when $T = 500$. For the remainder of the experiments we set $\lambda = 0.13$ and $\gamma = 0.25$. For all the experiments the search domain $\mathcal{T}$ is taken as $\{n_0, \ldots, T - n_0\}$, for a minimum sample size n_0 from $\{0.01T, 0.05T, 0.1T\}$.

We initialize $\tau^{(0)}$ to a randomly selected value in $\mathcal{T}$. The initial value $\theta_1^{(0)}$ and $\theta_2^{(0)}$ are taken as $\theta_j^{(0)} = (S_j(\tau^{(0)}) + \epsilon I)^{-1}$ where ϵ is a constant chosen to maintain positive definiteness. For cases where $p < \tau$ and $p < T - \tau$ we used $\epsilon = 0$, while for larger values of p we set $\epsilon = 0.2$.

For the data generation in the simulations, we typically choose $\tau_\star = T/2$ unless otherwise specified, and unless otherwise specified, we generate independently the matrices $\theta_{\star,1}$ and $\theta_{\star,2}$ as follows. First we generate a random symmetric sparse matrix M such that the proportion of non-zero entries is 0.25. We add 4 to all positive entries and subtract 4 from all negative entries. Then we set the actual precision matrix as $\theta_{\star,j} = M + (1 - \lambda_{\min}(M))I_p$ where $\lambda_{\min}(M)$ is the smallest eigenvalue of M. The resulting precision matrices contain roughly 25% non-zero off-diagonal elements. For each simulation a new pair of precision matrices was generated as well as the corresponding data set.

For Algorithm 3 we also experimented with a number of MCMC kernel $\mathcal{K}_{\beta,\theta_1,\theta_2}$. We experiment with the independence Metropolis sampler with proposal $\mathbf{U}(n_0, T - n_0)$. We also tried a Random Walk Metropolis with a truncated Gaussian proposal $\mathbf{N}(\tau^{(k-1)}, \sigma^2)$, for some scale parameter $\sigma > 0$. Finally, we also experimented with a mixture of these two Metropolis-Hastings kernels. We found that for our simulations the Independent Metropolis kernel works best, although the mixture kernel also performed well. For the cooling schedule of simulated annealing we use $\beta^{(0)} = 1$, and a geometric decay $\beta^{(n)} = \alpha\beta^{(n-1)}$ with $\alpha = \left(\frac{\beta^{(M)}}{\beta^{(0)}}\right)^{1/M}$ where $\beta^{(M)} = 0.001$, and M is the maximum number of iterations.

An implementation of the algorithms presented here for the Gaussian graphical model context is available in the changepointsHD package, Bybee (2017), available on the Comprehensive R Archive Network (CRAN).

3.1 Time Comparison

First we compare the running times of the proposed algorithms and the brute force approach. We consider two settings: $(p = 100, T = 1000)$ and $(p = 500, T = 500)$. In the setting $(p = 100, T = 1000)$, 100 independent runs of Algorithms 2 and 3 are performed and the average run-times are reported in Table 1. In the setting $(p = 500, T = 500)$ 10 independent runs of Algorithms 2 and 3 are used, and the results are presented in Table 2. We compare these times to results from one simulation run of the brute-force approach, the results of which are given in the description (caption) of Tables 1 and 2.

We consider two stopping criteria for Algorithm 2 or 3. The first criterion stops the iterations if

$$\frac{1}{T}|\tau^{(k)} - \tau_\star| < 0.005 \quad \text{and} \quad \frac{\|\theta_1^{(k)} - \hat{\theta}_1\|_F}{\|\hat{\theta}_1\|_F} + \frac{\|\theta_2^{(k)} - \hat{\theta}_2\|_F}{\|\hat{\theta}_2\|_F} < 0.05, \tag{V1}$$

where $\hat{\theta}_1$ and $\hat{\theta}_2$ are obtained by performing 1000 proximal-gradient steps at the true τ value. An interesting feature of the proposed approximate MM algorithms is that the

Variant		Approx. MM	Simulated Annealing
(V1)	Time (Seconds)	195.95 (48.94)	3.03 (0.40)
	Iterations	658.68 (82.93)	662.62 (88.51)
(V2)	Time (Seconds)	0.39 (0.10)	0.48 (0.46)
	Iterations	1.03 (0.17)	101.96 (100.29)

Table 1: Run-times of Algorithm 2 and 3 for $(p = 100, T = 1000)$. For comparison the run-time of the brute force algorithm for this problem is 2374.82.

Variant		Approx. MM	Simulated Annealing
(V1)	Time (Seconds)	3554.30 (404.24)	94.64 (5.50)
	Iterations	939.70 (11.03)	941.70 (16.23)
(V2)	Time (Seconds)	4.27 (1.10)	10.96 (8.26)
	Iterations	1.10 (0.32)	111.20 (90.71)

Table 2: Run-times of Algorithm 2 and 3 for $(p = 500, T = 500)$. For comparison the run-time of the brute force algorithm for this problem is 10854.44.

change-point sequence $\tau^{(k)}$ can converge well before $\theta_1^{(k)}$ and $\theta_2^{(k)}$. To illustrate this, we also explore the alternative approach of stopping the iterations only based on $\tau^{(k)}$, namely when

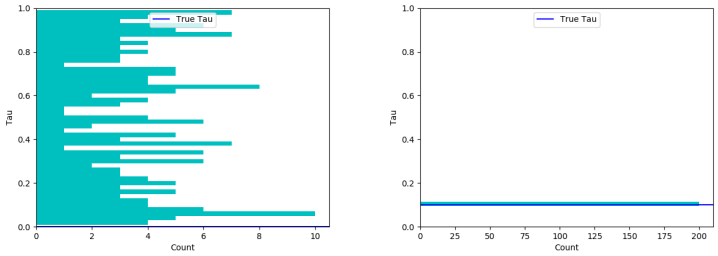
$$\frac{1}{T}|\tau^{(k)} - \tau_\star| < 0.005. \tag{V2}$$

Finally, we note that we implement the brute force approach by running 500 proximal-gradient steps for each possible value of τ. Note that 500 iterations is typically smaller than the number of iterations needed to satisfy (V1).

Tables 1 and 2 highlight the benefits of Algorithm 2 and Algorithm 3 as the run-time is several orders of magnitude lower than the brute force approach. Additionally, while Algorithm 3 requires more iterations than Algorithm 2 its run-time is typically smaller. The benefits of Algorithm 3 are particularly clear for large values of p and T (under stopping criterion (V1)). The stopping criteria (V2) highlights the fact that the $\tau^{(k)}$ sequence in the proposed algorithms can converge well before the θ-sequences.

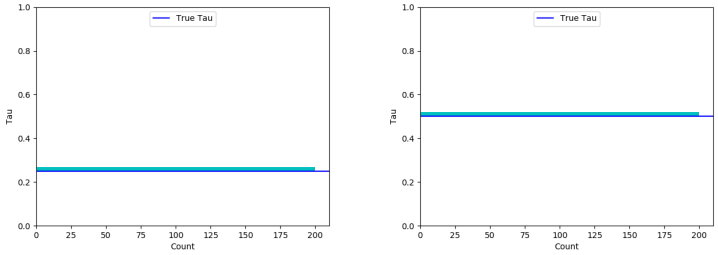
3.2 Behavior of the Algorithm when the Change-Point is at the Edge

We investigate how the brute force algorithm, Algorithm 2, and Algorithm 3 perform when change-points are non-existent or close to the edges. The results for the brute force algorithm are presented in Figure 1, the results for Algorithm 2 are presented on Figure 2 and the results for Algorithm 3 are presented on Figure 3. For Algorithm 2 and Algorithm 3 the figure contains two subfigures, the first showing the sequences $\{\tau^{(k)}\}$ of solutions produced by the algorithm (trace plots) for all 200 replications, and the second showing a histogram of the final change-point estimate, based on 200 replications. Additionally, a line is included to show the location of the true τ. The trace plots show how quickly each algorithm converges under the various settings. For the brute force algorithm the trace plot is not relevant since the brute force algorithm is not an iterative algorithm. The results suggest that Algorithm 2 and Algorithm 3 have more trouble when the true τ is close to the edge of the sample. For $\tau = 0.1T$, Algorithm 3 performed slightly better, with 136 simulations ending within 5 units of the true τ compared to 90 for Algorithm 2.



(a) No change-point

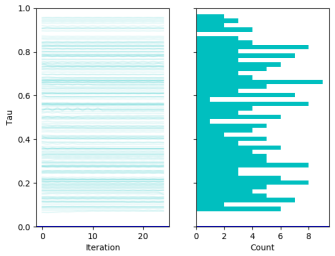
(b) Change-point at $\tau = 0.1T$



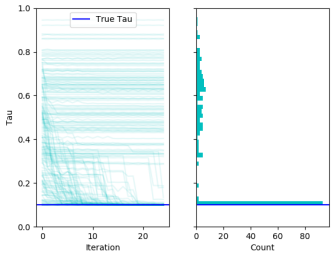
(c) Change-point at $\tau = 0.25T$

(d) Change-point at $\tau = 0.5T$

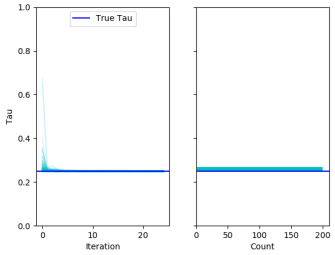
Figure 1: Behavior of the brute force approach as the location of the true change-point is varied. Each plot is a histogram of the change-point estimates based on 200 replications.



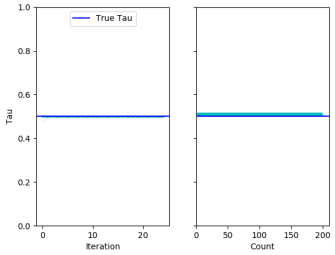
(a) No change-point



(b) Change-point at $\tau = 0.1T$

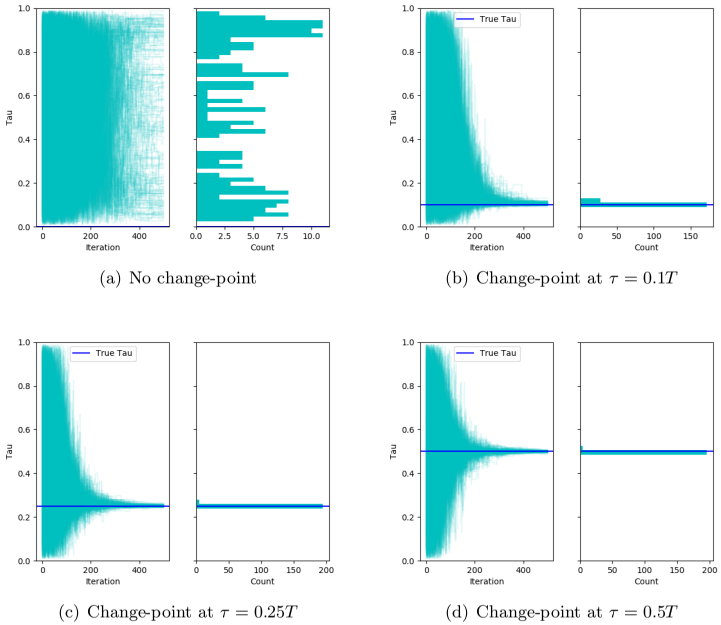


(c) Change-point at $\tau = 0.25T$



(d) Change-point at $\tau = 0.5T$

Figure 2: Behavior of Algorithm 2 as the location of the true change-point is varied. Each plot gives a trace plot of produced estimates, and a histogram of the final change-point estimate. Based on 200 replications.



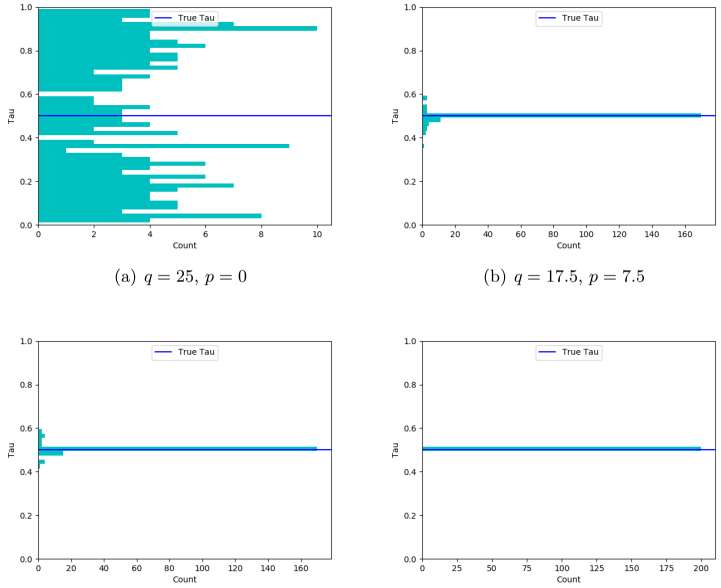
(a) No change-point

(b) Change-point at $\tau = 0.1T$

(c) Change-point at $\tau = 0.25T$

(d) Change-point at $\tau = 0.5T$

Figure 3: Behavior of Algorithm 3 as the location of the true change-point is varied. Each plot gives a trace plot of produced estimates, and a histogram of the final change-point estimate. Based on 200 replications.



(a) $q = 25$, $p = 0$

(b) $q = 17.5$, $p = 7.5$

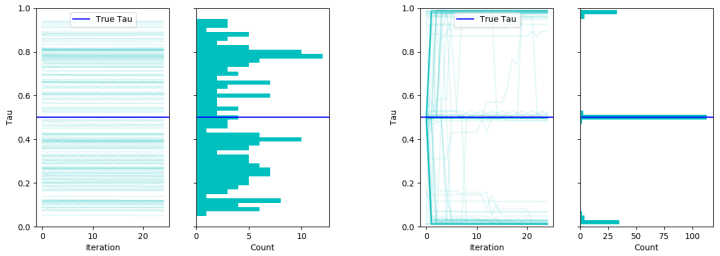
(c) $q = 10$, $p = 15$

(d) $q = 0$, $p = 25$

Figure 4: Behavior of the brute force approach for varying signals. Each plot is a histogram of the final change-point estimate. Based on 200 replications.

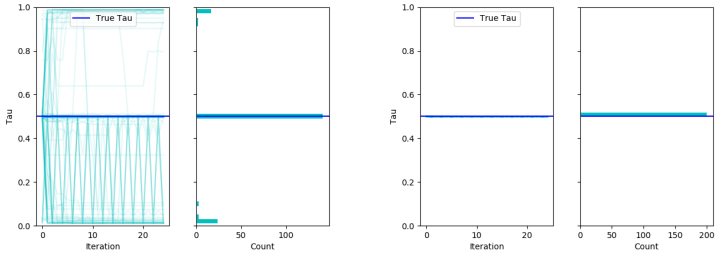
3.3 Behavior of the Algorithms when θ_1 and θ_2 are Similar

As θ_1 and θ_2 get increasingly similar, the location of the change-point becomes increasingly more difficult to find. We investigate the behavior of the proposed algorithms in such settings. We generate the true precision matrices θ_1 and θ_2 as follows. We draw a random precision matrix θ with $q\%$ non-zero off-diagonal elements, and C_1 and C_2 two random precision matrix with $p\%$ non-zero off-diagonal elements. We choose C_1 and C_2 to have the same diagonal elements. Then we set $\theta_1 = \theta + C_1$ and $\theta_2 = \theta + C_2$, which are then used to generate the data set for the experiment. The ratio p/q is a rough indication of the signal. Figure 4-6 show the behavior of the three algorithms for different values of q and p. For Algorithms 2 and 3 we found that similar precision matrices sometimes leads the algorithm to converge to the edge of the search domain. This makes sense, since a strong similarity between the two precision matrices implies a weak signal-to-noise ratio, which makes the model with no change-point more attractive. Putting the estimated change-point at the boundary of the search domain is roughly equivalent to fitting a model with no change-point.



(a) $q = 25$, $p = 0$

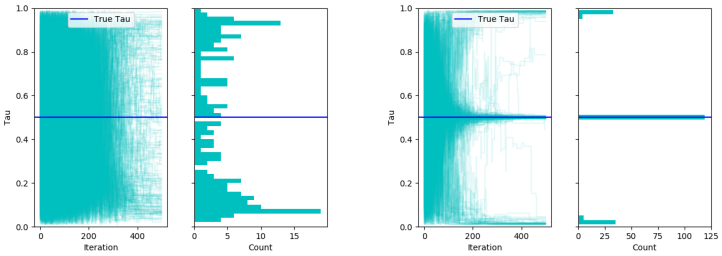
(b) $q = 17.5$, $p = 7.5$



(c) $q = 10$, $p = 15$

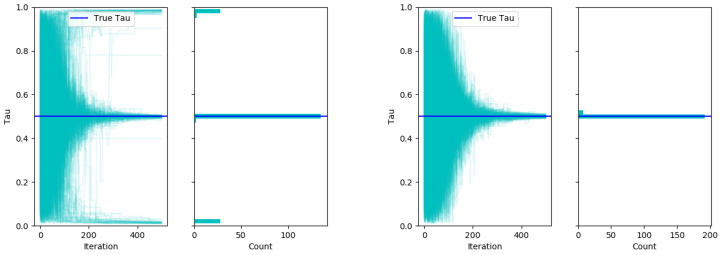
(d) $q = 0$, $p = 25$

Figure 5: Behavior of Algorithm 2 for varying signals. Each plot gives a trace plot of produced estimates, and a histogram of the final change-point estimate. Based on 200 replications.



(a) $q = 25$, $p = 0$

(b) $q = 17.5$, $p = 7.5$



(c) $q = 10$, $p = 15$

(d) $q = 0$, $p = 25$

Figure 6: Behavior of Algorithm 3 for varying signals. Each plot gives a trace plot of produced estimates, and a histogram of the final change-point estimate. Based on 200 replications.

3.4 Sensitivity to the Stopping Criteria in Binary Segmentation

This section considers the stopping condition for the binary segmentation algorithm (see Section 2.2) and how it performs with different configurations. A condition is required for determining when the binary segmentation splitting should reject a change-point and stop running. The stopping condition that we use is the following, stop if

$$\ell_\tau + Cp \geq \ell_F,$$

where ℓ_τ is the penalized negative log-likelihood obtained with the additional change-point τ, and ℓ_F is the penalized negative log-likelihood without the change-point. The term C is a user-defined parameter.

As mentioned above, the proposed algorithms can diverge when the step-size γ is not appropriately selected. In particular the appropriate value of γ is highly dependent on the length of the data set, and the binary segmentation splittings of the data can result in data segments with very different lengths. We use this feature to our advantage. We have chosen not to tune γ to the data segment, and to stop the binary segmentation splitting if the sequence $\hat{\theta}_1^{(k)}$ or $\hat{\theta}_2^{(k)}$ appear to diverge. This has the effect of constraining the lengths of the change-point segments from being too small. We achieve this result without directly setting a minimum length constraint—which be hard to do in practice. We found that stopping the algorithm when $||\hat{\theta}_i^{(k)}||_2^2 > 2 \times 10^3$ was sufficient for our data.

In the binary segmentation, since the estimates of θ_1 and θ_2 may not have converged by the end of the search for τ it may be worth continuing the estimation procedure for θ_1 and θ_2 so that the resulting penalized log-likelihoods are comparable. Hence after each split from the binary segmentation search, we perform an additional 500 iterations to estimate θ_1 and θ_2 at the resulting τ.

See Figure 7 for a series of heatmaps showing how often the binary segmentation method finds a given number of change-points for different values of C. These results suggest that the choice of C in the interval $(0, 4)$ is reasonable. These results are produced using Algorithm 3 for speed, however, the results are identical for the other two algorithms considered. Note that since an additional change-point should always improve the log-likelihood, when $C \leq 0$ we only stop on the secondary stopping condition that $||\hat{\theta}_i^{(k)}||_2^2 > 2 \times 10^3$.

3.5 High Dimensional Experiments

We also investigate the behavior of the proposed algorithms for larger values of p. We performed several (100) runs of Algorithm 3 for $T = 1000$, and $p \in \{100, 500, 750, 1000\}$. From these 100 runs we estimate the distributions of the iterates (by boxplots) after $10, 100, 200, \ldots, 1000$ iterations. The results are presented in Figure 8. The results show again a very quick convergence toward $\tau_\star$ and this convergence persists even as p gets large.

3.6 A Real Data Analysis

In finance and econometrics there is considerable interest in regime-switching models in the context of volatility, particularly because these switches may correspond to real events in the economy (Banerjee and Urga, 2005; Beltratti and Morana, 2006; Günay, 2014; Choi et al., 2010). However, much of the literature is limited to the low dimensional case, due to the difficulty involved in estimating change-points for higher dimensions. We are able to use our method to estimate change-points in the covariance structure of the Standard & Poor's (S&P) 500—an American stock market index.

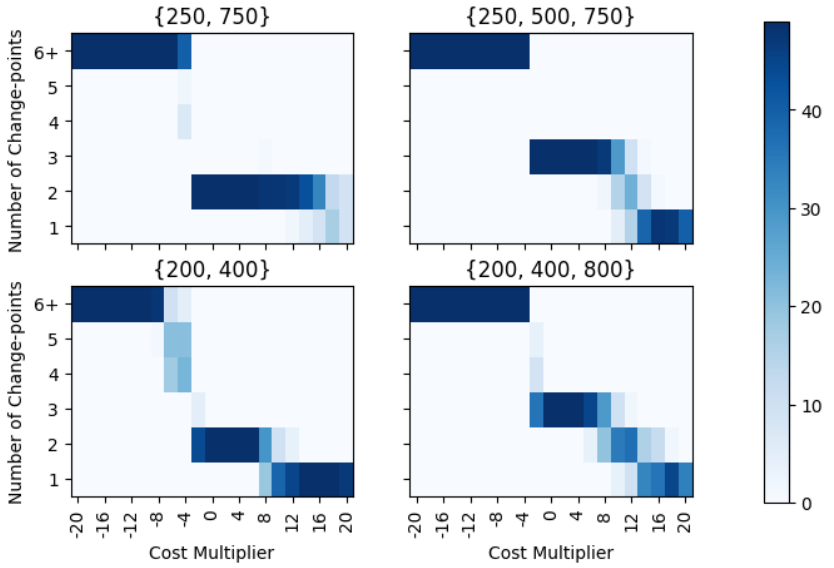


Figure 7: Number of change-points detected by binary segmentation as function of the cost multiplier C. The set of true change-points is indicated on top of the plots.

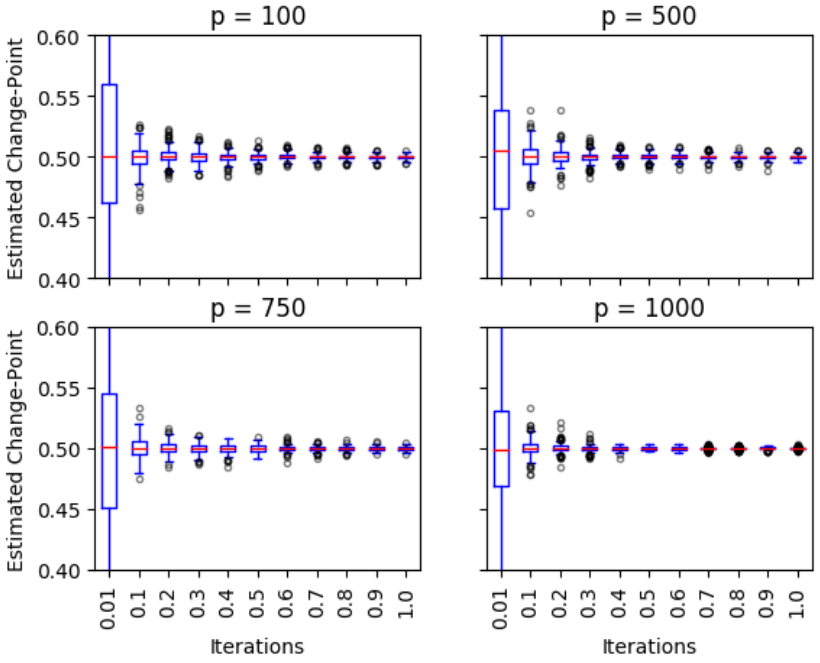


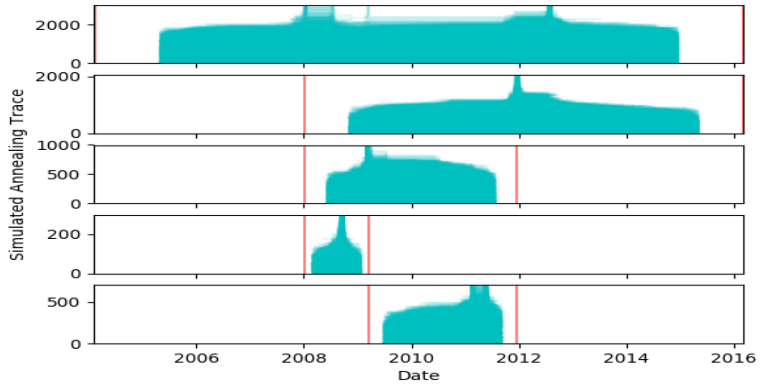
Figure 8: Boxplots of the iterates produced by Algorithm 4. Based on 100 replications.

Data from the S&P 500 was collected for the period from 2000-01-01 to 2016-03-03. From this initial sample a subset of stocks (or tickers) was selected for which at least 3000 corresponding observations exist. This produced a sample extending from 2004-02-06 to 2016-03-03, consisting of 3039 observations and 436 stocks. We follow a similar data cleaning procedure to Lafferty et al. (2012), who investigate a comparable problem without change-points. For each stock we generate the log returns, $\log \frac{X_t}{X_{t-1}}$, and standarize the resulting returns. Following Lafferty et al. (2012), we then truncate (or clip) all observations beyond three standard deviations of the same mean, thereby limiting unwanted outliers in our sample. The reason for this cleaning procedure is that these outliers often correspond to stock splits instead of meaningful price changes.

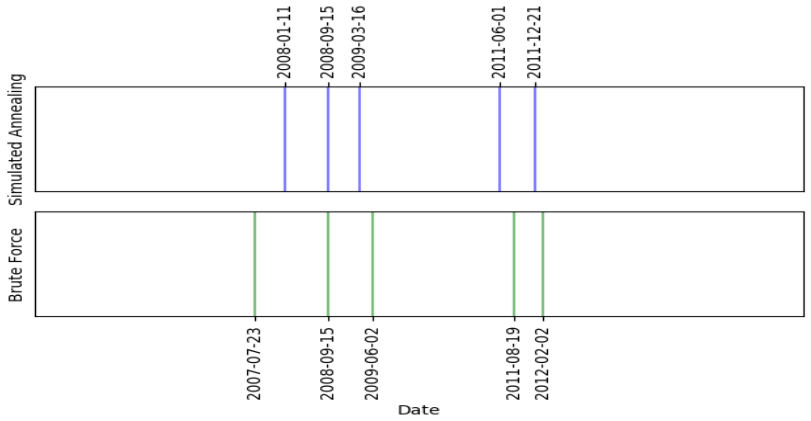
For our setting $\lambda = 0.002$ and $\gamma = 0.5$. We initialize $\hat{\theta}^{(0)} = (S(\tau^{(0)}) + I\epsilon)^{-1}$ where $\epsilon = 10^{-4}$ and $\tau^{(0)}$ is selected randomly. After the simulated annealing run the proximal gradient algorithm was run an additional 2000 steps, to produces estimates of θ_1 and θ_2. Here we increase the step-size to $\gamma = 350$ to accelerate the convergence. For the binary segmentation we found that selecting the threshold constant, $C = 0.005$, found a reasonable set of change-points. We found the choice of parameters important in this application, in particular, variation from the values used here can lead the algorithm to diverge. We use the same stopping criterion as with the prior binary-segmentation simulations. That is, a) stop when $\ell_\tau + Cp \geq \ell_F$ or b) stop when $||\hat{\theta}_i^{(k)}||_2^2 > 2 \times 10^3$.

Figure 9 presents the results of the change-point analysis using binary segmentation with Algorithm 4. As a reference we also present the results obtained using binary segmentation together with the brute force approach. For the brute force approach, we set $\gamma = 35$ and ran 10 iterations for each possible change-point, before running 2000 steps at $\gamma = 350$ to get the estimates for θ_1 and θ_2. The brute force approach took approximately an hour to run one layer of the search, while simulated annealing took approximately 15 minutes. Figure 9-(a) shows the trace plots from simulated annealing based on 100 replications. The red lines mark the detected time segments. Figure 9-(b) shows the resulting segmentation of the data. We note that simulated annealing and brute force produce slightly different sets of change-points. This brings up an important point: the resulting solution is a local optimum. Binary segmentation does introduce an element of path dependency to the results so there may be more than one viable set of change-points—in this particular case, the brute force approach starts with the first change-point on August 19th 2011 while simulated annealing starts with January 11th 2008.

We next look at how well the estimated change-points correspond to real world events. Our change-point set seems to do a good job of capturing both the Great Recession and a fall in stock prices during August of 2011 related to the European debt crisis and the downgrading of United State's credit-rating. The first change-point in our set is January 11th 2008. The National Bureau of Economic Research (NBER) identifies December of 2007 as the beginning of the Great Recession, which this change-point seems to capture. Additionally, 10 days after the change-point, the Financial Times Stock Exchange (FTSE) would experience its biggest fall since September 11th 2001. The brute force approach places this first change-point earlier in the series on July 23rd 2007, possibly capturing a relatively positive time in the economy before the downturn. The Second change-point occurred on September 15th 2008, the day on which Lehman Brothers filed for bankruptcy protection, one of the key events of the Great Recession (both methods agree on this change-point). The third change-point takes place on March 16th 2009, corresponding to the end of the bear market in the United States. For bthe brute force approach, this change-point is June 2nd 2009—June of 2009 was when the NBER officially declared the end of the recession. The fourth change-point, on June 1st 2011, and the fifth change-point, on December 21st 2011, likely capture a period of heightened concerns over the possible



(a) Simulated Annealing trace plots from 100 replications. The red lines represent the prior set of relevant change-points.



(b) Simulated annealing (top) and brute force segmentations of the data.

Figure 9: Change-points analysis of the S&P 500 data set over the period 2004-02-06 to 2016-03-03.

spread of the European debt crisis to Spain and Italy, during August of 2011. This period also saw the downgrading of the S&P's credit rating of the United States from AAA to

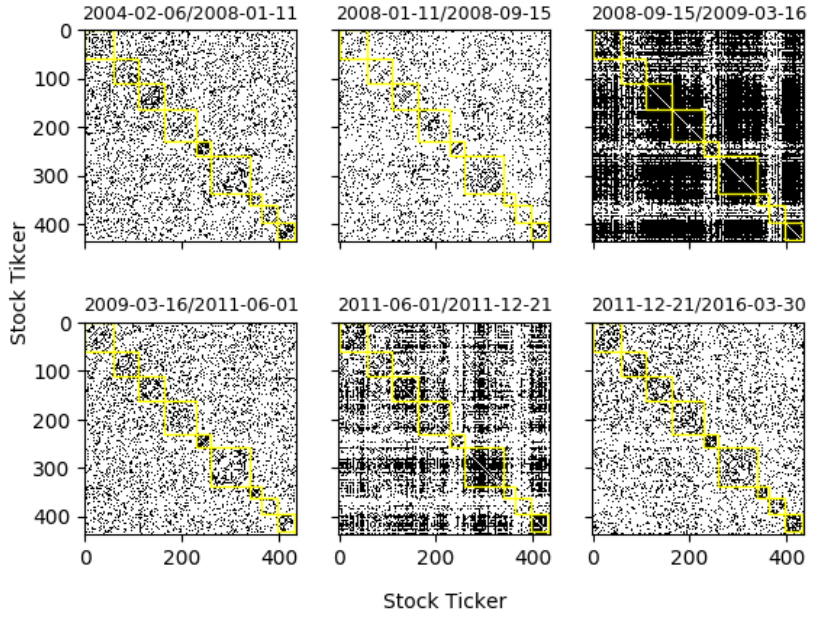


Figure 10: Adjacency matrices between stocks based on estimated precision matrices $\hat{\theta}$ for each time segment. A black dot represents an edge between two stocks.

AA+. The August 19th 2011 brute force change-point more precisely identifies this August downturn.

Given that the change-point set identified seems sensible, we then investigate what the corresponding $\hat{\theta}$ estimates look like, and whether any interesting conclusions can be drawn from our estimates. Here we focus only on the simulated annealing change-point set. See Figure 10 for a plot of the adjacency matrix for each $\hat{\theta}$ estimate. The black squares correspond to non-zero edges and he yellow boxes correspond to Global Industry Classification Standard (GICS) sectors. These results tell an intuitive story about how the economy behaves during financial crises. Following both the collapse of Lehamn Brother's and the events of August 2011, we see a dramatic increase in connectivity between returns even outside of GICS sectors. To get a better sense of this see Figure 11 for a similar series of plots where edges are summed over each sector. Figure 12 gives an expanded version of the summed edge plot for the first $\hat{\theta}$ estimate, as well as the corresponding sector labels for reference. Again, we can see that during periods of crisis, the off diagonal elements—corresponding to edges between different sectors—become more significant than during periods of general stability.

From these figures we can get a sense of which sectors are most affected during times of crisis. To expand upon this some, see Figure 12 for the edge count between each sector

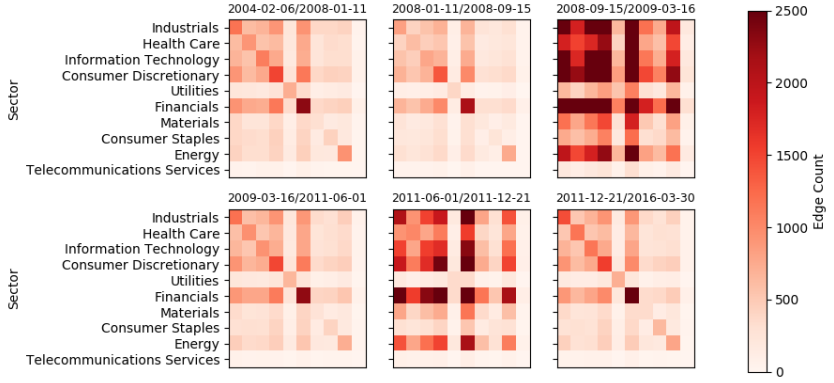


Figure 11: Adjacency matrices between sectors for each time segment. Based on the number of edges going from stocks of one sector to another as given by the estimated precision matrices $\hat{\theta}$.

and the Financial sector for each $\hat{\theta}$ estimate. We can see that during times of crisis, there is considerable connection between Industrials, Information Technology, Consumer Discretionary, and to a lesser extend Healthcare, and the Financial sector. Consumer Staples, Utilities, and Materials appear to be more stable during these periods and do not experience as much correlation with Financials. This might suggest that our method could be used as a tool to identify investment strategies that are likely to be resilient to periods of crisis in the market.

4. Proofs

4.1 Proof of Theorem 5

We will need the following lemma.

Lemma 12 *Set*

$$g(\theta) \stackrel{\text{def}}{=} -\log\det(\theta) + \mathit{Tr}(\theta S),$$

$$\textit{and} \quad \phi(\theta) \stackrel{\text{def}}{=} g(\theta) + \lambda\left[\alpha\|\theta\|_1 + \frac{1-\alpha}{2}\|\theta\|_F^2\right], \quad \theta \in \mathcal{M}_p^+,$$

for some symmetric matrix S, $\alpha \in (0,1)$, and $\lambda > 0$. Fix $0 < b < B \leq \infty$.

1. *For $\theta, \vartheta \in \mathcal{M}_p^+(b,B)$, we have*

$$g(\theta) + \langle \nabla g(\theta), \vartheta - \theta\rangle + \frac{1}{2B^2}\|\vartheta-\theta\|_F^2 \leq g(\vartheta) \leq g(\theta) + \langle \nabla g(\theta), \vartheta - \theta\rangle + \frac{1}{2b^2}\|\vartheta-\theta\|_F^2.$$

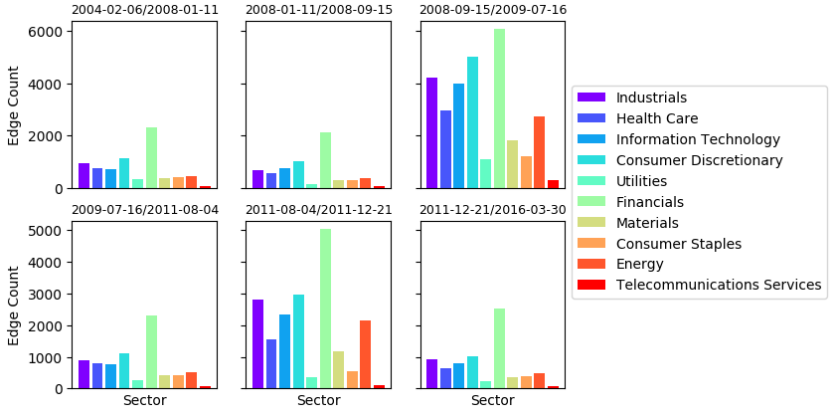


Figure 12: Number of edges between the financial sector and the remaining sectors, for each time segment. Based on the estimated precision matrices $\hat{\theta}$.

More generally, If $\theta, \vartheta \in \mathcal{M}_p^+$, then

$$g(\vartheta) - g(\theta) - \langle \nabla g(\theta), \vartheta - \theta\rangle \geq \frac{\|\vartheta-\theta\|_F^2}{4\|\theta\|_2\left(\|\theta\|_2 + \frac{1}{2}\|\vartheta-\theta\|_F\right)}.$$

2. *Let $\gamma \in (0, b^2]$, and $\theta, \bar{\theta}, \theta_0 \in \mathcal{M}_p^+(b,B)$. Suppose that*

$$\bar{\theta} = \mathrm{Prox}_{\gamma\lambda}\left(\theta - \gamma(S - \theta^{-1})\right),$$

then

$$2\gamma\left(\phi(\bar{\theta}) - \phi(\theta_0)\right) + \|\bar{\theta} - \theta_0\|_F^2 \leq \left(1 - \frac{\gamma}{B^2}\right)\|\theta - \theta_0\|_F^2.$$

Proof The first part of (1) is Lemma 12 of Atchadé et al. (2015), and Part (2) is Lemma 14 of Atchadé et al. (2015). The second part of (1) can be proved along similar lines. For completeness we give the details below.

Take $\theta_0, \theta_1 \in \mathcal{M}_p^+$. By Taylor expansion we have

$$g(\theta_1) - g(\theta_0) - \langle \nabla g(\theta_0), \theta_1 - \theta_0\rangle = -\int_0^1 \left\langle (\theta_0 + tH)^{-1} - \theta_0^{-1}, H\right\rangle \mathrm{d}t,$$

where $H \stackrel{\text{def}}{=} \theta_1 - \theta_0$. We have $(\theta_0 + tH)^{-1} - \theta_0^{-1} = -t\theta_0^{-1}H(\theta_0 + tH)^{-1}$, which leads to

$$g(\theta_1) - g(\theta_0) - \langle \nabla g(\theta_0), \theta_1 - \theta_0\rangle = \int_0^1 \mathrm{Tr}\left(\theta_0^{-1}H(\theta_0 + tH)^{-1}H\right) t\mathrm{d}t.$$

If $\theta_0 = \sum_{i=1}^p \rho_j u_j u_j'$ is the eigendecomposition of θ_0, we see that $\mathsf{Tr}\left(\theta_0^{-1} H(\theta_0+tH)^{-1}H\right) = \sum_{j=1}^p \frac{1}{\rho_j} u_j' H(\theta_0+tH)^{-1} H u_j$. Hence

$$g(\theta_1) - g(\theta_0) - \langle \nabla g(\theta_0), \theta_1 - \theta_0 \rangle \geq \sum_{j=1}^p \|Hu_j\|_2^2 \int_0^1 \frac{t\mathrm{d}t}{\|\theta_0\|_2 \left(\|\theta_0\|_2 + t\|H\|_{\mathsf{F}}\right)} \geq \frac{\sum_{j=1}^p \|Hu_j\|_2^2}{4\|\theta_0\|_2 \left(\|\theta_0\|_2 + \frac{1}{2}\|H\|_{\mathsf{F}}\right)},$$

and the result follows by noting that $\sum_{j=1}^p \|Hu_j\|_2^2 = \|H\|_{\mathsf{F}}^2$. ∎

Set

$$\mathcal{F}(\tau, \theta_1, \theta_2) = g_{1,\tau}(\theta_1) + \lambda_{1,\tau} p(\theta) + g_{2,\tau}(\theta_2) + \lambda_{2,\tau} p(\theta_2),$$

$\underline{\mathcal{F}} = \mathcal{F}(\hat{\tau}, \hat{\theta}_{1,\hat{\tau}}, \hat{\theta}_{1,\hat{\tau}})$ the value of Problem (3), and $\mathcal{F}_k = \mathcal{F}(\tau^{(k)}, \theta_1^{(k)}, \theta_2^{(k)}) - \underline{\mathcal{F}}$.

Lemma 13 *Suppose that $\gamma \in (0, b_1^2 \wedge b_2^2]$, and for $j = 1, 2$, $\theta_j^{(0)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$. Then $\lim_k \left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}} = 0$, $\lim_k \left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}} = 0$. Furthermore the sequence $\{\mathcal{F}_k\}$ is non-increasing, and $\lim_k \mathcal{F}_k$ exists.*

Proof We know from Lemma 2 that for $\gamma \in (0, \mathsf{b}_1^2 \wedge \mathsf{b}_2^2]$, and $\theta_j^{(0)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$, we have $\theta_j^{(k)} \in \mathcal{M}_p^+(\mathsf{b}_j, \mathsf{B}_j)$ for all $k \geq 0$, for $j = 1, 2$. We have,

$$\mathcal{F}_{k+1} - \mathcal{F}_k = \mathcal{F}(\tau^{(k+1)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) + \mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \theta_1^{(k)}, \theta_2^{(k)}).$$

By definition, $\mathcal{F}(\tau^{(k+1)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) \leq 0$, and by Lemma 12-Part(2),

$$\mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \theta_1^{(k)}, \theta_2^{(k)}) \leq -\frac{1}{2\gamma}\left\|\theta_1^{(k+1)} - \theta_1^{(k)}\right\|_{\mathsf{F}}^2 - \frac{1}{2\gamma}\left\|\theta_2^{(k+1)} - \theta_2^{(k)}\right\|_{\mathsf{F}}^2$$

It follows that

$$\mathcal{F}_{k+1} \leq \mathcal{F}_k - \frac{1}{2\gamma}\left\|\theta_1^{(k+1)} - \theta_1^{(k)}\right\|_{\mathsf{F}}^2 - \frac{1}{2\gamma}\left\|\theta_2^{(k+1)} - \theta_2^{(k)}\right\|_{\mathsf{F}}^2,$$

which implies that

$$\lim_k \left\|\theta_1^{(k+1)} - \theta_1^{(k)}\right\|_{\mathsf{F}} = 0, \quad \text{and} \quad \lim_k \left\|\theta_2^{(k+1)} - \theta_2^{(k)}\right\|_{\mathsf{F}} = 0. \tag{15}$$

It also implies that the sequence $\{\mathcal{F}_k\}$ is non-increasing and bounded from below by 0. Hence converges. Another application of Lemma 12 gives

$$2\gamma\left(\mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \hat{\theta}_{1,\tau^{(k)}}, \hat{\theta}_{2,\tau^{(k)}})\right) + \left\|\theta_1^{(k+1)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}}^2 + \left\|\theta_2^{(k+1)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\mathsf{B}_1^2}\right)\left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}}^2 + \left(1 - \frac{\gamma}{\mathsf{B}_2^2}\right)\left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}}^2.$$

And notice that $\mathcal{F}(\tau^{(k)}, \theta_1^{(k+1)}, \theta_2^{(k+1)}) - \mathcal{F}(\tau^{(k)}, \hat{\theta}_{1,\tau^{(k)}}, \hat{\theta}_{2,\tau^{(k)}}) \geq 0$. Hence

$$\left\|\theta_1^{(k+1)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}}^2 + \left\|\theta_2^{(k+1)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}}^2 \leq \left(1 - \frac{\gamma}{\mathsf{B}_1^2}\right)\left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}}^2 + \left(1 - \frac{\gamma}{\mathsf{B}_2^2}\right)\left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}}^2,$$

which can be written as

$$\frac{\gamma}{\mathsf{B}_1^2}\left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}}^2 + \frac{\gamma}{\mathsf{B}_2^2}\left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}}^2 \leq \left\|\theta_1^{(k+1)} - \theta_1^{(k)}\right\|_{\mathsf{F}}^2 + \left\|\theta_2^{(k+1)} - \theta_2^{(k)}\right\|_{\mathsf{F}}^2 - 2\left\langle\theta_1^{(k+1)} - \theta_1^{(k)}, \theta_1^{(k+1)} - \hat{\theta}_{1,\tau^{(k)}}\right\rangle - 2\left\langle\theta_2^{(k+1)} - \theta_2^{(k)}, \theta_2^{(k+1)} - \hat{\theta}_{2,\tau^{(k)}}\right\rangle.$$

Since $\{\theta_1^{(k)}\}$, $\{\theta_2^{(k)}\}$ $\{\hat{\theta}_{1,\tau^{(k)}}\}$, and $\{\hat{\theta}_{2,\tau^{(k)}}\}$ are bounded sequence, and given (15), letting $k \to \infty$, we conclude that

$$\lim_k \left\|\theta_1^{(k)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}} = 0, \quad \text{and} \quad \lim_k \left\|\theta_2^{(k)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}} = 0.$$

∎

Proof of Theorem 5 Let $\epsilon > 0$ as in H1. By Lemma 13, there exist $k_0 \geq 1$ such that for all $k \geq k_0$, $\left\|\theta_1^{(k+1)} - \hat{\theta}_{1,\tau^{(k)}}\right\|_{\mathsf{F}} \leq \epsilon$, and $\left\|\theta_2^{(k+1)} - \hat{\theta}_{2,\tau^{(k)}}\right\|_{\mathsf{F}} \leq \epsilon$. Since

$$\tau^{(k+1)} = \mathsf{Argmin}_{t \in \mathcal{T}} \mathcal{H}\left(t | \theta_1^{(k+1)}, \theta_2^{(k+1)}\right),$$

using H1 we conclude that for all $k \geq k_0$,

$$\left|\tau^{(k+1)} - \tau_\star\right| \leq \kappa\left|\tau^{(k)} - \tau_\star\right| + c \leq \kappa^{k-k_0+1}\left|\tau^{(k_0)} - \tau_\star\right| + \frac{c}{1-\kappa},$$

which implies the stated result. ∎

4.2 Proof of Theorem 9

We introduce some more notation. Given $M \in \mathbb{R}^{p\times p}$ the sparsity structure of M is the matrix $\delta \in \{0,1\}^{p\times p}$ such that $\delta_{jk} = \mathbf{1}_{\{|M_{jk}|>0\}}$. In particular we will write $\delta_{\star,j}$ $(j = 1, 2)$ to denote the sparsity structure of $\theta_{\star,j}$. Given matrices $A \in \mathbb{R}^{p\times p}$, and $\delta \in \{0,1\}^{p\times p}$, we will use the notation A_δ (resp. A_{δ^c}) to denote the component-wise product of A and δ (resp A and $1-\delta$). Given $j \in \{1, 2\}$, we define

$$\mathcal{C}_j \stackrel{\text{def}}{=} \left\{M \in \mathcal{M}_p : \ \|M_{\delta_{\star,j}^c}\|_1 \leq 7\|M_{\delta_{\star,j}}\|_1.\right\}. \tag{16}$$

We will need the following deviation bound.

Lemma 14 *Suppose that $X_i \overset{ind}{\sim} \boldsymbol{N}(0, \theta_i^{-1})$, $i = 1, \ldots, N$, where $\theta_i \in \mathcal{M}_p^+$. We set $\Sigma_i \stackrel{\text{def}}{=} \theta_i^{-1}$, and define*

$$\underline{\kappa}_i(2) \stackrel{\text{def}}{=} \inf\left\{u'\Sigma_i u, \ \|u\|_2 = 1, \ \|u\|_0 \leq 2\right\}, \quad \bar{\kappa}_i(2) \stackrel{\text{def}}{=} \sup\left\{u'\Sigma_i u, \ \|u\|_2 = 1, \ \|u\|_0 \leq 2\right\},$$

and suppose that $\underline{\kappa}_i(2) > 0$ *for* $i = 1, \ldots, N$. *Set* $G_N \stackrel{\text{def}}{=} N^{-1}\sum_{i=1}^N (X_i X_i' - \theta_i^{-1})$. *Then for* $0 < \delta \leq 2\left(\frac{\min_k \underline{\kappa}_k(2)}{\max_k \bar{\kappa}_k(2)}\right)^2$, *we have*

$$\mathbb{P}\left(\|G_N\|_\infty > \left(\max_k \bar{\kappa}_k(2)\right)\delta\right) \leq 4p^2 e^{-\frac{N\delta^2}{4}}.$$

Proof The proof is similar to the proof of Lemma 1 of Ravikumar et al. (2010), which itself builds on Bickel and Levina (2008). For $1 \leq i, j \leq p$, arbitrary, set $Z_{ij}^{(k)} = X_{k,i}X_{k,j}$, and $\sigma_{ij}^{(k)} = \Sigma_{k,ij}$, so that the (i,j)-th component of G_N is $N^{-1}\sum_{k=1}^N (Z_{ij}^{(k)} - \sigma_{ij}^{(k)})$. Suppose that $i \neq j$. The case $i = j$ is simpler. It is easy to check that

$$\sum_{k=1}^N \left[Z_{ij}^{(k)} - \sigma_{ij}^{(k)}\right] = \frac{1}{4}\sum_{k=1}^N \left[(X_{k,i} + X_{k,j})^2 - \sigma_{ii}^{(k)} - \sigma_{jj}^{(k)} - 2\sigma_{ij}^{(k)}\right] - \frac{1}{4}\sum_{k=1}^N \left[(X_{k,i} - X_{k,j})^2 - \sigma_{ii}^{(k)} - \sigma_{jj}^{(k)} + 2\sigma_{ij}^{(k)}\right].$$

Notice that $X_{k,i}+X_{k,j} \sim \mathbf{N}(0, \sigma_{ii}^{(k)}+\sigma_{jj}^{(k)}+2\sigma_{ij}^{(k)})$, and $X_{k,i}-X_{k,j} \sim \mathbf{N}(0, \sigma_{ii}^{(k)}+\sigma_{jj}^{(k)}-2\sigma_{ij}^{(k)})$. It follows that for all $x \geq 0$,

$$\mathbb{P}\left[\left|\sum_{k=1}^N \left[Z_{ij}^{(k)} - \sigma_{ij}^{(k)}\right]\right| > x\right] \leq \mathbb{P}\left[\left|\sum_{k=1}^N a_{ij}^{(k)}(W_k - 1)\right| > 2x\right] + \mathbb{P}\left[\left|\sum_{k=1}^N b_{ij}^{(k)}(W_k - 1)\right| > 2x\right],$$

where $W_{1:N} \stackrel{i.i.d.}{\sim} \chi_1^2$, $a_{ij}^{(k)} = \sigma_{ii}^{(k)} + \sigma_{jj}^{(k)} + 2\sigma_{ij}^{(k)}$, and $b_{ij}^{(k)} = \sigma_{ii}^{(k)} + \sigma_{jj}^{(k)} - 2\sigma_{ij}^{(k)}$. For any $x \geq 0$ and a sequence $a = (a_1, \ldots, a_N)$ of positive numbers, with $|a|_\infty = \max_i |a_i|$, $|a|_2 = \sqrt{\sum_i a_i^2}$, we write

$$2x = 2|a|_2\left(\frac{x}{2|a|_2}\right) + 2|a|_\infty\left(\frac{4|a|_2^2}{2x|a|_\infty}\right)\left(\frac{x}{2|a|_2}\right)^2.$$

Therefore if $2x|a|_\infty \leq 4|a|_2^2$, we can apply Lemma 1 of Laurent and Massart (2000) to conclude that

$$\mathbb{P}\left(\left|\sum_{k=1}^N a_k(W_k - 1)\right| \geq 2x\right) \leq 2e^{-\frac{x^2}{4|a|_2^2}}.$$

In particular, we can apply the above bound with $x = |a|_\infty N\delta$ for $\delta \in (0, \frac{2\min_j a_i^2}{\max_i a_i^2}]$ to get that

$$\mathbb{P}\left(\left|\sum_{k=1}^N a_k(W_k - 1)\right| \geq 2|a|_\infty N\delta\right) \leq 2e^{-\frac{N\delta^2}{4}}.$$

In the particular case above, $a_{ij}^{(k)} = \sigma_{ii}^{(k)} + \sigma_{jj}^{(k)} + 2\sigma_{ij}^{(k)} = u'\Sigma^{(k)}u$, where $u_i = u_j = 1$, and $u_r = 0$ for $r \notin \{i, j\}$. And

$$\frac{\min_k u'\Sigma^{(k)}u}{\max_k u'\Sigma^{(k)}u} \geq \frac{\min_k \underline{\kappa}_k(2)}{\max_k \bar{\kappa}(2)}.$$

A similar bound holds for $b_{ij}^{(k)}$. The lemma follows from a standard union-sum argument.

■

The following event plays an important role in the analysis.

$$\mathcal{E}_n \stackrel{\text{def}}{=} \bigcap_{\tau\in\mathcal{T}} \left\{\frac{1}{\lambda_{1,\tau}}\|\nabla g_{1,\tau}(\theta_{\star,1})\|_\infty \leq \frac{\alpha}{2}, \text{ and } \frac{1}{\lambda_{2,\tau}}\|\nabla g_{2,\tau}(\theta_{\star,2})\|_\infty \leq \frac{\alpha}{2}\right\}, \tag{17}$$

Lemma 15 *Under the assumptions of the theorem*

$$\mathbb{P}(\mathcal{E}_n) \geq 1 - \frac{8}{pT}.$$

Proof We have

$$\mathbb{P}(\mathcal{E}_n^c) \leq \mathbb{P}\left(\max_{\tau\in\mathcal{T}} \frac{1}{\lambda_{1,\tau}}\|\nabla g_{1,\tau}(\theta_{\star,1})\|_\infty > \frac{\alpha}{2}\right) + \mathbb{P}\left(\max_{\tau\in\mathcal{T}} \frac{1}{\lambda_{2,\tau}}\|\nabla g_{2,\tau}(\theta_{\star,2})\|_\infty > \frac{\alpha}{2}\right).$$

We show how to bound the first term. A similar bound follows for $g_{2,\tau}$ by working on the reversed sequence $X^{(T)}, \ldots, X^{(1)}$. We have $\nabla g_{1,\tau}(\theta) = \frac{\tau}{2T}(S_1(\tau) - \theta^{-1})$. Setting $U^{(t)} \stackrel{\text{def}}{=} X^{(t)}(X^{(t)})' - \mathbb{E}\left(X^{(t)}(X^{(t)})'\right)$, we can write

$$\nabla g_{1,\tau}(\theta_{\star,1}) = \frac{1}{2T}\sum_{t=1}^\tau U^{(t)} + \frac{(\tau - \tau_\star)_+}{2T}(\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}),$$

where $a_+ \stackrel{\text{def}}{=} \max(a, 0)$. Hence by a standard union-bound argument,

$$\mathbb{P}\left(\max_{\tau\in\mathcal{T}} \frac{1}{\lambda_{1,\tau}}\|\nabla g_{1,\tau}(\theta_{\star,1})\|_\infty > \frac{\alpha}{2}\right) \leq \sum_{\tau\in\mathcal{T}} \mathbb{P}\left(\left\|\sum_{t=1}^\tau U^{(t)}\right\|_\infty > \alpha\lambda_{1,\tau}T - (\tau - \tau_\star)_+\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty\right).$$

Given the choice of $\lambda_{1,\tau}$ in (8), $\alpha\lambda_{1,\tau}T/2 = 2\sqrt{3}\bar{\kappa}\sqrt{\tau\log(pT)} \geq (\tau - \tau_\star)_+\|\theta_{\star,2}^{-1} - \theta_{\star,1}^{-1}\|_\infty$, by assumption (11). In view of (10) we can apply Lemma 14 to deduce that

$$\begin{aligned}\mathbb{P}\left(\max_{\tau\in\mathcal{T}} \frac{1}{\lambda_{1,\tau}}\|\nabla g_{1,\tau}(\theta_{\star,1})\|_\infty > \frac{\alpha}{2}\right) &\leq \sum_{\tau\in\mathcal{T}} \mathbb{P}\left(\left\|\frac{1}{\tau}\sum_{t=1}^\tau U^{(t)}\right\|_\infty > \frac{\alpha\lambda_{1,\tau}T}{2\tau}\right)\\ &\leq 4Tp^2 e^{-\frac{\tau}{4}\left(\frac{\alpha\lambda_{1,\tau}T}{2\tau\bar{\kappa}}\right)^2}\\ &\leq 4\exp\left(2\log(pT) - 3\log(pT)\right) \leq \frac{4}{pT}.\end{aligned}$$

■

Lemma 16 *Under the assumptions of the theorem, and on the event* $\mathcal{E}_n$, *we have*

$$\left\|\hat{\theta}_{1,\tau} - \theta_{\star,1}\right\|_F \leq A\bar{\kappa}\|\theta_{\star,1}\|_2^2\sqrt{\frac{s_1\log(pT)}{\tau}},$$

and

$$\left\|\hat{\theta}_{2,\tau} - \theta_{\star,2}\right\|_F \leq A\bar{\kappa}\|\theta_{\star,2}\|_2^2\sqrt{\frac{s_2\log(pT)}{T - \tau}},$$

for all $\tau \in \mathcal{T}$, *where* A *is an absolute constant that can be taken as* $A = 16 \times 20 \times \sqrt{48}$.

Proof Fix $j \in \{1,2\}$, and $\tau \in \mathcal{T}$. Set $\bar{g}_{j,\tau}(\theta) \stackrel{\text{def}}{=} g_{j,\tau}(\theta) + (1-\alpha)\lambda_{j,\tau} \|\theta\|_{\mathsf{F}} /2$, and recall that $\phi_{j,\tau}(\theta) \stackrel{\text{def}}{=} g_{j,\tau}(\theta) + \lambda_{j,\tau}\wp(\theta)$. Hence $\phi_{j,\tau}(\theta) = \bar{g}_{j,\tau}(\theta) + \alpha\lambda_{j,\tau}\|\theta\|_1$. By a very standard argument that can be found for instance in Negahban et al. (2012), it is known that on the event $\mathcal{E}_n$, and if α satisfies (9) then we have $\hat{\theta}_{j,\tau} - \theta_{\star,j} \in \mathcal{C}_j$, where the cones $\mathcal{C}_j$ are as defined in (16). We write

$$\begin{aligned}\phi_{j,\tau}(\hat{\theta}_{j,\tau}) - \phi_{j,\tau}(\theta_{\star,j}) &= \left\langle \nabla g_{j,\tau}(\theta_{\star,j}) + (1-\alpha)\lambda_{j,\tau}\theta_{\star,j}, \hat{\theta}_{j,\tau} - \theta_{\star,j}\right\rangle \\ &\quad + \bar{g}_{j,\tau}(\hat{\theta}_{j,\tau}) - \bar{g}_{j,\tau}(\theta_{\star,j}) - \left\langle \nabla \bar{g}_{j,\tau}(\theta_{\star,j}), \hat{\theta}_{j,\tau} - \theta_{\star,j}\right\rangle \\ &\quad + \alpha\lambda_{j,\tau}\left(\|\hat{\theta}_{j,\tau}\|_1 - \|\theta_{\star,j}\|_1\right).\end{aligned}$$

On $\mathcal{E}_n$, $\hat{\theta}_{j,\tau} - \theta_{\star,j} \in \mathcal{C}_j$. Therefore

$$\alpha\lambda_{j,\tau}\left|\|\hat{\theta}_{j,\tau}\|_1 - \|\theta_{\star,j}\|_1\right| \le \alpha\lambda_{j,\tau}\left\|\hat{\theta}_{j,\tau} - \theta_{\star,j}\right\|_1 \le 8\alpha\lambda_{j,\tau}\sqrt{s_j}\left\|\hat{\theta}_{j,\tau} - \theta_{\star,j}\right\|_{\mathsf{F}},$$

and

$$\begin{aligned}&\left|\left\langle \nabla g_{j,\tau}(\theta_{\star,j}) + (1-\alpha)\lambda_{j,\tau}\theta_{\star,j}, \hat{\theta}_{j,\tau} - \theta_{\star,j}\right\rangle\right| \\ &\qquad \le \frac{\lambda_{j,\tau}}{2}\left(\alpha + 2(1-\alpha)\|\theta_{\star,j}\|_\infty\right)\left\|\hat{\theta}_{j,\tau} - \theta_{\star,j}\right\|_1 \\ &\qquad\qquad \le 4\lambda_{j,\tau}\left(\alpha + 2(1-\alpha)\|\theta_{\star,j}\|_\infty\right)\sqrt{s_j}\left\|\hat{\theta}_{j,\tau} - \theta_{\star,j}\right\|_{\mathsf{F}}.\end{aligned}$$

Suppose $j = 1$. The case $j = 2$ is similar. We then set $\Delta_{1,\tau} \stackrel{\text{def}}{=} \hat{\theta}_{1,\tau} - \theta_{\star,1}$, and use the second part of Lemma 12 (1) to deduce that

$$\begin{aligned}&\bar{g}_{1,\tau}(\hat{\theta}_{1,\tau}) - \bar{g}_{1,\tau}(\theta_{\star,1}) - \left\langle \nabla \bar{g}_{1,\tau}(\theta_{\star,1}), \hat{\theta}_{1,\tau} - \theta_{\star,1}\right\rangle \\ &\qquad \ge g_{1,\tau}(\hat{\theta}_{1,\tau}) - g_{1,\tau}(\theta_{\star,1}) - \left\langle \nabla g_{1,\tau}(\theta_{\star,1}), \hat{\theta}_{1,\tau} - \theta_{\star,1}\right\rangle \\ &\qquad\qquad \ge \frac{\tau}{2T}\frac{\|\Delta_{1,\tau}\|_{\mathsf{F}}^2}{2\|\theta_{\star,1}\|_2\left(2\|\theta_{\star,1}\|_2 + \|\Delta_{1,\tau}\|_{\mathsf{F}}\right)}.\end{aligned}$$

Set $c_1 = \frac{\tau}{4T\|\theta_{\star,1}\|_2^2}$, $c_2 = 4\lambda_{1,\tau}\sqrt{s_1}\left(3\alpha + 2(1-\alpha)\|\theta_{\star,1}\|_\infty\right)$. Since $\phi_{1,\tau}(\hat{\theta}_{1,\tau}) - \phi_{1,\tau}(\theta_{\star,1}) \le 0$, the above derivation shows that on the event $\mathcal{E}_n$,

$$\frac{c_1\|\Delta_{1,\tau}\|_{\mathsf{F}}^2}{2 + \frac{1}{\|\theta_{\star,1}\|_2}\|\Delta_{1,\tau}\|_{\mathsf{F}}} - c_2\|\Delta_{1,\tau}\|_{\mathsf{F}} \le 0,$$

Under the assumption that $c_1 \ge 2c_2/\|\theta_{\star,1}\|_2$ (which we impose in (10)), this implies that

$$\|\Delta_{1,\tau}\|_{\mathsf{F}} \le \frac{4c_2}{c_1} \le A\bar{\kappa}\|\theta_{\star,1}\|_2^2\sqrt{\frac{s_1\log(pT)}{\tau}},$$

where $A = 16 \times 20 \times \sqrt{48}$, as claimed. ∎

Proof of Theorem 9 For $\tau \in \mathcal{T}$, let

$$r_{1,\tau} \stackrel{\text{def}}{=} A\bar{\kappa}\|\theta_{\star,1}\|_2^2\sqrt{\frac{s_1\log(pT)}{\tau}}, \quad r_{2,\tau} \stackrel{\text{def}}{=} A\bar{\kappa}\|\theta_{\star,2}\|_2^2\sqrt{\frac{s_2\log(pT)}{T-\tau}},$$

be the convergence rates obtained in Lemma 16. Let $\epsilon > 0$ be given by

$$\epsilon \stackrel{\text{def}}{=} \min_{\tau\in\mathcal{T}}(r_{1,\tau} \wedge r_{1,\tau}).$$

For $j = 1, 2$, let $\theta_j \in \mathcal{M}_p^+$ be such that $\|\theta_j - \hat{\theta}_{\tau,j}\|_1 \le \epsilon$. Set $\check{\tau} = \mathsf{Argmin}_{t\in\mathcal{T}}\ \mathcal{H}(t|\theta_1,\theta_2)$, where $\mathcal{H}$ is as defined in (4). Set

$$C_0 = \min\left[\frac{\|\theta_{\star,2} - \theta_{\star,1}\|_{\mathsf{F}}^4}{128B^4\|\theta_{\star,2} - \theta_{\star,1}\|_1^2}, \left(\frac{\bar{\kappa}}{\bar{\kappa}}\right)^4\right].$$

We will show below that

$$\mathbb{P}\left(|\check{\tau} - \tau_\star| > \frac{4\log(p)}{C_0}\right) \le \frac{8}{pT} + \frac{4}{p^2\left(1 - e^{-C_0}\right)}. \tag{18}$$

This implies that with probability at least $1 - \frac{8}{pT} - \frac{4}{p^2(1-e^{-C_0})}$, Assumption H1 holds (with $\epsilon \leftarrow \epsilon/\sqrt{p}$, $\kappa = 0$, and $c = (4/C_0)\log(p)$). The theorem then follows by applying Theorem 5.

Given $\theta_j \in \mathcal{M}_p^+$ be such that $\|\theta_j - \hat{\theta}_{\tau,j}\|_1 \le \epsilon$, we will now show that (18) holds. We shall bound $\mathbb{P}(\check{\tau} > \tau_\star + \delta)$, $\delta = (4/C_0)\log(p)$. The bound on $\mathbb{P}(\check{\tau} < \tau_\star - \delta)$ follows similarly by working with the reversed sequence $X^{(T)}, \ldots, X^{(1)}$.

Note that θ_j can be written as

$$\theta_j = (\theta_j - \hat{\theta}_{\tau,j}) + (\hat{\theta}_{\tau,j} - \theta_{\star,j}) + \theta_{\star,j}. \tag{19}$$

This implies that on $\mathcal{E}_n$, for $\epsilon \le r_{j,\tau}$, and $r_{j,\tau} \le \min\left(\frac{\lambda_{\min}(\theta_{\star,j})}{4}, \frac{\|\theta_{\star,j}\|_\infty}{2}, \frac{\|\theta_{\star,j}\|_1}{1+8s_j^{1/2}}\right)$, we have

$$\begin{aligned}\lambda_{\min}(\theta_j) \ge \frac{1}{2}\lambda_{\min}(\theta_{\star,j}), \quad \lambda_{\max}(\theta_j) \le 2\lambda_{\max}(\theta_{\star,j}),& \\ \|\theta_j\|_\infty \le 2\|\theta_{\star,j}\|_\infty, \quad \text{and} \quad \|\theta_j\|_1 \le 2\|\theta_{\star,j}\|_1.&\end{aligned} \tag{20}$$

Using the event $\mathcal{E}_n$ introduced in (17), we have

$$\begin{aligned}\mathbb{P}\left(\check{\tau} > \tau_\star + \delta\right) &\le \mathbb{P}(\mathcal{E}_n^c) + \sum_{j\ge 0:\ \tau_\star+\delta+j\in\mathcal{T}} \mathbb{P}\left(\mathcal{E}_n,\ \check{\tau} = \tau_\star + \delta + j\right) \\ &\le \mathbb{P}(\mathcal{E}_n^c) + \sum_{j\ge 0:\ \tau_\star+\delta+j\in\mathcal{T}} \mathbb{P}\left(\mathcal{E}_n,\ \phi_{1,\tau_\star+\delta+j}(\theta_1) + \phi_{2,\tau_\star+\delta+j}(\theta_2) \le \phi_{1,\tau_\star}(\theta_1) + \phi_{2,\tau_\star}(\theta_2)\right),\end{aligned} \tag{21}$$

where $\phi_{j,\tau}(\theta) \stackrel{\text{def}}{=} g_{j,\tau}(\theta) + \lambda_{j,\tau}\wp(\theta)$. First we are going to bound the probability

$$\mathbb{P}\left(\mathcal{E}_n,\ \phi_{1,\tau}(\theta_1) + \phi_{2,\tau}(\theta_2) \le \phi_{1,\tau_\star}(\theta_1) + \phi_{2,\tau_\star}(\theta_2)\right),$$

for some arbitrary $\tau \in \mathcal{T}$, $\tau > \tau_\star$. A simple calculation shows that

$$\begin{aligned}\frac{2T}{\tau - \tau_\star}\left[\phi_{1,\tau}(\theta_1) + \phi_{2,,\tau}(\theta_2) - \phi_{1,\tau_\star}(\theta_1) - \phi_{2,\tau_\star}(\theta_2)\right] &= -\log\det(\theta_1) + \log\det(\theta_2) \\ &\quad + \langle\theta_1 - \theta_2, \theta_{\star,2}^{-1}\rangle + \left\langle \theta_1 - \theta_2, \frac{1}{\tau - \tau_\star}\sum_{t=\tau_\star+1}^{\tau}\left(X^{(t)}X^{(t)\prime} - \theta_{\star,2}^{-1}\right)\right\rangle \\ &\quad + 2T\left(\frac{\lambda_{1,\tau} - \lambda_{1,\tau_\star}}{\tau - \tau_\star}\right)\left(\frac{1-\alpha}{2}\|\theta_1\|_{\mathsf{F}}^2 + \alpha\|\theta_1\|_1\right) \\ &\quad + 2T\left(\frac{\lambda_{2,\tau} - \lambda_{2,\tau_\star}}{\tau - \tau_\star}\right)\left(\frac{1-\alpha}{2}\|\theta_2\|_{\mathsf{F}}^2 + \alpha\|\theta_2\|_1\right).\end{aligned}$$

We have $2T\left(\frac{\lambda_{1,\tau}-\lambda_{1,\tau_\star}}{\tau-\tau_\star}\right)\left(\frac{1-\alpha}{2}\|\theta_1\|_{\mathsf{F}}^2+\alpha\|\theta_1\|_1\right)\geq 0$, and

$$2T\left|\frac{\lambda_{2,\tau}-\lambda_{2,\tau_\star}}{\tau-\tau_\star}\right|\leq\frac{\bar{\kappa}}{\alpha}\sqrt{\frac{48\log(pT)}{T-\tau}}=\frac{c_0 r_{2,\tau}}{\alpha s_2^{1/2}\|\theta_{\star,2}\|_2^2},$$

for some absolute constant c_0. Using the infinity-norm and 1-norm bounds in (20) together with (9), we have

$$\frac{1-\alpha}{2}\|\theta_2\|_{\mathsf{F}}^2+\alpha\|\theta_2\|_1=\alpha\left[\frac{1-\alpha}{2\alpha}\|\theta_2\|_\infty+1\right]\|\theta_2\|_1\leq 4\alpha\|\theta_{\star,2}\|_1,$$

and it follows that

$$2T\left|\frac{\lambda_{2,\tau}-\lambda_{2,\tau_\star}}{\tau-\tau_\star}\right|\left(\frac{1-\alpha}{2}\|\theta_2\|_{\mathsf{F}}^2+\alpha\|\theta_2\|_1\right)\leq C_\tau\stackrel{\text{def}}{=}\left(\frac{4c_0\|\theta_{\star,2}\|_1}{s_2^{1/2}\|\theta_{\star,2}\|_2^2}\right)r_{2,\tau}.$$

Set

$$b\stackrel{\text{def}}{=}\min\left(\lambda_{\mathsf{min}}(\theta_{\star,1}),\lambda_{\mathsf{min}}(\theta_{\star,2})\right),\quad B\stackrel{\text{def}}{=}\max\left(\|\theta_{\star,1}\|_2,\|\theta_{\star,2}\|_2\right).$$

By the strong convexity of $\log\det$ (Lemma 12 Part(1)) we have:

$$-\log\det(\theta_1)+\log\det(\theta_2)+\left\langle\theta_1-\theta_2,\theta_{\star,2}^{-1}\right\rangle\geq\left\langle\theta_{\star,2}^{-1}-\theta_2^{-1},\theta_1-\theta_2\right\rangle+\frac{1}{2B^2}\|\theta_1-\theta_2\|_{\mathsf{F}}^2.$$

Since $\theta_{\star,2}^{-1}-\theta_2^{-1}=\theta_{\star,2}^{-1}(\theta_2-\theta_{\star,2})\theta_2^{-1}$, and using the fact that $\|AB\|_{\mathsf{F}}\leq\|A\|_2\|B\|_{\mathsf{F}}$, we have that on $\mathcal{E}_n$,

$$\left|\left\langle\theta_{\star,2}^{-1}-\theta_2^{-1},\theta_1-\theta_2\right\rangle\right|\leq 2r_{2,\tau}\|\theta_{\star,2}^{-1}\|_2\|\theta_2^{-1}\|_2\|\theta_2-\theta_1\|_{\mathsf{F}}\leq 4r_{2,\tau}\|\theta_{\star,2}^{-1}\|_2^2\|\theta_2-\theta_1\|_{\mathsf{F}}.$$

We conclude that on $\mathcal{E}_n$,

$$\frac{2T}{\tau-\tau_\star}\left[\phi_{1,\tau}(\theta_1)+\phi_{2,\tau}(\theta_2)-\phi_{1,\tau_\star}(\theta_1)-\phi_{2,\tau_\star}(\theta_2)\right]\geq\left\langle\theta_1-\theta_2,\frac{1}{\tau-\tau_\star}\sum_{t=\tau_\star+1}^{\tau}\left(X^{(t)}X^{(t)'}-\theta_{\star,2}^{-1}\right)\right\rangle-C_\tau-4r_{2,\tau}\|\theta_{\star,2}^{-1}\|_2^2\|\theta_2-\theta_1\|_{\mathsf{F}}+\frac{1}{2B^2}\|\theta_1-\theta_2\|_{\mathsf{F}}^2.$$

Under the assumption (12) imposed on $r_{j,\tau}$ and for $\epsilon\leq r_{1,\tau}\wedge r_{2,\tau}$, it can be shown that on $\mathcal{E}_n$, and for $\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}\geq\frac{8c_0\|\theta_{\star,2}\|_1}{s_2^{1/2}\|\theta_{\star,2}\|_2^2\|\theta_{\star,2}^{-1}\|_2^2}$, we have

$$-C_\tau-2\left(\epsilon+r_{2,\tau}\right)\|\theta_{\star,2}^{-1}\|_2^2\|\theta_2-\theta_1\|_{\mathsf{F}}+\frac{1}{4B^2}\|\theta_1-\theta_2\|_{\mathsf{F}}^2\geq 0.\tag{22}$$

To see this, note that (22) holds if $\|\theta_2-\theta_1\|_{\mathsf{F}}\geq 8B^2r_{2,\tau}\|\theta_{\star,2}^{-1}\|_2^2+2B\sqrt{C_\tau+16B^2\|\theta_{\star,2}^{-1}\|_2^4r_{2,\tau}^2}$. Then it can be checked that if $r_{2,\tau}\leq\frac{c_0\|\theta_{\star,2}\|_1}{16B^2s_2^{1/2}\|\theta_{\star,2}\|_2^2\|\theta_{\star,2}^{-1}\|_2^2}$, then

$$8B^2\|\theta_{\star,2}^{-1}\|_2^2r_{2,\tau}\leq\frac{C_\tau}{2\|\theta_{\star,2}^{-1}\|_2^2r_{2,\tau}},\quad\text{and}\quad 4B\sqrt{C_\tau}\leq\frac{C_\tau}{2\|\theta_{\star,2}^{-1}\|_2^2r_{2,\tau}}.$$

Therefore, (22) holds if

$$\|\theta_2-\theta_1\|_{\mathsf{F}}\geq\frac{C_\tau}{\|\theta_{\star,2}^{-1}\|_2^2r_{2,\tau}}=\frac{4c_0\|\theta_{\star,2}\|_1}{s_2^{1/2}\|\theta_{\star,2}\|_2^2\|\theta_{\star,2}^{-1}\|_2^2}.$$

Now we write

$$\theta_2-\theta_1=(\theta_2-\hat{\theta}_{\tau,2})+(\hat{\theta}_{\tau,2}-\theta_{\star,2})+(\theta_{\star,2}-\theta_{\star,1})+(\theta_{\star,1}-\hat{\theta}_{\tau,1})+(\hat{\theta}_{\tau,1}-\theta_1),$$

and use the fact that $\epsilon\leq r_{1,\tau}\wedge r_{2,\tau}$, and $r_{j,\tau}\leq\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}/8$ to deduce that on $\mathcal{E}_n$, $\|\theta_2-\theta_1\|_{\mathsf{F}}\geq\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}/2$, and this completes the proof of the claim.

It follows from the above that

$$\mathbb{P}\left(\mathcal{E}_n;\phi_{1,\tau}(\theta_1)+\phi_{2,\tau}(\theta_2)-\phi_{1,\tau_\star}(\theta_1)-\phi_{2,\tau_\star}(\theta_2)\leq 0\right)\leq\mathbb{P}\left(\left\|\frac{1}{\tau-\tau_\star}\sum_{t=\tau_\star+1}^{\tau}\left(X^{(t)}X^{(t)'}-\theta_{\star,2}^{-1}\right)\right\|_\infty>\frac{\|\theta_2-\theta_1\|_{\mathsf{F}}^2}{4B^2\|\theta_2-\theta_1\|_1}\right).\tag{23}$$

Proceeding as above, it is easy to see that if $\epsilon\leq r_{1,\tau}\wedge r_{2,\tau}$, and $r_{j,\tau}\leq\frac{\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}}{2(1+8s^{1/2})}$, then

$$\frac{\|\theta_2-\theta_1\|_{\mathsf{F}}^2}{4B^2\|\theta_2-\theta_1\|_1}\geq\frac{\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}^2}{32B^2\|\theta_{\star,2}-\theta_{\star,1}\|_1}.$$

Using this, and by Lemma 15, it follows that the probability on the right-hand side of (23) is upper-bounded by

$$4p^2\exp\left(-(\tau-\tau_\star)\min\left[\frac{\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}^4}{128B^4\|\theta_{\star,2}-\theta_{\star,1}\|_1^2},\left(\frac{\bar{\kappa}}{\bar{\kappa}}\right)^4\right]\right).$$

We apply this to (21) to get:

$$\mathbb{P}(\check{\tau}>\tau_\star+\delta)\leq\mathbb{P}(\mathcal{E}_n^c)+\sum_{j\geq 0}4p^2e^{-C_0(\delta+j)}\leq\frac{8}{pT}+\frac{4}{p^2(1-e^{-C_0})},$$

where $C_0=\min\left[\frac{\|\theta_{\star,2}-\theta_{\star,1}\|_{\mathsf{F}}^4}{128B^4\|\theta_{\star,2}-\theta_{\star,1}\|_1^2},\left(\frac{\kappa}{\bar{\kappa}}\right)^4\right]$, and by taking $\delta=4\log(p)/C_0$. This completes the proof. ■

Acknowledgments

This work is partially supported by the NSF grant DMS 1513040

References

Y. F. Atchadé, R. Mazumder, and J. Chen. Scalable Computation of Regularized Precision Matrices via Stochastic Optimization. *ArXiv e-prints*, September 2015.

Y. F. Atchade, G. Fort, and E. Moulines. On stochastic proximal gradient algorithms. *Journal of Machine Learning Research*, 18:1–33, 2017.

Alexander Aue, Siegfried Hormann, Lajos Horváth, and Matthew Reimherr. Break detection in the covariance structure of multivariate time series models. *Ann. Statist.*, 37(6B):4046–4087, 12 2009.

Jushan Bai. Estimation of a change point in multiple regression models. *The Review of Economics and Statistics*, 79(4):551–563, 1997.

Anindya Banerjee and Giovanni Urga. Modelling structural breaks, long memory and stock market volatility: an overview. *Journal of Econometrics*, 129(1):1–34, 2005.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008.

Andrea Beltratti and Claudio Morana. Breaks and persistency: macroeconomic causes of stock market volatility. *Journal of econometrics*, 131(1):151–177, 2006.

Peter J. Bickel and Elizaveta Levina. Covariance regularization by thresholding. *Ann. Statist.*, 36 (6):2577–2604, 2008.

Leland Bybee. *changepointsHD: Change-Point Estimation for Expensive and High-Dimensional Models*, 2017. R package version 0.3.0.

Hao Chen and Nancy Zhang. Graph-based change-point detection. *Ann. Statist.*, 43(1):139–176, 02 2015.

Haeran Cho and Piotr Fryzlewicz. Multiple-change-point detection for high dimensional time series via sparsified binary segmentation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 77(2):475–507, 2015.

Kyongwook Choi, Wei-Choun Yu, and Eric Zivot. Long memory versus structural breaks in modeling and forecasting realized volatility. *Journal of International Money and Finance*, 29(5):857–875, 2010.

Jerome Friedman, Trevor Hastie, Holger Höfling, and Robert Tibshirani. Pathwise coordinate optimization. *Ann. Appl. Stat.*, 1(2):302–332, 2007.

Piotr Fryzlewicz. Wild binary segmentation for multiple change-point detection. *Ann. Statist.*, 42 (6):2243–2281, 12 2014.

Samet Günay. Long memory property and structural breaks in volatility: Evidence from turkey and brazil. *International Journal of Economics and Finance*, 6(12):119, 2014.

T. Hastie, R Tibshirani, and M. Wainwright. *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman and Hall/CRC, 2015. ISBN 978-1-4987-1216-3.

Holger Höfling and Robert Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *J. Mach. Learn. Res.*, 10:883–906, 2009.

M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimating time-varying networks. *Ann. Appl. Statist.*, 4(1):94–123, 2010.

Mladen Kolar and Eric P. Xing. Estimating networks with jumps. *Electron. J. Statist.*, 6:2069–2106, 2012.

John Lafferty, Han Liu, Larry Wasserman, et al. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Ann. Statist.*, 28(5):1302–1338, 2000.

F. Leonardi and P. Bühlmann. Computationally efficient change point detection for high-dimensional regression. *ArXiv e-prints*, 2016.

Céline Lévy-Leduc and Franois Roueff. Detection and localization of change-points in high-dimensional network traffic data. *Ann. Appl. Stat.*, 3(2):637–662, 06 2009.

Song Liu, John A. Quinn, Michael U. Gutmann, and Masashi Sugiyama. Direct learning of sparse changes in markov networks by density ratio estimation. In Hendrik Blockeel, Kristian Kersting, Siegfried Nijssen, and Filip Železný, editors, *Machine Learning and Knowledge Discovery in Databases*, pages 596–611, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

N. Meinshausen and P. Buhlmann. High-dimensional graphs with the lasso. *Annals of Stat.*, 34: 1436–1462, 2006.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 11 2012. doi: 10.1214/12-STS400.

Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1 (3):123–231, 2013.

Pradeep Ravikumar, Martin J. Wainwright, and John D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *Ann. Statist.*, 38(3):1287–1319, 2010.

Pradeep Ravikumar, Martin J. Wainwright, Garvesh Raskutti, and Bin Yu. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electron. J. Stat.*, 5: 935–980, 2011.

Benjamin Rolfs, Bala Rajaratnam, Dominique Guillot, Ian Wong, and Arian Maleki. Iterative thresholding algorithm for sparse inverse covariance estimation. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1574–1582. Curran Associates, Inc., 2012.

Sandipan Roy, Yves Atchadé, and George Michailidis. Change point estimation in high dimensional markov random-field models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, pages 1187–1206, 2017.

Tong Tong Wu and Kenneth Lange. The mm alternative to em. *Statist. Sci.*, 25(4):492–505, 11 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

S. Zhou, J. Lafferty, and L. Wasserman. Time-varying undirected graphs. *In Rocco A. Dervedio and Tong Zhang, editors, Conference on learning Theory*, pages 455–466, 2009.

Journal of Machine Learning Research 19 (2018) 1-39 Submitted 05/17; Revised 05/18; Published 07/18

Statistical Analysis and Parameter Selection for Mapper

Mathieu Carrière mathieu.carriere@inria.fr
Inria Saclay
91120 Palaiseau, France

Bertrand Michel bertrand.michel@ec-nantes.fr
LMJL UMR 6629– Ecole Centrale Nantes
44322 Nantes, France

Steve Oudot steve.oudot@inria.fr
Inria Saclay
91120 Palaiseau, France

Editor: Kevin Murphy and Bernhard Schölkopf

Abstract

In this article, we study the question of the statistical convergence of the 1-dimensional Mapper to its continuous analogue, the Reeb graph. We show that the Mapper is an optimal estimator of the Reeb graph, which gives, as a byproduct, a method to automatically tune its parameters and compute confidence regions on its topological features, such as its loops and flares. This allows to circumvent the issue of testing a large grid of parameters and keeping the most stable ones in the brute-force setting, which is widely used in visualization, clustering and feature selection with the Mapper.

Keywords: Topological Data Analysis, Mapper, Parameter Selection, Confidence Regions, Extended Persistence

1. Introduction

In statistical learning, a large class of problems can be categorized into supervised or unsupervised problems. For supervised learning problems, an output quantity Y must be predicted or explained from the input measures X. On the contrary, for unsupervised problems there is no output quantity Y to predict and the aim is to explain and model the underlying structure or distribution in the data. In a sense, unsupervised learning can be thought of as extracting features from the data, assuming that the latter come with unstructured noise. Many methods in data sciences can be qualified as unsupervised methods, among the most popular examples are association methods, clustering methods, linear and non linear dimension reduction methods and matrix factorization to cite a few (see for instance Chapter 14 in Friedman et al. (2001)). Topological Data Analysis (TDA) has emerged in the recent years as a new field whose aim is to uncover, understand and exploit the topological and geometric structure underlying complex and possibly high-dimensional data. Most of TDA methods can thus be qualified as unsupervised. In this paper, we study a recent TDA algorithm called Mapper which was first introduced in Singh et al. (2007).

Starting from a point cloud $\mathbb{X}_n$ sampled from a metric space $\mathcal{X}$, the idea of the Mapper is to study the topology of the sublevel sets of a function $f : \mathbb{X}_n \to \mathbb{R}$ defined on the point cloud[1]. The function f is called a *filter function* and it has to be chosen by the user. The Mapper construction depends on the choice of a cover $\mathcal{I}$ of the image of f by open sets. Pulling back $\mathcal{I}$ through f gives an open cover of the domain $\mathbb{X}_n$. It is then refined into a connected cover by splitting each element into its various clusters using a clustering algorithm whose choice is left to the user. Then, the Mapper is defined as the nerve of the connected cover, having one vertex per element, one edge per pair of intersecting elements, and more generally, one k-simplex per non-empty $(k+1)$-fold intersection. It can also be seen as a discrete approximation of its continuous counterpart called the *Reeb graph*, which was originally introduced in Reeb (1946).

In practice, the Mapper has two major applications. The first one is data visualization and clustering. Indeed, when the cover $\mathcal{I}$ is minimal in terms of cardinality, i.e. no more than two cover elements can intersect at once, the Mapper provides a visualization of the data in the form of a graph whose topology reflects that of the data. As such, it brings additional information to the usual clustering algorithms by identifying *flares* and *loops* that outline potentially remarkable subpopulations in the various clusters. See e.g. Yao et al. (2009); Lum et al. (2013); Sarikonda et al. (2014); Hinks et al. (2015) for examples of applications. The second application of Mapper is about feature selection. Indeed, each feature of the data can be evaluated on its ability to discriminate the interesting subpopulations mentioned above (flares, loops) from the rest of the data, using for instance Kolmogorov-Smirnov tests. See e.g. Lum et al. (2013); Nielson et al. (2015); Rucco et al. (2015) for examples of applications.

Unsupervised methods generally depend on parameters that need to be chosen by the user, such as the number of selected dimensions for dimension reduction methods or the number of clusters for clustering methods. Contrarily to supervised problems, it can be very difficult to evaluate the output of unsupervised methods and thus to select parameters. Regarding Mapper, the only answer proposed in the literature consists in selecting parameters in a range of values for which the Mapper seems to be stable—see for instance Nielson et al. (2015). But with non trivial data sets, it is not easy to tune Mapper this way. The problem is illustrated for instance in Figure 1 on a data set that we study further in Section 5. More generally, we believe that such an approach is not satisfactory since it does not provide statistical guarantees on the inferred Mapper. This major drawback of Mapper is an important obstacle to its use in exploratory data analysis.

Contributions. Our main goal in this article is to provide a statistical method to tune the parameters of the Mapper automatically in various settings (Equations (8), (9) and (10)) by computing its rate of convergence (Propositions 11 and 13 and Corollary 14) to its continuous counterpart called the Reeb graph, avoiding the computational cost of testing millions of candidates and selecting the most stable ones in the brute-force setting of many practitioners. We also provide methods to assess stability, rates of convergence and confidence regions (Proposition 15) for the topological features of the Mapper. We believe that this set of methods open the way to an accessible and intuitive utilization of Mapper for non expert researchers in applied topology.

1. The Mapper was originally defined more generally for functions with values in $\mathbb{R}^d$, with arbitrary $d > 0$. In this work, we restrict the focus to scalar-valued functions, since the mathematical analysis is much easier in that case, and since it also corresponds to many use cases of the Mapper.

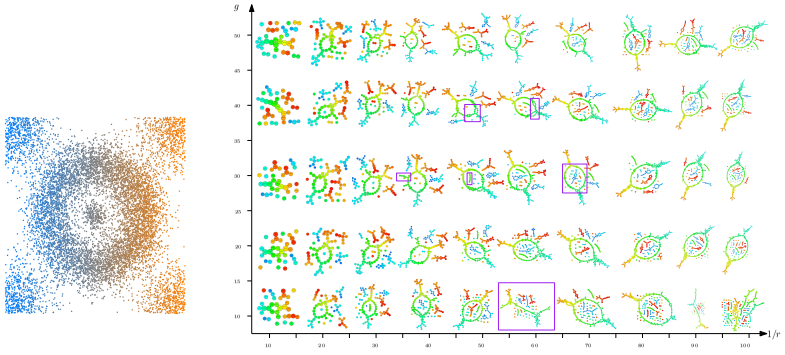


Figure 1: A collection of Mappers computed with various parameters. Left: crater data set. Right: outputs of Mapper with various parameters. One can see that for some Mappers (the ones with purple squares), topological features suddenly appear and disappear. These are discretization artifacts, that we overcome in this article by appropriately tuning the parameters.

Related work. Theoretical properties of Reeb graphs and Mappers have been the topic of several recent articles. Reeb graphs are now well understood and have been used in a wide range of applications. Algorithms for their computation have been proposed, as well as studies of their homology groups, like in Dey et al. (2017), and metrics for their comparison, such as the *functional distortion distance* of Bauer et al. (2014), the *interleaving distance* of de Silva et al. (2016) and the *edit distance* of di Fabio and Landi (2016). We refer the interested reader to the survey Biasotti et al. (2008) and to the introductions of Bauer et al. (2014) and Bauer et al. (2015) for a comprehensive list of references.

Concerning the Mapper, Babu (2013) characterized the Mapper with coarsened levelset zigzag persistence modules and showed that, as the lengths of the intervals in the cover $\mathcal{I}$ go to zero uniformly, the Mapper of a real-valued function converges to the Reeb graph in the bottleneck distance (defined in Section 2.2). Similarly, Munch and Wang (2016) recently characterized the Mapper with constructible cosheaves and showed the same type of convergence for the Mapper in the interleaving distance. Their result holds in the general case of vector-valued functions. However, in both approaches, the quantification of convergence is not precise enough to enable parameter selection.

Statistics in TDA have been so far focused on persistence diagrams, with the computation of rates of convergence, confidence regions and bootstrap—see e.g. Chazal et al. (2013); Fasy et al. (2014); Chazal et al. (2015a,b). In this article, we build on this line of work to provide results in the same vein for the Mapper. The integration of the Mapper in this framework is not straightforward since it encodes a different type of information than persistence diagrams. However, this is made possible by the recent work (in a deterministic setting) of Carrière and Oudot (2017b) about the structure and the stability of the Mapper. In this article, the authors provide a way to go from the input space to the Mapper using small perturbations. We build on this precise relation between the input space and its Mapper to show that the Mapper is itself a measurable construction. In Carrière and Oudot (2017b), the authors also show that the topological structure of the Mapper can actually be predicted from the cover $\mathcal{I}$ by looking at appropriate *signatures* that take the form of *extended persistence diagrams*. In this article, we use this observation, together with an approximation inequality, to show that the Mapper, computed with a specific set of parameters, is actually an optimal estimator of its continuous analogue, the so-called *Reeb graph*. Moreover, these specific parameters act as natural candidates to obtain a reliable Mapper with no artifacts.

Plan of the article. Section 2 presents the necessary background on the Reeb graph and the Mapper, and it also gives an approximation inequality—Theorem 7—for the Reeb graph with the Mapper. From this approximation result, we derive rates of convergences as well as candidate parameters in Section 3, and we show how to get confidence regions in Section 4. Section 5 illustrates the validity of our parameter tuning and confidence regions with numerical experiments on smooth and noisy data.

2. Approximation of a Reeb graph with the Mapper

2.1 Background on the Reeb graph and the Mapper

We start with some background on the Reeb graph and the Mapper. In particular, we present the specific Mapper algorithm that we study in this article.

Reeb graph. Let $\mathcal{X}$ be a topological space and let $f : \mathcal{X} \to \mathbb{R}$ be a continuous function. Such a function on $\mathcal{X}$ is called a *filter function* in the following. Then, we define the equivalence relation $\sim_f$ as follows: for all x and x' in $\mathcal{X}$, x and x' are in the same class ($x \sim_f x'$) if and only if x and x' belong to the same connected component of $f^{-1}(y)$, for some y in the image of f.

Definition 1 *The* Reeb graph $\mathrm{R}_f(\mathcal{X})$ *of* $\mathcal{X}$ *computed with the filter function* f *is the quotient space* $\mathcal{X}/\sim_f$ *endowed with the quotient topology.*

See Figure 2 for an illustration. Note that, since f is constant on equivalence classes, there is an induced map $f_{\mathrm{R}} : \mathrm{R}_f(\mathcal{X}) \to \mathbb{R}$ such that $f = f_{\mathrm{R}} \circ \pi$, where π is the quotient map $\mathcal{X} \to \mathrm{R}_f(\mathcal{X})$. The topological structure of a Reeb graph can be described if the pair $(\mathcal{X}, f)$ is regular enough. From now on, we will assume that the filter function $f : \mathcal{X} \to \mathbb{R}$ is *Morse-type*. Morse-type functions are generalizations of classical Morse functions that share some of their properties without having to be differentiable (nor even defined over a smooth manifold).

Definition 2 *Let* f *be a continuous real-valued function defined on a compact space* $\mathcal{X}$. *Then* f *is called* of Morse type *if:*

(i) *There is a finite set* $\mathrm{Crit}(f) = \{a_1 < ... < a_n\}$, *called the set of* critical values, *such that over every open interval* $(a_0 = -\infty, a_1), ..., (a_i, a_{i+1}), ..., (a_n, a_{n+1} = +\infty)$ *there is a compact and locally connected space* $\mathcal{Y}_i$ *and a homeomorphism* $\mu_i : \mathcal{Y}_i \times$

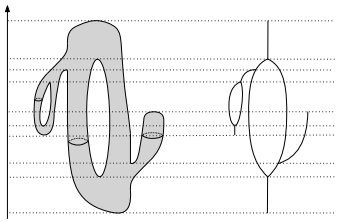


Figure 2: Example of Reeb graph computed on a double torus with the height function. Connected components of the level sets of the function (such as the three different ones drawn on the double torus) are contracted into single points.

$(a_i, a_{i+1}) \to f^{-1}((a_i, a_{i+1}))$ *such that* $\forall i = 0, ..., n, f|_{f^{-1}((a_i,a_{i+1}))} = \pi_2 \circ \mu_i^{-1}$, *where* π_2 *is the projection onto the second factor;*

(ii) $\forall i = 1, ..., n-1, \mu_i$ *extends to a continuous function* $\bar{\mu}_i : \mathcal{Y}_i \times [a_i, a_{i+1}] \to f^{-1}([a_i, a_{i+1}])$ *and similarly* μ_0 *extends to* $\bar{\mu}_0 : \mathcal{Y}_0 \times (-\infty, a_1] \to f^{-1}((-\infty, a_1])$ *and* μ_n *extends to* $\bar{\mu}_n : \mathcal{Y}_n \times [a_n, +\infty) \to f^{-1}([a_n, +\infty))$*;*

(iii) *Each levelset* $f^{-1}(t)$ *has a finitely-generated homology.*

Key fact 1a. (Proposition 2.10 in de Silva et al. (2016)) For $f : \mathcal{X} \to \mathbb{R}$ a Morse-type function, the Reeb graph $\mathrm{R}_f(\mathcal{X})$ is a multigraph.

For our purposes, in the following we further assume that $\mathcal{X}$ is a smooth and compact submanifold of $\mathbb{R}^D$. The set of Reeb graphs computed with Morse-type functions over such spaces is denoted $\mathcal{R}$ in this article. Whenever it is necessary, it will be equipped with extra structures in the following, such as pseudometrics or topologies.

Mapper. The Mapper is introduced in Singh et al. (2007) as a statistical version of the Reeb graph $\mathrm{R}_f(\mathcal{X})$ in the sense that it is a discrete and computable approximation of the Reeb graph computed with some filter function. Assume that we observe a point cloud $\mathbb{X}_n = \{X_1, \ldots, X_n\} \subset \mathcal{X}$ with known pairwise distances. A filter function is chosen and can be computed on each point of $\mathbb{X}_n$. The generic version of the Mapper algorithm on $\mathbb{X}_n$ computed with the filter function f can be summarized as follows:

1. Cover the range of values $\mathbb{Y}_n = f(\mathbb{X}_n)$ with a set of consecutive intervals $\{I_s\}_{1 \le s \le S}$ which overlap.

2. Apply a clustering algorithm to each pre-image $f^{-1}(I_s)$, $s \in \{1, ..., S\}$. This defines a *pullback cover* $\mathcal{C} = \{\mathcal{C}_{1,1}, \ldots, \mathcal{C}_{1,k_1}, \ldots, \mathcal{C}_{S,1}, \ldots, \mathcal{C}_{S,k_S}\}$ of the point cloud $\mathbb{X}_n$, where $\mathcal{C}_{s,k}$ denotes the kth cluster of $f^{-1}(I_s)$.

3. The Mapper is then the *nerve* of $\mathcal{C}$. Each vertex $v_{s,k}$ of the Mapper corresponds to one element $\mathcal{C}_{s,k}$ and two vertices $v_{s,k}$ and $v_{s',k'}$ are connected if and only if $\mathcal{C}_{s,k} \cap \mathcal{C}_{s',k'}$ is not empty, i.e. they have common points.

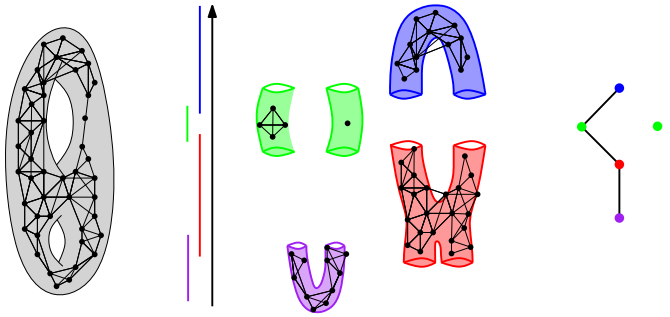


Figure 3: Example of Mapper computed on a sampling of the double torus with the height function f and a cover $\mathcal{I}$ of its range with four open intervals. Clusters are given by a neighborhood graph built on the sampling. Note that the rightmost green vertex is not connected to the other vertices of the Mapper since its corresponding cluster (which contains only one point) has no common points with the others.

See Figure 3 for an illustration. Even for one given filter function, many versions of the Mapper algorithm can be proposed depending on how one chooses the intervals that cover the image of f, and which method is used to cluster the pre-images. Moreover, note that the Mapper can be defined as well for continuous spaces. The definition is strictly the same except for the clustering step, which is replaced by taking the connected components of each pre-image $f^{-1}(I_s)$, $s \in \{1, ..., S\}$.

Our version of Mapper. In this article, we focus on a Mapper algorithm that uses neighborhood graphs. Of course, more sophisticated versions of Mapper can be used in practice but then the statistical analysis is more tricky. We assume that there exists a distance on $\mathbb{X}_n$ and that the matrix of pairwise distances is available. First, from the distance matrix we compute the δ-neighborhood graph built on top of $\mathbb{X}_n$, i.e. we draw an edge between two different points whenever their pairwise distance is less than δ. This object plays the role of an approximation of the underlying and unknown metric space $\mathcal{X}$ on which the data are sampled. Second, given $\mathbb{Y}_n = f(\mathbb{X}_n)$ the set of filter values, we choose a regular cover of $\mathbb{Y}_n$ with open intervals, where no more than two intervals can intersect at a time. More precisely, we use open intervals with same length r (apart from the first and the last one, which can have any positive length): $\forall s \in \{2, \ldots, S-1\}$,

$$r = \ell(I_s) \tag{1}$$

where ℓ is the Lebesgue measure on $\mathbb{R}$. The overlap g between two consecutive intervals is also a fixed constant: $\forall s \in \{1, \ldots, S-1\}$,

$$0 < g = \frac{\ell(I_s \cap I_{s+1})}{r} < \frac{1}{2}. \tag{2}$$

The parameters g and r are generally called the *gain* and the *resolution* in the literature on the Mapper algorithm. Finally, for the clustering step, we simply consider the connected components of the pre-images $f^{-1}(I_s)$ that are induced by the δ-neighborhood graph. The corresponding Mapper is denoted $\mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ or M_n for short in the following. When dealing with a continuous space $\mathcal{X}$, there is no need to compute a neighborhood graph since the connected components are well-defined, so we let $\mathrm{M}_{r,g}(\mathcal{X}, f)$ denote our version of the Mapper in this case.

Key fact 1b. The Mapper $\mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ is a combinatorial graph.

Moreover, following Carrière and Oudot (2017b), we can define a function on the nodes of M_n as follows.

Definition 3 *Let v be a node of M_n, i.e. v represents a connected component of $f^{-1}(I_s)$ for some $s \in \{1, \dots, S\}$. Then, we let*

$$f_{\mathcal{I}}(v) = \mathrm{mid}(\tilde{I}_s),$$

where $\tilde{I}_s = I_s \setminus (I_{s-1} \cup I_{s+1})$ and $\mathrm{mid}(\tilde{I}_s)$ denotes the midpoint of the interval $\tilde{I}_s$.

Filter functions. In practice, it is common to choose filter functions that are coordinate-independent, in order to avoid depending on solid transformations of the data like rotations or translations. The two most common filters that are used in the literature are:

- the *eccentricity*: $x \mapsto \sup_{y \in \mathcal{X}} d(x, y)$,
- the eigenfunctions of the covariance matrix as used in Principal Component Analysis.

2.2 Extended persistence signatures and the persistence metric

In this section, we introduce *extended persistence* and its associated metric, the *bottleneck distance*, which we will use later to compare Reeb graphs and Mappers. We merely provide a short introduction containing the necessary definitions since the statement of our results does not require a deep understanding of these notions. The understanding of the proofs of these results is more demanding, so we refer the reader willing to read proofs and already familiar with homology to Appendix C for more details, and to Edelsbrunner and Harer (2010); Oudot (2015) for a thorough treatment of extended persistence.

Extended persistence. Given any graph $G = (V, E)$ and a function defined on its nodes $f : V \to \mathbb{R}$, the so-called *extended persistence diagram* $\mathrm{Dg}(G, f)$, originally defined in Cohen-Steiner et al. (2009), is a multiset of points in the Euclidean plane $\mathbb{R}^2$ that can be computed with *extended persistence theory*. Each of the diagram points has a specific *type*, which is either Ord_0, Rel_1, Ext_0^+ or Ext_1^-. A rigorous connection between the Mapper and the Reeb graph was drawn recently by Carrière and Oudot (2017b), who show how extended persistence provides a relevant and efficient framework to compare a Reeb graph with a Mapper. We summarize below the main points of this work in the perspective of the present article.

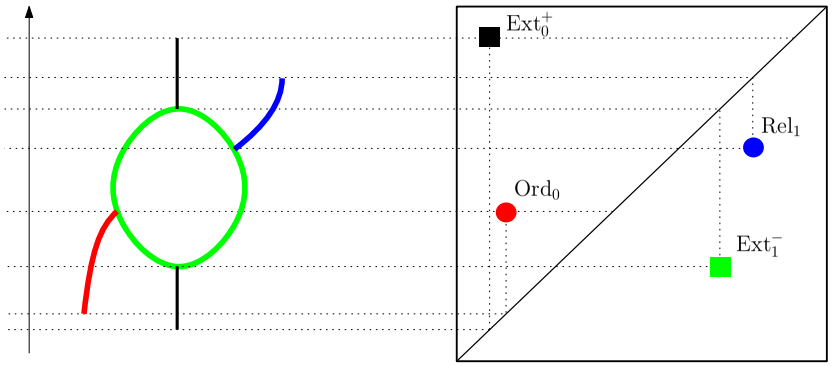


Figure 4: Example of correspondences between topological features of a graph and points in its corresponding extended persistence diagram. Note that ordinary persistence is unable to detect the blue upwards branch.

Topological dictionary. Given a topological space $\mathcal{X}$ and a Morse-type function $f : \mathcal{X} \to \mathbb{R}$, there is a nice interpretation of $\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}})$ in terms of the structure of $\mathrm{R}_f(\mathcal{X})$. Orienting the Reeb graph vertically so f_{R} is the height function, we can see each connected component of the graph as a trunk with multiple branches (some oriented upwards, others oriented downwards) and holes. Then, one has the following correspondences, where the *vertical span* of a feature is the span of its image by f_{R}:

- The vertical spans of the trunks are given by the points in $\mathrm{Ext}_0^+(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}})$;
- The vertical spans of the branches that are oriented downwards are given by the points in $\mathrm{Ord}_0(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}})$;
- The vertical spans of the branches that are oriented upwards are given by the points in $\mathrm{Rel}_1(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}})$;
- The vertical spans of the holes are given by the points in $\mathrm{Ext}_1^-(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}})$.

These correspondences provide a dictionary to read off the structure of the Reeb graph from the corresponding extended persistence diagram. See Figure 4 for an illustration.

Note that it is a bag-of-features type descriptor, taking an inventory of all the features (trunks, branches, holes) together with their vertical spans, but leaving aside the actual layout of the features. As a consequence, it is an incomplete descriptor: two Reeb graphs with the same persistence diagram may not be isomorphic.

Bottleneck distance. We now define the commonly used metric between persistence diagrams.

Definition 4 *Given two persistence diagrams D, D', a* partial matching *between D and D' is a subset Γ of $D \times D'$ such that:*

$$\forall p \in D, \text{ there is at most one } p' \in D' \text{ such that } (p, p') \in \Gamma,$$

$$\forall p' \in D', \text{ there is at most one } p \in D \text{ such that } (p, p') \in \Gamma.$$

Furthermore, Γ must match points of the same type (ordinary, relative, extended) and of the same homological dimension only. Let Δ be the diagonal $\Delta = \{(x, x) : x \in \mathbb{R}\}$. The cost *of Γ is:*

$$\text{cost}(\Gamma) = \max\left\{\max_{p \in D} \delta_D(p), \max_{p' \in D'} \delta_{D'}(p')\right\},$$

where

$$\delta_D(p) = \|p - p'\|_\infty \text{ if } \exists p' \in D' \text{ such that } (p, p') \in \Gamma, \text{ otherwise } \delta_D(p) = \inf_{q \in \Delta} \|p - q\|_\infty,$$

$$\delta_{D'}(p') = \|p - p'\|_\infty \text{ if } \exists p \in D \text{ such that } (p, p') \in \Gamma, \text{ otherwise } \delta_{D'}(p') = \inf_{q \in \Delta} \|p' - q\|_\infty.$$

Definition 5 *Let D, D' be two persistence diagrams. The* bottleneck distance *between D and D' is:*

$$d_\Delta(D, D') = \inf_\Gamma \text{ cost}(\Gamma),$$

where Γ ranges over all partial matchings between D and D'.

Note that d_Δ is only a pseudometric and not a true metric, because diagrams which only differ at the diagonal will have zero distance.

Definition 6 *Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two combinatorial graphs with real-valued functions $f_1 : V_1 \to \mathbb{R}$ and $f_2 : V_2 \to \mathbb{R}$ attached to their nodes. The* persistence metric d_Δ *between the pairs (G_1, f_1) and (G_2, f_2) is:*

$$d_\Delta(G_1, G_2) = d_\Delta\left(\text{Dg}(G_1, f_1), \text{Dg}(G_2, f_2)\right).$$

For a Morse-type function f defined on $\mathcal{X}$ and for a finite point cloud $\mathbb{X}_n \subset \mathcal{X}$, we can thus consider $\text{Dg}(\text{R}_f(\mathcal{X})) = \text{Dg}(\text{R}_f(\mathcal{X}), f_\text{R})$ and $\text{Dg}(\text{M}_n) = \text{Dg}(\text{M}_n, f_\mathcal{I})$, with $f_\mathcal{I}$ as in Definition 3. In this context the bottleneck distance $d_\Delta(\text{R}_f(\mathcal{X}), \text{M}_n) = d_\Delta(\text{Dg}(\text{R}_f(\mathcal{X})), \text{Dg}(\text{M}_n))$ is well defined and we use this quantity to assess if the Mapper M_n is a good approximation of the Reeb graph $\text{R}_f(\mathcal{X})$. Moreover, note that, even though d_Δ is only a pseudometric, it has been shown to be a true metric *locally* for Reeb graphs by Carrière and Oudot (2017a).

As noted in Carrière and Oudot (2017b), the choice of $f_\mathcal{I}$ is in some sense arbitrary since any function defined on the nodes of the Mapper that respects the ordering of the intervals of $\mathcal{I}$ carries the same information in its extended persistence diagram. To avoid this issue, Carrière and Oudot (2017b) define a pruned version of $\text{Dg}(\text{R}_f(\mathcal{X}), f_\text{R})$ as a canonical descriptor for the Mapper. The problem with this approach is that computing this canonical descriptor requires to know the critical values of f_R beforehand. Here, by considering $\text{Dg}(\text{M}_n, f_\mathcal{I})$ instead, the descriptor becomes computable. Moreover, one can see from the proofs in the Appendix that the canonical descriptor and its arbitrary version actually enjoy the same rate of convergence, up to some constant.

2.3 An approximation inequality for Mapper

We are now ready to give the key ingredient of this paper to derive a statistical analysis of the Mapper. The ingredient is an upper bound on the bottleneck distance between the Reeb graph of a pair $(\mathcal{X}, f)$ and the Mapper computed with the same filter function f and a specific cover $\mathcal{I}$ of a sampled point cloud $\mathbb{X}_n \subset \mathcal{X}$. From now on, it is assumed that the underlying space $\mathcal{X}$ is a smooth and compact submanifold embedded in $\mathbb{R}^D$, and that the filter function f is Morse-type on $\mathcal{X}$.

Regularity of the filter function. Intuitively, approximating a Reeb graph computed with a filter function f that has large variations is more difficult than for a smooth filter function, for some notion of regularity that we now specify. Our result is given in a general setting by considering the modulus of continuity of f. In our framework, f is assumed to be Morse-type and thus uniformly continuous on the compact set $\mathcal{X}$. Following for instance Section 6 in DeVore and Lorentz (1993), we define the exact modulus of continuity of f as:

$$\omega_f(\delta) = \sup_{\|x - x'\| \leq \delta} |f(x) - f(x')|$$

for any $\delta > 0$, where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^D$. Then ω_f satisfies :

1. $\omega_f(\delta) \to \omega(0) = 0$ when $\delta \to 0$;
2. ω_f is non-negative and non-decreasing on $\mathbb{R}^+$;
3. ω_f is subadditive : $\omega_f(\delta_1 + \delta_2) \leq \omega_f(\delta_1) + \omega_f(\delta_2)$ for any $\delta_1, \delta_2 > 0$;
4. ω_f is continous on $\mathbb{R}^+$.

In this paper we say that a function ω defined on $\mathbb{R}^+$ is *a modulus of continuity* if it satisfies the four properties above and we say that it is *a modulus of continuity for f* if, in addition, we have

$$|f(x) - f(x')| \leq \omega(\|x - x'\|),$$

for any $x, x' \in \mathcal{X}$.

Theorem 7 *Assume that $\mathcal{X}$ has positive reach* rch *and convexity radius ρ. Let $\mathbb{X}_n$ be a point cloud of n points, all lying in $\mathcal{X}$. Assume that the filter function f is Morse-type on $\mathcal{X}$. Let ω be a modulus of continuity for f. Finally, let r, g be Mapper parameters defined as per Equations (1) and (2). If the three following conditions hold:*

$$\delta \leq \frac{1}{4} \min\{\text{rch}, \rho\}, \tag{3}$$

$$\max\{|f(X) - f(X')| : X, X' \in \mathbb{X}_n \text{ and } \|X - X'\| \leq \delta\} < gr, \tag{4}$$

$$4 d_\text{H}(\mathcal{X}, \mathbb{X}_n) \leq \delta, \tag{5}$$

where d_H denotes the Hausdorff distance, then the Mapper $\text{M}_n = \text{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ with parameters r, g and δ is such that:

$$d_\Delta\left(\text{R}_f(\mathcal{X}), \text{M}_n\right) \leq r + 2\omega(\delta). \tag{6}$$

Remark 8 *Using the edge-based MultiNerve Mapper—as defined in Section 8 of Carrière and Oudot (2017b)—allows to weaken Assumption (4) since gr can be replaced by r in the corresponding equation, and r can be replaced by $r/2$ in Equation (6).*

Analysis of the hypotheses. On the one hand, the scale parameter δ of the neighborhood graph could not be smaller than the approximation error corresponding to the Hausdorff distance between the sample and the underlying space $\mathcal{X}$ (Assumption (5)). On the other hand, it must be smaller than the reach and convexity radius to provide a correct estimation of the geometry and topology of $\mathcal{X}$ (Assumption (3)). The quantity gr corresponds to the minimum scale at which the filter's codomain is analyzed. This minimum resolution has to be compared with the regularity of the filter at scale δ (Assumption (4)). Indeed the pre-images of a filter with strong variations will be more difficult to analyze than when the filter does not vary too fast.

Analysis of the upper bound. The upper bound given in (6) makes sense in that the approximation error is controlled by the resolution level in the codomain and by the regularity of the filter. If one uses a filter with strong variations, or if the grid in the codomain has a too rough resolution, then the approximation will be poor. On the other hand, a sufficiently dense sampling is required in order to take r small, as prescribed in the assumptions.

Lipschitz filters. A large class of filters used for the Mapper are actually Lipschitz functions and of course, in this case, one can take $\omega(\delta) = c\delta$ for some positive constant c. In particular, $c = 1$ for linear projections (PCA, SVD, Laplacian or coordinate filter for instance). The distance to a measure (DTM) is also a 1-Lipschitz function, see Chazal et al. (2011). On the other hand, the modulus of continuity of filter functions defined from estimators, e.g. density estimators, is less obvious although still well-defined.

Filter approximation. In some situations, the filter function $\hat{f}$ used to compute the Mapper is only an approximation of the filter function f with which the Reeb graph is computed. In this context, the pair $(\mathbb{X}_n, \hat{f})$ appears as an approximation of the pair $(\mathcal{X}, f)$. The following result is directly derived from Theorem 7 and Theorem 5.1 in Carrière and Oudot (2017b) (that derives stability for Mappers building on the stability theorem of extended persistence diagrams proved by Cohen-Steiner et al. (2009)):

Corollary 9 *Let $\hat{f} : \mathcal{X} \to \mathbb{R}$ be a Morse-type filter function approximating f. Assume that Assumptions (3) and (5) of Theorem 7 are satisfied, and assume moreover that*

$$\max\{\max\{|f(X) - f(X')|, |\hat{f}(X) - \hat{f}(X')|\} \,:\, X, X' \in \mathbb{X}_n, \|X - X'\| \leq \delta\} < gr. \qquad (7)$$

Then, the Mapper $\hat{\mathrm{M}}_n = \mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \hat{f}(\mathbb{X}_n))$ built on $\mathbb{X}_n$ with filter function $\hat{f}$ and parameters r, g, δ satisfies:

$$d_\Delta(\mathrm{R}_f(\mathcal{X}), \hat{\mathrm{M}}_n) \leq 2r + 2\omega(\delta) + \max_{1 \leq i \leq n} |f(X_i) - \hat{f}(X_i)|.$$

3. Statistical Analysis of Mapper

From now on, the set of observations $\mathbb{X}_n$ is assumed to be composed of n independent points $X_1, ..., X_n$ sampled from a probability distribution $\mathbb{P}$ in $\mathbb{R}^D$ (endowed with its Borel algebra). We assume that each point X_i comes with a filter value which is represented by a random variable Y_i. Contrarily to the X_i's, the filter values Y_i's are not necessarily independent. In the following, we consider two different settings: in the first one, $Y_i = f(X_i)$, where the

filter f is a deterministic function, in the second one, $Y_i = \hat{f}(X_i)$ where $\hat{f}$ is an estimator of the filter function f. In the latter case, the Y_i's are obviously dependent. We first provide the following proposition, whose proof is deferred to Appendix A.4, which states that computing probabilities on the Mapper makes sense:

Proposition 10 *For any fixed choice of parameters r, g, δ and for any fixed $n \in \mathbb{N}$, the function*

$$\Phi : \begin{cases} (\mathbb{R}^D)^n \times \mathbb{R}^n & \to & \mathcal{R} \\ (\mathbb{X}_n, \mathbb{Y}_n) & \mapsto & \mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n) \end{cases}$$

is measurable, where $\mathcal{R}$ denotes the set of Reeb graphs computed from Morse-type functions.

3.1 Statistical Model for the Mapper

In this section, we study the convergence of the Mapper for a general generative model and a class of filter functions. We first introduce the generative model and next we present different settings depending on the nature of the filter function.

Generative model. The set of observations $\mathbb{X}_n$ is assumed to be composed of n independent points $X_1, ..., X_n$ sampled from a probability distribution $\mathbb{P}$ in $\mathbb{R}^D$. The support of $\mathbb{P}$ is denoted $\mathcal{X}_\mathbb{P}$ and is assumed to be a smooth and compact submanifold of $\mathbb{R}^D$ with positive reach and positive convexity radius, as in the setting of Theorem 7. We also assume that $0 < \mathrm{diam}(\mathcal{X}_\mathbb{P}) \leq L$. Next, the probability distribution $\mathbb{P}$ is assumed to be (a, b)-standard for some constants $a > 0$ and $b \geq D$, that is for any Euclidean ball $B(x, t)$ centered on $x \in \mathcal{X}$ with radius t :

$$\mathbb{P}(B(x, t)) \geq \min(1, at^b).$$

This assumption is popular in the literature about set estimation (see for instance Cuevas, 2009; Cuevas and Rodríguez-Casal, 2004). It is also widely used in the TDA literature (Chazal et al., 2015b; Fasy et al., 2014; Chazal et al., 2015a). For instance, when $b = D$, this assumption is satisfied when the distribution is absolutely continuous with respect to the Hausdorff measure on $\mathcal{X}_\mathbb{P}$. We introduce the set $\mathcal{P}_{a,b} = \mathcal{P}_{a,b,\kappa,\rho,L}$ which is composed of all the (a, b)-standard probability distributions for which the support $\mathcal{X}_\mathbb{P}$ is a smooth and compact submanifold of $\mathbb{R}^D$ with reach larger than κ, convexity radius larger than ρ and diameter less than L.

Filter functions in the statistical setting. The filter function $f : \mathcal{X}_\mathbb{P} \to \mathbb{R}$ for the Reeb graph is assumed as before to be a Morse-type function. Two different settings have to be considered regarding how the filter function is defined. In the first setting, the same filter function is used to define the Reeb graph and the Mapper. The Mapper can be defined by taking the exact values of the filter function at the observation points $f(X_1), \ldots, f(X_n)$. Note that this does not mean that the function f is completely known since, in our framework, knowing f would imply to know its domain and thus $\mathcal{X}_\mathbb{P}$ would be known which is of course not the case in practice. This first setting is referred to as the *exact filter setting* in the following. It corresponds to the situations where the Mapper algorithm is used with coordinate functions for instance. In the second setting, the filter function used for the Mapper is not available and an estimation of this filter function has to be computed from the data. This second setting is referred to as the *inferred filter setting* in

the following. It corresponds to PCA or Laplacian eigenfunctions, distance functions (such as the DTM), or regression and density estimators.

Risk of the Mapper. We study, in various settings, the problem of inferring a Reeb graph using Mappers and we use the metric d_Δ to assess the performance of the Mapper, seen as an estimator of the Reeb graph. Hence, we study the following quantity:

$$\mathbb{E}\left[d_\Delta\left(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_\mathbb{P})\right)\right],$$

where M_n is computed with the exact filter f or the inferred filter $\hat{f}$, depending on the context.

3.2 Reeb graph inference with exact filter and known generative model

We first consider the exact filter setting in the simplest situation where the parameters a and b of the generative model are known. In this setting, for a given neighborhood graph parameter δ, gain g and resolution r, the Mapper $\mathrm{M}_n = \mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ is computed with $\mathbb{Y}_n = f(\mathbb{X}_n)$.

Parameter selection. We now tune the triple of parameters (r, g, δ) depending on the parameters a and b. More precisely, we take:

$$\text{an arbitrary } g \in \left(\frac{1}{3}, \frac{1}{2}\right), \quad \delta_n = 8\left(\frac{2\log(n)}{an}\right)^{1/b}, \quad r_n = \frac{V_n(\delta_n)^+}{g}, \tag{8}$$

where $V_n(\delta_n) = \max\{|f(X) - f(X')| : X, X' \in \mathbb{X}_n, \|X - X'\| \leq \delta_n\}$, and $V_n(\delta_n)^+$ denotes a value that is strictly larger but arbitrarily close to $V_n(\delta_n)$.

Upper bound. We give below a general upper bound on the risk of M_n with these parameters, which depends on the regularity of the filter function and on the parameters of the generative model. We show a uniform convergence over a class of possible filter functions. This class of filters necessarily depends on the support of $\mathbb{P}$, so we define the class of filters for each probability measure in $\mathcal{P}_{a,b}$. For any $\mathbb{P} \in \mathcal{P}_{a,b}$, we let $\mathcal{F}(\mathbb{P}, \omega)$ denote the set of filter functions $f : \mathcal{X}_\mathbb{P} \to \mathbb{R}$ such that f is Morse-type on $\mathcal{X}_\mathbb{P}$ with $\omega_f \leq \omega$.

Proposition 11 *Let ω be a modulus of continuity for f such that $\omega(x)/x$ is a non-increasing function on $\mathbb{R}^+$. For n large enough, the Mapper computed with parameters (r_n, g, δ_n) as per Equation (8) satisfies*

$$\sup_{\mathbb{P}\in\mathcal{P}_{a,b}} \mathbb{E}\left[\sup_{f\in\mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}), \mathrm{M}_n\right)\right] \leq C\,\omega\left(\frac{2\cdot 8^b}{a}\frac{\log(n)}{n}\right)^{1/b}$$

where the constant C only depends on a, b, and on the geometric parameters of the model.

Assuming that $\omega(x)/x$ is non-increasing is not a very strong assumption. This property is satisfied in particular when ω is concave, as in the case of concave majorant (see for instance Section 6 in DeVore and Lorentz (1993)). As expected, we see that the rate of convergence of the Mapper to the Reeb graph directly depends on the regularity of the

filter function and on the parameter b which roughly represents the intrinsic dimension of the data. For Lipschitz filter functions, the rate is similar to the one for persistence diagram inference in Chazal et al. (2015b), namely it corresponds to the one of support estimation for the Hausdorff metric (see for instance Cuevas and Rodríguez-Casal (2004)) and Genovese et al. (2012a)). In the other cases where the filters only admit a concave modulus of continuity, we see that the "distortion" created by the filter function slows down the convergence of the Mapper to the Reeb graph.

We now give a lower bound that matches with the upper bound of Proposition 11.

Proposition 12 *Let ω be a modulus of continuity for f. Then, for any estimator $\hat{\mathrm{R}}_n$ of $\mathrm{R}_f(\mathcal{X}_\mathbb{P})$,, we have*

$$\sup_{\mathbb{P}\in\mathcal{P}_{a,b}} \mathbb{E}\left[\sup_{f\in\mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}), \hat{\mathrm{R}}_n\right)\right] \geq C\,\omega\left(\frac{1}{an}\right)^{\frac{1}{b}},$$

where the constant C only depends on a, b and on the geometric parameters of the model.

Propositions 11 and 12 together show that, with the choice of parameters given before, M_n is minimax optimal up to a logarithmic factor $\log(n)$ inside the modulus of continuity. Note that the lower bound is also valid whether or not the coefficients a and b and the filter function f and its modulus of continuity are given.

3.3 Reeb graph inference with exact filter and unknown generative model

We still assume that the exact values $\mathbb{Y}_n = f(\mathbb{X}_n)$ of the filter on the point could can be computed and that at least a modulus of continuity for the filter is known. However, the parameters a and b are not assumed to be known anymore. We adapt a subsampling approach proposed by Fasy et al. (2014). As before, for a given neighborhood graph parameter δ, gain g and resolution r, the Mapper $\mathrm{M}_n = \mathrm{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ is computed with $\mathbb{Y}_n = f(\mathbb{X}_n)$.

Parameter selection. We introduce the sequence $s_n = \frac{n}{(\log n)^{1+\beta}}$ for some fixed value $\beta > 0$. Let $\hat{\mathbb{X}}_n^{s_n}$ be an arbitrary subset of $\mathbb{X}_n$ that contains s_n points. Then, we take:

$$\text{an arbitrary } g \in \left(\frac{1}{3}, \frac{1}{2}\right), \quad \delta_n = d_{\mathrm{H}}(\hat{\mathbb{X}}_n^{s_n}, \mathbb{X}_n), \quad r_n = \frac{V_n(\delta_n)^+}{g}, \tag{9}$$

where V_n^+ is defined as in Equation (8).

Upper bound. Using these parameters, we can then derive the following upper bound:

Proposition 13 *Let ω be a modulus of continuity for f such that $x \mapsto \omega(x)/x$ is a non-increasing function. Then, using the same notations as in the previous section, the Mapper M_n computed with parameters (r_n, g, δ_n) as per Equation (9) satisfies*

$$\sup_{\mathbb{P}\in\mathcal{P}_{a,b}} \mathbb{E}\left[\sup_{f\in\mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}), \mathrm{M}_n\right)\right] \leq C\,\omega\left(\frac{C'\log(n)^{2+\beta}}{n}\right)^{1/b},$$

where the constants C, C' only depends on a, b, and on the geometric parameters of the model.

Up to logarithmic factors inside the modulus of continuity, we find that this Mapper is still minimax optimal over the class $\mathcal{P}_{a,b}$ by Proposition 12.

3.4 Reeb graph inference with inferred filter and unknown generative model

One of the nice properties of the Mapper is that it can be easily computed with any filter function, including estimated filter functions such as PCA eigenfunctions, eccentricity functions, DTM functions, Laplacian eigenfunctions, density estimators, regression estimators, and many other filters directly estimated from the data. In this section, we assume that the *true filter* f is unknown but can be estimated from the data using an estimator $\hat{f}$. Without loss of generality, we assume that both f and $\hat{f}$ are defined on $\mathbb{R}^D$. As before, parameters a and b are not assumed to be known and we have to tune the triple of parameters (r_n, g, δ_n).

Parameter selection. In this context, the quantity V_n^+ of Equations (8) and (9) cannot be computed as before because there is no direct access to the values of f: we only know an estimation $\hat{f}$ of it. However, in many cases, a modulus of continuity ω_1 for f is known, which makes possible the tuning of the parameters. For instance, PCA (and kernel) projectors, eccentricity functions, DTM functions (see Chazal et al. (2011)) are all 1-Lipschitz functions, and Corollary 14 below can be applied.

Let $\hat{V}_n(\delta_n) = \max\{|\hat{f}(X) - \hat{f}(X')| \,:\, X, X' \in \mathbb{X}_n, \|X - X'\| \leq \delta_n\}$, and let ω_1 be a modulus of continuity for f. Then, we take:

$$\text{an arbitrary } g \in \left(\frac{1}{3}, \frac{1}{2}\right), \quad \delta_n = d_{\mathrm{H}}(\hat{\mathbb{X}}_n^{s_n}, \mathbb{X}_n), \quad r_n = \frac{\max\{\omega_1(\delta_n), \hat{V}_n(\delta_n)\}^+}{g}. \tag{10}$$

Upper bound. Following the lines of the proof of Proposition 13 and applying Corollary 9, we obtain:

Corollary 14 *Let $f : \mathbb{R}^D \to \mathbb{R}$ be a Morse-type filter function and let $\hat{f} : \mathbb{R}^D \to \mathbb{R}$ be a Morse-type estimator of f. Let ω_1 (resp. ω_2) be a modulus of continuity for f (resp. $\hat{f}$). Let $\omega = \max\{\omega_1, \omega_2\}$ such that $x \mapsto \omega(x)/x$ is a non-increasing function. Let also $\hat{\mathrm{M}}_n = \mathrm{M}_{r_n,g,\delta_n}(\mathbb{X}_n, \hat{f}(\mathbb{X}_n))$ be the Mapper built on $\mathbb{X}_n$ with function $\hat{f}$ and parameters g, δ_n, r_n as in Equation (10). Then, $\hat{\mathrm{M}}_n$ satisfies*

$$\mathbb{E}\left[d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}), \hat{\mathrm{M}}_n\right)\right] \leq C\omega\left(\frac{C'\log(n)^{2+\beta}}{n}\right)^{\frac{1}{b}} + \mathbb{E}\left[\max_{1\leq i\leq n} |f(X_i) - \hat{f}(X_i)|\right],$$

where the constants C, C' only depends on a, b, and on the geometric parameters of the model.

Note that ω_1 has to be known to compute $\hat{\mathrm{M}}_n$ in Corollary 14 since it appears in the definition of r_n. On the contrary, ω_2—and thus ω—is not required to tune the parameters.

PCA eigenfunctions. In the setting of this article, the measure μ has a finite second moment. Following Biau and Mas (2012), we define the covariance operator $\Gamma(\cdot) = \mathbb{E}(\langle X, \cdot\rangle X)$ and we let Π_k denote the orthogonal projection onto the space spanned by the k-th eigenvector of Γ. In practice, we consider the empirical version of the covariance operator

$$\hat{\Gamma}_n(\cdot) = \frac{1}{n}\sum_{i=1}^n \langle X_i, \cdot\rangle X_i$$

and the empirical projection $\hat{\Pi}_k$ onto the space spanned by the k-th eigenvector of $\hat{\Gamma}_n$. According to Biau and Mas (2012)(see also Blanchard et al. (2007); Shawe-Taylor et al. (2005)), we have

$$\mathbb{E}\left[\|\Pi_k - \hat{\Pi}_k\|_\infty\right] = O\left(\frac{1}{\sqrt{n}}\right).$$

This, together with Corollary 14 and the fact that both Π_k and $\hat{\Pi}_k$ are 1-Lipschitz, gives that the rate of convergence of the Mapper of $\hat{\Pi}_k(\mathbb{X}_n)$ computed with parameters δ_n, g and r_n as in Equation (10) (which gives $r_n = g^{-1}\delta_n^+$) satisfies

$$\mathbb{E}\left[d_\Delta\left(\mathrm{R}_{\Pi_k}(\mathcal{X}_\mathbb{P}), \mathrm{M}_{r_n,g,\delta_n}(\mathbb{X}_n, \hat{\Pi}_k(\mathbb{X}_n))\right)\right] = O\left(\max\left\{\left(\frac{\log(n)^{2+\beta}}{n}\right)^{\frac{1}{b}}, \frac{1}{\sqrt{n}}\right\}\right).$$

Hence, the rate of convergence of Mapper is not deteriorated by using $\hat{\Pi}_k$ instead of Π_k if the intrinsic dimension b of the support of μ is at least 2.

The distance to measure. It is well known that TDA methods may fail completely in the presence of outliers. To address this issue, Chazal et al. (2011) introduced an alternative distance function which is robust to noise, the *distance-to-measure* (DTM). A similar analysis as with the PCA filter can be carried out with the DTM filter using the rates of convergence proven in Chazal et al. (2016b).

4. Confidence sets for Reeb signatures

4.1 Confidence sets for extended persistence diagrams

In practice, computing a Mapper M_n and its signature $\mathrm{Dg}(\mathrm{M}_n, f_\mathcal{I})$ is not sufficient: we need to know how accurate these estimations are. One natural way to answer this problem is to provide a confidence set for the Mapper using the bottleneck distance. For $\alpha \in (0, 1)$, we look for some value $\eta_{n,\alpha}$ such that

$$\mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_\mathbb{P})) \geq \eta_{n,\alpha}\right) \leq \alpha$$

or at least such that

$$\limsup_{n\to\infty} \mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_\mathbb{P})) \geq \eta_{n,\alpha}\right) \leq \alpha.$$

Let

$$\mathcal{M}_\alpha = \{\mathrm{R} \in \mathcal{R} \,:\, d_\Delta(\mathrm{M}_n, \mathrm{R}) \leq \alpha\}$$

be the closed ball of radius α in the bottleneck distance and centered at the Mapper M_n in the space of Reeb graphs $\mathcal{R}$. Following Fasy et al. (2014), we can visualize the signatures of the points belonging to this ball in various ways. One first option is to center a box of side length 2α at each point of the extended persistence diagram of M_n—see the right columns of Figure 5 and Figure 6 for instance. An alternative solution is to visualize the confidence set by adding a band at (vertical) distance 2α from the diagonal (the bottleneck distance being defined for the ℓ_∞ norm). The points outside the band are then considered as significant topological features, see Fasy et al. (2014) for more details.

Several methods have been proposed in Fasy et al. (2014) and Chazal et al. (2014) to define confidence sets for persistence diagrams. We now adapt these ideas to provide confidence sets for Mappers. Except for the bottleneck bootstrap (see Section 4.3), all the methods proposed in these two articles rely on the stability results for persistence diagrams, which say that persistence diagrams equipped with the bottleneck distance are stable under Hausdorff or Wasserstein perturbations of the data. Confidence sets for diagrams are then directly derived from confidence sets in the sample space. Here, we follow a similar strategy using Theorem 7, as explained in the next section.

4.2 Confidence sets derived from Theorem 7

In this section, we always assume that an upper bound ω on the exact modulus of continuity ω_f of the filter function is known. We start with the following remark: if we can take δ of the order of $d_{\mathrm{H}}(\mathcal{X}_{\mathbb{P}}, \mathbb{X}_n)$ in Theorem 7 and if all the conditions of the theorem are satisfied, then $d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}}))$ can be bounded in terms of $\omega(d_{\mathrm{H}}(\mathcal{X}_{\mathbb{P}}, \mathbb{X}_n))$. This means that we can adapt the methods of Fasy et al. (2014) to Mappers.

Known generative model. Let us first consider the simplest situation where the parameters a an b are also known. Following Section 3.2, we choose for g, δ_n, r_n as per Equation (8). Let $\varepsilon_n = d_{\mathrm{H}}(\mathcal{X}_{\mathbb{P}}, \mathbb{X}_n)$. As shown in the proof of Proposition 11 (see Appendix A.5), for n large enough, Assumption (3) and (4) are always satisfied and then

$$\mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}})) \geq \eta\right) \leq \mathbb{P}\left(\delta_n \geq \omega^{-1}\left(\frac{\eta}{g^{-1}+2}\right)\right).$$

Consequently,

$$\begin{aligned}\mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}})) \geq \eta\right) &\leq \mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}})) \geq \eta \cap \varepsilon_n \leq 4\delta_n\right) + \mathbb{P}\left(\varepsilon_n > 4\delta_n\right) \\ &\leq \mathbb{I}_{\omega(\delta_n) \geq \frac{g}{1+2g}\eta} + \min\left\{1, \frac{2^b}{2\log(n)n}\right\} \\ &= \Phi_n(\eta).\end{aligned}$$

where Φ_n depends on the parameters of the model (or some bounds on these parameters) which are here assumed to be known. Hence, given a probability level α, one has:

$$\mathbb{P}\left(d_\Delta(\mathrm{M}_n, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}})) \geq \Phi_n^{-1}(\alpha)\right) \leq \alpha.$$

Unknown generative model. We now assume that a and b are unknown. To compute confidence sets for the Mapper in this context, we approximate the distribution of $d_{\mathrm{H}}(\mathcal{X}_{\mathbb{P}}, \mathbb{X}_n)$ using the distribution of $d_{\mathrm{H}}(\hat{\mathbb{X}}_n^{s_n}, \mathbb{X}_n)$ conditionally to $\mathbb{X}_n$. There are $N_1 = \binom{n}{s_n}$ subsets of size s_n inside $\mathbb{X}_n$, so we let $\mathbb{X}_{s_n}^1, \ldots, \mathbb{X}_{s_n}^{N_1}$ denote all the possible configurations. Define

$$L_n(t) = \frac{1}{N_1}\sum_{k=1}^{N_1} \mathbb{I}_{d_{\mathrm{H}}\left(\mathbb{X}_{s_n}^k, \mathbb{X}_n\right) > t}.$$

Let s be the function on $\mathbb{N}$ defined by $s(n) = s_n$ and let $s_n^2 = s(s(n))$. There are $N_2 = \binom{n}{s_n^2}$ subsets of size s_n^2 inside $\mathbb{X}_n$. Again, we let $\mathbb{X}_{s_n^2}^k$, $1 \leq k \leq N_2$, denote these configurations

and we also introduce

$$F_n(t) = \frac{1}{N_2}\sum_{k=1}^{N_2} \mathbb{I}_{d_{\mathrm{H}}\left(\mathbb{X}_{s_n^2}^k, \mathbb{X}_{s_n}\right) > t}.$$

Proposition 15 *Let $\eta > 0$. Then, one has the following confidence set:*

$$\mathbb{P}\left(d_\Delta(\mathrm{R}_f(\mathcal{X}_{\mathbb{P}}), \mathrm{M}_n) \geq \eta\right) \leq F_n\left(\frac{1}{4}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + L_n\left(\frac{1}{4}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + o\left(\frac{s_n}{n}\right)^{\frac{1}{4}}.$$

Both F_n and L_n can be computed in practice, or at least approximated using Monte Carlo procedures. The upper bound on $\mathbb{P}\left(d_\Delta(\mathrm{R}_f(\mathcal{X}_{\mathbb{P}}), \mathrm{M}_n) \geq \eta\right)$ then provides an asymptotic confidence region for the persistence diagram of the Mapper M_n, which can be explicitly computed in practice. See the green squares in the first row of Figure 5. The main drawback of this approach is that it requires knowing a modulus of continuity ω and, more importantly, the number of observations has to be very large, which is not the case on our examples in Section 5.

Modulus of continuity of the filter function. As shown in Proposition 15, the modulus of continuity of the filter function is a key quantity to describe the confidence regions. Inferring the modulus of continuity of the filter from the data is a tricky problem. Fortunately, in practice, even in the inferred filter setting, a modulus of continuity for the function is known in many situations. For instance, projections such as PCA eigenfunctions and DTM functions are 1-Lipschitz.

4.3 Bottleneck Bootstrap

The two methods given before both require an explicit upper bound on the modulus of continuity of the filter function. Moreover, these methods both rely on the approximation result Theorem 7, which often leads to conservative confidence sets. An alternative strategy is the bottleneck bootstrap introduced in Chazal et al. (2014), and which we now apply to our framework.

Bootstrap. The bootstrap is a general method for estimating standard errors and computing confidence intervals. Let $\mathbb{P}_n$ be the empirical measure defined from the sample $(X_1, Y_1), \ldots, (X_n, Y_n)$. Let $(X_1^*, Y_1^*) \ldots, (X_n^*, Y_n^*)$ be a sample from $\mathbb{P}_n$ and let also M_n^* be the random Mapper defined from this sample. We then take for $\hat{\eta}_{n,\alpha}$ the quantity $\hat{\eta}_{n,\alpha}^*$ defined by

$$\mathbb{P}\left(d_\Delta(\mathrm{M}_n^*, \mathrm{M}_n) > \hat{\eta}_{n,\alpha}^* \,|\, X_1, \ldots, X_n\right) = \alpha. \tag{11}$$

Note that $\hat{\eta}_{n,\alpha}^*$ can be easily estimated with Monte Carlo procedures. It has been shown in Chazal et al. (2014) that the bottleneck bootstrap is valid when computing the sublevel sets of a density estimator. The validity of the bottleneck bootstrap has not been proven for the extended persistence diagram of any distance function. For Mapper, it would require writing $d_\Delta(\mathrm{M}_n^*, \mathrm{M}_n)$ in terms of the distance between the extrema of the filter function and the ones of the interpolation of the filter function on the δ-neighborhood graph. We leave this problem open in this article.

Extension of the analysis. As pointed out in Section 2.1, many versions of the Mapper exist in the literature. One of them, called the *edge-based MultiNerve Mapper* $\overline{\mathrm{M}}^{\triangle}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$, is described in Section 8 of Carrière and Oudot (2017b). The main advantage of this version is that it allows for finer resolutions than the usual Mapper while remaining fast to compute. Our analysis can actually handle this version as well by replacing gr by r in Assumption (4) of Theorem 7—see Remark 8, and changing constants accordingly in the proofs. In particular, this improves the resolution r_n in Equation (9) since $g^{-1}V_n(\delta_n)^+$ becomes $V_n(\delta_n)^+$. Hence, we use this edge-based version in Section 5, where this improvement on the resolution r_n allows us to compensate for the low number of observations.

5. Numerical experiments

In this section, we provide few examples of parameter selections and confidence regions (which are unions of squares in the extended persistence diagrams) obtained with bottleneck bootstrap. The interpretation of these regions is that squares that intersect the diagonal, which are drawn in pink color, represent topological features in the Mappers that may be horizontal or artifacts due to the cover, and that may not be present in the Reeb graph. We show in Figure 5 various Mappers (in each node of the Mappers, the left number is the cluster ID and the right number is the number of observations in that cluster) and 85 percent confidence regions computed on various data sets. All δ parameters and resolutions were computed with Equation (9) (the δ parameters were also averaged over $N = 100$ subsamplings with $\beta = 0.001$), and all gains were set to 40%. The code we used is available in the Gudhi open source library (see Carrière (2017)). The confidence regions were computed by bootstrapping data 100 times. Note that computing confidence regions with Proposition 15 is possible, but the numbers of observations in all of our data sets were too low, leading to conservative confidence regions that did not allow for interpretation.

5.1 Mappers and confidence regions

Synthetic example. We computed the Mapper of an embedding of the Klein bottle into $\mathbb{R}^4$ with 10,000 points with the height function. In order to illustrate the conservativity of confidence regions computed with Proposition 15, we also plot these regions for an embedding with 10,000,000 points using the fact that the height function is 1-Lipschitz. Corresponding squares are drawn in green color. Their very large sizes show that Proposition 15 requires a very large number of observations in practice. See the first row of Figure 5.

3D shapes. We computed the Mapper of an ant shape and a human shape from Chen et al. (2009) embedded in $\mathbb{R}^3$ (with 4,706 and 6,370 points respectively) Both Mappers were computed with the height function. One can see that the confidence squares for the features that are almost horizontal (such as the small branches in the Mapper of the ant) intersect indeed the diagonal. See the second and third rows of Figure 5.

Miller-Reaven data set. The first data set comes from the Miller-Reaven diabetes study that contains 145 observations of patients suffering or not from diabete. Observations were mapped into $\mathbb{R}^5$ by computing various medical features. Data can be obtained in the "locfit" R-package. In Reaven and Miller (1979), the authors identified two groups of diseases with

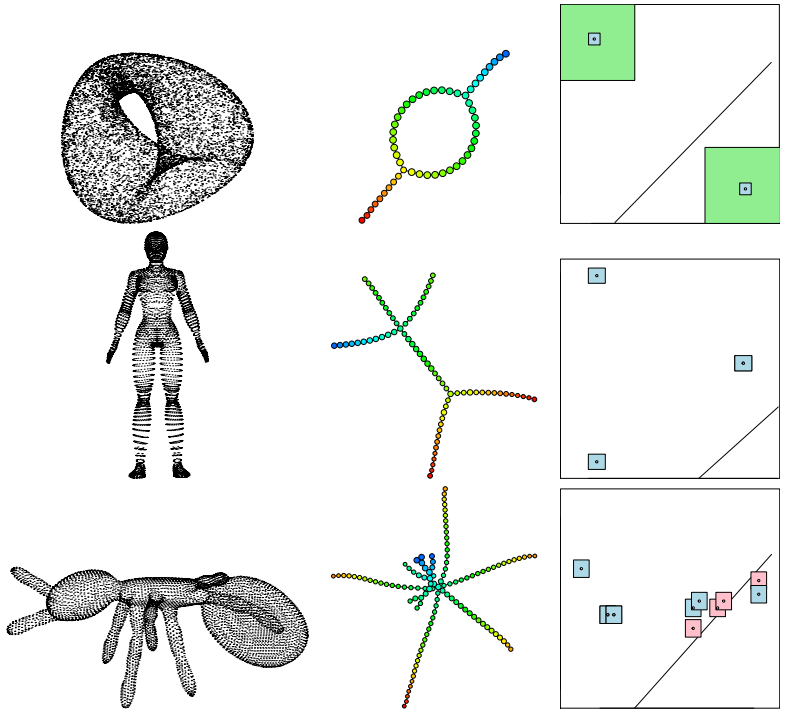


Figure 5: Mappers computed with automatic tuning (middle) and 85 percent confidence regions for their topological features (right) are provided for an embedding of the Klein Bottle into $\mathbb{R}^4$ (first row), a 3D human shape (second row) and a 3D ant shape (third row).

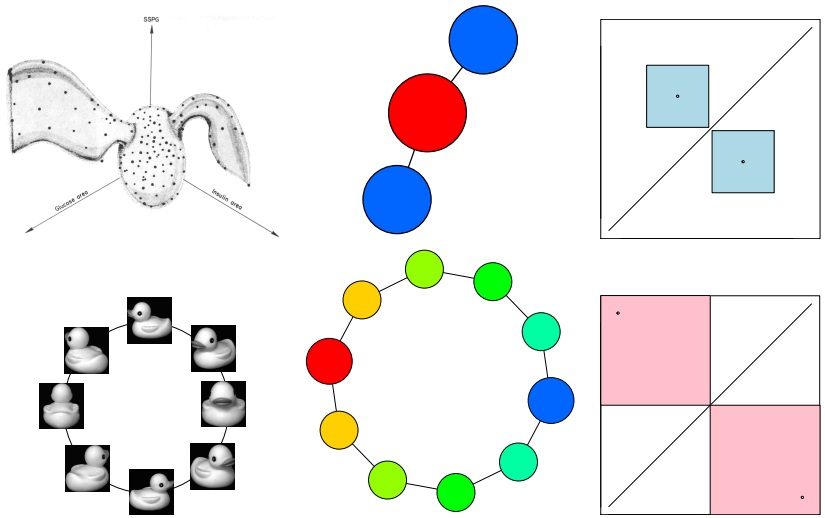


Figure 6: Mappers computed with automatic tuning (middle) and 85 percent confidence regions for their topological features (right) are provided for the Reaven-Miller data set (first row) and the COIL data set (second row).

the projection pursuit method, and in Singh et al. (2007), the authors applied Mapper with hand-crafted parameters to get back this result. Here, we normalized the data to zero mean and unit variance, and we obtained the two flares in the Mapper computed with the eccentricity function. Moreover, these flares are at least 85 percent sure since the confidence squares on the corresponding points in the extended persistence diagrams do not intersect the diagonal. See the first row of Figure 6.

COIL data set. The second data set is an instance of the 16,384-dimensional COIL data set of Nene et al. (1996). It contains 72 observations, each of which being a picture of a duck taken at a specific angle. Despite the low number of observations and the large number of dimensions, we managed to retrieve the intrinsic loop lying in the data using the first PCA eigenfunction. However, the low number of observations made the bootstrap fail since the confidence squares computed around the points that represent this loop in the extended persistence diagram intersect the diagonal. See the second row of Figure 6.

5.2 Noisy data

Denoising Mapper. An important drawback of Mapper is its sensitivity to noise and outliers. See the crater data set in Figure 7, for instance. Several answers have been proposed for recovering the correct persistence homology from noisy data. The idea is to

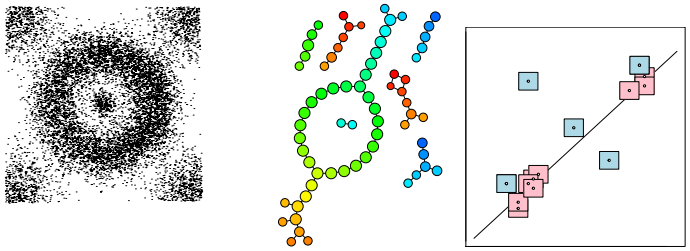


Figure 7: Mappers computed with automatic tuning (middle) and 85 percent confidence regions for their topological features (right) are provided for a noisy crater in the Euclidean plane.

use an alternative filtration of simplicial complexes instead of the Rips filtration. A first option is to consider the upper level sets of a density estimator rather than the distance to the sample (see Section 4.4 in Fasy et al. (2014)). Another solution is to consider the sublevel sets of the DTM and apply persistence homology inference in Chazal et al. (2014).

Crater data set. To handle noise in our crater data set, we simply smoothed the data set by computing the empirical DTM with 10 neighbors on each point and removing all points with DTM less than 40 percent of the maximum DTM in the data set. Then we computed the Mapper with the height function. One can see that all topological features in the Mapper that are most likely artifacts due to noise (like the small loops and connected components) have corresponding confidence squares that intersect the diagonal in the extended persistence diagram. See Figure 7.

6. Conclusion

In this article, we provided a statistical analysis of the Mapper. Namely, we proved the fact that the Mapper is a measurable construction in Proposition 10, and we used the approximation Theorem 7 to show that the Mapper is a minimax optimal estimator of the Reeb graph in various contexts—see Propositions 11, 12 and 13—and that corresponding confidence regions can be computed—see Proposition 15 and Section 4.3. Along the way, we derived rules of thumb to automatically tune the parameters of the Mapper with Equations (8), (9) and (10). Finally, we provided few examples of our methods on various data sets in Section 5.

Future directions. We plan to investigate several questions for future work.

- We will work on adapting results from Chazal et al. (2014) to prove the validity of bootstrap methods for computing confidence regions on the Mapper, since we only used bootstrap methods empirically in this article.

- We believe that using weighted versions of δ-neighborhood graphs, as defined in Buchet et al. (2015), would improve the quality of the confidence regions on the Mapper features, and would probably be a better way to deal with noise that our current solution.
- We plan to adapt our statistical setting to the question of selecting variables, which is one of the main applications of the Mapper in practice.

Acknowledgements. This work was supported by ERC grant Gudhi (ERC-2013-ADG-339025) and by ANR project TopData (ANR-13-BS01-0008). The authors would like to thank the anonymous referees for their constructive criticism and comments. The third author acknowledges the support of ICERM and Brown University, as part of this work was carried out while he was participating in the ICERM program *Topology in Motion* during the Fall of 2016.

Appendix A. Proofs

A.1 Preliminary results

In order to prove the results of this article, we need to state several preliminary definitions and theorems. All of them can be found, together with their proofs, in Dey and Wang (2013) and Carrière and Oudot (2017b). In this section, we let $\mathbb{X}_n \subset \mathcal{X}$ be a point cloud of n points sampled on a smooth and compact submanifold $\mathcal{X}$ embedded in $\mathbb{R}^D$, with positive reach rch and convexity radius ρ. Since δ-neighborhood graphs can be seen as 1-skeletons of Rips complexes with parameter δ, as per Definition 55 in Carrière and Oudot (2017b), and since many results are phrased with Rips complexes in the literature, we also use these complexes to state our results in this section. Let $f : \mathcal{X} \to \mathbb{R}$ be a Morse-type filter function, $\mathcal{I}$ be an open cover of the range of f with resolution r and gain $g < \frac{1}{2}$ (which ensures that no more than two cover elements can intersect at once, i.e. the cover is minimal), and $|\mathrm{Rips}_\delta(\mathbb{X}_n)|$ denote a geometric realization of the Rips complex built on top of $\mathbb{X}_n$ with parameter δ, and $f^{\mathrm{PL}} : |\mathrm{Rips}_\delta(\mathbb{X}_n)| \to \mathbb{R}$ be the piecewise-linear interpolation of f on the simplices of $\mathrm{Rips}_\delta(\mathbb{X}_n)$.

Definition 16 *Let $G = (\mathbb{X}_n, E)$ be a graph built on top of $\mathbb{X}_n$. Let $e = (X, X') \in E$ be an edge of G, and let $I(e)$ be the open interval $(\min\{f(X), f(X')\}, \max\{f(X), f(X')\})$. Then e is said to be* intersection-crossing *if there is a pair of consecutive intervals $I, J \in \mathcal{I}$ such that $\emptyset \neq I \cap J \subseteq I(e)$.*

Theorem 17 (Lemma 61 and 62 in Carrière and Oudot (2017b)). *Let $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$ denote the 1-skeleton of $\mathrm{Rips}_\delta(\mathbb{X}_n)$. If $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$ has no intersection-crossing edges, then $\mathrm{M}_{r,g,\delta}(\mathbb{X}_n, f(\mathbb{X}_n))$ and $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$ are isomorphic as combinatorial graphs.*

Theorem 18 (Theorem 54 in Carrière and Oudot (2017b)). *Let $f : \mathcal{X} \to \mathbb{R}$ be a Morse-type function. Then, we have the following inequality between extended persistence diagrams:*

$$d_\Delta(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{M}_{r,g}(\mathcal{X}, f), f_{\mathcal{I}})) \leq r. \tag{12}$$

Moreover, given another Morse-type function $\hat{f} : \mathcal{X} \to \mathbb{R}$, we have:

$$d_\Delta(\mathrm{Dg}(\mathrm{M}_{r,g}(\mathcal{X}, f), f_{\mathcal{I}}), \mathrm{Dg}(\mathrm{M}_{r,g}(\mathcal{X}, \hat{f}), \hat{f}_{\mathcal{I}})) \leq r + \|f - \hat{f}\|_\infty. \tag{13}$$

Theorem 19 (Theorem 4.6, Remark 2 in Dey and Wang (2013) and Theorem 59 in Carrière and Oudot (2017b)). *If $4d_{\mathrm{H}}(\mathcal{X}, \mathbb{X}_n) \leq \delta \leq \min\{\mathrm{rch}/4, \rho/4\}$, then:*

$$d_\Delta(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|), f^{\mathrm{PL}}_{\mathrm{R}})) \leq 2\omega(\delta).$$

Note that the original version of this theorem is only proven for Lipschitz functions in Dey and Wang (2013), but it extends at no cost to functions with modulus of continuity.

A.2 Proof of Theorem 7

Let $|\mathrm{Rips}_\delta(\mathbb{X}_n)|$ denote a geometric realization of the Rips complex built on top of $\mathbb{X}_n$ with parameter δ. Moreover, let $f^{\mathrm{PL}} : |\mathrm{Rips}_\delta(\mathbb{X}_n)| \to \mathbb{R}$ be the piecewise-linear interpolation of f on the simplices of $\mathrm{Rips}_\delta(\mathbb{X}_n)$, whose 1-skeleton is denoted by $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$. Since $(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$ is a metric space, we also consider its Reeb graph $\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|)$, with induced function $f^{\mathrm{PL}}_{\mathrm{R}}$, and its Mapper $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$, with induced function $f^{\mathrm{PL}}_{\mathcal{I}}$. See Figure 8. Then, the following inequalities lead to the result:

$$\begin{aligned}
d_\Delta(\mathrm{R}_f(\mathcal{X}), \mathrm{M}_n) &= d_\Delta\left(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{M}_n, f_{\mathcal{I}})\right) \\
&= d_\Delta\left(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}}), f^{\mathrm{PL}}_{\mathcal{I}})\right) && (14) \\
&\leq d_\Delta\left(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|), f^{\mathrm{PL}}_{\mathrm{R}})\right) \\
&\quad + d_\Delta\left(\mathrm{Dg}(\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|), f^{\mathrm{PL}}_{\mathrm{R}}), \mathrm{Dg}(\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}}), f^{\mathrm{PL}}_{\mathcal{I}})\right) && (15) \\
&\leq 2\omega(\delta) + r. && (16)
\end{aligned}$$

Let us prove every (in)equality:

Equality (14). Let $X_1, X_2 \in \mathbb{X}_n$ such that (X_1, X_2) is an edge of $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$ i.e. $\|X_1 - X_2\| \leq \delta$. Then, according to (4): $|f(X_1) - f(X_2)| < gr$. Hence, there is no $s \in \{1, \ldots, S-1\}$ such that $I_s \cap I_{s+1} \subseteq [\min\{f(X_1), f(X_2)\}, \max\{f(X_1), f(X_2)\}]$. It follows that there are no intersection-crossing edges in $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$. Then, according to Theorem 17, there is a graph isomorphism $i : \mathrm{M}_n = \mathrm{M}_{r,g,\delta}(\mathbb{X}_n, f(\mathbb{Y}_n)) \to \mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$. Since $f_{\mathcal{I}} = f^{\mathrm{PL}}_{\mathcal{I}} \circ i$ by definition of $f_{\mathcal{I}}$ and $f^{\mathrm{PL}}_{\mathcal{I}}$, the equality follows.

Inequality (15). This inequality is just an application of the triangle inequality.

Inequality (16). According to (3), we have $\delta \leq \min\{\mathrm{rch}/4, \rho/4\}$. According to (5), we also have $\delta \geq 4d_{\mathrm{H}}(\mathcal{X}, \mathbb{X}_n)$. Hence, we have

$$d_\Delta(\mathrm{Dg}(\mathrm{R}_f(\mathcal{X}), f_{\mathrm{R}}), \mathrm{Dg}(\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|), f^{\mathrm{PL}}_{\mathrm{R}})) \leq 2\omega(\delta),$$

according to Theorem 19. Moreover, we have

$$d_\Delta(\mathrm{Dg}(\mathrm{R}_{f^{\mathrm{PL}}}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|), f^{\mathrm{PL}}_{\mathrm{R}}), \mathrm{Dg}(\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}}), f^{\mathrm{PL}}_{\mathcal{I}})) \leq r,$$

according to Equation (12).

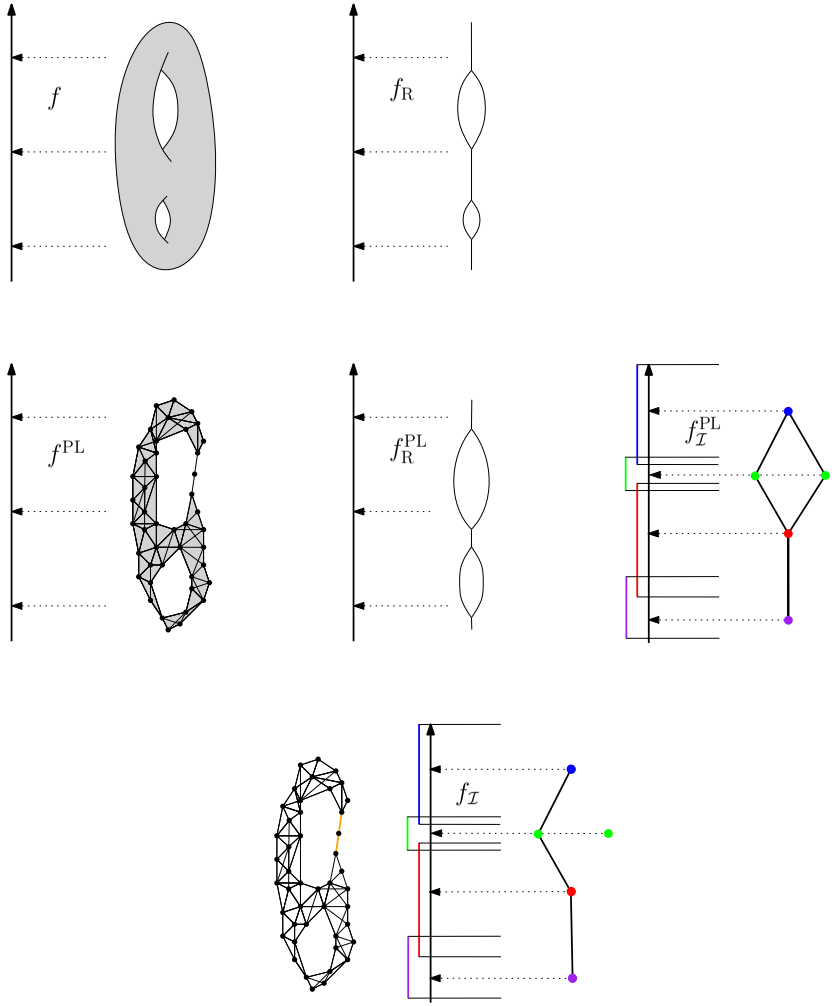


Figure 8: Examples of the function defined on the original space (left column), its induced function defined on the Reeb graph (middle column) and the function defined on the Mapper (right column). Note that the Mapper computed from the geometric realization of the Rips complex (middle row, right) is not isomorphic to the standard Mapper (last row), since there are two intersection-crossing edges in the Rips complex (outlined in orange).

A.3 Proof of Corollary 9

Let $|\mathrm{Rips}_\delta(\mathbb{X}_n)|$ denote a geometric realization of the Rips complex built on top of $\mathbb{X}_n$ with parameter δ. Moreover, let $f^{\mathrm{PL}} : |\mathrm{Rips}_\delta(\mathbb{X}_n)| \to \mathbb{R}$ be the piecewise-linear interpolation of f on the simplices of $\mathrm{Rips}_\delta(\mathbb{X}_n)$, whose 1-skeleton is denoted by $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$. Similarly, let $\hat{f}^{\mathrm{PL}}$ be the piecewise-linear interpolation of $\hat{f}$ on the simplices of $\mathrm{Rips}^1_\delta(\mathbb{X}_n)$. As before, since $(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$ and $(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, \hat{f}^{\mathrm{PL}})$ are metric spaces, we also consider their Mappers $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$ and $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, \hat{f}^{\mathrm{PL}})$. Then, the following inequalities lead to the result:

$$
\begin{aligned}
&d_\Delta(\mathrm{R}_f(\mathcal{X}), \hat{\mathrm{M}}_n) \leq d_\Delta(\mathrm{R}_f(\mathcal{X}), \mathrm{M}_n) + d_\Delta(\mathrm{M}_n, \hat{\mathrm{M}}_n) \text{ by the triangle inequality} \\
&\quad = d_\Delta(\mathrm{R}_f(\mathcal{X}), \mathrm{M}_n) + d_\Delta(\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}}), \mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, \hat{f}^{\mathrm{PL}})) \qquad (17) \\
&\quad \leq r + 2\omega(\delta) + d_\Delta(\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}}), \mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, \hat{f}^{\mathrm{PL}})) \text{ by Theorem 7} \\
&\quad \leq r + 2\omega(\delta) + r + \|f^{\mathrm{PL}} - \hat{f}^{\mathrm{PL}}\|_\infty \text{ by Equation (13)} \\
&\quad = 2r + 2\omega(\delta) + \max\{|f(X) - \hat{f}(X)| \,:\, X \in \mathbb{X}_n\}
\end{aligned}
$$

Let us prove Equality (17). By definition of r, there are no intersection-crossing edges for both f and $\hat{f}$. According to Theorem 17, $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, f^{\mathrm{PL}})$ and M_n are isomorphic and similarly for $\mathrm{M}_{r,g}(|\mathrm{Rips}_\delta(\mathbb{X}_n)|, \hat{f}^{\mathrm{PL}})$ and $\hat{\mathrm{M}}_n$. See also the proof of Equality (14).

A.4 Proof of Proposition 10

We check that not only the topological signature of the Mapper but also the Mapper itself is a measurable object and thus can be seen as an estimator of a target Reeb graph. This problem is more complicated than for the statistical framework of persistence diagram inference, for which the existing stability results give for free that persistence estimators are measurable for adequate sigma algebras.

Let $\bar{\mathbb{R}} = \mathbb{R} \cup \{+\infty\}$ denote the extended real line. Given a fixed integer $n \geq 1$, let $\mathcal{C}_{[n]}$ be the set of abstract simplicial complexes over a fixed set of n vertices. We see $\mathcal{C}_{[n]}$ as a subset of the power set $2^{2^{[n]}}$, where $[n] = \{1, \cdots, n\}$, and we implicitly identify $2^{[n]}$ with the set $[2^n]$ via the map assigning to each subset $\{i_1, \cdots, i_k\}$ the integer $1 + \sum_{j=1}^{k} 2^{i_j - 1}$. Given a fixed parameter $\delta > 0$, we define the application

$$
\Phi_1 : \left\{ \begin{array}{rcl} (\mathbb{R}^D)^n \times \mathbb{R}^n & \to & \mathcal{C}_{[n]} \times \bar{\mathbb{R}}^{2^{[n]}} \\ (\mathbb{X}_n, \mathbb{Y}_n) & \mapsto & (K, f_K) \end{array} \right.
$$

where K is the abstract Rips complex of parameter δ over the n labeled points in $\mathbb{R}^D$, minus the intersection-crossing edges and their cofaces, and where f_K is a function defined by:

$$
f_K : \left\{ \begin{array}{rcl} 2^{[n]} & \to & \bar{\mathbb{R}} \\ \sigma & \mapsto & \left\{ \begin{array}{ll} \max_{i \in \sigma} \mathbb{Y}_i & \text{if } \sigma \in K \\ +\infty & \text{otherwise.} \end{array} \right. \end{array} \right.
$$

The space $(\mathbb{R}^D)^n \times \mathbb{R}^n$ is equipped with the standard topology, denoted by T_1, inherited from $\mathbb{R}^{(D+1)n}$. The space $\mathcal{C}_{[n]} \times \bar{\mathbb{R}}^{2^{[n]}}$ is equipped with the product, denoted by T_2 hereafter, of the discrete topology on $\mathcal{C}_{[n]}$ and the topology induced by the extended distance $d(f,g) = \max\{|f(\sigma) - g(\sigma)| \,:\, \sigma \in 2^{[n]}, f(\sigma) \text{ or } g(\sigma) \neq +\infty\}$ on $\bar{\mathbb{R}}^{2^{[n]}}$. In particular, $K \neq K' \Rightarrow d(f_K, f_{K'}) = +\infty$.

Note that the map $(\mathbb{X}_n, \mathbb{Y}_n) \mapsto K$ is piecewise-constant, with jumps located at the hypersurfaces defined by $\|X_i - X_j\|^2 = \delta^2$ (for combinatorial changes in the Rips complex) or $Y_i = \text{cst} \in \text{End}((r,g))$ (for changes in the set of intersection-crossing edges) in $(\mathbb{R}^D)^n \times \mathbb{R}^n$, where $\text{End}((r,g))$ denotes the set of endpoints of elements of the gomic (r,g). We can then define a finite measurable partition $(\mathcal{C}_\ell)_{\ell \in L}$ of $(\mathbb{R}^D)^n \times \mathbb{R}^n$ whose boundaries are included in these hypersurfaces, and such that $(\mathbb{X}_n, \mathbb{Y}_n) \mapsto K$ is constant over each set $\mathcal{C}_\ell$. As a byproduct, we have that $(\mathbb{X}_n, \mathbb{Y}_n) \mapsto f$ is continuous over each set $\mathcal{C}_\ell$.

We now define the operator

$$\Phi_2 : \left\{ \begin{array}{rcl} \mathcal{C}_{[n]} \times \bar{\mathbb{R}}^{2^{[n]}} & \to & \mathcal{A} \\ (K, f) & \mapsto & (|K|, f^{\text{PL}}) \end{array} \right.$$

where $\mathcal{A}$ denotes the class of topological spaces filtered by Morse-type functions, and where f^{PL} is the piecewise-linear interpolation of f on the geometric realization $|K|$ of K. For a fixed simplicial complex K, the extended persistence diagram of the lower-star filtration induced by f and of the sublevel sets of f^{PL} are identical—see e.g. Morozov (2008), therefore the map Φ_2 is distance-preserving (hence continuous) in the pseudometrics d_Δ on the domain and codomain. Since the topology T_2 on $\mathcal{C}_{[n]} \times \bar{\mathbb{R}}^{2^n}$ is a refinement[2] of the topology induced by d_Δ, the map Φ_2 is also continuous when $\mathcal{C}_{[n]} \times \bar{\mathbb{R}}^{2^{[n]}}$ is equipped with T_2.

Let now $\Phi_3 : \mathcal{A} \to \mathcal{R}$ map each Morse-type pair $(\mathcal{X}, f)$ to its Mapper $\text{M}_f(\mathcal{X}, \mathcal{I})$, where $\mathcal{I} = (r,g)$ is the gomic induced by r and g. Note that, similarly to Φ_1, the map Φ_3 is piecewise-constant, since combinatorial changes in $\text{M}_f(\mathcal{X}, \mathcal{I})$ are located at the regions $\text{Crit}(f) \cap \text{End}(\mathcal{I}) \neq \emptyset$. Hence, Φ_3 is measurable in the pseudometric d_Δ. For more details on Φ_3, we refer the reader to Definition 7.6 in Carrière and Oudot (2017b).

Moreover, $\text{M}_{f^{\text{PL}}}(|K|, \mathcal{I})$ is isomorphic to $\text{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$ by Theorem 17 since all of the intersection-crossing edges were removed in the construction of K. Hence, the map Φ defined by $\Phi = \Phi_3 \circ \Phi_2 \circ \Phi_1$ is a measurable map that sends $(\mathbb{X}_n, \mathbb{Y}_n)$ to $\text{M}_{r,g,\delta}(\mathbb{X}_n, \mathbb{Y}_n)$.

A.5 Proof of Proposition 11

We fix some parameters $a > 0$ and $b \geq 1$. First note that Assumption (4) is always satisfied by definition of r_n. Next, there exists $n_0 \in \mathbb{N}$ such that for any $n \geq n_0$, Assumption (3) is satisfied because $\delta_n \to 0$ and $\omega(\delta_n) \to 0$ as $n \to +\infty$. Moreover, n_0 can be taken the same for all $f \in \bigcup_{\mathbb{P} \in \mathcal{P}(a,b)} \mathcal{F}(\mathbb{P}, \omega)$.

Let $\varepsilon_n = d_{\text{H}}(\mathcal{X}, \mathbb{X}_n)$. Under the (a,b)-standard assumption, it is well known that (see for instance Cuevas and Rodríguez-Casal (2004); Chazal et al. (2015b)):

$$\mathbb{P}\left(\varepsilon_n \geq u\right) \leq \min\left\{1, \frac{4^b}{au^b} \mathrm{e}^{-a\left(\frac{u}{2}\right)^b n}\right\}, \forall u > 0. \tag{18}$$

2. This is because singletons are open balls in the discrete topology, and also because of the stability theorem for persistence diagrams—see Chazal et al. (2016a); Cohen-Steiner et al. (2007)

In particular, regarding the complementary of (5) we have:

$$\mathbb{P}\left(\varepsilon_n > \frac{\delta_n}{4}\right) \leq \min\left\{1, \frac{2^b}{2\log(n)n}\right\}. \tag{19}$$

Recall that $\text{diam}(\mathcal{X}_\mathbb{P}) \leq L$. Let $\bar{C} = \omega(L)$ be a constant that only depends on the parameters of the model. Then, for any $\mathbb{P} \in \mathcal{P}(a,b)$, we have:

$$\sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\text{R}_f(\mathcal{X}_\mathbb{P}), \text{M}_n\right) \leq \bar{C}. \tag{20}$$

For $n \geq n_0$, we have :

$$\begin{aligned} \sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\text{R}_f(\mathcal{X}_\mathbb{P}), \text{M}_n\right) = & \sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\text{R}_f(\mathcal{X}_\mathbb{P}), \text{M}_n\right) \mathbb{I}_{\varepsilon_n > \delta_n/4} \\ & + \sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\text{R}_f(\mathcal{X}_\mathbb{P}), \text{M}_n\right) \mathbb{I}_{\varepsilon_n \leq \delta_n/4} \end{aligned}$$

and thus

$$\begin{aligned} \mathbb{E}\left[\sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\text{R}_f(\mathcal{X}_\mathbb{P}), \text{M}_n\right)\right] & \leq \bar{C}\mathbb{P}\left(\varepsilon_n > \frac{\delta_n}{4}\right) + r_n + 2\omega(\delta_n) \\ & \leq \bar{C}\min\left\{1, \frac{2^b}{2\log(n)n}\right\} + \left(\frac{1+2g}{g}\right)\omega(\delta_n) \end{aligned} \tag{21}$$

where we have used (20), Theorem 7 and the fact that $V_n(\delta_n)^+$ can be chosen less or equal to $\omega(\delta_n)$. For n large enough, the first term in (21) is of the order of δ_n^b, which can be upper bounded by δ_n and thus by $\omega(\delta_n)$ (up to a constant) since $\omega(\delta)/\delta$ is non-increasing. Since $\frac{1+2g}{g} < 6$ because $\frac{1}{3} < g < \frac{1}{2}$, we get that the risk is bounded by $\omega(\delta_n)$ for $n \geq n_0$ up to a constant that only depends on the parameters of the model. The same inequality is of course valid for any n by taking a larger constant, because n_0 itself only depends on the parameters of the model.

A.6 Proof of Proposition 12

The proof follows closely Section B.2 in Chazal et al. (2015b). Let $\mathcal{X}_0 = [0, a^{-1/b}] \subset \mathbb{R}^D$. Obviously, $\mathcal{X}_0$ is a smooth and compact submanifold of $\mathbb{R}^D$. Let $\mathcal{U}(\mathcal{X}_0)$ be the uniform measure on $\mathcal{X}_0$. Let $\mathcal{P}_{a,b,\mathcal{X}_0}$ denote the set of (a,b)-standard measures whose support is included in $\mathcal{X}_0$. Let $x_0 = 0 \in \mathcal{X}_0$ and $\{x_n\}_{n \in \mathbb{N}^*} \in \mathcal{X}_0^\mathbb{N}$ such that $\|x_n - x_0\| = (an)^{-1/b}$. Now, let

$$f_0 : \left\{ \begin{array}{ll} \mathcal{X}_0 & \to \mathbb{R} \\ x & \mapsto \omega(\|x - x_0\|) \end{array} \right.$$

By definition, we have $f_0 \in \mathcal{F}(\mathcal{U}(\mathcal{X}_0), \omega)$ because $\text{Dg}(\mathcal{X}_0, f_0) = \{(0, \omega(a^{-1/b}))\}$ since f_0 is increasing by definition of ω. Finally, given any measure $\mathbb{P} \in \mathcal{P}_{a,b,\mathcal{X}_0}$, we let $\theta_0(\mathbb{P}) =$

$\mathrm{R}_{f_0|_{\mathcal{X}_\mathbb{P}}}(\mathcal{X}_\mathbb{P})$. Then, we have:

$$\begin{aligned}
&\sup_{\mathbb{P}\in\mathcal{P}_{a,b}} \mathbb{E}\left[\sup_{f\in\mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}),\hat{\mathrm{R}}_n\right)\right] \\
&\quad\geq \sup_{\mathbb{P}\in\mathcal{P}_{a,b,\mathcal{X}_0}} \mathbb{E}\left[\sup_{f\in\mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}),\hat{\mathrm{R}}_n\right)\right] \\
&\quad\geq \sup_{\mathbb{P}\in\mathcal{P}_{a,b,\mathcal{X}_0}} \mathbb{E}\left[d_\Delta\left(\mathrm{R}_{f_0|_{\mathcal{X}_\mathbb{P}}}(\mathcal{X}_\mathbb{P}),\hat{\mathrm{R}}_n\right)\right] = \sup_{\mathbb{P}\in\mathcal{P}_{a,b,\mathcal{X}_0}} \mathbb{E}\left[\rho\left(\theta_0(\mathbb{P}),\hat{\mathrm{R}}_n\right)\right],
\end{aligned}$$

where $\rho = d_\Delta$. For any $n\in\mathbb{N}^*$, we let $\mathbb{P}_{0,n} = \delta_{x_0}$ be the Dirac measure on x_0 and $\mathbb{P}_{1,n} = (1-\frac{1}{n})\mathbb{P}_{0,n} + \frac{1}{n}\mathcal{U}([x_0,x_n])$. As a Dirac measure, $\mathbb{P}_{0,n}$ is obviously in $\mathcal{P}_{a,b,\mathcal{X}_0}$. We now check that $\mathbb{P}_{1,n}\in\mathcal{P}_{a,b,\mathcal{X}_0}$.

- Let us study $\mathbb{P}_{1,n}(B(x_0,r))$.

  Assume $r\leq(an)^{-1/b}$. Then

  $$\begin{aligned}
  \mathbb{P}_{1,n}(B(x_0,r)) = 1-\frac{1}{n}+\frac{1}{n}\frac{r}{(an)^{-1/b}} &\geq \left(1-\frac{1}{n}+\frac{1}{n}\right)\left(\frac{r}{(an)^{-1/b}}\right)^b \\
  &\geq \left(\frac{1}{2}+\frac{1}{n}\right)anr^b \geq ar^b.
  \end{aligned}$$

  Assume $r>(an)^{-1/b}$. Then

  $$\mathbb{P}_{1,n}(B(x_0,r)) = 1 \geq \min\{ar^b\}.$$

- Let us study $\mathbb{P}_{1,n}(B(x_n,r))$. Assume $r\leq(an)^{-1/b}$. Then

  $$\mathbb{P}_{1,n}(B(x_n,r)) = \frac{1}{n}\frac{r}{(an)^{-1/b}} \geq \frac{1}{n}\left(\frac{r}{(an)^{-1/b}}\right)^b = ar^b.$$

  Assume $r>(an)^{-1/b}$. Then

  $$\mathbb{P}_{1,n}(B(x_n,r)) = 1 \geq \min\{ar^b\}.$$

- Let us study $\mathbb{P}_{1,n}(B(x,r))$, where $x\in(x_0,x_n)$. Assume $r\leq x$. Then

  $$\mathbb{P}_{1,n}(B(x,r)) \geq \frac{1}{n}\frac{r}{(ab)^{-1/b}} \geq ar^b \text{ (see previous case).}$$

  Assume $r>x$. Then $\mathbb{P}_{1,n}(B(x,r)) = 1-\frac{1}{n}+\frac{1}{n}\frac{\left(x+\min\{r,\ (an)^{-1/b}-x\}\right)}{(an)^{-1/b}}$.

  If $\min\{r,\ (an)^{-1/b}-x\} = r$, then we have

  $$\mathbb{P}_{1,n}(B(x,r)) \geq 1-\frac{1}{n}+\frac{1}{n}\frac{r}{(ab)^{-1/b}} \geq ar^b \text{ (see previous case).}$$

  Otherwise, we have

  $$\mathbb{P}_{1,n}(B(x,r)) = 1 \geq \min\{ar^b\}.$$

Thus $\mathbb{P}_{1,n}$ is in $\mathcal{P}_{a,b,\mathcal{X}_0}$ as well. Hence, we apply Le Cam's lemma (see Section B) to get:

$$\sup_{\mathbb{P}\in\mathcal{P}_{a,b,\mathcal{X}_0}} \mathbb{E}\left[\rho\left(\theta_0(\mathbb{P}),\hat{\mathrm{R}}_n\right)\right] \geq \frac{1}{8}\rho(\theta_0(\mathbb{P}_{0,n}),\theta_0(\mathbb{P}_{1,n}))\left[1-\mathrm{TV}(\mathbb{P}_{0,n},\mathbb{P}_{1,n})\right]^{2n}.$$

By definition, we have:

$$\rho(\theta_0(\mathbb{P}_{0,n}),\theta_0(\mathbb{P}_{1,n})) = d_\Delta\left(\mathrm{R}_{f_0|_{\{x_0\}}}(\{x_0\}),\mathrm{R}_{f_0|_{[x_0,x_n]}}(\mathcal{U}[x_0,x_n])\right).$$

Since $\mathrm{Dg}\left(\mathrm{R}_{f_0|_{\{x_0\}}}(\{x_0\})\right) = \{(0,0)\}$ and $\mathrm{Dg}\left(\mathrm{R}_{f_0|_{[x_0,x_n]}}(\mathcal{U}[x_0,x_n])\right) = \{(f(x_0),f(x_n))\}$ because f_0 is increasing by definition of ω, it follows that

$$\rho(\theta_0(\mathbb{P}_{0,n}),\theta_0(\mathbb{P}_{1,n})) = \frac{1}{2}|f(x_n)-f(x_0)| = \frac{1}{2}\omega\left((an)^{-1/b}\right).$$

It remains to compute $\mathrm{TV}(\mathbb{P}_{0,n},\mathbb{P}_{1,n}) = \left|1-\left(1-\frac{1}{n}\right)\right|+\frac{1}{n}(an)^{-1/b} = \frac{1}{n}+o\left(\frac{1}{n}\right)$. The proposition follows then from the fact that $\left[1-\mathrm{TV}(\mathbb{P}_{0,n},\mathbb{P}_{1,n})\right]^{2n}\to\mathrm{e}^{-2}$.

A.7 Proof of Proposition 13

Let $\mathbb{P}\in\mathcal{P}_{a,b}$ and ω a modulus of continuity for f. Using the same notation as in the previous section, we have

$$\begin{aligned}
\mathbb{P}\left(\delta_n\geq u\right) &\leq \mathbb{P}\left(d_\mathrm{H}(\mathbb{X}_n,\mathcal{X}_\mathbb{P})\geq\frac{u}{2}\right)+\mathbb{P}\left(d_\mathrm{H}(\mathbb{X}_n^{s_n},\mathcal{X}_\mathbb{P})\geq\frac{u}{2}\right) \\
&\leq \mathbb{P}\left(\varepsilon_n\geq\frac{u}{2}\right)+\mathbb{P}\left(\varepsilon_{s_n}\geq\frac{u}{2}\right). \qquad (22)
\end{aligned}$$

Note that for any $f\in\mathcal{F}(\mathbb{P},\omega)$, according to (6) and (20)

$$d_\Delta\left(\mathrm{R}_f(\mathcal{X}_\mathbb{P}),\mathrm{M}_n\right) \leq \left[r+2\omega(\delta)\right]\mathbb{I}_{\Omega_n}+\bar{C}\,\mathbb{I}_{\Omega_n^c} \qquad (23)$$

where Ω_n is the event defined by

$$\Omega_n = \{4\delta_n\leq\min\{\kappa,\rho\}\}\cap\{4\varepsilon_n\leq\delta_n\}.$$

This gives

$$\begin{aligned}
\mathbb{E}\left[\sup_{f\in\mathcal{F}(P,\omega)} d_\Delta\left(\mathrm{M}_n,\mathrm{R}_f(\mathcal{X})\right)\right] \leq &\underbrace{\int_0^{\bar{C}}\mathbb{P}\left(\omega(\delta_n)\geq\frac{g}{1+2g}\alpha\right)\mathrm{d}\alpha}_{(A)}+\underbrace{\bar{C}\mathbb{P}\left(\varepsilon_n\geq\frac{\delta_n}{4}\right)}_{(B)} \\
&+\underbrace{\bar{C}\mathbb{P}\left(\delta_n\geq\min\left\{\frac{\kappa}{4},\frac{\rho}{4}\right\}\right)}_{(D)}.
\end{aligned}$$

Let us bound the three terms (A), (B) and (C).

- **Term** (C). It can be bounded using (22) then (18).

- **Term** (B). Let $t_n = 2\left(\frac{2\log(n)}{an}\right)^{1/b}$ and $A_n = \{\varepsilon_n < t_n\}$. We first prove that $\delta_n \geq 4\varepsilon_n$ on the event A_n, for n large enough. We follow the lines of the proof of Theorem 3 in Section 6 in Fasy et al. (2014).

  Let q_n be the t_n-*packing number* of $\mathcal{X}_{\mathbb{P}}$, i.e. the maximal number of Euclidean balls $B(X, t_n) \cap \mathcal{X}_{\mathbb{P}}$, where $X \in \mathcal{X}_{\mathbb{P}}$, that can be packed into $\mathcal{X}_{\mathbb{P}}$ without overlap. It is known (see for instance Lemma 17 in Fasy et al. (2014)) that $q_n = \Theta(t_n^{-d})$, where d is the (intrinsic) dimension of $\mathcal{X}_{\mathbb{P}}$. Let $\mathrm{Pack}_n = \{c_1, \cdots, c_{q_n}\}$ be a corresponding packing set, i.e. the set of centers of a family of balls of radius t_n whose cardinality achieves the packing number q_n. Note that $d_{\mathrm{H}}(\mathrm{Pack}_n, \mathcal{X}_{\mathbb{P}}) \leq 2t_n$. Indeed, for any $X \in \mathcal{X}_{\mathbb{P}}$, there must exist $c \in \mathrm{Pack}_n$ such that $\|X - c\| \leq 2t_n$, otherwise X could be added to Pack_n, contradicting the fact that Pack_n is maximal. By contradiction, assume $\epsilon_n < t_n$ and $\delta_n \leq 4\epsilon_n$. Then:

$$\begin{aligned} d_{\mathrm{H}}(\mathbb{X}_n^{s_n}, \mathrm{Pack}_n) &\leq d_{\mathrm{H}}(\mathbb{X}_n^{s_n}, \mathbb{X}_n) + d_{\mathrm{H}}(\mathbb{X}_n, \mathcal{X}_{\mathbb{P}}) + d_{\mathrm{H}}(\mathcal{X}_{\mathbb{P}}, \mathrm{Pack}_n) \\ &\leq 5 d_{\mathrm{H}}(\mathbb{X}_n, \mathcal{X}_{\mathbb{P}}) + 2t_n \leq 7t_n. \end{aligned}$$

  Now, one has $\frac{s_n}{q_n} = \Theta\left(\frac{n^{1-b/d}}{\log(n)^{1-b/d+\beta}}\right)$. Since $b \geq D \geq d$ by definition, it follows that $s_n = o(q_n)$. In particular, this means that $d_{\mathrm{H}}(\mathbb{X}_n^{s_n}, \mathrm{Pack}_n) > 7t_n$ for n large enough, which yields a contradiction.

  Hence, one has $\delta_n \geq 4\varepsilon_n$ on the event A_n. Thus, one has:

$$\mathbb{P}\left(\varepsilon_n \geq \frac{\delta_n}{4}\right) \leq \underbrace{\mathbb{P}\left(\varepsilon_n \geq \frac{\delta_n}{4} \mid A_n\right)}_{=0} \mathbb{P}(A_n) + \mathbb{P}(A_n^c) = \mathbb{P}(A_n^c).$$

  Finally, the probability of A_n^c is bounded with (18):

$$\mathbb{P}(A_n^c) \leq \frac{2^b}{2\log(n)n}.$$

- **Term** (A). This is the dominating term. First, note that since ω is increasing, one has for all $u > 0$:

$$\mathbb{P}(\omega(\delta_n) \geq u) = \mathbb{P}\left(\delta_n \geq \omega^{-1}(u)\right). \tag{24}$$

  Then, using (22) and (24), we have:

$$(A) \leq \int_0^{\bar{C}} \mathbb{P}\left(\varepsilon_n \geq \frac{1}{2}\omega^{-1}\left(\frac{g\alpha}{1+2g}\right)\right) \mathrm{d}\alpha + \int_0^{\bar{C}} \mathbb{P}\left(\varepsilon_{s_n} \geq \frac{1}{2}\omega^{-1}\left(\frac{g\alpha}{1+2g}\right)\right) \mathrm{d}\alpha.$$

  We only bound the first integral, but the analysis extends verbatim to the second integral when replacing n by s_n. Let

$$\alpha_n = \frac{1+2g}{g}\omega\left[\left(\frac{4^b \log(n)}{an}\right)^{1/b}\right].$$

Since $x \mapsto \frac{\omega(x)}{x}$ is non-increasing, it follows that $x \mapsto \frac{\omega^{-1}(x)}{x}$ is non-decreasing, and

$$\omega^{-1}(x) \geq \frac{x}{y}\omega^{-1}(y), \ \forall x \geq y > 0. \tag{25}$$

Taking inspiration from Section B.2 in Chazal et al. (2015b) and using (18), we have the following inequalities:

$$\begin{aligned} &\int_0^{\bar{C}} \mathbb{P}\left(\varepsilon_n \geq \frac{1}{2}\omega^{-1}\left(\frac{g\alpha}{1+2g}\right)\right) \mathrm{d}\alpha \\ &\leq \alpha_n + \frac{8^b}{a}\int_{\alpha_n}^{\bar{C}} \frac{1}{\omega^{-1}\left(\frac{g\alpha}{1+2g}\right)^b} \exp\left[-\frac{an}{4^b}\omega^{-1}\left(\frac{g\alpha}{1+2g}\right)^b\right] \mathrm{d}\alpha \\ &\leq \alpha_n + \frac{8^b}{a}\int_{\alpha_n}^{\bar{C}} \frac{\alpha_n^b}{\left[\alpha\omega^{-1}\left(\frac{g\alpha_n}{1+2g}\right)\right]^b} \exp\left[-\frac{an\alpha^b}{(4\alpha_n)^b}\omega^{-1}\left(\frac{g\alpha_n}{1+2g}\right)^b\right] \mathrm{d}\alpha \\ &\leq \alpha_n + \alpha_n \frac{2^b 4n^{1-1/b}}{ba^{1/b}\omega^{-1}\left(\frac{g\alpha_n}{1+2g}\right)} \int_{u \geq \frac{an}{4^b}\omega^{-1}\left(\frac{g\alpha_n}{1+2g}\right)^b} u^{1/b-2}\mathrm{e}^{-u}\mathrm{d}u \\ &= \alpha_n + \alpha_n \frac{2^b n}{b\log(n)^{1/b}} \int_{u \geq \log(n)} u^{1/b-2}\mathrm{e}^{-u}\mathrm{d}u \leq \left(1 + \frac{2^b}{b\log(n)^2}\right)\alpha_n \text{ since } b \geq 1 \\ &\leq C(b)\alpha_n, \end{aligned}$$

where we used (25) with $x = \frac{g\alpha}{1+2g}$ and $y = \frac{g\alpha_n}{1+2g}$ for the second inequality. The constant $C(b)$ only depends on b.

Hence, since $\frac{1+2g}{g} < 6$, there exist constants $K, K' > 0$ that depend only of the geometric parameters of the model such that:

$$(A) \leq K\omega\left(\frac{K'\log(s_n)}{s_n}\right)^{1/b}.$$

Final bound. Since $s_n = n\log(n)^{-(1+\beta)}$, by gathering all four terms, there exist constants $C, C' > 0$ such that:

$$\mathbb{E}\left[\sup_{f \in \mathcal{F}(\mathbb{P},\omega)} d_\Delta\left(\mathrm{R}_f(\mathcal{X}_{\mathbb{P}}), \mathrm{M}_n\right)\right] \leq C\omega\left(\frac{C'\log(n)^{2+\beta}}{n}\right)^{1/b}.$$

A.8 Proof of Proposition 15

We have the following bound by using (23) in the proof of Proposition 13:

$$\begin{aligned} &\mathbb{P}\left(d_\Delta(, \mathrm{R}_f(\mathcal{X}_{\mathbb{P}}), \mathrm{M}_n) \geq \eta\right) \\ &\leq \mathbb{P}\left(\omega(\delta_n) \geq \frac{g}{1+2g}\eta\right) + \mathbb{P}\left(\varepsilon_n \geq \frac{\delta_n}{4}\right) + \mathbb{P}\left(\delta_n \geq \min\left\{\frac{\kappa}{4}, \frac{\rho}{4}\right\}\right) \\ &\leq \mathbb{P}\left(\varepsilon_n \geq \frac{1}{2}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + \mathbb{P}\left(\varepsilon_{s_n} \geq \frac{1}{2}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + o\left(\frac{1}{n\log(n)}\right). \end{aligned}$$

Following the lines of Section 6 in Fasy et al. (2014), subsampling approximations give

$$\mathbb{P}\left(\varepsilon_n \geq \frac{1}{2}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) \leq L_n\left(\frac{1}{4}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + o\left(\frac{s_n}{n}\right)^{1/4},$$

and

$$\mathbb{P}\left(\varepsilon_{s_n} \geq \frac{1}{2}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) \leq F_n\left(\frac{1}{4}\omega^{-1}\left(\frac{g}{1+2g}\eta\right)\right) + o\left(\frac{s_n^2}{s_n}\right)^{1/4}.$$

The result follows by taking $s_n = n\log(n)^{-(1+\beta)}$.

Appendix B. Le Cam's lemma

The version of Le Cam's lemma given below is from Yu (1997) (see also Genovese et al., 2012b). Recall that the total variation distance between two distributions $\mathbb{P}_0$ and $\mathbb{P}_1$ on a measured space $(\mathcal{X}, \mathcal{B})$ is defined by

$$\mathrm{TV}(\mathbb{P}_0, \mathbb{P}_1) = \sup_{B\in\mathcal{B}} |\mathbb{P}_0(B) - \mathbb{P}_1(B)|.$$

Moreover, if $\mathbb{P}_0$ and $\mathbb{P}_1$ have densities p_0 and p_1 for the same measure λ on $\mathcal{X}$, then

$$\mathrm{TV}(\mathbb{P}_0, \mathbb{P}_1) = \frac{1}{2}\ell_1(p_0, p_1) = \int_{\mathcal{X}} |p_0 - p_1| d\lambda.$$

Lemma 20 *Let $\mathcal{P}$ be a set of distributions. For $\mathbb{P} \in \mathcal{P}$, let $\theta(\mathbb{P})$ take values in a pseudo-metric space $(\mathbb{X}, \rho)$. Let $\mathbb{P}_0$ and $\mathbb{P}_1$ in $\mathcal{P}$ be any pair of distributions. Let $X_1, \ldots, X_n$ be drawn i.i.d. from some $\mathbb{P} \in \mathcal{P}$. Let $\hat{\theta} = \hat{\theta}(X_1, \ldots, X_n)$ be any estimator of $\theta(\mathbb{P})$, then*

$$\sup_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}^n}\left[\rho(\theta, \hat{\theta})\right] \geq \frac{1}{8}\rho\left(\theta(\mathbb{P}_0), \theta(\mathbb{P}_1)\right)\left[1 - \mathrm{TV}(\mathbb{P}_0, \mathbb{P}_1)\right]^{2n}.$$

Appendix C. Extended Persistence

Let f be a real-valued function on a topological space $\mathcal{X}$. Given an interval I and a scalar value $t \in \mathbb{R}$, we let $\mathcal{X}^I$ denote $f^{-1}(I)$ and $\mathcal{X}^t = f^{-1}(t)$. The family $\{\mathcal{X}^{(-\infty,\alpha]}\}_{\alpha\in\mathbb{R}}$ of sublevel sets of f defines a *filtration*, that is, it is nested with respect to the inclusion: $\mathcal{X}^{(-\infty,\alpha]} \subseteq \mathcal{X}^{(-\infty,\beta]}$ for all $\alpha \leq \beta \in \mathbb{R}$. The family $\{\mathcal{X}^{[\alpha,+\infty)}\}_{\alpha\in\mathbb{R}}$ of superlevel sets of f is also nested but in the opposite direction: $\mathcal{X}^{[\alpha,+\infty)} \supseteq \mathcal{X}^{[\beta,+\infty)}$ for all $\alpha \leq \beta \in \mathbb{R}$. We can turn it into a filtration by reversing the real line. Specifically, let $\mathbb{R}^{\mathrm{op}} = \{\tilde{x} \mid x \in \mathbb{R}\}$, ordered by $\tilde{x} \leq \tilde{y} \Leftrightarrow x \geq y$. We index the family of superlevel sets by $\mathbb{R}^{\mathrm{op}}$, so now we have a filtration: $\{\mathcal{X}^{[\tilde{\alpha},+\infty)}\}_{\tilde{\alpha}\in\mathbb{R}^{\mathrm{op}}}$, with $\mathcal{X}^{[\tilde{\alpha},+\infty)} \subseteq \mathcal{X}^{[\tilde{\beta},+\infty)}$ for all $\tilde{\alpha} \leq \tilde{\beta} \in \mathbb{R}^{\mathrm{op}}$.

Extended persistence connects the two filtrations at infinity as follows. Replace each superlevel set $\mathcal{X}^{[\tilde{\alpha},+\infty)}$ by the pair of spaces $(\mathcal{X}, \mathcal{X}^{[\tilde{\alpha},+\infty)})$ in the second filtration. This maintains the filtration property since we have $(\mathcal{X}, \mathcal{X}^{[\tilde{\alpha},+\infty)}) \subseteq (\mathcal{X}, \mathcal{X}^{[\tilde{\beta},+\infty)})$ for all $\tilde{\alpha} \leq \tilde{\beta} \in \mathbb{R}^{\mathrm{op}}$. Then, let $\mathbb{R}_{\mathrm{Ext}} = \mathbb{R} \sqcup \{+\infty\} \sqcup \mathbb{R}^{\mathrm{op}}$, where the order is completed by $\alpha < +\infty < \tilde{\beta}$ for all $\alpha \in \mathbb{R}$ and $\tilde{\beta} \in \mathbb{R}^{\mathrm{op}}$. This poset is isomorphic to $(\mathbb{R}, \leq)$. Finally, define the *extended filtration* of f over $\mathbb{R}_{\mathrm{Ext}}$ by:

$$\begin{aligned} F_\alpha &= \mathcal{X}^{(-\infty,\alpha]} && \text{for } \alpha \in \mathbb{R} \\ F_{+\infty} &= \mathcal{X} \equiv (\mathcal{X}, \emptyset) \\ F_{\tilde{\alpha}} &= (\mathcal{X}, \mathcal{X}^{[\tilde{\alpha},+\infty)}) && \text{for } \tilde{\alpha} \in \mathbb{R}^{\mathrm{op}}, \end{aligned}$$

where we have identified the space $\mathcal{X}$ with the pair of spaces $(\mathcal{X}, \emptyset)$. This is a well-defined filtration since we have $\mathcal{X}^{(-\infty,\alpha]} \subseteq \mathcal{X} \equiv (\mathcal{X}, \emptyset) \subseteq (\mathcal{X}, \mathcal{X}^{[\tilde{\beta},+\infty)})$ for all $\alpha \in \mathbb{R}$ and $\tilde{\beta} \in \mathbb{R}^{\mathrm{op}}$. The subfamily $\{F_\alpha\}_{\alpha\in\mathbb{R}}$ is called the *ordinary* part of the filtration, and the subfamily $\{F_{\tilde{\alpha}}\}_{\tilde{\alpha}\in\mathbb{R}^{\mathrm{op}}}$ is called the *relative* part. See Figure 9 for an illustration.

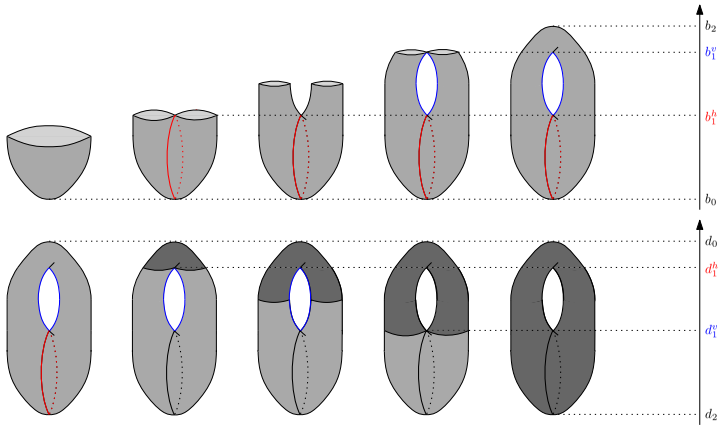


Figure 9: The extended filtration of the height function on a torus. The upper row displays the ordinary part of the filtration while the lower row displays the relative part. The red and blue cycles both correspond to extended points in dimension 1. The point corresponding to the red cycle is located above the diagonal ($d_1^h > b_1^h$), while the point corresponding to the blue cycle is located below the diagonal ($d_1^v > b_1^v$).

Applying the homology functor H_* to this filtration gives the so-called *extended persistence module* $\mathbb{V}$ of f:

$$\begin{aligned} V_\alpha &= H_*(F_\alpha) = H_*(\mathcal{X}^{(-\infty,\alpha]}) && \text{for } \alpha \in \mathbb{R} \\ V_{+\infty} &= H_*(F_{+\infty}) = H_*(\mathcal{X}) \cong H_*(\mathcal{X}, \emptyset) \\ V_{\tilde{\alpha}} &= H_*(F_{\tilde{\alpha}}) = H_*(\mathcal{X}, \mathcal{X}^{[\tilde{\alpha},+\infty)}) && \text{for } \tilde{\alpha} \in \mathbb{R}^{\mathrm{op}}, \end{aligned}$$

and where the linear maps between the spaces are induced by the inclusions in the extended filtration.

For Morse-type functions, the extended persistence module can be decomposed as a finite direct sum of half-open *interval modules*—see e.g. Chazal et al. (2016a):

$$\mathbb{V} \simeq \bigoplus_{k=1}^{n} \mathbb{I}[b_k, d_k),$$

where each summand $\mathbb{I}[b_k, d_k)$ is made of copies of the field of coefficients at each index $\alpha \in [b_k, d_k)$, and of copies of the zero space elsewhere, the maps between copies of the field being identities. Each summand represents the lifespan of a *homological feature* (connected component, hole, void, etc.) within the filtration. More precisely, the *birth time* b_k and *death time* d_k of the feature are given by the endpoints of the interval. Then, a convenient way to represent the structure of the module is to plot each interval in the decomposition as a point in the extended plane, whose coordinates are given by the endpoints. Such a plot is called the *extended persistence diagram* of f, denoted $\mathrm{Dg}(f)$. The distinction between ordinary and relative parts of the filtration allows to classify the points in $\mathrm{Dg}(f)$ in the following way:

- points whose coordinates both belong to $\mathbb{R}$ are called *ordinary* points; they correspond to homological features being born and then dying in the ordinary part of the filtration;
- points whose coordinates both belong to $\mathbb{R}^{\mathrm{op}}$ are called *relative* points; they correspond to homological features being born and then dying in the relative part of the filtration;
- points whose abscissa belongs to $\mathbb{R}$ and whose ordinate belongs to $\mathbb{R}^{\mathrm{op}}$ are called *extended* points; they correspond to homological features being born in the ordinary part and then dying in the relative part of the filtration.

Note that ordinary points lie strictly above the diagonal $\Delta = \{(x, x) \mid x \in \mathbb{R}\}$ and relative points lie strictly below Δ, while extended points can be located anywhere, including on Δ, e.g. cc that lie inside a single critical level. It is common to decompose $\mathrm{Dg}(f)$ according to this classification:

$$\mathrm{Dg}(f) = \mathrm{Ord}(f) \sqcup \mathrm{Rel}(f) \sqcup \mathrm{Ext}^{+}(f) \sqcup \mathrm{Ext}^{-}(f),$$

where by convention $\mathrm{Ext}^{+}(f)$ includes the extended points located on the diagonal Δ.

References

Aravindakshan Babu. Zigzag Coarsenings, Mapper Stability and Gene-network Analyses. PhD Thesis, 2013.

Ulrich Bauer, Xiaoyin Ge, and Yusu Wang. Measuring Distance Between Reeb Graphs. In *Proceedings of the 30th Symposium on Computational Geometry*, pages 464–473, 2014.

Ulrich Bauer, Elizabeth Munch, and Yusu Wang. Strong Equivalence of the Interleaving and Functional Distortion Metrics for Reeb Graphs. In *Proceedings of the 31st Symposium on Computational Geometry*, 2015.

Silvia Biasotti, Daniela Giorgi, Michela Spagnuolo, and Bianca Falcidieno. Reeb Graphs for Shape Analysis and Applications. *Theoretical Computer Science*, 392:5–22, 2008.

Gérard Biau and André Mas. PCA-Kernel estimation. *Statistics and Risk Modeling with Applications in Finance and Insurance*, 29(1):19–46, 2012.

Gilles Blanchard, Olivier Bousquet, and Laurent Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2-3):259–294, 2007.

Mickaël Buchet, Frédéric Chazal, Steve Oudot, and Donald Sheehy. Efficient and Robust Persistent Homology for Measures. In *Proceedings of the 26th Symposium on Discrete Algorithms*, pages 168–180, 2015.

Mathieu Carrière. Cover complex. In *GUDHI User and Reference Manual*. GUDHI Editorial Board, 2017. URL `http://gudhi.gforge.inria.fr/doc/latest/group__cover_complex.html`.

Mathieu Carrière and Steve Oudot. Local Equivalence and Induced Metrics for Reeb Graphs. In *Proceedings of the 33rd Symposium on Computational Geometry*, 2017a.

Mathieu Carrière and Steve Oudot. Structure and Stability of the 1-Dimensional Mapper. *Foundations of Computational Mathematics*, 2017b.

Frédéric Chazal, David Cohen-Steiner, and Quentin Mérigot. Geometric Inference for Probability Measures. *Foundations of Computational Mathematics*, 11(6):733–751, 2011.

Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. On the Bootstrap for Persistence Diagrams and Landscapes. *Modeling and Analysis of Information Systems*, 20(6):111–120, 2013.

Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Robust topological inference: distance to a measure and kernel distance. *CoRR*, abs/1412.7197, 2014. Accepted for publication in Journal of Machine Learning Research.

Frédéric Chazal, Brittany Fasy, Fabrizio Lecci, Bertrand Michel, Alessandro Rinaldo, and Larry Wasserman. Subsampling Methods for Persistent Homology. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 2143–2151, 2015a.

Frédéric Chazal, Marc Glisse, Catherine Labruère, and Bertrand Michel. Convergence rates for persistence diagram estimation in topological data analysis. *Journal of Machine Learning Research*, 16:3603–3635, 2015b.

Frédéric Chazal, Vin de Silva, Marc Glisse, and Steve Oudot. *The Structure and Stability of Persistence Modules*. Springer, 2016a.

Frédéric Chazal, Pascal Massart, and Bertrand Michel. Rates of convergence for robust geometric inference. *Electronic Journal of Statistics*, 10(2):2243–2286, 2016b.

X. Chen, A. Golovinskiy, and T. Funkhouser. A Benchmark for 3D Mesh Segmentation. *ACM Transactions on Graphics*, 28(3):1–12, 2009.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of Persistence Diagrams. *Discrete and Computational Geometry*, 37(1):103–120, 2007.

David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Extending persistence using Poincaré and Lefschetz duality. *Foundation of Computational Mathematics*, 9(1):79–103, 2009.

Antonio Cuevas. Set estimation: another bridge between statistics and geometry. *Boletín de Estadística e Investigación Operativa*, 25(2):71–85, 2009.

Antonio Cuevas and Alberto Rodríguez-Casal. On boundary estimation. *Advances in Applied Probability*, pages 340–354, 2004.

Vin de Silva, Elizabeth Munch, and Amit Patel. Categorified Reeb Graphs. *Discrete and Computational Geometry*, 55:854–906, 2016.

Ronald DeVore and George Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993.

Tamal Dey and Yusu Wang. Reeb Graphs: Approximation and Persistence. *Discrete and Computational Geometry*, 49(1):46–73, 2013.

Tamal Dey, Facundo Mémoli, and Yusu Wang. Topological Analysis of Nerves, Reeb Spaces, Mappers, and Multiscale Mappers. In *Proceedings of the 33rd Symposium on Computational Geometry*, volume 77, pages 36:1–36:16, 2017.

Barbara di Fabio and Claudia Landi. The Edit Distance for Reeb Graphs of Surfaces. *Discrete and Computational Geometry*, 55(2):423–461, 2016.

Herbert Edelsbrunner and John Harer. *Computational Topology: an introduction*. AMS Bookstore, 2010.

Brittany Fasy, Fabrizio Lecci, Alessandro Rinaldo, Larry Wasserman, Sivaraman Balakrishnan, and Aarti Singh. Confidence Sets for Persistence Diagrams. *The Annals of Statistics*, 42(6):2301–2339, 2014.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The Elements of Statistical Learning*. Springer series in statistics Springer, Berlin, 2001.

Christopher Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Manifold estimation and singular deconvolution under Hausdorff loss. *The Annals of Statistics*, 40:941–963, 2012a.

Christopher Genovese, Marco Perone-Pacifico, Isabella Verdinelli, and Larry Wasserman. Minimax Manifold Estimation. *Journal of Machine Learning Research*, 13:1263–1291, 2012b.

TS. Hinks, X. Zhou, KJ. Staples, BD. Dimitrov, A. Manta, T. Petrossian, P. Lum, CG. Smith, JA. Ward, PH Howarth, AF. Walls, SD. Gadola, and R. Djukanovic. Innate and adaptive t cells in asthmatic patients: Relationship to severity and disease mechanisms. *Journal of Allergy and Clinical Immunology*, 136(2):323–333, 2015.

P. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3, 2013.

Dmitriy Morozov. *Homological Illusions of Persistence and Stability*. Ph.D. dissertation, Department of Computer Science, Duke University, 2008.

Elizabeth Munch and Bei Wang. Convergence between Categorical Representations of Reeb Space and Mapper. In *Proceedings of the 32nd Symposium on Computational Geometry*, volume 51, pages 53:1–53:16, 2016.

Sameer Nene, Shree Nayar, and Hiroshi Murase. Columbia Object Image Library (COIL-100). Technical Report CUCS-006-96, 1996.

Jessica Nielson, Jesse Paquette, Aiwen Liu, Cristian Guandique, Amy Tovar, Tomoo Inoue, Karen-Amanda Irvine, John Gensel, Jennifer Kloke, Tanya Petrossian, Pek Lum, Gunnar Carlsson, Geoffrey Manley, Wise Young, Michael Beattie, Jacqueline Bresnahan, and Adam Ferguson. Topological data analysis for discovery in preclinical spinal cord injury and traumatic brain injury. *Nature Communications*, 6, 2015.

Steve Oudot. *Persistence Theory: From Quiver Representations to Data Analysis*. Number 209 in Mathematical Surveys and Monographs. American Mathematical Society, 2015.

Gerald Reaven and Rupert Miller. An attempt to define the nature of chemical diabetes using a multidimensional analysis. *Diabetologia*, 16(1):17–24, 1979.

Georges Reeb. Sur les points singuliers d'une forme de Pfaff complètement intégrable ou d'une fonction numérique. *Compte Rendu de l'Académie des Science de Paris*, 222:847–849, 1946.

Matteo Rucco, Emanuela Merelli, Damir Herman, Devi Ramanan, Tanya Petrossian, Lorenzo Falsetti, Cinzia Nitti, and Aldo Salvi. Using topological data analysis for diagnosis pulmonary embolism. *Journal of Theoretical and Applied Computer Science*, 9 (1):41–55, 2015.

G. Sarikonda, J. Pettus, S. Phatak, S. Sachithanantham, JF. Miller, JD. Wesley, E. Cadag, J. Chae, L. Ganesan, R. Mallios, S. Edelman, B. Peters, and M. von Herrath. Cd8 t-cell reactivity to islet antigens is unique to type 1 while cd4 t-cell reactivity exists in both type 1 and type 2 diabetes. *Journal of Autoimmunity*, 50:77–82, 2014.

John Shawe-Taylor, Christopher Williams, Nello Cristianini, and Jaz Kandola. On the eigenspectrum of the Gram matrix and the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005.

Gurjeet Singh, Facundo Mémoli, and Gunnar Carlsson. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In *Symposium on Point Based Graphics*, pages 91–100, 2007.

Yuan Yao, Jian Sun, Xuhui Huang, Greg Bowman, Gurjeet Singh, Michael Lesnick, Leonidas Guibas, Vijay Pande, and Gunnar Carlsson. Topological methods for exploring low-density states in biomolecular folding pathways. *Journal of Chemical Physics*, 130 (14), 2009.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997.

Journal of Machine Learning Research 19 (2018) 1-48 Submitted 6/17; Revised 4/18; Published 8/18

A Robust Learning Approach for Regression Models Based on Distributionally Robust Optimization

Ruidi Chen RCHEN15@BU.EDU
Division of Systems Engineering,
Boston University,
Boston, MA 02215, USA

Ioannis Ch. Paschalidis YANNISP@BU.EDU
Department of Electrical and Computer Engineering,
Division of Systems Engineering,
and Department of Biomedical Engineering,
Boston University,
Boston, MA 02215, USA
sites. bu. edu/paschalidis

Editor: Edo Airoldi

Abstract

We present a *Distributionally Robust Optimization (DRO)* approach to estimate a robustified regression plane in a linear regression setting, when the observed samples are potentially contaminated with adversarially corrupted outliers. Our approach mitigates the impact of outliers by hedging against a family of probability distributions on the observed data, some of which assign very low probabilities to the outliers. The set of distributions under consideration are close to the empirical distribution in the sense of the Wasserstein metric. We show that this DRO formulation can be relaxed to a convex optimization problem which encompasses a class of models. By selecting proper norm spaces for the Wasserstein metric, we are able to recover several commonly used regularized regression models. We provide new insights into the regularization term and give guidance on the selection of the regularization coefficient from the standpoint of a confidence region. We establish two types of performance guarantees for the solution to our formulation under mild conditions. One is related to its out-of-sample behavior (prediction bias), and the other concerns the discrepancy between the estimated and true regression planes (estimation bias). Extensive numerical results demonstrate the superiority of our approach to a host of regression models, in terms of the prediction and estimation accuracies. We also consider the application of our robust learning procedure to outlier detection, and show that our approach achieves a much higher AUC (Area Under the ROC Curve) than M-estimation (Huber, 1964, 1973).
Keywords: Robust Learning, Distributionally Robust Optimization, Wasserstein Metric, Regularized Regression, Generalization Guarantees.

1. Introduction

Consider a linear regression model with response $y \in \mathbb{R}$, predictor vector $\mathbf{x} \in \mathbb{R}^{m-1}$, regression coefficient $\boldsymbol{\beta}^* \in \mathbb{R}^{m-1}$ and error $\epsilon \in \mathbb{R}$:

$$y = \mathbf{x}'\boldsymbol{\beta}^* + \epsilon.$$

Given samples $(\mathbf{x}_i, y_i), i = 1, \ldots, N$, we are interested in estimating $\boldsymbol{\beta}^*$. The *Ordinary Least Squares (OLS)* minimizes the sum of squared residuals $\sum_{i=1}^N (y_i - \mathbf{x}_i'\boldsymbol{\beta})^2$, and works well if all the N samples are generated from the underlying true model. However, when faced with adversarial perturbations in the training data, the OLS estimator will deviate from the true regression plane to reduce large residuals. Alternatively, one can choose to minimize the sum of absolute residuals $\sum_{i=1}^N |y_i - \mathbf{x}_i'\boldsymbol{\beta}|$, as done in *Least Absolute Deviation (LAD)*, to mitigate the influence of large residuals. Another commonly used approach for hedging against outliers is M-estimation (Huber, 1964, 1973), which minimizes a symmetric loss function $\rho(\cdot)$ of the residuals in the form $\sum_{i=1}^N \rho(y_i - \mathbf{x}_i'\boldsymbol{\beta})$, which downweights the influence of samples with large absolute residuals. Several choices for $\rho(\cdot)$ include the Huber function (Huber, 1964, 1973), the Tukey's Biweight function (Rousseeuw and Leroy, 2005), the logistic function (Coleman et al., 1980), the Talwar function (Hinich and Talwar, 1975), and the Fair function (Fair, 1974).

Both LAD and M-estimation are not resistant to large deviations in the predictors. For contamination present in the predictor space, high breakdown value methods are required. Examples include the *Least Median of Squares (LMS)* (Rousseeuw, 1984), which minimizes the median of the absolute residuals, the *Least Trimmed Squares (LTS)* (Rousseeuw, 1985), which minimizes the sum of the q smallest squared residuals, and S-estimation (Rousseeuw and Yohai, 1984), which has a higher statistical efficiency than LTS with the same breakdown value. A combination of the high breakdown value method and M-estimation is the MM-estimation (Yohai, 1987). It has a higher statistical efficiency than S-estimation. We refer the reader to the book of Rousseeuw and Leroy (2005) for a detailed description of these robust regression methods.

The aforementioned robust estimation procedures focus on modifying the objective function in a heuristic way with the intent of minimizing the effect of outliers. A more rigorous line of research explores the underlying stochastic program that leads to the sample-based estimation procedures. For example, the OLS objective can be viewed as minimizing the expected squared residual under the uniform empirical distribution over the samples. It has been well recognized that optimizing under the empirical distribution yields estimators that are sensitive to perturbations in the data and suffer from overfitting. The reason is that when the data $(\mathbf{x}, y)$ are adversarially corrupted by outliers, the observed samples do not represent well the true underlying distribution of the data. Yet, the samples are typically the only information available. Instead of equally weighting all the samples as in the empirical distribution, we may wish to include more informative distributions that "drive out" the corrupted samples. One way to realize this is to hedge the expected loss against a family of distributions that include the true data-generating mechanism with a high confidence; an approach called *Distributionally Robust Optimization (DRO)*. DRO minimizes the worst-case expected loss over a probabilistic ambiguity set $\mathcal{P}$ that is constructed from the observed samples and characterized by certain known properties of the true data-generating distribution. For example, Mehrotra and Zhang (2014) study the distributionally robust least squares problem with $\mathcal{P}$ defined through either moment constraints, norm bounds with moment constraints, or a confidence region over a reference probability measure. Compared to the single distribution-based stochastic optimization, DRO often results in better out-of-sample performance due to its distributional robustness.

The existing literature on DRO can be split into two main branches according to the way in which $\mathcal{P}$ is defined. One is through a moment ambiguity set, which contains all distributions that satisfy certain moment constraints (see Popescu, 2007; Delage and Ye, 2010; Goh and Sim, 2010; Zymler et al., 2013; Wiesemann et al., 2014). In many cases, it leads to a tractable DRO problem but has been criticized for yielding overly conservative solutions (Wang et al., 2016). The other is to define $\mathcal{P}$ as a ball of distributions using some probabilistic distance functions such as the ϕ-divergences (Bayraksan and Love, 2015), which include the Kullback-Leibler (KL) divergence (Hu and Hong, 2013; Jiang and Guan, 2015) as a special case, the Prokhorov metric (Erdoğan and Iyengar, 2006), and the Wasserstein distance (Esfahani and Kuhn, 2015; Gao and Kleywegt, 2016; Zhao and Guan, 2015; Luo and Mehrotra, 2017; Blanchet and Murthy, 2016). Deviating from the stochastic setting, there are also some works focusing on deterministic robustness. El Ghaoui and Lebret (1997) consider the least squares problem with unknown but bounded, non-random disturbance and solve it in polynomial time. Xu et al. (2010) study the robust linear regression problem with norm-bounded feature perturbation and show that it is equivalent to the ℓ_1-regularized regression. See Yang and Xu (2013); Bertsimas and Copenhaver (2017) which also use a deterministic robustness approach.

In this paper we consider a DRO problem with $\mathcal{P}$ containing distributions that are close to the discrete empirical distribution in the sense of Wasserstein distance. The reason for choosing the Wasserstein metric is two-fold. On one hand, the Wasserstein ambiguity set is rich enough to contain both continuous and discrete relevant distributions, while other metrics such as the KL divergence, exclude all continuous distributions if the nominal distribution is discrete (Esfahani and Kuhn, 2015; Gao and Kleywegt, 2016). Furthermore, considering distributions within a KL distance from the empirical, does not allow for probability mass outside the support of the empirical distribution. On the other hand, measure concentration results guarantee that the Wasserstein set contains the true data-generating distribution with high confidence for a sufficiently large sample size (Fournier and Guillin, 2015). Moreover, the Wasserstein metric takes into account the closeness between support points while other metrics such as the ϕ-divergence only consider the probabilities of these points. The image retrieval example in Gao and Kleywegt (2016) suggests that the probabilistic ambiguity set constructed based on the KL divergence prefers the pathological distribution to the true distribution, whereas the Wasserstein distance does not exhibit such a problem. The reason lies in that ϕ-divergence does not incorporate a notion of closeness between two points, which in the context of image retrieval represents the perceptual similarity in color.

Our DRO problem minimizes the worst-case absolute residual over a Wasserstein ball of distributions, and could be relaxed to the following form:

$$\inf_{\boldsymbol{\beta}} \frac{1}{N} \sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon \|(-\boldsymbol{\beta}, 1)\|_*, \tag{1}$$

where ϵ is the radius of the Wasserstein ball, and $\|\cdot\|_*$ is the dual norm of the norm space where the Wasserstein metric is defined on. Formulation (1) incorporates a wide class of models whose specific form depends on the notion of transportation cost embedded in the Wasserstein metric (see Section 2). Although the Wasserstein DRO formulation

simply reduces to regularized regression models, we want to emphasize a few new insights brought by this methodology. First, the regularization term controls the conservativeness of the Wasserstein set, or the amount of ambiguity in the data, which differentiates itself from the heuristically added regularizers in traditional regression models that serve the purpose of preventing overfitting, error/variance reduction, or sparsity recovery. Second, the regularization term is determined by the dual norm of the regression coefficient, which controls the *growth rate* of the ℓ_1-loss function, and the radius of the Wasserstein set. This connection provides guidance on the selection of the regularization coefficient and may lead to significant computational savings compared to cross-validation. DRO essentially enables new and more accurate interpretations of the regularizer, and establishes its dependence on the *growth rate* of the loss, the underlying metric space and the reliability of the observed samples.

The connection between robustness and regularization has been established in several works. The earliest one may be credited to El Ghaoui and Lebret (1997), who show that minimizing the worst-case squared residual within a Frobenius norm-based perturbation set is equivalent to Tikhonov regularization. In more recent works, using properly selected uncertainty sets, Xu et al. (2010) have shown the equivalence between robust linear regression with feature perturbations and the *Least Absolute Shrinkage and Selection Operator (LASSO)*. Yang and Xu (2013) extend this to more general LASSO-like procedures, including versions of the grouped LASSO. Bertsimas and Copenhaver (2017) give a comprehensive characterization of the conditions under which robustification and regularization are equivalent for regression models with deterministic norm-bounded perturbations on the features. For classification problems, Xu et al. (2009) show the equivalence between the regularized Support Vector Machines (SVMs) and a robust optimization formulation, by allowing potentially correlated disturbances in the covariates. Shafieezadeh-Abadeh et al. (2015) consider a robust version of logistic regression under the assumption that the probability distributions under consideration lie in a Wasserstein ball, and they show that the regularized logistic regression is a special case of this robust formulation. Recently, Shafieezadeh-Abadeh et al. (2017); Gao et al. (2017) have provided a unified framework for connecting the Wasserstein DRO with regularized learning procedures, for various regression and classification models.

Our work is motivated by the problem of identifying patients who receive an abnormally high radiation exposure in CT exams, given the patient characteristics and exam-related variables (Chen et al., 2018). This could be casted as an outlier detection problem; specifically, estimating a robustified regression plane that is immunized against outliers and learns the underlying true relationship between radiation dose and the relevant predictors. We focus on robust learning of the parameter in regression models under distributional perturbations residing within a Wasserstein ball. While the applicability of the Wasserstein DRO methodology is not restricted to regression analysis (Sinha et al., 2017; Gao et al., 2017; Shafieezadeh-Abadeh et al., 2017), or a particular form of the loss function (as long as it satisfies certain smoothness conditions (Gao et al., 2017)), we focus on the absolute residual loss in linear regression in light of our motivating application and for the purpose of enhancing robustness. Our contributions can be summarized as follows:

1. We develop a DRO approach to robustify linear regression using an ℓ_1 loss function and an ambiguity set around the empirical distribution of the training samples defined based on the Wasserstein metric. The formulation is general enough to in-

clude any norm-induced Wasserstein metric and incorporate additional regularization constraints on the regression coefficients (e.g., ℓ_1-norm constraints). It provides an intuitive connection between the amount of ambiguity allowed and a regularization penalty term in the robust formulation, which provides a natural way to adjust the latter.

2. We establish novel performance guarantees on both the out-of-sample loss (prediction bias) and the discrepancy between the estimated and the true regression coefficients (estimation bias). Our guarantees elucidate the role of the regularizer, which is related to the dual norm of the regression coefficients, in bounding the biases and are in concert with the theoretical foundation that leads to the regularized problem. The generalization error bound, in particular, builds a connection between the loss function and the form of the regularizer via Rademacher complexity, providing a rigorous explanation for the commonly observed good out-of-sample performance of regularized regression. On the other hand, the estimation error bound corroborates the validity of the ℓ_1-loss function, which tends to incur a lower estimation bias than other candidates such as the ℓ_2 and ℓ_∞ losses. Our results are novel in the robust regression setting and different from earlier work in the DRO literature, enabling new perspectives and interpretations of the norm-based regularization, and providing justifications for the ℓ_1-loss-based learning algorithms.

3. We empirically explore three important aspects of the Wasserstein DRO formulation, including the advantages of the ℓ_1-loss function, the selection of a proper norm for the Wasserstein metric, and the implication of penalizing the *extended regression coefficient* $(-\boldsymbol{\beta}, 1)$, by comparing with a series of regression models on a number of synthetic datasets. We show the superiority of the Wasserstein DRO approach, presenting a thorough analysis under four different experimental setups. We also consider the application of our methodology to outlier detection and compare with M-estimation in terms of the ability of identifying outliers (*ROC (Receiver Operating Characteristic)* curves). The Wasserstein DRO formulation achieves significantly higher *AUC (Area Under Curve)* values.

The rest of the paper is organized as follows. In Section 2, we introduce the Wasserstein metric and derive the general Wasserstein DRO formulation in a linear regression framework. Section 3 establishes performance guarantees for both the general formulation and the special case where the Wasserstein metric is defined on the ℓ_1-norm space. Numerical experimental results are presented in Section 4. We conclude the paper in Section 5.

Notational conventions: We use boldfaced lowercase letters to denote vectors, ordinary lowercase letters to denote scalars, boldfaced uppercase letters to denote matrices, and calligraphic capital letters to denote sets. $\mathbb{E}$ denotes expectation and $\mathbb{P}$ probability of an event. All vectors are column vectors. For space saving reasons, we write $\mathbf{x} = (x_1, \ldots, x_{\dim(\mathbf{x})})$ to denote the column vector $\mathbf{x}$, where $\dim(\mathbf{x})$ is the dimension of $\mathbf{x}$. We use prime to denote the transpose of a vector, $\|\cdot\|$ for the general norm operator, $\|\cdot\|_2$ for the ℓ_2 norm, $\|\cdot\|_1$ for the ℓ_1 norm, and $\|\cdot\|_\infty$ for the infinity norm. $\mathcal{P}(\mathcal{Z})$ denotes the set of probability measures supported on $\mathcal{Z}$. $\mathbf{e}_i$ denotes the i-th unit vector, $\mathbf{e}$ the vector of ones, $\mathbf{0}$ a vector of zeros, and $\mathbf{I}$ the identity matrix. Given a norm $\|\cdot\|$ on $\mathbb{R}^m$, the dual

norm $\|\cdot\|_*$ is defined as: $\|\boldsymbol{\theta}\|_* \triangleq \sup_{\|\mathbf{z}\|\leq 1} \boldsymbol{\theta}'\mathbf{z}$. For a function $h(\mathbf{z})$, its convex conjugate $h^*(\cdot)$ is defined as: $h^*(\boldsymbol{\theta}) \triangleq \sup_{\mathbf{z}\in \text{dom } h} \{\boldsymbol{\theta}'\mathbf{z} - h(\mathbf{z})\}$, where dom h denotes the domain of the function h.

2. Problem Statement and Justification of Our Formulation

Consider a linear regression problem where we are given a predictor/feature vector $\mathbf{x} \in \mathbb{R}^{m-1}$, and a response variable $y \in \mathbb{R}$. Our goal is to obtain an accurate estimate of the regression plane that is robust with respect to the adversarial perturbations in the data. We consider an ℓ_1-loss function $h_{\boldsymbol{\beta}}(\mathbf{x}, y) \triangleq |y - \mathbf{x}'\boldsymbol{\beta}|$, motivated by the observation that the absolute loss function is more robust to large residuals than the squared loss (see Fig. 1). Moreover, the estimation error analysis presented in Section 3.2 suggests that the ℓ_1-loss function leads to a smaller estimation bias than others. Our Wasserstein DRO problem using the ℓ_1-loss function is formulated as:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \sup_{\mathbb{Q}\in\Omega} \mathbb{E}^{\mathbb{Q}}\big[|y - \mathbf{x}'\boldsymbol{\beta}|\big], \tag{2}$$

where $\boldsymbol{\beta}$ is the regression coefficient vector that belongs to some set $\mathcal{B}$. $\mathcal{B}$ could be $\mathbb{R}^{m-1}$, or $\mathcal{B} = \{\boldsymbol{\beta} : \|\boldsymbol{\beta}\|_1 \leq l\}$ if we wish to induce sparsity, with l being some pre-specified number. $\mathbb{Q}$ is the probability distribution of $(\mathbf{x}, y)$, belonging to some set Ω which is defined as:

$$\Omega \triangleq \{\mathbb{Q} \in \mathcal{P}(\mathcal{Z}) : W_p(\mathbb{Q},\ \hat{\mathbb{P}}_N) \leq \epsilon\},$$

where $\mathcal{Z}$ is the set of possible values for $(\mathbf{x}, y)$; $\mathcal{P}(\mathcal{Z})$ is the space of all probability distributions supported on $\mathcal{Z}$; ϵ is a pre-specified radius of the Wasserstein ball; and $W_p(\mathbb{Q},\ \hat{\mathbb{P}}_N)$ is the order-p Wasserstein distance between $\mathbb{Q}$ and $\hat{\mathbb{P}}_N$ (see definition in (3)), with $\hat{\mathbb{P}}_N$ the uniform empirical distribution over samples. The formulation in (2) is robust since it minimizes over the regression coefficients the worst case expected loss, that is, the expected loss maximized over all probability distributions in the ambiguity set Ω.

Before deriving a tractable reformulation for (2), let us first define the Wasserstein metric. Let $(\mathcal{Z}, s)$ be a metric space where $\mathcal{Z}$ is a set and s is a metric on $\mathcal{Z}$. The Wasserstein metric of order $p \geq 1$ defines the distance between two probability distributions $\mathbb{Q}_1$ and $\mathbb{Q}_2$ in the following way:

$$W_p(\mathbb{Q}_1, \mathbb{Q}_2) \triangleq \left(\min_{\Pi\in\mathcal{P}(\mathcal{Z}\times\mathcal{Z})} \left\{\int_{\mathcal{Z}\times\mathcal{Z}} \big(s((\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2))\big)^p\ \Pi\big(d(\mathbf{x}_1, y_1), d(\mathbf{x}_2, y_2)\big)\right\}\right)^{1/p}, \tag{3}$$

where Π is the joint distribution of $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$ with marginals $\mathbb{Q}_1$ and $\mathbb{Q}_2$, respectively. The Wasserstein distance between $\mathbb{Q}_1$ and $\mathbb{Q}_2$ represents the cost of an optimal mass transportation plan, where the cost is measured through the metric s. The order p should be selected in such a way as to ensure that the worst-case expected loss is meaningfully defined, i.e.,

$$\mathbb{E}^{\mathbb{Q}}\big[h_{\boldsymbol{\beta}}(\mathbf{x}, y)\big] < \infty,\ \forall \mathbb{Q} \in \Omega. \tag{4}$$

Notice that the ambiguity set Ω is centered at the empirical distribution $\hat{\mathbb{P}}_N$ and has radius ϵ. It may be desirable to translate (4) into:

$$\left|\mathbb{E}^{\mathbb{Q}}\big[h_{\boldsymbol{\beta}}(\mathbf{x}, y)\big] - \mathbb{E}^{\hat{\mathbb{P}}_N}\big[h_{\boldsymbol{\beta}}(\mathbf{x}, y)\big]\right| < \infty,\ \forall \mathbb{Q} \in \Omega. \tag{5}$$

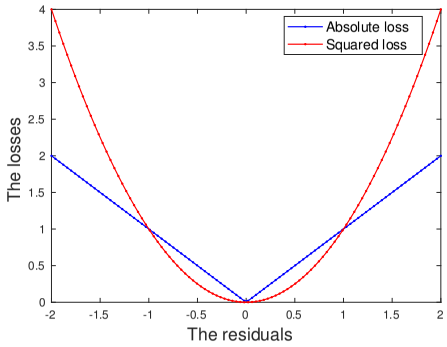


Figure 1: Comparison between ℓ_1 and ℓ_2 loss functions.

We want to relate (5) with the Wasserstein distance $W_p(\mathbb{Q}, \hat{\mathbb{P}}_N)$, which is no larger than ϵ for all $\mathbb{Q} \in \Omega$. The LHS of (5) could be written as:

$$\begin{aligned}
&\left|\mathbb{E}^{\mathbb{Q}}\big[h_{\boldsymbol{\beta}}(\mathbf{x},y)\big] - \mathbb{E}^{\hat{\mathbb{P}}_N}\big[h_{\boldsymbol{\beta}}(\mathbf{x},y)\big]\right| \\
=&\left|\int_{\mathcal{Z}} h_{\boldsymbol{\beta}}(\mathbf{x}_1,y_1)\mathbb{Q}(d(\mathbf{x}_1,y_1)) - \int_{\mathcal{Z}} h_{\boldsymbol{\beta}}(\mathbf{x}_2,y_2)\hat{\mathbb{P}}_N(d(\mathbf{x}_2,y_2))\right| \\
=&\left|\int_{\mathcal{Z}} h_{\boldsymbol{\beta}}(\mathbf{x}_1,y_1)\int_{\mathcal{Z}} \Pi_0(d(\mathbf{x}_1,y_1),d(\mathbf{x}_2,y_2)) - \int_{\mathcal{Z}} h_{\boldsymbol{\beta}}(\mathbf{x}_2,y_2)\int_{\mathcal{Z}} \Pi_0(d(\mathbf{x}_1,y_1),d(\mathbf{x}_2,y_2))\right| \\
\leq&\int_{\mathcal{Z}\times\mathcal{Z}} \big|h_{\boldsymbol{\beta}}(\mathbf{x}_1,y_1) - h_{\boldsymbol{\beta}}(\mathbf{x}_2,y_2)\big|\Pi_0(d(\mathbf{x}_1,y_1),d(\mathbf{x}_2,y_2)),
\end{aligned} \tag{6}$$

where Π_0 is the joint distribution of $(\mathbf{x}_1,y_1)$ and $(\mathbf{x}_2,y_2)$ with marginals $\mathbb{Q}$ and $\hat{\mathbb{P}}_N$, respectively. Comparing (6) with (3), we see that for (5) to hold, the following quantity which characterizes the *growth rate* of the loss function needs to be bounded:

$$\mathrm{GR}_{h_{\boldsymbol{\beta}}}((\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2)) \triangleq \frac{\big|h_{\boldsymbol{\beta}}(\mathbf{x}_1,y_1) - h_{\boldsymbol{\beta}}(\mathbf{x}_2,y_2)\big|}{\big(s((\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2))\big)^p},\ \forall (\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2) \in \mathcal{Z}. \tag{7}$$

A formal definition of the growth rate is due to Gao and Kleywegt (2016), which takes the limit of (7) as $s((\mathbf{x}_1,y_1),(\mathbf{x}_2,y_2)) \to \infty$, to eliminate its dependence on $(\mathbf{x},y)$. One important aspect they have pointed out is that when the growth rate of the loss function is infinite, strong duality for the worst-case problem $\sup_{\mathbb{Q}\in\Omega}\mathbb{E}^{\mathbb{Q}}\big[h_{\boldsymbol{\beta}}(\mathbf{x},y)\big]$ fails to hold, in which case the DRO problem (2) becomes intractable. Assuming that the metric s is

induced by some norm $\|\cdot\|$, the bounded *growth rate* requirement is expressed as follows:

$$\begin{aligned}
&\limsup_{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|\to\infty} \frac{|h_{\boldsymbol{\beta}}(\mathbf{x}_1,y_1) - h_{\boldsymbol{\beta}}(\mathbf{x}_2,y_2)|}{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|^p} \leq \limsup_{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|\to\infty} \frac{|y_1 - \mathbf{x}_1'\boldsymbol{\beta} - (y_2 - \mathbf{x}_2'\boldsymbol{\beta})|}{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|^p} \\
\leq &\limsup_{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|\to\infty} \frac{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|\|(-\boldsymbol{\beta},1)\|_*}{\|(\mathbf{x}_1,y_1)-(\mathbf{x}_2,y_2)\|^p} < \infty,
\end{aligned} \tag{8}$$

where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$, and the second inequality is due to the Cauchy-Schwarz inequality. Notice that by taking $p = 1$, (8) is equivalently translated into the condition that $\|(-\boldsymbol{\beta},1)\|_* < \infty$, which we will see in Section 3 is an essential requirement to guarantee a good generalization performance for the Wasserstein DRO estimator. The growth rate essentially reveals the underlying metric space used by the Wasserstein distance. Taking $p > 1$ leads to zero growth rate in the limit of (8), which is not desirable since it removes the Wasserstein ball structure from our formulation and renders it an optimization problem over a singleton distribution. This will be made more clear in the following analysis. We thus choose the order-1 Wasserstein metric with s being induced by some norm $\|\cdot\|$ to define our DRO problem.

Next, we will discuss how to convert (2) into a tractable formulation. Suppose we have N independently and identically distributed realizations of $(\mathbf{x},y)$, denoted by $(\mathbf{x}_i,y_i), i = 1,\ldots,N$. We make the assumption that $(\mathbf{x},y)$ comes from a mixture of two distributions, with probability q from the outlying distribution $\mathbb{P}_{out}$ and with probability $1-q$ from the true distribution $\mathbb{P}$. Recall that $\hat{\mathbb{P}}_N$ is the discrete uniform distribution over the N samples. Our goal is to generate estimators that are consistent with the true distribution $\mathbb{P}$. We claim that when q is small, if the Wasserstein ball radius ϵ is chosen judiciously, the true distribution $\mathbb{P}$ will be included in the set Ω while the outlying distribution $\mathbb{P}_{out}$ will be excluded. To see this, consider a simple example where $\mathbb{P}$ is a discrete distribution that assigns equal probability to 10 data points equally spaced between 0.1 and 1, and $\mathbb{P}_{out}$ assigns probability 0.5 to two data points 1 and 2. We generate 100 samples and plot the Wasserstein distances from $\hat{\mathbb{P}}_N$ for both $\mathbb{P}$ and $\mathbb{P}_{out}$. From Fig. 2 we observe that for q below 0.5, the true distribution $\mathbb{P}$ is closer to $\hat{\mathbb{P}}_N$ whereas the outlying distribution $\mathbb{P}_{out}$ is further away. If the radius ϵ is chosen between the red $(*-)$ and blue $(\circ-)$ lines, the Wasserstein ball that we are hedging against will exclude the outlying distribution and the resulting estimator will be robust to the adversarial perturbations. Moreover, as q becomes smaller, the gap between the red and blue lines becomes larger. One implication from this observation is that as the data becomes purer, the radius of the Wasserstein ball tends to be smaller, and the confidence in the observed samples is higher. For large q values, the DRO formulation seems to fail. However, as outliers are defined to be the data points that do not conform to the majority of data, we can safely claim that $\mathbb{P}_{out}$ is the distribution of the minority and q is always below 0.5.

We now consider the inner supremum in (2). Esfahani and Kuhn (2015, Theorem 6.3) show that when the set $\mathcal{Z}$ is closed and convex, and the loss function $h_{\boldsymbol{\beta}}(\mathbf{x},y)$ is convex in $(\mathbf{x},y)$,

$$\sup_{\mathbb{Q}\in\Omega} \mathbb{E}^{\mathbb{Q}}[h_{\boldsymbol{\beta}}(\mathbf{x},y)] \leq \kappa\epsilon + \frac{1}{N}\sum_{i=1}^{N} h_{\boldsymbol{\beta}}(\mathbf{x}_i,y_i),\ \forall \epsilon \geq 0, \tag{9}$$

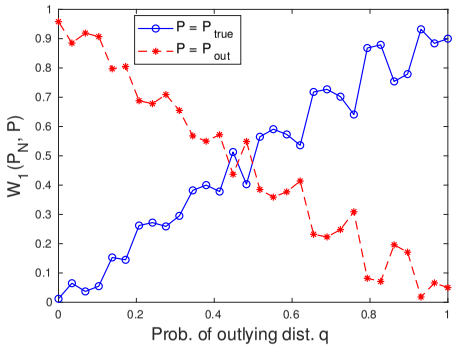


Figure 2: The order-1 Wasserstein distances from the empirical distribution.

where $\kappa(\boldsymbol{\beta}) = \sup\{\|\boldsymbol{\theta}\|_* : h^*_{\boldsymbol{\beta}}(\boldsymbol{\theta}) < \infty\}$, with $h^*_{\boldsymbol{\beta}}(\cdot)$ the convex conjugate function of $h_{\boldsymbol{\beta}}(\mathbf{x}, y)$. Through (9), we can relax problem (2) by minimizing the right hand side of (9) instead of the worst-case expected loss. Moreover, as shown in Esfahani and Kuhn (2015), (9) becomes an equality when $\mathcal{Z} = \mathbb{R}^m$. In Theorem 2.1, we compute the value of $\kappa(\boldsymbol{\beta})$ for the specific ℓ_1 loss function we use. The proof of this Theorem and all results hereafter are included in Appendix A.

Theorem 2.1 *Define $\kappa(\boldsymbol{\beta}) = \sup\{\|\boldsymbol{\theta}\|_* : h^*_{\boldsymbol{\beta}}(\boldsymbol{\theta}) < \infty\}$, where $\|\cdot\|_*$ is the dual norm of $\|\cdot\|$, and $h^*_{\boldsymbol{\beta}}(\cdot)$ is the conjugate function of $h_{\boldsymbol{\beta}}(\cdot)$. When the loss function is $h_{\boldsymbol{\beta}}(\mathbf{x}, y) = |y - \mathbf{x}'\boldsymbol{\beta}|$, we have $\kappa(\boldsymbol{\beta}) = \|(-\boldsymbol{\beta}, 1)\|_*$.*

Due to Theorem 2.1, (2) could be formulated as the following optimization problem:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \epsilon\|(-\boldsymbol{\beta}, 1)\|_* + \frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}|. \tag{10}$$

Note that the regularization term of (10) is the product of the *growth rate* of the loss and the Wasserstein ball radius. The growth rate is closely related to the way the Wasserstein metric defines the transportation costs on the data $(\mathbf{x}, y)$. As mentioned earlier, a zero growth rate diminishes the effect of the Wasserstein distributional uncertainty set, and the resulting formulation would simply be an empirical loss minimization problem. The parameter ϵ controls the conservativeness of the formulation, whose selection depends on the sample size, the dimensionality of the data, and the confidence that the Wasserstein ball contains the true distribution (see eq. (8) in Esfahani and Kuhn, 2015). Roughly speaking, when the sample size is large enough, and for a fixed confidence level, ϵ is inversely proportional to $N^{1/m}$.

Formulation (10) incorporates a class of models whose specific form depends on the norm space we choose, which could be application-dependent and practically useful. For example, when the Wasserstein metric s is induced by $\|\cdot\|_2$ and the set $\mathcal{B}$ is the intersection of a

polyhedron with convex quadratic inequalities, (10) is a convex quadratic problem which can be solved to optimality very efficiently. Specifically, it could be converted to:

$$\begin{aligned} \min_{a,\ b_1,\ldots,b_N,\ \boldsymbol{\beta}} \quad & a\epsilon + \frac{1}{N}\sum_{i=1}^{N} b_i \\ \text{s.t.} \quad & \|\boldsymbol{\beta}\|_2^2 + 1 \le a^2, \\ & y_i - \mathbf{x}_i'\boldsymbol{\beta} \le b_i,\ i = 1,\ldots,N, \\ & -(y_i - \mathbf{x}_i'\boldsymbol{\beta}) \le b_i,\ i = 1,\ldots,N, \\ & a,\ b_i \ge 0,\ i = 1,\ldots,N, \\ & \boldsymbol{\beta} \in \mathcal{B}. \end{aligned} \tag{11}$$

When the Wasserstein metric is defined using $\|\cdot\|_1$ and the set $\mathcal{B}$ is a polyhedron, (10) is a linear programming problem:

$$\begin{aligned} \min_{a,\ b_1,\ldots,b_N,\ \boldsymbol{\beta}} \quad & a\epsilon + \frac{1}{N}\sum_{i=1}^{N} b_i \\ \text{s.t.} \quad & a \ge \boldsymbol{\beta}'\mathbf{e}_i,\ i = 1,\ldots,m-1, \\ & a \ge -\boldsymbol{\beta}'\mathbf{e}_i,\ i = 1,\ldots,m-1, \\ & y_i - \mathbf{x}_i'\boldsymbol{\beta} \le b_i,\ i = 1,\ldots,N, \\ & -(y_i - \mathbf{x}_i'\boldsymbol{\beta}) \le b_i,\ i = 1,\ldots,N, \\ & a \ge 1, \\ & b_i \ge 0,\ i = 1,\ldots,N, \\ & \boldsymbol{\beta} \in \mathcal{B}. \end{aligned} \tag{12}$$

More generally, when the coordinates of $(\mathbf{x}, y)$ differ from each other substantially, a properly chosen, positive definite weight matrix $\mathbf{M} \in \mathbb{R}^{m\times m}$ could scale correspondingly different coordinates of $(\mathbf{x}, y)$ by using the $\mathbf{M}$-weighted norm:

$$\|(\mathbf{x}, y)\|_{\mathbf{M}} = \sqrt{(\mathbf{x}, y)'\mathbf{M}(\mathbf{x}, y)}.$$

It can be shown that (10) in this case becomes:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \epsilon\sqrt{(-\boldsymbol{\beta}, 1)'\mathbf{M}^{-1}(-\boldsymbol{\beta}, 1)} + \frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}|. \tag{13}$$

We note that this Wasserstein DRO framework could be applied to a broad class of loss functions and the tractable reformulations have been derived in Shafieezadeh-Abadeh et al. (2017); Gao et al. (2017) for regression and classification models. We adopt the absolute residual loss in this paper to enhance the robustness of the formulation, which is the focus of our work and serves the purpose of estimating robust parameters that are immunized against perturbations/outliers. Notice that (10) coincides with the regularized LAD models (Pollard, 1991; Wang et al., 2006), except that we are regularizing a variant of the regression coefficient. We would like to highlight several novel viewpoints that are

brought by the Wasserstein DRO framework and justify the value and novelty of (10). First, (10) is obtained as an outcome of a fundamental DRO formulation, which enables new interpretations of the regularizer from the standpoint of distributional robustness, and provides rigorous theoretical foundation on why the ℓ_2-regularizer prevents overfitting to the training data. The regularizer could be seen as a control over the amount of ambiguity in the data and reveals the reliability of the contaminated samples. Second, the geometry of the Wasserstein ball is embedded in the regularization term, which penalizes the regression coefficient on the dual Wasserstein space, with the magnitude of penalty being the radius of the ball. This offers an intuitive interpretation and provides guidance on how to set the regularization coefficient. Moreover, different from the traditional regularized LAD models that directly penalize the regression coefficient $\boldsymbol{\beta}$, we regularize the vector $(-\boldsymbol{\beta}, 1)$, where the 1 takes into account the transportation cost along the y direction. Penalizing only on $\boldsymbol{\beta}$ corresponds to an infinite transportation cost along y. Our model is more general in this sense, and establishes the connection between the metric space on data and the form of the regularizer.

3. Performance Guarantees

Having obtained a tractable reformulation for the Wasserstein DRO problem, we next establish guarantees on the predictive power and estimation quality for the solution to (10). Two types of results will be presented in this section, one of which bounds the prediction bias of the estimator on new, future data (given in Section 3.1). The other one that bounds the discrepancy between the estimated and true regression planes (estimation bias), is given in Section 3.2.

3.1 Out-of-Sample Performance

In this subsection we investigate generalization characteristics of the solution to (10), which involves measuring the error generated by our estimator on a new random sample $(\mathbf{x}, y)$. We would like to obtain estimates that not only explain the observed samples well, but, more importantly, possess strong generalization abilities. The derivation is mainly based on *Rademacher complexity* (see Bartlett and Mendelson, 2002), which is a measurement of the complexity of a class of functions. We would like to emphasize the applicability of such a proof technique to general loss functions, as long as their empirical Rademacher complexity could be bounded. The bound we derive for the prediction bias depends on both the sample average loss (the training error) and the dual norm of the regression coefficient (the regularizer), which corroborates the validity and necessity of our regularized formulation. Moreover, the generalization result also builds a connection between the loss function and the form of the regularizer via Rademacher complexity, which enables new insights into the regularization term and explains the commonly observed good out-of-sample performance of regularized regression in a rigorous way. We first make several mild assumptions that are needed for the generalization result.

Assumption A *The norm of the uncertainty parameter $(\mathbf{x}, y)$ is bounded above almost surely, i.e., $\|(\mathbf{x}, y)\| \leq R$.*

Assumption B *The dual norm of $(-\boldsymbol{\beta}, 1)$ is bounded above within the feasible region, namely,*

$$\sup_{\boldsymbol{\beta} \in \mathcal{B}} \|(-\boldsymbol{\beta}, 1)\|_* = \bar{B}.$$

Under these two assumptions, the absolute loss could be bounded via the Cauchy-Schwarz inequality.

Lemma 3.1 *For every feasible $\boldsymbol{\beta}$, it follows*

$$|y - \mathbf{x}'\boldsymbol{\beta}| \leq \bar{B}R, \quad \textit{almost surely.}$$

With the above result, the idea is to bound the generalization error using the empirical *Rademacher complexity* of the following class of loss functions:

$$\mathcal{H} = \{(\mathbf{x}, y) \mapsto h_{\boldsymbol{\beta}}(\mathbf{x}, y) : h_{\boldsymbol{\beta}}(\mathbf{x}, y) = |y - \mathbf{x}'\boldsymbol{\beta}|,\ \boldsymbol{\beta} \in \mathcal{B}\}.$$

We need to show that the empirical Rademacher complexity of $\mathcal{H}$, denoted by $\mathcal{R}_N(\mathcal{H})$, is upper bounded. The following result, similar to Lemma 3 in Bertsimas et al. (2015), provides a bound that is inversely proportional to the square root of the sample size.

Lemma 3.2

$$\mathcal{R}_N(\mathcal{H}) \leq \frac{2\bar{B}R}{\sqrt{N}}.$$

Let $\hat{\boldsymbol{\beta}}$ be an optimal solution to (10), obtained using the samples $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, N$. Suppose we draw a new i.i.d. sample $(\mathbf{x}, y)$. In Theorem 3.3 we establish bounds on the error $|y - \mathbf{x}'\hat{\boldsymbol{\beta}}|$.

Theorem 3.3 *Under Assumptions A and B, for any $0 < \delta < 1$, with probability at least $1 - \delta$ with respect to the sampling,*

$$\mathbb{E}[|y - \mathbf{x}'\hat{\boldsymbol{\beta}}|] \leq \frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}| + \frac{2\bar{B}R}{\sqrt{N}} + \bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}\ , \tag{14}$$

and for any $\zeta > \frac{2\bar{B}R}{\sqrt{N}} + \bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}$,

$$\mathbb{P}\left(|y - \mathbf{x}'\hat{\boldsymbol{\beta}}| \geq \frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}| + \zeta\right) \leq \frac{\frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}| + \frac{2\bar{B}R}{\sqrt{N}} + \bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}}{\frac{1}{N}\sum_{i=1}^{N} |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}| + \zeta}. \tag{15}$$

There are two probability measures in the statement of Theorem 3.3. One is related to the new data $(\mathbf{x}, y)$, while the other is related to the samples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)$. The expectation in (14) (and the probability in (15)) is taken w.r.t. the new data $(\mathbf{x}, y)$. For a given set of samples, (14) (and (15)) holds with probability at least $1-\delta$ w.r.t. the measure of samples. Theorem 3.3 essentially says that given typical samples, the expected loss on new data using our Wasserstein DRO estimator could be bounded above by the average sample loss plus extra terms that depend on the supremum of $\|(-\boldsymbol{\beta}, 1)\|_*$ (our regularizer), and

are proportional to $1/\sqrt{N}$. This result validates the dual norm-based regularized regression from the perspective of generalization ability, and could be generalized to any bounded loss function. It also provides implications on the form of the regularizer. For example, if given an ℓ_2-loss function, the dependency on $\bar{B}$ for the generalization error bound will be of the form $\bar{B}^2$, which suggests using $\|(-\boldsymbol{\beta}, 1)\|_*^2$ as a regularizer, reducing to a variant of ridge regression (Hoerl and Kennard, 1970) for $\|\cdot\|_2$ induced Wasserstein metric.

We also note that the upper bounds in (14) and (15) do not depend on the dimension of $(\mathbf{x}, y)$. This dimensionality-free characteristic implies direct applicability of our Wasserstein approach to high-dimensional settings and is particularly useful in many real applications where, potentially, hundreds of features may be present. Theorem 3.3 also provides guidance on the number of samples that are needed to achieve satisfactory out-of-sample performance.

Corollary 3.4 *Suppose $\hat{\boldsymbol{\beta}}$ is the optimal solution to (10). For a fixed confidence level δ and some threshold parameter $\tau \geq 0$, to guarantee that the percentage difference between the expected absolute loss on new data and the sample average loss is less than τ, that is,*

$$\frac{\mathbb{E}[|y - \mathbf{x}'\hat{\boldsymbol{\beta}}|] - \frac{1}{N}\sum_{i=1}^N |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}|}{\bar{B}R} \leq \tau,$$

the sample size N must satisfy

$$N \geq \left[\frac{2(1 + \sqrt{2\log(2/\delta)}\ )}{\tau}\right]^2. \tag{16}$$

Corollary 3.5 *Suppose $\hat{\boldsymbol{\beta}}$ is the optimal solution to (10). For a fixed confidence level δ, some $\tau \in (0, 1)$ and $\gamma \geq 0$, to guarantee that*

$$\mathbb{P}\left(\frac{|y - \mathbf{x}'\hat{\boldsymbol{\beta}}| - \frac{1}{N}\sum_{i=1}^N |y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}}|}{\bar{B}R} \geq \gamma\right) \leq \tau,$$

the sample size N must satisfy

$$N \geq \left[\frac{2(1 + \sqrt{2\log(2/\delta)}\ )}{\tau \cdot \gamma + \tau - 1}\right]^2, \tag{17}$$

provided that $\tau \cdot \gamma + \tau - 1 > 0$.

In Corollaries 3.4 and 3.5, the sample size is inversely proportional to both δ and τ, which is reasonable since the more confident we want to be, the more samples we need. Moreover, the smaller τ is, the stricter a requirement we impose on the performance, and thus more samples are needed.

3.2 Discrepancy between Estimated and True Regression Planes

In addition to the generalization performance, we are also interested in the accuracy of the estimator. In this section we seek to bound the difference between the estimated and true regression coefficients, under a certain distributional assumption on $(\mathbf{x}, y)$. Throughout the

section we will use $\hat{\boldsymbol{\beta}}$ to denote the estimated regression coefficients, obtained as an optimal solution to (18), and $\boldsymbol{\beta}^*$ for the true (unknown) regression coefficients. The bound we will derive turns out to be related to the Gaussian width (see definition in the Appendix) of the unit ball in $\|\cdot\|_\infty$, the sub-Gaussian norm of the uncertainty parameter $(\mathbf{x}, y)$, as well as the geometric structure of the true regression coefficients. We note that this proof technique may be applied to several other loss functions, e.g., ℓ_2 and ℓ_∞ losses, with slight modifications. However, we will see that the ℓ_1-loss function incurs a relatively low estimation bias compared to others, further demonstrating the superiority of our absolute error minimization formulation.

To facilitate the analysis, we will use the following equivalent form of problem (10):

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|(-\boldsymbol{\beta}, 1)\|_* \\ \text{s.t.} \quad & \|(-\boldsymbol{\beta}, 1)'\mathbf{Z}\|_1 \leq \gamma_N, \\ & \boldsymbol{\beta} \in \mathcal{B}, \end{aligned} \tag{18}$$

where $\mathbf{Z} = [(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)]$ is the matrix with columns $(\mathbf{x}_i, y_i), i = 1, \ldots, N$, and γ_N is some exogenous parameter related to ϵ. One can show that for properly chosen γ_N, (18) produces the same solution with (10) (Bertsekas, 1999). (18) is similar to (11) in Chen and Banerjee (2016), with the difference lying in that we impose a constraint on the error instead of the gradient, and we consider a more general notion of norm on the coefficient. On the other hand, due to their similarity, we will follow the line of development in Chen and Banerjee (2016). Still, our analysis is self-contained and the bound we obtain is in a different form, which provides meaningful insights into our specific problem. We list below the assumptions that are needed to bound the estimation error.

Assumption C *The ℓ_2 norm of $(-\boldsymbol{\beta}, 1)$ is bounded above within the feasible region, namely,*

$$\sup_{\boldsymbol{\beta} \in \mathcal{B}} \|(-\boldsymbol{\beta}, 1)\|_2 = \bar{B}_2.$$

Assumption D (Restricted Eigenvalue Condition) *For some set $\mathcal{A}(\boldsymbol{\beta}^*) = cone\{\mathbf{v} | \|(-\boldsymbol{\beta}^*, 1) + \mathbf{v}\|_* \leq \|(-\boldsymbol{\beta}^*, 1)\|_*\} \cap \mathbb{S}^m$ and some positive scalar $\underline{\alpha}$, where $\mathbb{S}^m$ is the unit sphere in the m-dimensional Euclidean space,*

$$\inf_{\mathbf{v} \in \mathcal{A}(\boldsymbol{\beta}^*)} \mathbf{v}'\mathbf{Z}\mathbf{Z}'\mathbf{v} \geq \underline{\alpha},$$

where $\mathbb{S}^m$ denotes the unit sphere in the m-dimensional Euclidean space.

Assumption E *The true coefficient $\boldsymbol{\beta}^*$ is a feasible solution to (18), i.e.,*

$$\|\mathbf{Z}'(-\boldsymbol{\beta}^*, 1)\|_1 \leq \gamma_N, \quad \boldsymbol{\beta}^* \in \mathcal{B}.$$

Assumption F *$(\mathbf{x}, y)$ is a centered sub-Gaussian random vector (see definition in the Appendix), i.e., it has zero mean and satisfies the following condition:*

$$\|\|(\mathbf{x}, y)\|\|_{\psi_2} = \sup_{\mathbf{u} \in \mathbb{S}^m} \left\|\left|(\mathbf{x}, y)'\mathbf{u}\right|\right\|_{\psi_2} \leq \mu.$$

Assumption G *The covariance matrix of* $(\mathbf{x}, y)$ *has bounded positive eigenvalues. Set* $\mathbf{\Gamma} = \mathbb{E}[(\mathbf{x}, y)(\mathbf{x}, y)']$*; then,*

$$0 < \lambda_{min} \triangleq \lambda_{min}(\mathbf{\Gamma}) \leq \lambda_{max}(\mathbf{\Gamma}) \triangleq \lambda_{max} < \infty.$$

Notice that both $\underline{\alpha}$ in Assumption D and γ_N in Assumption E are related to the random observation matrix $\mathbf{Z}$. A probabilistic description for these two quantities will be provided later. We next present a preliminary result, similar to Lemma 2 in Chen and Banerjee (2016), that bounds the ℓ_2-norm of the estimation bias in terms of a quantity that is related to the geometric structure of the true coefficients. This result gives a rough idea on the factors that affect the estimation error, and shows the advantages of using the ℓ_1-loss from the perspective of its dual norm. The bound derived in Theorem 3.6 is crude in the sense that it is a function of several random parameters that are related to the random observation matrix $\mathbf{Z}$. This randomness will be described in a probabilistic way in the subsequent analysis.

Theorem 3.6 *Suppose the true regression coefficient vector is* $\boldsymbol{\beta}^*$ *and the solution to (18) is* $\hat{\boldsymbol{\beta}}$*. For the set* $\mathcal{A}(\boldsymbol{\beta}^*) = cone\{\mathbf{v} |\ \|(-\boldsymbol{\beta}^*, 1) + \mathbf{v}\|_* \leq \|(-\boldsymbol{\beta}^*, 1)\|_*\} \cap \mathbb{S}^m$*, under Assumptions A, D, and E, we have:*

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \frac{2R\gamma_N}{\underline{\alpha}} \Psi(\boldsymbol{\beta}^*), \tag{19}$$

where $\Psi(\boldsymbol{\beta}^*) = \sup_{\mathbf{v} \in \mathcal{A}(\boldsymbol{\beta}^*)} \|\mathbf{v}\|_*$.

Notice that the bound in (19) does not explicitly depend on the sample size N. If we change to the ℓ_2-loss function, problem (18) will become:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|(-\boldsymbol{\beta}, 1)\|_* \\ \text{s.t.} \quad & \|(-\boldsymbol{\beta}, 1)'\mathbf{Z}\|_2 \leq \gamma_N, \\ & \boldsymbol{\beta} \in \mathcal{B}. \end{aligned}$$

The proof of Theorem 3.6 still applies with slight modification. We will find out that in the case of ℓ_2-loss, the estimation error bound takes the following form:

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \frac{2R\sqrt{N}\gamma_N}{\underline{\alpha}} \Psi(\boldsymbol{\beta}^*).$$

Similarly, the ℓ_∞-loss, which considers only the maximum absolute loss among the samples, turns (18) into:

$$\begin{aligned} \min_{\boldsymbol{\beta}} \quad & \|(-\boldsymbol{\beta}, 1)\|_* \\ \text{s.t.} \quad & \|(-\boldsymbol{\beta}, 1)'\mathbf{Z}\|_\infty \leq \gamma_N, \\ & \boldsymbol{\beta} \in \mathcal{B}. \end{aligned}$$

The corresponding bound becomes:

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \frac{2RN\gamma_N}{\underline{\alpha}} \Psi(\boldsymbol{\beta}^*).$$

We see that by using either ℓ_2 or ℓ_∞-loss, an explicit dependency on N is introduced. As a result, the estimation error bounds become worse. The reason is that for the ℓ_1-loss function, its dual norm operator only picks out the maximum absolute coordinate and thus avoids the dependence on the dimension, which in our case is the sample size (see Eq.(28)), whereas other norms, e.g., ℓ_2-norm, sum over all the coordinates and thus introduce a dependence on N.

As mentioned earlier, (19) provides a random upper bound, revealed in $\underline{\alpha}$ and γ_N, that depends on the randomness in $\mathbf{Z}$. We therefore would like to replace these two parameters by non-random quantities. The $\underline{\alpha}$ acts as the minimum eigenvalue of the matrix $\mathbf{ZZ}'$ restricted to a subspace of $\mathbb{R}^m$, and thus a proper substitute should be related to the minimum eigenvalue of the covariance matrix of $(\mathbf{x}, y)$, i.e., the $\mathbf{\Gamma}$ matrix (cf. Assumption G), given that $(\mathbf{x}, y)$ is zero mean. See Lemmas 3.7, 3.8 and 3.9 for the derivation.

Lemma 3.7 *Consider the set* $\mathcal{A}_{\mathbf{\Gamma}} = \{\mathbf{w} \in \mathbb{S}^m | \mathbf{\Gamma}^{-1/2}\mathbf{w} \in cone(\mathcal{A}(\boldsymbol{\beta}^*))\}$*, where* $\mathcal{A}(\boldsymbol{\beta}^*)$ *is defined as in Theorem 3.6, and* $\mathbf{\Gamma} = \mathbb{E}[(\mathbf{x}, y)(\mathbf{x}, y)']$*. Under Assumptions F and G, when the sample size* $N \geq C_1 \bar{\mu}^4 (w(\mathcal{A}_{\mathbf{\Gamma}}))^2$*, where* $\bar{\mu} = \mu\sqrt{\frac{1}{\lambda_{min}}}$*, and* $w(\mathcal{A}_{\mathbf{\Gamma}})$ *is the Gaussian width of* $\mathcal{A}_{\mathbf{\Gamma}}$*, with probability at least* $1 - \exp(-C_2 N/\bar{\mu}^4)$*, we have*

$$\mathbf{v}'\mathbf{ZZ}'\mathbf{v} \geq \frac{N}{2}\mathbf{v}'\mathbf{\Gamma}\mathbf{v}, \quad \forall\ \mathbf{v} \in \mathcal{A}(\boldsymbol{\beta}^*),$$

where C_1 *and* C_2 *are positive constants.*

Note that the sample size requirement stated in Lemma 3.7 depends on the Gaussian width of $\mathcal{A}_{\mathbf{\Gamma}}$, where $\mathcal{A}_{\mathbf{\Gamma}}$ relates to $\mathcal{A}(\boldsymbol{\beta}^*)$. The following lemma shows that their Gaussian widths are also related. This relation is built upon the square root of the eigenvalues of $\mathbf{\Gamma}$, which measures the extent to which $\mathcal{A}_{\mathbf{\Gamma}}$ expands $\mathcal{A}(\boldsymbol{\beta}^*)$.

Lemma 3.8 (Lemma 4 in Chen and Banerjee (2016)) *Let* μ_0 *be the* ψ_2*-norm of a standard Gaussian random vector* $\mathbf{g} \in \mathbb{R}^m$*, and* $\mathcal{A}_{\mathbf{\Gamma}}$*,* $\mathcal{A}(\boldsymbol{\beta}^*)$ *be defined as in Lemma 3.7. Then, under Assumption G,*

$$w(\mathcal{A}_{\mathbf{\Gamma}}) \leq C_3 \mu_0 \sqrt{\frac{\lambda_{max}}{\lambda_{min}}} \Big(w(\mathcal{A}(\boldsymbol{\beta}^*)) + 3 \Big),$$

for some positive constant C_3*.*

Combining Lemmas 3.7 and 3.8, and expressing the covariance matrix $\mathbf{\Gamma}$ using its eigenvalues, we arrive at the following result.

Corollary 3.9 *Under Assumptions F and G, and the conditions in Lemmas 3.7 and 3.8, when* $N \geq \bar{C}_1 \bar{\mu}^4 \mu_0^2 \cdot \frac{\lambda_{max}}{\lambda_{min}} \Big(w(\mathcal{A}(\boldsymbol{\beta}^*)) + 3 \Big)^2$*, with probability at least* $1 - \exp(-C_2 N/\bar{\mu}^4)$*,*

$$\mathbf{v}'\mathbf{ZZ}'\mathbf{v} \geq \frac{N\lambda_{min}}{2}, \quad \forall\ \mathbf{v} \in \mathcal{A}(\boldsymbol{\beta}^*),$$

where $\bar{C}_1$ *and* C_2 *are positive constants.*

Next, we derive the smallest possible value of γ_N such that $\boldsymbol{\beta}^*$ is feasible. The derivation uses the dual norm operator of the ℓ_1-loss, resulting in a bound that depends on the Gaussian width of the unit ball in the dual norm space ($\|\cdot\|_\infty$). See Lemma 3.10 for details.

Lemma 3.10 *Under Assumptions C and F, for any feasible $\boldsymbol{\beta}$, with probability at least $1 - C_4 \exp(-\frac{C_5^2 (w(\mathcal{B}_u))^2}{4\rho^2})$,*

$$\|(-\boldsymbol{\beta}, 1)'\mathbf{Z}\|_1 \leq C\mu \bar{B}_2 w(\mathcal{B}_u),$$

where $\mathcal{B}_u$ is the unit ball of norm $\|\cdot\|_\infty$, $\rho = \sup_{\mathbf{v}\in\mathcal{B}_u} \|\mathbf{v}\|_2$, and C_4, C_5, C positive constants.

We note that for other loss functions, e.g., the ℓ_2 and ℓ_∞ losses, similar results can be obtained, where $\mathcal{B}_u$ is defined to be the unit $\|\cdot\|_*^{\text{loss}}$-ball in $\mathbb{R}^m$, with $\|\cdot\|_*^{\text{loss}}$ being the dual norm of the loss. Combining Theorem 3.6, Corollary 3.9 and Lemma 3.10, we have the following main performance guarantee result that bounds the estimation bias of the solution to (18).

Theorem 3.11 *Under Assumptions A, C, D, E, F, G, and the conditions of Theorem 3.6, Corollary 3.9 and Lemma 3.10, when $N \geq \bar{C}_1 \bar{\mu}^4 \mu_0^2 \cdot \frac{\lambda_{max}}{\lambda_{min}} \Big(w(\mathcal{A}(\boldsymbol{\beta}^*)) + 3 \Big)^2$, with probability at least $1 - \exp(-C_2 N/\bar{\mu}^4) - C_4 \exp(-C_5^2 (w(\mathcal{B}_u))^2/(4\rho^2))$,*

$$\|\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*\|_2 \leq \frac{\bar{C} R \bar{B}_2 \mu}{N \lambda_{min}} w(\mathcal{B}_u) \Psi(\boldsymbol{\beta}^*). \tag{20}$$

From (20) we see that the bias is decreased as the sample size increases and the uncertainty embedded in $(\mathbf{x}, y)$ (revealed in R and μ) is reduced. The estimation error bound depends on the geometric structure of the true coefficients, defined using the dual norm space of the Wasserstein metric, the Gaussian width of the unit $\|\cdot\|_*^{\text{loss}}$-ball in $\mathbb{R}^m$, and the minimum eigenvalue of the covariance matrix of $(\mathbf{x}, y)$, with a convergence rate $1/N$ for the ℓ_1-loss we applied. As mentioned earlier, other loss functions may incur a dependence on N in the numerator of the bound, thus resulting in a slower convergence rate, which substantiates the benefit of using an ℓ_1-loss function.

4. Simulation Experiments on Synthetic Datasets

In this section we will explore the robustness of the Wasserstein formulation in terms of its *Absolute Deviation (AD)* loss function and the dual norm regularizer on the *extended regression coefficient* $(-\boldsymbol{\beta}, 1)$. Recall that our Wasserstein formulation is in the following form:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N} \sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon \|(-\boldsymbol{\beta}, 1)\|_*. \tag{21}$$

We will focus on the following three aspects of this formulation:

1. How to choose a proper norm $\|\cdot\|$ for the Wasserstein metric?
2. Why do we penalize the extended regression coefficient $(-\boldsymbol{\beta}, 1)$ rather than $\boldsymbol{\beta}$?
3. What is the advantage of the AD loss compared to the *Squared Residuals (SR)* loss?

To answer Question 1, we will connect the choice of $\|\cdot\|$ for the Wasserstein metric with the characteristics/structures of the data $(\mathbf{x}, y)$. Specifically, we will design two sets of experiments, one with a dense regression coefficient $\boldsymbol{\beta}^*$, where all coordinates of $\mathbf{x}$ play a role in determining the value of the response y, and another with a sparse $\boldsymbol{\beta}^*$ implying that only a few predictors are relevant/important in predicting y. Two Wasserstein formulations will be tested and compared, one induced by the $\|\cdot\|_2$ (Wasserstein ℓ_2), which leads to an ℓ_2-regularizer in (21), and the other one induced by the $\|\cdot\|_\infty$ (Wasserstein ℓ_∞) and resulting in an ℓ_1-regularizer in (21). Intuitively, and based on the past experience in implementing the regularization techniques, the Wasserstein ℓ_2 should outperform the Wasserstein ℓ_∞ in the dense setting, while in the sparse setting, the reverse is true. Researchers have well identified the sparsity inducing property of the ℓ_1-regularizer and provided a nice geometrical interpretation for it (Friedman et al., 2001). Here, we try to offer a different explanation from the perspective of the Wasserstein DRO formulation, through projecting the sparsity of $\boldsymbol{\beta}^*$ onto the $(\mathbf{x}, y)$ space and establishing a *sparse* distance metric that only extracts a subset of coordinates from $(\mathbf{x}, y)$ to measure the closeness between samples.

For the second question, we first note that if the Wasserstein metric is induced by the following metric s_c:

$$s_c(\mathbf{x}, y) = \|(\mathbf{x}, cy)\|_2,$$

for a positive constant c, then as $c \to \infty$, the resulting Wasserstein DRO formulation becomes:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N} \sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon \|\boldsymbol{\beta}\|_2,$$

which is the ℓ_2-regularized LAD. This can be proved by recognizing that $s_c(\mathbf{x}, y) = \|(\mathbf{x}, y)\|_{\mathbf{M}}$, with $\mathbf{M} \in \mathbb{R}^{m\times m}$ a diagonal matrix whose diagonal elements are $(1, \ldots, 1, c^2)$, and then applying (13). Alternatively, if we let

$$s_c(\mathbf{x}, y) = \|(\mathbf{x}, cy)\|_\infty,$$

it can be shown that as $c \to \infty$, the corresponding Wasserstein formulation becomes:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N} \sum_{i=1}^{N} |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon \|\boldsymbol{\beta}\|_1,$$

which is the ℓ_1-regularized LAD (see proof in the Appendix). It follows that regularizing over $\boldsymbol{\beta}$ implies an infinite transportation cost along y. In other words, for two data points $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$, if $y_1 \neq y_2$, then they are considered to be infinitely far away. By contrast, our Wasserstein formulation, which regularizes over the extended regression coefficient $(-\boldsymbol{\beta}, 1)$, stems from a finite cost along y that is equally weighted with $\mathbf{x}$. We will see the disadvantages of penalizing only $\boldsymbol{\beta}$ in the analysis of the experimental results.

To answer Question 3, we will compare against several commonly used regression models that employ the SR loss function, e.g., ridge regression (Hoerl and Kennard, 1970), LASSO (Tibshirani, 1996), and *Elastic Net (EN)* (Zou and Hastie, 2005). We will also compare against M-estimation (Huber, 1964, 1973), which uses a variant of the SR loss and is equivalent to solving a weighted least squares problem, where the weights are determined

by the residuals. These models will be compared under two different experimental setups, one involving adversarial perturbations in both $\mathbf{x}$ and y, and the other with perturbations only in $\mathbf{x}$. The purpose is to investigate the behavior of these approaches when the noise in y is substantially reduced. As shown by Fig. 1, compared to the SR loss, the AD loss is less vulnerable to large residuals, and hence, it is advantageous in the scenarios where large perturbations appear in y. We are interested in studying whether its performance is consistently good when the corruptions appear mainly in $\mathbf{x}$.

We next describe the data generation process. Each training sample has a probability q of being drawn from the outlying distribution, and a probability $1-q$ of being drawn from the true (clean) distribution. Given the true regression coefficient $\boldsymbol{\beta}^*$, we generate the training data as follows:

- Generate a uniform random variable on $[0,1]$. If it is no larger than $1-q$, generate a clean sample as follows:
  1. Draw the predictor $\mathbf{x} \in \mathbb{R}^{m-1}$ from the normal distribution $N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma}$ is the covariance matrix of $\mathbf{x}$, which is just the top left block of the matrix $\boldsymbol{\Gamma}$ in Assumption G. Specifically, $\boldsymbol{\Gamma} = \mathbb{E}[(\mathbf{x}, y)(\mathbf{x}, y)']$ is equal to
  $$\boldsymbol{\Gamma} = \begin{pmatrix} \boldsymbol{\Sigma} & \boldsymbol{\Sigma}\boldsymbol{\beta}^* \\ (\boldsymbol{\beta}^*)'\boldsymbol{\Sigma} & (\boldsymbol{\beta}^*)'\boldsymbol{\Sigma}\boldsymbol{\beta}^* + \sigma^2 \end{pmatrix},$$
  with σ^2 being the variance of the noise term. In our implementation, $\boldsymbol{\Sigma}$ has diagonal elements equal to 1 (unit variance) and off-diagonal elements equal to ρ, with ρ the correlation between predictors.
  2. Draw the response variable y from $N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2)$.
- Otherwise, depending on the experimental setup, generate an outlier that is either:
  - Abnormal in both $\mathbf{x}$ and y, with outlying distribution:
    1. $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(5\mathbf{e}, \mathbf{I})$, or $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(\mathbf{0}, 0.25\mathbf{I})$;
    2. $y \sim N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2) + 5\sigma$.
  - Abnormal only in $\mathbf{x}$:
    1. $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(5\mathbf{e}, \mathbf{I})$;
    2. $y \sim N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2)$.
- Repeat the above procedure for N times, where N is the size of the training set.

To test the generalization ability of various formulations, we generate a test dataset containing M samples from the clean distribution. It is worth noting that only clean samples are included in the test set, since we only care about the prediction accuracy on clean data points, and our estimator is supposed to be consistent with the clean distribution and stay away from the outlying one. We are interested in studying the performance of various methods as the following factors are varied:

- *Signal to Noise Ratio (SNR)*, defined as:
  $$\text{SNR} = \frac{(\boldsymbol{\beta}^*)'\boldsymbol{\Sigma}\boldsymbol{\beta}^*}{\sigma^2},$$
  which is equally spaced between 0.05 and 2 on a log scale.
- The correlation between predictors: ρ, which takes values in $(0.1, 0.2, \ldots, 0.9)$.

The performance metrics we use include:

- *Mean Squared Error (MSE)* on the test dataset, which is defined to be $\sum_{i=1}^{M}(y_i - \mathbf{x}_i'\hat{\boldsymbol{\beta}})^2/M$, with $\hat{\boldsymbol{\beta}}$ being the estimate of $\boldsymbol{\beta}^*$ obtained from the training set, and $(\mathbf{x}_i, y_i),\ i = 1, \ldots, M$, being the observations from the test dataset;
- *Relative Risk (RR)* of $\hat{\boldsymbol{\beta}}$ defined as:
  $$\text{RR}(\hat{\boldsymbol{\beta}}) \triangleq \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)'\boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)}{(\boldsymbol{\beta}^*)'\boldsymbol{\Sigma}\boldsymbol{\beta}^*}.$$
- *Relative Test Error (RTE)* of $\hat{\boldsymbol{\beta}}$ defined as:
  $$\text{RTE}(\hat{\boldsymbol{\beta}}) \triangleq \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)'\boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) + \sigma^2}{\sigma^2}.$$
- *Proportion of Variance Explained (PVE)* of $\hat{\boldsymbol{\beta}}$ defined as:
  $$\text{PVE}(\hat{\boldsymbol{\beta}}) \triangleq 1 - \frac{(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*)'\boldsymbol{\Sigma}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^*) + \sigma^2}{(\boldsymbol{\beta}^*)'\boldsymbol{\Sigma}\boldsymbol{\beta}^* + \sigma^2}.$$

For the metrics that evaluate the accuracy of the estimator, i.e., the RR, RTE and PVE, we list below two types of scores, one achieved by the best possible estimator $\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}^*$, called the perfect score, and the other one achieved by the null estimator $\hat{\boldsymbol{\beta}} = 0$, called the null score.

- RR: a perfect score is 0 and the null score is 1.
- RTE: a perfect score is 1 and the null score is SNR+1.
- PVE: a perfect score is $\frac{\text{SNR}}{\text{SNR}+1}$, and the null score is 0.

During the training process, all the regularization parameters are tuned on a separate validation dataset. Specifically, we divide all the N training samples into two sets, dataset 1 and dataset 2 (validation set). For a pre-specified range of values for the penalty parameters, dataset 1 is used to train the models and derive $\hat{\boldsymbol{\beta}}$, and the performance of $\hat{\boldsymbol{\beta}}$ is evaluated on dataset 2. We choose the regularization parameter that yields the minimum *Median Absolute Deviation (MAD)* on the validation set. Using MAD as a selection criterion serves to hedge against the potentially large noise in the validation samples. As to the range of values for the tuned parameters, we borrow ideas from Hastie et al. (2017), where the

LASSO was tuned over 50 values ranging from $\lambda_m = \|\mathbf{X}'\mathbf{y}\|_\infty$ to a small fraction of λ_m on a log scale, with $\mathbf{X} \in \mathbb{R}^{N\times(m-1)}$ the design matrix whose i-th row is $\mathbf{x}_i'$, and $\mathbf{y} = (y_1, \ldots, y_N)$ the response vector. In our experiments, this range is properly adjusted for procedures that use the AD loss. Specifically, for Wasserstein ℓ_2 and ℓ_∞, ℓ_1- and ℓ_2-regularized LAD, the range of values for the regularization parameter is:

$$\sqrt{\exp\Big(\mathrm{lin}\Big(\log(0.005 * \|\mathbf{X}'\mathbf{y}\|_\infty), \log(\|\mathbf{X}'\mathbf{y}\|_\infty), 50\Big)\Big)},$$

where $\mathrm{lin}(a, b, n)$ is a function that takes in scalars a, b and n (integer) and outputs a set of n values equally spaced between a and b; the exp function is applied elementwise to a vector. The square root operator is in consideration of the AD loss that is the square root of the SR loss if evaluated on a single sample.

The regularization coefficient ϵ in formulation (10), which is the radius of the Wasserstein ball, allows for a more efficient tuning procedure. It has been noted in Esfahani and Kuhn (2015) that for a large enough sample size, ϵ is inversely proportional to $N^{1/m}$. This proportionality could be used as a guidance on setting ϵ, where only the proportional factor needs to be tuned (using cross-validation or a separate validation dataset as described earlier). In our implementation, given the small size of the simulated datasets, we will still adopt the validation dataset approach to tune the regularization parameter.

4.1 Dense $\boldsymbol{\beta}^*$, outliers in both $\mathbf{x}$ and y

In this subsection, we choose a dense regression coefficient $\boldsymbol{\beta}^*$, set the intercept $\beta_0^* = 0.3$, and the coefficient for each predictor x_i to be $\beta_i^* = 0.5, i = 1, \ldots, 20$. The adversarial perturbations are present in both $\mathbf{x}$ and y. Specifically, the outlying distribution is described by:

1. $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(5\mathbf{e}, \mathbf{I})$;
2. $y \sim N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2) + 5\sigma$.

We generate 10 datasets consisting of $N = 100, M = 60$ observations. The probability of a training sample being drawn from the outlying distribution is $q = 30\%$. The mean values of the performance metrics (averaged over the 10 datasets), as we vary the SNR and the correlation between predictors, are shown in Figs. 3 and 4. Note that when SNR is varied, the correlation between predictors is set to 0.8 times a random noise uniformly distributed on the interval $[0.2, 0.4]$. When the correlation ρ is varied, the SNR is fixed to 0.5.

It can be seen that as the SNR decreases or the correlation between the predictors increases, the estimation problem becomes harder, and the performance of all approaches gets worse. In general the Wasserstein ℓ_2 achieves the best performance in terms of all four metrics. Specifically,

- It is better than the ℓ_2-regularized LAD, which assumes an infinite transportation cost along y.
- It is better than the Wasserstein ℓ_∞ and ℓ_1-regularized LAD which use the ℓ_1-regularizer.

- It is better than the approaches that use the SR loss function.

Empirically we have found out that in most cases, the approaches that use the AD loss, including the ℓ_1- and ℓ_2-regularized LAD, and the Wasserstein ℓ_∞ formulation, drive all the coordinates of $\boldsymbol{\beta}$ to zero, due to the relatively small magnitude of the AD loss compared to the norm of the coefficient, so that the regularizer dominates the solution. The approaches that use the SR loss, e.g., ridge regression and EN, do not exhibit such a problem, since the squared residuals weaken the dominance of the regularization term.

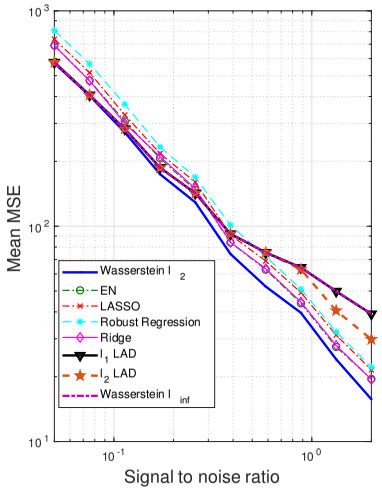
Overall the ℓ_2-regularizer outperforms the ℓ_1-regularizer, since the true regression coefficient is dense, which implies that a proper distance metric on the $(\mathbf{x}, y)$ space should take into account all the coordinates. From the perspective of the Wasserstein DRO framework, the ℓ_1-regularizer corresponds to an $\|\cdot\|_\infty$-based distance metric on the $(\mathbf{x}, y)$ space that only picks out the most influential coordinate to determine the closeness between data points, which in our case is not reasonable since every coordinate plays a role (reflected in the dense $\boldsymbol{\beta}^*$). In contrast, if $\boldsymbol{\beta}^*$ is sparse, using the $\|\cdot\|_\infty$ as a distance metric on $(\mathbf{x}, y)$ is more appropriate. A more detailed discussion of this will be presented in Sections 4.3 and 4.4.

4.2 Dense $\boldsymbol{\beta}^*$, outliers only in $\mathbf{x}$

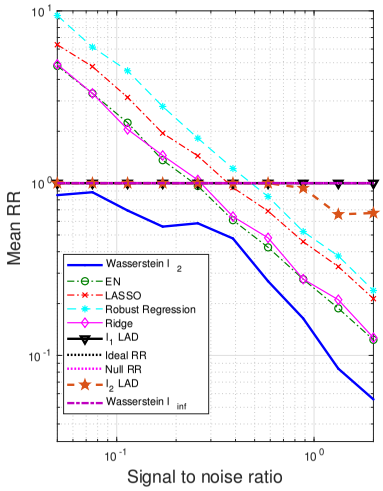
In this subsection we will experiment with the same $\boldsymbol{\beta}^*$ as in Section 4.1, but with perturbations only in $\mathbf{x}$, i.e., for a given $\mathbf{x}$ of the outlier, the corresponding y value is drawn in the same way as the clean samples. Our goal is to investigate the performance of the Wasserstein formulation when the response y is not subjected to large perturbations. The motivation for introducing the AD loss in the Wasserstein formulation is to hedge against large residuals, as illustrated in Fig. 1. We are interested in comparing the AD and SR loss functions when the residuals have moderate magnitudes.

Interestingly, we have observed that although the ℓ_1- and ℓ_2-regularized LAD, as well as the Wasserstein ℓ_∞ formulation, exhibit unsatisfactory performance, the Wasserstein ℓ_2, which shares the same loss function with them, is able to achieve a comparable performance with the best among all – EN and ridge regression (see Figs. 5 and 6). Notably, the ℓ_2-regularized LAD, which is just slightly different from our Wasserstein ℓ_2 formulation, shows a much worse performance. This is because the ℓ_2-regularized LAD implicitly assumes an infinite transportation cost along y, which gives zero tolerance to the variation in the response. For example, given two data points $(\mathbf{x}_1, y_1)$ and $(\mathbf{x}_2, y_2)$, as long as $y_1 \neq y_2$, the distance between them is infinity. Therefore, a reasonable amount of fluctuation, caused by the intrinsic randomness of y, would be overly exaggerated by the underlying metric used by the ℓ_2-regularized LAD. In contrast, our Wasserstein approach uses a proper notion of norm to evaluate the distance in the $(\mathbf{x}, y)$ space and is able to effectively distinguish abnormally high variations from moderate, acceptable noise.

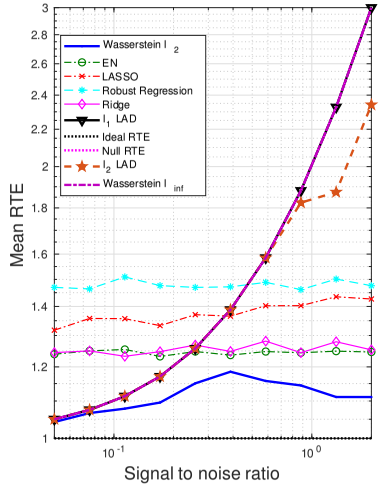
It is also worth noting that the formulations with the AD loss, e.g., ℓ_2- and ℓ_1-regularized LAD, and the Wasserstein ℓ_∞, perform worse than the approaches with the SR loss. One reasonable explanation is that the AD loss, introduced primarily for hedging against large perturbations in y, is less useful when the noise in y is moderate, in which case the sensitivity to response noise is needed. Although the AD loss is not a wise choice, penalizing



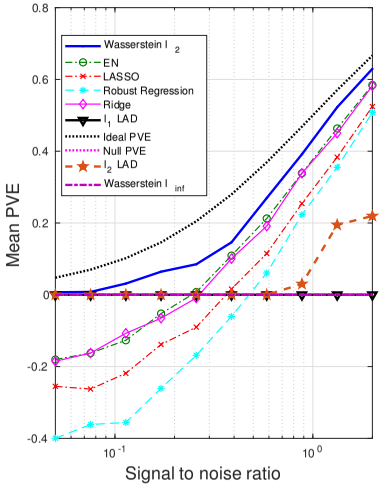
(a) Mean Squared Error.



(b) Relative risk.

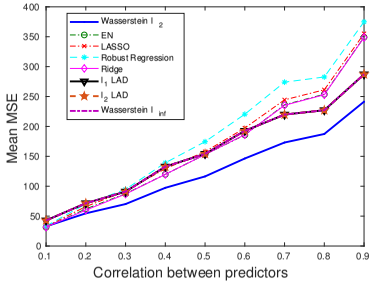


(c) Relative test error.

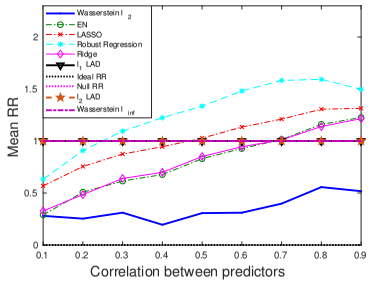


(d) Proportion of variance explained.

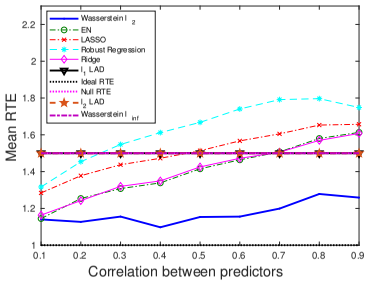
Figure 3: The impact of SNR on the performance metrics: dense $\boldsymbol{\beta}^*$, outliers in both $\mathbf{x}$ and y.



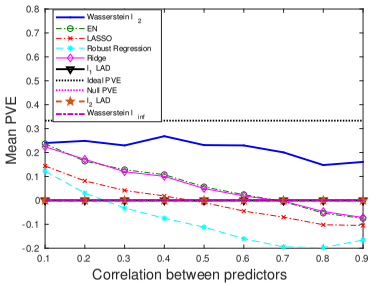
(a) Mean Squared Error.



(b) Relative risk.



(c) Relative test error.



(d) Proportion of variance explained.

Figure 4: The impact of predictor correlation on the performance metrics: dense $\boldsymbol{\beta}^*$, outliers in both $\mathbf{x}$ and y.

the extended coefficient vector $(-\boldsymbol{\beta}, 1)$ seems to make up, making the Wasserstein ℓ_2 a competitive method even when the perturbations appear only in $\mathbf{x}$.

4.3 Sparse $\boldsymbol{\beta}^*$, outliers in both x and y

In this subsection we will experiment with a sparse $\boldsymbol{\beta}^*$. The intercept is set to $\beta_0^* = 3$, and the coefficients for the 20 predictors are set to $\boldsymbol{\beta}^* = (0.05, 0, 0.006, 0, -0.007, 0, 0.008, 0, \ldots, 0)$. The adversarial perturbations are present in both $\mathbf{x}$ and y. Specifically, the distribution of outliers is characterized by:

1. $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(\mathbf{0}, 0.25\mathbf{I})$;
2. $y \sim N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2) + 5\sigma$.

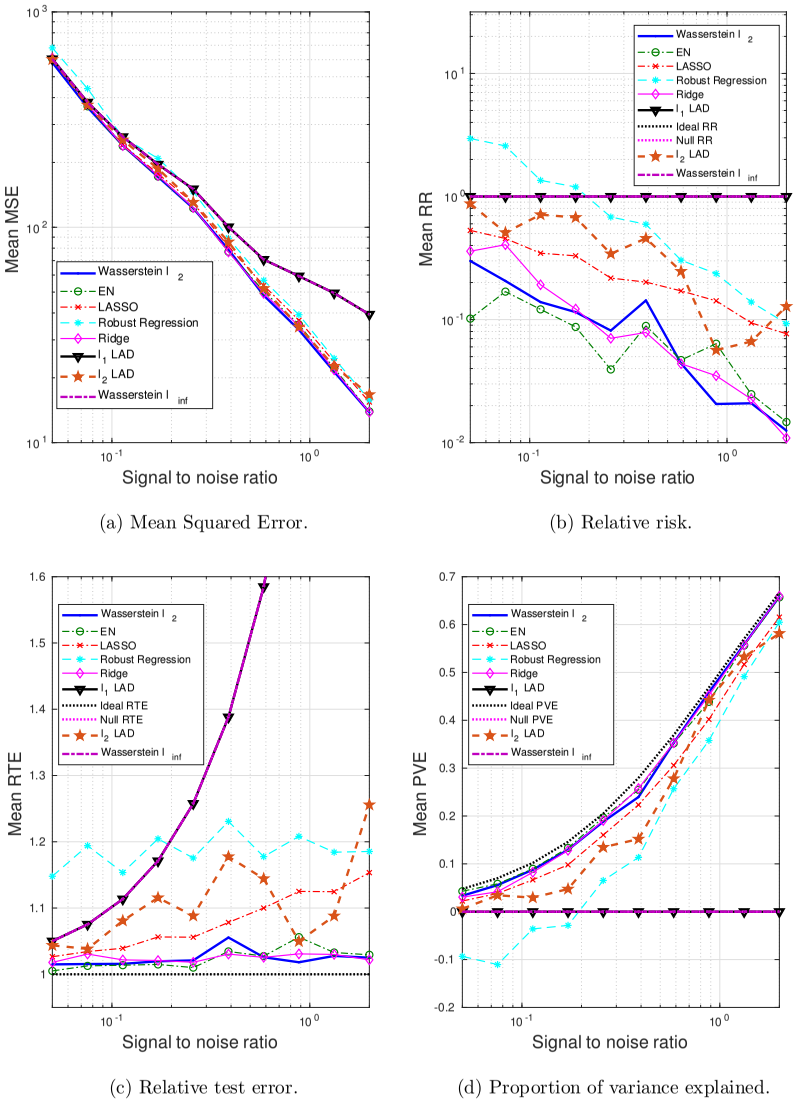
Our goal is to study the impact of the sparsity of $\boldsymbol{\beta}^*$ on the choice of the norm space for the Wasserstein metric. We know that the ℓ_1-regularizer works better than the ℓ_2-regularizer for sparse data, which has been validated by our results in Figs. 7 and 8. We will see that the Wasserstein ℓ_∞ formulation significantly outperforms the Wasserstein ℓ_2. An intuitively appealing interpretation for the sparsity inducing property of the ℓ_1-regularizer is made available by the Wasserstein DRO framework, which we explain as follows. The sparse regression coefficient $\boldsymbol{\beta}^*$ implies that only a few predictors are relevant to the regression model, and thus when measuring the distance in the $(\mathbf{x}, y)$ space, we need a metric that only extracts the subset of relevant predictors. The $\|\cdot\|_\infty$, which takes only the most influential coordinate of its argument, roughly serves this purpose. Compared to the $\|\cdot\|_2$ which takes into account all the coordinates, most of which are redundant due to the sparsity assumption, $\|\cdot\|_\infty$ results in a better performance, and hence, the Wasserstein ℓ_∞ formulation that stems from the $\|\cdot\|_\infty$ distance metric on $(\mathbf{x}, y)$ and induces the ℓ_1-regularizer is expected to outperform others.

We note that the ℓ_1-regularized LAD achieves similar performance to ours, since replacing $\|\boldsymbol{\beta}\|_1$ by $\|(-\boldsymbol{\beta}, 1)\|_1$ only adds a constant term to the objective function. The generalization performance (mean MSE) of the AD loss-based formulations is consistently better than those with the SR loss, since the AD loss is less affected by large perturbations in y. Also note that choosing a wrong norm for the Wasserstein metric, e.g., the Wasserstein ℓ_2, could lead to an enormous estimation error, whereas with a right norm space, we are guaranteed to outperform all others. Even when the SNR is very low, our performance is at least as good as the null estimator (see Fig. 7). Although EN and LASSO achieve similar performance to ours for moderate SNR values, they have a chance of performing even worse than the null estimator when there is little signal/information to learn from.

4.4 Sparse $\boldsymbol{\beta}^*$, outliers only in x

In this subsection, we will use the same sparse coefficient as in Section 4.3, but the perturbations are present only in $\mathbf{x}$. Specifically, for outliers, their predictors and responses are drawn from the following distributions:

1. $\mathbf{x} \sim N_{m-1}(\mathbf{0}, \boldsymbol{\Sigma}) + N_{m-1}(5\mathbf{e}, \mathbf{I})$;
2. $y \sim N(\mathbf{x}'\boldsymbol{\beta}^*, \sigma^2)$.



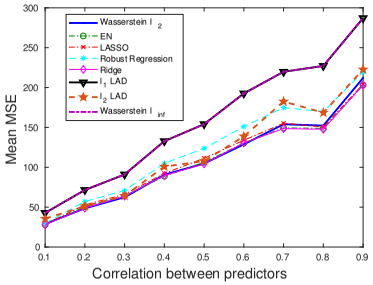
(a) Mean Squared Error.

(b) Relative risk.

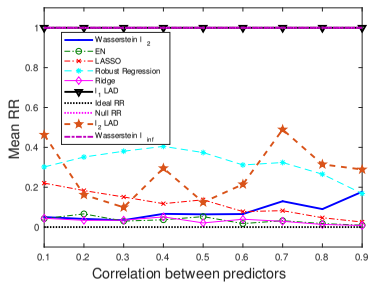
(c) Relative test error.

(d) Proportion of variance explained.

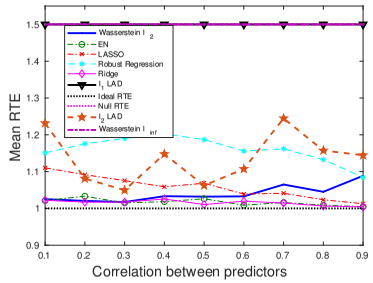
Figure 5: The impact of SNR on the performance metrics: dense $\boldsymbol{\beta}^*$, outliers only in $\mathbf{x}$.



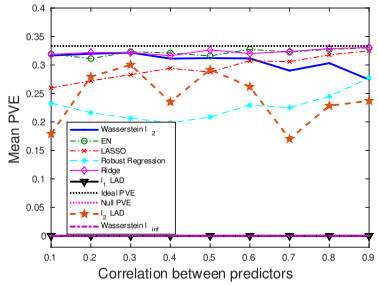
(a) Mean Squared Error.



(b) Relative risk.

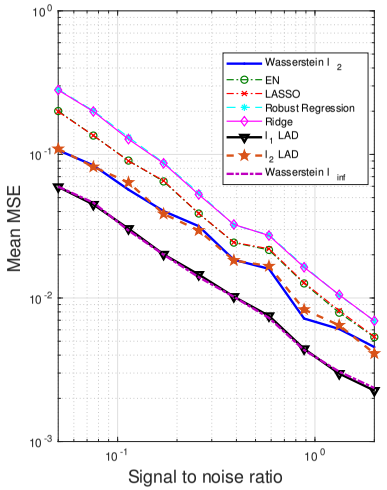


(c) Relative test error.

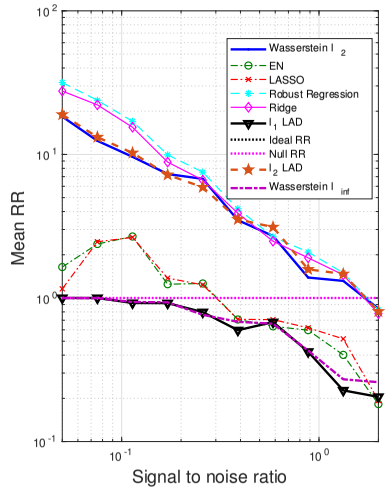


(d) Proportion of variance explained.

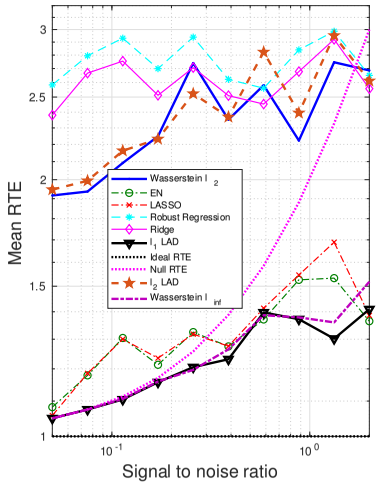
Figure 6: The impact of predictor correlation on the performance metrics: dense $\boldsymbol{\beta}^*$, outliers only in $\mathbf{x}$.



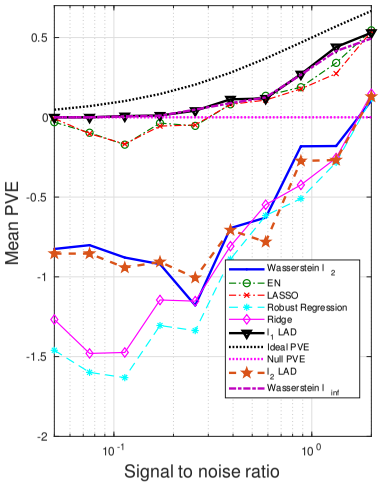
(a) Mean Squared Error.



(b) Relative risk.

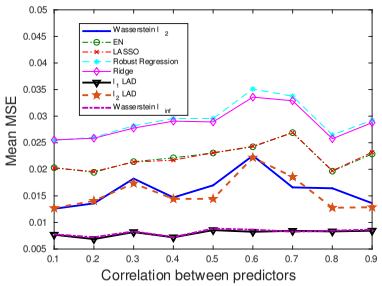


(c) Relative test error.

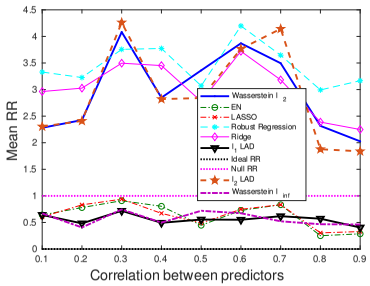


(d) Proportion of variance explained.

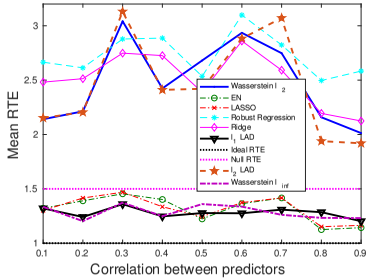
Figure 7: The impact of SNR on the performance metrics: sparse $\boldsymbol{\beta}^*$, outliers in both $\mathbf{x}$ and y.



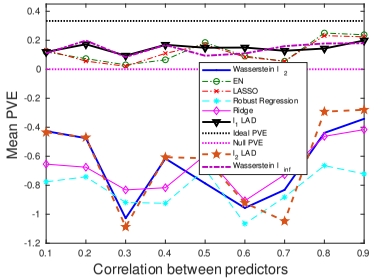
(a) Mean Squared Error.



(b) Relative risk.



(c) Relative test error.



(d) Proportion of variance explained.

Figure 8: The impact of predictor correlation on the performance metrics: sparse $\boldsymbol{\beta}^*$, outliers in both $\mathbf{x}$ and y.

Not surprisingly, the Wasserstein ℓ_∞ and the ℓ_1-regularized LAD achieve the best performance. Notice that in Section 4.3, where perturbations appear in both $\mathbf{x}$ and y, the AD loss-based formulations have smaller generalization and estimation errors than the SR loss-based formulations. When we reduce the variation in y, the SR loss seems superior to the AD loss, if we restrict attention to the improperly regularized (ℓ_2-regularizer) formulations (see Fig. 9). For the ℓ_1-regularized formulations, our Wasserstein ℓ_∞ formulation, as well as the ℓ_1-regularized LAD, is comparable with the EN and LASSO. Moreover, when there is little information to utilize (low SNR), EN and LASSO are worse than the null estimator, whereas our performance is at least as good as the null estimator.

We summarize below our main findings from all sets of experiments we have presented:

1. When a proper norm space is selected for the Wasserstein metric, the Wasserstein DRO formulation outperforms all others in terms of the generalization and estimation qualities.

2. Penalizing the extended regression coefficient $(-\boldsymbol{\beta}, 1)$ implicitly assumes a more reasonable distance metric on $(\mathbf{x}, y)$ and thus leads to a better performance.

3. The AD loss is remarkably superior to the SR loss when there is large variation in the response y.

4. The Wasserstein DRO formulation shows a more stable estimation performance than others when the correlation between predictors is varied.

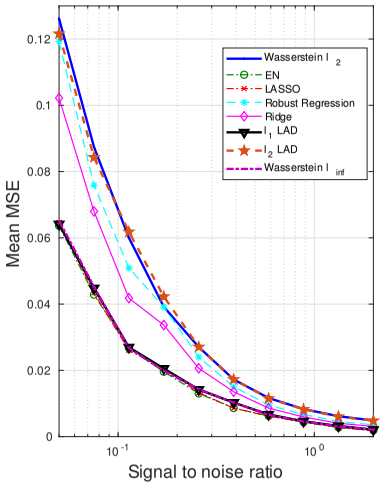
4.5 An outlier detection example

As an application, we consider an unlabeled two-class classification problem, where our goal is to identify the abnormal class of data points based on the predictor and response information using the Wasserstein formulation. We do not know a priori whether the samples are normal or abnormal, and thus classification models do not apply. The commonly used regression model for this type of problem is the M-estimation (Huber, 1964, 1973), against which we will compare in terms of the outlier detection capability.

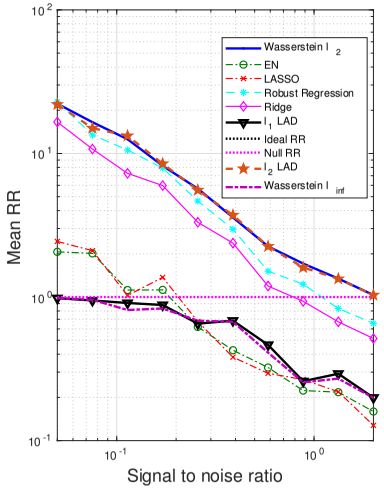
The data are generated in the same fashion as before. For clean samples, all predictors $x_1, \ldots, x_{30}$ come from a normal distribution with mean 7.5 and standard deviation 4.0. The response is a linear function of the predictors with $\beta_0^* = 0.3$, $\beta_1^* = \cdots = \beta_{30}^* = 0.5$, plus a Gaussian distributed noise term with zero mean and standard deviation σ. The outliers concentrate in a cloud that is randomly placed in the interior of the $\mathbf{x}$-space. Specifically, their predictors are uniformly distributed on $(u - 0.125, u + 0.125)$, where u is a uniform random variable on $(7.5 - 3 \times 4, 7.5 + 3 \times 4)$. The response values of the outliers are at a δ_R distance off the regression plane.

$$y = \beta_0^* + \beta_1^* x_1 + \cdots + \beta_{30}^* x_{30} + \delta_R.$$

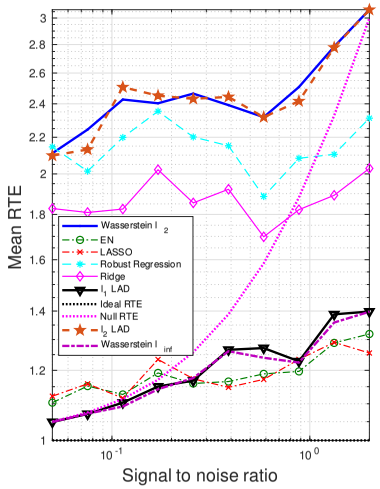
We will compare the performance of the Wasserstein ℓ_2 formulation (10) with the ℓ_1-regularized LAD and M-estimation with three cost functions – Huber (Huber, 1964, 1973), Talwar (Hinich and Talwar, 1975), and Fair (Fair, 1974). The performance metrics include



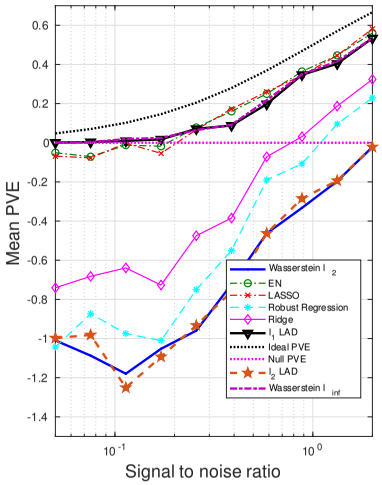
(a) Mean Squared Error.



(b) Relative risk.

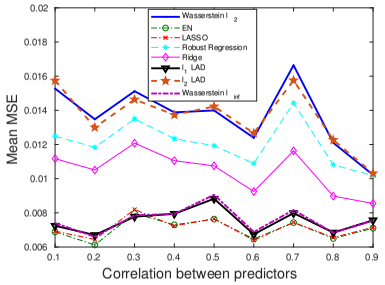


(c) Relative test error.

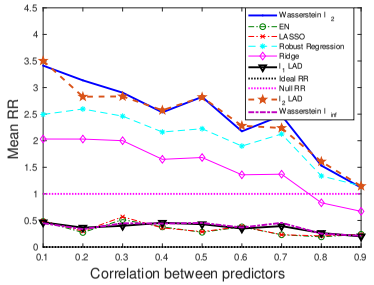


(d) Proportion of variance explained.

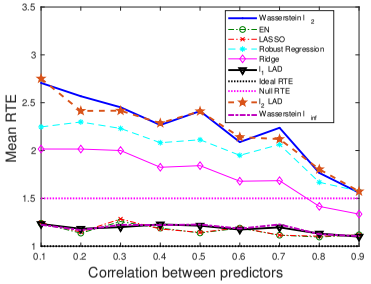
Figure 9: The impact of SNR on the performance metrics: sparse $\boldsymbol{\beta}^*$, outliers only in $\mathbf{x}$.



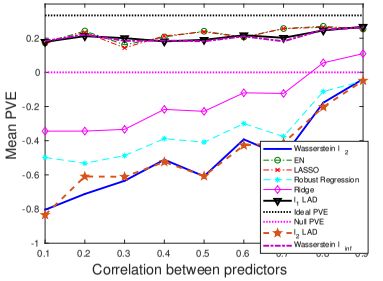
(a) Mean Squared Error.



(b) Relative risk.



(c) Relative test error.



(d) Proportion of variance explained.

Figure 10: The impact of predictor correlation on the performance metrics: sparse $\boldsymbol{\beta}^*$, outliers only in $\mathbf{x}$.

the *Receiver Operating Characteristic (ROC)* curve which plots the true positive rate against the false positive rate, and the related *Area Under Curve (AUC)*.

Notice that all the regression methods under consideration only generate an estimated regression coefficient. The identification of outliers is based on the residual and estimated standard deviation of the noise. Specifically,

$$\text{Outlier} = \begin{cases} \text{YES}, & \text{if } |\text{residual}| > \text{threshold} \times \hat{\sigma}, \\ \text{NO}, & \text{otherwise}, \end{cases}$$

where $\hat{\sigma}$ is the standard deviation of residuals in the entire training set. ROC curves are obtained through adjusting the threshold value.

The regularization parameters for Wasserstein DRO and regularized LAD are tuned using a separate validation set as done in previous sections. We would like to highlight a salient advantage of our approach reflected in its robustness w.r.t. the choice of ϵ. In Fig. 11 we plot the out-of-sample AUC as the radius ϵ (regularization parameter) varies, for the ℓ_2-induced Wasserstein DRO and the ℓ_1-regularized LAD. For the Wasserstein DRO curve, when ϵ is small, the Wasserstein ball contains the true distribution with low confidence and thus AUC is low. On the other hand, too large ϵ makes our solution overly conservative. Note that the robustness of our approach, indicated by the flatness of the Wasserstein DRO curve, constitutes another advantage, whereas the performance of LAD dramatically deteriorates once the regularizer deviates from the optimum. Moreover, the maximal achievable AUC for Wasserstein DRO is significantly higher than LAD.

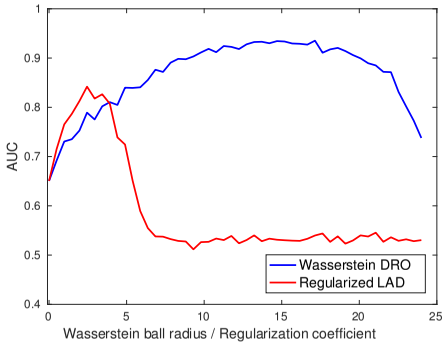
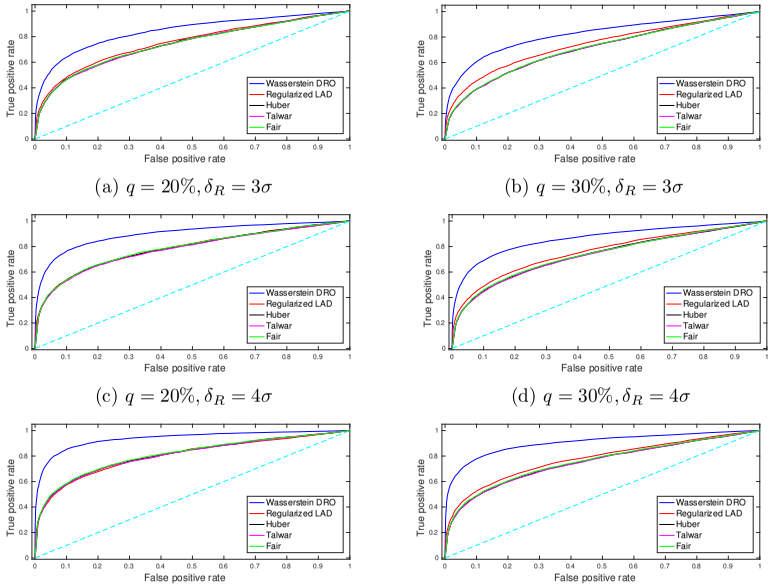


Figure 11: Out-of-sample AUC v.s. Wasserstein ball radius (regularization coefficient).

In Fig. 12 we show the ROC curves for different approaches, where q represents the percentage of outliers, and δ_R the outlying distance along y. We see that the Wasserstein DRO formulation consistently outperforms all other approaches, with its ROC curve lying well above others. In general, all approaches have better performance when the percentage of outliers is lower, and the outlying distance is larger. The approaches that use the AD loss function (e.g., Wasserstein DRO and regularized LAD) tend to outperform those that adopt

the SR loss (e.g., M-estimation which uses a variant of the SR loss). The superiority of our formulation could be attributed to the AD loss function, and the distributional robustness since we hedge against a family of plausible distributions, including the true distribution with high confidence. By contrast, M-estimation adopts an *Iteratively Reweighted Least Squares (IRLS)* procedure which assigns weights to data points based on the residuals from previous iterations, and then solves a weighted least squares estimation problem. With such an approach, there is a chance of exaggerating the influence of outliers while downplaying the importance of clean observations, especially when the initial residuals are obtained through *Ordinary Least Squares (OLS)*.



(a) $q = 20\%, \delta_R = 3\sigma$ (b) $q = 30\%, \delta_R = 3\sigma$

(c) $q = 20\%, \delta_R = 4\sigma$ (d) $q = 30\%, \delta_R = 4\sigma$

(e) $q = 20\%, \delta_R = 5\sigma$ (f) $q = 30\%, \delta_R = 5\sigma$

Figure 12: ROC curves for outliers in a randomly placed cloud, $N = 60, \sigma = 0.5$.

5. Conclusions

We presented a novel ℓ_1-loss based robust learning procedure using *Distributionally Robust Optimization (DRO)* in a linear regression framework, through which a delicate connection between the metric space on data and the regularization term has been established. The Wasserstein metric was utilized to construct the ambiguity set and a tractable reformulation was derived. It is worth noting that the linear law assumption does not necessarily limit the applicability of our model. In fact, by appropriately pre-processing the data, one can

often find a roughly linear relationship between the response and transformed explanatory variables. Our Wasserstein formulation incorporates a class of models whose specific form depends on the norm space that the Wasserstein metric is defined on. We provide out-of-sample generalization guarantees, and bound the estimation bias of the general formulation. Extensive numerical examples demonstrate the superiority of the Wasserstein formulation and shed light on the advantages of the ℓ_1-loss, the implication of the regularizer, and the selection of the norm space for the Wasserstein metric. We also presented an outlier detection example as an application of this robust learning procedure. A remarkable advantage of our approach rests in its flexibility to adjust the form of the regularizer based on the characteristics of the data.

Acknowledgments

Research partially supported by the NSF under grants DMS-1664644, CNS-1645681, CCF-1527292, and IIS-1237022, by the ARO under grant W911NF-12-1-0390, by the ONR under grant MURI N00014-16-1-2832, by the NIH under grant 1UL1TR001430 to the Clinical & Translational Science Institute at Boston University, by the Boston University Digital Health Initiative and the Center for Information and Systems Engineering, and by the joint Boston University and Brigham & Women's Hospital program in Engineering and Radiology. We thank Jenifer Siegelman and Vladimir Valtchinov for useful motivating discussions. We also thank the Editor and an anonymous reviewer whose comments helped us improve and better position this work.

Appendix A. Omitted Definitions and Proofs

This section includes proofs for the theorems and lemmas, in the order they appear in the paper.

A.1 Proof of Theorem 2.1

Proof We will adopt the notation $\mathbf{z} \triangleq (\mathbf{x}, y), \tilde{\boldsymbol{\beta}} \triangleq (-\boldsymbol{\beta}, 1)$ for ease of analysis. First rewrite $\kappa(\boldsymbol{\beta})$ as:

$$\kappa(\boldsymbol{\beta}) = \sup\Big\{\|\boldsymbol{\theta}\|_* : \sup_{\mathbf{z}|\mathbf{z}'\tilde{\boldsymbol{\beta}}\geq 0} \{(\boldsymbol{\theta}-\tilde{\boldsymbol{\beta}})'\mathbf{z}\} < \infty, \quad \sup_{\mathbf{z}|\mathbf{z}'\tilde{\boldsymbol{\beta}}\leq 0} \{(\boldsymbol{\theta}+\tilde{\boldsymbol{\beta}})'\mathbf{z}\} < \infty\Big\}.$$

Consider now the two linear optimization problems A and B:

$$\text{Problem A:} \quad \begin{array}{ll} \max & (\boldsymbol{\theta}-\tilde{\boldsymbol{\beta}})'\mathbf{z} \\ \text{s.t.} & \mathbf{z}'\tilde{\boldsymbol{\beta}} \geq 0. \end{array}$$

$$\text{Problem B:} \quad \begin{array}{ll} \max & (\boldsymbol{\theta}+\tilde{\boldsymbol{\beta}})'\mathbf{z} \\ \text{s.t.} & \mathbf{z}'\tilde{\boldsymbol{\beta}} \leq 0. \end{array}$$

Form the dual problems using dual variables r_A and r_B, respectively:

$$\text{Dual-A:} \quad \begin{array}{ll} \min & 0 \cdot r_A \\ \text{s.t.} & \tilde{\boldsymbol{\beta}} r_A = \boldsymbol{\theta} - \tilde{\boldsymbol{\beta}}, \\ & r_A \leq 0, \end{array}$$

$$\text{Dual-B:} \quad \begin{array}{ll} \min & 0 \cdot r_B \\ \text{s.t.} & \tilde{\boldsymbol{\beta}} r_B = \boldsymbol{\theta} + \tilde{\boldsymbol{\beta}}, \\ & r_B \geq 0. \end{array}$$

We want to find the set of $\boldsymbol{\theta}$ such that the optimal values of problems A and B are finite. Then, Dual-A and Dual-B need to have non-empty feasible sets, which implies the following two conditions:

$$\exists\, r_A \leq 0, \quad \text{s.t.} \quad \tilde{\boldsymbol{\beta}} r_A = \boldsymbol{\theta} - \tilde{\boldsymbol{\beta}}, \tag{22}$$

$$\exists\, r_B \geq 0, \quad \text{s.t.} \quad \tilde{\boldsymbol{\beta}} r_B = \boldsymbol{\theta} + \tilde{\boldsymbol{\beta}}. \tag{23}$$

For all i with $\tilde{\beta}_i \leq 0$, (22) implies $\theta_i - \tilde{\beta}_i \geq 0$ and (23) implies $\theta_i \leq -\tilde{\beta}_i$. On the other hand, for all j with $\tilde{\beta}_j \geq 0$, (22) and (23) imply $-\tilde{\beta}_j \leq \theta_j \leq \tilde{\beta}_j$. It is not hard to conclude that:

$$|\theta_i| \leq |\tilde{\beta}_i|, \quad \forall\, i.$$

It follows,

$$\kappa(\boldsymbol{\beta}) = \sup\{\|\boldsymbol{\theta}\|_* : |\theta_i| \leq |\tilde{\beta}_i|,\ \forall i\} = \|\tilde{\boldsymbol{\beta}}\|_*.$$

■

A.2 Proof of Lemma 3.2

Proof Suppose that $\sigma_1, \ldots, \sigma_N$ are i.i.d. uniform random variables on $\{1, -1\}$. Then, by the definition of the Rademacher complexity and Lemma 3.1,

$$\begin{aligned} \mathcal{R}_N(\mathcal{H}) &= \mathbb{E}\left[\sup_{h\in\mathcal{H}} \frac{2}{N}\left|\sum_{i=1}^N \sigma_i h_{\boldsymbol{\beta}}(\mathbf{x}_i, y_i)\right| \middle| (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\right] \\ &\leq \frac{2\bar{B}R}{N}\mathbb{E}\left[\left|\sum_{i=1}^N \sigma_i\right|\right] \\ &\leq \frac{2\bar{B}R}{N}\mathbb{E}\left[\sqrt{\sum_{i=1}^N \sigma_i^2}\,\right] \\ &= \frac{2\bar{B}R}{\sqrt{N}}. \end{aligned}$$

■

A.3 Proof of Theorem 3.3

Proof We use Theorem 8 in Bartlett and Mendelson (2002), setting the following correspondences with the notation used there: $\mathcal{L}(\mathbf{x}, y) = \phi(\mathbf{x}, y) = |y - \mathbf{x}'\boldsymbol{\beta}|$. This yields the

bound (14) on the expected loss. For Eq. (15), we apply Markov's inequality to obtain:

$$\mathbb{P}\Big(|y-\mathbf{x}'\hat{\boldsymbol{\beta}}| \geq \frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\zeta\Big) \leq \frac{\mathbb{E}[|y-\mathbf{x}'\hat{\boldsymbol{\beta}}|]}{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\zeta} \leq \frac{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\frac{2\bar{B}R}{\sqrt{N}}+\bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}}{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\zeta}.$$

■

A.4 Proof of Corollary 3.4

Proof The percentage difference requirement can be translated into:

$$\frac{2}{\sqrt{N}}+\sqrt{\frac{8\log(2/\delta)}{N}} \leq \tau,$$

from which (16) can be easily derived. ■

A.5 Proof of Corollary 3.5

Proof Based on Theorem 3.3, we just need the following inequality to hold:

$$\frac{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\frac{2\bar{B}R}{\sqrt{N}}+\bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}}{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\gamma\bar{B}R} \leq \tau,$$

which is equivalent to:

$$\frac{\gamma\bar{B}R-\frac{2\bar{B}R}{\sqrt{N}}-\bar{B}R\sqrt{\frac{8\log(2/\delta)}{N}}}{\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}|+\gamma\bar{B}R} \geq 1-\tau. \tag{24}$$

We cannot obtain a lower bound for N by directly solving (24) since N appears in a summation operator. A proper relaxation to (24) is:

$$\frac{\gamma-\frac{2}{\sqrt{N}}-\sqrt{\frac{8\log(2/\delta)}{N}}}{1+\gamma} \geq 1-\tau, \tag{25}$$

due to the fact that $\frac{1}{N}\sum_{i=1}^{N}|y_i-\mathbf{x}_i'\hat{\boldsymbol{\beta}}| \leq \bar{B}R$. By solving (25), we obtain (17). ■

A.6 Sub-Gaussian Random Variables and Gaussian Width

Definition 1 (Sub-Gaussian random variable) *A random variable z is sub-Gaussian if the ψ_2-norm defined below is finite, i.e.,*

$$\|z\|_{\psi_2} \triangleq \sup_{q\geq 1}\frac{\mathbb{E}|z|^q}{\sqrt{q}} < +\infty.$$

An equivalent property for sub-Gaussian random variables is that their tail distribution decays as fast as a Gaussian, namely,

$$\mathbb{P}(|z|\geq t) \leq 2\exp\{-t^2/C^2\}, \quad \forall t\geq 0,$$

for some constant C.

A random vector $\mathbf{z}\in\mathbb{R}^m$ is sub-Gaussian if $\mathbf{z}'\mathbf{u}$ is sub-Gaussian for any $\mathbf{u}\in\mathbb{R}^m$. The ψ_2-norm of a vector $\mathbf{z}$ is defined as:

$$\|\mathbf{z}\|_{\psi_2} \triangleq \sup_{\mathbf{u}\in\mathbb{S}^m}\big\|\|\mathbf{z}'\mathbf{u}\|\big\|_{\psi_2},$$

where $\mathbb{S}^m$ denotes the unit sphere in the m-dimensional Euclidean space. For the properties of sub-Gaussian random variables/vectors, please refer to the book by Vershynin (2017).

Definition 2 (Gaussian width) *For any set $\mathcal{A}\subseteq\mathbb{R}^m$, its Gaussian width is defined as:*

$$w(\mathcal{A}) \triangleq \mathbb{E}\Big[\sup_{\mathbf{u}\in\mathcal{A}}\mathbf{u}'\mathbf{g}\Big], \tag{26}$$

where $\mathbf{g}\sim\mathcal{N}(\mathbf{0},\mathbf{I})$ is an m-dimensional standard Gaussian random vector.

A.7 Proof of Theorem 3.6

In all the following proofs related to Section 3.2, we will adopt the notation $\mathbf{z}\triangleq(\mathbf{x},y)$, $\mathbf{z}_i\triangleq(\mathbf{x}_i,y_i)$, $\tilde{\boldsymbol{\beta}}\triangleq(-\boldsymbol{\beta},1)$, $\tilde{\boldsymbol{\beta}}_{\text{est}}\triangleq(-\hat{\boldsymbol{\beta}},1)$, $\tilde{\boldsymbol{\beta}}_{\text{true}}\triangleq(-\boldsymbol{\beta}^*,1)$ for ease of exposition.

Proof Since both $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^*$ are feasible (the latter due to Assumption E), we have:

$$\|\mathbf{Z}'\tilde{\boldsymbol{\beta}}_{\text{est}}\|_1 \leq \gamma_N,$$
$$\|\mathbf{Z}'\tilde{\boldsymbol{\beta}}_{\text{true}}\|_1 \leq \gamma_N,$$

from which we derive that $\|\mathbf{Z}'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})\|_1 \leq 2\gamma_N$. Since $\hat{\boldsymbol{\beta}}$ is an optimal solution to (18) and $\boldsymbol{\beta}^*$ a feasible solution, it follows that $\|\tilde{\boldsymbol{\beta}}_{\text{est}}\|_* \leq \|\tilde{\boldsymbol{\beta}}_{\text{true}}\|_*$. This implies that $\boldsymbol{\nu}=\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}$ satisfies the condition $\|\tilde{\boldsymbol{\beta}}_{\text{true}}+\mathbf{v}\|_* \leq \|\tilde{\boldsymbol{\beta}}_{\text{true}}\|_*$ included in the definition of $\mathcal{A}(\boldsymbol{\beta}^*)$ and, furthermore, $(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})/\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_2 \in \mathcal{A}(\boldsymbol{\beta}^*)$. Together with Assumption D, this yields

$$(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})'\mathbf{Z}\mathbf{Z}'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}) \geq \underline{\alpha}\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_2^2. \tag{27}$$

On the other hand, from the Cauchy-Schwarz inequality:

$$\begin{aligned}(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})'\mathbf{Z}\mathbf{Z}'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}) &\leq \|\mathbf{Z}'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})\|_1\|\mathbf{Z}'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})\|_\infty \\ &\leq 2\gamma_N\max_i|\mathbf{z}_i'(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})| \\ &\leq 2\gamma_N\max_i\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_*\|\mathbf{z}_i\| \\ &\leq 2R\gamma_N\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_*.\end{aligned} \tag{28}$$

Combining (27) and (28), we have:

$$\begin{aligned}\|\hat{\boldsymbol{\beta}}-\boldsymbol{\beta}^*\|_2 &= \|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_2\\ &\le \frac{2R\gamma_N}{\underline{\alpha}}\frac{\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_*}{\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_2}\\ &\le \frac{2R\gamma_N}{\underline{\alpha}}\Psi(\boldsymbol{\beta}^*),\end{aligned}$$

where the last step follows from the fact that $(\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}})/\|\tilde{\boldsymbol{\beta}}_{\text{est}}-\tilde{\boldsymbol{\beta}}_{\text{true}}\|_2 \in \mathcal{A}(\boldsymbol{\beta}^*)$. ■

A.8 Proof of Lemma 3.7

Proof Define $\hat{\boldsymbol{\Gamma}}=\frac{1}{N}\sum_{i=1}^N \mathbf{z}_i\mathbf{z}_i'$. Consider the set of functions $\mathcal{F}=\{f_{\mathbf{w}}(\mathbf{z})=\mathbf{z}'\boldsymbol{\Gamma}^{-1/2}\mathbf{w} | \mathbf{w}\in \mathcal{A}_{\boldsymbol{\Gamma}}\}$. Then, for any $f_{\mathbf{w}}\in\mathcal{F}$,

$$\begin{aligned}\mathbb{E}[f_{\mathbf{w}}^2] &= \mathbb{E}[\mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\mathbf{z}\mathbf{z}'\boldsymbol{\Gamma}^{-1/2}\mathbf{w}]\\ &= \mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\mathbb{E}[\mathbf{z}\mathbf{z}']\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\\ &= \mathbf{w}'\mathbf{w}\\ &= 1,\end{aligned}$$

where we used $\boldsymbol{\Gamma}=\mathbb{E}[\mathbf{z}\mathbf{z}']$ and the fact that $\mathbf{w}\in\mathcal{A}_{\boldsymbol{\Gamma}}$.

For any $f_{\mathbf{w}}\in\mathcal{F}$ we have

$$\begin{aligned}|||f_{\mathbf{w}}|||_{\psi_2} &= \left|\left|\left|\mathbf{z}'\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\right|\right|\right|_{\psi_2}\\ &= \left|\left|\left|\mathbf{z}'\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\right|\right|\right|_{\psi_2}\frac{\|\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\|_2}{\|\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\|_2}\\ &= \left|\left|\left|\mathbf{z}'\frac{\boldsymbol{\Gamma}^{-1/2}\mathbf{w}}{\|\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\|_2}\right|\right|\right|_{\psi_2}\|\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\|_2\\ &\le \mu\sqrt{\mathbf{w}'\boldsymbol{\Gamma}^{-1}\mathbf{w}}\\ &\le \mu\sqrt{\frac{1}{\lambda_{\min}}\|\mathbf{w}\|_2^2}\\ &= \mu\sqrt{\frac{1}{\lambda_{\min}}} = \bar{\mu},\end{aligned}$$

where the first inequality used Assumption F and the second inequality used Assumption G.

Applying Theorem D from Mendelson et al. (2007), for any $\theta>0$ and when

$$\tilde{C}_1\bar{\mu}\gamma_2(\mathcal{F},|||\cdot|||_{\psi_2})\le\theta\sqrt{N},$$

with probability at least $1-\exp(-\tilde{C}_2\theta^2N/\bar{\mu}^4)$ we have

$$\begin{aligned}\sup_{f_{\mathbf{w}}\in\mathcal{F}}\left|\frac{1}{N}\sum_{i=1}^N f_{\mathbf{w}}^2(\mathbf{z}_i)-\mathbb{E}[f_{\mathbf{w}}^2]\right| &= \sup_{f_{\mathbf{w}}\in\mathcal{F}}\left|\frac{1}{N}\sum_{i=1}^N \mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\mathbf{z}_i\mathbf{z}_i'\boldsymbol{\Gamma}^{-1/2}\mathbf{w}-1\right|\\ &= \sup_{\mathbf{w}\in\mathcal{A}_{\boldsymbol{\Gamma}}}\left|\mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\hat{\boldsymbol{\Gamma}}\boldsymbol{\Gamma}^{-1/2}\mathbf{w}-1\right|\\ &\le \theta,\end{aligned}\tag{29}$$

where $\tilde{C}_1$ is some positive constant and $\gamma_2(\mathcal{F},|||\cdot|||_{\psi_2})$ is defined in Mendelson et al. (2007) as a measure of the size of the set $\mathcal{F}$ with respect to the metric $|||\cdot|||_{\psi_2}$. Using $\theta=1/2$, and properties of $\gamma_2(\mathcal{F},|||\cdot|||_{\psi_2})$ outlined in Chen and Banerjee (2016), we can set N to satisfy

$$\begin{aligned}\tilde{C}_1\bar{\mu}\gamma_2(\mathcal{F},|||\cdot|||_{\psi_2}) &\le \tilde{C}_1\bar{\mu}^2\gamma_2(\mathcal{A}_{\boldsymbol{\Gamma}},\|\cdot\|_2)\\ &\le \tilde{C}_1\bar{\mu}^2C_0w(\mathcal{A}_{\boldsymbol{\Gamma}})\\ &\le \frac{1}{2}\sqrt{N},\end{aligned}$$

for some positive constant C_0, where we used Eq. (44) in Chen and Banerjee (2016). This implies

$$N\ge C_1\bar{\mu}^4(w(\mathcal{A}_{\boldsymbol{\Gamma}}))^2$$

for some positive constant C_1. Thus, for such N and with probability at least $1-\exp(-C_2N/\bar{\mu}^4)$, for some positive constant C_2, (29) holds with $\theta=1/2$. This implies that for all $\mathbf{w}\in\mathcal{A}_{\boldsymbol{\Gamma}}$,

$$\left|\mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\hat{\boldsymbol{\Gamma}}\boldsymbol{\Gamma}^{-1/2}\mathbf{w}-1\right|\le\frac{1}{2}$$

or

$$\mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\hat{\boldsymbol{\Gamma}}\boldsymbol{\Gamma}^{-1/2}\mathbf{w}\ge\frac{1}{2}=\frac{1}{2}\mathbf{w}'\boldsymbol{\Gamma}^{-1/2}\boldsymbol{\Gamma}\boldsymbol{\Gamma}^{-1/2}\mathbf{w}.$$

By the definition of $\mathcal{A}_{\boldsymbol{\Gamma}}$, for any $\mathbf{v}\in\mathcal{A}(\boldsymbol{\beta}^*)$,

$$\mathbf{v}'\hat{\boldsymbol{\Gamma}}\mathbf{v}\ge\frac{1}{2}\mathbf{v}'\boldsymbol{\Gamma}\mathbf{v}.$$

Noting that $\hat{\boldsymbol{\Gamma}}=(1/N)\mathbf{Z}\mathbf{Z}'$ yields the desired result. ■

A.9 Proof of Lemma 3.8

We follow the proof of Lemma 4 in Chen and Banerjee (2016), adapted to our setting. We include all key steps for completeness.

Proof Recall the definition of the Gaussian width $w(\mathcal{A}_{\boldsymbol{\Gamma}})$ (cf. (26)):

$$w(\mathcal{A}_{\boldsymbol{\Gamma}})=\mathbb{E}\Big[\sup_{\mathbf{u}\in\mathcal{A}_{\boldsymbol{\Gamma}}}\mathbf{u}'\mathbf{g}\Big],$$

where $\mathbf{g} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. We have:

$$\begin{aligned}\sup_{\mathbf{w}\in\mathcal{A}_{\mathbf{\Gamma}}} \mathbf{w}'\mathbf{g} &= \sup_{\mathbf{w}\in\mathcal{A}_{\mathbf{\Gamma}}} \mathbf{w}'\mathbf{\Gamma}^{-1/2}\mathbf{\Gamma}^{1/2}\mathbf{g} \\ &= \sup_{\mathbf{w}\in\mathcal{A}_{\mathbf{\Gamma}}} \|\mathbf{\Gamma}^{-1/2}\mathbf{w}\|_2 \frac{\mathbf{w}'\mathbf{\Gamma}^{-1/2}}{\|\mathbf{\Gamma}^{-1/2}\mathbf{w}\|_2}\mathbf{\Gamma}^{1/2}\mathbf{g} \\ &\le \sqrt{\frac{1}{\lambda_{\min}}} \sup_{\mathbf{v}\in\mathrm{cone}(\mathcal{A}(\boldsymbol{\beta}^*))\cap\mathbb{B}^m} \mathbf{v}'\mathbf{\Gamma}^{1/2}\mathbf{g},\end{aligned}$$

where $\mathbb{B}^m$ is the unit ball in the m-dimensional Euclidean space and the inequality used Assumption G and the fact that $\mathbf{w}'\mathbf{\Gamma}^{-1/2}/\|\mathbf{\Gamma}^{-1/2}\mathbf{w}\|_2 \in \mathbb{B}^m$ and $\mathbf{w} \in \mathcal{A}_{\mathbf{\Gamma}}$.

Define $\mathcal{T} = \mathrm{cone}(\mathcal{A}(\boldsymbol{\beta}^*)) \cap \mathbb{B}^m$, and consider the stochastic process $\{S_{\mathbf{v}} = \mathbf{v}'\mathbf{\Gamma}^{1/2}\mathbf{g}\}_{\mathbf{v}\in\mathcal{T}}$. For any $\mathbf{v}_1, \mathbf{v}_2 \in \mathcal{T}$,

$$\begin{aligned}\|\!|S_{\mathbf{v}_1} - S_{\mathbf{v}_2}|\!\|_{\psi_2} &= \Big|\!\Big|\!\Big|(\mathbf{v}_1 - \mathbf{v}_2)'\mathbf{\Gamma}^{1/2}\mathbf{g}\Big|\!\Big|\!\Big|_{\psi_2} \\ &= \|\mathbf{\Gamma}^{1/2}(\mathbf{v}_1 - \mathbf{v}_2)\|_2 \Big|\!\Big|\!\Big|\frac{(\mathbf{v}_1 - \mathbf{v}_2)'\mathbf{\Gamma}^{1/2}\mathbf{g}}{\|\mathbf{\Gamma}^{1/2}(\mathbf{v}_1 - \mathbf{v}_2)\|_2}\Big|\!\Big|\!\Big|_{\psi_2} \\ &\le \|\mathbf{\Gamma}^{1/2}(\mathbf{v}_1 - \mathbf{v}_2)\|_2 \sup_{\mathbf{u}\in\mathbb{S}^m} \|\!|\mathbf{u}'\mathbf{g}|\!\|_{\psi_2} \\ &= \mu_0\|\mathbf{\Gamma}^{1/2}(\mathbf{v}_1 - \mathbf{v}_2)\|_2 \\ &\le \mu_0\sqrt{\lambda_{\max}}\|\mathbf{v}_1 - \mathbf{v}_2\|_2,\end{aligned}$$

where the last step used Assumption G.

Then, by the tail behavior of sub-Gaussian random variables (see Hoeffding bound, Thm. 2.6.2 in (Vershynin, 2017)), we have:

$$\mathbb{P}(|S_{\mathbf{v}_1} - S_{\mathbf{v}_2}| \ge \delta) \le 2\exp\Big(-\frac{C_{01}\delta^2}{\mu_0^2\lambda_{\max}\|\mathbf{v}_1 - \mathbf{v}_2\|_2^2}\Big),$$

for some positive constant C_{01}.

To bound the supremum of $S_{\mathbf{v}}$, we define the metric $s(\mathbf{v}_1, \mathbf{v}_2) = \mu_0\sqrt{\lambda_{\max}}\|\mathbf{v}_1 - \mathbf{v}_2\|_2$. Then, by Lemma B in Chen and Banerjee (2016),

$$\begin{aligned}\mathbb{E}\Big[\sup_{\mathbf{v}\in\mathcal{T}} \mathbf{v}'\mathbf{\Gamma}^{1/2}\mathbf{g}\Big] &\le C_{02}\gamma_2(\mathcal{T}, s) \\ &= C_{02}\mu_0\sqrt{\lambda_{\max}}\gamma_2(\mathcal{T}, \|\cdot\|_2) \\ &\le C_3\mu_0\sqrt{\lambda_{\max}}w(\mathcal{T}),\end{aligned}$$

for positive constants C_{02}, C_3, where $\gamma_2(\mathcal{T}, s)$ is the γ_2-functional we referred to in the proof of Lemma 3.7. Since $\mathcal{T} = \mathrm{cone}(\mathcal{A}(\boldsymbol{\beta}^*)) \cap \mathbb{B}^m \subseteq \mathrm{conv}(\mathcal{A}(\boldsymbol{\beta}^*) \cup \{\mathbf{0}\})$, by Lemma 2 in Maurer et al. (2014),

$$\begin{aligned}w(\mathcal{T}) &\le w(\mathrm{conv}(\mathcal{A}(\boldsymbol{\beta}^*) \cup \{\mathbf{0}\})) \\ &= w(\mathcal{A}(\boldsymbol{\beta}^*) \cup \{\mathbf{0}\}) \\ &\le \max\{w(\mathcal{A}(\boldsymbol{\beta}^*)), w(\{\mathbf{0}\})\} + 2\sqrt{\ln 4} \\ &\le w(\mathcal{A}(\boldsymbol{\beta}^*)) + 3.\end{aligned}$$

Thus,

$$\begin{aligned}w(\mathcal{A}_{\mathbf{\Gamma}}) &= \mathbb{E}\Big[\sup_{\mathbf{w}\in\mathcal{A}_{\mathbf{\Gamma}}} \mathbf{w}'\mathbf{g}\Big] \\ &\le \sqrt{\frac{1}{\lambda_{\min}}}\mathbb{E}\Big[\sup_{\mathbf{v}\in\mathcal{T}} \mathbf{v}'\mathbf{\Gamma}^{1/2}\mathbf{g}\Big] \\ &\le C_3\sqrt{\frac{1}{\lambda_{\min}}}\mu_0\sqrt{\lambda_{\max}}w(\mathcal{T}) \\ &\le C_3\mu_0\sqrt{\frac{\lambda_{\max}}{\lambda_{\min}}}\Big(w(\mathcal{A}(\boldsymbol{\beta}^*)) + 3\Big).\end{aligned}$$

■

A.10 Proof of Corollary 3.9

Proof Combining Lemmas 3.7 and 3.8, and using the fact that for any $\mathbf{v} \in \mathcal{A}(\boldsymbol{\beta}^*)$,

$$\frac{N}{2}\mathbf{v}'\mathbf{\Gamma}\mathbf{v} \ge \frac{N\lambda_{\min}}{2},$$

we can derive the desired result. ■

A.11 Proof of Lemma 3.10

Proof By the definition of dual norm, we know that:

$$\|\tilde{\boldsymbol{\beta}}'\mathbf{Z}\|_1 = \sup_{\mathbf{v}\in\mathcal{B}_u} \tilde{\boldsymbol{\beta}}'\mathbf{Z}\mathbf{v} = \sup_{\mathbf{v}\in\mathcal{B}_u} \sum_{i=1}^N v_i\tilde{\boldsymbol{\beta}}'\mathbf{z}_i.$$

Since $v_i\tilde{\boldsymbol{\beta}}'\mathbf{z}_i,\ i = 1, \ldots, N$ are independent centered sub-Gaussian random variables, and

$$\Big|\!\Big|\!\Big|v_i\tilde{\boldsymbol{\beta}}'\mathbf{z}_i\Big|\!\Big|\!\Big|_{\psi_2} \le \mu\|v_i\tilde{\boldsymbol{\beta}}\|_2,$$

we have that $\sum_{i=1}^N v_i\tilde{\boldsymbol{\beta}}'\mathbf{z}_i$ is also a centered sub-Gaussian random variable with

$$\begin{aligned}\Big|\!\Big|\!\Big|\sum_{i=1}^N v_i\tilde{\boldsymbol{\beta}}'\mathbf{z}_i\Big|\!\Big|\!\Big|_{\psi_2}^2 &\le C_{03}^2\sum_{i=1}^N \mu^2\|v_i\tilde{\boldsymbol{\beta}}\|_2^2 \\ &= C_{03}^2\mu^2\|\tilde{\boldsymbol{\beta}}\|_2^2\|\mathbf{v}\|_2^2,\end{aligned}$$

for a positive constant C_{03}.

Consider the stochastic process $\{S_{\mathbf{v}} = \tilde{\boldsymbol{\beta}}'\mathbf{Z}\mathbf{v}\}_{\mathbf{v}\in\mathcal{B}_u}$. As in the proof of Lemma 3.8,

$$\|\!|S_{\mathbf{v}_1} - S_{\mathbf{v}_2}|\!\|_{\psi_2} \le C_{03}\mu\|\tilde{\boldsymbol{\beta}}\|_2\|\mathbf{v}_1 - \mathbf{v}_2\|_2.$$

By the tail behavior of sub-Gaussian random variables (Vershynin, 2017), we know:

$$\mathbb{P}(|S_{\mathbf{v}_1} - S_{\mathbf{v}_2}| \geq \delta) \leq 2\exp\Big(-\frac{C_{04}\delta^2}{\mu^2\|\tilde{\boldsymbol{\beta}}\|_2^2\|\mathbf{v}_1 - \mathbf{v}_2\|_2^2}\Big),$$

for a positive constant C_{04}.

Define the metric $s(\mathbf{v}_1, \mathbf{v}_2) = \mu\|\tilde{\boldsymbol{\beta}}\|_2\|\mathbf{v}_1 - \mathbf{v}_2\|_2$. Then, by Lemma B in Chen and Banerjee (2016),

$$\mathbb{P}\Big(\sup_{\mathbf{v}_1,\mathbf{v}_2\in\mathcal{B}_u} |S_{\mathbf{v}_1} - S_{\mathbf{v}_2}| \geq C_{05}\big(\gamma_2(\mathcal{B}_u, s) + \delta\cdot\mathrm{diam}(\mathcal{B}_u, s)\big)\Big) \leq C_4\exp(-\delta^2),$$

for positive constants C_{05}, C_4. Also,

$$\begin{aligned}
\gamma_2(\mathcal{B}_u, s) &= \mu\|\tilde{\boldsymbol{\beta}}\|_2\gamma_2(\mathcal{B}_u, \|\cdot\|_2) \leq C_5\mu\|\tilde{\boldsymbol{\beta}}\|_2 w(\mathcal{B}_u),\\
\mathrm{diam}(\mathcal{B}_u, s) &= \sup_{\mathbf{v}_1,\mathbf{v}_2\in\mathcal{B}_u} s(\mathbf{v}_1, \mathbf{v}_2)\\
&= \mu\|\tilde{\boldsymbol{\beta}}\|_2 \sup_{\mathbf{v}_1,\mathbf{v}_2\in\mathcal{B}_u} \|\mathbf{v}_1 - \mathbf{v}_2\|_2\\
&\leq 2\mu\|\tilde{\boldsymbol{\beta}}\|_2 \sup_{\mathbf{v}\in\mathcal{B}_u} \|\mathbf{v}\|_2\\
&= 2\mu\|\tilde{\boldsymbol{\beta}}\|_2\rho,
\end{aligned}$$

for positive constants C_5. Therefore, noting that $\sup_{\mathbf{v}_1,\mathbf{v}_2\in\mathcal{B}_u} |S_{\mathbf{v}_1} - S_{\mathbf{v}_2}| \geq 2\sup_{\mathbf{v}\in\mathcal{B}_u} S_{\mathbf{v}}$, we obtain

$$\begin{aligned}
&\mathbb{P}\Big(\sup_{\mathbf{v}\in\mathcal{B}_u} S_{\mathbf{v}} \geq C_{05}\Big(\frac{C_5}{2}\mu\|\tilde{\boldsymbol{\beta}}\|_2 w(\mathcal{B}_u) + \delta\mu\|\tilde{\boldsymbol{\beta}}\|_2\rho\Big)\Big)\\
\leq\ &\mathbb{P}\Big(\sup_{\mathbf{v}_1,\mathbf{v}_2\in\mathcal{B}_u} |S_{\mathbf{v}_1} - S_{\mathbf{v}_2}| \geq C_{05}\big(\gamma_2(\mathcal{B}_u, s) + \delta\mathrm{diam}(\mathcal{B}_u, s)\big)\Big)\\
\leq\ &C_4\exp(-\delta^2).
\end{aligned}$$

Set $\delta = \frac{C_5 w(\mathcal{B}_u)}{2\rho}$; then with probability at least $1 - C_4\exp(-\frac{C_5^2(w(\mathcal{B}_u))^2}{4\rho^2})$,

$$\sup_{\mathbf{v}\in\mathcal{B}_u} S_{\mathbf{v}} \leq C\mu\bar{B}_2 w(\mathcal{B}_u).$$

The result follows. ■

A.12 Proof of the Result in Section 4

We will show that if the Wasserstein metric is defined by the following metric s_c:

$$s_c(\mathbf{x}, y) = \|(\mathbf{x}, cy)\|_\infty,$$

then as $c \to \infty$, the corresponding Wasserstein DRO formulation becomes:

$$\inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N}\sum_{i=1}^N |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon\|\boldsymbol{\beta}\|_1,$$

which is the ℓ_1-regularized LAD.

Proof We first define a new notion of norm on $(\mathbf{x}, y)$ where $\mathbf{x} = (x_1, \ldots, x_{m-1})$:

$$\|(\mathbf{x}, y)\|_{\mathbf{w},p} \triangleq \|(x_1 w_1, \ldots, x_{m-1}w_{m-1}, yw_m)\|_p,$$

for some m-dimensional weighting vector $\mathbf{w} = (w_1, \ldots, w_m)$, and $p \geq 1$. Then, $s_c(\mathbf{x}, y) = \|(\mathbf{x}, y)\|_{\mathbf{w},\infty}$ with $\mathbf{w} = (1, \ldots, 1, c)$. To obtain the Wasserstein DRO formulation, the key is to derive the dual norm of $\|\cdot\|_{\mathbf{w},\infty}$. Hölder's inequality (Rogers, 1888) will be used for the derivation. We state it below for convenience.

Theorem 1 (Hölder's inequality) *Suppose we have two scalars $p, q > 1$ and $1/p + 1/q = 1$. For any two vectors $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$, the following holds.*

$$\sum_{i=1}^n |a_i b_i| \leq \|\mathbf{a}\|_p\|\mathbf{b}\|_q.$$

We will use the notation $\mathbf{z} \triangleq (\mathbf{x}, y)$. Based on the definition of dual norm, we are interested in solving the following optimization problem for $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^m$:

$$\begin{aligned}
\max_{\mathbf{z}}\quad & \mathbf{z}'\tilde{\boldsymbol{\beta}}\\
\text{s.t.}\quad & \|\mathbf{z}\|_{\mathbf{w},\infty} \leq 1.
\end{aligned} \tag{30}$$

The optimal value of problem (30), which is a function of $\tilde{\boldsymbol{\beta}}$, gives the dual norm evaluated at $\tilde{\boldsymbol{\beta}}$. Using Hölder's inequality, we can write

$$\mathbf{z}'\tilde{\boldsymbol{\beta}} = \sum_{i=1}^m (w_i z_i)\Big(\frac{1}{w_i}\tilde{\beta}_i\Big) \leq \|\mathbf{z}\|_{\mathbf{w},\infty}\|\tilde{\boldsymbol{\beta}}\|_{\mathbf{w}^{-1},1} \leq \|\tilde{\boldsymbol{\beta}}\|_{\mathbf{w}^{-1},1},$$

where $\mathbf{w}^{-1} \triangleq (\frac{1}{w_1}, \ldots, \frac{1}{w_m})$. The last inequality is due to the constraint $\|\mathbf{z}\|_{\mathbf{w},\infty} \leq 1$. It follows that the dual norm of $\|\cdot\|_{\mathbf{w},\infty}$ is just $\|\cdot\|_{\mathbf{w}^{-1},1}$. Back to our problem setting, using $\mathbf{w} = (1, \ldots, 1, c)$, and evaluating the dual norm at $(-\boldsymbol{\beta}, 1)$, we have the following Wasserstein DRO formulation as $c \to \infty$:

$$\lim_{c\to\infty}\inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N}\sum_{i=1}^N |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon\|(-\boldsymbol{\beta}, 1)\|_{\mathbf{w}^{-1},1} = \inf_{\boldsymbol{\beta}\in\mathcal{B}} \frac{1}{N}\sum_{i=1}^N |y_i - \mathbf{x}_i'\boldsymbol{\beta}| + \epsilon\|\boldsymbol{\beta}\|_1.$$

■

References

Peter L Bartlett and Shahar Mendelson. Rademacher and Gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

Güzin Bayraksan and David K Love. Data-driven stochastic programming using ϕ-divergences. *Tutorials in Operations Research*, pages 1–19, 2015.

Dimitri P Bertsekas. *Nonlinear programming*. Athena scientific Belmont, 1999.

Dimitris Bertsimas and Martin S Copenhaver. Characterization of the equivalence of robustification and regularization in linear and matrix regression. *European Journal of Operational Research*, 2017.

Dimitris Bertsimas, Vishal Gupta, and Ioannis Ch Paschalidis. Data-driven estimation in equilibrium using inverse optimization. *Mathematical Programming*, 153(2):595–633, 2015.

Jose Blanchet and Karthyek Murthy. Quantifying distributional model risk via optimal transport. Technical Report arXiv:1604.01446, 2016.

Ruidi Chen, Ioannis Ch Paschalidis, Hiroto Hatabu, Vladimir Valtchinov, and Jenifer Siegelman. Detection of unwarranted CT radiation exposure from patient and imaging protocol meta-data using regularized regression. *Working paper*, 2018.

Sheng Chen and Arindam Banerjee. Alternating estimation for structured high-dimensional multi-response models. *arXiv preprint arXiv:1606.08957*, 2016.

David Coleman, Paul Holland, Neil Kaden, Virginia Klema, and Stephen C Peters. A system of subroutines for iteratively reweighted least squares computations. *ACM Transactions on Mathematical Software (TOMS)*, 6(3):327–336, 1980.

Erick Delage and Yinyu Ye. Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612, 2010.

Laurent El Ghaoui and Hervé Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM Journal on Matrix Analysis and Applications*, 18(4):1035–1064, 1997.

E Erdoğan and Garud Iyengar. Ambiguous chance constrained problems and robust optimization. *Mathematical Programming*, 107(1-2):37–61, 2006.

Peyman Mohajerin Esfahani and Daniel Kuhn. Data-driven distributionally robust optimization using the Wasserstein metric: performance guarantees and tractable reformulations. *Available at Optimization Online*, 2015.

Ray C Fair. On the robust estimation of econometric models. In *Annals of Economic and Social Measurement, Volume 3, number 4*, pages 667–677. NBER, 1974.

Nicolas Fournier and Arnaud Guillin. On the rate of convergence in Wasserstein distance of the empirical measure. *Probability Theory and Related Fields*, 162(3-4):707–738, 2015.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Rui Gao and Anton J Kleywegt. Distributionally robust stochastic optimization with Wasserstein distance. *arXiv preprint arXiv:1604.02199*, 2016.

Rui Gao, Xi Chen, and Anton J Kleywegt. Wasserstein distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*, 2017.

Joel Goh and Melvyn Sim. Distributionally robust optimization and its tractable approximations. *Operations research*, 58(4-part-1):902–917, 2010.

Trevor Hastie, Robert Tibshirani, and Ryan J Tibshirani. Extended comparisons of best subset selection, forward stepwise selection, and the LASSO. *arXiv preprint arXiv:1707.08692*, 2017.

Melvin J Hinich and Prem P Talwar. A simple method for robust regression. *Journal of the American Statistical Association*, 70(349):113–119, 1975.

Arthur E Hoerl and Robert W Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

Zhaolin Hu and L Jeff Hong. Kullback-Leibler divergence constrained distributionally robust optimization. *Available at Optimization Online*, 2013.

Peter J Huber. Robust estimation of a location parameter. *The Annals of Mathematical Statistics*, 35(1):73–101, 1964.

Peter J Huber. Robust regression: asymptotics, conjectures and Monte Carlo. *The Annals of Statistics*, 1(5):799–821, 1973.

Ruiwei Jiang and Yongpei Guan. Data-driven chance constrained stochastic program. *Mathematical Programming*, pages 1–37, 2015.

Fengqiao Luo and Sanjay Mehrotra. Decomposition algorithm for distributionally robust optimization using Wasserstein metric. *arXiv preprint arXiv:1704.03920*, 2017.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. An inequality with applications to structured sparsity and multitask dictionary learning. In *COLT*, pages 440–460, 2014.

Sanjay Mehrotra and He Zhang. Models and algorithms for distributionally robust least squares problems. *Mathematical Programming*, 146(1-2):123–141, 2014.

Shahar Mendelson, Alain Pajor, and Nicole Tomczak-Jaegermann. Reconstruction and subGaussian operators in asymptotic geometric analysis. *Geometric and Functional Analysis*, 17(4):1248–1282, 2007.

David Pollard. Asymptotics for least absolute deviation regression estimators. *Econometric Theory*, 7(02):186–199, 1991.

Ioana Popescu. Robust mean-covariance solutions for stochastic optimization. *Operations Research*, 55(1):98–112, 2007.

Leonhard James Rogers. An extension of a certain theorem in inequalities. *Messenger of Math*, 17(2):145–150, 1888.

Peter Rousseeuw and Victor Yohai. Robust regression by means of S-estimators. In *Robust and nonlinear time series analysis*, pages 256–272. Springer, 1984.

Peter J Rousseeuw. Least median of squares regression. *Journal of the American statistical association*, 79(388):871–880, 1984.

Peter J Rousseeuw. Multivariate estimation with high breakdown point. *Mathematical statistics and applications*, 8:283–297, 1985.

Peter J Rousseeuw and Annick M Leroy. *Robust regression and outlier detection*. John Wiley & Sons, 2005.

Soroosh Shafieezadeh-Abadeh, Peyman Mohajerin Esfahani, and Daniel Kuhn. Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, pages 1576–1584, 2015.

Soroosh Shafieezadeh-Abadeh, Daniel Kuhn, and Peyman Mohajerin Esfahani. Regularization via mass transportation. *arXiv preprint arXiv:1710.10016*, 2017.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*. Cambridge University Press (to appear), 2017.

Li Wang, Michael D Gordon, and Ji Zhu. Regularized least absolute deviations regression and an efficient algorithm for parameter tuning. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 690–700. IEEE, 2006.

Zizhuo Wang, Peter W Glynn, and Yinyu Ye. Likelihood robust optimization for data-driven problems. *Computational Management Science*, 13(2):241–261, 2016.

Wolfram Wiesemann, Daniel Kuhn, and Melvyn Sim. Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376, 2014.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robustness and regularization of support vector machines. *Journal of Machine Learning Research*, 10(Jul):1485–1510, 2009.

Huan Xu, Constantine Caramanis, and Shie Mannor. Robust regression and LASSO. *IEEE Transactions on Information Theory*, 56(7):3561–3574, 2010.

Wenzhuo Yang and Huan Xu. A unified robust regression model for LASSO-like algorithms. In *International Conference on Machine Learning*, pages 585–593, 2013.

Victor J Yohai. High breakdown-point and high efficiency robust estimates for regression. *The Annals of Statistics*, pages 642–656, 1987.

C Zhao and Y Guan. Data-driven risk-averse stochastic optimization with Wasserstein metric. *Available on optimization online*, 2015.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

S. Zymler, D. Kuhn, and B. Rustem. Distributionally robust joint chance constraints with second-order moment information. *Mathematical Programming*, 137(1-2):167–198, 2013.

Journal of Machine Learning Research 19 (2018) 1-25 Submitted 6/17; Revised 4/18; Published 8/18

Model-Free Trajectory-based Policy Optimization with Monotonic Improvement

Riad Akrour[1] riad@robot-learning.de
Abbas Abdolmaleki[2] aabdolmaleki@google.com
Hany Abdulsamad[1] hany@robot-learning.de
Jan Peters[1,3] jan@robot-learning.de
Gerhard Neumann[1,4] geri@robot-learning.de
[1] *CLAS/IAS, Technische Universitäte Darmstadt, Hochschulstr. 10, D-64289 Darmstadt, Germany*
[2] *DeepMind, London N1C 4AG, UK*
[3] *Max Planck Institute for Intelligent Systems, Max-Planck-Ring 4, Tübingen, Germany*
[4] *L-CAS, University of Lincoln, Lincoln LN6 7TS, UK*

Editor: George Konidaris

Abstract

Many of the recent trajectory optimization algorithms alternate between linear approximation of the system dynamics around the mean trajectory and conservative policy update. One way of constraining the policy change is by bounding the Kullback-Leibler (KL) divergence between successive policies. These approaches already demonstrated great experimental success in challenging problems such as end-to-end control of physical systems. However, the linear approximation of the system dynamics can introduce a bias in the policy update and prevent convergence to the optimal policy. In this article, we propose a new model-free trajectory-based policy optimization algorithm with guaranteed monotonic improvement. The algorithm backpropagates a local, quadratic and time-dependent Q-Function learned from trajectory data instead of a model of the system dynamics. Our policy update ensures exact KL-constraint satisfaction without simplifying assumptions on the system dynamics. We experimentally demonstrate on highly non-linear control tasks the improvement in performance of our algorithm in comparison to approaches linearizing the system dynamics. In order to show the monotonic improvement of our algorithm, we additionally conduct a theoretical analysis of our policy update scheme to derive a lower bound of the change in policy return between successive iterations.

Keywords: Reinforcement Learning, Policy Optimization, Trajectory Optimization, Robotics

1. Introduction

Trajectory Optimization methods based on stochastic optimal control (Todorov, 2006; Theodorou et al., 2009; Todorov and Tassa, 2009) have been very successful in learning high dimensional controls in complex settings such as end-to-end control of physical systems (Levine and Abbeel, 2014). These methods are based on a time-dependent linearization of the dynamics model around the mean trajectory in order to obtain a closed form update of the policy as a Linear-Quadratic Regulator (LQR). This linearization is then repeated locally for the new policy at every iteration. However, this iterative process does not offer convergence guarantees as the linearization of the dynamics might introduce a bias and impede the algorithm from converging to the optimal policy. To circumvent this limitation, we propose in this paper a novel model-free trajectory-based policy optimization algorithm (MOTO) couched in the approximate policy iteration framework. At each iteration, a Q-Function is estimated locally around the current trajectory distribution using a time-dependent quadratic function. Afterwards, the policy is updated according to a new information-theoretic trust region that bounds the KL-divergence between successive policies in closed form.

MOTO is well suited for high dimensional continuous state and action spaces control problems. The policy is represented by a time-dependent stochastic linear-feedback controller which is updated by a Q-Function propagated backward in time. We extend the work of (Abdolmaleki et al., 2015), which was proposed in the domain of stochastic search (having no notion of state space nor that of sequential decisions), to that of sequential decision making and show that our policy class can be updated under a KL-constraint in closed form, when the learned Q-Function is a quadratic function of the state and action space. In order to maximize sample efficiency, we rely on importance sampling to reuse transition samples from policies of all time-steps and all previous iterations in a principled way. MOTO is able to solve complex control problems despite the simplicity of the Q-Function thanks to two key properties: i) the learned Q-Function is fitted to samples of the current policy, which ensures that the function is *valid locally* and ii) the closed form update of the policy ensures that the KL-constraint is satisfied exactly irrespective of the number of samples or the non-linearity of the dynamics, which ensures that the Q-Function is *used locally*.

The experimental section demonstrates that on tasks with highly non-linear dynamics MOTO outperforms similar methods that rely on a linearization of these dynamics. Additionally, it is shown on a simulated Robot Table Tennis Task that MOTO is able to scale to high dimensional tasks while keeping the sample complexity relatively low; amenable to a direct application to a physical system.

Compared to Akrour et al. (2016), we report new experimental results comparing MOTO to TRPO (Schulman et al., 2015), a state-of-the-art reinforcement learning algorithm. These results showcase settings in which the time-dependent linear-Gaussian policies used by MOTO are a suitable alternative to neural networks. We also conduct a theoretical analysis of the policy update (Sec. 5) and lower bound the increase in policy return between successive iterations of the algorithm. The resulting lower bound validates the use of an expected KL-constraint (Sec. 3.1) in a trajectory-based policy optimization setting for ensuring a monotonic improvement of the policy return. Prior theoretical studies reported similar results when the *maximum* (over the state space) KL is upper bounded which is hard to enforce in practice (Schulman et al., 2015). Leveraging standard trajectory optimization assumptions, we extend prior analysis of the policy update to the specific setting of MOTO when only the *expected* policy KL is bounded.

2. Notation

Consider an undiscounted finite-horizon Markov Decision Process (MDP) of horizon T with state space $\mathcal{S} = \mathbb{R}^{\mathrm{d}_s}$ and action space $\mathcal{A} = \mathbb{R}^{\mathrm{d}_a}$. The transition function $p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$,

which gives the probability (density) of transitioning to state $\boldsymbol{s}_{t+1}$ upon the execution of action $\boldsymbol{a}_t$ in $\boldsymbol{s}_t$, is assumed to be time-independent; while there are T time-dependent reward functions $r_t : \mathcal{S} \times \mathcal{A} \mapsto \mathbb{R}$. A policy π is defined by a set of time-dependent density functions π_t, where $\pi_t(\boldsymbol{a}|\boldsymbol{s})$ is the probability of executing action $\boldsymbol{a}$ in state $\boldsymbol{s}$ at time-step t. The goal is to find the optimal policy $\pi^* = \{\pi_1^*, \dots, \pi_T^*\}$ maximizing the policy return $J(\pi) = \mathbb{E}_{\boldsymbol{s}_1, \boldsymbol{a}_1, \dots}\left[\sum_{t=1}^{T} r_t(\boldsymbol{s}_t, \boldsymbol{a}_t)\right]$, where the expectation is taken w.r.t. all the random variables $\boldsymbol{s}_t$ and $\boldsymbol{a}_t$ such that $\boldsymbol{s}_1 \sim \rho_1$ follows the distribution of the initial state, $\boldsymbol{a}_t \sim \pi_t(.|\boldsymbol{s}_t)$ and $\boldsymbol{s}_{t+1} \sim p(\boldsymbol{s}_{t+1}|\boldsymbol{s}_t, \boldsymbol{a}_t)$.

As is common in Policy Search (Deisenroth et al., 2013), our algorithm operates on a restricted class of parameterized policies $\pi_{\boldsymbol{\theta}}, \boldsymbol{\theta} \in \mathbb{R}^{\mathrm{d}_\theta}$ and is an iterative algorithm comprising two main steps, *policy evaluation* and *policy update*. Throughout this article, we will assume that each time-dependent policy is parameterized by $\boldsymbol{\theta}_t = \{K_t, \boldsymbol{k}_t, \Sigma_t\}$ such that $\pi_{\boldsymbol{\theta}_t}$ is of linear-Gaussian form $\pi_{\boldsymbol{\theta}_t}(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(K_t s + \boldsymbol{k}_t, \Sigma_t)$, where the gain matrix K_t is a $d_a \times d_s$ matrix, the bias term $\boldsymbol{k}_t$ is a d_a dimensional column vector and the covariance matrix Σ_t, which controls the exploration of the policy, is of dimension $d_a \times d_a$; yielding a total number of parameters across all time-steps of $d_\theta = T(d_a d_s + \frac{1}{2} d_a(d_a + 3))$.

The policy at iteration i of the algorithm is denoted by π^i and following standard definitions, the Q-Function of π^i at time-step t is given by $Q_t^i(\boldsymbol{s}, \boldsymbol{a}) = \mathbb{E}_{\boldsymbol{s}_t, \boldsymbol{a}_t, \dots}\left[\sum_{t'=t}^{T} r_{t'}(\boldsymbol{s}_{t'}, \boldsymbol{a}_{t'})\right]$ with $(\boldsymbol{s}_t, \boldsymbol{a}_t) = (\boldsymbol{s}, \boldsymbol{a})$ and $\boldsymbol{a}_{t'} \sim \pi_{t'}^i(.|\boldsymbol{s}_{t'}), \forall t' > t$. While the V-Function is given by $V_t^i(\boldsymbol{s}) = \mathbb{E}_{\boldsymbol{a} \sim \pi_t(.|\boldsymbol{s})}\left[Q_t^\pi(\boldsymbol{s}, \boldsymbol{a})\right]$ and the Advantage Function by $A_t^i(\boldsymbol{s}, \boldsymbol{a}) = Q_t^i(\boldsymbol{s}, \boldsymbol{a}) - V_t^i(\boldsymbol{s})$. Furthermore the state distribution at time-step t, related to policy π^i, is denoted by $\rho_t^i(\boldsymbol{s})$. In order to keep the notations uncluttered, the time-step or the iteration number is occasionally dropped when a definition applies similarly for all time-steps or iteration number.

3. Model-free Policy Update for Trajectory-based Policy Optimization

MOTO alternates between policy evaluation and policy update. At each iteration i, the policy evaluation step generates a set of M rollouts[1] from the policy π^i in order to estimate a (quadratic) Q-Function $\tilde{Q}^i$ (Sec. 4.1) and a (Gaussian) state distribution $\tilde{\rho}^i$ (Sec. 4.3). Using these quantities, an information-theoretic policy update is derived at each time-step that uses a KL-bound as a trust region to obtain the policy π^{i+1} of the next iteration.

3.1 Optimization Problem

The goal of the policy update is to return a new policy π^{i+1} that maximizes the Q-Function $\tilde{Q}^i$ in expectation under the state distribution $\tilde{p}_i$ of the previous policy π^i. In order to limit policy oscillation between iterations (Wagner, 2011), the KL w.r.t. π^i is upper bounded. The use of the KL divergence to define the step-size of the policy update has already been successfully applied in prior work (Peters et al., 2010; Levine and Abbeel, 2014; Schulman et al., 2015). Additionally, we lower bound the entropy of π^{i+1} in order to better control

1. A rollout is a Monte Carlo simulation of a trajectory according to ρ_1, π and p or the execution of π on a physical system.

the reduction of exploration yielding the following non-linear program:

$$\underset{\pi}{\text{maximize}} \quad \int\int \tilde{\rho}_t^i(\boldsymbol{s})\pi(\boldsymbol{a}|\boldsymbol{s})\tilde{Q}_t^i(\boldsymbol{s}, \boldsymbol{a})\mathrm{d}a\mathrm{d}s, \tag{1}$$

$$\text{subject to} \quad \mathbb{E}_{s \sim \tilde{\rho}_t^i(\boldsymbol{s})}\left[\mathrm{KL}(\pi(.|\boldsymbol{s}) \parallel \pi_t^i(.|\boldsymbol{s}))\right] \leq \epsilon, \tag{2}$$

$$\mathbb{E}_{s \sim \tilde{\rho}_t^i(\boldsymbol{s})}\left[\mathcal{H}\left(\pi(.|\boldsymbol{s})\right)\right] \geq \beta. \tag{3}$$

The KL between two distributions p and q is given by $\mathrm{KL}(p \parallel q) = \int p(x) \log \frac{p(x)}{q(x)} \mathrm{d}x$ while the entropy $\mathcal{H}$ is given by $\mathcal{H} = -\int p(x) \log p(x) \mathrm{d}x$. The step-size ϵ is a hyper-parameter of the algorithm kept constant throughout the iterations while β is set according to the entropy of the current policy π_t^i, $\beta = \mathbb{E}_{s \sim \tilde{\rho}_t^i(\boldsymbol{s})}\left[\mathcal{H}\left(\pi_t^i(.|\boldsymbol{s})\right)\right] - \beta_0$ and β_0 is the entropy reduction hyper-parameter kept constant throughout the iterations.

Eq. (1) indicates that π_t^{i+1} maximizes $\tilde{Q}_t^i$ in expectation under its own action distribution and the state distribution of π_t^i. Eq. (2) bounds the average change in the policy to the step-size ϵ while Eq. (3) controls the exploration-exploitation trade-off and ensures that the exploration in the action space (which is directly linked to the entropy of the policy) is not reduced too quickly. A similar constraint was introduced in the stochastic search domain by (Abdolmaleki et al., 2015), and was shown to avoid premature convergence. This constraint is even more crucial in our setting because of the inherent *non-stationarity* of the objective function being optimized at each iteration. The cause for the non-stationarity of the objective optimized at time-step t in the policy update is twofold: i) updates of policies $\pi_{t'}$ with time-step $t' > t$ will modify in the next iteration of the algorithm $\tilde{Q}_t$ as a function of s and a and hence the optimization landscape as a function of the policy parameters, ii) updates of policies with time-step $t' < t$ will induce a change in the state distribution ρ_t. If the policy had unlimited expressiveness, the optimal solution of Eq. (1) would be to choose $\arg\max_a \tilde{Q}_t$ irrespective of ρ_t. However, due to the restricted class of the policy, any change in ρ_t will likely change the optimization landscape including the position of the optimal policy parameter. Hence, Eq. (3) ensures that exploration in action space is maintained as the optimization landscape evolves and avoids premature convergence.

3.2 Closed Form Update

Using the method of Lagrange multipliers, the solution of the optimization problem in section 3.1 is given by

$$\pi_t'(\boldsymbol{a}|\boldsymbol{s}) \propto \pi_t(\boldsymbol{a}|\boldsymbol{s})^{\eta^*/(\eta^* + \omega^*)} \exp\left(\frac{\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})}{\eta^* + \omega^*}\right), \tag{4}$$

with η^* and ω^* being the optimal Lagrange multipliers related to the KL and entropy constraints respectively. Assuming that $\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})$ is of quadratic form in a and s

$$\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T Q_{aa}\boldsymbol{a} + \boldsymbol{a}^T Q_{as}\boldsymbol{s} + \boldsymbol{a}^T \boldsymbol{q}_a + q(\boldsymbol{s}), \tag{5}$$

with $q(\boldsymbol{s})$ grouping all terms of $\tilde{Q}_t(\boldsymbol{s},\boldsymbol{a})$ that do not depend[2] on a, then $\pi'_t(\boldsymbol{a}|\boldsymbol{s})$ is again of linear-Gaussian form

$$\pi'_t(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}|FL\boldsymbol{s} + F\boldsymbol{f}, F(\eta^* + \omega^*)),$$

such that the gain matrix, bias and covariance matrix of π'_t are function of matrices F and L and vector $\boldsymbol{f}$ where

$$F = (\eta^*\Sigma_t^{-1} - Q_{aa})^{-1}, \qquad L = \eta^*\Sigma_t^{-1}K_t + Q_{as}, \qquad \boldsymbol{f} = \eta^*\Sigma_t^{-1}\boldsymbol{k}_t + \boldsymbol{q}_a.$$

Note that $\eta\Sigma_t^{-1} - Q_{aa}$ needs to be invertible and positive semi-definite as it defines the new covariance matrix of the linear-Gaussian policy. For this to hold, either $Q_t(\boldsymbol{s},.)$ needs to be concave in a (i.e. Q_{aa} is negative semi-definite), or η needs to be large enough (and for any Q_{aa} such η always exists). A too large η is not desirable as it would barely yield a change to the current policy (too small KL divergence) and could negatively impact the convergence speed. Gradient based algorithms for learning model parameters with a specific semi-definite shape are available (Bhojanapalli et al., 2015) and could be used for learning a concave Q_t. However, we found in practice that the resulting η was always small enough (resulting in a maximally tolerated KL divergence of ϵ between successive policies) while F remains well defined, without requiring additional constraints on the nature of Q_{aa}.

3.3 Dual Minimization

The Lagrangian multipliers η and ω are obtained by minimizing the convex dual function

$$g_t(\eta,\omega) = \eta\epsilon - \omega\beta + (\eta+\omega)\int \tilde{\rho}_t(\boldsymbol{s}) \log\left(\int \pi_t(\boldsymbol{a}|\boldsymbol{s})^{\eta/(\eta+\omega)} \exp\left(\tilde{Q}_t(\boldsymbol{s},\boldsymbol{a})/(\eta+\omega)\right) \mathrm{d}\boldsymbol{a}\right) \mathrm{d}\boldsymbol{s}.$$

Exploiting the structure of the quadratic Q-Function $\tilde{Q}_t$ and the linear-Gaussian policy $\pi_t(\boldsymbol{a}|\boldsymbol{s})$, the inner integral over the action space can be evaluated in closed form and the dual simplifies to

$$g(\eta,\omega) = \eta\epsilon_t - \omega\beta_t + \int \rho_t(\boldsymbol{s})(\boldsymbol{s}^T M\boldsymbol{s} + \boldsymbol{s}^T\boldsymbol{m} + m_0)\mathrm{d}\boldsymbol{s}.$$

The dual function further simplifies, by additionally assuming normality of the state distribution $\tilde{\rho}_t(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_s, \Sigma_s)$, to the function

$$g_t(\eta,\omega) = \eta\epsilon - \omega\beta + \boldsymbol{\mu}_s^T M\boldsymbol{\mu}_s + \mathrm{tr}(\Sigma_s M) + \boldsymbol{\mu}_s^T\boldsymbol{m} + m_0,$$

which can be efficiently optimized by gradient descent to obtain η^* and ω^*. The full expression of the dual function, including the definition of M, $\boldsymbol{m}$ and m_0 in addition to the partial derivatives $\frac{\partial g_t(\eta,\omega)}{\partial\eta}$ and $\frac{\partial g_t(\eta,\omega)}{\partial\omega}$ are given in Appendix A.

2. Constant terms and terms depending on s but not a won't appear in the policy update. As such, and albeit we only refer in this article to $Q_t(\boldsymbol{s},\boldsymbol{a})$, the Advantage Function $A_t(\boldsymbol{s},\boldsymbol{a})$ can be used interchangeably in lieu of $Q_t(\boldsymbol{s},\boldsymbol{a})$ for updating the policy.

4. Sample Efficient Policy Evaluation

The KL constraint introduced in the policy update gives rise to a non-linear optimization problem. This problem can still be solved in closed form for the class of linear-Gaussian policies, if the learned function $\tilde{Q}_t^i$ is quadratic in $\boldsymbol{s}$ and $\boldsymbol{a}$. The first subsection introduces the main supervised learning problem solved during the policy evaluation for learning $\tilde{Q}_t^i$ while the remaining subsections discuss how to improve its sample efficiency.

4.1 The Q-Function Supervised Learning Problem

In the remainder of the section, we will be interested in finding the parameter $\boldsymbol{w}$ of a linear model $\tilde{Q}_t^i = \langle \boldsymbol{w}, \phi(\boldsymbol{s},\boldsymbol{a})\rangle$, where the feature function ϕ contains a bias and all the linear and quadratic terms of s and a, yielding $1 + (d_a + d_s)(d_a + d_s + 3)/2$ parameters. $\tilde{Q}_t^i$ can subsequently be written as in Eq. (5) by extracting Q_{aa}, Q_{as} and $\boldsymbol{q}_a$ from $\boldsymbol{w}$.

At each iteration i, M rollouts are performed following π^i. Let us initially assume that $\tilde{Q}_t^i$ is learned only from samples $\mathcal{D}_t^i = \{\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]}, \boldsymbol{s}_{t+1}^{[k]}; k = 1..M\}$ gathered from the execution of the M rollouts. The parameter $\boldsymbol{w}$ of $\tilde{Q}_t^i$ is learned by regularized linear least square regression

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \frac{1}{M}\sum_{k=1}^{M}\left(\langle \boldsymbol{w}, \phi(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]})\rangle - \hat{Q}_t^i(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]})\right)^2 + \lambda\boldsymbol{w}^T\boldsymbol{w}, \tag{6}$$

where the target value $\hat{Q}_t^i(\boldsymbol{s}^{[k]}, \boldsymbol{a}^{[k]})$ is a noisy estimate of the true value $Q_t^i(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]})$. We will distinguish two cases for obtaining the estimate $\hat{Q}_t^i(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]})$.

4.1.1 Monte-Carlo Estimate

This estimate is obtained by summing the future rewards for each trajectory k, yielding $\hat{Q}_t^i(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]}) = \sum_{t'=t}^{T} r_{t'}(\boldsymbol{s}_{t'}^{[k]}, \boldsymbol{a}_{t'}^{[k]})$. This estimator is known to have no bias but high variance. The variance can be reduced by averaging over multiple rollouts, assuming we can reset to states $\boldsymbol{s}_t^{[k]}$. However, such an assumption would severely limit the applicability of the algorithm on physical systems.

4.1.2 Dynamic Programming

In order to reduce the variance, this estimate exploits the V-Function to reduce the noise of the expected rewards of time-steps $t' > t$ through the following identity

$$\hat{Q}_t^i(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]}) = r_t(\boldsymbol{s}_t^{[k]}, \boldsymbol{a}_t^{[k]}) + \hat{V}_{t+1}^i(\boldsymbol{s}_{t+1}^{[k]}), \tag{7}$$

which is unbiased if $\hat{V}_{t+1}^i$ is. However, we will use for $\hat{V}_{t+1}^i$ an approximate V-Function $\tilde{V}_{t+1}^i$ learned recursively in time. This approximation might introduce a bias which will accumulate as t goes to 1. Fortunately, $\tilde{V}$ is not restricted by our algorithm to be of a particular class as it does not appear in the policy update. Hence, the bias can be reduced by increasing the complexity of the function approximator class. Nonetheless, in this article, a quadratic function will also be used for the V-Function which worked well in our experiments.

The V-Function is learned by first assuming that $\tilde{V}^i_{T+1}$ is the zero function.[3] Subsequently and recursively in time, the function $\tilde{V}^i_{t+1}$ and the transition samples in $\mathcal{D}^i_t$ are used to fit the parametric function $\tilde{V}^i_t$ by minimizing the loss $\sum_{k=1}^{M} \left(\hat{Q}^i_t(\boldsymbol{s}^{[k]}_t, \boldsymbol{a}^{[k]}_t) - \tilde{V}^i_t(\boldsymbol{s}^{[k]}_t)\right)^2$.

In addition to reducing the variance of the estimate $\hat{Q}^i_t(\boldsymbol{s}^{[k]}_t, \boldsymbol{a}^{[k]}_t)$, the choice of learning a V-Function is further justified by the increased possibility of reusing sample transitions from all time-steps and previous iterations.

4.2 Sample Reuse

In order to improve the sample efficiency of our approach, we will reuse samples from different time-steps and iterations using importance sampling. Let the expected loss which $\tilde{Q}^i_t$ minimizes under the assumption of an infinite number of samples be

$$\boldsymbol{w} = \arg\min_{\boldsymbol{w}} \mathbb{E}[\ell^i_t(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}'; \boldsymbol{w})],$$

where the loss ℓ^i_t is the inner term within the sum in Eq. (6); the estimate $\hat{Q}^i_t(\boldsymbol{s}^{[k]}_t, \boldsymbol{a}^{[k]}_t)$ is taken as in Eq. (7) and the expectation is with respect to the current state $\boldsymbol{s} \sim \rho^i_t$, the action $\boldsymbol{a} \sim \pi^i_t(.|\boldsymbol{s})$ and the next state $\boldsymbol{s}' \sim p(.|\boldsymbol{s}, \boldsymbol{a})$.

4.2.1 Reusing samples from different time-steps

To use transition samples from all time-steps when learning $\tilde{Q}^i_t$, we rely on importance sampling, where the importance weight (IW) is given by the ratio between the state-action probability of the current time-step $z^i_t(\boldsymbol{s}, \boldsymbol{a}) = \rho^i_t(\boldsymbol{s})\pi^i_t(\boldsymbol{a}|\boldsymbol{s})$ divided by the time-independent state-action probability of π^i given by $z^i(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{T}\sum_{t=1}^{T} z^i_t(\boldsymbol{s}, \boldsymbol{a})$. The expected loss minimized by $\tilde{Q}^i_t$ becomes

$$\min_{\boldsymbol{w}} \mathbb{E}\left[\frac{z^i_t(\boldsymbol{s}, \boldsymbol{a})}{z^i(\boldsymbol{s}, \boldsymbol{a})} \ell^i_t(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{s}'; \boldsymbol{w}) \mid (\boldsymbol{s}, \boldsymbol{a}) \sim z^i(\boldsymbol{s}, \boldsymbol{a})\right]. \tag{8}$$

Since the transition probabilities are not time-dependent they cancel out from the IW. Upon the computation of the IW, weighted least square regression is used to minimize an empirical estimate of (8) for the data set $\mathcal{D}^i = \cup_{t=1}^{T} \mathcal{D}^i_t$. Note that the (numerator of the) IW needs to be recomputed at every time-step for all samples $(\boldsymbol{s}, \boldsymbol{a}) \in \mathcal{D}^i$. Additionally, if the rewards are time-dependent, the estimate $\hat{Q}^i_t(\boldsymbol{s}^{[k]}_t, \boldsymbol{a}^{[k]}_t)$ in Eq. (7) needs to be recomputed with the current time-dependent reward, assuming the reward function is known.

4.2.2 Reusing samples from previous iterations

Following a similar reasoning, at a given time-step t, samples from previous iterations can be reused for learning $\tilde{Q}^i_t$. In this case, we have access to the samples of the state-action distribution $z^{1:i}_t(\boldsymbol{s}, \boldsymbol{a}) \propto \sum_{j=1}^{i} z^j_t(\boldsymbol{s}, \boldsymbol{a})$. The computation of $z^{1:i}_t$ requires the storage of all previous policies and state distributions. Thus, we will in practice limit ourselves to the K last iterations.

3. Alternatively one could assume the presence of a final reward $R_{T+1}(\boldsymbol{s}_{t+1})$, as is usually formulated in control tasks (Bertsekas, 1995), to which V^i_{T+1} could be initialized to.

Finally, both forms of sample reuse will be combined for learning $\tilde{Q}^i_t$ under the complete data set up to iteration i, $\mathcal{D}^{1:i} = \cup_{j=1}^{i} \mathcal{D}^j$ using weighted least square regression where the IW are given by $z^i_t(\boldsymbol{s}, \boldsymbol{a})/z^{1:i}(\boldsymbol{s}, \boldsymbol{a})$ with $z^{1:i}(\boldsymbol{s}, \boldsymbol{a}) \propto \sum_{t=1}^{T} z^{1:i}_t(\boldsymbol{s}, \boldsymbol{a})$.

4.3 Estimating the State Distribution

To compute the IW, the state distribution at every time-step ρ^i_t needs to be estimated. Since M rollouts are sampled for every policy π^i only M state samples are available for the estimation of ρ^i_t, necessitating again the reuse of previous samples to cope with higher dimensional control tasks.

4.3.1 Forward propagation of the state distribution

The first investigated solution for the estimation of the state distribution is the propagation of the estimate $\tilde{\rho}^i_t$ forward in time. Starting from $\tilde{\rho}^i_1$ which is identical for all iterations, importance sampling is used to learn $\tilde{\rho}^i_{t+1}$ with $t > 1$ from samples $(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1}) \in \mathcal{D}^{1:i}_t$ by weighted maximum-likelihood; where each sample $\boldsymbol{s}_{t+1}$ is weighted by $z^i_t(\boldsymbol{s}_t, \boldsymbol{a}_t)/z^{1:i}_t(\boldsymbol{s}_t, \boldsymbol{a}_t)$. And the computation of this IW only depends on the previously estimated state distribution $\tilde{\rho}^i_t$. In practice however, the estimate $\tilde{\rho}^i_t$ might entail errors despite the use of all samples from past iterations, which are propagated forward leading to a degeneracy of the number of effective samples in latter time-steps.

4.3.2 State distribution of a mixture policy

The second considered solution for the estimation of $\tilde{\rho}^i_t$ is heuristic but behaved better in practice. It is based on the intuition that the KL constraint of the policy update will yield state distributions that are close to each other (see Sec. 5 for a theoretical justification of the closeness in state distributions) and state samples from previous iterations can be reused in a simpler manner. Specifically, $\tilde{\rho}^i_t$ will be learned from samples of the mixture policy $\pi^{1:i} \propto \sum_{j=1}^{i} \gamma^{i-j}\pi^j$ which selects a policy from previous iterations with an exponentially decaying (w.r.t. the iteration number) probability and executes it for a whole rollout. In practice, the decay factor γ is selected according to the dimensionality of the problem, the number of samples per iterations M and the KL upper bound ϵ (intuitively, a smaller ϵ yields closer policies and henceforth more reusable samples). The estimated state distribution $\tilde{\rho}^i_t$ is learned as a Gaussian distribution by weighted maximum likelihood from samples of $\mathcal{D}^{1:i}_t$ where a sample of iteration j is weighted by γ^{i-j}.

4.4 The MOTO Algorithm

MOTO is summarized in Alg. 1. The innermost loop is split between policy evaluation (Sec. 4) and policy update (Sec. 3). For every time-step t, once the state distribution $\tilde{\rho}^i_t$ is estimated, the IWs of all the transition samples are computed and used to learn the Q-Function (and the V-Function using the same IWs, if dynamic programming is used when estimating the Q-Function), concluding the policy evaluation part. Subsequently, the components of the quadratic model $\tilde{Q}^i_t$ that depend on the action are extracted and used to find the optimal dual parameters η^* and ω^* that are respectively related to the KL and the entropy constraint, by minimizing the convex dual function g^i_t using gradient descent.

Algorithm 1 Model-Free Trajectory-based Policy Optimization (MOTO)

Input: Initial policy π^0, number of trajectories per iteration M, step-size ϵ and entropy reduction rate β_0
Output: Policy π^N
for $i = 0$ **to** $N-1$ **do**
  Sample M trajectories from π^i
  for $t = T$ **to** 1 **do**
    Estimate state distribution $\tilde{\rho}_t^i$ (Sec. 4.3)
    Compute IW for all $(\boldsymbol{s}, a, s') \in \mathcal{D}^{1:i}$ (Sec. 4.2)
    Estimate the Q-Function $\tilde{Q}_t^i$ (Sec. 4.1)
    Optimize: $(\eta^*, \omega^*) = \arg\min g_t^i(\eta, \omega)$ (Sec. 3.3)
    Update π_t^{i+1} using η^*, ω^*,$\tilde{\rho}_t^i$ and $\tilde{Q}_t^i$ (Sec. 3.2)
  end for
end for

The policy update then uses η^* and ω^* to return the new policy π_{t+1} and the process is iterated.

In addition to the simplification of the policy update, the rationale behind the use of a local quadratic approximation for Q_t^i is twofold: i) since Q_t^i is only optimized locally (because of the KL constraint), a quadratic model would potentially contain as much information as a Hessian matrix in a second order gradient descent setting ii) If $\tilde{Q}_t$ in Eq. (4) is an arbitrarily complex model then it is common that π_t', of linear-Gaussian form, is fit by weighted maximum-likelihood (Deisenroth et al., 2013); it is clear though from Eq. (4) that however complex $\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})$ is, if both π_t and π_t' are of linear-Gaussian form then there exist a quadratic model that would result in the same policy update. Additionally, note that $\tilde{Q}_t$ is not used when learning $\tilde{Q}_{t-1}$ (sec. 4.1) and hence the bias introduced by $\tilde{Q}_t$ will not propagate back. For these reasons, we think that choosing a more complex class for $\tilde{Q}_t$ than that of quadratic functions might not necessarily lead to an improvement of the resulting policy, for the class of linear-Gaussian policies.

5. Monotonic Improvement of the Policy Update

We analyze in this section the properties of the constrained optimization problem solved during our policy update. Kakade and Langford (2002) showed that in the approximate policy iteration setting, a monotonic improvement of the policy return can be obtained if the successive policies are close enough. While in our algorithm the optimization problem defined in Sec. 3.1 bounds the expected policy KL under the state distribution of the current iteration i, it does not tell us how similar the policies are under the new state distribution and a more careful analysis needs to be conducted.

The analysis we present builds on the results of Kakade and Langford (2002) to lower-bound the change in policy return $J(\pi^{i+1}) - J(\pi^i)$ between the new policy π^{i+1} (solution of the optimization problem defined in Sec. 3.1) and the current policy π^i. Unlike Kakade and Langford (2002), we enforce closeness between successive policies with a KL constraint instead of by mixing π^{i+1} and π^i. Related results were obtained when a KL constraint is

used in Schulman et al. (2015). Our main contribution is to extend these results to the trajectory optimization setting with continuous states and actions and where the expected KL between the policies is bounded instead of the maximum KL over the state space (which is hard to achieve in practice).

In what follows, p and q denote the next policy π^{i+1} and the current policy π^i respectively. We will denote the state distribution and policy at time-step t by p_t and $p_t(.|\boldsymbol{s})$ respectively (and similarly for q). First, we start by writing the difference between policy returns in term of advantage functions.

Lemma 1 *For any two policies p and q, and where A_t^q denotes the advantage function at time-step t of policy q, the difference in policy return is given by*

$$J(p) - J(q) = \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s}\sim p_t, \boldsymbol{a}\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}, \boldsymbol{a})\right].$$

The proof of Lemma 1 is given by the proof of Lemma 5.2.1 in (Kakade, 2003). Note that Lemma 1 expresses the change in policy return in term of expected advantage under the current state distribution while we optimize the advantage function under the state distribution of policy q, which is made apparent in Lemma 2.

Lemma 2 *Let $\epsilon_t = \mathrm{KL}(p_t \parallel q_t)$ be the KL divergence between state distributions $p_t(.)$ and $q_t(.)$ and let $\delta_t = \max_s |\mathbb{E}_{\boldsymbol{a}\sim p_t(.|\boldsymbol{s})}[A_t^q(\boldsymbol{s}, \boldsymbol{a})]|$, then for any two policies p and q we have*

$$J(p) - J(q) \geq \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{s}\sim q_t, \boldsymbol{a}\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}, \boldsymbol{a})\right] - 2\sum_{t=1}^{T} \delta_t \sqrt{\frac{\epsilon_t}{2}}.$$

Proof

$$\begin{aligned}
\mathbb{E}_{s\sim p_t, a\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}_t, \boldsymbol{a}_t)\right] &= \int p_t(\boldsymbol{s}) \int p_t(\boldsymbol{a}_t|\boldsymbol{s}_t) A_t^q(\boldsymbol{s}_t, \boldsymbol{a}_t), \\
&= \int q_t(\boldsymbol{s}) \int p_t(\boldsymbol{a}_t|\boldsymbol{s}_t) A_t^q(\boldsymbol{s}_t, \boldsymbol{a}_t) \\
&\quad + \int (p_t(\boldsymbol{s}) - q_t(\boldsymbol{s})) \int p_t(\boldsymbol{a}_t|\boldsymbol{s}_t) A_t^q(\boldsymbol{s}_t, \boldsymbol{a}_t), \\
&\geq \mathbb{E}_{s\sim q_t, a\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}, \boldsymbol{a})\right] - \delta_t \int (p_t(\boldsymbol{s}) - q_t(\boldsymbol{s})), \\
&\geq \mathbb{E}_{s\sim q_t, a\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}, \boldsymbol{a})\right] - 2\delta_t \frac{1}{2} \int |p_t(\boldsymbol{s}) - q_t(\boldsymbol{s})|, \\
&\geq \mathbb{E}_{s\sim q_t, a\sim p_t(.|\boldsymbol{s})} \left[A_t^q(\boldsymbol{s}, \boldsymbol{a})\right] - 2\delta_t \sqrt{\frac{1}{2}\mathrm{KL}(p_t \parallel q_t)}. \\
&\qquad \text{(Pinsker's inequality)}
\end{aligned}$$

Summing over the time-steps and using Lemma 1 completes the proof. ■

Lemma 2 lower-bounds the change in policy return by the advantage term optimized during the policy update and a negative change that quantifies the change in state distributions between successive policies. The core of our contribution is given by Lemma 3 which

relates the change in state distribution to the expected KL constraint between policies of our policy update.

Lemma 3 *If for every time-step, the state distributions p_t and q_t are Gaussian and the policies $p_t(.|\boldsymbol{s}_t)$ and $q_t(.|\boldsymbol{s}_t)$ are linear-Gaussian and if $\mathbb{E}_{s\sim q_t}\left[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))\right] \leq \epsilon$ for every time-step then $\mathrm{KL}(p_t \parallel q_t) = \mathcal{O}(\epsilon)$ as $\epsilon \to 0$ for every time-step.*

Proof We will demonstrate the lemma by induction noting that for $t = 1$ the state distributions are identical and hence their KL is zero. Assuming $\epsilon_t = \mathrm{KL}(p_t \parallel q_t) = \mathcal{O}(\epsilon)$ as $\epsilon \to 0$, let us compute the KL between state distributions for $t+1$

$$\begin{aligned}
\mathrm{KL}(p_{t+1} \parallel q_{t+1}) &= \int p_{t+1}(\boldsymbol{s}') \log \frac{p_{t+1}(\boldsymbol{s}')}{q_{t+1}(\boldsymbol{s}')}, \\
&\leq \iiint p_t(\boldsymbol{s},\boldsymbol{a}) p(\boldsymbol{s}'|\boldsymbol{a},\boldsymbol{s}) \log \frac{p_t(\boldsymbol{s},\boldsymbol{a}) p(\boldsymbol{s}'|\boldsymbol{a},\boldsymbol{s})}{q_t(\boldsymbol{s},\boldsymbol{a}) p(\boldsymbol{s}'|\boldsymbol{a},\boldsymbol{s})}, \qquad \text{(log sum inequality)} \\
&= \int p_t(\boldsymbol{s}') \int p_t(\boldsymbol{a}|\boldsymbol{s}') \log \frac{p_t(\boldsymbol{s}') p_t(\boldsymbol{a}|\boldsymbol{s}')}{q_t(\boldsymbol{s}') q_t(\boldsymbol{a}|\boldsymbol{s}')}, \\
&= \epsilon_t + \mathbb{E}_{s\sim p_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))]. \qquad (9)
\end{aligned}$$

Hence we have bounded the KL between state distributions at $t+1$ by the KL between state distributions and the expected KL between policies of the previous time-step t. Now we will express the KL between policies under the new state distributions, given by $\mathbb{E}_{s\sim p_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))]$, in terms of KL between policies under the previous state distribution, $\mathbb{E}_{s\sim q_t}\left[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))\right]$ which is bounded during policy update by ϵ, and $\mathrm{KL}(p_t \parallel q_t)$. To do so, we will use the assumption that the state distribution and the policy are Gaussian and linear-Gaussian. The complete demonstration is given in Appendix B, and we only report the following result

$$\mathbb{E}_{s\sim p_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))] \leq 2\epsilon \left(3\epsilon_t + d_s + 1\right). \qquad (10)$$

It is now easy to see that the combination of (9) and (10) together with the induction hypothesis yields $\mathrm{KL}(p_{t+1} \parallel q_{t+1}) = \mathcal{O}(\epsilon)$ as $\epsilon \to 0$. ■

Finally, the combination of Lemma 2 and Lemma 3 results in the following theorem, lower-bounding the change in policy return.

Theorem 4 *If for every time-step the state distributions p_t and q_t are Gaussian and the policies $p_t(.|\boldsymbol{s}_t)$ and $q_t(.|\boldsymbol{s}_t)$ are linear-Gaussian and if $\mathbb{E}_{s\sim q_t}\left[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))\right] \leq \epsilon$ for every time-step then*

$$J(p) - J(q) \geq \sum_{t=1}^{T} \mathbb{E}_{s\sim q_t, a\sim p_t(.|\boldsymbol{s})}\left[A_t^q(\boldsymbol{s},\boldsymbol{a})\right] - \sum_{t=1}^{T} \delta_t \mathcal{O}(\sqrt{\epsilon}).$$

Theorem 4 shows that we are able to obtain similar bounds than those derived in (Schulman et al., 2015) for our continuous state-action trajectory optimization setting with a bounded KL policy update in expectation under the previous state distribution. While, it

is not easy to apply Theorem 4 in practice to choose an appropriate step-size ϵ since $A_t^q(\boldsymbol{s},\boldsymbol{a})$ is generally only known approximately, Theorem 4 still shows that our constrained policy update will result in small changes in the overall behavior of the policy between successive iterations which is crucial in the approximate RL setting.

6. Related Work

In the Approximate Policy Iteration scheme (Szepesvari, 2010), policy updates can potentially decrease the expected reward, leading to policy oscillations (Wagner, 2011), unless the updated policy is 'close' enough to the previous one (Kakade and Langford, 2002). Bounding the change between π^i and π^{i+1} during the policy update step is thus a well studied idea in the Approximate Policy Iteration literature. Already in 2002, Kakade and Langford proposed the Conservative Policy Iteration (CPI) algorithm where the new policy π^{i+1} is obtained as a mixture of π^i and the greedy policy w.r.t. Q^i. The mixture parameter is chosen such that a lower bound of $J(\pi^{i+1}) - J(\pi^i)$ is positive and improvement is guaranteed. However, convergence was only asymptotic and in practice a single policy update would require as many samples as other algorithms would need to find the optimal solution (Pirotta et al., 2013b). Pirotta et al. (2013b) refined the lower bound of CPI by adding an additional term capturing the closeness between policies (defined as the matrix norm of the difference between the two policies), resulting in a more aggressive updates and better experimental results. However, both approaches only considered discrete action spaces. Pirotta et al. (2013a) provide an extension to continuous domains but only for single dimensional actions.

When the action space is continuous, which is typical in e.g. robotic applications, using a stochastic policy and updating it under a KL constraint to ensure 'closeness' of successive policies has shown several empirical successes (Daniel et al., 2012; Levine and Koltun, 2014; Schulman et al., 2015). However, only an empirical sample estimate of the objective function is generally optimized (Peters et al., 2010; Schulman et al., 2015), which typically requires a high number of samples and precludes it from a direct application to physical systems. The sample complexity can be reduced when a model of the dynamics is available (Levine and Koltun, 2014) or learned (Levine and Abbeel, 2014). In the latter work, empirical evidence suggests that good policies can be learned on high dimensional continuous state-action spaces with only a few hundred episodes. The counter part being that time-dependent dynamics are assumed to be linear, which is a simplifying assumption in many cases. Learning more sophisticated models using for example Gaussian Processes was experimented by Deisenroth and Rasmussen (2011) and Pan and Theodorou (2014) in the Policy Search and Trajectory Optimization context, but it is still considered to be a challenging task, see Deisenroth et al. (2013), chapter 3.

The policy update in Eq. (4) resembles that of (Peters et al., 2010; Daniel et al., 2012) with three main differences. First, without the assumption of a quadratic Q-Function, an additional weighted maximum likelihood step is required for fitting π^{i+1} to weighted samples as in the r.h.s of Eq. (4), since this policy might not be of the same policy class. As a result, the KL between π^i and π^{i+1} is no longer respected. Secondly, we added an entropy constraint in order to cope with the inherent non-stationary objective function maximized by the policy (Eq. 1) and to ensure that exploration is sustained, resulting in better quality

policies. Thirdly, their sample based optimization algorithm requires the introduction of a number of dual variables typically scaling at least linearly with the dimension of the state space, while we only have to optimize over two dual variables irrespective of the state space.

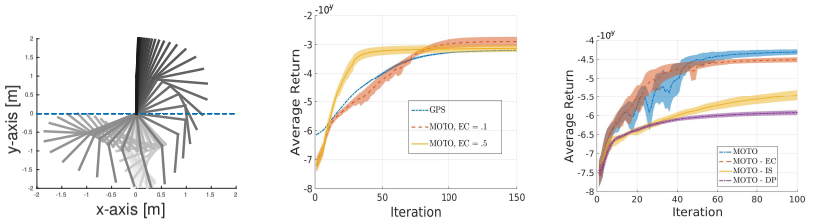
Most trajectory optimization methods are based on stochastic optimal control. These methods linearize the system dynamics and update the policy in closed form as a LQR. Instances of such algorithms are for example iLQG (Todorov, 2006), DDP (Theodorou et al., 2010), AICO (Toussaint, 2009) and its more robust variant (Rückert et al., 2014) and the trajectory optimization algorithm used in the GPS algorithm (Levine and Abbeel, 2014). These methods share the same assumptions as MOTO for ρ_t^i and π_t^i respectively considered to be of Gaussian and linear-Gaussian form. These methods face issues in maintaining the stability of the policy update and, similarly to MOTO, introduce additional constraints and regularizers to their update step. DDP, iLQG and AICO regularize the update by introducing a damping term in the matrix inversion step, while GPS uses a KL bound on successive trajectory distributions. However, as demonstrated in Sec. 7, the quadratic approximation of the Q-Function performed by MOTO seems to be empirically less detrimental to the quality of the policy update than the linearization of the system dynamics around the mean trajectory performed by related approaches.

7. Experimental Validation

MOTO is experimentally validated on a set of multi-link swing-up tasks and on a robot table tennis task. The experimental section aims at analyzing the proposed algorithm from four different angles: i) the quality of the returned policy comparatively to state-of-the-art trajectory optimization algorithms, ii) the effectiveness of the proposed variance reduction and sample reuse schemes, iii) the contribution of the added entropy constraint during policy updates in finding better local optima and iv) the ability of the algorithm to scale to higher dimensional problems. The experimental section concludes with a comparison to TRPO (Schulman et al., 2015), a state-of-the-art reinforcement learning algorithm that bounds the KL between successive policies; showcasing settings in which the time-dependent linear-Gaussian policies used by MOTO are a suitable alternative to neural networks.

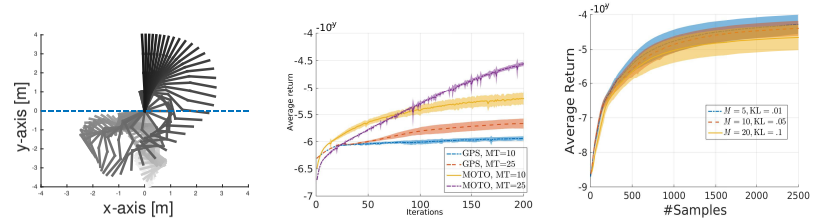
7.1 Multi-link Swing-up Tasks

A set of swing-up tasks involving a multi-link pole with respectively two and four joints (Fig. 1.a and 2.a) is considered in this section. The set of tasks includes several variants with different torque and joint limits, introducing additional non-linearities in the dynamics and resulting in more challenging control problems for trajectory optimization algorithms based on linearizing the dynamics. The state space consists of the joint positions and joint velocities while the control actions are the motor torques. In all the tasks, the reward function is split between an action cost and a state cost. The action cost is constant throughout the time-steps while the state cost is time-dependent and is equal to zero for all but the 20 last time-steps. During this period, a quadratic cost penalizes the state for not being the null vector, i.e. having zero velocity and reaching the upright position. Examples of successful swing-ups learned with MOTO are depicted in Fig. 1.a and 2.a.



(a) (b) (c)

Figure 1: a) Double link swing-up policy found by MOTO. b) Comparison between GPS and MOTO on the double link swing-up task (different torque limits and state costs are applied compared to c) and f). c) MOTO and its variants on the double link swing-up task: MOTO without the entropy constraint (EC), importance sampling (IS) or dynamic programming (DP). All plots are averaged over 15 runs.



(a) (b) (c)

Figure 2: a) Quad link swing-up policy found by MOTO. b) Comparison between GPS and MOTO on the quad link swing-up task with restricted joint limits and two different torque limits. c) MOTO on the double link swing-up task for varying number of rollouts per episode and step-sizes. All plots are averaged over 15 runs.

MOTO is compared to the trajectory optimization algorithm proposed in Levine and Abbeel (2014), that we will refer to as GPS.[4] We chose to compare MOTO and GPS as both use a KL constraint to bound the change in policy. As such, the choice of approximating the Q-Function with time-dependent quadratic models (as done in MOTO) in order to solve the policy update instead of linearizing the system dynamics around the mean trajectory (as done in most trajectory optimization algorithms) is better isolated. GPS and MOTO both use a time-dependent linear-Gaussian policy. In order to learn the linear

4. This is a slight abuse of notation as the GPS algorithm of (Levine and Abbeel, 2014) additionally feeds the optimized trajectory to an upper level policy. However, in this article, we are only interested in the trajectory optimization part.

model of the system dynamics, GPS reuses samples from different time-steps by learning a Gaussian mixture model on all the samples and uses this model as a prior to learn a joint Gaussian distribution $p(\boldsymbol{s}_t, \boldsymbol{a}_t, \boldsymbol{s}_{t+1})$ for every time-step. To single out the choice of linearizing the dynamics model or lack thereof from the different approaches to sample reuse, we give to both algorithm a high number of samples (200 and 400 rollouts per iteration for the double and quad link respectively) and bypass any form of sample reuse for both algorithms.

Fig. 1.b compares GPS to two configurations of MOTO on the double-link swing up task. The same initial policy and step-size ϵ are used by both algorithm. However, we found that GPS performs better with a smaller initial variance, as otherwise actions quickly hit the torque limits making the dynamics modeling harder. Fig. 1.b shows that even if the dynamics of the system are not linear, GPS manages to improve the policy return, and eventually finds a swing-up policy. The two configurations of MOTO have an entropy reduction constant β_0 of .1 and .5. The effect of the entropy constraint is similar to the one observed in the stochastic search domain by (Abdolmaleki et al., 2015). Specifically, a smaller entropy reduction constant β_0 results in an initially slower convergence but ultimately leads to higher quality policies. In this particular task, MOTO with $\beta_0 = .1$ manages to slightly outperform GPS.

Next, GPS and MOTO are compared on the quad link swing-up task. We found this task to be significantly more challenging than the double link and to increase the difficulty further, soft joint limits are introduced on the three last joints in the following way: whenever a joint angle exceeds in absolute value the threshold $\frac{2}{3}\pi$, the desired torque of the policy is ignored in favor of a linear-feedback controller that aims at pushing back the joint angle within the constrained range. As a result, Fig. 2.b shows that GPS can barely improve its average return (with the torque limits set to 25, as in the double link task.) while MOTO performs significantly better. Finally, the torque limits are reduced even further but MOTO still manages to find a swing-up policy as demonstrated by Fig. 2.a.

In the last set of comparisons, the importance of each of the components of MOTO is assessed on the double link experiment. The number of rollouts per iteration is reduced to $M = 20$. Fig. 1.c shows that: i) the entropy constraint provides an improvement on the quality of the policy in the last iterations in exchange of a slower initial progress, ii) importance sampling greatly helps in speeding-up the convergence and iii) the Monte-Carlo estimate of $\hat{Q}_i^t$ is not adequate for the smaller number of rollouts per iterations, which is further exacerbated by the fact that sample reuse of transitions from different time-steps is not possible with the Monte-Carlo estimate.

Finally, we explore on the double-link swing-up task several values of M, trying to find the balance between performing a small number of rollouts per iterations with a small step-size ϵ versus having a large number of rollouts for the policy evaluation that would allow to take larger update steps. To do so, we start with an initial $M = 20$ and successively divide this number by two until $M = 5$. In each case, the entropy reduction constant is set such that, for a similar number of rollouts, the entropy is reduced by the same amount, while we choose γ', the discount of the state sample weights as $\gamma' = \gamma^{M/M'}$ to yield again

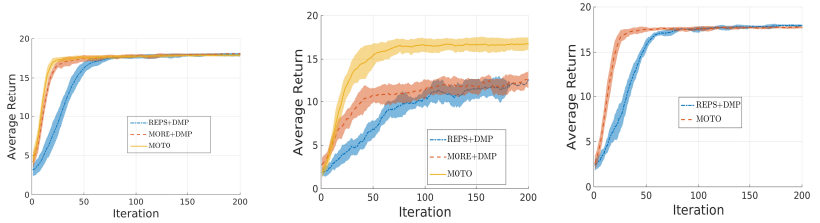


Figure 3: a) Comparison on the robot table tennis task with no noise on the initial velocity of the ball. b) Comparison on the robot table tennis task with Gaussian noise during the ball bounce on the table. c) Comparison on the robot table tennis task with initial velocity sampled uniformly in a 15cm range.

a similar sample decay after the same number of rollouts have been performed. Tuning ϵ was, however, more complicated and we tested several values on non-overlapping ranges for each M and selected the best one. Fig. 2.c shows that, on the double link swing-up task, a better sample efficiency is achieved with a smaller M. However, the improvement becomes negligible from $M = 10$ to $M = 5$. We also noticed a sharp decrease in the number of effective samples when M tends to 1. In this limit case, the complexity of the mixture policy $z^{1:i}$ in the denominator of the importance ratio increases with the decrease of M and might become a poor representation of the data set. Fitting a simpler state-action distribution that is more representative of the data can be the subject of future work in order to further improve the sample efficiency of the algorithm, which is crucial for applications on physical systems.

7.2 Robot Table Tennis

The considered robot table tennis task consists of a simulated robotic arm mounted on a floating base, having a racket on the end effector. The task of the robot is to return incoming balls using a forehand strike to the opposite side of the table (Fig. 4). The arm has 9 degrees of freedom comprising the six joints of the arm and the three linear joints of the base allowing (small) 3D movement. Together with the joint velocities and the 3D position of the incoming ball, the resulting state space is of dimension $d_s = 21$ and the action space is of dimension $d_a = 9$ and consists of direct torque commands.

We use the analytical player of Mülling et al. (2011) to generate a single forehand stroke, which is subsequently used to learn from demonstration the initial policy π^1. The analytica player comprises a waiting phase (keeping the arm still), a preparation phase, a hitting phase and a return phase, which resets the arm to the waiting position of the arm. Only the preparation and the hitting phase are replaced by a learned policy. The total control time for the two learned phases is of 300 time-steps at 500hz, although for the MOTO

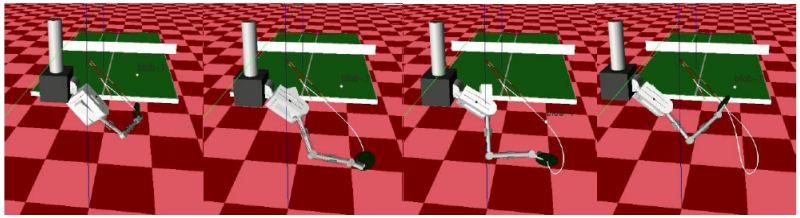


Figure 4: Robot table tennis setting and a forehand stroke learned by MOTO upon a spinning ball.

algorithm we subsequently divide the control frequency by a factor of 10 resulting in a time-dependent policy of 30 linear-Gaussian controllers.

The learning from demonstration step is straightforward and only consists in averaging the torque commands of every 10 time-steps and using these quantities as the initial bias for each of the 30 controllers. Although this captures the basic template of the forehand strike, no correlation between the action and the state (e.g. the ball position) is learned from demonstration as the initial gain matrix K for all the time-steps is set to the null matrix. Similarly, the exploration in action space is uninformed and initially set to the identity matrix.

Three settings of the task are considered, a noiseless case where the ball is launched with the same initial velocity, a varying context setting where the initial velocity is sampled uniformly within a fixed range and the noisy bounce setting where a Gaussian noise is added to both the x and y velocities of the ball upon bouncing on the table, to simulate the effect of a spin.

We compare MOTO to the policy search algorithm REPS (Kupcsik et al., 2013) and the stochastic search algorithm MORE (Abdolmaleki et al., 2015) that shares a related information-theoretic update. Both algorithms will optimize the parameters of a Dynamical Movement Primitive (DMP) (Ijspeert and Schaal, 2003). A DMP is a non-linear attractor system commonly used in robotics. The DMP is initialized from the same single trajectory and the two algorithm will optimize the goal joint positions and velocities of the attractor system. Note that the DMP generates a trajectory of states, which will be tracked by a linear controller using the inverse dynamics. While MOTO will directly output the torque commands and does not rely on this model.

Fig. 3.a and 3.c show that our algorithm converges faster than REPS and to a smaller extent than MORE in both the noiseless and the varying context setting. This is somewhat surprising since MOTO with its time-dependent linear policy have a much higher number of parameters to optimize than the 18 parameters of the DMP's attractor. However, the resulting policy in both cases is slightly less good than that of MORE and REPS. Note that for the varying context setting, we used a contextual variant of REPS that learns a mapping from the initial ball velocity to the DMP's parameters. MORE, on the other hand couldn't be compared in this setting. Finally, Fig. 3.b shows that our policy is successfully

capable of adapting to noise at ball bounce, while the other methods fail to do so since the trajectory of the DMP is not updated once generated.

7.3 Comparison to Neural Network Policies

Recent advances in reinforcement learning using neural network policies and supported by the ability of generating and processing large amounts of data allowed impressive achievements such as playing Atari at human level (Mnih et al., 2015) or mastering the game of Go (Silver et al., 2016). On continuous control tasks, success was found by combining trajectory optimization and supervised learning of a neural network policy (Levine and Abbeel, 2014), or by directly optimizing the policy's neural network using reinforcement learning (Lillicrap et al., 2015; Schulman et al., 2015). The latter work, side-stepping trajectory optimization to directly optimize a neural network policy raises the question as to whether the linear-Gaussian policies used in MOTO and related algorithms provide any benefit compared to neural network policies.

To this end, we propose to compare on the multi-link swing-up tasks of Sec. 7.1, MOTO learning a time-dependent linear-Gaussian policy to TRPO (Schulman et al., 2015) learning a neural network policy. We chose TRPO as our reinforcement learning baseline for its state-of-the-art performance and because of its similar policy update than that of MOTO (both bound the KL between successive policies). Three variants of TRPO are considered while for MOTO, we refrain from using importance sampling (Sec. 4.2) since similar techniques such as off-policy policy evaluation can be used for TRPO.

First, MOTO is compared to a default version of TRPO using OpenAI's baselines implementation (Dhariwal et al., 2017) where TRPO optimizes a neural network for both learning the policy and the V-Function. Default parameters are used except for the KL divergence constraint where we set $\epsilon = .1$ for TRPO to match MOTO's setting. Note that because the rewards are time-dependent (distance to the upright position penalized only for the last 20 steps, see Sec. 7.1) we add time as an additional entry to the state description. Time entry is in the interval $[0, 1]$ (current time-step divided by horizon T) and is fed to both the policy and V-Function neural networks. This first variant of TRPO answers the question: is there any benefit for using MOTO with its time-dependent linear-Gaussian policy instead of a state-of-the-art deep RL implementation with a neural network policy.

The second considered baseline uses the same base TRPO algorithm but replaces the policy evaluation using a neural network V-Function with the same policy evaluation used by MOTO (Sec. 4), back-propagating a quadratic V-Function. In this variant of TRPO the time-entry is dropped for the V-Function. This second baseline better isolates the policy update, which is the core of both algorithms, from the learning of the V-Function which could be interchanged.

Finally, we consider a third variant of TRPO that uses both the quadratic V-Function and a time-dependent linear-Gaussian policy with diagonal covariance matrix (standard formulation and implementation of TRPO does not support full covariance exploration noise). The time entry is dropped for both the V-Function and the policy in this third baseline. While both algorithms bound the KL divergence between successive policies, there are still a few differentiating factors between this third baseline and MOTO. First, TRPO bounds the KL of the whole policy while MOTO solves a policy update for each time-step independently

(but still results in a well-founded approximate policy iteration algorithm as discussed in Sec. 5). In practice the KL divergence upon update for every time-step for MOTO is often equal to ϵ and hence both MOTO and TRPO result in the same KL divergence of the overall policy (in expectation of the state distribution) while the KL divergence of the sub-policies (w.r.t. the time-step) may vary. Secondly, MOTO performs a quadratic approximation of the Q-Function and solves the policy update exactly while TRPO performs a quadratic approximation of the KL constraint and solves the policy update using conjugate gradient descent. TRPO does not solve the policy update in closed form because it would require a matrix inversion and the matrix to invert has the dimensionality of the number of policy parameters. In contrast, MOTO can afford the closed form solution because the matrix to invert has the dimensionality of the action space which is generally significantly smaller than the number of policy parameters.

Fig. 5 shows the learning performance of MOTO and three TRPO variants on the double link and quadruple link swing-up tasks (Sec. 7.1). In both tasks MOTO outperforms all three TRPO variants albeit when TRPO is combined with the quadratic V-Function (second variant), it initially outperforms MOTO on the double link swing-up task. The quadratic V-Function befits these two tasks in particular and the quadratic regulation setting more generally because the reward is a quadratic function of the state-action pair (here the negative squared distance to the upright position and a quadratic action cost). However, MOTO makes better use of the task's nature and largely outperforms the third variant of TRPO despite having a similar policy evaluation step and using the same policy class. In conclusion, while neural networks can be a general purpose policy class demonstrating success on a wide variety of tasks, on specific settings such as on quadratic regulator tasks, trajectory-based policy optimization is able to outperform deep RL algorithms. MOTO in particular, which does not rely on a linearization of the dynamics around the mean trajectory is able to handle quadratic reward problems with highly non-linear dynamics such as the quadruple link swing-up task and outperform state-of-the-art trajectory optimization algorithms (Sec. 7.1) as a result.

8. Conclusion

We proposed in this article MOTO, a new trajectory-based policy optimization algorithm that does not rely on a linearization of the dynamics. Yet, an efficient policy update could be derived in closed form by locally fitting a quadratic Q-Function. We additionally conducted a theoretical analysis of the constrained optimization problem solved during the policy update. We showed that the upper bound on the expected KL between successive policies leads to only a small drift in successive state distributions which is a key property in the approximate policy iteration scheme.

The use of a KL constraint is widely spread including in other trajectory optimization algorithms. The experiments demonstrate however that our algorithm has an increased robustness towards non-linearities of the system dynamics when compared to a closely related trajectory optimization algorithm. It appears as such that the simplification resulting from considering a local linear approximation of the dynamics is more detrimental to the overall convergence of the algorithm than a local quadratic approximation of the Q-Function.

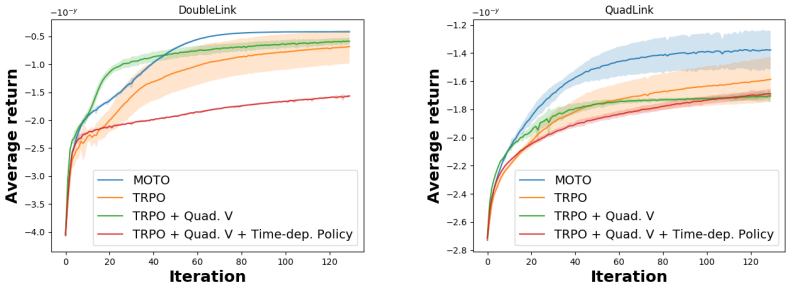


Figure 5: Comparisons on multi-link swing-up tasks between MOTO and TRPO. TRPO uses a neural network policy and V-Function (default) or a quadratic V-Function and a time-dependent linear-Gaussian policy as in MOTO. Quadratic V-Function is a good fit for such tasks and allows MOTO to outperform neural network policies on the double and quadruple link swing-up tasks. Rewards of the original task divided by 1e4 to accomodate with the neural network V-Function. Plots averaged over 11 independent runs.

On simulated robotics tasks, we demonstrated the merits of our approach compared to direct policy search algorithms that optimize commonly used low dimensional parameterized policies. The main strength of our approach is its ability to learn reactive policies capable of adapting to external perturbations in a sample efficient way. However, the exploration scheme of our algorithm based on adding Gaussian noise at every time-step is less structured than that of low dimensional parameterized policies and can be harmful to the robot. One of the main addition that would ease the transition from simulation to physical systems is thus to consider the safety of the exploration scheme of the algorithm. On a more technical note, and as the V-Function can be of any shape in our setting, the use of a more complex function approximator such as a deep network can be considered in future extensions to allow for a more refined bias-variance trade-off.

Acknowledgments

The research leading to these results has received funding from the DFG Project LearnRobotS under the SPP 1527 Autonomous Learning, from the Intel Corporation, and from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 640554 (SKILLS4ROBOTS). Computing time for the experiments was granted from Lichtenberg cluster.

Appendix A. Dual Function Derivations

Recall the quadratic form of the Q-Function $\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})$ in the action a and state s

$$\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a}) = \frac{1}{2}\boldsymbol{a}^T Q_{aa}\boldsymbol{a} + \boldsymbol{a}^T Q_{as}\boldsymbol{s} + \boldsymbol{a}^T \boldsymbol{q}_a + q(\boldsymbol{s}).$$

The new policy $\pi'_t(\boldsymbol{a}|\boldsymbol{s})$ solution of the constrained maximization problem is again of linear-Gaussian form and given by

$$\pi'_t(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}|FL\boldsymbol{s} + F\boldsymbol{f}, F(\eta^* + \omega^*)),$$

such that the gain matrix, bias and covariance matrix of π'_t are function of matrices F and L and vector $\boldsymbol{f}$ where

$$F = (\eta^*\Sigma_t^{-1} - Q_{aa})^{-1}, \qquad L = \eta^*\Sigma_t^{-1}K_t + Q_{as},$$
$$\boldsymbol{f} = \eta^*\Sigma_t^{-1}\boldsymbol{k}_t + \boldsymbol{q}_a,$$

with η^* and ω^* being the optimal Lagrange multipliers related to the KL and entropy constraints, obtained by minimizing the dual function

$$g_t(\eta, \omega) \;=\; \eta\epsilon - \omega\beta + (\eta + \omega)\int \tilde{\rho}_t(\boldsymbol{s}) \log\left(\int \pi(\boldsymbol{a}|\boldsymbol{s})^{\eta/(\eta+\omega)} \exp\left(\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})/(\eta + \omega)\right)\right).$$

From the quadratic form of $\tilde{Q}_t(\boldsymbol{s}, \boldsymbol{a})$ and by additionally assuming that the state distribution is approximated by $\tilde{\rho}_t(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_s, \Sigma_s)$, the dual function simplifies to

$$g_t(\eta, \omega) = \eta\epsilon - \omega\beta + \boldsymbol{\mu}_s^T M\boldsymbol{\mu}_s + \mathrm{tr}(\Sigma_s M) + \boldsymbol{\mu}_s^T\boldsymbol{m} + m_0,$$

where M, $\boldsymbol{m}$ and m_0 are defined by

$$M = \frac{1}{2}\left(L^T F L - \eta K_t^T\Sigma_t^{-1}K_t\right), \; \boldsymbol{m} = L^T F\boldsymbol{f} - \eta K_t^T\Sigma_t^{-1}\boldsymbol{k}_t,$$
$$m_0 = \frac{1}{2}(\boldsymbol{f}^T F\boldsymbol{f} - \eta\boldsymbol{k}_t^T\Sigma_t^{-1}\boldsymbol{k}_t - \eta\log|2\pi\Sigma_t| + (\eta + \omega)\log|2\pi(\eta + \omega)F|).$$

The convex dual function g_t can be efficiently minimized by gradient descent and the policy update is performed upon the computation of η^* and ω^*. The gradient w.r.t. η and ω is given by[5]

$$\frac{\partial g_t(\eta, \omega)}{\partial \eta} = \mathrm{cst} + \mathrm{lin} + \mathrm{quad}$$
$$\mathrm{cst} = \epsilon - \frac{1}{2}\left(\boldsymbol{k}_t - F\boldsymbol{f}\right)^T\Sigma_t^{-1}\left(\boldsymbol{k}_t - F\boldsymbol{f}\right) - \frac{1}{2}[\log|2\pi\Sigma_t| - \log|2\pi(\eta + \omega)F| + (\eta + \omega)\mathrm{tr}(\Sigma_t^{-1}F) - d_a].$$
$$\mathrm{lin} = \left((K_t - FL)\boldsymbol{\mu}_s\right)^T\Sigma_t^{-1}(F\boldsymbol{f} - \boldsymbol{k}_t).$$
$$\mathrm{quad} = \boldsymbol{\mu}_s^T(K_t + FL)^T\Sigma_t^{-1}(K_t + FL)\boldsymbol{\mu}_s + \mathrm{tr}(\Sigma_s(K_t + FL)^T\Sigma_t^{-1}(K_t + FL))$$
$$\frac{\partial g_t(\eta, \omega)}{\partial \omega} = -\beta + \frac{1}{2}(d_a + \log|2\pi(\eta + \omega)F|).$$

5. cst, lin, quad, F, L and $\boldsymbol{f}$ all depend on η and ω. We dropped the dependency from the notations for compactness. d_a is the dimensionality of the action.

Appendix B. Bounding the Expected Policy KL Under the Current State Distribution

Let the state distributions and policies be parameterized as following: $p_t(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_p, \Sigma_p)$, $q_t(\boldsymbol{s}) = \mathcal{N}(\boldsymbol{s}|\boldsymbol{\mu}_q, \Sigma_q)$, $p_t(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}|K\boldsymbol{s} + \boldsymbol{b}, \Sigma)$ and $q_t(\boldsymbol{a}|\boldsymbol{s}) = \mathcal{N}(\boldsymbol{a}|K'\boldsymbol{s} + \boldsymbol{b}', \Sigma')$. The change of the state distribution in the expected KL constraint of our policy update, given by $\mathbb{E}_{\boldsymbol{s}\sim q_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))]$ from state distribution q_t to p_t will only affect the part of the KL that depends on the state.

We give as a reminder the general formula for the KL between two Gaussian distributions $l = \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ and $l' = \mathcal{N}(\boldsymbol{\mu}', \Sigma')$

$$\mathrm{KL}(l \parallel l') = \frac{1}{2}\left(\mathrm{tr}(\Sigma'^{-1}\Sigma) + (\boldsymbol{\mu} - \boldsymbol{\mu}')^T\Sigma'^{-1}(\boldsymbol{\mu} - \boldsymbol{\mu}') - dim + \log\frac{|\Sigma'|}{|\Sigma|}\right).$$

For the linear-Gaussian policies, and since only the mean of the policies depend on the state, the change of state distribution in the expected KL will only affect the term

$$(K\boldsymbol{s} - K'\boldsymbol{s})^T\Sigma'(K\boldsymbol{s} - K'\boldsymbol{s}) = \boldsymbol{s}^T M\boldsymbol{s},$$

with p.s.d. matrix $M = (K - K')^T\Sigma'(K - K')$. Thus it suffices to bound the expectation $\int p_t(\boldsymbol{s})\boldsymbol{s}^T M\boldsymbol{s}$ since the rest of the KL terms are already bounded by ϵ, yielding

$$\mathbb{E}_{s\sim p_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))] \leq \epsilon + \frac{1}{2}\int p_t(\boldsymbol{s})\boldsymbol{s}^T M\boldsymbol{s},$$
$$= \epsilon + \frac{1}{2}\left(\boldsymbol{\mu}_p M\boldsymbol{\mu}_p + \mathrm{tr}(M\Sigma_p)\right),$$

where we exploited the Gaussian nature of p_t in the second line of the equation. We will now bound both $\boldsymbol{\mu}_p M\boldsymbol{\mu}_p$ and $\mathrm{tr}(M\Sigma_p)$. First, note that for any two p.d. matrices Σ and Σ' we have

$$\mathrm{tr}(\Sigma'^{-1}\Sigma) + -d_s + \log\frac{|\Sigma'|}{|\Sigma|} \geq 0. \tag{11}$$

This immediately follows from the non-negativity of the KL. Since, if for some Σ and Σ', eq. (11) is negative then the KL for two Gaussian distributions having Σ and Σ' as covariance matrices and sharing the same mean would be negative which is not possible. Hence it also follows that

$$(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p)^T\Sigma_q^{-1}(\boldsymbol{\mu}_q - \boldsymbol{\mu}_p) \leq 2\epsilon_t, \tag{12}$$

from the bounded KL induction hypothesis between p_t and q_t.

For the expected policy KL, since the part that does not depend on s is positive as in eq. (11), it can thus be dropped out yielding

$$\mathbb{E}_{s\sim q_t}[\mathrm{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))] \leq \epsilon \Rightarrow \int q_t(\boldsymbol{s})\boldsymbol{s}^T M\boldsymbol{s} \leq 2\epsilon,$$
$$\Rightarrow \boldsymbol{\mu}_q M\boldsymbol{\mu}_q + \mathrm{tr}(M\Sigma_q) \leq 2\epsilon. \tag{13}$$

Also note that for any p.s.d. matrices A and B, $\text{tr}(AB) \geq 0$. Letting $\boldsymbol{x} = \boldsymbol{\mu}_p - \boldsymbol{\mu}_q$, we have

$$\begin{aligned}\boldsymbol{x}^T M \boldsymbol{x} &= \text{tr}(\boldsymbol{x}\boldsymbol{x}^T M),\\ &= \text{tr}(\Sigma_q^{-1}\boldsymbol{x}\boldsymbol{x}^T M \Sigma_q),\\ &\leq \text{tr}(\Sigma_q^{-1}\boldsymbol{x}\boldsymbol{x}^T)\text{tr}(M\Sigma_q),\\ &\leq 4\epsilon_t\epsilon.\end{aligned}$$

Third line is due to Cauchy-Schwarz inequality and positiveness of traces while the last one is from eq. (12) and (13). Finally, from the reverse triangular inequality, we have

$$\begin{aligned}\boldsymbol{\mu}_p M \boldsymbol{\mu}_p &\leq \boldsymbol{x}^T M \boldsymbol{x} + \boldsymbol{\mu}_q M \boldsymbol{\mu}_q,\\ &\leq 2\epsilon(1 + 2\epsilon_t),\end{aligned}$$

Which concludes the bounding of $\boldsymbol{\mu}_p M \boldsymbol{\mu}_p$.

To bound $\text{tr}(M\Sigma_p)$ we can write

$$\begin{aligned}\text{tr}(M\Sigma_p) &= \text{tr}(M\Sigma_q\Sigma_q^{-1}\Sigma_p),\\ &\leq \text{tr}(M\Sigma_q)\text{tr}(\Sigma_q^{-1}\Sigma_p).\end{aligned}$$

We know how to bound $\text{tr}(M\Sigma_q)$ from Eq. (13). While $\text{tr}(\Sigma_q^{-1}\Sigma_p)$ appears in the bounded KL between state distributions. Bounding $\text{tr}(\Sigma_q^{-1}\Sigma_p)$ is equivalent to solving $\max \sum \lambda_i$ under constraint $\sum \lambda_i - d_s - \sum \log \lambda_i \leq 2\epsilon_t$, where the $\{\lambda_i\}$ are the eigenvalues of $\Sigma_q^{-1}\Sigma_p$. For any solution $\{\lambda_i\}$, we can keep the same optimization objective using equal $\{\lambda_i'\}$ where for each i, $\lambda_i' = \bar{\lambda} = \sum \lambda_i / d_s$ is the average lambda. This transformation will at the same time reduce the value of the constraint since $-d_s \log \bar{\lambda} \leq -\sum \log \lambda_i$ from Jensen's inequality. Hence the optimum is reached when all the λ_i are equal, and the constraint is active (i.e. $d_s\bar{\lambda} - d_s - d_s \log \bar{\lambda} = 2\epsilon_t$). Finally, the constraint is at a minimum for $\bar{\lambda} = 1$, hence $\bar{\lambda} > 1$. The maximum is reached at

$$\begin{aligned}& d_s\bar{\lambda} - d_s - d_s \log \bar{\lambda} = 2\epsilon_t\\ \Leftrightarrow & \bar{\lambda} - \log \bar{\lambda} = \frac{2\epsilon_t}{d_s} + 1\\ \Rightarrow & \bar{\lambda} \leq \left(\frac{2\epsilon_t}{d_s} + 1\right)\frac{e}{e-1}\\ \Rightarrow & \text{tr}(\Sigma_q^{-1}\Sigma_p) \leq 4\epsilon_t + 2d_s \qquad (14)\end{aligned}$$

The equation in the second line has a unique solution ($f(\lambda) = \lambda - \log \lambda$ is a strictly increasing function for $\lambda > 1$) for which no closed form expression exists. We thus lower bound f by $g(\lambda) = \frac{e-1}{e}\lambda$ and solve the equation for g which yields an upper bound of the original equation that is further simplified in the last inequality.
Eq. (13) and (14) yield $\text{tr}(M\Sigma_p) \leq 2\epsilon(4\epsilon_t + 2d_s)$ and grouping all the results yields

$$\mathbb{E}_{s\sim p_t}[\text{KL}(p_t(.|\boldsymbol{s}) \parallel q_t(.|\boldsymbol{s}))] \leq 2\epsilon\,(3\epsilon_t + d_s + 1)$$

References

A. Abdolmaleki, R. Lioutikov, J. Peters, N. Lau, L. Pualo Reis, and G. Neumann. Model-based relative entropy stochastic search. In *Advances in Neural Information Processing Systems (NIPS)*. 2015.

R. Akrour, A. Abdolmaleki, H. Abdulsamad, and G. Neumann. Model-free trajectory optimization for reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2016.

D. P. Bertsekas. *Dynamic programming and optimal control*. Athena Scientific, 1995.

S. Bhojanapalli, A. T. Kyrillidis, and S. Sanghavi. Dropping convexity for faster semi-definite optimization. *CoRR*, 2015.

C. Daniel, G. Neumann, and J. Peters. Hierarchical Relative Entropy Policy Search. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2012.

M. Deisenroth and C. Rasmussen. PILCO: A Model-Based and Data-Efficient Approach to Policy Search. In *International Conference on Machine Learning (ICML)*, 2011.

M. P. Deisenroth, G. Neumann, and J. Peters. A Survey on Policy Search for Robotics. *Foundations and Trends in Robotics*, 2013.

P. Dhariwal, C. Hesse, O. Klimov, A. Nichol, M. Plappert, A. Radford, J. Schulman, S. Sidor, and Y. Wu. Openai baselines. `https://github.com/openai/baselines`, 2017.

A. Ijspeert and S. Schaal. Learning attractor landscapes for learning motor primitives. In *Advances in Neural Information Processing Systems (NIPS)*. 2003.

S. Kakade. *On the Sample Complexity of Reinforcement Learning*. PhD thesis, University College London, 2003.

S. Kakade and J. Langford. Approximately optimal approximate reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2002.

A. G. Kupcsik, M. P. Deisenroth, J. Peters, and G. Neumann. Data-efficient generalization of robot skills with contextual policy search. In *The Conference on Artificial Intelligence (AAAI)*, 2013.

S. Levine and P. Abbeel. Learning neural network policies with guided policy search under unknown dynamics. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

S. Levine and V. Koltun. Learning complex neural network policies with trajectory optimization. In *International Conference on Machine Learning (ICML)*, 2014.

T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra. Continuous control with deep reinforcement learning. *CoRR*, 2015.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 2015.

K. Mülling, J. Kober, and J. Peters. A biomimetic approach to robot table tennis. *Adaptive Behavior Journal*, 2011.

Y. Pan and E. Theodorou. Probabilistic differential dynamic programming. In *Advances in Neural Information Processing Systems (NIPS)*. 2014.

J. Peters, K. Mülling, and Y. Altun. Relative entropy policy search. In *Conference on Artificial Intelligence (AAAI)*, 2010.

M. Pirotta, M. Restelli, and L. Bascetta. Adaptive step-size for policy gradient methods. In *Advances in Neural Information Processing Systems (NIPS)*. 2013a.

M. Pirotta, M. Restelli, A. Pecorino, and D. Calandriello. Safe policy iteration. In *International Conference on Machine Learning (ICML)*, 2013b.

E. A. Rückert, M. Mindt, J. Peters, and G. Neumann. Robust policy updates for stochastic optimal control. In *International Conference on Humanoid Robots (Humanoids)*, 2014.

J. Schulman, S. Levine, P. Abbeel, M. I. Jordan, and P. Moritz. Trust region policy optimization. In *International Conference on Machine Learning (ICML)*, 2015.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 2016.

C. Szepesvari. *Algorithms for Reinforcement Learning*. Morgan & Claypool, 2010.

E. Theodorou, J. Buchli, and S. Schaal. Path integral stochastic optimal control for rigid body dynamics. In *International Symposium on Approximate Dynamic Programming and Reinforcement Learning (ADPRL)*, 2009.

E. Theodorou, Y. Tassa, and E. Todorov. Stochastic differential dynamic programming. In *American Control Conference (ACC)*, 2010.

E. Todorov. Optimal control theory. *Bayesian Brain*, 2006.

E. Todorov and Y. Tassa. Iterative local dynamic programming. In *International Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL)*, 2009.

M. Toussaint. Robot trajectory optimization using approximate inference. In *International Conference on Machine Learning (ICML)*, 2009.

P. Wagner. A reinterpretation of the policy oscillation phenomenon in approximate policy iteration. In *Advances in Neural Information Processing Systems (NIPS)*. 2011.

Journal of Machine Learning Research 19 (2018) 1-53 Submitted 6/17; Revised 7/18; Published 8/18

Regularized Optimal Transport and the Rot Mover's Distance

Arnaud Dessein arnaud.dessein@qucit.com
Institut de Mathématiques de Bordeaux
CNRS, Université de Bordeaux
351 Cours de la Libération, 33405 Talence, France
Qucit
213 Cours Victor Hugo, 33130 Bègles, France

Nicolas Papadakis nicolas.papadakis@math.u-bordeaux.fr
Institut de Mathématiques de Bordeaux
CNRS, Université de Bordeaux
351 Cours de la Libération, 33405 Talence, France

Jean-Luc Rouas jean-luc.rouas@labri.fr
Laboratoire Bordelais de Recherche en Informatique
CNRS, Université de Bordeaux
351 Cours de la Libération, 33405 Talence, France

Editor: Francis Bach

Abstract

This paper presents a unified framework for smooth convex regularization of discrete optimal transport problems. In this context, the regularized optimal transport turns out to be equivalent to a matrix nearness problem with respect to Bregman divergences. Our framework thus naturally generalizes a previously proposed regularization based on the Boltzmann-Shannon entropy related to the Kullback-Leibler divergence, and solved with the Sinkhorn-Knopp algorithm. We call the regularized optimal transport distance the rot mover's distance in reference to the classical earth mover's distance. By exploiting alternate Bregman projections, we develop the alternate scaling algorithm and non-negative alternate scaling algorithm, to compute efficiently the regularized optimal plans depending on whether the domain of the regularizer lies within the non-negative orthant or not. We further enhance the separable case with a sparse extension to deal with high data dimensions. We also instantiate our framework and discuss the inherent specificities for well-known regularizers and statistical divergences in the machine learning and information geometry communities. Finally, we demonstrate the merits of our methods with experiments using synthetic data to illustrate the effect of different regularizers, penalties and dimensions, as well as real-world data for a pattern recognition application to audio scene classification.

Keywords: alternate projections, convex analysis, regularized optimal transport, rot mover's distance, statistical divergences

1. Introduction

A recurrent problem in statistical machine learning is the choice of a relevant distance measure to compare probability distributions. Various information divergences are famous, among which Euclidean, Mahalanobis, Kullback-Leibler, Itakura-Saito, Hellinger, χ^2, ℓ_p (quasi-)norm, total variation, logistic loss function, or more general Csiszár and Bregman divergences and parametric families of such divergences such as α- and β-divergences.

An alternative family of distances between probability distributions can be introduced in the framework of optimal transport (OT). Rather than performing a pointwise comparison of the distributions, the idea is to quantify the minimal effort for moving the probability mass of one distribution to the other, where the transport plan to move the mass is optimized according to a given ground cost. This makes OT distances suitable and robust in certain applications, notably in the field of computer vision where the discrete OT distance, also known as earth mover's distance (EMD), has been popularized to compare histograms of features for pattern recognition tasks (Rubner et al., 2000).

Despite its appealing theoretical properties, intuitive formulation, and excellent performance in various problems of information retrieval, the computation of the EMD involves solving a linear program whose cost quickly becomes prohibitive with the data dimension. In practice, the best algorithms currently proposed, such as the network simplex (Ahuja et al., 1993), scale at least with a super-cubic complexity. Embeddings of the distributions can be used to approximate the EMD with linear complexity (Indyk and Thaper, 2003; Grauman and Darrell, 2004; Shirdhonkar and Jacobs, 2008), and the network simplex can be modified to run in quadratic time (Gudmundsson et al., 2007; Ling and Okada, 2007; Pele and Werman, 2009). Nevertheless, the distortions inherent to such embeddings (Naor and Schechtman, 2007), and the exponential increase of costs incurred by such modifications, make these approaches inapplicable for dimensions higher than four. Instead, multi-scale strategies (Oberman and Ruan, 2015) and shortcut paths (Schmitzer, 2016a) can speed up the estimation of the exact optimal plan. These approaches are yet limited to particular convex costs such as ℓ_2, while other costs such as ℓ_1 and truncated or compressed versions are often preferred in practice for an increased robustness to data outliers (Pele and Werman, 2008, 2009; Rabin et al., 2009). For general applications, a gain in performance can also be obtained with a cost directly learned from labeled data (Cuturi and Avis, 2014). The aforementioned accelerated methods that are dedicated to ℓ_2 or convex costs are thus not adapted in this context.

On another line of research, the regularization of the transport plan, for example via graph modeling (Ferradans et al., 2014), has been considered to deal with noisy data, though this latter approach does not address the computational issue of efficiency for high dimensions. In this continuity, an entropic regularization was shown to admit an efficient algorithm with quadratic complexity that speeds up the computation of solutions by several orders of magnitude, and to improve performance on applications such as handwritten digit recognition (Cuturi, 2013). In addition, a tailored computation can be obtained via convolution for specific ground costs (Solomon et al., 2015). Since the introduction of the entropic regularization, OT has benefited from extensive developments in the machine learning community, with applications such as label propagation (Solomon et al., 2014), domain adaptation (Courty et al., 2015), matrix factorization (Zen et al., 2014), dictionary learning (Rolet et al., 2016; Schmitz et al., 2018), barycenter computation (Cuturi and Peyré, 2016), geodesic principal component analysis (Bigot et al., 2013; Seguy and Cuturi, 2015; Cazelles et al., 2017), data fitting (Frogner et al., 2015), statistical inference (Bernton et al.,

2017), training of Boltzmann machines (Montavon et al., 2016) and generative adversarial networks (Arjovsky et al., 2017; Bousquet et al., 2017; Genevay et al., 2017).

With the entropic regularization, the gain in computational time is only important for high dimensions or large levels of regularization. For low regularization, advanced optimization strategies can still be used to obtain a significant speed-up (Thibault et al., 2017; Schmitz et al., 2018). It is also a well-known effect that the entropic regularization overspreads the transported mass, which may be undesirable for certain applications as in the case of interpolation purposes. An interesting perspective of these works, however, is that many more regularizers are worth investigating to solve OT problems both efficiently and robustly (Galichon and Salanié, 2015; Muzellec et al., 2018; Blondel et al., 2017). This is the idea we address in the present work, focusing on smooth convex regularization.

1.1 Notations

For the sake of simplicity, we consider distributions with same dimension d, and thus work with the Euclidean space $\mathbb{R}^{d\times d}$ of square matrices. It is straightforward, however, to extend all results for a different number of bins m, n by using rectangular matrices in $\mathbb{R}^{m\times n}$ instead. We denote the null matrix of $\mathbb{R}^{d\times d}$ by $\mathbf{0}$, and the matrix full of ones by $\mathbf{1}$. The Frobenius inner product between two matrices $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}^{d\times d}$ is defined by:

$$\langle \boldsymbol{\pi}, \boldsymbol{\xi} \rangle = \sum_{i=1}^{d}\sum_{j=1}^{d} \pi_{ij}\xi_{ij} \ . \tag{1}$$

When the intended meaning is clear from the context, we also write $\mathbf{0}$ for the null vector of $\mathbb{R}^d$, and $\mathbf{1}$ for the vector full of ones. The notation $\cdot^\top$ represents the transposition operator for matrices or vectors. The probability simplex of $\mathbb{R}^d$ is defined as follows:

$$\Sigma_d = \{\mathbf{p} \in \mathbb{R}^d_+ \colon \mathbf{p}^\top \mathbf{1} = 1\} \ . \tag{2}$$

The operator $\mathrm{diag}(\mathbf{v})$ transforms a vector $\mathbf{v} \in \mathbb{R}^d$ into a diagonal matrix $\boldsymbol{\pi} \in \mathbb{R}^{d\times d}$ such that $\pi_{ii} = v_i$, for all $1 \leq i \leq d$. The operator $\mathrm{vec}(\boldsymbol{\pi})$ transforms a matrix $\boldsymbol{\pi} \in \mathbb{R}^{d\times d}$ into a vector $\mathbf{x} \in \mathbb{R}^{d^2}$ such that $x_{i+(j-1)d} = \pi_{ij}$, for all $1 \leq i, j \leq d$. The operator $\mathrm{sgn}(x)$ for $x \in \mathbb{R}$ returns $-1, 0, +1$, if x is negative, null, positive, respectively. Functions of a real variable, such as the absolute value, sign, exponential or power functions, are considered element-wise when applied to matrices. The max operator and inequalities between matrices should also be interpreted element-wise. Matrix divisions are similarly considered element-wise, whereas element-wise matrix multiplications, also known as Hadamard or Schur products, are denoted by $\odot$ to remove any ambiguity with standard matrix multiplications. Lastly, addition or subtraction of a scalar and a matrix should be understood element-wise by replicating the scalar.

1.2 Background and Related Work

Given two probability vectors $\mathbf{p}, \mathbf{q} \in \Sigma_d$, and a cost matrix $\boldsymbol{\gamma} \in \mathbb{R}^{d\times d}_+$ whose coefficients γ_{ij} represent the cost of moving the mass from bin p_i to q_j, the total cost of a given transport plan, or coupling, $\boldsymbol{\pi} \in \Pi(\mathbf{p}, \mathbf{q})$ can be quantified as $\langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle$. An optimal cost is then obtained

by solving a linear program:

$$d_{\boldsymbol{\gamma}}(\mathbf{p}, \mathbf{q}) = \min_{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle \ , \tag{3}$$

with the transport polytope of $\mathbf{p}$ and $\mathbf{q}$, also known as the polytope of couplings between $\mathbf{p}$ and $\mathbf{q}$, defined as the following polyhedron:

$$\Pi(\mathbf{p}, \mathbf{q}) = \{\boldsymbol{\pi} \in \mathbb{R}^{d\times d}_+ \colon \boldsymbol{\pi}\mathbf{1} = \mathbf{p}, \boldsymbol{\pi}^\top \mathbf{1} = \mathbf{q}\} \ . \tag{4}$$

The EMD associated to the cost matrix $\boldsymbol{\gamma}$ is given by $d_{\boldsymbol{\gamma}}$ and is a true distance metric on the probability simplex Σ_d whenever $\boldsymbol{\gamma}$ is itself a distance matrix. In general, the optimal plans, or earth mover's plans, have at most $2d-1$ nonzero entries, and consist either of a single vertex or of a whole facet of the transport polytope. One of the earth mover's plans can be obtained with the network simplex (Ahuja et al., 1993) among other approaches. For a general cost matrix $\boldsymbol{\gamma}$, the complexity of solving an OT problem scales at least in $O(d^3 \log d)$ for the best algorithms currently proposed, including the network simplex, and turns out to be super-cubic in practice as well.

Cuturi (2013) proposed a new family of OT distances, called Sinkhorn distances, from the perspective of maximum entropy. The idea is to smooth the original problem with a strictly convex regularization via the Boltzmann-Shannon entropy. The primal problem involves the entropic regularization as an additional constraint:

$$d'_{\boldsymbol{\gamma},\alpha}(\mathbf{p}, \mathbf{q}) = \min_{\boldsymbol{\pi} \in \Pi_\alpha(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle \ , \tag{5}$$

with the regularized transport polytope defined as follows:

$$\Pi_\alpha(\mathbf{p}, \mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p}, \mathbf{q}) \colon E(\boldsymbol{\pi}) \leq E(\mathbf{p}\mathbf{q}^\top) + \alpha\} \ , \tag{6}$$

where $\alpha \geq 0$ is a regularization term and E is minus the Boltzmann-Shannon entropy as defined in (28). It is also straightforward to prove that we have:

$$\Pi_\alpha(\mathbf{p}, \mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p}, \mathbf{q}) \colon K(\boldsymbol{\pi} \| \mathbf{1}) \leq K(\mathbf{p}\mathbf{q}^\top \| \mathbf{1}) + \alpha\} \ , \tag{7}$$

$$\Pi_\alpha(\mathbf{p}, \mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p}, \mathbf{q}) \colon K(\boldsymbol{\pi} \| \mathbf{p}\mathbf{q}^\top) \leq \alpha\} \ . \tag{8}$$

where K is the Kullback-Leibler divergence as defined in (27). This enforces the solution to have sufficient entropy, or equivalently small enough mutual information, by constraining it to the Kullback-Leibler ball of radius $K(\mathbf{p}\mathbf{q}^\top \| \mathbf{1}) + \alpha$, respectively α, and center the matrix $\mathbf{1} \in \mathbb{R}^{d\times d}_{++}$, respectively the transport plan $\mathbf{p}\mathbf{q}^\top \in \mathbb{R}^{d\times d}_{++}$, which have maximum entropy. The dual problem exploits a Lagrange multiplier to relax the entropic regularization as a penalty:

$$d_{\boldsymbol{\gamma},\lambda}(\mathbf{p}, \mathbf{q}) = \langle \boldsymbol{\pi}^\star_\lambda, \boldsymbol{\gamma} \rangle \ , \tag{9}$$

with the regularized optimal plan $\boldsymbol{\pi}^\star_\lambda$ defined as follows:

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma} \rangle + \lambda E(\boldsymbol{\pi}) \ , \tag{10}$$

where $\lambda > 0$ is a regularization term. The problem can then be solved empirically in quadratic complexity with linear convergence using the Sinkhorn-Knopp algorithm (Sinkhorn

and Knopp, 1967) based on iterative matrix scaling, where rows and columns are rescaled in turn so that they respectively sum up to $\mathbf{p}$ and $\mathbf{q}$ until convergence. Finally, it is easy to prove that we have:

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda K(\boldsymbol{\pi}\|\mathbf{1}) \ , \tag{11}$$

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda K(\boldsymbol{\pi}\|\mathbf{p}\mathbf{q}^\top) \ . \tag{12}$$

This again shows that the regularization enforces the solution to have sufficient entropy, or equivalently small enough mutual information, by shrinking it toward the matrix $\mathbf{1}$ and the joint distribution $\mathbf{p}\mathbf{q}^\top$ which have maximum entropy.

Benamou et al. (2015) revisited the entropic regularization in a geometrical framework with iterative information projections. They showed that computing a Sinkhorn distance in dual form actually amounts to the minimization of a Kullback-Leibler divergence:

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} K(\boldsymbol{\pi}\|\exp(-\boldsymbol{\gamma}/\lambda)) \ . \tag{13}$$

Precisely, this amounts to computing the Kullback-Leibler projection of $\exp(-\boldsymbol{\gamma}/\lambda) \in \mathbb{R}^{d\times d}_{++}$ onto the transport polytope $\Pi(\mathbf{p},\mathbf{q})$. In this context, the Sinkhorn-Knopp algorithm turns out to be a special instance of Bregman projection onto the intersection of convex sets via alternate projections. Specifically, we see $\Pi(\mathbf{p},\mathbf{q})$ as the intersection of the non-negative orthant with two affine subspaces containing all matrices with rows and columns summing to $\mathbf{p}$ and $\mathbf{q}$ respectively, and we alternate projection on these two subspaces according to the Kullback-Leibler divergence until convergence.

Kurras (2015) further studied this equivalence in the wider context of iterative proportional fitting. He notably showed that the Sinkhorn-Knopp and alternate Bregman projections can be extended to account for infinite entries in the cost matrix $\boldsymbol{\gamma}$, and thus null entries in the regularized optimal plan. Hence, it is possible to develop a sparse version of the entropic regularization to OT problems. This becomes interesting to store the $d \times d$ matrix variables and perform the required computations when the data dimension gets large.

Dhillon and Tropp (2007) had already enlightened such an equivalence in the field of matrix analysis. They actually considered the estimation of contingency tables with fixed marginals as a matrix nearness problem based on the Kullback-Leibler divergence. In more detail, they use a rough estimate $\boldsymbol{\xi} \in \mathbb{R}^{d\times d}_{++}$ to produce a contingency table $\boldsymbol{\pi}^\star$ that has fixed marginals $\mathbf{p},\mathbf{q}$ by Kullback-Leibler projection of $\boldsymbol{\xi}$ onto $\Pi(\mathbf{p},\mathbf{q})$:

$$\boldsymbol{\pi}^\star = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} K(\boldsymbol{\pi}\|\boldsymbol{\xi}) \ . \tag{14}$$

They showed that alternate Bregman projections specialize to the Sinkhorn-Knopp algorithm in this context. However, no relationship to OT problems was highlighted.

1.3 Contributions and Organization

Our main contribution is to formulate a unified framework for discrete regularized optimal transport (ROT) by considering a large class of smooth convex regularizers. We call the

underlying distance the rot mover's distance (RMD) and show that a given ROT problem actually amounts to the minimization of an associated Bregman divergence. This allows the derivation of two schemes that we call the alternate scaling algorithm (ASA) and the non-negative alternate scaling algorithm (NASA), to compute efficiently the regularized optimal plans depending on whether the domain of the regularizer lies within the non-negative orthant or not. These schemes are based on the general form of alternate projections for Bregman divergences. They also exploit the Newton-Raphson method to approximate the projections for separable divergences. The separable case is further enhanced with a sparse extension to deal with high data dimensions. We also instantiate our two generic schemes with widely-used regularizers and statistical divergences.

The proposed framework naturally extends the Sinkhorn-Knopp algorithm for the regularization based on the Boltzmann-Shannon entropy (Cuturi, 2013), or equivalently the minimization of a Kullback-Leibler divergence (Benamou et al., 2015), and their sparse version (Kurras, 2015), which turn out to be special instances of ROT problems. It also relates to matrix nearness problems via minimization of Bregman divergences, and it is straightforward to construct more general estimators for contingency tables with fixed marginals than the classical estimator based on the Kullback-Leibler divergence (Dhillon and Tropp, 2007). Lastly, it brings some new insights between transportation theory (Villani, 2009) and information geometry (Amari and Nagaoka, 2000), where Bregman divergences are known to possess a dually flat structure with a generalized Pythagorean theorem in relation to information projections.

The remainder of this paper is organized as follows. In Section 2, we introduce some necessary preliminaries. In Section 3, we present our theoretical results for a unified framework of ROT problems. We then derive the algorithmic methods for solving ROT problems in Section 4. We also discuss the inherent specificities of ROT problems for classical regularizers and associated divergences in Section 5. In Section 6, we provide experiments to illustrate our methods on synthetic data and real-world audio data in a classification problem. Finally, in Section 7, we draw some conclusions and perspectives for future work.

2. Theoretical Preliminaries

In this section, we introduce the required preliminaries to our framework. We begin with elements of convex analysis (Section 2.1) and of Bregman geometry (Section 2.2). We proceed with theoretical results for convergence of alternate Bregman projections (Section 2.3) and of the Newton-Raphson method (Section 2.4).

2.1 Convex Analysis

Let $\mathcal{E}$ be a Euclidean space with inner product $\langle\cdot,\cdot\rangle$ and induced norm $\|\cdot\|$. The boundary, interior and relative interior of a subset $\mathcal{X} \subseteq \mathcal{E}$ are respectively denoted by $\mathrm{bd}(\mathcal{X})$, $\mathrm{int}(\mathcal{X})$, and $\mathrm{ri}(\mathcal{X})$, where we recall that for a convex set $\mathcal{C}$, we have:

$$\mathrm{ri}(\mathcal{C}) = \{\mathbf{x}\in\mathcal{C} \colon \forall \mathbf{y}\in\mathcal{C}, \exists\lambda>1, \lambda\mathbf{x}+(1-\lambda)\mathbf{y}\in\mathcal{C}\} \ . \tag{15}$$

In convex analysis, scalar functions are defined over the whole space $\mathcal{E}$ and take values in the extended real number line $\mathbb{R}\cup\{-\infty,+\infty\}$. The effective domain, or simply domain,

of a function f is then defined as the set:

$$\operatorname{dom} f = \{\mathbf{x} \in \mathcal{E} \colon f(\mathbf{x}) < +\infty\} \ . \tag{16}$$

A convex function f is proper if $f(\mathbf{x}) < +\infty$ for at least one $\mathbf{x} \in \mathcal{E}$ and $f(\mathbf{x}) > -\infty$ for all $\mathbf{x} \in \mathcal{E}$, and it is closed if its lower level sets $\{\mathbf{x} \in \mathcal{E} \colon f(\mathbf{x}) \leq \alpha\}$ are closed for all $\alpha \in \mathbb{R}$. If $\operatorname{dom} f$ is closed, then f is closed, and a proper convex function is closed if and only if it is lower semi-continuous. Moreover, a closed function f is continuous relative to any simplex, polytope or polyhedral subset in $\operatorname{dom} f$. It is also well-known that a convex function f is always continuous in the relative interior $\operatorname{ri}(\operatorname{dom} f)$ of its domain.

A function f is essentially smooth if it is differentiable on $\operatorname{int}(\operatorname{dom} f) \neq \emptyset$ and verifies $\lim_{k \to +\infty} \|\nabla f(\mathbf{x}_k)\| = +\infty$ for any sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ from $\operatorname{int}(\operatorname{dom} f)$ that converges to a point $\mathbf{x} \in \operatorname{bd}(\operatorname{dom} f)$. A function f is of Legendre type if it is a closed proper convex function that is also essentially smooth and strictly convex on $\operatorname{int}(\operatorname{dom} f)$.

The Fenchel conjugate $f^\star$ of a function f is defined for all $\mathbf{y} \in \mathcal{E}$ as follows:

$$f^\star(\mathbf{y}) = \sup_{\mathbf{x} \in \operatorname{int}(\operatorname{dom} f)} \langle \mathbf{x}, \mathbf{y} \rangle - f(\mathbf{x}) \ . \tag{17}$$

The Fenchel conjugate $f^\star$ is always a closed convex function. Moreover, if f is a closed convex function, then $(f^\star)^\star = f$, and f is of Legendre type if and only if $f^\star$ is of Legendre type. In this latter case, the gradient mapping ∇f is a homeomorphism between $\operatorname{int}(\operatorname{dom} f)$ and $\operatorname{int}(\operatorname{dom} f^\star)$, with inverse mapping $(\nabla f)^{-1} = \nabla f^\star$, which guarantees the existence of dual coordinate systems $\mathbf{x}(\mathbf{y}) = \nabla f^\star(\mathbf{y})$ and $\mathbf{y}(\mathbf{x}) = \nabla f(\mathbf{x})$ on $\operatorname{int}(\operatorname{dom} f)$ and $\operatorname{int}(\operatorname{dom} f^\star)$.

Finally, we say that a function f is cofinite if it verifies:

$$\lim_{\lambda \to +\infty} f(\lambda \mathbf{x})/\lambda = +\infty \ , \tag{18}$$

for all nonzero $\mathbf{x} \in \mathcal{E}$. Intuitively, it means that f grows super-linearly in every direction. In particular, a closed proper convex function is cofinite if and only if $\operatorname{dom} f^\star = \mathcal{E}$.

2.2 Bregman Geometry

Let ϕ be a convex function on $\mathcal{E}$ that is differentiable on $\operatorname{int}(\operatorname{dom} \phi) \neq \emptyset$. The Bregman divergence generated by ϕ is defined as follows:

$$B_\phi(\mathbf{x} \| \mathbf{y}) = \phi(\mathbf{x}) - \phi(\mathbf{y}) - \langle \mathbf{x} - \mathbf{y}, \nabla \phi(\mathbf{y}) \rangle \ , \tag{19}$$

for all $\mathbf{x} \in \operatorname{dom} \phi$ and $\mathbf{y} \in \operatorname{int}(\operatorname{dom} \phi)$. We have $B_\phi(\mathbf{x} \| \mathbf{y}) \geq 0$ for any $\mathbf{x} \in \operatorname{dom} \phi$ and $\mathbf{y} \in \operatorname{int}(\operatorname{dom} \phi)$. If in addition ϕ is strictly convex on $\operatorname{int}(\operatorname{dom} \phi)$, then $B_\phi(\mathbf{x} \| \mathbf{y}) = 0$ if and only if $\mathbf{x} = \mathbf{y}$. Bregman divergences are also always convex in the first argument, and are invariant under adding an arbitrary affine term to their generator.

Bregman divergences are not symmetric and do not verify the triangle inequality in general, and thus are not necessarily distances in the strict sense. However, they still enjoy some nice geometrical properties that somehow generalize the Euclidean geometry. In particular, they verify a four-point identity similar to a parallelogram law:

$$B_\phi(\mathbf{x} \| \mathbf{y}) + B_\phi(\mathbf{x}' \| \mathbf{y}') = B_\phi(\mathbf{x}' \| \mathbf{y}) + B_\phi(\mathbf{x} \| \mathbf{y}') - \langle \mathbf{x} - \mathbf{x}', \nabla \phi(\mathbf{y}) - \nabla \phi(\mathbf{y}') \rangle \ , \tag{20}$$

for all $\mathbf{x}, \mathbf{x}' \in \operatorname{dom} \phi$ and $\mathbf{y}, \mathbf{y}' \in \operatorname{int}(\operatorname{dom} \phi)$. A special instance of this relation gives rise to a three-point property similar to a triangle law of cosines:

$$B_\phi(\mathbf{x} \| \mathbf{y}) = B_\phi(\mathbf{x} \| \mathbf{y}') + B_\phi(\mathbf{y}' \| \mathbf{y}) - \langle \mathbf{x} - \mathbf{y}', \nabla \phi(\mathbf{y}) - \nabla \phi(\mathbf{y}') \rangle \ , \tag{21}$$

for all $\mathbf{x} \in \operatorname{dom} \phi$ and $\mathbf{y}, \mathbf{y}' \in \operatorname{int}(\operatorname{dom} \phi)$.

Suppose now that ϕ is of Legendre type, and let $\mathcal{C} \subseteq \mathcal{E}$ be a closed convex set such that $\mathcal{C} \cap \operatorname{int}(\operatorname{dom} \phi) \neq \emptyset$. Then, for any point $\mathbf{y} \in \operatorname{int}(\operatorname{dom} \phi)$, the following problem:

$$P_{\mathcal{C}}(\mathbf{y}) = \operatorname*{argmin}_{\mathbf{x} \in \mathcal{C}} B_\phi(\mathbf{x} \| \mathbf{y}) \ , \tag{22}$$

has a unique solution, then called the Bregman projection of $\mathbf{y}$ onto $\mathcal{C}$. This solution actually belongs to $\mathcal{C} \cap \operatorname{int}(\operatorname{dom} \phi)$, and is also characterized as the unique point $\mathbf{y}' \in \mathcal{C} \cap \operatorname{int}(\operatorname{dom} \phi)$ that verifies the variational relation:

$$\langle \mathbf{x} - \mathbf{y}', \nabla \phi(\mathbf{y}) - \nabla \phi(\mathbf{y}') \rangle \leq 0 \ , \tag{23}$$

for all $\mathbf{x} \in \mathcal{C} \cap \operatorname{dom} \phi$. This characterization is equivalent to a well-known generalized Pythagorean theorem for Bregman divergences, which states that the Bregman projection of $\mathbf{y}$ onto $\mathcal{C}$ is the unique point $\mathbf{y}' \in \mathcal{C} \cap \operatorname{int}(\operatorname{dom} \phi)$ that verifies the following inequality:

$$B_\phi(\mathbf{x} \| \mathbf{y}) \geq B_\phi(\mathbf{x} \| \mathbf{y}') + B_\phi(\mathbf{y}' \| \mathbf{y}) \ , \tag{24}$$

for all $\mathbf{x} \in \mathcal{C} \cap \operatorname{dom} \phi$. When $\mathcal{C}$ is further an affine subspace, or more generally when the Bregman projection further belongs to $\operatorname{ri}(\mathcal{C})$, the scalar product actually vanishes:

$$\langle \mathbf{x} - \mathbf{y}', \nabla \phi(\mathbf{y}) - \nabla \phi(\mathbf{y}') \rangle = 0 \ , \tag{25}$$

leading to an equality in the generalized Pythagorean theorem:

$$B_\phi(\mathbf{x} \| \mathbf{y}) = B_\phi(\mathbf{x} \| \mathbf{y}') + B_\phi(\mathbf{y}' \| \mathbf{y}) \ . \tag{26}$$

A famous example of Bregman divergence is the Kullback-Leibler divergence, defined for matrices $\boldsymbol{\pi} \in \mathbb{R}_+^{d \times d}$ and $\boldsymbol{\xi} \in \mathbb{R}_{++}^{d \times d}$ as follows:

$$K(\boldsymbol{\pi} \| \boldsymbol{\xi}) = \sum_{i=1}^{d} \sum_{j=1}^{d} \left(\pi_{ij} \log \left(\frac{\pi_{ij}}{\xi_{ij}} \right) - \pi_{ij} + \xi_{ij} \right) \ . \tag{27}$$

This divergence is generated by a function of Legendre type for $\boldsymbol{\pi} \in \mathbb{R}_+^{d \times d}$ given by minus the Boltzmann-Shannon entropy:

$$E(\boldsymbol{\pi}) = K(\boldsymbol{\pi} \| \mathbf{1}) = \sum_{i=1}^{d} \sum_{j=1}^{d} \left(\pi_{ij} \log(\pi_{ij}) - \pi_{ij} + 1 \right) \ , \tag{28}$$

with the convention $0 \log(0) = 0$. Another well-known example is the Itakura-Saito divergence, defined for matrices $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}_{++}^{d \times d}$ as follows:

$$I(\boldsymbol{\pi} \| \boldsymbol{\xi}) = \sum_{i=1}^{d} \sum_{j=1}^{d} \left(\frac{\pi_{ij}}{\xi_{ij}} - \log \left(\frac{\pi_{ij}}{\xi_{ij}} \right) - 1 \right) \ . \tag{29}$$

This divergence is generated by a function of Legendre type for $\boldsymbol{\pi} \in \mathbb{R}^{d\times d}_{++}$ given by minus the Burg entropy:

$$F(\boldsymbol{\pi}) = \sum_{i=1}^{d}\sum_{j=1}^{d}(\pi_{ij} - \log \pi_{ij} - 1) \ . \tag{30}$$

On the one hand, these examples belong to a particular type of so-called separable Bregman divergences between matrices on $\mathbb{R}^{d\times d}$, that can be seen as the aggregation of element-wise Bregman divergences between scalars on $\mathbb{R}$:

$$B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}) = \sum_{i=1}^{d}\sum_{j=1}^{d} B_{\phi_{ij}}(\pi_{ij}\|\xi_{ij}) \ , \tag{31}$$

$$\phi(\boldsymbol{\pi}) = \sum_{i=1}^{d}\sum_{j=1}^{d} \phi_{ij}(\pi_{ij}) \ . \tag{32}$$

Often, all element-wise generators ϕ_{ij} are chosen equal, and are thus simply written as ϕ with a slight abuse of notation. Other examples of such divergences are discussed in Section 5, and include the logistic loss function generated by minus the Fermi-Dirac entropy, or the squared Euclidean distance generated by the Euclidean norm.

On the other hand, a classical example of non-separable Bregman divergence is half the squared Mahalanobis distance, defined for matrices $\boldsymbol{\pi}, \boldsymbol{\xi} \in \mathbb{R}^{d\times d}$ as follows:

$$M(\boldsymbol{\pi}\|\boldsymbol{\xi}) = \frac{1}{2}\mathrm{vec}(\boldsymbol{\pi} - \boldsymbol{\xi})^\top \mathbf{P}\, \mathrm{vec}(\boldsymbol{\pi} - \boldsymbol{\xi}) \ , \tag{33}$$

for a positive-definite matrix $\mathbf{P} \in \mathbb{R}^{d^2\times d^2}$. This divergence is generated by a function of Legendre type for $\boldsymbol{\pi} \in \mathbb{R}^{d\times d}$ given by a quadratic form:

$$Q(\boldsymbol{\pi}) = \frac{1}{2}\mathrm{vec}(\boldsymbol{\pi})^\top \mathbf{P}\, \mathrm{vec}(\boldsymbol{\pi}) \ . \tag{34}$$

This example is also discussed in Section 5.

2.3 Alternate Bregman Projections

Let ϕ be a function of Legendre type with Fenchel conjugate $\phi^\star = \psi$. In general, computing Bregman projections onto an arbitrary closed convex set $\mathcal{C} \subseteq \mathcal{E}$ such that $\mathcal{C} \cap \mathrm{int}(\mathrm{dom}\,\phi) \neq \emptyset$ is nontrivial. Sometimes, it is possible to decompose $\mathcal{C}$ into the intersection of finitely many closed convex sets:

$$\mathcal{C} = \bigcap_{l=1}^{s} \mathcal{C}_l \ , \tag{35}$$

where the individual Bregman projections onto the respective sets $\mathcal{C}_1, \ldots, \mathcal{C}_s$ are easier to compute. It is then possible to obtain the Bregman projection onto $\mathcal{C}$ by alternate projections onto $\mathcal{C}_1, \ldots, \mathcal{C}_s$ according to Dykstra's algorithm.

In more detail, let $\sigma\colon \mathbb{N} \to \{1, \ldots, s\}$ be a control mapping that determines the sequence of subsets onto which we project. For a given point $\mathbf{x}_0 \in \mathcal{C} \cap \mathrm{int}(\mathrm{dom}\,\phi)$, the Bregman

projection $P_{\mathcal{C}}(\mathbf{x}_0)$ of $\mathbf{x}_0$ onto $\mathcal{C}$ can be approximated with Dykstra's algorithm by iterating the following updates:

$$\mathbf{x}_{k+1} \leftarrow P_{\mathcal{C}_{\sigma(k)}}(\nabla\psi(\nabla\phi(\mathbf{x}_k) + \mathbf{y}^{\sigma(k)})) \ , \tag{36}$$

where the correction terms $\mathbf{y}^1, \ldots, \mathbf{y}^s$ for the respective subsets are initialized with the null element of $\mathcal{E}$, and are updated after projection as follows:

$$\mathbf{y}^{\sigma(k)} \leftarrow \mathbf{y}^{\sigma(k)} + \nabla\phi(\mathbf{x}_k) - \nabla\phi(\mathbf{x}_{k+1}) \ . \tag{37}$$

Under some technical conditions, the sequence of updates $(\mathbf{x}_k)_{k\in\mathbb{N}}$ then converges in norm to $P_{\mathcal{C}}(\mathbf{x}_0)$ with a linear rate. Several sets of such conditions have been studied, notably by Tseng (1993), Bauschke and Lewis (2000), Dhillon and Tropp (2007).

We here use the conditions proposed by Dhillon and Tropp (2007), which reveal to be the less restrictive ones in our framework. Specifically, the convergence of Dykstra's algorithm is guaranteed as soon as the function ϕ is cofinite, the constraint qualification $\mathrm{ri}(\mathcal{C}_1) \cap \cdots \cap \mathrm{ri}(\mathcal{C}_s) \cap \mathrm{int}(\mathrm{dom}\,\phi) \neq \emptyset$ holds, and the control mapping σ is essentially cyclic, that is, there exists a number $t \in \mathbb{N}$ such that σ takes each output value at least once during any t consecutive input values. If a given $\mathcal{C}_l$ is a polyhedral set, then the relative interior can be dropped from the constraint qualification. Hence, when all subsets $\mathcal{C}_l$ are polyhedral, the constraint qualification simply reduces to $\mathcal{C} \cap \mathrm{int}(\mathrm{dom}\,\phi) \neq \emptyset$, which is already enforced for the definition of Bregman projections.

Finally, if all subsets $\mathcal{C}_l$ are further affine, then we can relax other assumptions. Notably, we do not require ϕ to be cofinite (18), or equivalently $\mathrm{dom}\,\psi = \mathcal{E}$, but only $\mathrm{dom}\,\psi$ to be open. The control mapping need not be essentially cyclic anymore, as long as it takes each output value an infinite number of times. More importantly, we can completely drop the correction terms from the updates, leading to a simpler technique known as projections onto convex sets (POCS):

$$\mathbf{x}_{k+1} \leftarrow P_{\mathcal{C}_{\sigma(k)}}(\mathbf{x}_k) \ . \tag{38}$$

2.4 Newton-Raphson Method

Let f be a continuously differentiable scalar function on an open interval $I \subseteq \mathbb{R}$. Assume f is increasing on a non-empty closed interval $[x^-, x^+] \subset I$, and write $y^- = f(x^-)$ and $y^+ = f(x^+)$. Then, for any $y \in [y^-, y^+]$, the equation $f(x) = y$ has at least one solution $x^\star \in [x^-, x^+]$. Such a solution can be approximated by iterative updates according to the Newton-Raphson method:

$$x \leftarrow \max\left\{x^-, \min\left\{x^+, x - \frac{f(x) - y}{f'(x)}\right\}\right\} \ , \tag{39}$$

where the fraction takes infinite values when $f'(x) = 0$ and $f(x) \neq y$, and a null value by convention when $f'(x) = 0$ and $f(x) = y$. It is well-known that the Newton-Raphson method converges to a solution $x^\star$ as soon as x is initialized sufficiently close to $x^\star$. Convergence is then quadratic provided that $f'(x^\star) \neq 0$. However, this local convergence has little importance in practice because it is hard to quantify the required proximity to the solution.

(A) Affine constraints	(B) Polyhedral constraints
(A1) ϕ is of Legendre type.	(B1) ϕ is of Legendre type.
(A2) $(0,1)^{d\times d} \subseteq \operatorname{dom}\phi$.	(B2) $(0,1)^{d\times d} \subseteq \operatorname{dom}\phi$.
(A3) $\operatorname{dom}\phi \subseteq \mathbb{R}_+^{d\times d}$.	(B3) $\operatorname{dom}\phi \nsubseteq \mathbb{R}_+^{d\times d}$.
(A4) $\operatorname{dom}\psi$ is open.	(B4) $\operatorname{dom}\psi = \mathbb{R}^{d\times d}$.
(A5) $\mathbb{R}_-^{d\times d} \subset \operatorname{dom}\psi$.	

Table 1: Set of assumptions for the considered regularizers ϕ.

Thorlund-Petersen (2004) elucidated results on global convergence of the Newton-Raphson method. He proved a necessary and sufficient condition of convergence for an arbitrary value $y \in [y^-, y^+]$ and from any starting point $x \in [x^-, x^+]$. This condition is that for any $a, b \in [x^-, x^+]$, $f(b) > f(a)$ implies:

$$f'(a) + f'(b) > \frac{f(b) - f(a)}{b - a} \ . \tag{40}$$

In particular, a sufficient condition is that the underlying function f is an increasing convex or increasing concave function on $[x^-, x^+]$, or can be decomposed as the sum of such functions. In addition, if f satisfies the necessary and sufficient condition and is strictly increasing with $f'(x) > 0$ for all $x \in [x^-, x^+]$, then initializing with a boundary point $x^- \neq x^\star$ or $x^+ \neq x^\star$ ensures that the entire sequence of updates is interior to (x^-, x^+), so that we can actually drop the min and max truncation operators in the updates:

$$x \leftarrow x - \frac{f(x) - y}{f'(x)} \ . \tag{41}$$

3. Mathematical Formulation

In this section, we develop a unified framework to define ROT problems. We start by drawing some technical assumptions for our generalized framework to hold (Section 3.1). We then formulate primal ROT problems and study their properties (Section 3.2). We also formulate dual ROT problems and discuss their properties in relation to primal ones (Section 3.3). Finally, we provide some geometrical insights to summarize our developments in the light of information geometry (Section 3.4).

3.1 Technical Assumptions

Some mild technical assumptions are required on the convex regularizer ϕ and its Fenchel conjugate $\psi = \phi^*$ for the proposed framework to hold. Some assumptions relate to required conditions for the definition of Bregman projections and convergence of the algorithms, while others are more specific to ROT problems. In our framework, we also need to distinguish between two situations where the underlying closed convex set can be described as the intersection of either affine subspaces or polyhedral subsets. The two sets of assumptions (A) and (B) are summarized in Table 1.

For the first assumptions (A1) and (B1), we recall that a closed proper convex function is of Legendre type if and only if it is essentially smooth and strictly convex on the interior

of its domain (Section 2.1). This is required for the definition of Bregman projections (Section 2.2). In addition, it guarantees the existence of dual coordinate systems on $\operatorname{int}(\operatorname{dom}\phi)$ and $\operatorname{int}(\operatorname{dom}\psi)$ via the homeomorphism $\nabla\phi = \nabla\psi^{-1}$:

$$\boldsymbol{\pi}(\boldsymbol{\theta}) = \nabla\psi(\boldsymbol{\theta}) \ , \tag{42}$$

$$\boldsymbol{\theta}(\boldsymbol{\pi}) = \nabla\phi(\boldsymbol{\pi}) \ . \tag{43}$$

With a slight abuse of notation, we omit the reparameterization to simply denote corresponding primal and dual parameters by $\boldsymbol{\pi}$ and $\boldsymbol{\theta}$.

The second assumptions (A2) and (B2) imply that $\operatorname{ri}(\Pi(\mathbf{p}, \mathbf{q})) \subset \operatorname{dom}\phi$ and ensure the constraint qualification $\Pi(\mathbf{p}, \mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi) \neq \emptyset$ for Bregman projection onto the transport polytope, independently of the input distributions $\mathbf{p}, \mathbf{q}$ as long as they do not have null or unit entries. We assume hereafter that this implicitly holds, and discuss in the practical considerations (Section 4.6) how our methods actually generalize to deal explicitly with null or unit entries in the input distributions.

The third assumptions (A3) and (B3) separate between two cases depending on whether $\operatorname{dom}\phi$ lies within the non-negative orthant or not for the alternate Bregman projections (Section 2.3). In the former case, non-negativity is already ensured by the domain of the regularizer, so that the underlying closed convex set is made of two affine subspaces for the row and column sum constraints, and the POCS method can be considered. The fourth assumption (A4) thus requires that $\operatorname{dom}\psi$ be open for convergence of this algorithm. In the latter case, there is one additional polyhedral subset for the non-negative constraints and Dykstra's algorithm should be used. The fourth assumption (B4) hence further requires that $\operatorname{dom}\psi = \mathbb{R}^{d\times d}$, or equivalently that ϕ be cofinite (18), for convergence. In both cases, we remark that we necessarily have $\operatorname{dom}\psi = \operatorname{dom}\nabla\psi$.

The fifth assumption (A5) in the affine constraints ensures that $-\boldsymbol{\gamma}/\lambda$ belongs to $\operatorname{dom}\nabla\psi$ for definition of ROT problems, independently of the non-negative cost matrix $\boldsymbol{\gamma}$ and positive regularization term λ. Notice that this is already guaranteed by the fourth assumption in the polyhedral constraints. We also show in the sparse extension (Section 4.5) how to deal with infinite entries in the cost matrix $\boldsymbol{\gamma}$ for separable regularizers, so as to enforce null entries in the regularized optimal plan.

On the one hand, some common regularizers under assumptions (A) are the Boltzmann-Shannon entropy associated to the Kullback-Leibler divergence, the Burg entropy associated to the Itakura-Saito divergence, and the Fermi-Dirac entropy associated to the logistic loss function. To solve the underlying ROT problems, we employ our method called ASA based on the POCS technique, where alternate Bregman projections onto the two affine subspaces for the row and column sum constraints are considered (Section 4.3). On the other hand, examples under assumptions (B) include the Euclidean norm associated to the Euclidean distance, and the quadratic form associated to the Mahalanobis distance. For these ROT problems, we use our second method called NASA based on Dykstra's algorithm, where correction terms and a further Bregman projection onto the polyhedral non-negative orthant are needed (Section 4.4).

3.2 Primal Problem

We start our primal formulation with the following lemmas and definition for the RMD.

Lemma 1 *The regularizer ϕ attains its global minimum uniquely at $\boldsymbol{\xi}' = \nabla\psi(\mathbf{0})$.*

Proof Using the assumptions (A4) and (A5), respectively (B4), we have that $\mathbf{0} \in \operatorname{dom}\psi = \operatorname{int}(\operatorname{dom}\psi)$. Thus, there exists a unique $\boldsymbol{\xi}' \in \operatorname{int}(\operatorname{dom}\phi)$ such that $\nabla\phi(\boldsymbol{\xi}') = \mathbf{0}$, or equivalently $\boldsymbol{\xi}' = \nabla\psi(\mathbf{0})$, via the homeomorphism $\nabla\phi = \nabla\psi^{-1}$ ensured by assumption (A1), respectively (B1). Hence, ϕ attains its global minimum uniquely at $\boldsymbol{\xi}'$ by strict convexity on $\operatorname{int}(\operatorname{dom}\phi)$. ■

Lemma 2 *The restriction of the regularizer ϕ to the transport polytope $\Pi(\mathbf{p},\mathbf{q})$ attains its global minimum uniquely at the Bregman projection $\boldsymbol{\pi}'$ of $\boldsymbol{\xi}'$ onto $\Pi(\mathbf{p},\mathbf{q})$.*

Proof Using the assumption (A2), respectively (B2), we have that $\Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi) \neq \emptyset$. Since $\boldsymbol{\xi}' \in \operatorname{int}(\operatorname{dom}\phi)$ and $\Pi(\mathbf{p},\mathbf{q})$ is a closed convex set, the Bregman projection $\boldsymbol{\pi}'$ of $\boldsymbol{\xi}'$ onto $\Pi(\mathbf{p},\mathbf{q})$ according to the function ϕ of Legendre type is well-defined. Moreover, it is characterized by the variational relation (23) as follows:

$$\langle \boldsymbol{\pi} - \boldsymbol{\pi}', \nabla\phi(\boldsymbol{\pi}')\rangle \geq 0 \ , \tag{44}$$

for all $\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{dom}\phi$. We also have $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') > 0$ when $\boldsymbol{\pi} \neq \boldsymbol{\pi}'$ by strict convexity of ϕ on $\operatorname{int}(\operatorname{dom}\phi)$. As a result, we have:

$$\phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\pi}') > \langle \boldsymbol{\pi} - \boldsymbol{\pi}', \nabla\phi(\boldsymbol{\pi}')\rangle \ . \tag{45}$$

Combining the two inequalities, we obtain $\phi(\boldsymbol{\pi}) > \phi(\boldsymbol{\pi}')$ and the restriction of ϕ to $\Pi(\mathbf{p},\mathbf{q})$ attains its global minimum uniquely at $\boldsymbol{\pi}'$. ■

Lemma 3 *The restriction of the cost $\langle\cdot,\boldsymbol{\gamma}\rangle$ to the regularized transport polytope:*

$$\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})\colon \phi(\boldsymbol{\pi}) \leq \phi(\boldsymbol{\pi}') + \alpha\} \ , \tag{46}$$

where $\alpha \geq 0$, attains its global minimum.

Proof The regularized transport polytope is the intersection of the compact set $\Pi(\mathbf{p},\mathbf{q})$ with a lower level set of ϕ which is also closed since ϕ is closed. Hence, $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ is compact and the restriction of $\langle\cdot,\boldsymbol{\gamma}\rangle$ to $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ attains its global minimum by continuity on a compact set. ■

Definition 4 *The primal rot mover's distance is the quantity defined as:*

$$d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q}) = \min_{\boldsymbol{\pi}\in\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})} \langle\boldsymbol{\pi},\boldsymbol{\gamma}\rangle \ . \tag{47}$$

A minimizer $\boldsymbol{\pi}'^{\star}_{\alpha}$ is then called a primal rot mover's plan.

Remark 1 *For the sake of notation, we omit the dependence on $\mathbf{p},\mathbf{q},\boldsymbol{\gamma},\phi$ in the index of primal rot mover's plans $\boldsymbol{\pi}'^{\star}_{\alpha}$.*

The regularization enforces the associated minimizers to have small enough Bregman information $\phi(\boldsymbol{\pi}'^{\star}_{\alpha}) \leq \phi(\boldsymbol{\pi}') + \alpha$ compared to the minimal one $\phi(\boldsymbol{\pi}')$ for transport plans. We also have a geometrical interpretation where the solutions are constrained to a Bregman ball whose center $\boldsymbol{\xi}'$ is the matrix with minimal Bregman information.

Proposition 5 *The regularized transport polytope is the intersection of the transport polytope with the Bregman ball of radius $B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') + \alpha$ and center $\boldsymbol{\xi}'$:*

$$\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})\colon B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') \leq B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') + \alpha\} \ . \tag{48}$$

Proof Expanding the Bregman divergences from their definition (19), we obtain:

$$B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') = \phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\xi}') - \langle \boldsymbol{\pi} - \boldsymbol{\xi}', \nabla\phi(\boldsymbol{\xi}')\rangle \ , \tag{49}$$

$$B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') = \phi(\boldsymbol{\pi}') - \phi(\boldsymbol{\xi}') - \langle \boldsymbol{\pi}' - \boldsymbol{\xi}', \nabla\phi(\boldsymbol{\xi}')\rangle \ . \tag{50}$$

Since $\nabla\phi(\boldsymbol{\xi}') = \mathbf{0}$, the last terms with scalar products vanish, leading to:

$$\phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\pi}') = B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') - B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') \ . \tag{51}$$

Therefore, in the definition (46) of $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$, we have $\phi(\boldsymbol{\pi}) \leq \phi(\boldsymbol{\pi}') + \alpha$ if and only if $\boldsymbol{\pi}$ is in the Bregman ball of radius $B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') + \alpha$ and center $\boldsymbol{\xi}'$. ■

Under some additional conditions, this geometrical interpretation still holds with a Bregman ball whose center $\boldsymbol{\pi}'$ has minimal Bregman information for transport plans.

Proposition 6 *If $\boldsymbol{\pi}' \in \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$, then the regularized transport polytope is the intersection of the transport polytope with the Bregman ball of radius α and center $\boldsymbol{\pi}'$:*

$$\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})\colon B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') \leq \alpha\} \ . \tag{52}$$

Proof Since $\boldsymbol{\pi}' \in \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$, there is equality in the generalized Pythagorean theorem (26):

$$B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') = B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') + B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') \ . \tag{53}$$

The regularized transport polytope as seen from (48) is then the intersection of the transport polytope $\Pi(\mathbf{p},\mathbf{q})$ with the Bregman ball of radius α and center $\boldsymbol{\pi}'$. ■

Remark 2 *The proposition also holds trivially when the global minimum is attained on the transport polytope, that is, when $\boldsymbol{\xi}' = \boldsymbol{\pi}'$.*

Corollary 7 *Under assumptions (A), the regularized transport polytope is the intersection of the transport polytope with the Bregman ball of radius α and center $\boldsymbol{\pi}'$:*

$$\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q})\colon B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') \leq \alpha\} \ . \tag{54}$$

Proof This is a result of $\boldsymbol{\pi}' \in \Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi) = \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$ when $\operatorname{dom}\phi \subseteq \mathbb{R}_+^{d\times d}$. Indeed, we then have $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q})) \subset \Pi(\mathbf{p},\mathbf{q})$ and $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q})) \subset \operatorname{int}(\operatorname{dom}\phi)$, so that $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q})) \subseteq \Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi)$. Conversely, let $\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi)$ so that $\boldsymbol{\pi} \in \mathbb{R}_{++}^{d\times d}$. Then, for a given $\overline{\boldsymbol{\pi}} \in \Pi(\mathbf{p},\mathbf{q})$, let us pose $\boldsymbol{\pi}_\lambda = \lambda\boldsymbol{\pi} + (1-\lambda)\overline{\boldsymbol{\pi}}$ for $\lambda > 1$. We easily have $\boldsymbol{\pi}_\lambda \mathbf{1} = \mathbf{p}$ and $\boldsymbol{\pi}_\lambda^\top \mathbf{1} = \mathbf{q}$. Moreover, since all entries of $\boldsymbol{\pi}$ are positive and that of $\overline{\boldsymbol{\pi}}$ are non-negative, we can always choose a given λ sufficiently close to 1 such that $\boldsymbol{\pi}_\lambda \in \mathbb{R}_+^{d\times d}$. We then have $\boldsymbol{\pi}_\lambda \in \Pi(\mathbf{p},\mathbf{q})$ so that $\boldsymbol{\pi} \in \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$ as characterized by (15), and thus $\Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi) \subseteq \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$. ■

Remark 3 *Under assumptions (B), the Bregman projection $\boldsymbol{\pi}'$ does not necessarily lie within* $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$. *Hence, the geometrical interpretation in terms of a Bregman ball might break down, although the solutions are still constrained to have a small enough Bregman information above that of $\boldsymbol{\pi}'$.*

Although Sinkhorn distances verify the triangular inequality when $\boldsymbol{\gamma}$ is a distance matrix, thanks to specific chain rules and information inequalities for the Bolzmann-Shannon entropy and Kullback-Leibler divergence, it is not necessarily the case for the RMD with other regularizations, even for separable regularizers. Hence, the RMD does not provide a true distance metric on Σ_d in general even if $\boldsymbol{\gamma}$ is a distance matrix. Nonetheless, the RMD is symmetric as soon as ϕ is invariant by transposition, which holds for separable regularizers $\phi_{ij} = \phi$, and $\boldsymbol{\gamma}$ is symmetric. We now study some properties of the RMD that hold for general regularizers.

Property 1 *The primal rot mover's distance $d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q})$ is a decreasing convex and continuous function of α.*

Proof The fact that it is decreasing is a direct consequence of the regularized transport polytope $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ growing with α. The convexity can be proved as follows. Let $\alpha_0, \alpha_1 \geq 0$, and $0 < \lambda < 1$. We pose $\alpha_\lambda = (1-\lambda)\alpha_0 + \lambda\alpha_1 \geq 0$. We also choose arbitrary rot mover's plans $\boldsymbol{\pi}'^\star_{\alpha_0}, \boldsymbol{\pi}'^\star_{\alpha_1}, \boldsymbol{\pi}'^\star_{\alpha_\lambda}$. We finally pose $\boldsymbol{\pi}_\lambda = (1-\lambda)\boldsymbol{\pi}'^\star_{\alpha_0} + \lambda\boldsymbol{\pi}'^\star_{\alpha_1}$. By convexity of ϕ, we have:

$$\begin{aligned}
\phi(\boldsymbol{\pi}_\lambda) &\leq (1-\lambda)\phi(\boldsymbol{\pi}'^\star_{\alpha_0}) + \lambda\phi(\boldsymbol{\pi}'^\star_{\alpha_1}) && (55)\\
&\leq (1-\lambda)(\alpha_0 + \phi(\boldsymbol{\pi}')) + \lambda(\alpha_1 + \phi(\boldsymbol{\pi}')) && (56)\\
&= \alpha_\lambda + \phi(\boldsymbol{\pi}') \ . && (57)
\end{aligned}$$

Hence, $\boldsymbol{\pi}_\lambda \in \Pi_{\alpha_\lambda,\phi}(\mathbf{p},\mathbf{q})$, and by construction we have $\langle \boldsymbol{\pi}'^\star_{\alpha_\lambda}, \boldsymbol{\gamma}\rangle \leq \langle \boldsymbol{\pi}_\lambda, \boldsymbol{\gamma}\rangle$, or equivalently:

$$\langle \boldsymbol{\pi}'^\star_{\alpha_\lambda}, \boldsymbol{\gamma}\rangle \leq (1-\lambda)\langle \boldsymbol{\pi}'^\star_{\alpha_0}, \boldsymbol{\gamma}\rangle + \lambda\langle \boldsymbol{\pi}'^\star_{\alpha_1}, \boldsymbol{\gamma}\rangle \ . \quad (58)$$

The continuity for $\alpha > 0$ is a direct consequence of convexity for $\alpha > 0$, since a convex function is always continuous on the relative interior of its domain. Lastly, the continuity at $\alpha = 0$ can be seen as follows. Let $(\alpha_k)_{k\in\mathbb{N}}$ be a sequence of positive numbers that converges to 0. We choose arbitrary rot mover's plans $(\boldsymbol{\pi}'^\star_{\alpha_k})_{k\in\mathbb{N}}$. By compactness of $\Pi(\mathbf{p},\mathbf{q})$, we can extract a subsequence of rot mover's plans that converges in norm to a point $\boldsymbol{\pi}'^\star \in \Pi(\mathbf{p},\mathbf{q})$.

For the sake of simplicity, we do not relabel this subsequence. By construction, we have $\phi(\boldsymbol{\pi}') \leq \phi(\boldsymbol{\pi}'^\star_{\alpha_k}) \leq \phi(\boldsymbol{\pi}') + \alpha_k$, and $\phi(\boldsymbol{\pi}'^\star_{\alpha_k})$ converges to $\phi(\boldsymbol{\pi}')$. By lower semi-continuity of ϕ, we thus have $\phi(\boldsymbol{\pi}'^\star) \leq \phi(\boldsymbol{\pi}')$. Since the global minimum of ϕ on $\Pi(\mathbf{p},\mathbf{q})$ is attained uniquely at $\boldsymbol{\pi}'$, we must have $\boldsymbol{\pi}'^\star = \boldsymbol{\pi}'$, and the original sequence also converges in norm to $\boldsymbol{\pi}'$. By continuity of the total cost $\langle\cdot,\boldsymbol{\gamma}\rangle$ on $\mathbb{R}^{d\times d}$, $\langle \boldsymbol{\pi}'^\star_{\alpha_k}, \boldsymbol{\gamma}\rangle$ converges to $\langle \boldsymbol{\pi}', \boldsymbol{\gamma}\rangle$. Hence, the limit of the RMD when α tends to 0 from above is $\langle \boldsymbol{\pi}', \boldsymbol{\gamma}\rangle$, which equals the RMD for $\alpha = 0$ as shown in the next property. ■

Property 2 *When $\alpha = 0$, the primal rot mover's distance reduces to:*

$$d'_{\boldsymbol{\gamma},0,\phi}(\mathbf{p},\mathbf{q}) = \langle \boldsymbol{\pi}', \boldsymbol{\gamma}\rangle \ , \quad (59)$$

and the unique primal rot mover's plan is the transport plan with minimal Bregman information:

$$\boldsymbol{\pi}'^\star_0 = \boldsymbol{\pi}' \ . \quad (60)$$

Proof Since $\boldsymbol{\pi}'$ is the unique global minimizer of ϕ on $\Pi(\mathbf{p},\mathbf{q})$, the regularized transport polytope reduces to the singleton $\Pi_{0,\phi}(\mathbf{p},\mathbf{q}) = \{\boldsymbol{\pi} \in \Pi(\mathbf{p},\mathbf{q}) \colon \phi(\boldsymbol{\pi}) \leq \phi(\boldsymbol{\pi}')\} = \{\boldsymbol{\pi}'\}$. The property follows immediately. ■

Property 3 *When α tends to $+\infty$, the primal rot mover's distance converges to the earth mover's distance:*

$$\lim_{\alpha\to+\infty} d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q}) = d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) \ . \quad (61)$$

Proof Let $\boldsymbol{\pi}^\star \in \Pi(\mathbf{p},\mathbf{q})$ be an earth mover's plan so that $d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) = \langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma}\rangle$. By continuity of the total cost $\langle\cdot,\boldsymbol{\gamma}\rangle$ on $\mathbb{R}^{d\times d}$, we have that for all $\epsilon > 0$, there exists an open neighborhood of $\boldsymbol{\pi}^\star$ such that $\langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle \leq \langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma}\rangle + \epsilon$ for any transport plan $\boldsymbol{\pi}$ within this neighborhood. We can always choose a transport plan such that $\boldsymbol{\pi} \in \operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$. Since $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q})) \subset \operatorname{dom}\phi$, $\phi(\boldsymbol{\pi})$ is finite and we can fix $\alpha_\epsilon = \phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\pi}') \geq 0$. Hence, $\boldsymbol{\pi} \in \Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ for any $\alpha \geq \alpha_\epsilon$, and we have $d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) \leq d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q}) \leq \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle \leq d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) + \epsilon$. ■

Property 4 *If $[0,1)^{d\times d} \subseteq \operatorname{dom}\phi$, then there exists a minimal $\alpha' \geq 0$ such that for all $\alpha \geq \alpha'$, the primal rot mover's distance reduces to the earth mover's distance:*

$$d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q}) = d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) \ . \quad (62)$$

Proof The extra condition guarantees that $\Pi(\mathbf{p},\mathbf{q}) \subset \operatorname{dom}\phi$, and thus that ϕ is bounded on the closed set $\Pi(\mathbf{p},\mathbf{q})$. The property is then a direct consequence of $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q}) = \Pi(\mathbf{p},\mathbf{q})$ for α large enough. ■

Property 5 *If $[0,1)^{d\times d} \subseteq \operatorname{dom}\phi$ and ϕ is strictly convex on $[0,1)^{d\times d}$, then the unique primal rot mover's plan for $\alpha = \alpha'$ is the earth mover's plan $\boldsymbol{\pi}_0^\star$ with minimal Bregman information:*

$$\boldsymbol{\pi}'^\star_{\alpha'} = \boldsymbol{\pi}_0^\star \ . \tag{63}$$

Proof First, we recall that the set of earth mover's plans $\boldsymbol{\pi}^\star$ is either a single vertex or a whole facet of $\Pi(\mathbf{p},\mathbf{q})$. Hence, it forms a closed convex subset in $\Pi(\mathbf{p},\mathbf{q})$, and there is a unique earth mover's plan $\boldsymbol{\pi}_0^\star$ with minimal Bregman information by strict convexity of ϕ on this subset. Second, it is trivial that all primal rot mover's plan $\boldsymbol{\pi}'^\star_{\alpha'}$ must be earth mover's plans. If there is a single vertex as earth mover's plan, then the property follows immediately. Otherwise, we can see the property geometrically as follows. The whole facet of earth mover's plans is orthogonal to $\boldsymbol{\gamma}$. Nevertheless, by strict convexity of ϕ on $[0,1)^{d\times d}$, the facet must be tangent to $\Pi_{\alpha',\phi}(\mathbf{p},\mathbf{q})$ at the unique earth mover's plan $\boldsymbol{\pi}_0^\star$ with minimal Bregman information $\phi(\boldsymbol{\pi}_0^\star) = \phi(\boldsymbol{\pi}') + \alpha'$, and $\boldsymbol{\pi}_0^\star$ is also the rot mover's plan $\boldsymbol{\pi}'^\star_{\alpha'}$. Another way to prove the property more formally is as follows. Suppose that a primal rot mover's plan $\boldsymbol{\pi}^\star$ is not the earth mover's plan with minimal Bregman information. We thus have $\phi(\boldsymbol{\pi}_0^\star) < \phi(\boldsymbol{\pi}^\star) \leq \phi(\boldsymbol{\pi}') + \alpha'$. We can then choose a smaller α' such that $\phi(\boldsymbol{\pi}_0^\star) \leq \phi(\boldsymbol{\pi}') + \alpha'$ and the RMD still equals the EMD for this smaller value, and actually all values in between by monotonicity. This leads to a contradiction and $\boldsymbol{\pi}_0^\star$ must be the earth mover's plan with minimal Bregman information. ■

Remark 4 *When $\alpha > \alpha'$, the regularized transport polytope might grow to include several earth mover's plans with different Bregman information, which are then all minimizers for the RMD. When we do not have strict convexity outside $(0,1)^{d\times d}$, there might also be multiple earth mover's plans with minimal Bregman information.*

If $[0,1)^{d\times d} \subseteq \operatorname{dom}\phi$, then it is easy to check that the strict convexity of ϕ on $[0,1)^{d\times d}$ is always verified when ϕ is separable under assumptions (A) or (B), or when $[0,1)^{d\times d} \subset \operatorname{int}(\operatorname{dom}\phi)$ under assumptions (B). This holds for almost all typical regularizers, notably for all regularizers considered in this paper except from minus the Burg entropy as defined in (30) and associated to the Itakura-Saito divergence in (29). For this latter regularizer, the solutions for an increasing α all lie within $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$, and the RMD never reaches the EMD. In such cases where the minimal α' does not exist, we can use the convention $\alpha' = +\infty$ since the RMD always converges to the EMD in the limit when α tends to $+\infty$. We can then prove that there is a unique rot mover's plan $\boldsymbol{\pi}'^\star_{\alpha}$ as long as $0 < \alpha < \alpha'$, which can be seen informally as follows. The solutions geometrically lie at the intersection of $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ and of a supporting hyperplane with normal $\boldsymbol{\gamma}$. By strict convexity of ϕ on $\operatorname{ri}(\Pi(\mathbf{p},\mathbf{q}))$, this intersection is a singleton inside the polytope. When the intersection reaches a facet, the only facet that can coincide locally with the hyperplane is the one that contains the earth mover's plans. Hence, we also have a singleton on the boundary of the polytope before reaching an earth mover's plan. We formally prove this uniqueness result next by exploiting duality.

3.3 Dual Problem

We now present the following two lemmas before defining our dual formulation for the RMD.

Lemma 8 *The regularized cost $\langle\cdot,\boldsymbol{\gamma}\rangle + \lambda\phi(\cdot)$, where $\lambda > 0$, attains its global minimum uniquely at $\boldsymbol{\xi} = \nabla\psi(-\boldsymbol{\gamma}/\lambda)$.*

Proof The regularized cost is convex with same domain as ϕ, and is strictly convex on $\operatorname{int}(\operatorname{dom}\phi)$. Thus, it attains its global minimum at a unique point $\boldsymbol{\xi} \in \operatorname{int}(\operatorname{dom}\phi)$ if and only if $\boldsymbol{\gamma} + \lambda\nabla\phi(\boldsymbol{\xi}) = \mathbf{0}$, or equivalently $\nabla\phi(\boldsymbol{\xi}) = -\boldsymbol{\gamma}/\lambda$. By assumptions (A4) and (A5), respectively (B4), $-\boldsymbol{\gamma}/\lambda \in \operatorname{dom}\nabla\psi$, so that the global minimum is attained uniquely at $\boldsymbol{\xi} = \nabla\psi(-\boldsymbol{\gamma}/\lambda)$ in virtue of the homeomorphism in (42) and (43). ■

Lemma 9 *The restriction of the regularized cost $\langle\cdot,\boldsymbol{\gamma}\rangle + \lambda\phi(\cdot)$ to the transport polytope $\Pi(\mathbf{p},\mathbf{q})$ attains its global minimum uniquely.*

Proof We notice that the regularized cost is equal to a Bregman divergence up to a positive factor and additive constant:

$$\langle\boldsymbol{\pi},\boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}) - \lambda\phi(\boldsymbol{\xi}) = \lambda B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}) \ . \tag{64}$$

Hence, its minimization over the closed convex set $\Pi(\mathbf{p},\mathbf{q})$ is equivalent to the Bregman projection of $\boldsymbol{\xi} \in \operatorname{int}(\operatorname{dom}\phi)$ onto $\Pi(\mathbf{p},\mathbf{q})$ according to the function ϕ of Legendre type. Since $\Pi(\mathbf{p},\mathbf{q}) \cap \operatorname{int}(\operatorname{dom}\phi) \neq \emptyset$, this projection exists and is unique. ■

Definition 10 *The dual rot mover's distance is the quantity defined as:*

$$d_{\boldsymbol{\gamma},\lambda,\phi}(\mathbf{p},\mathbf{q}) = \langle\boldsymbol{\pi}_\lambda^\star,\boldsymbol{\gamma}\rangle \ , \tag{65}$$

where the dual rot mover's plan $\boldsymbol{\pi}_\lambda^\star$ is given by:

$$\boldsymbol{\pi}_\lambda^\star = \underset{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})}{\operatorname{argmin}} \ \langle\boldsymbol{\pi},\boldsymbol{\gamma}\rangle + \lambda\phi(\boldsymbol{\pi}) \ . \tag{66}$$

Remark 5 *For the sake of notation, we omit the dependence on $\mathbf{p},\mathbf{q},\boldsymbol{\gamma},\phi$ in the index of dual rot mover's plans $\boldsymbol{\pi}_\lambda^\star$.*

We proceed with the following proposition that enlightens the relation between the RMD and associated Bregman divergence.

Proposition 11 *The dual rot mover's plan is the Bregman projection of $\boldsymbol{\xi}$ onto the transport polytope:*

$$\boldsymbol{\pi}_\lambda^\star = \underset{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})}{\operatorname{argmin}} \ B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}) \ . \tag{67}$$

Proof This is a consequence of the proof for Lemma 9. Indeed, from the definition in (66), we see that the rot mover's plan also minimizes (64). Therefore, it is the unique Bregman projection of $\boldsymbol{\xi}$ onto the transport polytope. ■

We have a geometrical interpretation where the regularization shrinks the solution toward the matrix $\boldsymbol{\xi}'$ that has minimal Bregman information.

Proposition 12 *The dual rot mover's plan $\boldsymbol{\pi}^\star_\lambda$ can be obtained as:*

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') \ . \tag{68}$$

Proof Developing the Bregman divergence based on its definition (19), we have:

$$B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') = \phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\xi}') - \langle \boldsymbol{\pi} - \boldsymbol{\xi}', \nabla\phi(\boldsymbol{\xi}')\rangle \ . \tag{69}$$

Since $\nabla\phi(\boldsymbol{\xi}') = \mathbf{0}$, the last term with scalar product vanishes and we are left out with $\phi(\boldsymbol{\pi})$ plus a constant term with respect to $\boldsymbol{\pi}$. Hence, we can replace $\phi(\boldsymbol{\pi})$ by $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}')$ in the minimization (66) that defines $\boldsymbol{\pi}^\star_\lambda$. ■

Under some additional conditions, this interpretation can also be seen as shrinking toward the transport plan $\boldsymbol{\pi}'$ with minimal Bregman information.

Proposition 13 *If $\boldsymbol{\pi}' \in \mathrm{ri}(\Pi(\mathbf{p},\mathbf{q}))$, then the dual rot mover's plan $\boldsymbol{\pi}^\star_\lambda$ can be obtained as:*

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') \ . \tag{70}$$

Proof If $\boldsymbol{\pi}' \in \mathrm{ri}(\Pi(\mathbf{p},\mathbf{q}))$, then we have equality in the generalized Pythagorean theorem (26), leading to:

$$B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}') = B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') + B_\phi(\boldsymbol{\pi}'\|\boldsymbol{\xi}') \ . \tag{71}$$

Since the last term is constant with respect to $\boldsymbol{\pi}$, we can replace $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}')$ by $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}')$ in the minimization (68) that characterizes $\boldsymbol{\pi}^\star_\lambda$. ■

Remark 6 *The proposition also holds trivially when the global minimum is attained on the transport polytope, that is, when $\boldsymbol{\xi}' = \boldsymbol{\pi}'$.*

Corollary 14 *Under assumptions (A), the dual rot mover's plan $\boldsymbol{\pi}^\star_\lambda$ can be obtained as:*

$$\boldsymbol{\pi}^\star_\lambda = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}') \ . \tag{72}$$

Proof This is a result of $\boldsymbol{\pi}' \in \Pi(\mathbf{p},\mathbf{q}) \cap \mathrm{int}(\mathrm{dom}\,\phi) = \mathrm{ri}(\Pi(\mathbf{p},\mathbf{q}))$ when $\mathrm{dom}\,\phi \subseteq \mathbb{R}_+^{d\times d}$, as shown in the proof of Corollary 7. ■

In the sequel, we also extend naturally the definition of the dual RMD for $\lambda = 0$ as the EMD. We then do not necessarily have uniqueness of dual rot mover's plans for $\lambda = 0$, and the geometrical interpretation in terms of a Bregman projection does not hold anymore for $\lambda = 0$. However, we have the following theorem based on duality theory that shows the equivalence between primal and dual ROT problems.

Theorem 15 *For all $\alpha > 0$, there exists $\lambda \geq 0$ such that the primal and dual rot mover's distances are equal:*

$$d'_{\boldsymbol{\gamma},\alpha,\phi}(\mathbf{p},\mathbf{q}) = d_{\boldsymbol{\gamma},\lambda,\phi}(\mathbf{p},\mathbf{q}) \ . \tag{73}$$

Moreover, if $\alpha < \alpha'$, then a corresponding value is such that $\lambda > 0$, and the primal and dual rot mover's plans are unique and equal:

$$\boldsymbol{\pi}'^\star_\alpha = \boldsymbol{\pi}^\star_\lambda \ . \tag{74}$$

Proof The primal problem can be seen as the minimization $p^\star$ of the cost $\langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle$ on $\Pi(\mathbf{p},\mathbf{q})$ subject to $\phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\pi}') - \alpha \leq 0$. The domain of this constrained convex problem is $\mathcal{D} = \Pi(\mathbf{p},\mathbf{q}) \cap \mathrm{dom}\,\phi \neq \emptyset$. The Lagrangian on $\mathcal{D}\times\mathbb{R}$ is given by $\mathcal{L}(\boldsymbol{\pi},\lambda) = \langle \boldsymbol{\pi}, \boldsymbol{\gamma}\rangle + \lambda(\phi(\boldsymbol{\pi}) - \phi(\boldsymbol{\pi}') - \alpha)$, and its minimization over $\mathcal{D}$ for a fixed $\lambda \geq 0$ has the same solutions $\boldsymbol{\pi}^\star$ as the dual problem. In addition, Slater's condition for convex problems, stating that there is a strictly feasible point in the relative interior of the domain, is verified as long as $\alpha > 0$. Indeed, we have $\mathrm{ri}(\mathcal{D}) = \mathrm{ri}(\Pi(\mathbf{p},\mathbf{q}))$. The existence of a strictly feasible point $\phi(\boldsymbol{\pi}) < \phi(\boldsymbol{\pi}') + \alpha$ then holds by continuity of ϕ at $\boldsymbol{\pi}' \in \mathrm{int}(\mathrm{dom}\,\phi)$. As a result, we have strong duality with a zero duality gap $p^\star = d^\star$, where $d^\star$ is the maximization of $g(\lambda)$ subject to $\lambda \geq 0$. Moreover, if $d^\star$ is finite, then it is attained at least once at a point $\lambda^\star$. This is the case since we already know that $p^\star$ is finite. Since $p^\star$ is also attained at least once at a point $\boldsymbol{\pi}^\star$ solution of the primal problem, we have the following chain:

$$\begin{aligned}
p^\star &= d^\star && (75)\\
&= \min_{\boldsymbol{\pi}\in\mathcal{D}} \mathcal{L}(\boldsymbol{\pi},\lambda^\star) && (76)\\
&\leq \mathcal{L}(\boldsymbol{\pi}^\star,\lambda^\star) && (77)\\
&= \langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma}\rangle + \lambda^\star(\phi(\boldsymbol{\pi}^\star) - \phi(\boldsymbol{\pi}') - \alpha) && (78)\\
&\leq \langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma}\rangle && (79)\\
&= p^\star \ . && (80)
\end{aligned}$$

Therefore, all inequalities are in fact equalities, $\boldsymbol{\pi}^\star$ also minimizes the Lagrangian over $\mathcal{D}$ and thus is a solution of the dual problem. In other words, the primal and dual RMD for α and $\lambda^\star$ are equal, and the primal solutions must be dual solutions too. For $\alpha < \alpha'$, the RMD has not reached the EMD yet, and thus we must have $\lambda^\star > 0$. Hence, the dual solution is unique, so that the primal solution is unique too and equal to the dual one. ■

Remark 7 *Corresponding values of α and λ depend on $\mathbf{p},\mathbf{q},\boldsymbol{\gamma},\phi$. In addition, there might be multiple values of λ that correspond to a given α.*

Again, the RMD does not verify the triangular inequality in general, and hence does not provide a true distance metric on Σ_d even if $\boldsymbol{\gamma}$ is a distance matrix. Nevertheless, we still have the result that the RMD is symmetric as soon as ϕ is invariant by transposition, which holds for separable regularizers $\phi_{ij} = \phi$, and $\boldsymbol{\gamma}$ is symmetric. We also obtain properties for the dual RMD that are similar to the ones for the primal RMD.

Property 6 *The dual rot mover's distance $d_{\boldsymbol{\gamma},\lambda,\phi}(\mathbf{p},\mathbf{q})$ is an increasing and continuous function of λ.*

Proof The fact that it is increasing can be seen as follows. Let $0 \leq \lambda_1 < \lambda_2$. By construction, we have the following inequalities:

$$\langle \boldsymbol{\pi}^\star_{\lambda_1}, \boldsymbol{\gamma} \rangle + \lambda_1 \phi(\boldsymbol{\pi}^\star_{\lambda_1}) \leq \langle \boldsymbol{\pi}^\star_{\lambda_2}, \boldsymbol{\gamma} \rangle + \lambda_1 \phi(\boldsymbol{\pi}^\star_{\lambda_2}) \ , \tag{81}$$

$$\langle \boldsymbol{\pi}^\star_{\lambda_2}, \boldsymbol{\gamma} \rangle + \lambda_2 \phi(\boldsymbol{\pi}^\star_{\lambda_2}) \leq \langle \boldsymbol{\pi}^\star_{\lambda_1}, \boldsymbol{\gamma} \rangle + \lambda_2 \phi(\boldsymbol{\pi}^\star_{\lambda_1}) \ . \tag{82}$$

Subtracting these two inequalities, we obtain that $\phi(\boldsymbol{\pi}^\star_{\lambda_1}) \geq \phi(\boldsymbol{\pi}^\star_{\lambda_2})$. Reinserting this result in the first inequality, we finally get $\langle \boldsymbol{\pi}^\star_{\lambda_1}, \boldsymbol{\gamma} \rangle \leq \langle \boldsymbol{\pi}^\star_{\lambda_2}, \boldsymbol{\gamma} \rangle$. The continuity of the dual RMD results from that of the primal RMD. Let $\lambda \geq 0$, and choose an arbitrary dual rot mover's plan $\boldsymbol{\pi}^\star_\lambda$ and earth mover's plan $\boldsymbol{\pi}^\star$. On the one hand, we have $\langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma} \rangle \leq \langle \boldsymbol{\pi}^\star_\lambda, \boldsymbol{\gamma} \rangle$. On the other hand, we have $\langle \boldsymbol{\pi}^\star_\lambda, \boldsymbol{\gamma} \rangle + \lambda \phi(\boldsymbol{\pi}^\star_\lambda) \leq \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle + \lambda \phi(\boldsymbol{\pi}')$, and thus $\langle \boldsymbol{\pi}^\star_\lambda, \boldsymbol{\gamma} \rangle \leq \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle + \lambda(\phi(\boldsymbol{\pi}') - \phi(\boldsymbol{\pi}^\star_\lambda)) \leq \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle$. Suppose we have a discontinuity of the dual RMD at λ. Then by monotonicity, there is a value $\langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma} \rangle < d < \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle$ that is not in the image of the dual RMD. But d is in the image of the primal RMD for a given $\alpha > 0$ by continuity. It means that $\nexists \lambda > 0$ such that $\langle \pi^\star_\lambda, C \rangle = d$, whereas, by continuity of the primal problem, we know that there exist $\alpha > 0$ such that $\langle \pi^\star_\alpha, C \rangle \geq d$. This is in contradiction with the duality result in Theorem 15, which implies that the image of the primal RMD for $\alpha > 0$ must be included in that of the dual RMD for $\lambda \geq 0$. ■

Property 7 *When λ tends to $+\infty$, the dual rot mover's distance converges to:*

$$\lim_{\lambda \to +\infty} d_{\boldsymbol{\gamma},\lambda,\phi}(\mathbf{p},\mathbf{q}) = \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle \ , \tag{83}$$

and the dual rot mover's plan converges in norm to the transport plan with minimal Bregman information:

$$\lim_{\lambda \to +\infty} \boldsymbol{\pi}^\star_\lambda = \boldsymbol{\pi}' \ . \tag{84}$$

Proof Let $(\lambda_k)_{k \in \mathbb{N}}$ be a sequence of positive numbers that tends to $+\infty$, and $(\boldsymbol{\pi}^\star_{\lambda_k})_{k \in \mathbb{N}}$ the associated rot mover's plans. By compactness of $\Pi(\mathbf{p},\mathbf{q})$, we can extract a subsequence of rot mover's plans that converges in norm to a point $\boldsymbol{\pi}^\star \in \Pi(\mathbf{p},\mathbf{q})$. For the sake of simplicity, we do not relabel this subsequence. By construction, we have $\langle \boldsymbol{\pi}^\star_{\lambda_k}, \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}') \leq \langle \boldsymbol{\pi}^\star_{\lambda_k}, \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}^\star_{\lambda_k}) \leq \langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}')$. The scalar products are bounded, so dividing the inequalities by λ_k and taking the limit, we obtain that $\phi(\boldsymbol{\pi}^\star_{\lambda_k})$ converges to $\phi(\boldsymbol{\pi}')$. By lower semi-continuity of ϕ, we thus have $\phi(\boldsymbol{\pi}^\star) \leq \phi(\boldsymbol{\pi}')$. Since the global minimum of ϕ on $\Pi(\mathbf{p},\mathbf{q})$ is attained uniquely at $\boldsymbol{\pi}'$, we must have $\boldsymbol{\pi}^\star = \boldsymbol{\pi}'$, and the original sequence also converges in norm to $\boldsymbol{\pi}'$. Hence, the dual rot mover's plan $\boldsymbol{\pi}_\lambda$ converges in norm to $\boldsymbol{\pi}'$ when λ tends to $+\infty$. By continuity of the total cost $\langle \cdot, \boldsymbol{\gamma} \rangle$ on $\mathbb{R}^{d \times d}$, $\langle \boldsymbol{\pi}^\star_{\lambda_k}, \boldsymbol{\gamma} \rangle$ converges to $\langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle$. Hence, the limit of the RMD when λ tends to $+\infty$ is $\langle \boldsymbol{\pi}', \boldsymbol{\gamma} \rangle$. ■

Property 8 *When λ tends to 0, the dual rot mover's distance converges to the earth mover's distance:*

$$\lim_{\lambda \to 0} d_{\boldsymbol{\gamma},\lambda,\phi}(\mathbf{p},\mathbf{q}) = d_{\boldsymbol{\gamma}}(\mathbf{p},\mathbf{q}) \ . \tag{85}$$

Proof This is a direct consequence of the dual RMD being continuous at $\lambda = 0$. ■

Property 9 *If $[0,1)^{d \times d} \subseteq \operatorname{dom} \phi$ and ϕ is strictly convex on $[0,1)^{d \times d}$, then the dual rot mover's plan converges in norm when λ tends to 0 to the earth mover's plan $\boldsymbol{\pi}^\star_0$ with minimal Bregman information:*

$$\lim_{\lambda \to 0} \boldsymbol{\pi}^\star_\lambda = \boldsymbol{\pi}^\star_0 \ . \tag{86}$$

Proof Let $(\lambda_k)_{k \in \mathbb{N}}$ be a sequence of positive numbers that converges to 0, and $(\boldsymbol{\pi}^\star_{\lambda_k})_{k \in \mathbb{N}}$ the associated rot mover's plans. By compactness of $\Pi(\mathbf{p},\mathbf{q})$, we can extract a subsequence of rot mover's plans that converges in norm to a point $\boldsymbol{\pi}^\star \in \Pi(\mathbf{p},\mathbf{q})$. For the sake of simplicity, we do not relabel this subsequence. By construction, we have $\langle \boldsymbol{\pi}^\star_0, \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}^\star_{\lambda_k}) \leq \langle \boldsymbol{\pi}^\star_{\lambda_k}, \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}^\star_{\lambda_k}) \leq \langle \boldsymbol{\pi}^\star_0, \boldsymbol{\gamma} \rangle + \lambda_k \phi(\boldsymbol{\pi}^\star_0)$. The regularizer ϕ is continuous on the polytope $\Pi(\mathbf{p},\mathbf{q}) \subseteq \operatorname{dom} \phi$, so taking the limit, we obtain that $\langle \boldsymbol{\pi}^\star_{\lambda_k}, \boldsymbol{\gamma} \rangle$ converges to $\langle \boldsymbol{\pi}^\star_0, \boldsymbol{\gamma} \rangle$. Therefore, $\boldsymbol{\pi}^\star$ must be an earth mover's plan. Now dividing by λ_k and taking the limit, we obtain that $\phi(\boldsymbol{\pi}^\star) \leq \phi(\boldsymbol{\pi}^\star_0)$. Since $\boldsymbol{\pi}^\star_0$ is the unique earth mover's plan with minimal Bregman information, we must have $\boldsymbol{\pi}^\star = \boldsymbol{\pi}^\star_0$. ■

3.4 Geometrical Insights

Our primal and dual formulations enlighten some intricate relations between optimal transportation theory (Villani, 2009) and information geometry (Amari and Nagaoka, 2000), where Bregman divergences are known to possess a dually flat structure with a generalized Pythagorean theorem for information projections. A schematic view of the underlying geometry for ROT problems is represented in Figure 1, and can be discussed as follows.

Our constructions start from the global minimizer $\boldsymbol{\xi}'$ of the regularizer ϕ (Lemma 1). The Bregman projection $\boldsymbol{\pi}'$ of $\boldsymbol{\xi}'$ onto the transport polytope $\Pi(\mathbf{p},\mathbf{q})$ has minimal Bregman information on $\Pi(\mathbf{p},\mathbf{q})$ (Lemma 2). The linear cost restricted to the regularized transport polytope $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ also attains its global minimum (Lemma 3). Such a minimizer $\boldsymbol{\pi}'^\star_\alpha$ is a primal rot mover's plan (Definition 4). We can interpret $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ as the intersection of $\Pi(\mathbf{p},\mathbf{q})$ with the Bregman ball of radius $B_\phi(\boldsymbol{\pi}' \| \boldsymbol{\xi}') + \alpha$ and center $\boldsymbol{\xi}'$ (Proposition 5). In certain cases, $\Pi_{\alpha,\phi}(\mathbf{p},\mathbf{q})$ is also the intersection of $\Pi(\mathbf{p},\mathbf{q})$ with the Bregman ball of radius α and center $\boldsymbol{\pi}'$, as a result of the generalized Pythagorean theorem $B_\phi(\boldsymbol{\pi} \| \boldsymbol{\xi}') = B_\phi(\boldsymbol{\pi} \| \boldsymbol{\pi}') + B_\phi(\boldsymbol{\pi}' \| \boldsymbol{\xi}')$ (Proposition 6, Corollary 7). All in all, this enforces the solutions to have small enough Bregman information, by constraining them to lie close to the matrix $\boldsymbol{\xi}'$ or transport plan $\boldsymbol{\pi}'$ with minimal Bregman information.

In our developments, we next introduce the global minimizer $\boldsymbol{\xi}$ of the regularized cost (Lemma 8). The regularized cost restricted to $\Pi(\mathbf{p},\mathbf{q})$ also attains its global minimum uniquely (Lemma 9). This minimizer defines the dual rot mover's plan $\boldsymbol{\pi}^\star_\lambda$ (Definition 10). Actually, $\boldsymbol{\pi}^\star_\lambda$ can be seen as the Bregman projection of $\boldsymbol{\xi}$ onto $\Pi(\mathbf{p},\mathbf{q})$ (Proposition 11). The regularization by the Bregman information is also equivalent to regularizing the solution toward $\boldsymbol{\xi}'$ (Proposition 12). In some cases, this can also be seen as regularizing toward $\boldsymbol{\pi}'$, as a result of the generalized Pythagorean theorem $B_\phi(\boldsymbol{\pi} \| \boldsymbol{\xi}') = B_\phi(\boldsymbol{\pi} \| \boldsymbol{\pi}') + B_\phi(\boldsymbol{\pi}' \| \boldsymbol{\xi}')$ (Proposition 13, Corollary 14). Again, this enforces the solutions to have small enough

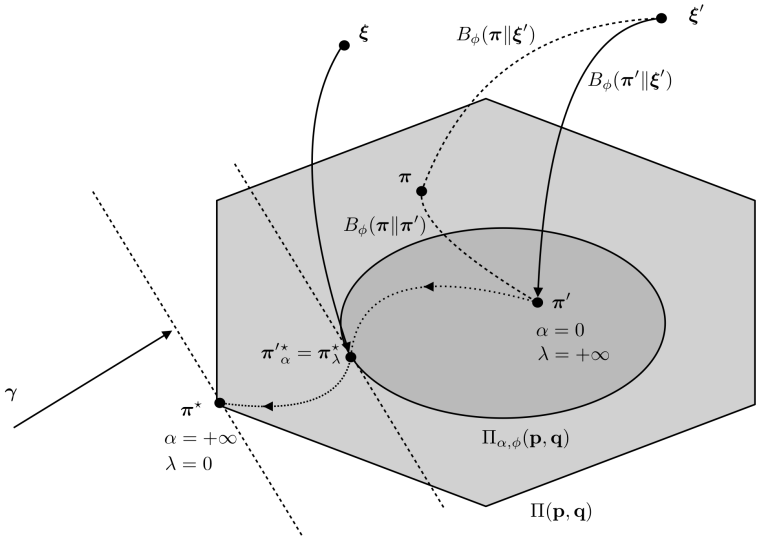


Figure 1: Geometry of regularized optimal transport.

Bregman information, by shrinking them toward the matrix $\boldsymbol{\xi}'$ or transport plan $\boldsymbol{\pi}'$ with minimal Bregman information.

We have duality between the primal and dual formulations, so that primal and dual rot mover's plans follow the same path on $\Pi(\mathbf{p},\mathbf{q})$ from no regularization ($\alpha = +\infty$, $\lambda = 0$) to full regularization ($\alpha = 0$, $\lambda = +\infty$) (Theorem 15). In the limit of no regularization, we obviously retrieve an earth mover's plan $\boldsymbol{\pi}^\star$ for the cost matrix $\boldsymbol{\gamma}$. By duality, it is also intuitive that the additional constraint for the primal formulation, seen in the equivalent forms of $\phi(\boldsymbol{\pi})$, $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}')$ or $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\pi}')$, leads to an analog penalty for the dual formulation in the same respective form.

Since $\boldsymbol{\xi}' = \mathbf{1}$, $\boldsymbol{\pi}' = \mathbf{p}\mathbf{q}^\top$, $\boldsymbol{\xi} = \exp(-\boldsymbol{\gamma}/\lambda)$, for minus the Boltzmann-Shannon entropy and Kullback-Leibler divergence, we retrieve the existing results discussed in Section 1.2 as a specific case (Cuturi, 2013; Benamou et al., 2015). In addition, we can readily generalize the estimation of contingency tables with fixed marginals to a matrix nearness problem based on other divergences than the Kullback-Leibler divergence (Dhillon and Tropp, 2007). Given a rough estimate $\boldsymbol{\xi} \in \operatorname{int}(\operatorname{dom}\phi)$, a contingency table with fixed marginals $\mathbf{p},\mathbf{q}$ can be estimated by Bregman projection of $\boldsymbol{\xi}$ onto $\Pi(\mathbf{p},\mathbf{q})$:

$$\boldsymbol{\pi}^\star = \operatorname*{argmin}_{\boldsymbol{\pi}\in\Pi(\mathbf{p},\mathbf{q})} B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi}) \ . \tag{87}$$

This simply amounts to solving a dual ROT problem with an arbitrary penalty $\lambda > 0$ and a cost matrix $\boldsymbol{\gamma} = -\lambda\nabla\phi(\boldsymbol{\xi})$.

Finally, since Bregman divergences are invariant under adding an affine term to their generator, it is straightforward to generalize ROT problems by shrinking toward an arbitrary prior matrix $\boldsymbol{\xi},\boldsymbol{\xi}' \in \operatorname{int}(\operatorname{dom}\phi)$, or transport plan $\boldsymbol{\pi}' \in \Pi(\mathbf{p},\mathbf{q})$. This is indeed equivalent to translating the regularizer by the appropriate amount $\phi(\boldsymbol{\pi}) + \langle\boldsymbol{\pi},\boldsymbol{\delta}\rangle$, so that the global minimizer is now attained at the desired point. Equivalently, this amounts to translating the cost matrix as $\boldsymbol{\gamma} + \lambda\boldsymbol{\delta}$ instead.

4. Algorithmic Derivations

In this section, we introduce algorithmic methods to solve ROT problems. We focus without lack of generality on the dual problem, which can be solved efficiently via alternate Bregman projections. The primal problem can then easily be solved for $0 < \alpha < \alpha'$ by a bisection search on $\lambda > 0$. For $\alpha = 0$, we could simply use alternate Bregman projections to project $\nabla\psi(\mathbf{0})$ instead of $\nabla\psi(-\boldsymbol{\gamma}/\lambda)$ in virtue of Lemmas 1 and 2, which actually corresponds to the special case $\boldsymbol{\gamma} = \mathbf{0}$ in our algorithms, though this is not really relevant in practice since this completely removes the linear influence of the total cost from the ROT problem. In the limit $\alpha \geq \alpha'$, a classical OT solver such as the network simplex can directly be used. We first study the underlying Bregman projections in their generic form (Section 4.1) and specifically develop the case of separable divergences (Section 4.2). We then derive the two generic schemes of ASA (Section 4.3) and NASA (Section 4.4) to solve dual ROT problems, depending on whether the domain of the smooth convex regularizer lies within the non-negative orthant or not. We also enhance both algorithms in the separable case with a sparse extension (Section 4.5), and finally discuss some practical considerations of our methods (Section 4.6). To simplify notations, we omit the penalty value λ in the index and simply write $\boldsymbol{\pi}^\star$ for the rot mover's plan.

4.1 Generic Projections

The closed convex transport polytope $\Pi(\mathbf{p},\mathbf{q})$ is the intersection of the non-negative orthant:

$$\mathcal{C}_0 = \mathbb{R}_+^{d\times d} \ , \tag{88}$$

which is a polyhedral subset, with two affine subspaces:

$$\mathcal{C}_1 = \{\boldsymbol{\pi} \in \mathbb{R}^{d\times d} \colon \boldsymbol{\pi}\mathbf{1} = \mathbf{p}\} \ , \tag{89}$$

$$\mathcal{C}_2 = \{\boldsymbol{\pi} \in \mathbb{R}^{d\times d} \colon \boldsymbol{\pi}^\top\mathbf{1} = \mathbf{q}\} \ . \tag{90}$$

The Bregman projection $\boldsymbol{\pi}^\star$ onto $\Pi(\mathbf{p},\mathbf{q})$ can then be obtained by alternate Bregman projections onto $\mathcal{C}_0,\mathcal{C}_1,\mathcal{C}_2$, where we expect that these latter projections are easier to compute.

On the one hand, the Karush-Kuhn-Tucker conditions for Bregman projection $\boldsymbol{\pi}_0^\star$ of a given matrix $\overline{\boldsymbol{\pi}} \in \operatorname{int}(\operatorname{dom}\phi)$ onto $\mathcal{C}_0$ are necessary and sufficient, and write as follows:

$$\boldsymbol{\pi}_0^\star \geq \mathbf{0} \ , \tag{91}$$

$$\nabla\phi(\boldsymbol{\pi}_0^\star) - \nabla\phi(\overline{\boldsymbol{\pi}}) \geq \mathbf{0} \ , \tag{92}$$

$$(\nabla\phi(\boldsymbol{\pi}_0^\star) - \nabla\phi(\overline{\boldsymbol{\pi}})) \odot \boldsymbol{\pi}_0^\star = \mathbf{0} \ . \tag{93}$$

While these conditions are nontrivial to solve in general, we shall see that they admit an elegant solver specific to the non-separable squared Mahalanobis distances defined in (33) and generated by the quadratic form in (34). In addition, they also greatly simplify for separable divergences, which encompass all other divergences used in this paper.

On the other hand, the Lagrangians with Lagrange multipliers $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^d$ for the Bregman projections $\boldsymbol{\pi}_1^\star$ and $\boldsymbol{\pi}_2^\star$ of a given matrix $\overline{\boldsymbol{\pi}} \in \operatorname{int}(\operatorname{dom}\phi)$ onto $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively write as follows:

$$\mathcal{L}_1(\boldsymbol{\pi}, \boldsymbol{\mu}) = \phi(\boldsymbol{\pi}) - \langle \boldsymbol{\pi}, \nabla\phi(\overline{\boldsymbol{\pi}}) \rangle + \boldsymbol{\mu}^\top(\boldsymbol{\pi}\mathbf{1} - \mathbf{p}) \ , \tag{94}$$

$$\mathcal{L}_2(\boldsymbol{\pi}, \boldsymbol{\nu}) = \phi(\boldsymbol{\pi}) - \langle \boldsymbol{\pi}, \nabla\phi(\overline{\boldsymbol{\pi}}) \rangle + \boldsymbol{\nu}^\top(\boldsymbol{\pi}^\top\mathbf{1} - \mathbf{q}) \ . \tag{95}$$

Their gradients are given on $\operatorname{int}(\operatorname{dom}\phi)$ by:

$$\nabla\mathcal{L}_1(\boldsymbol{\pi}, \boldsymbol{\mu}) = \nabla\phi(\boldsymbol{\pi}) - \nabla\phi(\overline{\boldsymbol{\pi}}) + \boldsymbol{\mu}\mathbf{1}^\top \ , \tag{96}$$

$$\nabla\mathcal{L}_2(\boldsymbol{\pi}, \boldsymbol{\nu}) = \nabla\phi(\boldsymbol{\pi}) - \nabla\phi(\overline{\boldsymbol{\pi}}) + \mathbf{1}\boldsymbol{\nu}^\top \ , \tag{97}$$

and vanish at $\boldsymbol{\pi}_1^\star, \boldsymbol{\pi}_2^\star \in \operatorname{int}(\operatorname{dom}\phi)$ if and only if:

$$\boldsymbol{\pi}_1^\star = \nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \boldsymbol{\mu}\mathbf{1}^\top) \ , \tag{98}$$

$$\boldsymbol{\pi}_2^\star = \nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \mathbf{1}\boldsymbol{\nu}^\top) \ . \tag{99}$$

By duality, the Bregman projections onto $\mathcal{C}_1, \mathcal{C}_2$ are thus equivalent to finding the unique vectors $\boldsymbol{\mu}, \boldsymbol{\nu}$, such that the rows of $\boldsymbol{\pi}_1^\star$ sum up to $\mathbf{p}$, respectively the columns of $\boldsymbol{\pi}_2^\star$ sum up to $\mathbf{q}$:

$$\nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \boldsymbol{\mu}\mathbf{1}^\top)\mathbf{1} = \mathbf{p} \ , \tag{100}$$

$$\nabla\psi(\nabla\phi(\overline{\boldsymbol{\pi}}) - \mathbf{1}\boldsymbol{\nu}^\top)^\top\mathbf{1} = \mathbf{q} \ . \tag{101}$$

Similarly, solving for the Lagrange multipliers is an expensive problem in general, since the search space is of dimension d and we evaluate matrix functions of size $d \times d$. This is because a given entry μ_i, ν_j can actually modify any entry of the $d \times d$ matrix functions being evaluated. Again, we shall see that they can nevertheless be computed efficiently for separable divergences as well as the non-separable Mahalanobis distances.

4.2 Separable Case

Assuming that the regularizer ϕ is separable, the underlying Bregman projections can be computed more efficiently. To keep notations simple, we focus on separable divergences with same element-wise regularizer, and thus chiefly omit the indices $\phi_{ij} = \phi$. We emphasize, however, that it is straightforward to apply all our methods for separable divergences with different element-wise regularizers, which notably enables weighting a given element-wise regularizer.

In case of separability, the Karush-Kuhn-Tucker conditions for projection onto $\mathcal{C}_0$ simplify to provide a closed-form solution on primal parameters:

$$\pi_{0,ij}^\star = \max\{0, \overline{\pi}_{ij}\} \ . \tag{102}$$

Since ϕ' is increasing, this is equivalent on dual parameters to:

$$\theta_{0,ij}^\star = \max\{\phi'(0), \overline{\theta}_{ij}\} \ . \tag{103}$$

Now turning to projections onto $\mathcal{C}_1, \mathcal{C}_2$ for primal parameters $\pi_{1,ij}^\star, \pi_{2,ij}^\star$, we can divide the initial problems into d parallel subproblems in search space of dimension 1 each. This is much more efficient to solve than in the non-separable case. This can be summarized as looking for d separate Lagrange multipliers μ_i, respectively ν_j, such that:

$$\sum_{j=1}^{d} \psi'(\overline{\theta}_{ij} - \mu_i) = p_i \ , \tag{104}$$

$$\sum_{i=1}^{d} \psi'(\overline{\theta}_{ij} - \nu_j) = q_j \ . \tag{105}$$

Finding the optimal values $\mu_i, \nu_j \in \mathbb{R}$ through ψ' and the sums over rows or columns, however, is still nontrivial in general.

An analytical solution can be obtained in specific cases. Intuitively, we need to factor μ_i, ν_j out of ψ' as additive or multiplicative terms. This is related to Pexider's functional equations, which hold only for functions with a linear form $\psi'(\theta) = a\theta + b$, or exponential form $\psi'(\theta) = a\exp(b\theta)$, with $a, b \in \mathbb{R}$. This leads to regularizers with a quadratic form $\phi(\pi) = a\pi^2 + b\pi + c$, or entropic form $\phi(\pi) = a\pi\log\pi + b\pi + c$, with $a, b, c \in \mathbb{R}$. The constants a, b actually only scale and translate the cost matrix, whereas the constant c has no effect. Referring to Table 1, the quadratic case holds under assumptions (B), and thus requires Dykstra's algorithm for alternate Bregman projections with correction terms to ensure non-negativity by projection onto the polyhedral non-negative orthant. The entropic case holds under assumptions (A), using the POCS technique for alternate Bregman projection with no correction terms since the non-negativity is already ensured by the domain of the regularizer. The latter case reduces to the regularization of Cuturi (2013) and Benamou et al. (2015), so that we actually end up with the Sinkhorn-Knopp algorithm. Hence, the Euclidean norm associated to the squared Euclidean distance, and the entropic case associated to the Kullback-Leibler divergence, are reasonably the only two existing analytical schemes to find the sum constraint projections. For other ROT problems, available solvers for line search can be employed instead.

For simplicity, we assume hereafter that ψ is twice continuously differentiable with ψ'' positive and ψ' verifying the necessary and sufficient condition (40) on its whole domain. Therefore, we can use the Newton-Raphson method with guarantees of global convergence. This encompasses most of the common regularizers, and notably all regularizers used in this paper except from the Fermi-Dirac entropy, ℓ_p norms and Hellinger distance. When the condition (40) for global convergence is not met on the whole domain, it is still possible to apply the Newton-Raphson method after careful initialization, so as to restrict to a smaller interval where the condition holds. This is discussed in more detail with practical examples for the Fermi-Dirac entropy, ℓ_p norms and Hellinger distance in Section 5, where the first-order derivatives are increasing convex on half of the domain and increasing concave on the other half. When the second-order derivatives do not exist, are not continuous or vanish at some points, a similar strategy can be applied. This is again discussed for the ℓ_p norms in

Section 5, where the second-order derivative is undefined or vanishes at 0 depending on the value of the parameter. If such an initialization is not possible, then a bisection search can always be applied instead of the Newton-Raphson method.

To apply the Newton-Raphson method, we exploit the following functions:

$$f(\mu_i) = -\sum_{j=1}^{d} \psi'(\overline{\theta}_{ij} - \mu_i) \ , \tag{106}$$

$$g(\nu_j) = -\sum_{i=1}^{d} \psi'(\overline{\theta}_{ij} - \nu_j) \ , \tag{107}$$

defined respectively on the open intervals $(\hat{\theta}_i - \overline{\theta}, +\infty)$ and $(\check{\theta}_j - \overline{\theta}, +\infty)$, where $0 < \overline{\theta} \leq +\infty$ is such that $\operatorname{dom} \psi = (-\infty, \overline{\theta})$, and $\hat{\theta}_i = \max\{\overline{\theta}_{ij}\}_{1 \leq j \leq d}$, $\check{\theta}_j = \max\{\overline{\theta}_{ij}\}_{1 \leq i \leq d}$. Their continuous derivatives are given by:

$$f'(\mu_i) = \sum_{j=1}^{d} \psi''(\overline{\theta}_{ij} - \mu_i) \ , \tag{108}$$

$$g'(\nu_j) = \sum_{i=1}^{d} \psi''(\overline{\theta}_{ij} - \nu_j) \ , \tag{109}$$

and are positive, so that f, g are strictly increasing on their whole domain, and thus on any closed interval with endpoints consisting of a feasible point and a solution. By construction, f, g also verify the necessary and sufficient condition (40) for global convergence, and we know that there are unique solutions to $f(\mu_i) = -p_i$ and $g(\nu_j) = -q_j$. Hence, the Newton-Raphson updates:

$$\mu_i \leftarrow \mu_i + \frac{\sum_{j=1}^{d} \psi'(\overline{\theta}_{ij} - \mu_i) - p_i}{\sum_{j=1}^{d} \psi''(\overline{\theta}_{ij} - \mu_i)} \ , \tag{110}$$

$$\nu_j \leftarrow \nu_j + \frac{\sum_{i=1}^{d} \psi'(\overline{\theta}_{ij} - \nu_j) - q_j}{\sum_{i=1}^{d} \psi''(\overline{\theta}_{ij} - \nu_j)} \ , \tag{111}$$

converge to the optimal solutions with a quadratic rate for any feasible starting points. By construction, we also know that initialization can be done with $\mu_i \leftarrow 0$, $\nu_j \leftarrow 0$. To avoid storing the intermediate Lagrange multipliers, the updates can then directly be written on dual parameters:

$$\theta^\star_{1,ij} \leftarrow \theta^\star_{1,ij} - \frac{\sum_{j=1}^{d} \psi'(\theta^\star_{1,ij}) - p_i}{\sum_{j=1}^{d} \psi''(\theta^\star_{1,ij})} \ , \tag{112}$$

$$\theta^\star_{2,ij} \leftarrow \theta^\star_{2,ij} - \frac{\sum_{i=1}^{d} \psi'(\theta^\star_{2,ij}) - q_j}{\sum_{i=1}^{d} \psi''(\theta^\star_{2,ij})} \ , \tag{113}$$

after initialization by $\theta^\star_{1,ij} \leftarrow \overline{\theta}_{ij}$, $\theta^\star_{2,ij} \leftarrow \overline{\theta}_{ij}$.

Algorithm 1 Alternate scaling algorithm.

$\boldsymbol{\theta}^\star \leftarrow -\boldsymbol{\gamma}/\lambda$
repeat
  $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \boldsymbol{\mu}\mathbf{1}^\top$, where $\boldsymbol{\mu}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}^\star - \boldsymbol{\mu}\mathbf{1}^\top)\mathbf{1} = \mathbf{p}$
  $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\nu}^\top$, where $\boldsymbol{\nu}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\nu}^\top)^\top\mathbf{1} = \mathbf{q}$
until convergence
$\boldsymbol{\pi}^\star \leftarrow \nabla\psi(\boldsymbol{\theta}^\star)$

4.3 Alternate Scaling Algorithm

Under assumptions (A), we can drop the non-negative constraint since it is already ensured by $\operatorname{dom}\phi \subseteq \mathbb{R}_+^{d \times d}$ (Table 1). The POCS technique in its basic form (38) then states that the projection of $\boldsymbol{\xi}$ onto $\Pi(\mathbf{p}, \mathbf{q})$ can be obtained by alternate Bregman projections onto the affine subspaces $\mathcal{C}_1$ and $\mathcal{C}_2$ with linear convergence. Clearly, the underlying control mapping takes each output value an infinite number of times. Since we have just two sets, the only possible alternative in the control mapping is to swap the order of projections starting from $\mathcal{C}_2$ instead of $\mathcal{C}_1$, which actually amounts to swapping the input distributions $\mathbf{p}, \mathbf{q}$ and transposing the cost matrix $\boldsymbol{\gamma}$, to obtain the transposed of the rot mover's plan. We thus focus on the first choice without lack of generality.

Starting from $\boldsymbol{\xi}$ and writing the successive vectors $\boldsymbol{\mu}^{(k)}, \boldsymbol{\nu}^{(k)}$ along iterations, we have the following sequence:

$$\nabla\psi(-\boldsymbol{\gamma}/\lambda) \rightarrow \nabla\psi\left(-\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top\right) \tag{114}$$

$$\rightarrow \nabla\psi\left(-\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\right) \tag{115}$$

$$\rightarrow \dots \tag{116}$$

$$\rightarrow \nabla\psi\left(-\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top} - \dots - \boldsymbol{\mu}^{(k)}\mathbf{1}^\top\right) \tag{117}$$

$$\rightarrow \nabla\psi\left(-\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top} - \dots - \boldsymbol{\mu}^{(k)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k)\top}\right) \tag{118}$$

$$\rightarrow \dots \tag{119}$$

$$\rightarrow \boldsymbol{\pi}^\star \ . \tag{120}$$

In other terms, we obtain the rot mover's plan $\boldsymbol{\pi}^\star$ by scaling iteratively the rows and columns of the successive estimates through $\nabla\psi$. An efficient algorithm, called ASA, is to store a unique $d \times d$ matrix in dual parameter space and update it by alternating the projections in primal parameter space (Algorithm 1). The updates have a complexity in $O(d^2)$ once the vectors $\boldsymbol{\mu}, \boldsymbol{\nu}$ are obtained.

In the separable case, the projections can be obtained by iterating the respective Newton-Raphson update steps, which can be written compactly with matrix and vector operations (Algorithm 2). The complexity for the updates are now clearly in $O(d^2)$. In more detail, each update step features one vector row or column replication, one vector element-wise division, one vector subtraction, one matrix subtraction, two matrix row or column sums, and two element-wise matrix function evaluations. Because of separability, we can expect

Algorithm 2 Alternate scaling algorithm in the separable case.

$\boldsymbol{\theta}^\star \leftarrow -\boldsymbol{\gamma}/\lambda$
repeat
  repeat
    $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \frac{\psi'(\boldsymbol{\theta}^\star)\mathbf{1} - \mathbf{p}}{\psi''(\boldsymbol{\theta}^\star)\mathbf{1}}\mathbf{1}^\top$
  until convergence
  repeat
    $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \mathbf{1}\frac{\mathbf{1}^\top\psi'(\boldsymbol{\theta}^\star) - \mathbf{q}^\top}{\mathbf{1}^\top\psi''(\boldsymbol{\theta}^\star)}$
  until convergence
until convergence
$\boldsymbol{\pi}^\star \leftarrow \psi'(\boldsymbol{\theta}^\star)$

the required number of iterations for convergence in the different loops to be independent of the data dimension, and thus expect a quadratic empirical complexity as well.

4.4 Non-negative Alternate Scaling Algorithm

Under assumptions (B), we must now include the non-negative constraint since $\operatorname{dom}\phi \nsubseteq \mathbb{R}_+^{d\times d}$ (Table 1). We suggest to ensure non-negativity of each update, and thus follow a cycle of projections onto $\mathcal{C}_0, \mathcal{C}_1, \mathcal{C}_0, \mathcal{C}_2$. The underlying control mapping is a fortiori essentially cyclic. For practical reasons, we also ensure non-negativity of the output solution with a final projection onto $\mathcal{C}_0$. Again, swapping the order of projections onto $\mathcal{C}_1, \mathcal{C}_2$ is equivalent to swapping the input distributions $\mathbf{p}, \mathbf{q}$ and transposing the cost matrix $\boldsymbol{\gamma}$ to obtain the transposed of the rot mover's plan. Other control mappings could also be exploited, for example by ensuring non-negativity every two or more sum constraint projections. We do not discuss such variants here and focus on the above-mentioned sequence. The non-negative orthant being polyhedral but not affine, we also need to incorporate correction terms $\boldsymbol{\vartheta}, \boldsymbol{\varrho}, \boldsymbol{\varsigma}$ for all three projections. In more detail, the projections are computed after correction so that we do not directly project the obtained updates $\boldsymbol{\theta}^\star$ but the corrected updates $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star + \boldsymbol{\vartheta}$, $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star + \boldsymbol{\varrho}$, and $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star + \boldsymbol{\varsigma}$ for the respective subsets. The correction terms are also updated as the difference $\overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$ between the projected point and its projection. Dykstra's algorithm (36) for Bregman divergences with corrections (37) then guarantees that the projection of $\boldsymbol{\xi}$ onto $\Pi(\mathbf{p}, \mathbf{q})$ is obtained with linear convergence.

A general algorithm, called NASA, is to store $d \times d$ matrices for projected points, projections and correction terms in dual parameter space, update them accordingly and finally go back to primal parameter space (Algorithm 3). The updates have a complexity in $O(d^2)$ once the Karush-Kuhn-Tucker conditions are solved or Lagrange multipliers $\boldsymbol{\mu}, \boldsymbol{\nu}$ are obtained.

In the separable case, the non-negativity constraint can be obtained analytically and the sequence of updates greatly simplifies. Starting from $\boldsymbol{\xi}$ and writing the successive vectors $\boldsymbol{\mu}^{(k)}, \boldsymbol{\nu}^{(k)}$ along iterations, we have:

$$
\begin{aligned}
\psi'(-\boldsymbol{\gamma}/\lambda) &\to \psi'\big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda\}\big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda\} - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top\Big)
\end{aligned}
$$

Algorithm 3 Non-negative alternate scaling algorithm.

$\boldsymbol{\theta}^\star \leftarrow -\boldsymbol{\gamma}/\lambda$
$\boldsymbol{\vartheta} \leftarrow \mathbf{0}$
$\boldsymbol{\varrho} \leftarrow \mathbf{0}$
$\boldsymbol{\varsigma} \leftarrow \mathbf{0}$
$\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^\star + \boldsymbol{\vartheta}$
$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}$, $\boldsymbol{\theta} \geq \overline{\boldsymbol{\theta}}$, $(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}$
$\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$
repeat
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^\star + \boldsymbol{\varrho}$
  $\boldsymbol{\theta}^\star \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\mu}\mathbf{1}^\top$, where $\boldsymbol{\mu}$ uniquely solves $\nabla\psi(\overline{\boldsymbol{\theta}} - \boldsymbol{\mu}\mathbf{1}^\top)\mathbf{1} = \mathbf{p}$
  $\boldsymbol{\varrho} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^\star + \boldsymbol{\vartheta}$
  $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}$, $\boldsymbol{\theta} \geq \overline{\boldsymbol{\theta}}$, $(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}$
  $\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^\star + \boldsymbol{\varsigma}$
  $\boldsymbol{\theta}^\star \leftarrow \overline{\boldsymbol{\theta}} - \mathbf{1}\boldsymbol{\nu}^\top$, where $\boldsymbol{\nu}$ uniquely solves $\nabla\psi(\overline{\boldsymbol{\theta}} - \mathbf{1}\boldsymbol{\nu}^\top)^\top\mathbf{1} = \mathbf{q}$
  $\boldsymbol{\varsigma} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$
  $\overline{\boldsymbol{\theta}} \leftarrow \boldsymbol{\theta}^\star + \boldsymbol{\vartheta}$
  $\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}$, where $\boldsymbol{\theta}$ uniquely solves $\nabla\psi(\boldsymbol{\theta}) \geq \mathbf{0}$, $\boldsymbol{\theta} \geq \overline{\boldsymbol{\theta}}$, $(\boldsymbol{\theta} - \overline{\boldsymbol{\theta}}) \odot \nabla\psi(\boldsymbol{\theta}) = \mathbf{0}$
  $\boldsymbol{\vartheta} \leftarrow \overline{\boldsymbol{\theta}} - \boldsymbol{\theta}^\star$
until convergence
$\boldsymbol{\pi}^\star \leftarrow \nabla\psi(\boldsymbol{\theta}^\star)$

$$
\begin{aligned}
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top\}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top\} - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\} + \boldsymbol{\mu}^{(1)}\mathbf{1}^\top - \boldsymbol{\mu}^{(2)}\mathbf{1}^\top\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(1)\top}\} + \mathbf{1}\boldsymbol{\nu}^{(1)\top} - \mathbf{1}\boldsymbol{\nu}^{(2)\top}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(2)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(2)\top}\}\Big) \\
&\to \dots \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k)\top}\}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k)\top}\} + \boldsymbol{\mu}^{(k)}\mathbf{1}^\top - \boldsymbol{\mu}^{(k+1)}\mathbf{1}^\top\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k)\top}\}\Big) \\
&\to \psi'\Big(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k)\top}\} + \mathbf{1}\boldsymbol{\nu}^{(k)\top} - \mathbf{1}\boldsymbol{\nu}^{(k+1)\top}\Big)
\end{aligned}
$$

Algorithm 4 Non-negative alternate scaling algorithm in the separable case.

$\widetilde{\boldsymbol{\theta}} \leftarrow -\boldsymbol{\gamma}/\lambda$
$\boldsymbol{\theta}^\star \leftarrow \max\{\phi'(\mathbf{0}), \widetilde{\boldsymbol{\theta}}\}$
repeat
  $\boldsymbol{\tau} \leftarrow \mathbf{0}$
  repeat
    $\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} + \frac{\psi'(\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top)\mathbf{1} - \mathbf{p}}{\psi''(\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top)\mathbf{1}}$
  until convergence
  $\widetilde{\boldsymbol{\theta}} \leftarrow \widetilde{\boldsymbol{\theta}} - \boldsymbol{\tau}\mathbf{1}^\top$
  $\boldsymbol{\theta}^\star \leftarrow \max\{\phi'(\mathbf{0}), \widetilde{\boldsymbol{\theta}}\}$
  $\boldsymbol{\sigma} \leftarrow \mathbf{0}$
  repeat
    $\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \frac{\mathbf{1}^\top\psi'(\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top) - \mathbf{q}^\top}{\mathbf{1}^\top\psi''(\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top)}$
  until convergence
  $\widetilde{\boldsymbol{\theta}} \leftarrow \widetilde{\boldsymbol{\theta}} - \mathbf{1}\boldsymbol{\sigma}^\top$
  $\boldsymbol{\theta}^\star \leftarrow \max\{\phi'(\mathbf{0}), \widetilde{\boldsymbol{\theta}}\}$
until convergence
$\boldsymbol{\pi}^\star \leftarrow \psi'(\boldsymbol{\theta}^\star)$

$$
\begin{aligned}
&\to \psi'\left(\max\{\phi'(\mathbf{0}), -\boldsymbol{\gamma}/\lambda - \boldsymbol{\mu}^{(k+1)}\mathbf{1}^\top - \mathbf{1}\boldsymbol{\nu}^{(k+1)\top}\}\right)\\
&\to \dots\\
&\to \boldsymbol{\pi}^\star \ .
\end{aligned}
$$

An efficient algorithm then exploits the differences $\boldsymbol{\tau}^{(k)} = \boldsymbol{\mu}^{(k)} - \boldsymbol{\mu}^{(k-1)}$ and $\boldsymbol{\sigma}^{(k)} = \boldsymbol{\nu}^{(k)} - \boldsymbol{\nu}^{(k-1)}$ to scale the rows and columns (Algorithm 4). We store $d \times d$ matrices as well as difference vectors instead of correction matrices. The algorithm can then be interpreted as producing interleaved updates between the projections according to the max operator and according to the respective scalings. The updates in NASA now clearly have a complexity in $O(d^2)$ when using the Newton-Raphson method for scaling, with similar matrix and vector operations to ASA in the separable case, and an expected empirical complexity that is quadratic.

4.5 Sparse Extension

In the separable case, it is possible to develop a sparse extension of both our methods ASA and NASA. Storing and updating full $d \times d$ matrices becomes expensive with the data dimension. Instead, we allow for infinite entries in the cost matrix $\boldsymbol{\gamma}$, meaning that the transport of mass between certain bins is proscribed. As a result, the corresponding entries of $\boldsymbol{\pi}^\star$ must be null. Eventually, we can drop all these entries so that we just need to store and update the remaining ones. The RMD via the Frobenius inner product $\langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma} \rangle$ is then computed without accounting for discarded entries, or equivalently by setting indefinite element-wise products $0 \times \infty = 0$ by convention, so it naturally costs nothing to move no mass on a path that is forbidden. This leads to an expected complexity in $O(r)$, where r is

the number of finite entries in $\boldsymbol{\gamma}$. Typically, r can be chosen in the order of magnitude of d, so as to obtain a linear instead of quadratic empirical complexity.

In practice, both ASA and NASA are compatible with this strategy. We always have $\lim_{\theta \to -\infty} \psi'(\theta) = 0$ under assumptions (A) for ASA. Under assumptions (B) for NASA, this limit might be finite or infinite but is necessarily negative, so also leads to 0 after enforcing non-negativity by projection onto the non-negative orthant. As a result, the obtained sequence of projections preserves the desired zeros in both algorithms, and an infinite element-wise cost does lead to no mass transport at all between the corresponding bins. In theory, we can understand this extension in light of the dual formulation seen as a Bregman projection in (67). Under assumptions (B), we always have $0 \in \operatorname{int}(\operatorname{dom}\phi)$ and thus $B_\phi(0\|0) = 0$. Hence, Dykstra's algorithm is readily applicable in the sparse version. Under assumptions (A), however, we have $0 \notin \operatorname{int}(\operatorname{dom}\phi)$, and even sometimes $0 \notin \operatorname{dom}\phi$ as for the Itakura-Saito divergence. We can nonetheless extend the domain of the element-wise divergence at the origin by continuity on the diagonal, that is, by setting it null as $B_\phi(0\|0) = 0$. This is akin to considering absolutely continuous measures, also known as dominated measures, and Radon-Nikodym derivatives to generalize the definition of Bregman divergences. Kurras (2015) then showed that the POCS method still holds with this convention by introducing a notion of locally affine spaces.

With such a sparse extension, however, we must take care that a sparse solution does exist, meaning that there is a transport plan in the transport polytope that has the desired zeros. For example, if all entries of $\boldsymbol{\gamma}$ are infinite, then there are obviously no possible sparse solutions since we enforce all entries of the plan to be null. A necessary condition for the existence of a sparse solution is that for any entry q_j, all entries p_k from which we are allowed to transport mass must provide enough total mass to fill q_j completely. Similarly, for any entry p_i, all entries q_k to which we are allowed to transport mass must require enough total mass to empty p_i completely. Unfortunately, sufficient conditions are not so intuitive. Idel (2016, Theorem 4.1) studied such problems thoroughly and elucidated several necessary and sufficient conditions for sparse solutions to exist, but these conditions are nontrivial to use from in practice. Kurras (2015) advocates trying first to compute a solution with the desired sparsity, and if no solution can be found, then gradually reduce sparsity until a solution is found. This might still speed up computation drastically because of the linear instead of quadratic complexity. Lastly, we remark that it is not evident to propose a sparse extension for the non-separable case in general, since a given entry of $\boldsymbol{\gamma}$ might influence all entries of $\boldsymbol{\pi}^\star$.

4.6 Practical Considerations

As noticed by Cuturi (2013) and Benamou et al. (2015), the Sinkhorn-Knopp algorithm might fail to converge because of numerical instability when the penalty λ gets small. In particular, unless taking special care of numerical stabilization (Schmitzer, 2016b), a direct limitation is the machine precision under which some entries of $\exp(-\boldsymbol{\gamma}/\lambda)$ are represented as zeros in memory. Such issues occur similarly for other regularizations, notably via the representation $\nabla\psi(-\boldsymbol{\gamma}/\lambda)$ of the unconstrained solution to project. Therefore, the proposed methods are actually competitive in a range where the penalty λ is not too small, and for which the rot mover's plan $\boldsymbol{\pi}^\star$ exhibits a significant amount of smoothing. Hence, we do

not target the same problems as traditional schemes such as interior point methods or the network simplex.

In addition, the different Bregman projections in our algorithms are most of the time approximate up to a given tolerance depending on the termination criterion used for convergence. Exceptions occur for the sum constraints with the Euclidean distance or Kullback-Leibler divergence, as well as the non-negativity constraints in the separable case, which are obtained analytically. A natural question to raise is then whether our algorithms still converge when the projections are approximate only. However, this is relatively hard to answer in theory. We did not observe in practice any problem of convergence when using sufficiently good approximations. Furthermore, first approximations can be quite rough without affecting convergence as long as final approximations are good enough. Sometimes, even alternating a single or two steps of the Newton-Raphson method throughout the main iterations the algorithm still works, though this is not systematic. Thus, we advocate for safety to use a tight tolerance for the auxiliary projections.

We also observed numerical instability of the Newton-Raphson updates for separable divergences under assumptions (A). This is due to the denominator being based on ψ'' with limit $\lim_{\theta \to -\infty} \psi''(\theta) = 0$, that is, for entries π close to zero. It is possible, however, to make the updates of μ_i, ν_j much more stable in practice by using the max truncation operator, despite theoretical guarantees of convergence without it. Specifically, we know that the entries $\pi^\star_{1,ij}$ must lie between 0 and p_i, and $\pi^\star_{2,ij}$ between 0 and q_j. Hence, we can lower bound μ_i and ν_j by $\hat{\theta}_i - \phi'(p_i)$ and $\hat{\theta}_j - \phi'(q_j)$, respectively. Interestingly, this also speeds up the convergence of the updates significantly when the initialization by 0 is far from the actual solution.

A possible termination criterion for the main and auxiliary iterations is to compute the marginal difference between the updated matrix and $\mathbf{p}, \mathbf{q}$. In the auxiliary iterations for the two scaling projections, we compare the sums of rows or columns to $\mathbf{p}$ or $\mathbf{q}$ respectively, and in the main iterations of the algorithm, we compare both marginals simultaneously. Typically, we can use the ℓ_p (quasi-)norm with $0 < p \leq +\infty$ to assess the marginal difference, and the auxiliary tolerance should be at least the main one for sufficient precision of the approximations. Two alternative quantities in absolute or relative scales can also be used for termination, either the variation with ℓ_p (quasi-)norm in the updated matrix or in the updated distance. Here the auxiliary tolerance should be at least the square of the main one. This seems reasonable for $\boldsymbol{\pi}^\star$ given the quadratic rate of convergence for the Newton-Raphson method versus the linear one for alternate Bregman projections, as well as for $\langle \boldsymbol{\pi}^\star, \boldsymbol{\gamma} \rangle$ under the Cauchy-Schwarz inequality. In all cases, convergence can be checked either after each iteration or after a given number of iterations to reduce the underlying cost of computing the termination criterion. We can also fix a maximum number of main and auxiliary iterations to limit the overall running time.

Regarding implementation, the matrix and vector operations used for ASA and NASA in the separable case are well-suited for fast calculation on a GPU and for processing of multiple input distributions in parallel. By working directly in the primal parameter space, the Sinkhorn-Knopp algorithm is also readily suited for dealing with sparse plans, based on existing libraries. In more general ROT problems, however, a specific library should be written for the sparse extension because null entries in the transport plan are not represented by null entries in the dual parameter space, so that tailored data structures and operations

for such matrices need to be coded. Therefore, we only implemented and will focus in our experiments on the non-sparse version of our methods.

Finally, although we implicitly assumed throughout that the entries of $\mathbf{p}$ and $\mathbf{q}$ are strictly comprised between 0 and 1 for theoretical issues, it is often possible in practice to deal explicitly with null or unit entries in the input distributions. Intuitively, no mass can be moved from or to a null entry, so the transport plans have null rows and columns for the corresponding null entries of $\mathbf{p}$ and $\mathbf{q}$, respectively. In the separable case, we can thus simply remove these entries, solve the reduced ROT problem, and reinsert the corresponding null entries in the rot mover's plan $\boldsymbol{\pi}^\star$. The same reasoning as for the sparse extension can be made to show that our two algorithms still hold with this strategy from a theoretical standpoint. In the non-separable case, however, this is not as straightforward again because the influences of the different entries of $\boldsymbol{\pi}^\star$ are interleaved through the regularizer ϕ. Nonetheless, as long as we have $[0,1)^{d \times d} \subset \operatorname{int}(\operatorname{dom} \phi)$ under assumptions (B), then we have $\Pi(\mathbf{p}, \mathbf{q}) \subset \operatorname{int}(\operatorname{dom} \phi)$ and we can apply NASA without modification. This is notably the case for the Mahalanobis distances whose domain is $\mathbb{R}^{d \times d}$. For a non-separable regularizer under assumptions (A), it is not easy to account for null entries because the constraint qualification $\Pi(\mathbf{p}, \mathbf{q}) \cap \operatorname{int}(\operatorname{dom} \phi) \neq \emptyset$ never holds due to mandatory null entries in the transport plans. Nevertheless, common regularizers under assumptions (A), including the ones used in this paper, are separable in general. Lastly, it is direct to cope with unit entries in $\mathbf{p}$ or $\mathbf{q}$ in all cases, since the transport polytope then reduces to a singleton, so that there is a unique transport plan $\mathbf{p}\mathbf{q}^\top$ which is the rot mover's plan.

5. Classical Regularizers and Divergences

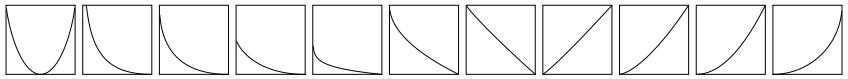
In this section, we discuss the specificities of the ASA (Algorithm 1) and NASA (Algorithm 3) methods to solve ROT problems for classical regularizers and associated divergences. We start with several separable regularizers under assumptions (A), based on the Boltzmann-Shannon entropy related to the Kullback-Leibler divergence (`BSKL`, Section 5.1), the Burg entropy related to the Itakura-Saito divergence (`BIS`, Section 5.2), and the Fermi-Dirac entropy related to a logistic loss function (`FDLOG`, Section 5.3), as well as the parametric families of β-potentials related to the β-divergences (`BETA`, Section 5.4). We then discuss the separable ℓ_p quasi-norms (`LPQN`, Section 5.5), which require a slight adaptation of assumptions (A). We also consider separable regularizers under assumptions (B) related to ℓ_p norms (`LPN`, Section 5.6), as well as the Euclidean norm related to the Euclidean distance (`EUC`, Section 5.7) and the Hellinger distance (`HELL`, Section 5.8). Finally, we study a non-separable regularizer under assumptions (B) via quadratic forms in relation to Mahalanobis distances (Section 5.9). We plot all separable regularizers in Figure 2. All regularizers and their corresponding divergences are also summed up in Table 2. Lastly, we provide in Table 3 the related terms based on derivatives that are needed to instantiate the separable versions of ASA (Algorithm 2) or NASA (Algorithm 4) accordingly.

5.1 Boltzmann-Shannon Entropy and Kullback-Leibler Divergence

Assumptions (A) hold for minus the Boltzmann-Shannon entropy $\pi \log \pi - \pi + 1$ associated to the Kullback-Leibler divergence. Hence, the ROT problem can be solved with the ASA scheme. In addition, the updates in the POCS technique can be written analytically, leading

$\phi(\pi)$ / $\phi(\boldsymbol{\pi})$	$B_\phi(\pi\|\xi)$ / $B_\phi(\boldsymbol{\pi}\|\boldsymbol{\xi})$	$\operatorname{dom}\phi$	$\operatorname{dom}\psi$
Boltzmann-Shannon entropy $\pi\log\pi-\pi+1$	*Kullback-Leibler divergence* $\pi\log\frac{\pi}{\xi}-\pi+\xi$	$\mathbb{R}_+$	$\mathbb{R}$
Burg entropy $\pi-\log\pi-1$	*Itakura-Saito divergence* $\frac{\pi}{\xi}-\log\frac{\pi}{\xi}-1$	$\mathbb{R}_{++}$	$(-\infty,1)$
Fermi-Dirac entropy $\pi\log\pi+(1-\pi)\log(1-\pi)$	*Logistic loss function* $\pi\log\frac{\pi}{\xi}+(1-\pi)\log\frac{1-\pi}{1-\xi}$	$[0,1]$	$\mathbb{R}$
β-potentials $(0<\beta<1)$ $\frac{1}{\beta(\beta-1)}(\pi^\beta-\beta\pi+\beta-1)$	*β-divergences* $\frac{1}{\beta(\beta-1)}(\pi^\beta+(\beta-1)\xi^\beta-\beta\pi\xi^{\beta-1})$	$\mathbb{R}_+$	$(-\infty,\frac{1}{1-\beta})$
ℓ_p quasi-norms $(0<p<1)$ $-\pi^p$	$-\pi^p+p\pi\xi^{p-1}-(p-1)\xi^p$	$\mathbb{R}_+$	$\mathbb{R}_{--}$
ℓ_p norms $(1<p<+\infty)$ $\lvert\pi\rvert^p$	$\lvert\pi\rvert^p-p\pi\ \operatorname{sgn}(\xi)\lvert\xi\rvert^{p-1}+(p-1)\lvert\xi\rvert^p$	$\mathbb{R}$	$\mathbb{R}$
Euclidean norm $\frac{1}{2}\pi^2$	*Euclidean distance* $\frac{1}{2}(\pi-\xi)^2$	$\mathbb{R}$	$\mathbb{R}$
Hellinger distance $-(1-\pi^2)^{\frac{1}{2}}$	$(1-\pi\xi)(1-\xi^2)^{-\frac{1}{2}}-(1-\pi^2)^{\frac{1}{2}}$	$[-1,1]$	$\mathbb{R}$
Quadratic forms $(\mathbf{P}\succ\mathbf{0})$ $\frac{1}{2}\operatorname{vec}(\boldsymbol{\pi})^\top\mathbf{P}\operatorname{vec}(\boldsymbol{\pi})$	*Mahalanobis distances* $\frac{1}{2}\operatorname{vec}(\boldsymbol{\pi}-\boldsymbol{\xi})^\top\mathbf{P}\operatorname{vec}(\boldsymbol{\pi}-\boldsymbol{\xi})$	$\mathbb{R}^{d\times d}$	$\mathbb{R}^{d\times d}$

Table 2: Convex regularizers and associated Bregman divergences.



—	$\beta=0$	$\beta=0.5$	$\beta=1$	$p=0.1$	$p=0.5$	$p=0.9$	$p=1.1$	$p=1.5$	$p=2$	—
FDLOG	BETA	BETA	BETA	LPQN	LPQN	LPQN	LPN	LPN	LPN	HELL
—	BIS	—	BSKL	—	—	—	—	—	EUC	—

Figure 2: Separable regularizers on $(0,1)$.

to the Sinkhorn-Knopp algorithm. Specifically, the two projections amount to normalizing in turn the rows and columns of $\boldsymbol{\pi}^\star$ so that they sum up to $\mathbf{p}$ and $\mathbf{q}$ respectively:

$$\boldsymbol{\pi}^\star \leftarrow \operatorname{diag}\left(\frac{\mathbf{p}}{\boldsymbol{\pi}^\star\mathbf{1}}\right)\boldsymbol{\pi}^\star \ , \tag{121}$$

$$\boldsymbol{\pi}^\star \leftarrow \boldsymbol{\pi}^\star\operatorname{diag}\left(\frac{\mathbf{q}}{\boldsymbol{\pi}^{\star\top}\mathbf{1}}\right) \ . \tag{122}$$

$\phi(\pi)$	$\phi'(\pi)$	$\psi'(\theta)$	$\psi''(\theta)$
Boltzmann-Shannon entropy $\pi\log\pi-\pi+1$	$\log\pi$	$\exp\theta$	$\exp\theta$
Burg entropy $\pi-\log\pi-1$	$1-\pi^{-1}$	$(1-\theta)^{-1}$	$(1-\theta)^{-2}$
Fermi-Dirac entropy $\pi\log\pi+(1-\pi)\log(1-\pi)$	$\log\frac{\pi}{1-\pi}$	$\frac{\exp\theta}{(1+\exp\theta)}$	$\frac{\exp\theta}{(1+\exp\theta)^2}$
β-potentials $(0<\beta<1)$ $\frac{1}{\beta(\beta-1)}(\pi^\beta-\beta\pi+\beta-1)$	$\frac{1}{\beta-1}(\pi^{\beta-1}-1)$	$((\beta-1)\theta+1)^{\frac{1}{\beta-1}}$	$((\beta-1)\theta+1)^{\frac{1}{\beta-1}-1}$
ℓ_p quasi-norms $(0<p<1)$ $-\pi^p$	$-p\pi^{p-1}$	$p^{-\frac{1}{p-1}}(-\theta)^{\frac{1}{p-1}}$	$-\frac{p^{-\frac{1}{p-1}}}{p-1}(-\theta)^{\frac{1}{p-1}-1}$
ℓ_p norms $(1<p<+\infty)$ $\lvert\pi\rvert^p$	$p\ \operatorname{sgn}(\pi)\lvert\pi\rvert^{p-1}$	$p^{-\frac{1}{p-1}}\ \operatorname{sgn}(\theta)\lvert\theta\rvert^{\frac{1}{p-1}}$	$\frac{p^{-\frac{1}{p-1}}}{p-1}\lvert\theta\rvert^{\frac{1}{p-1}-1}$
Euclidean norm $\frac{1}{2}\pi^2$	π	θ	1
Hellinger distance $-(1-\pi^2)^{\frac{1}{2}}$	$\pi(1-\pi^2)^{-\frac{1}{2}}$	$\theta(1+\theta^2)^{-\frac{1}{2}}$	$(1+\theta^2)^{-\frac{3}{2}}$

Table 3: Separable regularizers and related terms based on derivatives.

This can be optimized by remarking that the iterates $\boldsymbol{\pi}^{\star(k)}$ after each couple of projections verify:

$$\boldsymbol{\pi}^{\star(k)} = \operatorname{diag}(\mathbf{u}^{(k)})\boldsymbol{\xi}\operatorname{diag}(\mathbf{v}^{(k)}) \ , \tag{123}$$

where $\boldsymbol{\xi}=\exp(-\boldsymbol{\gamma}/\lambda)$, and vectors $\mathbf{u}^{(k)},\mathbf{v}^{(k)}$ satisfy the following recursion:

$$\mathbf{u}^{(k)} = \frac{\mathbf{p}}{\boldsymbol{\xi}\mathbf{v}^{(k-1)}} \ , \tag{124}$$

$$\mathbf{v}^{(k)} = \frac{\mathbf{q}}{\boldsymbol{\xi}^\top\mathbf{u}^{(k)}} \ , \tag{125}$$

with convention $\mathbf{v}^{(0)}=\mathbf{1}$. This allows a fast implementation by performing only matrix-vector multiplications using a fixed matrix $\boldsymbol{\xi}=\exp(-\boldsymbol{\gamma}/\lambda)$. We can further save one element-wise vector multiplication per update:

$$\mathbf{u} \leftarrow \frac{\mathbf{1}}{\operatorname{diag}\left(\frac{\mathbf{1}}{\mathbf{p}}\right)\boldsymbol{\xi}\,\mathbf{v}} \ , \tag{126}$$

$$\mathbf{v} \leftarrow \frac{\mathbf{1}}{\operatorname{diag}\left(\frac{\mathbf{1}}{\mathbf{q}}\right)\boldsymbol{\xi}^\top\mathbf{u}} \ , \tag{127}$$

where the matrices $\operatorname{diag}\left(\frac{\mathbf{1}}{\mathbf{p}}\right)\boldsymbol{\xi}$ and $\operatorname{diag}\left(\frac{\mathbf{1}}{\mathbf{q}}\right)\boldsymbol{\xi}^\top$ are precomputed and stored.

5.2 Burg Entropy and Itakura-Saito Divergence

Assumptions (A) also hold for minus the Burg entropy $\pi - \log\pi - 1$ associated to the Itakura-Saito divergence, so the ROT problem can be solved with the ASA scheme. Eventually, the Newton-Raphson steps to update the alternate projections in POCS technique can be written as follows:

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \frac{(1-\boldsymbol{\theta}^\star)^{-1}\,\mathbf{1} - \mathbf{p}}{(1-\boldsymbol{\theta}^\star)^{-2}\,\mathbf{1}}\,\mathbf{1}^\top \ , \tag{128}$$

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \mathbf{1}\,\frac{\mathbf{1}^\top(1-\boldsymbol{\theta}^\star)^{-1} - \mathbf{q}^\top}{\mathbf{1}^\top(1-\boldsymbol{\theta}^\star)^{-2}} \ . \tag{129}$$

Each step can be optimized by computing first an element-wise matrix inverse $(1-\boldsymbol{\theta}^\star)^{-1}$ for the numerator, and then performing an element-wise matrix multiplication of this matrix by itself to obtain a matrix for the denominator instead of applying an additional element-wise matrix power. Since ψ' is convex and strictly increasing with ψ'' positive everywhere, the convergence of the updates is guaranteed.

5.3 Fermi-Dirac Entropy and Logistic Loss Function

Assumptions (A) again hold for minus the Fermi-Dirac entropy $\pi\log\pi + (1-\pi)\log(1-\pi)$, also known as bit entropy, associated to a logistic loss function. The ROT problem can thus be solved with the ASA scheme, and the Newton-Raphson steps to update the alternate projections in the POCS technique can be written as follows:

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \frac{\frac{\exp\boldsymbol{\theta}^\star}{1+\exp\boldsymbol{\theta}^\star}\,\mathbf{1} - \mathbf{p}}{\frac{\exp\boldsymbol{\theta}^\star}{(1+\exp\boldsymbol{\theta}^\star)^2}\,\mathbf{1}}\,\mathbf{1}^\top \ , \tag{130}$$

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \mathbf{1}\,\frac{\frac{\exp\boldsymbol{\theta}^\star}{1+\exp\boldsymbol{\theta}^\star} - \mathbf{q}^\top}{\mathbf{1}^\top\frac{\exp\boldsymbol{\theta}^\star}{(1+\exp\boldsymbol{\theta}^\star)^2}} \ . \tag{131}$$

Each step can be optimized by storing first the element-wise matrix exponential $\exp\boldsymbol{\theta}^\star$, then applying an element-wise matrix division by the temporary matrix $1+\exp\boldsymbol{\theta}^\star$ to obtain a matrix for the numerator, and lastly performing an element-wise matrix division of these two matrices to obtain a matrix for the denominator and thus save an additional element-wise matrix power as well as several element-wise matrix exponentials. However, even if ψ' is strictly increasing with ψ'' positive everywhere, ψ' is neither convex nor concave and does not verify the necessary and sufficient condition (40) for global convergence of the Newton-Raphson method.

Nevertheless, ψ' is convex on $\mathbb{R}_-$ and concave on $\mathbb{R}_+$. It thus divides for a given $1 \leq i \leq d$, respectively $1 \leq j \leq d$, the real line into at most $d+1$ intervals $-\infty < \hat{\theta}_i^{(1)} \leq \hat{\theta}_i^{(2)} \leq \cdots \leq \hat{\theta}_i^{(d-1)} \leq \hat{\theta}_i^{(d)} < +\infty$, respectively $-\infty < \check{\theta}_j^{(1)} \leq \check{\theta}_j^{(2)} \leq \cdots \leq \check{\theta}_j^{(d-1)} \leq \check{\theta}_j^{(d)} < +\infty$, with the values $(\hat{\theta}_i^{(k)})_{1\leq k\leq d}$ from row i of $\overline{\boldsymbol{\theta}}$, respectively $(\check{\theta}_j^{(k)})_{1\leq k\leq d}$ from column j of $\overline{\boldsymbol{\theta}}$, sorted

in increasing order. On each of these intervals, the necessary and sufficient condition (40) is verified since we can decompose $f(\mu_i)$, respectively $g(\nu_j)$, as the sum of an increasing convex and an increasing concave function. Hence, we have global convergence on the interval that contains the solution. It is further possible to restrict the search to the two last intervals only. Indeed, we have $\sum_{j=1}^d \psi'(\overline{\theta}_{ij} - \hat{\theta}_i^{(d-1)}) \geq \psi'(\hat{\theta}_i^{(d-1)} - \hat{\theta}_i^{(d-1)}) + \psi'(\hat{\theta}_i^{(d)} - \hat{\theta}_i^{(d-1)}) \geq 2\psi'(0) = 1$, so that $\hat{\theta}_i^{(d-1)} < \mu_i < +\infty$. Similarly, we have $\sum_{i=1}^d \psi'(\overline{\theta}_{ij} - \check{\theta}_j^{(d-1)}) \geq \psi'(\check{\theta}_j^{(d-1)} - \check{\theta}_j^{(d-1)}) + \psi'(\check{\theta}_j^{(d)} - \check{\theta}_j^{(d-1)}) \geq 2\psi'(0) = 1$, so that $\check{\theta}_j^{(d-1)} < \nu_j < +\infty$. As a result, it suffices to initialize μ_i with $\hat{\theta}_i = \hat{\theta}_i^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq j\leq d}$, respectively ν_j with $\check{\theta}_j = \check{\theta}_j^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq i\leq d}$, to guarantee convergence of the updates.

5.4 β-potentials and β-divergences

Assumptions (A) hold for the β-potentials $(\pi^\beta - \beta\pi + \beta - 1)/(\beta(\beta-1))$ with $0 < \beta < 1$, associated to the so-called β-divergences. Hence, the ROT problem can be solved with the ASA scheme, and the Newton-Raphson steps to update the alternate projections in the POCS technique can be written as follows:

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \frac{((\beta-1)\boldsymbol{\theta}^\star + 1)^{\frac{1}{\beta-1}}\,\mathbf{1} - \mathbf{p}}{((\beta-1)\boldsymbol{\theta}^\star + 1)^{\frac{1}{\beta-1}-1}\,\mathbf{1}}\,\mathbf{1}^\top \ , \tag{132}$$

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star - \mathbf{1}\,\frac{\mathbf{1}^\top((\beta-1)\boldsymbol{\theta}^\star + 1)^{\frac{1}{\beta-1}} - \mathbf{q}^\top}{\mathbf{1}^\top((\beta-1)\boldsymbol{\theta}^\star + 1)^{\frac{1}{\beta-1}-1}} \ . \tag{133}$$

Each step can be optimized by computing first the temporary matrix $(\beta-1)\boldsymbol{\theta}^\star + 1$, then applying an element-wise matrix power of $1/(\beta-1) - 1$ to this temporary matrix to obtain a matrix for the denominator, and lastly performing an element-wise matrix multiplication of these two matrices to obtain a matrix for the numerator and thus save one element-wise matrix power. Since ψ' is convex and strictly increasing with ψ'' positive, the convergence of the updates is guaranteed.

Interestingly, the regularizer tends to minus the Burg and Boltzmann-Shannon entropies in the limit $\beta = 0$ and $\beta = 1$, respectively. Therefore, the β-divergences interpolate between the Itakura-Saito and Kullback-Leibler divergences. We finally remark that the regularizer can also be defined for other values of the parameter β using the same formula, but do not verify assumptions (A) for these values.

5.5 ℓ_p quasi-norms

Considering regularizers $-\pi^p$ with $0 < p < 1$, all assumptions (A) are verified except from (A5) since $\mathbb{R}_-^{d\times d} \not\subset \operatorname{dom}\psi = \mathbb{R}_{--}^{d\times d}$. Hence, our primal formulation does not hold here because $\mathbf{0} \notin \operatorname{dom}\nabla\psi$. However, it is straightforward to check that our dual formulation for ROT problems with the ASA scheme can still be applied as long as the cost matrix $\boldsymbol{\gamma}$ does not have null entries so that $-\boldsymbol{\gamma}/\lambda \in \operatorname{dom}\nabla\psi$. Eventually, the Newton-Raphson steps to update the alternate projections in the POCS technique can be written as follows:

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star + \frac{(-\boldsymbol{\theta}^\star)^{\frac{1}{p-1}}\,\mathbf{1} - p^{\frac{1}{p-1}}\,\mathbf{p}}{\frac{1}{p-1}(-\boldsymbol{\theta}^\star)^{\frac{1}{p-1}-1}\,\mathbf{1}}\,\mathbf{1}^\top \ , \tag{134}$$

$$\boldsymbol{\theta}^\star \leftarrow \boldsymbol{\theta}^\star + \mathbf{1}\,\frac{\mathbf{1}^\top\,(-\boldsymbol{\theta}^\star)^{\frac{1}{p-1}} - p^{\frac{1}{p-1}}\,\mathbf{q}^\top}{\frac{1}{p-1}\,\mathbf{1}^\top\,(-\boldsymbol{\theta}^\star)^{\frac{1}{p-1}-1}} \enspace . \tag{135}$$

Each step can be optimized by computing first the temporary matrix $-\boldsymbol{\theta}^\star$, then applying an element-wise matrix power of $1/(p-1)-1$ to obtain a matrix for the denominator, and lastly performing an element-wise matrix multiplication of these two matrices to obtain a matrix for the numerator and thus save one element-wise matrix power. Since ψ' is convex and strictly increasing with ψ'' positive everywhere, the convergence of the updates is guaranteed.

5.6 ℓ_p norms

Assumptions (B) hold for the ℓ_p norms $|\pi|^p$ with $1 < p < +\infty$, so the ROT problem can be solved with the NASA scheme. For $p \neq 2$, the Newton-Raphson steps to update the alternate projections in Dykstra's algorithm can be written as follows:

$$\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} + \frac{\left\{\operatorname{sgn}(\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top) \odot |\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top|^{\frac{1}{p-1}}\right\}\mathbf{1} - p^{\frac{1}{p-1}}\,\mathbf{p}}{\frac{1}{p-1}|\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top|^{\frac{1}{p-1}-1}\,\mathbf{1}} \enspace , \tag{136}$$

$$\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \frac{\mathbf{1}^\top\left\{\operatorname{sgn}(\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top) \odot |\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top|^{\frac{1}{p-1}}\right\} - p^{\frac{1}{p-1}}\,\mathbf{q}^\top}{\frac{1}{p-1}\,\mathbf{1}^\top\,|\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top|^{\frac{1}{p-1}-1}} \enspace . \tag{137}$$

Denoting $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top$ or $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top$ in the respective updates, each step can be optimized by computing first the temporary matrix $|\overline{\boldsymbol{\theta}}|$, then applying an element-wise matrix power of $1/(p-1)-1$ to obtain a matrix for the denominator, and lastly performing an element-wise matrix multiplication of these two matrices and of $\operatorname{sgn}\overline{\boldsymbol{\theta}}$ to obtain a matrix for the numerator and thus save one element-wise matrix power as well as several vector replications and matrix subtractions. However, even if ψ' is strictly increasing with $\psi'' > 0$ on $\mathbb{R}^*$, ψ' is neither convex nor concave and does not verify the necessary and sufficient condition (40) for global convergence of the Newton-Raphson method. Moreover, ψ'' vanishes at 0 for $p < 2$, and ψ' is not differentiable at 0 for $p > 2$.

Nevertheless, ψ' is concave on $\mathbb{R}_-$ and convex on $\mathbb{R}_+$ for $p < 2$, as well as convex on $\mathbb{R}_-$ and concave on $\mathbb{R}_+$ for $p > 2$. It thus divides for a given $1 \leq i \leq d$, respectively $1 \leq j \leq d$, the real line into at most $d+1$ intervals $-\infty < \hat{\theta}_i^{(1)} \leq \hat{\theta}_i^{(2)} \leq \cdots \leq \hat{\theta}_i^{(d-1)} \leq \hat{\theta}_i^{(d)} < +\infty$, respectively $-\infty < \check{\theta}_j^{(1)} \leq \check{\theta}_j^{(2)} \leq \cdots \leq \check{\theta}_j^{(d-1)} \leq \check{\theta}_j^{(d)} < +\infty$, with the values $(\hat{\theta}_i^{(k)})_{1\leq k\leq d}$ from row i of $\overline{\boldsymbol{\theta}}$, respectively $(\check{\theta}_j^{(k)})_{1\leq k\leq d}$ from column j of $\overline{\boldsymbol{\theta}}$, sorted in increasing order. The necessary and sufficient condition (40) is verified on the interior of each of these intervals since we can decompose $f(\mu_i)$, respectively $g(\nu_j)$, as the sum of an increasing convex and an increasing concave function. Hence, we have global convergence on the interior of the interval that contains the solution. In both cases, we must remove the finite endpoints to ensure differentiability of ψ' and positivity of ψ''. It is also further possible to prune the last interval from the search. Indeed, we have $\sum_{j=1}^d \psi'(\overline{\theta}_{ij} - \hat{\theta}_i^{(d)}) \leq \sum_{j=1}^d \psi'(0) = 0$, so that $\mu_i < \hat{\theta}_i = \hat{\theta}_i^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq j\leq d}$. Similarly, we have $\sum_{i=1}^d \psi'(\overline{\theta}_{ij} - \check{\theta}_j^{(d)}) \leq \sum_{i=1}^d \psi'(0) = 0$, so that $\nu_j < \check{\theta}_j = \check{\theta}_j^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq i\leq d}$. Lastly, we can restrict the first interval with a finite lower bound instead. Indeed, we have $\sum_{j=1}^d \psi'(\overline{\theta}_{ij} - \hat{\theta}_i^{(1)} + \phi'(p_i/d)) \geq \sum_{j=1}^d \psi'(\phi'(p_i/d)) = p_i$, so that $\mu_i \geq \hat{\theta}_i^{(1)} - \phi'(p_i/d)$. Similarly, we have $\sum_{i=1}^d \psi'(\overline{\theta}_{ij} - \check{\theta}_j^{(1)} + \phi'(q_j/d)) \geq \sum_{i=1}^d \psi'(\phi'(q_j/d)) = q_j$, so that $\nu_j \geq \check{\theta}_j^{(1)} - \phi'(q_j/d)$. As a result, we can perform at most d binary searches in parallel to determine within which of the remaining bounded intervals the solutions μ_i, respectively ν_j, lie. Initialization is then done with the midpoint to guarantee convergence of the updates. A given search thus requires a worst-case logarithmic number of tests, each of which requires a linear number of operations, for a total complexity in $O(d^2 \log d)$ instead of $O(d^2)$ if no such binary search were needed.

Now for $p = 2$, the regularizer specializes to the Euclidean norm, leading to the squared Euclidean distance as the associated divergence. In addition, the formula for ψ'' still holds with the convention $0^0 = 1$, and ψ'' is actually constant equal to $1/2$. Eventually, the projections can be written in closed form, and we can resort to the analytical algorithm derived in the next example specifically for the Euclidean distance, after doubling the penalty λ to account for the regularizer being halved.

5.7 Euclidean Norm and Euclidean Distance

Assumptions (B) hold for half the Euclidean norm $\pi^2/2$ associated to half the squared Euclidean distance. Therefore, the ROT problem can be solved with the NASA scheme, where Dykstra's algorithm can actually be written in closed form. Specifically, the non-negative projection reduces to:

$$\boldsymbol{\pi}^\star \leftarrow \max\{\mathbf{0}, \widetilde{\boldsymbol{\pi}}\} \enspace , \tag{138}$$

and is interleaved with the scaling projections which amount to offsetting the rows and columns of $\widetilde{\boldsymbol{\pi}}$ by an amount such that the rows and columns of $\boldsymbol{\pi}^\star$ sum up to $\mathbf{p}$ and $\mathbf{q}$ respectively:

$$\widetilde{\boldsymbol{\pi}} \leftarrow \widetilde{\boldsymbol{\pi}} - \frac{1}{d}(\boldsymbol{\pi}^\star\mathbf{1} - \mathbf{p})\,\mathbf{1}^\top \enspace , \tag{139}$$

$$\widetilde{\boldsymbol{\pi}} \leftarrow \widetilde{\boldsymbol{\pi}} - \frac{1}{d}\,\mathbf{1}\,(\mathbf{1}^\top\boldsymbol{\pi}^\star - \mathbf{q}^\top) \enspace . \tag{140}$$

As a remark, we notice that half the squared Euclidean distance can be seen as a β-divergence using the provided formula for $\beta = 2$. However, the β-divergence generated is not of Legendre type because the domain is restricted to $\mathbb{R}_+$, whereas it could actually be extended to $\mathbb{R}$ so that the regularizer would then be of Legendre type. This is why we fall under assumptions (B) rather than assumptions (A) in this case.

5.8 Hellinger Distance

Assumptions (B) hold for the regularizer $-(1-\pi^2)^{\frac{1}{2}}$ akin to a Hellinger distance. Hence, the ROT problem can be solved with the NASA scheme, and the Newton-Raphson steps to

update the alternate projections in Dykstra's algorithm can be written as follows:

$$\boldsymbol{\tau} \leftarrow \boldsymbol{\tau} + \frac{\left\{(\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top) \odot \left(1 + (\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top)^2\right)^{-\frac{1}{2}}\right\}\mathbf{1} - \mathbf{p}}{\left(1 + (\boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top)^2\right)^{-\frac{3}{2}}\mathbf{1}}\ , \tag{141}$$

$$\boldsymbol{\sigma} \leftarrow \boldsymbol{\sigma} + \frac{\mathbf{1}^\top\left\{(\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top) \odot \left(1 + (\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top)^2\right)^{-\frac{1}{2}}\right\} - \mathbf{q}^\top}{\mathbf{1}^\top\left(1 + (\boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top)^2\right)^{-\frac{3}{2}}}\ . \tag{142}$$

Denoting $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - \boldsymbol{\tau}\mathbf{1}^\top$ or $\overline{\boldsymbol{\theta}} = \boldsymbol{\theta}^\star - \mathbf{1}\boldsymbol{\sigma}^\top$ in the respective updates, each step can be optimized by computing first the temporary matrix $1/(1+\overline{\boldsymbol{\theta}}^2)$, then applying an element-wise matrix square root to this temporary matrix, performing an element-wise matrix multiplication of these two matrices to obtain a matrix for the denominator, and lastly an element-wise matrix multiplication of the temporary matrix with $\overline{\boldsymbol{\theta}}$ to obtain a matrix for the numerator and thus save one element-wise matrix power as well as several vector replications and matrix subtractions. However, even if ψ' is strictly increasing with ψ'' positive everywhere, ψ' is neither convex nor concave and does not verify the necessary and sufficient condition (40) for global convergence of the Newton-Raphson method.

Nevertheless, ψ' is convex on $\mathbb{R}_-$ and concave on $\mathbb{R}_+$. It thus divides for a given $1 \leq i \leq d$, respectively $1 \leq j \leq d$, the real line into at most $d+1$ intervals $-\infty < \hat{\theta}_i^{(1)} \leq \hat{\theta}_i^{(2)} \leq \cdots \leq \hat{\theta}_i^{(d-1)} \leq \hat{\theta}_i^{(d)} < +\infty$, respectively $-\infty < \check{\theta}_j^{(1)} \leq \check{\theta}_j^{(2)} \leq \cdots \leq \check{\theta}_j^{(d-1)} \leq \check{\theta}_j^{(d)} < +\infty$, with the values $(\hat{\theta}_i^{(k)})_{1\leq k\leq d}$ from row i of $\overline{\boldsymbol{\theta}}$, respectively $(\check{\theta}_j^{(k)})_{1\leq k\leq d}$ from column j of $\overline{\boldsymbol{\theta}}$, sorted in increasing order. On each of these intervals, the necessary and sufficient condition (40) is verified since we can decompose $f(\mu_i)$, respectively $g(\nu_j)$, as the sum of an increasing convex and an increasing concave function. Hence, we have global convergence on the interval that contains the solution. It is further possible to prune the last interval from the search. Indeed, we have $\sum_{j=1}^d \psi'(\overline{\theta}_{ij} - \hat{\theta}_i^{(d)}) \leq \sum_{j=1}^d \psi'(0) = 0$, so that $\mu_i < \hat{\theta}_i = \hat{\theta}_i^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq j\leq d}$. Similarly, we have $\sum_{i=1}^d \psi'(\overline{\theta}_{ij} - \check{\theta}_j^{(d)}) \leq \sum_{i=1}^d \psi'(0) = 0$, so that $\nu_j < \check{\theta}_j = \check{\theta}_j^{(d)} = \max\{\overline{\theta}_{ij}\}_{1\leq i\leq d}$. Lastly, we can restrict the first interval with a finite lower bound instead. Indeed, we have $\sum_{j=1}^d \psi'(\overline{\theta}_{ij} - \hat{\theta}_i^{(1)} + \phi'(p_i/d)) \geq \sum_{j=1}^d \psi'(\phi'(p_i/d)) = p_i$, so that $\mu_i \geq \hat{\theta}_i^{(1)} - \phi'(p_i/d)$. Similarly, we have $\sum_{i=1}^d \psi'(\overline{\theta}_{ij} - \check{\theta}_j^{(1)} + \phi'(q_j/d)) \geq \sum_{i=1}^d \psi'(\phi'(q_j/d)) = q_j$, so that $\nu_j \geq \check{\theta}_j^{(1)} - \phi'(q_j/d)$. As a result, we can perform d binary searches in parallel to determine within which of the remaining intervals the solutions μ_i, respectively ν_j, lie. Initialization is then done with the midpoint to guarantee convergence of the updates. A given search requires a worst-case logarithmic number of tests, each of which requires a linear number of operations, for a total complexity in $O(d^2 \log d)$ instead of $O(d^2)$ if no such binary search were needed.

5.9 Quadratic Forms and Mahalanobis Distances

Assumptions (B) hold for the quadratic forms $(1/2)\,\mathrm{vec}(\boldsymbol{\pi})^\top \mathbf{P}\,\mathrm{vec}(\boldsymbol{\pi})$ with positive-definite matrix $\mathbf{P} \in \mathbb{R}^{d^2\times d^2}$, associated to the Mahalanobis distances, so the ROT problem can be

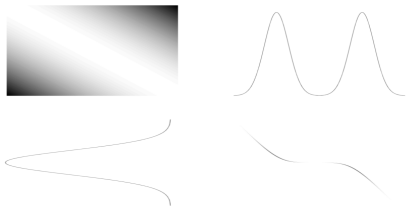


Figure 3: Earth mover's plan $\boldsymbol{\pi}^\star$ for the cost matrix $\boldsymbol{\gamma}$ and input distributions $\mathbf{p}, \mathbf{q}$.

solved with the NASA scheme. For a diagonal matrix $\mathbf{P}$, the regularizer is separable and the Newton-Raphson steps to update the alternate projections in Dykstra's algorithm are similar to that for the Euclidean distance with appropriate weights. For a non-diagonal matrix $\mathbf{P}$, however, the regularizer is not separable anymore and we must resort to the generic NASA scheme.

In this general case, the scaling projections amount to convex quadratic programs with linear equality constraints. They can be solved using classical techniques such as the range-space and null-space approaches, Krylov subspace methods or active set strategies. The non-negative projection reduces to a convex quadratic program with a linear inequality constraint. It can be solved elegantly with an iterative algorithm for non-negative quadratic programming proposed by Sha et al. (2007) using multiplicative updates with a complexity in $O(d^4)$. All in all, we recommend using a sparse matrix $\mathbf{P}$ with a block-diagonal structure and an order of magnitude of d^2 non-null entries, so as to obtain a quadratic instead of quartic empirical complexity.

6. Experimental Results

In this section, we present the results of our methods on different experiments. We first design an synthetic test to showcase the behavior of different regularizers and penalties on the output solutions or computational times (Section 6.1). We then consider a pattern recognition application to audio scene classification on a real-world dataset (Section 6.2).

6.1 Synthetic Data

We start by visualizing the effects of different regularizers ϕ and varying penalties λ on synthetic data. For the input distributions, we discretize and normalize continuous densities on a uniform grid $(x_i)_{1\leq i\leq d}$ of $[0,1]$ with dimension $d = 256$. We use for $\mathbf{p}$ a univariate normal with mean 0.5 and variance 0.2, and for $\mathbf{q}$ a mixture of two normals with equal weights, respective means 0.25 and 0.75, and same variance 0.1. We set the cost matrix $\boldsymbol{\gamma}$ as the squared Euclidean distance $\gamma_{ij} = (x_i - x_j)^2$ on the grid. The input distributions (bottom left and top right), cost matrix (top left) and unique earth mover's plan (bottom right) computed for classical OT using the solver of Rubner et al. (2000) with standard settings, are shown in Figure 3.

We test all separable regularizers ϕ introduced in Section 5. Because these regularizers have different ranges in the sensible values of the rot mover's plans $\boldsymbol{\pi}^\star$, we manually tune the penalties λ so that they feature similar amounts of regularization. For ease of comparison, we set $\lambda = \overline{\lambda}\,\lambda'$, with $\overline{\lambda}$ constant for each ϕ, and λ' varying similarly for all ϕ. The limit case when λ tends to infinity is simply obtained by setting $\boldsymbol{\gamma}/\lambda = \mathbf{0}$ in the algorithms, except from ℓ_p quasi-norms for which we use $\lambda = 10^{10}$. The null values of $\boldsymbol{\gamma}$ are also fixed to 10^{-12} for ℓ_p quasi-norms. We do not limit the number of iterations in the different algorithms, and use a small tolerance of 10^{-8} for convergence with the ℓ_∞ norm on the marginal difference checked after each iteration as a termination criterion.

The rot mover's plans obtained for ROT for $d = 256$ with the different regularizers and penalties are visualized in Figure 4. We first observe that all rot mover's plans converge to the earth mover's plan for low values of the penalty as shown theoretically in Property 9. Nevertheless, the rot mover's plans exhibit different shapes depending on the regularizers for intermediary and large values of the penalty. In the limit when the penalty grows to infinity, we obtain the transport plan with minimal Bregman information as shown theoretically in Property 7. In particular, this leads to $\mathbf{p}\mathbf{q}^\top$ with an ellipsoidal shape for `BSKL` (Boltzmann-Shannon entropy and Kullback-Leibler divergence), meaning that the mass is relatively spread among neighbor bins. The same pattern is observed for `FDLOG` (Fermi-Dirac entropy and logistic loss function), which can be explained in this synthetic example by the rot mover's plans having low values and the two regularizers being equivalent up to a constant in the neighborhood of zero. The profile gets more rectangular for `BIS` (Burg entropy and Itakura-Saito divergence), implying that the mass is even more spread across the different bins. Using an intermediary value $\beta = 0.5$ in `BETA` (β-potentials and β-divergences) allows the interpolation between these two limits of a rectangle for $\beta = 1$ and an ellipsoid for $\beta = 0$, so that the parameter β actually helps to control the spread of mass in the regularization. We observe similar results for `LPQN` (ℓ_p quasi-norms) with an ellipsoid for $p = 0.9$, a rectangle for $p = 0.1$, and a shape in between for $p = 0.5$. When the power parameter further increases in `LPN` (ℓ_p norms), we obtain new shapes that feature less spread of mass. These shapes for $p = 1.1$ and $p = 1.5$ now interpolate up to a lozenge for $p = 2$ in `EUC` (Euclidean norm and Euclidean distance), so that the parameter p also provides control on the spread of mass. A similar diamond profile is obtained for `HELL` (Hellinger distance), which is due again to the rot mover's plans having low values and the two regularizers being equivalent up to a constant in the neighborhood of zero. Lastly, we remark that varying the penalty between the two extremes allows a smooth interpolation of the earth mover's plan and optimal plan with minimal Bregman information, while keeping similar shapes and effects in terms of spreading of mass.

We next report in Table 4 the computational times required to reach convergence for the different regularizers and penalties. As a stopping criterion, we use the relative variation with tolerance 10^{-2} in ℓ_2 norm for the main loop of alternate Bregman projections, and the absolute variation with tolerance 10^{-5} in ℓ_2 norm for the auxiliary loops of the Newton-Raphson method. We use the same synthetic data as above but also vary the dimension d to assess its influence on speed. As already observed specifically for Sinkhorn distances (Cuturi, 2013), computing ROT distances is faster for important regularization with larger values of λ. The regularizers under assumptions (A) do not require the extra projections onto the non-negative orthant, and thus intuitively require less computational effort than the ones

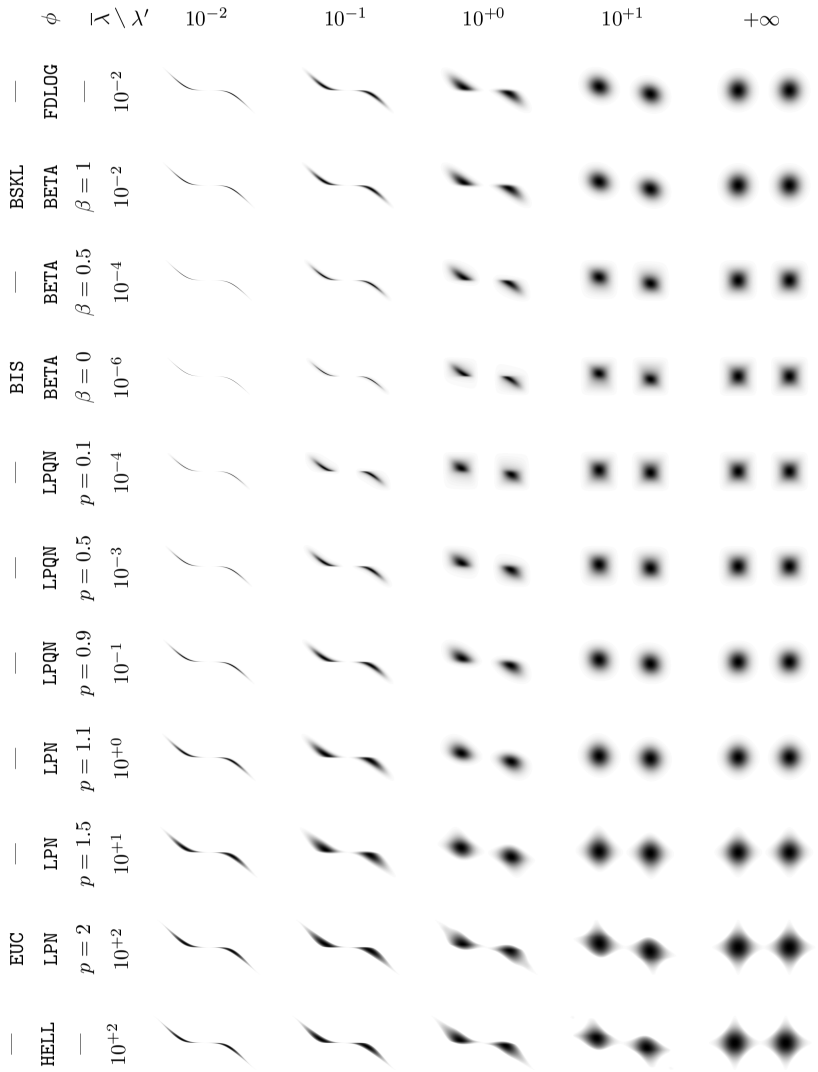


Figure 4: Rot mover's plans $\boldsymbol{\pi}^\star$ for different regularizers ϕ and penalties $\lambda = \overline{\lambda}\,\lambda'$.

that verify assumptions (B). In addition, we notice that when the projections have closed-form expressions, the algorithms are also faster. The results further illustrate the influence of the data dimension d and the difference between ROT and classical OT performances. For a low dimension d, the RMD is competitive with EMD in his historical implementation `EMD OLD` (Rubner et al., 2000). The super-cubic complexity of the EMD with `EMD OLD` becomes prohibitive as the data dimension increases in contrast to the RMD which scales better. It should nevertheless be underlined that for reasonable dimensions, fast computation of the EMD can be obtained with a more recent, optimized implementation of the network simplex solver `EMD NEW` (Bonneel et al., 2011). For higher dimensions, the super-cubic complexity makes `EMD NEW` less attractive, though it stays competitive with the RMD under a dimension $d = 4096$.

As a consequence, a numerical alternative to our algorithms for solving ROT problems with reasonable dimensions is to rely on conditional gradient methods similar to (Ferradans et al., 2014). Indeed, such methods imply the iterative resolution of linearized ROT problems, that can be reformulated as EMD problems and therefore be solved with the fast network simplex approach (Bonneel et al., 2011). Lastly, for a fair interpretation of the above timing results, we must mention that the two EMD schemes tested were run under MATLAB from native C/C++ implementations[1,2] via compiled MEX files[3,4]. Hence, these EMD codes are quite optimized in comparison to our pure MATLAB prototype codes[5] for the RMD. It is thus plausible that optimized C/C++ implementations of our algorithms would be even more competitive in this context.

6.2 Audio Classification

We now assess our methods in the context of audio classification, and specifically address the task of acoustic scene classification where the goal is to assign a test recording to one of predefined classes that characterizes the environment in which it was captured. We consider the framework of the DCASE 2016 IEEE AASP challenge with the TUT Acoustic Scenes 2016 database (Mesaros et al., 2016). The data set consists of audio recordings at 44.1 kHz sampling rate and 24-bit resolution. The metadata contains ground-truth annotations on the type of acoustic scene for all files, with a total of 15 classes: home, office, library, café/restaurant, grocery store, city center, residential area, park, forest path, beach, car, train, bus, tram, metro station. The audio material is cut into 30-second segments, and is split into two subsets of 75%–25% containing respectively 78–26 segments per class for development and evaluation, resulting in a total of 1170–390 files for training and testing. A 4-fold cross-validation setup is given with the training set. The classification accuracy, that is, the number of correctly classified segments among the total number of segments, is used as a score to evaluate systems.

A baseline system is also provided with the database for comparison. This system is based on Mel-frequency cepstral coefficient (MFCC) timbral features with Gaussian mixture model (GMM) classification. One GMM with diagonal covariance matrix is learned per class

[1] `http://robotics.stanford.edu/~rubner/emd/default.htm`
[2] `http://liris.cnrs.fr/~nbonneel/FastTransport/`
[3] `https://github.com/francopestilli/life/tree/master/external/emd`
[4] `https://arolet.github.io/code/`
[5] `https://www.math.u-bordeaux.fr/~npapadak/GOTMI/codes.php`

Algorithm	— ϕ —	β/p	$\overline{\lambda}/\lambda'$	d	— 128 —			— 256 —			— 512 —		
					10^{-2}	10^{-1}	10^{+0}	10^{-2}	10^{-1}	10^{+0}	10^{-2}	10^{-1}	10^{+0}
RMD	— FDLOG	—	10^{-2}		0.366	0.079	0.044	1.091	0.311	0.116	1.865	0.571	0.273
RMD	BSKL BETA	1.00	10^{-2}		0.105	0.013	0.008	0.259	0.055	0.017	0.680	0.100	0.038
RMD	— BETA	0.50	10^{-4}		0.971	0.102	0.044	1.922	0.251	0.147	3.526	1.339	0.281
RMD	BIS BETA	0.00	10^{-6}		0.916	0.106	0.019	1.466	0.108	0.053	2.598	0.398	0.096
RMD	— LPQN	0.10	10^{-4}		0.968	0.068	0.055	0.732	0.173	0.152	0.416	0.309	0.305
RMD	— LPQN	0.50	10^{-3}		0.404	0.057	0.042	0.778	0.163	0.160	0.780	0.305	0.304
RMD	— LPQN	0.90	10^{-1}		0.226	0.047	0.040	0.751	0.178	0.131	1.110	2.492	0.214
RMD	— LPN	1.10	10^{+0}		1.570	0.349	0.148	5.941	1.557	0.492	6.357	0.293	0.926
RMD	— LPN	1.50	10^{+1}		0.399	0.099	0.053	1.170	0.474	0.166	6.688	2.163	0.532
RMD	EUC LPN	2.00	10^{+2}		0.074	0.043	0.043	0.253	0.240	0.237	7.308	3.190	0.966
RMD	— HELL	—	10^{+2}		0.197	0.097	0.087	0.429	0.316	0.299	5.570	1.826	0.823
EMD OLD	— —	—	—		— 0.231 —			— 1.912 —			— 10.95 —		
EMD NEW	— —	—	—		— 0.003 —			— 0.011 —			— 0.076 —		

Algorithm	— ϕ —	β/p	$\overline{\lambda}/\lambda'$	d	— 1024 —			— 2048 —			— 4096 —		
					10^{-2}	10^{-1}	10^{+0}	10^{-2}	10^{-1}	10^{+0}	10^{-2}	10^{-1}	10^{+0}
RMD	— FDLOG	—	10^{-2}		4.156	3.517	1.410	15.85	9.663	5.109	54.19	33.87	17.24
RMD	BSKL BETA	1.00	10^{-2}		2.992	0.705	0.192	13.42	1.923	0.630	49.95	7.074	2.548
RMD	— BETA	0.50	10^{-4}		8.015	2.769	0.888	42.95	7.538	3.557	101.3	21.13	10.86
RMD	BIS BETA	0.00	10^{-6}		4.439	0.777	0.550	6.590	3.262	2.218	41.80	12.96	6.742
RMD	— LPQN	0.10	10^{-4}		4.068	2.174	1.291	6.962	4.890	4.334	51.15	15.86	14.02
RMD	— LPQN	0.50	10^{-3}		7.819	4.198	1.314	26.34	6.129	4.301	53.74	15.65	11.98
RMD	— LPQN	0.90	10^{-1}		3.584	2.264	1.054	13.80	4.571	3.285	43.83	14.22	11.51
RMD	— LPN	1.10	10^{+0}		9.110	4.924	1.956	38.98	16.47	8.400	145.6	65.95	32.82
RMD	— LPN	1.50	10^{+1}		18.97	9.509	2.539	61.92	20.41	9.314	236.6	77.87	45.94
RMD	EUC LPN	2.00	10^{+2}		11.90	5.805	2.161	31.43	14.22	4.906	117.1	50.88	27.67
RMD	— HELL	—	10^{+2}		18.22	6.629	3.456	35.45	20.03	7.199	205.0	48.42	31.75
EMD OLD	— —	—	—		— 85.56 —			— 482.2 —			— $+\infty$ —		
EMD NEW	— —	—	—		— 0.482 —			— 2.760 —			— 13.23 —		

Table 4: Computational times in seconds required to reach convergence for different regularizers ϕ and penalties $\lambda = \overline{\lambda}\,\lambda'$, with varying dimensions d.

by expectation-maximization (EM), after concatenating and normalizing in mean and variance the extracted MFCCs from the training segments in that class. A test file is assigned to the class whose trained GMM leads to maximum likelihood for the extracted MFCCs for that file, where the MFCCs are considered as independent samples and normalized with the learned mean and variance for the respective classes. The baseline system is ran with its default parameters: 40 ms frame size, 20 ms hop size, 60-dimensional MFCCs comprising 20 static (including energy) plus 20 delta and 20 acceleration coefficients extracted with

standard settings in RASTAMAT, 16 GMM components learned with standard settings in VOICEBOX.

Since MFCCs potentially take negative values, OT tools cannot be applied directly to this kind of features. Therefore, the common approach is to compute OT appropriately on GMMs estimated from MFCCs instead. Our proposed system follows this principle, and is implemented in the very same pipeline as the baseline for a fair comparison, with the following differences. One GMM is learned by EM for each training segment instead of class. Any normalization on the MFCCs per class is thus removed. Since less components are typically required to model one segment compared to one class, the spurious GMM components are further discarded as post-processing by keeping only those with weight and variances all greater than 10^{-2}. Instead of applying a GMM classifier, all individual models are exploited to train a support vector machine (SVM) classifier. An exponential kernel for the SVM is designed by introducing a distance between two mixtures P, Q based on the RMD as follows:

$$\kappa(P,Q) = \exp(-d_{\boldsymbol{\gamma},\lambda,\phi}(\boldsymbol{\omega},\boldsymbol{\upsilon})/\tau) \ , \tag{143}$$

where the exponential decay rate $\tau > 0$ is a kernel parameter, and $\boldsymbol{\omega}, \boldsymbol{\upsilon} \in \Sigma_d$ are the respective weights of the $d = 16$ (or less) components for the two GMMs P, Q. The cost matrix $\boldsymbol{\gamma} \in \mathbb{R}_+^{d\times d}$ depends on P, Q and is the square root of a symmetrized Kullback-Leibler divergence, called the Jeffrey divergence, between the pairwise Gaussian components:

$$\gamma_{ij} = \sqrt{\frac{1}{4}\sum_{k=1}^{l} \frac{(\sigma_{ik}^2 - \varsigma_{jk}^2)^2 + (\sigma_{ik}^2 + \varsigma_{jk}^2)(\mu_{ik} - \nu_{jk})^2}{\sigma_{ik}^2 \varsigma_{jk}^2}} \ , \tag{144}$$

where $\boldsymbol{\mu}_i$ and $\boldsymbol{\sigma}_i^2$, respectively $\boldsymbol{\nu}_j$ and $\boldsymbol{\varsigma}_j^2$, are the means and variances of the $l = 60$ MFCC features for component i in the first mixture P, respectively component j in the second mixture Q. The SVM classifier is implemented with standard settings in LIBSVM, and requires an additional soft-margin parameter $C > 0$ to be tuned. Notice that, even if the kernel is not positive-definite, LIBSVM is still able to provide a relevant classification by guaranteeing convergence to a stationary point (Lin and Lin, 2003; Haasdonk, 2005; Alabdulmohsin et al., 2014). All separable regularizers ϕ from Section 5 with different penalties $\lambda > 0$ are tested for the RMD in comparison to the EMD. The two distances between $\mathbf{p}, \mathbf{q}$ and $\mathbf{q}, \mathbf{p}$ with cost matrix $\boldsymbol{\gamma}$ transposed are computed and averaged, so as to remove any asymmetry due to practical issues. The number of iterations is limited to 100 for the main loop of the algorithm and to 10 for the auxiliary loops of the Newton-Raphson method, and the tolerance is set to 10^{-6} in all loops for convergence with the ℓ_∞ norm on the marginal difference checked after each iteration as a termination criterion. The parameters τ, $C \in 10^{\{-1,+0,+1,+2\}}$ and penalty $\lambda \in \Lambda$, where Λ is a manually chosen set of four successive powers of ten depending on the range of the regularizer ϕ, are tuned automatically by cross-validation.

The obtained results on this experiment in terms of accuracy per system are reported in Table 5. The optimal penalties $\lambda \in \Lambda$ selected by cross-validation for each regularizer ϕ are also included, while the optimal parameters τ, C are not displayed since they actually all equal 10^{+1} independently of the kernel used. We first notice that the proposed system `SVM` (support vector machine classifier) consistently outperforms the baseline system `GMM`

(Gaussian mixture model classifier). This proves the benefits of incorporating individual information per sound via an SVM rather than exploiting global information per class with a GMM. This further demonstrates the relevance of OT and more general ROT problems for the design of kernels between GMMs in the SVM pipeline. We also notice that `RMD` (rot mover's distance kernel) is at least competitive with `EMD` (earth mover's distance kernel) for all proposed regularizers, except from `EUC` which does not perform as well. This might be a consequence of the regularization profile for `EUC`, or equivalently `LPN` with $p = 2$, which does not spread enough mass across similar bins, implying a lack of robustness to slight variations in the means and variances of the GMM components. Reducing the power parameter in `LPN` brings back to a competitive system with `EMD` for $p = 1.1$, and even a better trade-off with improved accuracy for $p = 1.5$. We obtain similar results for `LPQN` with $p = 0.9$ and $p = 0.5$, with now the best compromise for the lowest power value $p = 0.1$ which clearly outperforms `EMD`. As a remark, the accuracy for `LPN` and `LPQN` is not unimodal with respect to p which controls the spread of mass in the regularization. We suspect this is because the performance is a function of both the spread of mass and the amount of regularization, whose coupling allows for similar compromises in terms of results within different regimes of use. Concerning `BETA` now, we observe that the existing Sinkhorn-Knopp algorithm `BSKL` for $\beta = 1$ does not improve the accuracy compared to `EMD`. Increasing the spread of mass with $\beta = 0$ in `BIS` is even worse. The best performance is obtained with a range in between for $\beta = 0.5$, which slightly improves results over `EMD`. Using `LOG` here slightly degrades the performance compared to `EMD` and `BSKL`. Interestingly, the overall best accuracy on this application is obtained for `HELL` which beats all other systems, including `EUC`, by a safe margin. In contrast to the experiment on synthetic data with dimension 256 presented in Section 6.1, where both `BSKL` and `LOG`, respectively `EUC` and `HELL`, behave similarly due to equivalence up to a constant for low values in the transport plans, the range of the transport plans here is much higher since the dimension of the input distributions is at most 16 (typically less than 10). This raises the importance of choosing a good regularizer depending on the actual task and its inherent design criteria such as the data dimension.

7. Conclusion

In this paper, we formulated a unified framework for smooth convex regularization of discrete OT problems. We also derived some algorithmic methods to solve such ROT problems, and detailed their specificities for classical regularizers and associated divergences from the literature. We finally designed a synthetic experiment to illustrate our proposed methods, and proved the relevance of ROT problems and the RMD on a real-world application to audio scene classification. The obtained results are encouraging for further development of the present work, and we now discuss some interesting perspectives for future investigation.

Firstly, we want to assess the effect of other regularizers on the solutions, notably when adding an affine term. From a geometrical viewpoint, such a transformation is equivalent to simply translating the cost matrix, with no effect on the Bregman divergence itself. For a given regularizer, we could therefore parametrize a whole family of interpolating regularizers, and tune the translation parameter according to the application. In particular, a recent work developed independently of ours makes use of Tsallis entropies to regularize OT problems with ad hoc solvers (Muzellec et al., 2018). These regularizers could be integrated readily to

Classifier		— ϕ —		β/p	Λ	λ	Accuracy
GMM	—	—	—	—	—	—	77.2%
	EMD	—	—	—	—	—	81.3%
		—	FDLOG	—	$10^{\{-2,-1,+0,+1\}}$	10^{-1}	81.0%
		BSKL	BETA	1.00	$10^{\{-2,-1,+0,+1\}}$	10^{+0}	81.3%
		—	BETA	0.50	$10^{\{-3,-2,-1,+0\}}$	10^{-2}	81.5%
		BIS	BETA	0.00	$10^{\{-4,-3,-2,-1\}}$	10^{-2}	81.3%
		—	LPQN	0.10	$10^{\{-2,-1,+0,+1\}}$	10^{-1}	82.1%
SVM	RMD	—	LPQN	0.50	$10^{\{-2,-1,+0,+1\}}$	10^{-1}	81.3%
		—	LPQN	0.90	$10^{\{-2,-1,+0,+1\}}$	10^{+0}	81.0%
		—	LPN	1.10	$10^{\{-1,+0,+1,+2\}}$	10^{+0}	81.0%
		—	LPN	1.50	$10^{\{+0,+1,+2,+3\}}$	10^{+1}	81.8%
		EUC	LPN	2.00	$10^{\{+1,+2,+3,+4\}}$	10^{+3}	77.4%
		—	HELL	—	$10^{\{+1,+2,+3,+4\}}$	10^{+2}	82.8%

Table 5: Results of the experiment on audio classification.

our more general framework based on alternate Bregman projections, since Tsallis entropies are equivalent to β-potentials and ℓ_p (quasi)-norms up to an affine term.

In another direction, we would like to extend some theoretical results that hold for the Boltzmann-Shannon entropy and associated Kullback-Leibler divergence. Specifically, it is known that the related rot mover's plan converges in norm to the earth mover's plan with an exponential rate as the penalty decreases (Cominetti and San Martín, 1994). It is not straightforward, however, to generalize this to other regularizers and divergences. In addition, it would be worth elucidating some technical restrictions under which metric properties such as the triangular inequality can be proved similarly to Sinkhorn distances.

We also plan to study other pattern recognition tasks in text, image and audio signal processing. Intuitive possibilities include retrieval and classification for various kinds of data modeled via histograms of features or GMMs. Among potential approaches, this can be addressed by exploiting the RMD either directly in a nearest-neighbor search, or in the design of kernels for an SVM as done here for acoustic scenes. For such tasks, it would be relevant to provide insight into the choice of a good regularizer for the actual problem, or develop methods for automatic tuning of regularization parameters, and for learning the cost matrix from the data as can be done for the EMD (Cuturi and Avis, 2014). Even if we mostly focused on separable regularizers, it would be relevant to further use the quadratic forms associated to Mahalanobis distances in certain applications, and maybe propose a parametric learning scheme for the quadratic regularizer from the data.

Lastly, a more prospective idea is to use the RMD instead of Sinkhorn distances in the recent works built on the entropic regularization mentioned in Section 1. We also think that variational ROT problems could be formulated for statistical inference, notably parameter estimation in finite mixture models by minimizing loss functions based on the RMD (Dessein et al., 2017). This would leverage new applications of our ROT framework for more general machine learning problems. Such developments are yet involved and require some theoretical effort before reaching enough maturity to address practical setups.

Acknowledgments

This study has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the "Investments for the future" Program IdEx Bordeaux (ANR-10-IDEX-03-02), Cluster of excellence CPU, and the GOTMI project (ANR-16-CE33-0010-01). The authors would like to thank Annamaria Mesaros for her kind help with the evaluation on the DCASE 2016 IEEE AASP challenge, Charles-Alban Deledalle for his valuable advice on the use of the computing platform PlaFRIM, and Marco Cuturi for the insightful discussions about this work.

References

Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

Ibrahim Alabdulmohsin, Xin Gao, and Xiangliang Zhang. Support vector machines with indefinite kernels. In *Asian Conference on Machine Learning (ACML)*, pages 32–47, 2014.

Shun-ichi Amari and Hiroshi Nagaoka. *Methods of Information Geometry*, volume 191 of *Translations of Mathematical Monographs*. American Mathematical Society, Providence, RI, USA, 2000.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. Technical report, arXiv:1701.07875, 2017.

Heinz H. Bauschke and Adrian S. Lewis. Dykstra's algorithm with Bregman projections: A convergence proof. *Optimization*, 48(4):409–427, 2000.

Jean-David Benamou, Guillaume Carlier, Marco Cuturi, Luca Nenna, and Gabriel Peyré. Iterative Bregman projections for regularized transportation problems. *SIAM Journal on Scientific Computing*, 37(2):A1111–A1138, 2015.

Espen Bernton, Pierre E. Jacob, Mathieu Gerber, and Christian P. Robert. Inference in generative models using the Wasserstein distance. Technical report, arXiv:1701.05146, 2017.

Jérémie Bigot, Raúl Gouet, Thierry Klein, and Alfredo López. Geodesic PCA in the Wasserstein space. Technical report, arXiv:1307.7721, 2013.

Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. Technical report, arXiv:1710.06276, 2017.

Nicolas Bonneel, Michiel van de Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. *ACM Transactions on Graphics*, 30(6): 158:1–158:12, 2011.

Olivier Bousquet, Sylvain Gelly, Ilya Tolstikhin, Carl-Johann Simon-Gabriel, and Bernhard Schölkopf. From optimal transport to generative modeling: the VEGAN cookbook. Technical report, arXiv:1705.07642, 2017.

Elsa Cazelles, Vivien Seguy, Jérémie Bigot, Marco Cuturi, and Nicolas Papadakis. Log-PCA versus geodesic PCA of histograms in the Wasserstein space. Technical report, arXiv:1708.08143, 2017.

Roberto Cominetti and Jaime San Martín. Asymptotic analysis of the exponential penalty trajectory in linear programming. *Mathematical Programming*, 67(1–3):169–187, 1994.

Nicolas Courty, Rémi Flamary, Devis Tuia, and Alain Rakotomamonjy. Optimal transport for domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(9):1853–1865, 2015.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 2292–2300, 2013.

Marco Cuturi and David Avis. Ground metric learning. *Journal of Machine Learning Research*, 15(1):533–564, 2014.

Marco Cuturi and Gabriel Peyré. A smoothed dual approach for variational Wasserstein problems. *SIAM Journal on Imaging Sciences*, 9(1):320–343, 2016.

Arnaud Dessein, Nicolas Papadakis, and Charles-Alban Deledalle. Parameter estimation in finite mixture models by regularized optimal transport: A unified framework for hard and soft clustering. Technical report, arXiv:1711.04366, 2017.

Inderjit S. Dhillon and Joel A. Tropp. Matrix nearness problems with Bregman divergences. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1120–1146, 2007.

Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-François Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. Learning with a Wasserstein loss. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 2053–2061, 2015.

Alfred Galichon and Bernard Salanié. Cupid's invisible hand: Social surplus and identification in matching models. Technical report, SSRN:1804623, 2015.

Aude Genevay, Gabriel Peyré, and Marco Cuturi. GAN and VAE from an optimal transport point of view. Technical report, arXiv:1706.01807, 2017.

Kristen Grauman and Trevor Darrell. Fast contour matching using approximate earth mover's distance. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 220–227, 2004.

Joachim Gudmundsson, Oliver Klein, Christian Knauer, and Michiel Smid. Small Manhattan networks and algorithmic applications for the earth mover's distance. In *European Workshop on Computational Geometry (EuroCG)*, pages 174–177, 2007.

Bernard Haasdonk. Feature space interpretation of SVMs with indefinite kernels. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(4):482–492, 2005.

Martin Idel. A review of matrix scaling and Sinkhorn's normal form for matrices and positive maps. Technical report, arXiv:1609.06349, 2016.

Piotr Indyk and Nitin Thaper. Fast image retrieval via embeddings. In *International Workshop on Statistical and Computational Theories of Vision (SCTV)*, 2003.

Sven Kurras. Symmetric iterative proportional fitting. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 526–534, 2015.

Hsuan-Tien Lin and Chih-Jen Lin. A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, National Taiwan University, 2003.

Haibin Ling and Kazunori Okada. An efficient earth mover's distance algorithm for robust histogram comparison. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(5):840–853, 2007.

Annamaria Mesaros, Toni Heittola, and Tuomas Virtanen. TUT database for acoustic scene classification and sound event detection. In *European Signal Processing Conference (EUSIPCO)*, pages 1128–1132, 2016.

Grégoire Montavon, Klaus-Robert Müller, and Marco Cuturi. Wasserstein training of restricted Boltzmann machines. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 3718–3726, 2016.

Boris Muzellec, Richard Nock, Giorgio Patrini, and Frank Nielsen. Tsallis regularized optimal transport and ecological inference. In *AAAI Conference on Artificial Intelligence (AAAI)*, pages 2387–2393, 2018.

Assaf Naor and Gideon Schechtman. Planar earthmover is not in l_1. *SIAM Journal on Computing*, 37(3):804–826, 2007.

Adam M. Oberman and Yuanlong Ruan. An efficient linear programming method for optimal transportation. Technical report, arXiv:1509.03668, 2015.

Ofir Pele and Michael Werman. A linear time histogram metric for improved SIFT matching. In *European Conference on Computer Vision (ECCV)*, pages 495–508, 2008.

Ofir Pele and Michael Werman. Fast and robust earth mover's distances. In *IEEE International Conference on Computer Vision (ICCV)*, pages 460–467, 2009.

Julien Rabin, Julie Delon, and Yann Gousseau. A statistical approach to the matching of local features. *SIAM Journal on Imaging Sciences*, 2(3):931–958, 2009.

Antoine Rolet, Marco Cuturi, and Gabriel Peyré. Fast dictionary learning with a smoothed Wasserstein loss. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 630–638, 2016.

Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

Morgan A. Schmitz, Matthieu Heitz, Nicolas Bonneel, Fred Ngolè, David Coeurjolly, Marco Cuturi, Peyré Gabriel, and Jean-Luc Starck. Wasserstein dictionary learning: Optimal transport-based unsupervised non-linear dictionary learning. *SIAM Journal on Imaging Sciences*, 11(1):643–678, 2018.

Bernhard Schmitzer. A sparse multi-scale algorithm for dense optimal transport. *Journal of Mathematical Imaging and Vision*, 56(2):238–259, 2016a.

Bernhard Schmitzer. Stabilized sparse scaling algorithms for entropy regularized transport problems. Technical report, arXiv:1610.06519, 2016b.

Vivien Seguy and Marco Cuturi. Principal geodesic analysis for probability measures under the optimal transport metric. In *International Conference on Neural Information Processing Systems (NIPS)*, pages 3312–3320, 2015.

Fei Sha, Yuanqing Lin, Lawrence K. Saul, and Daniel D. Lee. Multiplicative updates for nonnegative quadratic programming. *Neural Computation*, 19(8):2004–2031, 2007.

Sameer Shirdhonkar and David W. Jacobs. Approximate earth mover's distance in linear time. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, 2008.

Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967.

Justin Solomon, Raif M. Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *International Conference on Machine Learning (ICML)*, pages 306–314, 2014.

Justin Solomon, Fernando de Goes, Gabriel Peyré, Marco Cuturi, Adrian Butscher, Andy Nguyen, Tao Du, and Leonidas Guibas. Convolutional Wasserstein distances: Efficient optimal transportation on geometric domains. *ACM Transactions on Graphics*, 34(4): 66:1–66:11, 2015.

Alexis Thibault, Léénaïc Chizat, Charles Dossal, and Nicolas Papadakis. Overrelaxed sinkhorn-knopp algorithm for regularized optimal transport. Technical report, arXiv:1711.01851, 2017.

Lars Thorlund-Petersen. Global convergence of Newton's method on an interval. *Mathematical Methods of Operations Research*, 59(1):91–110, 2004.

Paul Tseng. Dual coordinate ascent methods for non-strictly convex minimization. *Mathematical Programming*, 59(1–3):231–247, 1993.

Cédric Villani. *Optimal Transport: Old and New*, volume 338 of *Comprehensive Studies in Mathematics*. Springer, Berlin Heidelberg, Germany, 2009.

Gloria Zen, Elisa Ricci, and Nicu Sebe. Simultaneous ground metric learning and matrix factorization with earth mover's distance. In *International Conference on Pattern Recognition (ICPR)*, pages 3690–3695, 2014.

Journal of Machine Learning Research 19 (2018) 1-7 Submitted 7/17; Revised 5/18; Published 8/18

ELFI: Engine for Likelihood-Free Inference

Jarno Lintusaari jarno.lintusaari@aalto.fi
Henri Vuollekoski henri.vuollekoski@aalto.fi
Antti Kangasrääsiö antti.kangasraasio@aalto.fi
Kusti Skytén kusti.skyten@aalto.fi
Marko Järvenpää marko.j.jarvenpaa@aalto.fi
Pekka Marttinen pekka.marttinen@aalto.fi
Department of Computer Science
Aalto University
00076 Aalto, Finland

Michael U. Gutmann michael.gutmann@ed.ac.uk
School of Informatics
The University of Edinburgh
Edinburgh, EH8 9AB, UK

Aki Vehtari * aki.vehtari@aalto.fi
Department of Computer Science
Aalto University
00076 Aalto, Finland

Jukka Corander * jukka.corander@medisin.uio.no
Department of Biostatistics
University of Oslo
0317 OSLO, Norway

Samuel Kaski * samuel.kaski@aalto.fi
Department of Computer Science
Aalto University
00076 Aalto, Finland

Editor: Alexandre Gramfort

Abstract

Engine for Likelihood-Free Inference (ELFI) is a Python software library for performing likelihood-free inference (LFI). ELFI provides a convenient syntax for arranging components in LFI, such as priors, simulators, summaries or distances, to a network called ELFI graph. The components can be implemented in a wide variety of languages. The stand-alone ELFI graph can be used with any of the available inference methods without modifications. A central method implemented in ELFI is Bayesian Optimization for Likelihood-Free Inference (BOLFI), which has recently been shown to accelerate likelihood-free inference up to several orders of magnitude by surrogate-modelling the distance. ELFI also has an inbuilt support for output data storing for reuse and analysis, and supports parallelization of computation from multiple cores up to a cluster environment. ELFI is designed to be extensible and provides interfaces for widening its functionality. This makes the adding of new inference methods to ELFI straightforward and automatically compatible with the inbuilt features.

∗. Equal contribution

Keywords: Likelihood-free inference, approximate Bayesian computation, Python, BOLFI, parallel computing

1. Introduction

Engine for Likelihood-Free Inference (ELFI) is a statistical software package for likelihood-free inference written in Python. The term "likelihood-free inference" (LFI) refers to a family of inference methods that can be used when the likelihood function is not computable or otherwise available, but it is possible to simulate from the model (see e.g. Lintusaari et al., 2017). Other names for likelihood-free inference or closely related approaches include Approximate Bayesian Computation (ABC) (see e.g. Marin et al., 2012; Lintusaari et al., 2017), simulator-based inference, approximative Bayesian inference and indirect inference.

In LFI, generative models are commonly composed of priors and user-specified simulators. The inference is based on the outputs of the generative model, that is, on the simulated data for various parameter configurations, as opposed to the likelihoods of the observed data under the configurations. To facilitate the inference, the observed and simulated data are usually summarized after which distances between the summaries are taken. In ELFI, the simulators, summaries, distances, etc. are called components and can be implemented in a wide variety of languages.

One of the main features in ELFI is the convenient syntax of combining all of the components into a single network (Figure 1) that we call an ELFI graph. Once the ELFI graph is specified, it can be used with any of the available inference algorithms. ELFI also supports parallelization of the inference from a single computer up to a computational cluster, and storing the generated data for reuse, post-processing and further analysis. ELFI has emerged from the prior research on the subject by the authors (Lintusaari et al., 2016; Gutmann and Corander, 2016; Lintusaari et al., 2017; Kangasrääsiö et al., 2017) and was used by Kangasrääsiö et al. (2017) and Kangasrääsiö and Kaski (2017).

2. Software Design Principles

ELFI is designed to support likelihood-free inference research both from the practitioners' and methodologists' point of view. We aim for an easy-to-use ecosystem where practitioners will find the state-of-the-art inference methods, whereas methodologists will find simulators along with accompanying ELFI graphs and data to aid in method development and assessment.

2.1 Features for Practitioners

For practitioners ELFI provides a convenient interface for quickly arranging the components needed in LFI into an ELFI graph. Inherently ELFI graphs are directed acyclic graphs (DAGs), that represent how quantities used by the inference algorithm (e.g. distances) are computed (Figure 1). The DAG structure makes it possible to construct detailed hierarchies between the components (nodes). Under the hood, the ELFI graph is converted to a computation graph that will include e.g. nodes for the observed data (see the documentation[1] for more details). The nodes (components) are either data or operations that output data. Users are free to implement components as needed, but ELFI provides also ready made implementations for common components. Once specified, the ELFI graph can be directly

```python
# Define the simulator, the summary and the observed data
def simulator(t1, t2, batch_size=1, random_state=None):
    # Implementation comes here. Return `batch_size`
    # simulations wrapped to a NumPy array.
def summary(data, argument=0):
    # Implementation comes here...
y = # Observed data, as one element of a batch.

# Specify the ELFI graph
t1 = elfi.Prior('uniform', -2, 4)
t2 = elfi.Prior('normal', t1, 5)  # depends on t1
SIM = elfi.Simulator(simulator, t1, t2, observed=y)
S1 = elfi.Summary(summary, SIM)
S2 = elfi.Summary(summary, SIM, 2)
d = elfi.Distance('euclidean', S1, S2)

# Run the rejection sampler
rej = elfi.Rejection(d, batch_size=10000)
result = rej.sample(1000, threshold=0.1)
```

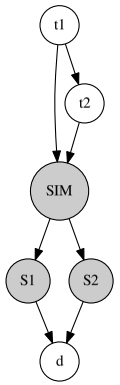


Figure 1: Example of an ELFI graph and of running ABC rejection sampling in ELFI. The observed data are given to the operation that produces corresponding output. Two summaries are defined, where summary S2 is given an additional argument 2.

used with any of the available inference methods. We have provided an initial set of methods that can handle different types of scenarios: basic rejection sampling for cheap simulators, the general-purpose sequential Monte Carlo as well as BOLFI (Gutmann and Corander, 2016) for expensive simulators. BOLFI combines probabilistic modelling of the distance with decision-making under uncertainty to decide for which parameter value to next run the simulator, significantly reducing the number of simulations needed.

Since likelihood-free inference often requires a moderate amount of experimentation (e.g. trying different summary statistics) it is important that specifying the components is made flexible and that already generated data can be reused. We found the DAG structure to be ideal for these tasks. First, ELFI allows any part of the ELFI graphs (e.g. nodes and their dependencies) to be redefined keeping the rest of the structure intact. Second, ELFI provides automatic storing of the full output of any node (e.g. the simulator). These data will be automatically reused when for example the summaries are changed, potentially resulting in significant savings in compute time. These features are demonstrated in the ELFI tutorial in the documentation.

Another important factor is the ability to use non-Python components in the ELFI graph. For instance, it may not always be practical or even possible to rewrite existing simulators in Python. ELFI provides both tools and examples in the documentation on how to use simulators written in other languages.

Other practical features include the ability to progress the inference iteratively and to stop early if necessary. The provided visualization functions support assessing the current state of the inference. Finally, the ELFI graph can be saved to a file and shared with

others. ELFI also guarantees that the results will be identical for the same seeds making the reproduction of the results easy.

2.2 Features for Methodologists

For methodologists ELFI provides a convenient platform for testing new algorithms with models from the literature (e.g. Ricker, 1954; Marin et al., 2012; Lintusaari et al., 2017) and comparing their performance against existing algorithms. The framework provides means for parallelization, data storing, seeding of pseudo random number generation and other important technicalities out of the box. The documentation includes instructions on how to implement new algorithms for ELFI. One of the major benefits is that all existing ELFI graphs will be usable with the new algorithms without modifications.

3. Performance and Scalability

Performance is an important factor in computationally heavy inference such as LFI. ELFI uses batches of computations to control execution performance and parallelization. A batch consists of a fixed number of consecutive evaluations of a node in the ELFI graph before moving to the next (e.g. 100 draws from the prior and then 100 simulations using those parameters). The standard parallelization strategy is to compute multiple batches in parallel. This provides several benefits. First, the computation of a single batch can often be vectorized with, for example, NumPy (van der Walt et al., 2011) for many of the basic operations (e.g. computing summaries or distances), making them efficient in Python. This is especially beneficial when experimenting with different summaries and distances with precomputed simulations. Batches are also often relatively constant in their time and memory consumption, allowing flexibility in planning the parallel execution of multiple batches. This helps in avoiding unnecessary message passing, progressing the inference in meaningful steps, and makes it possible to know in advance the size of the returned output data for storing purposes.

4. Comparison to Other Similar Software

There exist multiple LFI libraries for parameter inference. Many of them are either restricted to a specific problem domain (Liepe et al., 2014; Cornuet et al., 2014; Louppe et al., 2016), or require existing simulated data (Thornton, 2009; Csilléry et al., 2012; Nunes and Prangle, 2015). Edward (Tran et al., 2016) provides some LFI methods with a GPU acceleration, but requires the simulator to be differentiable. ELFI makes no extra requirements for the simulator (or other components), and can also be used with implementations taking benefit of hardware accelerations (e.g. GPU). General-purpose LFI software similar to ELFI are, to our knowledge, ABCtoolbox (Wegmann et al., 2010), EasyABC (Jabot et al., 2013), and ABCpy (Dutta et al., 2017). A relatively recent categorization of LFI software is provided by Nunes and Prangle (2015).

Among the general-purpose LFI software, only ELFI separates the LFI component specification from inference (Table 1). The graph-based specification provides considerable flexibility in both defining the components and experimenting with them. For example, it

Software	Language	Latest release	Data reuse	Parallelization	Graph-based	Iterative processing
ABCtoolbox	C++	2009	Partial	cluster (manual)	✗	✗
EasyABC	R	2015	Partial	local	✗	✗
ABCpy	Python	2017	✗	local and cluster	✗	✗
ELFI	Python	2017	✓	local and cluster	✓	✓

Table 1: Comparison of general-purpose LFI frameworks

is possible to embed multiple simulators into a single ELFI graph without modifying their codes. We refer the reader to the documentation for illustrations.[1]

Regarding parallelization, EasyABC supports multiple cores on a single computer while the others can also run in cluster environments. By default, ABCpy uses Spark (Zaharia et al., 2010) and ELFI ipyparallel for parallelization but both can be used with alternative backends.[2] ABCtoolbox does not provide a parallel solution out of the box.

ABCtoolbox, EasyABC and ELFI support reusing generated data. ELFI is more flexible in that it allows the output of any node of the ELFI graph to be stored, and it automatically uses that data to compute the output of its current or future child nodes. There is thus no need to manually transform existing data.

Only ELFI supports advancing the inference sample-by-sample, which facilitates debugging and enables e.g. convergence monitoring and early stopping. Also, ELFI is currently the only general-purpose software to implement the BOLFI method (Gutmann and Corander, 2016), which can handle expensive-to-evaluate simulators outside the reach of other methods.

5. Source Code and Dependencies

ELFI has been designed to be open source and modular, and can be extended through interfaces. For instance, it is possible to add new types of components, data stores or parallel clients. All the dependencies of ELFI are also open source.

ELFI is written in Python and is officially tested under Linux and MacOS but also works in Windows. The code style follows PEP 8 and documentation NumPy format. Code development uses the continuous integration practice with code review and automated tests to ensure the quality and usability of the software. The venue for distributing the source is GitHub that among the above features also allows anyone to raise issues regarding the software and make pull requests for new features.[3] Online documentation is hosted in the Read The Docs.[1] ELFI also has a community chat for the users.

Acknowledgments

We would like to acknowledge support for this project from the Academy of Finland (Finnish Centre of Excellence in Computational Inference Research COIN) and grants 294238, 292334. We acknowledge the computational resources provided by the Aalto Science-IT project.

1. ELFI documentation can be found at `http://elfi.readthedocs.io`.
2. The ipyparallel project can be found at `https://github.com/ipython/ipyparallel`.
3. The ELFI GitHub repository can be found at `https://github.com/elfi-dev/elfi`.

References

J. M. Cornuet, P. Pudlo, J. Veyssier, A. Dehne-Garcia, M. Gautier, R. Leblois, J. M. Marin, and A. Estoup. DIYABC v2.0: a software to make approximate Bayesian computation inferences about population history using single nucleotide polymorphism, DNA sequence and microsatellite data. *Bioinformatics*, 30(8):1187–1189, Apr 2014.

K. Csilléry, O. François, and M. G. B. Blum. abc: an R package for approximate Bayesian computation (ABC). *Methods in Ecology and Evolution*, 3(3):475–479, 2012. doi: 10.1111/j.2041-210X.2011.00179.x.

R. Dutta, M. Schoengens, J-P. Onnela, and A. Mira. ABCpy: A user-friendly, extensible, and parallel library for approximate Bayesian computation. In *Proceedings of the Platform for Advanced Scientific Computing Conference*, PASC '17, pages 8:1–8:9, New York, NY, USA, 2017. ACM. doi: 10.1145/3093172.3093233. URL `http://doi.acm.org/10.1145/3093172.3093233`.

M. U. Gutmann and J. Corander. Bayesian optimization for likelihood-free inference of simulator-based statistical models. *Journal of Machine Learning Research*, 17(125):1–47, 2016.

F. Jabot, T. Faure, and N. Dumoulin. EasyABC: performing efficient approximate Bayesian computation sampling schemes using R. *Methods in Ecology and Evolution*, 4(7):684–687, 2013. doi: 10.1111/2041-210X.12050.

A. Kangasrääsiö and S. Kaski. Inverse reinforcement learning from summary data. *arXiv preprint arXiv:1703.09700*, 2017.

A. Kangasrääsiö, K. Athukorala, A. Howes, J. Corander, S. Kaski, and A. Oulasvirta. Inferring cognitive models from data using approximate Bayesian computation. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, CHI '17, pages 1295–1306, New York, NY, USA, 2017. ACM. doi: 10.1145/3025453.3025576.

J. Liepe, P. Kirk, S. Filippi, T. Toni, C. P. Barnes, and M. P. Stumpf. A framework for parameter estimation and model selection from experimental data in systems biology using approximate Bayesian computation. *Nat Protoc*, 9(2):439–456, Feb 2014.

J. Lintusaari, M. U. Gutmann, S. Kaski, and J. Corander. On the identifiability of transmission dynamic models for infectious diseases. *Genetics*, 2016. doi: 10.1534/genetics.115.180034.

J. Lintusaari, M. U. Gutmann, R. Dutta, S. Kaski, and J. Corander. Fundamentals and recent developments in approximate Bayesian computation. *Systematic Biology*, 66(1):e66, 2017. doi: 10.1093/sysbio/syw077.

G. Louppe, K Cranmer, and J. Pavez. carl: a likelihood-free inference toolbox, March 2016. URL `http://dx.doi.org/10.5281/zenodo.47798`.

J-M. Marin, P. Pudlo, C. P. Robert, and R. J. Ryder. Approximate Bayesian computational methods. *Statistics and Computing*, 22(6):1167–1180, 2012. doi: 10.1007/s11222-011-9288-2.

M. A. Nunes and D. Prangle. abctools: An R Package for Tuning Approximate Bayesian Computation Analyses. *The R Journal*, 7(2):189–205, 2015.

W. E. Ricker. Stock and recruitment. *Journal of the Fisheries Research Board of Canada*, 11(5):559–623, 1954. doi: 10.1139/f54-039.

K. R. Thornton. Automating approximate Bayesian computation by local linear regression. *BMC Genetics*, 10(1):35, 2009. doi: 10.1186/1471-2156-10-35.

D. Tran, A. Kucukelbir, A. B. Dieng, M Rudolph, D. Liang, and D. M. Blei. Edward: A library for probabilistic modeling, inference, and criticism. *arXiv preprint arXiv:1610.09787*, 2016.

S. van der Walt, S. C. Colbert, and G. Varoquaux. The numpy array: A structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011. doi: 10.1109/MCSE.2011.37.

D. Wegmann, C. Leuenberger, S. Neuenschwander, and L. Excoffier. ABCtoolbox: a versatile toolkit for approximate Bayesian computations. *BMC Bioinformatics*, 11(1):116, 2010. doi: 10.1186/1471-2105-11-116.

M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *Proceedings of the 2Nd USENIX Conference on Hot Topics in Cloud Computing*, HotCloud'10, pages 10–10, Berkeley, CA, USA, 2010. USENIX Association.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 07/17; Revised 07/18; Published 08/18

Streaming kernel regression with provably adaptive mean, variance, and regularization

Audrey Durand AUDREY.DURAND@MCGILL.CA
Université Laval, Québec, Canada

Odalric-Ambrym Maillard ODALRIC.MAILLARD@INRIA.FR
INRIA, Lille, France

Joelle Pineau JPINEAU@CS.MCGILL.CA
McGill University & Facebook AI Research, Montreal, Canada

Editor: Csaba Szepesvari

Abstract

We consider the problem of streaming kernel regression, when the observations arrive sequentially and the goal is to recover the underlying mean function, assumed to belong to an RKHS. The variance of the noise is not assumed to be known. In this context, we tackle the problem of tuning the regularization parameter adaptively at each time step, while maintaining tight confidence bounds estimates on the value of the mean function at each point. To this end, we first generalize existing results for finite-dimensional linear regression with fixed regularization and known variance to the kernel setup with a regularization parameter allowed to be a measurable function of past observations. Then, using appropriate self-normalized inequalities we build upper and lower bound estimates for the variance, leading to Bernstein-like concentration bounds. The latter is used in order to define the adaptive regularization. The bounds resulting from our technique are valid uniformly over all observation points and all time steps, and are compared against the literature with numerical experiments. Finally, the potential of these tools is illustrated by an application to kernelized bandits, where we revisit the Kernel UCB and Kernel Thompson Sampling procedures, and show the benefits of the novel adaptive kernel tuning strategy.

Keywords: kernel, regression, online learning, adaptive tuning, bandits

1. Introduction

Many applications require solving an online optimization problem for an unknown, noisy, function defined over a possibly large domain space. Kernel regression methods can learn such possibly non-linear functions by sharing information gathered across observations. These techniques are being used in many fields where they serve a variety of applications like hyperparameters optimization (Snoek et al., 2012), active preference learning (Brochu et al., 2008), and reinforcement learning (Marchant and Ramos, 2014; Wilson et al., 2014). The idea is generally to rely on kernel regression to estimate a function that can be used for decision making and selecting the next observation point. Algorithmically speaking, standard kernel regression involves a regularization parameter that accounts for both the complexity of the unknown target function, and the variance of the noise. While most theoretical approaches rely on a fixed regularization parameter, in practice, people have often used heuristics in order to tune this parameter adaptively with time.

This however comes at the price of loosing theoretical guarantees. Indeed, in order for theoretical guarantees (based on concentration inequalities) to hold, existing approaches (Srinivas et al., 2010; Valko et al., 2013) require the regularization parameter in the kernel regression to be a fixed quantity. Further, they assume a prior and tight knowledge of the variance of the noise, which is unrealistic in practice. The reason for this cumbersome assumption is to adjust the regularization parameter in the kernel regression based on this deterministic quantity, as such a choice of regularization conveys a natural Bayesian interpretation (Rasmussen and Williams, 2006). Following this intuition, given an empirical estimate of the function noise based on gathered observations, one should be able to tune the regularization automatically. This is however non-trivial, first due to the streaming nature of the data, that allows the noise to be a measurable function of the past observations, second because concentration bounds on the empirical variance are currently unknown in such a general kernel setup, and finally because all existing theoretical bounds require the regularization parameter to be a deterministic constant, while we require here a parameterization that explicitly depends on past observations. The goal of this work is to provide the rigorous tools for performing an online tuning of the kernel regularization while preserving theoretical guarantees and confidence intervals in the context of streaming kernel regression with unknown noise. We thus hope to provide a sound method for adaptive tuning that is both interesting from a practical perspective and retains theoretical guarantees.

We gently start our contributions by Theorem 1 that generalizes existing concentration results (such as in Abbasi-Yadkori et al. (2011); Wang and de Freitas (2014)), and is explicitly stated for a regularization parameter that may differ from the noise. This result paves the way to an even more general result (Theorem 2) that holds when the regularization is tuned online at each step. Afterwards, we introduce a streaming variance estimator (Theorem 3) that yields empirical upper- and lower-bounds on the function noise. Plugging-in the resulting estimates leads to empirical Bernstein-like concentration results (Corollary 1) for the kernel regression, where we use the variance estimates in order to tune the regularization parameter. Section 4 presents an application to kernelized bandits, where regret bounds for Kernel UCB and Kernel Thompson Sampling procedures are derived. Section 5 discusses our results and compares them against other approaches. Finally, Section 6 shows the potential of all the previously introduced results while comparing them to existing alternatives through different numerical experiments. We postpone most of the proofs to the appendix.

2. Kernel streaming regression with a predictable noise process

Let us consider a sequential regression problem. At each time step $t \in \mathbb{N}$, a learner picks a point $x_t \in \mathcal{X} \subset \mathbb{R}^d$ and gets the observation

$$y_t = f_\star(x_t) + \xi_t \,,$$

where $f_\star$ is an unknown function assumed to belong to some function space $\mathcal{F}$, and ξ_t is a random noise. In the following, we assume a sub-Gaussian streaming predictable model:

Assumption 1 (Predictability) *The process generating the observations is* predictable *in the sense that there is a filtration $\mathcal{H} = (\mathcal{H}_t)_{t\in\mathbb{N}}$ such that x_t is $\mathcal{H}_{t-1}$-measurable and y_t is $\mathcal{H}_t$-measurable. Such an example is given by $\mathcal{H}_t = \sigma(x_1, \dots, x_{t+1}, y_1, \dots, y_t)$.*

Assumption 2 (Sub-Gaussian streaming model) *In the sub-Gaussian streaming predictable model, for some non-negative constant σ^2, the following holds*

$$\forall t \in \mathbb{N}, \forall \gamma \in \mathbb{R}, \quad \ln \mathbb{E}\Big[\exp(\gamma \xi_t)\Big|\mathcal{H}_{t-1}\Big] \leqslant \frac{\gamma^2\sigma^2}{2}\,.$$

Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a kernel function (that is continuous, symmetric positive definite) on a compact set $\mathcal{X}$ equipped with a positive finite Borel measure, and denote $\mathcal{K}$ the corresponding RKHS.

Information gain This quantity measures the information obtained about function $f_\star$ by sampling at points $(x_1, \dots, x_t)$. It is defined (Cover and Thomas, 1991) as the *mutual information* between $f_\star$ and the observations $(y_1, \dots, y_t)$:

$$I(y_1, \dots, y_t; f_\star) = H(y_1, \dots, y_t) - H(y_1, \dots, y_t|f_\star),$$

that is the difference between the *marginal entropy* and the *conditional entropy* of the distributions of observations. The information gain thus quantifies the reduction of uncertainty about $f_\star$ following these observations. For a multidimensional Gaussian, we have $H(\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})) = \frac{1}{2}\ln|2\pi e\boldsymbol{\Sigma}|$, such that for $\lambda = \sigma^2$ (Srinivas et al., 2010),

$$\gamma_t(\sigma^2) = I(y_1, \dots, y_t; f_\star) = \frac{1}{2}\ln\det(I_t + \sigma^{-2}\mathbf{K}_t),$$

where $\mathbf{K}_t = (k(s, s'))_{s,s'\leqslant t}$. In the linear case when $k(x, x') = x^\top x'$ for $x \in \mathbb{R}^d$, the information gain typically scales as $\gamma_t(\sigma^2) = \mathcal{O}(d\ln t)$ (Srinivas et al., 2010). The information gain can be shown to scale with the *effective dimensionality* (Valko et al., 2013) instead of the dimension, where effective dimensions correspond to the most informative ones. More effective dimensions require more observations for a good space coverage, which increases the information gain. We now extend the information gain to any regularization λ.

Definition 1 (Information gain with unknown variance) *We define the information gain at time t for a regularization parameter λ to be*

$$\gamma_t(\lambda) = \frac{1}{2}\sum_{t'=1}^{t}\ln\left(1 + \frac{1}{\lambda}k_{\lambda,t'-1}(x_{t'}, x_{t'})\right).$$

This generalization is natural in view of Theorem 1 below. The information gain is inversely proportional to the regularization λ. By controlling the flexibility of the regression model, the regularization limits the impact of a new observation on the resulting model, therefore limiting the information that can be gained out of it.

Concentration We first provide a result bounding the prediction error of a standard regularized kernel estimate, where the regularization is given by a fixed parameter $\lambda > 0$.

Theorem 1 (Streaming kernel least-squares (Maillard, 2016)) *Assume we are in the sub-Gaussian streaming predictable model. For a parameter $\lambda \in \mathbb{R}$, let us define the posterior mean and variances after observing $Y_t = (y_1, \dots, y_t)^\top \in \mathbb{R}^{t\times 1}$ as*

$$\begin{cases} f_{\lambda,t}(x) = & k_t(x)^\top(\mathbf{K}_t + \lambda I_t)^{-1}Y_t \\ s^2_{\lambda,t}(x) = & \frac{\sigma^2}{\lambda}k_{\lambda,t}(x,x) \text{ with } k_{\lambda,t}(x,x) = k(x,x) - k_t(x)^\top(\mathbf{K}_t + \lambda I_t)^{-1}k_t(x)\,. \end{cases}$$

where $k_t(x) = (k(x, x_{t'}))_{t'\leqslant t}$ is a $t \times 1$ (column) vector and $\mathbf{K}_t = (k(x_s, x_{s'}))_{s,s'\leqslant t}$. Then $\forall\delta \in [0, 1]$, with probability higher than $1-\delta$, it holds simultaneously over all $x \in \mathcal{X}$ and $t \geqslant 0$,

$$|f_\star(x) - f_{\lambda,t}(x)| \leqslant \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}}\left[\sqrt{\lambda}\|f_\star\|_{\mathcal{K}} + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda)}\right],$$

where the quantity $\gamma_t(\lambda) = \frac{1}{2}\sum_{t'=1}^{t}\ln\left(1+\frac{1}{\lambda}k_{\lambda,t'-1}(x_{t'}, x_{t'})\right)$ *is the information gain.*

Remark 1 *This result should be considered as an extension of Abbasi-Yadkori et al. (2011, Theorem 2) from finite-dimensional to possibly infinite dimensional function space. More specifically, when considering the linear kernel, the result of Theorem 1 recovers exactly Theorem 2 from Abbasi-Yadkori et al. (2011). The generalization is non trivial as the Laplace method must be amended in order to be applied beyond the linear case.*

Remark 2 *This result holds uniformly over all $x \in \mathcal{X}$ and most importantly over all $t \geqslant 0$, thanks to a random stopping time construction (related to the occurrence of* bad events*) and a self-normalized inequality handling this stopping time. This is in contrast with results such as Wang and de Freitas (2014), that are only stated separately for each t.*

The case when $\lambda = \lambda_\star \overset{\text{def}}{=} \sigma^2/\|f_\star\|^2_{\mathcal{K}}$ is of special interest, since we get on the one hand

$$\begin{aligned} f_{\lambda_\star,t}(x) &= k_t(x)^\top(\mathbf{K_t} + \lambda_\star I_t)^{-1}Y_t \\ s^2_{\lambda_\star,t}(x) &= \|f_\star\|^2_{\mathcal{K}}k_t(x,x) \text{ with } k_t(x,x) = k(x,x) - k_t(x)^\top(\mathbf{K_t} + \lambda_\star I_t)^{-1}k_t(x) \end{aligned}$$

and on the other hand

$$|f_\star(x) - f_{\lambda_\star,t}(x)| \leqslant \|f_\star\|_{\mathcal{K}}\sqrt{k_t(x,x)}\left[1 + \sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_\star)}\right].$$

In practice however, neither $\|f_\star\|^2_{\mathcal{K}}$ nor σ^2 may be known exactly. In this paper, we make the following assumption on the former:

Assumption 3 (Bounded norm in RKHS) *An upper bound C is given on $\|f_\star\|_{\mathcal{K}}$. This essentially means that the kernel is well chosen for capturing $f_\star$. For more details, see Canu et al. (2009); Loustau (2009); Wasserman (2017).*

Then, we want to build an estimate of σ^2 at each time t in order to tune λ. Using a sequence of regularization parameters $(\lambda_t)_{t\geqslant 1}$ that is tuned adaptively based on the past observations requires to modify the previous result (it is only valid for a deterministic λ) into the following more general statement:

Theorem 2 (Streaming kernel least-squares with online tuning) *Under the same assumption as Theorem 1, let $\boldsymbol{\lambda} = (\lambda_t)_{t\geqslant 1}$ be a predictable positive sequence of parameters, that is λ_t is $\mathcal{H}_{t-1}$-measurable for each t. Assume that for each t, $\lambda_t \geqslant \lambda_\star$ holds for a positive constant $\lambda_\star$. Let us define the modified posterior mean and variances after observing $Y_t \in \mathbb{R}^t$ as*

$$\begin{cases} f_{\boldsymbol{\lambda},t}(x) &= k_t(x)^\top (\mathbf{K}_t + \lambda_{t+1} I_t)^{-1} Y_t \\ s^2_{\boldsymbol{\lambda},t}(x) &= \frac{\sigma^2}{\lambda_{t+1}} k_{\lambda_{t+1},t}(x,x) \text{ with } k_{\lambda,t}(x,x) = k(x,x) - k_t(x)^\top (\mathbf{K}_t + \lambda I_t)^{-1} k_t(x)\,, \end{cases}$$

where $k_t(x) = (k(x,x_{t'}))_{t'\leqslant t}$, and $\mathbf{K}_t = (k(x_s,x_{s'}))_{s,s'\leqslant t}$. Then for all $\delta \in [0,1]$, with probability higher than $1-\delta$, it holds simultaneously over all $x \in \mathcal{X}$ and $t \geqslant 0$

$$|f_\star(x) - f_{\boldsymbol{\lambda},t}(x)| \leqslant \sqrt{\frac{k_{\lambda_{t+1},t}(x,x)}{\lambda_{t+1}}} \left[\sqrt{\lambda_{t+1}}\|f_\star\|_{\mathcal{K}} + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_\star)}\right].$$

The proof is presented in Appendix A.

The regularization parameter λ_{t+1} is therefore used in conjunction with previous data up to time t to provide the posterior regression model (mean and variance) that is used in return to acquire the next observation y_{t+1} on point x_{t+1}.

Remark 3 *Since λ_t is allowed to be $\mathcal{H}_{t-1}$-measurable, this gives theoretical guarantees for virtually any adaptive tuning procedure of the regularization parameter.*

Remark 4 *The assumption that $\lambda_t \geqslant \lambda_\star$ will be naturally satisfied for the choice of regularization we consider.*

3. Variance estimation

We now focus on the estimation of the variance parameter of the noise in the case when it is unknown, or loosely known. Theorem 2 suggests to define the sequence $(\lambda_t)_{t\geqslant 1}$ by

$$\lambda_t = \sigma^2_{+,t-1}/C^2 \quad \text{with} \quad \sigma_{+,t} = \min\{\tilde{\sigma}_{+,t}, \sigma_{+,t-1}\} \quad \text{and} \quad \sigma_{+,0} = \sigma_+\,, \tag{1}$$

where $\sigma_+ \geqslant \sigma$ is an initial loose upper bound on σ and $\tilde{\sigma}_{+,t}$ is an upper-bound estimate on σ built from all observations gathered up to time t (inclusively). This ensures that λ_t is $\mathcal{H}_{t-1}$ measurable for all t and satisfies $\lambda_t \geqslant \lambda_\star$ with high probability, where $\lambda_\star = \sigma^2/C^2$. The crux is now to define the upper-bound estimate $\sigma_{+,t}$ on σ. In order to get a variance estimate, one obviously requires more than the sub-Gaussian assumption, since the term σ^2 has no reason to be tight (the inequality remains valid when σ^2 is replaced with any larger value). In order to convey the minimality of σ^2, we assume that the noise sequence is both σ-sub-Gaussian and second-order[1] σ-sub-Gaussian, in the sense that

$$\forall t, \forall \gamma < \frac{1}{2\sigma^2} \quad \ln \mathbb{E}\left[\exp(\gamma \xi_t^2) \middle| \mathcal{H}_{t-1}\right] \leqslant -\frac{1}{2}\ln\left(1 - 2\gamma\sigma^2\right).$$

1. The term on the right-hand side corresponds to the cumulant generating function of the chi-squared distribution with 1 degree of freedom. This assumption naturally holds for Gaussian variables.

Remark 5 *To avoid any technicality, one may assume that $\xi_t|\mathcal{H}_{t-1}$ is exactly $\mathcal{N}(0,\sigma^2)$, in which case it is trivially second-order σ-sub-Gaussian.*

Now let $\widehat{\sigma}^2_{\lambda,T} = \frac{1}{T}\sum_{t=1}^T (y_t - f_{\lambda,T}(x_t))^2$ denote the (slightly biased) variance estimate for a regularization parameter λ.

Theorem 3 (Streaming kernel variance estimate) *Assume we are in the predictable second-order σ-sub-Gaussian streaming regression model, with a predictable positive sequence $\boldsymbol{\lambda}$ such that $\lambda_t \geqslant \lambda_\star$ holds for all t. Let us introduce the following quantities*

$$C_t(\delta) = \ln(e/\delta)\left[1 + \ln(\pi^2 \ln(t)/6)/\ln(1/\delta)\right], \qquad D_{\lambda,t}(\delta) = 2\ln(1/\delta) + 2\gamma_t(\lambda)$$
$$\text{and finally} \quad \alpha = \max\left(1 - \sqrt{\tfrac{C_t(\delta')}{t}} - \sqrt{\tfrac{C_t(\delta')+2D_{\lambda_\star,t}(\delta')}{t}}, 0\right).$$

Then, let us introduce the following variance bounds, defined differently depending on whether a deterministic upper bound $\sigma_+ \geqslant \sigma$ is known (case 1) or not (case 2).

$$\sigma_{+,t}(\lambda,\lambda_\star) = \begin{cases} \widehat{\sigma}_{\lambda,t} + \sigma_+\left(\sqrt{\frac{C_t(\delta')}{t}} + \sqrt{\frac{C_t(\delta')+2D_{\lambda_\star,t}(\delta')}{t}}\right) + \sqrt{\frac{2\sigma_+\|f_\star\|_{\mathcal{K}}\sqrt{\lambda D_{\lambda_\star,t}(\delta')}}{t}} & \text{(case 1)} \\ \frac{1}{\alpha^2}\left(\sqrt{\widehat{\sigma}_{\lambda,t}\alpha + \frac{\|f_\star\|_{\mathcal{K}}\sqrt{\lambda D_{\lambda_\star,t}(\delta')}}{2t}} + \sqrt{\frac{\|f_\star\|_{\mathcal{K}}\sqrt{\lambda D_{\lambda_\star,t}(\delta')}}{2t}}\right)^2 & \text{(case 2)} \end{cases}$$

$$\sigma_{-,t}(\lambda) = \begin{cases} \widehat{\sigma}_{\lambda,t} - \sigma_+\sqrt{\frac{2C_t(\delta')}{t}} - \|f_\star\|_{\mathcal{K}}\sqrt{\frac{\lambda}{t}\left(1 - \frac{1}{\max_{t'\leqslant t}(1+\frac{1}{\lambda}k_{\lambda,t'-1}(x_{t'},x_{t'}))}\right)} & \text{(case 1)} \\ \left[\widehat{\sigma}_{\lambda,t} - \|f_\star\|_{\mathcal{K}}\sqrt{\frac{\lambda}{t}\left(1 - \frac{1}{\max_{t'\leqslant t}(1+\frac{1}{\lambda}k_{\lambda,t'-1}(x_{t'},x_{t'}))}\right)}\right]\left(1 + \sqrt{\frac{2C_t(\delta')}{t}}\right)^{-1} & \text{(case 2).} \end{cases}$$

Then with probability higher than $1-3\delta'$, it holds simultaneously for all $t \geqslant 0$

$$\sigma_{-,t}(\lambda_t) \leqslant \sigma \leqslant \sigma_{+,t}(\lambda_t,\lambda_\star)\,.$$

The proof is presented in Appendix B.

Remark 6 *The case when absolutely no bound is known on the noise σ^2 is challenging in practice. In this case, it is intuitive that one should not be able to recover the noise with too few samples. The bound stated in Theorem 3 (see Appendix B) supports this intuition, as when the number of observations is too small, then $\alpha = 0$ and the corresponding bound becomes trivial ($\sigma \leqslant \infty$).*

Remark 7 *In the variance bounds of Theorem 6 the term $\|f_\star\|_{\mathcal{K}}$ appears systematically with the factor $\sqrt{\lambda}$. This suggests we need to choose λ proportional to $1/\|f_\star\|^2_{\mathcal{K}}$, which gives further justification to the target $\lambda_\star = \sigma^2/C^2$, where C is a known upper bound on $\|f_\star\|$.*

Remark 8 *In practice, we advice to choose the best of case 1 and case 2 bounds when $\sigma_+ \geqslant \sigma$ is known.*

Because $\lambda_\star$ is not known in practice, the quantity $\sigma_{+,t}(\lambda_t, \lambda_\star)$ is not computable directly. However, we observe that $\sigma_{+,t}(\lambda_t, \lambda_\star)$ scales with $D_{\lambda_\star,t}$ (directly and through α), and that $D_{\lambda_\star,t}$ scales with the information gain $\gamma_t(\lambda_\star)$. Recall that the information gain scales inversely with the regularization. Hence we have that for any $\lambda_- \leqslant \lambda_\star$, we also have $\sigma_{+,t}(\lambda, \lambda_-) \geqslant \sigma_{+,t}(\lambda, \lambda_\star)$. Therefore, in order to *estimate* the upper bound $\sigma_{+,t}(\lambda, \lambda_\star)$, one only needs a lower-bound on $\lambda_\star$. Let us define

$$\sigma_{-,t} = \max\{\tilde{\sigma}_{-,t}, \sigma_{-,t-1}\} \quad \text{with} \quad \sigma_{-,0} = \sigma_-\,, \tag{2}$$

where $0 \leqslant \sigma_- \leqslant \sigma$ is a initial lower-bound on σ and $\tilde{\sigma}_{-,t}$ is a lower-bound estimate on σ built from all observations gathered up to time t (inclusively). Then, one way to proceed is, at each time step $t \geqslant 1$, to build an estimate $\tilde{\sigma}_{-,t} = \sigma_{-,t}(\lambda)$, which in return can be used to compute the lower quantity $\lambda_- = \sigma^2_{-,t}/C^2 \leqslant \sigma^2/C^2 = \lambda_\star$, and obtain the estimate $\tilde{\sigma}_{+,t} = \sigma_{+,t}(\lambda, \lambda_-) \geqslant \sigma_{+,t}(\lambda, \lambda_\star)$. This "sandwich estimates" procedure allows us to build an upper bound without prior knowledge of $\lambda_\star$, and then compute the predictable sequence $\boldsymbol{\lambda}$ as described by Equation 1. Given Theorem 3, we have that $\sigma_{-,t}(\lambda_t) \leqslant \sigma$ such that $\lambda_- \leqslant \lambda_\star$ and $\sigma_{+,t}(\lambda_t, \lambda_-) \geqslant \sigma$, hence $\lambda_t \geqslant \lambda_\star$, simultaneously for all $t \geqslant 0$, with high probability. Further replacing the variance σ with its estimate $\sigma_{+,t}$ using a union bound in the result of Theorem 2, we derive confidence bounds that are *fully computable empirically* in the context where the regularization parameter is adaptively tuned and the function noise is unknown. This is summarized in the following empirical Bernstein-style inequality:

Corollary 1 (Kernel empirical-Bernstein inequality) *Assume that $C \geqslant \|f_\star\|_{\mathcal{K}}$. Let us define the following noise lower-bound for each $t \geqslant 1$*

$$\sigma_{-,t} = \max\{\sigma_{-,t}(\lambda_{t-1}), \sigma_{-,t-1}\}$$

and define $\lambda_- = \sigma^2_{-,t}/C^2$ as the corresponding lower bound on $\lambda_\star$. Then, let us define the following noise upper bound for each $t \geqslant 1$

$$\sigma_{+,t} = \min\{\sigma_{+,t}(\lambda_{t-1}, \lambda_-), \sigma_{+,t-1}\}\,.$$

Define the regularization parameterizing the regression model used for acquiring observation at time t to be $\lambda_t = \sigma^2_{+,t}/C^2$, according to Equation 1. Then with probability higher than $1 - 4\delta$, the following is valid simultaneously for all $x \in \mathcal{X}$ and $t \geqslant 0$,

$$\big|f_\star(x) - f_{\lambda_t,t}(x)\big| \leqslant \sqrt{\frac{k_{\lambda_t,t}(x,x)}{\lambda_t}}\, B_{\lambda_t,t}(\delta) \qquad \textit{where}$$

$$B_{\lambda_t,t}(\delta) = \sqrt{\lambda_t}C + \sigma_{+,t}\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_-)}\,. \tag{3}$$

Proof Let E_f denote the event that

$$|f_\star(x) - f_{\boldsymbol{\lambda},t}(x)| \leqslant \sqrt{\frac{k_{\lambda_{t+1},t}(x,x)}{\lambda_{t+1}}}\Big[\sqrt{\lambda_{t+1}}\|f_\star\|_{\mathcal{K}} + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_\star)}\Big]$$

simultaneously for all $x \in \mathcal{X}$ and $t \geqslant 0$, and let $E_{\boldsymbol{\lambda}}$ denote the event that $\lambda_t \geqslant \lambda_\star$ holds for all t. We can decompose

$$\mathbb{P}[E_f^c] \leqslant \mathbb{P}[E_f^c \cap E_{\boldsymbol{\lambda}}] + \mathbb{P}[E_f^c \cap E_{\boldsymbol{\lambda}}^c] \leqslant \mathbb{P}[E_f^c \cap E_{\boldsymbol{\lambda}}] + \mathbb{P}[E_{\boldsymbol{\lambda}}^c]$$

By Theorem 2, we have that $\mathbb{P}[E_f^c \cap E_{\boldsymbol{\lambda}}] \leqslant \delta$. We need to show that $\lambda_t \geqslant \lambda_\star$ for all $t \geqslant 0$ by tuning λ_t with the proposed procedure. Let us look at what happens at each time t. Using the proposed procedure, we have $\lambda_0 = \sigma^2_+/C^2 \geqslant \lambda_\star$. Then we have

$$\begin{aligned}
\sigma_{-,1}(\lambda_0) &\leqslant \sigma \quad \text{L1}\\
\sigma_{+,1}(\lambda_0, \sigma_{-,1}(\lambda_0)^2/C^2) &\geqslant \sigma \quad \text{U1}\\
\to \lambda_1 = \sigma_{+,1}(\lambda_0, \sigma_{-,1}(\lambda_0)^2/C^2)^2/C^2 &\geqslant \lambda_\star\\
\sigma_{-,2}(\lambda_1) &\leqslant \sigma \quad \text{L2}\\
\sigma_{+,2}(\lambda_1, \sigma_{-,2}(\lambda_1)^2/C^2) &\geqslant \sigma \quad \text{U2}\\
\to \lambda_2 = \sigma_{+,2}(\lambda_1, \sigma_{-,1}(\lambda_1)^2/C^2)^2/C^2 &\geqslant \lambda_\star\\
&\dots
\end{aligned}$$

such that $E_{\boldsymbol{\lambda}}$ holds given that steps L1, U1, L2, U2, ... hold simultaneously. Therefore, $\mathbb{P}[E_{\boldsymbol{\lambda}}^c]$ is bounded by the probability that these steps do not hold simultaneously. Following Theorem 3, we have that $\mathbb{P}[E_{\boldsymbol{\lambda}}^c] \leqslant 3\delta$ and thus $\mathbb{P}[E_f^c] \leqslant 4\delta$. Naturally, under the event $E_{\boldsymbol{\lambda}}$, we have $\sigma_{+,t} \geqslant \sigma$ and $\lambda_- \leqslant \lambda_\star$. Therefore, given $C \geqslant \|f_\star\|_{\mathcal{K}}$, we have

$$\sqrt{\lambda_{t+1}}\|f_\star\|_{\mathcal{K}} + \sigma\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_\star)} \leqslant \sqrt{\lambda_{t+1}}C + \sigma_{+,t}\sqrt{2\ln(1/\delta) + 2\gamma_t(\lambda_-)}.$$

■

Remark 9 *This result is especially interesting since it provides a fully empirical confidence envelope function around $f_\star$. When an initial bound on the noise σ_+ is known and considered to be tight, one may simply choose the constant deterministic sequence $\boldsymbol{\lambda} = (\lambda, \dots, \lambda)$, in which case the same result holds for $\lambda_- = \lambda$ and $\sigma_{+,t} = \sigma_+$.*

We observe from Theorem 3 that the tightness of the noise estimates depends on the λ parameter that is used for computing $\tilde{\sigma}_{-,t}$ and $\tilde{\sigma}_{+,t}$. Since $\sigma^2/C^2 \leqslant \lambda_t \leqslant \sigma^2_+/C^2$ holds with high probability by construction, using such an adaptive λ_t should yield tighter bounds than using a fixed σ^2_+/C^2. This is supported by the numerical experiments of Section 6.2.

4. Application to kernelized bandits

Here is a direct application of our results in the framework of stochastic multi-armed bandits with structured arms embedded in an RKHS (Srinivas et al., 2010; Valko et al., 2013). At each time step $t \geqslant 1$, a bandit algorithm recommends a point x_t to sample from a compact set $\mathbb{X} \subseteq \mathcal{X}$, and observes a noisy outcome $y_t = f_\star(x_t) + \xi_t$, where $\xi_t \sim \mathcal{N}(0, \sigma^2)$. Let $x_\star = \operatorname{argmax}_{x \in \mathbb{X}} f_\star(x)$ denote the optimal arm. The goal of an algorithm is to pick a sequence of points $(x_t)_{t \leqslant T}$ that minimizes the cumulative regret

$$\mathfrak{R}_T = \sum_{t=1}^{T} f_\star(x_\star) - f_\star(x_t). \tag{4}$$

In this context, one needs to build tight confidence sets on the mean of each arm, and this will be given by Corollary 1. We illustrate our technique on two main bandit strategies: Upper Confidence Bound (UCB) (Auer et al., 2002) and Thompson Sampling (TS) (Thompson, 1933); both are adapted here to the kernel setting with unknown variance.

The following extension of Lemma 7 from Wang and de Freitas (2014) (see also Srinivas et al. (2012)) to the case when the variance is estimated plays an important role in the regret analysis of both algorithms.

Lemma 1 (From sum of variances to information gain) *Let us assume that the kernel is bounded by 1 in the sense that* $\sup_{x\in\mathcal{X}} k(x,x) \leqslant 1$. *Let* $\boldsymbol{\lambda}$ *be any sequence such that* $\forall \lambda \in \boldsymbol{\lambda}, \lambda \geqslant \sigma^2/C^2$. *For instance, this is satisfied with high probability when using Equation 1. Then, it holds*

$$\sum_{t=1}^{T} s^2_{\boldsymbol{\lambda},t-1}(x_t) = \sigma^2 \sum_{t=1}^{T} \frac{1}{\lambda_t} k_{\lambda_t,t-1}(x_t,x_t) \leqslant \frac{2C^2}{\ln(1+C^2/\sigma^2)} \gamma_T(\sigma^2/C^2)\,.$$

In the sequel, it is useful to bound the confidence bound term $B_{\lambda_t,t}(\delta)$ from Equation 3.

Lemma 2 (Deterministic bound on the confidence bound) *Assume that we are given a constant* $0 < \sigma_- < \sigma$, *so that* $\sigma_{t,-} \geqslant \sigma_-$ *holds for all* t. *Then for all* $t \leqslant T$, *the confidence bound term is upper-bounded by the following deterministic quantity*

$$B_{\lambda_t,t}(\delta) \leqslant \sigma_+ \left(1 + \sqrt{2\ln(1/\delta) + 2\gamma_T(\sigma^2_-/C^2)}\right).$$

Further, we have $\gamma_t(\sigma^2_{t,-}/C^2) = \gamma_t(\sigma^2_-/C^2) + O(1/\sqrt{t})$.

Remark 10 *The term* σ_+ *can be replaced with a more refined term* $\sigma + O(1/\sqrt{t})$ *thanks to the confidence bounds on the variance estimates.*

Kernel UCB with unknown variance The upper bound on the error can be used directly in order to build a UCB-style algorithm. Formally, the vanilla UCB algorithm (Auer et al., 2002) corresponding to our setting picks at time t the arm

$$x_t \in \operatorname*{argmax}_{x\in\mathbb{X}} f^+_{\lambda_t,t-1}(x) \quad \text{where} \quad f^+_{\lambda,t}(x) = f_{\lambda,t}(x) + \sqrt{\frac{k_{\lambda,t}(x,x)}{\lambda}} B_{\lambda,t}(\delta)\,. \tag{5}$$

Following the regret proof strategy of Abbasi-Yadkori et al. (2011), with some minor modifications, yields the following guarantee on the regret of this strategy:

Theorem 4 (Kernel UCB with unknown noise and adaptive regularization) *With probability higher than* $1-\delta$, *the regret of Kernel UCB with adaptive regularization and variance estimation satisfies for all* $T \geqslant 0$ *(recall that* $B_{\lambda_{t+1},t}(\delta)$ *is defined in Equation 3):*

$$\mathfrak{R}_T \leqslant 2\sum_{t=1}^{T} \sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}} B_{\lambda_t,t-1}(\delta/4)\,.$$

In particular, we have

$$\mathfrak{R}_T \;\leqslant\; 2\frac{\sigma_+}{\sigma}\Big(1 + \sqrt{2\ln(4/\delta) + 2\gamma_T(\sigma^2_-/C^2)}\Big) C \sqrt{T \frac{2\gamma_T(\sigma^2/C^2)}{\ln(1+C^2/\sigma^2)}}\,.$$

Remark 11 *This result that holds simultaneously over all time horizon* T *extends that of Abbasi-Yadkori et al. (2011) first to kernel regression and then to the case when the variance of the noise is unknown. This should also be compared to Valko et al. (2013) that assumes bounded observations, which implies a bounded noise (with known bound) and a bounded* $f_\star$, *and Srinivas et al. (2010) that provides looser bounds.*

Kernel TS with unknown variance Another application of our confidence bounds is in the analysis of Thompson sampling in the kernel scenario. Before presenting the result, let us say a few words about the design of TS algorithm in a kernel setting. Such an algorithm requires sampling from a posterior distribution over the arms. It is natural to consider a Gaussian posterior with posterior means and variances given by the kernel estimates. However, it has been noted in a series of papers (Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) that, in order to obtain provable regret minimization guarantees, the posterior variance should be inflated (although in practice, the vanilla version without inflation may work better). Following these lines of research, and owing to our novel confidence bounds, we derive the following TS algorithm using a posterior variance inflation factor v_t^2.

Algorithm 1 Kernel TS with adaptive variance estimation and regularization tuning.

Input: discrete space $\mathbb{X}$.

Parameters: regularization sequence $\boldsymbol{\lambda}$, variance inflation factor v_t^2 for each t.

for all $t \geqslant 1$ **do**
  compute the posterior mean $\widehat{\mathbf{f}}_{t-1} = (f_{\lambda_t,t-1}(x))_{x\in\mathbb{X}}$
  compute the posterior covariance $\widehat{\boldsymbol{\Sigma}}_{t-1} = \frac{\sigma^2_{+,t-1}}{\lambda_t}\big(k_{\lambda_t,t-1}(x,x')\big)_{x,x'\in\mathbb{X}}$
  sample $\tilde{f}_t = \mathcal{N}(\widehat{\mathbf{f}}_{t-1}, v_t^2\widehat{\boldsymbol{\Sigma}}_{t-1})$
  play $x_t = \operatorname{argmax}_{x\in\mathbb{X}} \tilde{f}_t(x)$
  observe outcome $y_t = f_\star(x_t) + \xi_t$
end for

Remark 12 *The algorithm does not know the variance* σ^2 *of the noise, but uses an upper estimate* $\sigma^2_{+,t-1}$.

Remark 13 *We assume that the set of arms* $\mathbb{X}$ *is* discrete. *This is merely for practical reasons since otherwise updating the estimate of* $f_\star$ *in a RKHS requires memory and computational times that are unbounded with* t. *This also simplifies the analysis.*

The following regret bound can then be obtained after some careful but easy adaptation of Agrawal and Goyal (2013). We provide the proof of this result in Appendix C, which can be of independent interest, being a more rigorous and somewhat simpler rewriting of the original proof technique from Agrawal and Goyal (2013).

Theorem 5 (Regularized Kernel TS with variance estimate) *Assume that the maximal instantaneous pseudo-regret* $R = \max_{x\in\mathbb{X}}\big(f_\star(x_\star) - f_\star(x)\big)$ *is finite. Then, the regret of Kernel TS (Algorithm 1) with* $v_t = \frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma_{+,t-1}}$ *after* T *episodes is* $\mathcal{O}(C\sqrt{T\ln(T|\mathbb{X}|)}\gamma_T(\sigma^2/C^2))$

with probability $1-3\delta$. *More precisely, with probability* $1-3\delta$, *the regret is bounded for all* $T \geqslant 0$:

$$\mathfrak{R}_T \;\leqslant\; C_{1,T}\Bigg(\sum_{t=1}^{T}\sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}}B_{\lambda_t,t-1}(\delta/4)\Bigg) + C_2R\sqrt{T\ln(1/\delta)} + 4\pi eR\delta\,,$$

where $C_{1,T} = (4\sqrt{\pi e}+1)\Big(1+\sqrt{2\ln\Big(\frac{T(T+1)|\mathbb{X}|}{\sqrt{\pi}\delta}\Big)}\Big)$ *and* $C_2 = \sqrt{8\pi e(1+\delta\sqrt{4\pi e})^2}$.

Further, we have

$$\mathfrak{R}_T \;\leqslant\; C_{1,T}\frac{\sigma_+}{\sigma}\Big(1+\sqrt{2\ln(4/\delta)+2\gamma_T(\sigma_-^2/C^2)}\Big)C\sqrt{T\frac{2\gamma_T(\sigma^2/C^2)}{\ln(1+C^2/\sigma^2)}} + C_2R\sqrt{T\ln(1/\delta)} + 4\pi eR\delta\,.$$

Remark 14 *As our confidence intervals do not require a bounded noise, likewise we can control the regret with high probability without requiring bounded observations, contrary to earlier works such as Valko et al. (2013).*

5. Discussion and related works

Concentration results Theorem 1 extends the self-normalized bounds of Abbasi-Yadkori et al. (2011) from the setting of linear function spaces to that of an RKHS with sub-Gaussian noise. Based on a nontrivial adaptation of the Laplace method, it yields self-normalized inequalities in a setting of possibly infinite dimension. It generalizes the following result of Wang and de Freitas (2014) to kernel regression with $\lambda \neq \sigma^2$, which was already a generalization of a previous result by Srinivas et al. (2010) for bounded noise. It is also more general than the concentration result from Valko et al. (2013), for kernel regression with $\lambda \neq \sigma^2$, which holds *under the assumption of bounded observations.*

Lemma 3 (Proposition 1 from Wang and de Freitas (2014)) *Let* $f_\star$ *denote a function in the RKHS* $\mathcal{K}$ *induced by kernel* k *and let us define the posterior mean and variances with* $\lambda = \sigma^2$, *for (arbitrary) data* $(x_{t'})_{t'\leqslant t}$. *Assuming* σ*-sub-Gaussian noise variables, then for all* $\delta' \in (0,1)$ *we have that*

$$\mathbb{P}\big[\exists x \in \mathcal{X} : |f_{\lambda,t}(x) - f_\star(x)| \geqslant \ell_{\lambda,t+1}(\delta')k_{\lambda,t}^{1/2}(x,x)\big] \leqslant \delta', \quad \textit{where}$$

$$\ell_{\lambda,t}^2(\delta') = \|f_\star\|_{\mathcal{K}}^2 + \sqrt{8\gamma_{t-1}(\lambda)\ln\frac{2}{\delta'}} + \sqrt{2\ln\frac{4}{\delta'}}\|f_\star\|_{\mathcal{K}} + 2\gamma_{t-1}(\lambda) + 2\sigma\ln\frac{2}{\delta'}$$

and $\gamma_t(\lambda) = \frac{1}{2}\sum_{t'=1}^{t}\ln\big(1+\frac{1}{\lambda}k_{\lambda,t'-1}(x_{t'},x_{t'})\big)$ *is the information gain.*

Remark 15 *This results provides a bound that is valid for each* t, *with probability higher than* $1-\delta$. *In contrast, results from Abbasi-Yadkori et al. (2011), as well as Theorem 1 hold with probability higher than* $1-\delta$, *uniformly for all* t, *and are thus much stronger in this sense.*

Theorem 2 extends Theorem 1 to the case when *the regularization is tuned online* based on gathered observations. To the best of our knowledge, no such result exists in the literature at the time of writing this paper. Moreover, Theorem 3 provides variance estimates with confidence bounds scaling with $1/\sqrt{t}$, in the spirit of the results from Maurer and Pontil (2009), that were provided in the i.i.d. case. Thus, Theorem 3 also appears to be new. Finally, Corollary 1 further specifies Theorem 2 to the situation where the regularization is tuned according to Theorem 3, yielding a fully adaptive regularization procedure with explicit confidence bounds.

Bandits optimization When applied to the setting of multi-armed bandits, Theorems 5 and 4 respectively extend linear TS (Agrawal and Goyal, 2013; Abeille and Lazaric, 2017) and UCB (Li et al., 2010; Chu et al., 2011) to the RKHS setting. Similar extensions have been provided in the literature: GP-UCB (Srinivas et al., 2010) generalizes UCB from the linear to the RKHS setting through the use of Gaussian processes; this corresponds to the case when $\lambda = \sigma^2$. The bounds they provide in the case when the target function belongs to an RKHS is however quite loose. KernelUCB (Valko et al., 2013) also generalizes UCB from the linear to the RKHS setting through the use of kernel regression. However the analysis of this algorithm was out of reach of their proof technique (that requires independence between arms) and they analyze instead the arguably less appealing variant called SupKernelUCB. Also, the analysis of both GP-UCB and SupKernelUCB in the agnostic setting are respectively limited to bounded noise and bounded observations.

6. Illustrative numerical experiments

In this section, we illustrate the results introduced in the previous Sections 2 and 3 on a few examples. The first one is the concentration result on the mean from Theorem 1, the second one is the variance estimate from Theorem 3, and the last one combines the formers by using the noise estimate to tune $\lambda_{t+1} = \sigma_t^2/C^2$ in Theorem 2, which corresponds to Corollary 1. We finally show the performance of kernelized bandits techniques using the provided variance estimates and adaptative regularization schemes.

We conduct the experiments using the function $f_\star$ shown by Figure 1, which has norm $\|f_\star\|_{\mathcal{K}} = \|\theta_\star\|_2 = 2.06$ in the RKHS induced by a Gaussian kernel $\quad k(x,x') = e^{-\frac{(x-x')^2}{2\rho^2}}$ with length scale $\rho = 0.3$. This function results from the linear product between features $\varphi(x)$, explicited using a Taylor expansion[2], and a randomly generated parameter vector $\theta_\star$. We consider the space $\mathcal{X} = [0,1]$ and zero-centered Gaussian noise with $\sigma = 0.1$. All further experiments use the upper-bound $C = 5$ on $\|f_\star\|_{\mathcal{K}}$ and the lower-bound $\sigma_- = 0.01$ on σ.

6.1 Kernel concentration bound with fixed regularization

The following experiments compare the concentration result given by Theorem 1 with the kernel concentration bounds from Wang and de Freitas (2014) reported by Lemma 3. The true noise $\sigma = 0.1$ is assumed to be known and all observations are uniformly sampled from $\mathcal{X}$. In both cases, we use a fixed confidence level $\delta = 0.1$. Figure 2 shows that for $\lambda = \sigma^2$, the

2. If $x \in \mathbb{R}$, the i-th feature of a Gaussian kernel $\varphi(x) = e^{-\frac{x^2}{2\rho^2}}\frac{x^{i-1}}{\rho^{i-1}\sqrt{(i-1)!}}$ (Cotter et al., 2011).

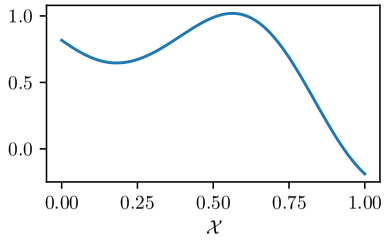


Figure 1: Test function $f_\star$ used in the following numerical experiments.

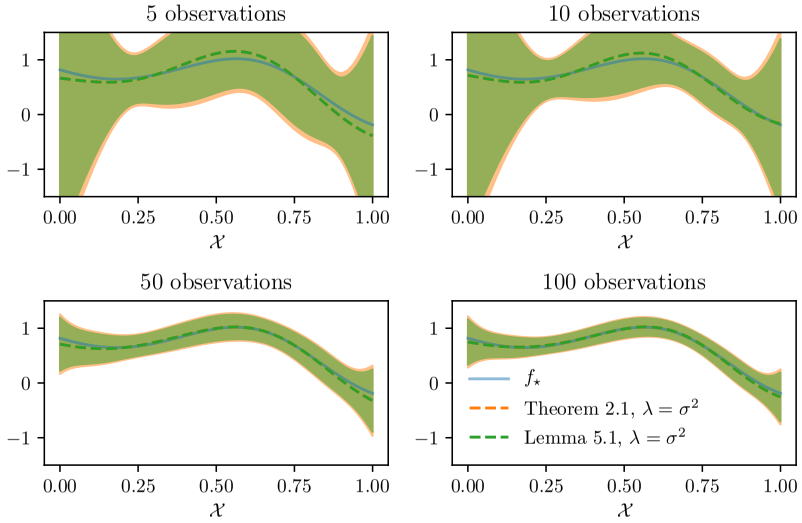


Figure 2: Confidence interval of Theorem 1 and Lemma 3 (Wang and de Freitas, 2014).

result given by Theorem 1 recovers the confidence envelope of Wang and de Freitas (2014). Note however that the confidence bound that we plot for Theorem 1 are valid *uniformly* over all time steps, while the one derived from Wang and de Freitas (2014) is only valid separately for each time. Further, Theorem 1 generalizes the latter result to the case where $\lambda \neq \sigma^2$. For illustration, Figure 3 illustrates the confidence envelopes in the special case where $\lambda = \sigma^2/C^2$, which also shows the potential benefit of such a tuning.

6.2 Empirical variance estimate

We now illustrate the convergence rate of the noise estimates $\sigma_{-,t} = \max\{\sigma_{-,t}(\lambda), \sigma_{-,t-1}\}$ and $\sigma_{+,t} = \min\{\sigma_{+,t}(\lambda, \lambda_-), \sigma_{+,t-1}\}$ computed using Theorem 3, where $\lambda_- = \sigma^2_{-,t}/C^2$

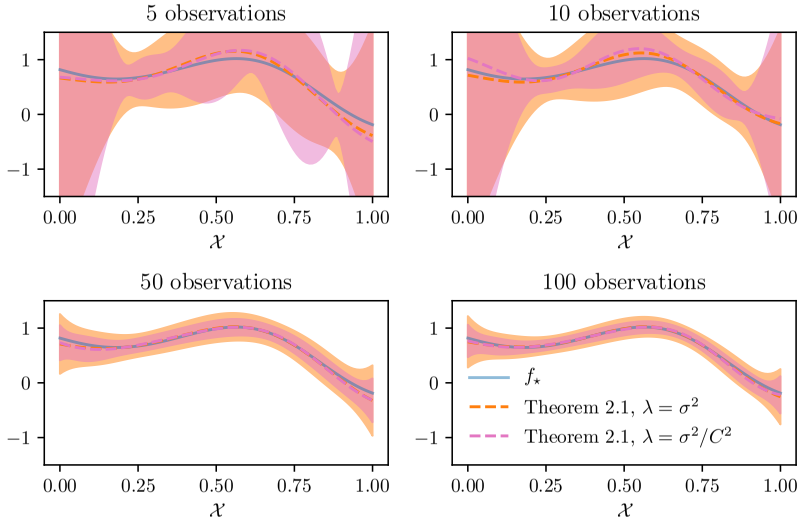


Figure 3: Confidence interval of Theorem 1 for different λ.

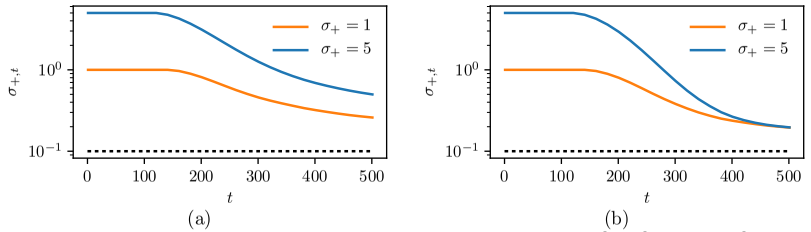


Figure 4: Noise estimate from Theorem 3 with σ_+ for a) fixed $\lambda = \sigma_+^2/C^2$; b) $\lambda = \sigma^2_{+,t-1}/C^2$. Dotted line indicates σ.

and $\delta = 0.1$. All observations are uniformly sampled from $\mathcal{X}$. Section 3 suggests that $\lambda = \sigma^2_{+,t-1}/C^2$ should provide tighter bounds than a fixed $\lambda = \sigma_+^2/C^2$. Figure 4 shows that this is indeed the case especially for large values of t. We also see that the adaptive update of λ converges to the same value, whatever the initial bound σ_+. This is especially interesting when σ_+ is a loose initial upper bound on σ.

In practice, the bound of Theorem 3 not using the knowledge of σ_+ may be useful even when σ_+ is known. This is illustrated by Figure 5a that plots the upper-bound variance estimate $\sigma_{+,t}(\lambda, \lambda_-)$ for $\lambda = \sigma^2_{+,t-1}/C^2$ in both cases. In practice, we suggest to use the minimum of the bound using the knowledge of σ_+ (case 1) and of the agnostic one (case 2) to

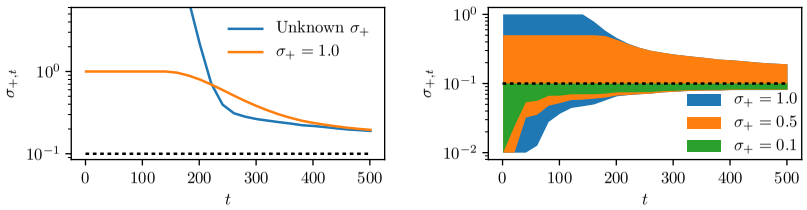


Figure 5: Variance estimate a) from Theorem 3, with and without σ_+; b) as minimum of the bounds and σ_+, for different upper-bounds. Dotted line indicates σ.

set $\sigma_{+,t}(\lambda, \lambda_-)$ and the maximum for $\sigma_{-,t}(\lambda)$. Figure 5b shows the resulting noise estimate envelopes for different σ_+ values (recall that $\sigma = 0.1$).

6.3 Kernel concentration bound with adaptive regularization

We now combine the previous experiments and use the estimated noise in order to tune the regularization. Recall that we consider $\sigma_{-,0} = \sigma_-$, $\sigma_{+,0} = \sigma_+$, and $\lambda_0 = \sigma_+^2/C^2$. On each time $t \geqslant 1$, we estimate the noise lower-bound $\sigma_{-,t} = \max\{\sigma_{-,t}(\lambda_{t-1}), \sigma_{-,t-1}\}$ using Theorem 3 and set $\lambda_- = \sigma_{-,t}^2/C^2$. We then compute the upper-bound noise estimate $\sigma_{+,t} = \min\{\sigma_{+,t}(\lambda_{t-1}, \lambda_-), \sigma_{+,t-1}\}$ using Theorem 3 and set $\lambda_t = \sigma_{+,t}^2/C^2$. We are now ready to compute the confidence interval given by Corollary 1. Note that $\delta = 0.1$ is used everywhere and all observations are uniformely sampled from $\mathcal{X}$. Figure 6 illustrates the resulting confidence envelope of this fully empirical model for noise upper-bound $\sigma_+ = 1$ (recall that the noise satisfies $\sigma = 0.1$) plotted against the confidence envelope obtained with Theorem 1 with fixed $\lambda = \sigma_+^2/C^2$. We observe the improvement of the confidence intervals with the number of observations. Recall that this setting is especially challenging since the variance is unknown, the regularization parameter is tuned online, and the confidence bounds are valid uniformly over all time steps.

6.4 Kernelized bandits optimization

In this section, we now evaluate the potential of kernelized bandits algorithms with variance estimate. We consider $\mathbb{X}$ as the linearly discretized space $\mathcal{X} = [0, 1]$ into 100 arms. Recall that the goal is to minimize the cumulative regret (Equation 4) and that we are optimizing the function shown by Figure 1 with $\sigma = 0.1$. We evaluate Kernel UCB (Equation 5) and Kernel TS (Algorithm 1 with $v_t = B_{\lambda_t, t-1}(\delta)/\sigma_{+,t-1}$) with three different configurations:

a) the oracle, that is with fixed $\lambda_t = \sigma^2/C^2$, assuming knowledge of σ;
b) the fixed $\lambda_t = \sigma_+^2/C^2$, that is the best one can do without prior knowledge of σ^2;
c) the adaptative regularization tuned with Corollary 1.

All configurations use $C = 5$. Kernel UCB uses $\delta = 0.1/4$ and Kernel TS uses $\delta = 0.1/12$ such that their regret bounds respectively hold with probability 0.9. Recall that observations are now sampled from $\mathbb{X}$ using the bandits algorithms (they are not i.i.d.). Configurations b) and c) use $\sigma_+ = 1$, while the oracle a) uses $\sigma_+ = \sigma$. Figure 7 shows the cumulative regret

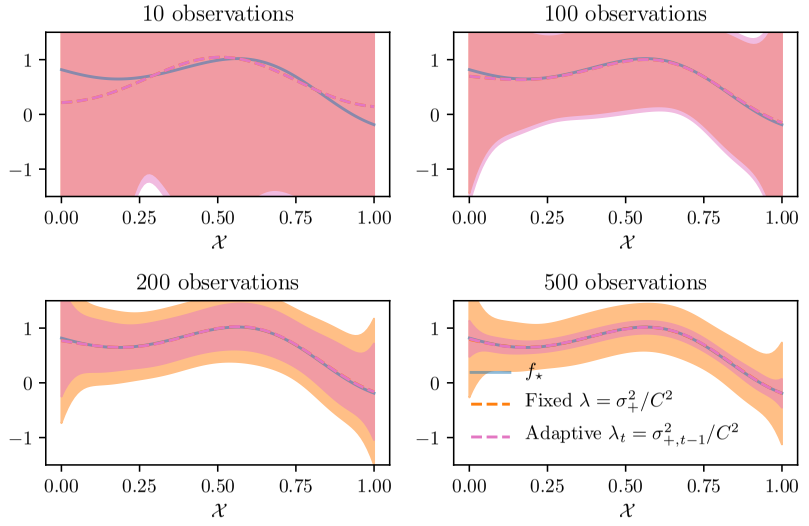


Figure 6: Confidence interval using fixed (Theorem 1) and adaptive (Corollary 1) regularization, for $\sigma_+ = 1$ and $\delta = 0.1$.

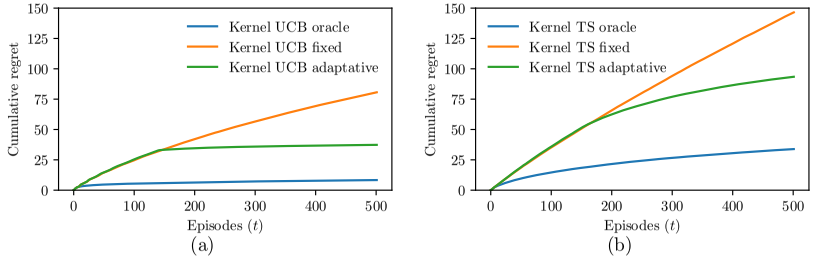


Figure 7: Averaged cumulative regret along episodes for a) Kernel UCB and b) Kernel TS.

averaged over 100 repetitions. Note that the oracle corresponds to the best performance that could be expected by Kernel UCB and Kernel TS given knowledge of the noise. The plots confirm that adaptively tuning the regularization using the variance estimates can lead to a major improvement compared to using a fixed, non-accurate guess: after an initial burn-in phase, the regret of the adaptively tuned algorithm increases at the same rate as that of the oracle algorithm knowing the noise exactly. The fact that Kernel UCB outperforms Kernel TS much implies that inflating the variance in Kernel TS, as suggested per the theory presented previously, may not be optimal in practice. Further attention should be given to this question.

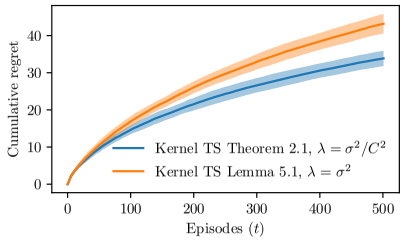


Figure 8: Averaged cumulative regret and one standard deviation along episodes for Kernel TS oracle with Theorem 1 and Lemma 3 (Wang and de Freitas, 2014).

In order to evaluate the benefit of the concentration bound provided by Theorem 1, we compare the Kernel TS (Algorithm 1) oracle using $v_t = B_{\lambda,t-1}/\sigma$ and $\lambda = \sigma^2/C^2$, where $B_{\lambda,t-1}$ is given by Theorem 1, against $v_t = \ell_t(\delta)$ where $\ell_t(\delta)$ is given by Lemma 3 (Wang and de Freitas, 2014) with $\delta = 0.1$. Figure 8 shows that the concentration bound given by Theorem 1 improves the performance of Kernel TS compared with existing concentration results (Wang and de Freitas, 2014). It highlights the relevance of expliciting the regularization parameter, which allows us to take advantage of regularization rates that may be better adapted.

7. Conclusion

This work addresses two problems: the online tuning of the regularization parameter in streaming kernel regression and the online estimation of the noise variance. To this extent, we introduce novel concentration bounds on the posterior mean estimate in streaming kernel regression with fixed and explicit regularization (Theorem 1), which we then extend to the setting where the regularization parameter is tuned (Theorem 2). We further introduce upper- and lower-bound estimates of the noise variance (Theorem 3). Putting these tools together, we show how the estimate of the noise variance can be used to tune the kernel regularization in an online fashion (Corollary 1) while retaining theoretical guarantees. We also show how to use the proposed results in order to derive kernelized variations of the most common bandits algorithms UCB and Thompson sampling, for which regret bounds are also provided (Theorems 4 and 5).

All the proposed results and tools are illustrated through numerical experiments. The obtained results show the relevance of the introduced kernel regression concentration intervals for explicit regularization, which hold when the regularization does not correspond to the noise variance. The potential of the proposed regularization tuning procedure is illustrated through the application to kernelized bandits, where the benefits of adaptive regularization is undeniable when the noise variance is unknown (this is usually the case in practice). Finally, one must note that a major strength of the tools proposed in this work is to allow for an adaptively tuned regularization parameter while preserving theoretical guarantees, which is not the case when regularization is tuned for example by cross-validation.

Future work includes a natural extension of these techniques to obtain an empirical estimate of the kernel length scales. This information is often assumed to be known, while in practice it is often not available. Although some preliminary work has been done in that direction (Wang and de Freitas, 2014), designing theoretically motivated algorithms addressing these concerns would help to fill an important gap between theory and practice. On a different matter, the current work gives the basis for performing Thompson sampling in RKHS, and could be extended to the contextual setting in a near future, as was done with CGP-UCB (Krause and Ong, 2011; Valko et al., 2013).

Acknowledgments

This work was supported through funding from the Natural Sciences and Engineering Research Council of Canada (NSERC, Canada), the REPARTI strategic network (FRQ-NT, Québec), MITACS, and E Machine Learning Inc. O.-A. M. acknowledges the support of the French Agence Nationale de la Recherche (ANR), under grant ANR-16-CE40-0002 (project BADASS), Inria Lille – Nord europe, CPER Nord-Pas de Calais/FEDER DATA Advanced data science and technologies 2015-2020, and the French Ministry of Higher Education and Research.

Appendix A. Laplace method for tuned kernel regression

In this section, we want to control the term $|f_{\lambda,t}(x) - f_\star(x)|$ simultaneously over all $t \leqslant T$. To this end, we resort to a version of the Laplace method carefully extended to the RKHS setting.

Before proceeding, we note that since $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a kernel function (that is continuous, symmetric positive definite) on a compact set $\mathcal{X}$ equipped with a positive finite Borel measure μ, then there is an at most countable sequence $(\sigma_i, \psi_i)_{i\in\mathbb{N}^\star}$ where $\sigma_i \geqslant 0$, $\lim_{i\to\infty} \sigma_i = 0$ and $\{\psi_i\}$ form an orthonormal basis of $L_{2,\mu}(\mathcal{X})$, such that

$$k(x,y) = \sum_{j=1}^{\infty} \sigma_j \psi_j(x)\psi_j(y') \quad \text{and} \quad \|f_\star\|_{\mathcal{K}}^2 = \sum_{j=1}^{\infty} \frac{\langle f, \psi_j\rangle_{L_{2,\mu}}^2}{\sigma_j}$$

Let $\varphi_i = \sqrt{\sigma_i}\psi_i$. Note that $\|\varphi_i\|_{L_2} = \sqrt{\sigma_i}$, $\|\varphi_i\|_{\mathcal{K}} = 1$. Further, if $f = \sum_i \theta_i\varphi_i$, then $\|f_\star\|_{\mathcal{K}}^2 = \sum_i \theta_i^2$ and $\|f_\star\|_{L_2}^2 = \sum_i \theta_i^2\sigma_i$. In particular f belongs to the RKHS if and only if $\sum_i \theta_i^2 < \infty$. For $\varphi(x) = (\varphi_1(x), \dots)$ and $\theta = (\theta_1, \dots)$, we now denote $\theta^\top\varphi(x)$ for $\sum_{i\in\mathbb{N}} \theta_i\varphi_i(x)$, by analogy with the finite dimensional case. Note that $k(x,y) = \varphi(x)^\top\varphi(y)$.

In the sequel, the following Martingale control will be a key component of the analysis.

Lemma 4 (Hilbert Martingale Control) *Assume that the noise sequence $\{\xi_t\}_{t=0}^{\infty}$ is conditionally σ^2-sub-Gaussian*

$$\forall t \in \mathbb{N}, \forall \gamma \in \mathbb{R}, \quad \ln \mathbb{E}[\exp(\gamma\xi_t)|\mathcal{H}_{t-1}] \leqslant \frac{\gamma^2\sigma^2}{2}\,.$$

Let τ be a stopping time with respect to the filtration $\{\mathcal{H}_t\}_{t=0}^{\infty}$ generated by the variables $\{x_t, \xi_t\}_{t=0}^{\infty}$. For any $\mathbf{q} = (q_1, q_2, \dots)$ such that $\mathbf{q}^\top \varphi_i(x) = \sum_{i\in\mathbb{N}} q_i \varphi(x) < \infty$, and deterministic positive λ, let us denote

$$M_{m,\lambda}^{\mathbf{q}} = \exp\left(\sum_{t=1}^{m} \frac{\mathbf{q}^\top \varphi(x_t)}{\sqrt{\lambda}}\xi_t - \frac{\sigma^2}{2}\sum_{t=1}^{m}\frac{(\mathbf{q}^\top\varphi(x_t))^2}{\lambda}\right)$$

Then, for all such $\mathbf{q}$ the quantity $M_{\tau,\lambda}^{\mathbf{q}}$ is well defined and satisfies

$$\ln \mathbb{E}[M_{\tau,\lambda}^{\mathbf{q}}] \leqslant 0\,.$$

Proof The only difficulty in the proof is to handle the stopping time. Indeed, for all $m \in \mathbb{N}$, thanks to the conditional R-sub-Gaussian property, it is immediate to show that $\{M_{m,\lambda}^{\mathbf{q}}\}_{m=0}^{\infty}$ is a non-negative super-martingale and actually satisfies $\ln \mathbb{E}[M_{m,\lambda}^{\mathbf{q}}] \leqslant 0$.

By the convergence theorem for nonnegative super-martingales, $M_{\infty}^{\mathbf{q}} = \lim_{m\to\infty} M_{m,\lambda}^{\mathbf{q}}$ is almost surely well-defined, and thus $M_{\tau,\lambda}^{\mathbf{q}}$ is well-defined (whether $\tau < \infty$ or not) as well. In order to show that $\ln \mathbb{E}[M_{\tau,\lambda}^{\mathbf{q}}] \leqslant 0$, we introduce a stopped version $Q_m^{\mathbf{q}} = M_{\min\{\tau,m\},\lambda}^{\mathbf{q}}$ of $\{M_{m,\lambda}^{\mathbf{q}}\}_m$. Now $\mathbb{E}[M_{\tau,\lambda}^{\mathbf{q}}] = \mathbb{E}[\liminf_{m\to\infty} Q_m^{\mathbf{q}}] \leqslant \liminf_{m\to\infty} \mathbb{E}[Q_m^{\mathbf{q}}] \leqslant 1$ by Fatou's lemma, which concludes the proof. We refer to (Abbasi-Yadkori et al., 2011) for further details. ■

We are now ready to prove the following result.

Proof of Theorem 2 (Streaming Kernel Least-Squares) We make use of the features in an explicit way. Let $\lambda = \lambda_{t+1}$. For $f_\star \in \mathcal{K}$, we denote $\theta^\star$ its corresponding parameter sequence. We let $\mathbf{\Phi}_t = (\varphi(x_{t'}))_{t'\leqslant t}$ be a $t\times\infty$ matrix built from the features and introduce the bi-infinite matrix $V_{\lambda,t} = I + \frac{1}{\lambda}\mathbf{\Phi}_t^\top\mathbf{\Phi}_t$ as well as the noise vector $E_t = (\xi_1, \dots, \xi_t)$. In order to control the term $|f_{\boldsymbol{\lambda},t} - f_\star(x)|$, we first decompose the estimation term. Indeed, using the feature map, it holds that

$$\begin{aligned}
f_{\boldsymbol{\lambda},t}(x) &= k_t(x)^\top(\mathbf{K_t} + \lambda I_t)^{-1}Y_t \\
&= \varphi(x)^\top\mathbf{\Phi}_t^\top(\mathbf{\Phi}_t\mathbf{\Phi}_t^\top + \lambda I_t)^{-1}Y_t \\
&= \varphi(x)^\top\mathbf{\Phi}_t^\top\left(\frac{I_t}{\lambda} - \frac{1}{\lambda}\mathbf{\Phi}_t\big(\lambda I + \mathbf{\Phi}_t^\top\mathbf{\Phi}_t\big)^{-1}\mathbf{\Phi}_t^\top\right)Y_t \\
&= \varphi(x)^\top(\mathbf{\Phi}_t^\top\mathbf{\Phi}_t + \lambda I)^{-1}\mathbf{\Phi}_t^\top(\mathbf{\Phi}_t\theta^\star + E_t)
\end{aligned}$$

where in the third line, we used the Shermann-Morrison formula. From this, simple algebra yields

$$f_{\boldsymbol{\lambda},t}(x) - f^\star(x) = \frac{1}{\lambda}\varphi(x)^\top V_{\lambda,t}^{-1}\big(\mathbf{\Phi}_t^\top E_t - \lambda\theta^\star\big)\,.$$

We then obtain, from a simple Hölder inequality using the appropriate matrix norm, the following decomposition, that is valid provided that all terms involved are finite.

$$|f_{\boldsymbol{\lambda},t}(x) - f(x)| \leqslant \frac{1}{\sqrt{\lambda}}\|\varphi(x)\|_{V_{\lambda,t}^{-1}}\left[\frac{1}{\sqrt{\lambda}}\|\mathbf{\Phi}_t^\top E_t\|_{V_{\lambda,t}^{-1}} + \sqrt{\lambda}\|\theta^\star\|_{V_{\lambda,t}^{-1}}\right]$$

Now, we note that a simple application of the Shermann-Morrison formula yields

$$\|\varphi(x)\|_{V_{\lambda,t}^{-1}}^2 = k_t(x,x)\,.$$

On the other hand, the last term of the bound is controlled as $\|\theta^\star\|_{V_{\lambda,t}^{-1}} \leqslant \|\theta^\star\|$. Thus,

$$|f_{\boldsymbol{\lambda},t}(x) - f(x)| \leqslant \frac{k_{\boldsymbol{\lambda},t}^{1/2}(x,x)}{\sqrt{\lambda_{t+1}}}\left[\frac{1}{\sqrt{\lambda_{t+1}}}\|\mathbf{\Phi}_t^\top E_t\|_{V_{\lambda_{t+1},t}^{-1}} + \sqrt{\lambda_{t+1}}\|\theta^\star\|_2\right].$$

In order to control the remaining term, $\frac{1}{\sqrt{\lambda_{t+1}}}\|\mathbf{\Phi}_t^\top E_t\|_{V_{\lambda_{t+1},t}^{-1}}$, for all t, we now want to apply Lemma 4. However, the lemma does not apply since λ_{t+1} is $\mathcal{H}_t$-measurable. Thus, before proceeding, we upper-bound it by the similar expression involving $\lambda_\star$:

$$\begin{aligned}
\frac{1}{\lambda}\|\mathbf{\Phi}_t^\top E_t\|_{V_{\lambda,t}^{-1}}^2 &= E_t^\top\frac{\mathbf{\Phi}_t^\top}{\lambda}(I + \frac{1}{\lambda}\mathbf{\Phi}_t^\top\mathbf{\Phi}_t)^{-1}\frac{\mathbf{\Phi}_t}{E}_t \\
&= E_t^\top\mathbf{\Phi}_t^\top(\lambda I + \mathbf{\Phi}_t^\top\mathbf{\Phi}_t)^{-1}\mathbf{\Phi}_t E_t \\
&\leqslant E_t^\top\mathbf{\Phi}_t^\top(\lambda_\star I + \mathbf{\Phi}_t^\top\mathbf{\Phi}_t)^{-1}\mathbf{\Phi}_t E_t\,,
\end{aligned}$$

where in the last line, we use the fact that the function $f : \lambda \to u^\top(\lambda I + A)^{-1}u$, for $u = \mathbf{\Phi}_t E_t$ and $A = \mathbf{\Phi}_t^\top\mathbf{\Phi}_t$ is non increasing (see Lemma 5 below). Thus, $\frac{1}{\sqrt{\lambda_{t+1}}}\|\mathbf{\Phi}_t^\top E_t\|_{V_{\lambda_{t+1},t}^{-1}} \leqslant \frac{1}{\sqrt{\lambda_\star}}\|\mathbf{\Phi}_{\lambda_\star,t}^\top E_t\|_{V_{\lambda_\star,t}^{-1}}$. Next, we introduce a random stopping time τ, to be defined later and apply Lemma 4.

More precisely, let $Q \sim \mathcal{N}(0, I)$ be an infinite Gaussian random sequence which is independent of all other random variables. We denote $Q^\top\varphi(x) = \sum_{i\in\mathbb{N}} Q_i\varphi_i(x)$. For all x, $k(x,x) = \sum_{i\in\mathbb{N}}\varphi_i^2(x) < \infty$ and thus $\mathbb{V}(Q^\top\varphi(x)) < \infty$. We define $M_{m,\lambda_\star} = \mathbb{E}[M_{m,\lambda_\star}^Q]$. Clearly, we still have $\mathbb{E}[M_{\lambda_\star,\tau}] = \mathbb{E}[\mathbb{E}[M_{m,\lambda_\star}^Q]|Q] \leqslant 1$. Since $V_{\lambda_\star,\tau} = I + \frac{1}{\lambda_\star}\mathbf{\Phi}_\tau^\top\mathbf{\Phi}_\tau$, elementary algebra gives

$$\begin{aligned}
\det(V_{\lambda_\star,\tau}) &= \det(V_{\lambda_\star,\tau-1} + \frac{1}{\lambda_\star}\varphi(x_\tau)\varphi(x_\tau)^\top) = \det(V_{\lambda_\star,\tau-1})(1 + \frac{1}{\lambda_\star}\|\varphi(x_\tau)\|_{V_{\lambda_\star,\tau-1}^{-1}}^2) \\
&= \det(V_{\lambda_\star,0})\prod_{t'=1}^{\tau}\left(1 + \frac{1}{\lambda_\star}\|\varphi(x_{t'})\|_{V_{\lambda_\star,t'-1}^{-1}}^2\right),
\end{aligned}$$

where we used the fact that the eigenvalues of a matrix of the form $I + xx^\top$ are all ones except for the eigenvalue $1 + \|x\|^2$ corresponding to x. Then, note that $\det(V_{\lambda_\star,0}) = 1$ and thus

$$\begin{aligned}
\ln(\det(V_{\lambda_\star,\tau})) &= \sum_{t'=1}^{\tau}\ln(1 + \frac{1}{\lambda_\star}\|\varphi(x_{t'})\|_{V_{\lambda_\star,t'-1}^{-1}}^2) \\
&= \frac{1}{2}\sum_{t'=1}^{\tau}\ln\left(1 + \frac{1}{\lambda_\star}k_{\lambda_\star,t'-1}(x_{t'},x_{t'})\right).
\end{aligned}$$

In particular, $\ln(\det(V_{\lambda_\star,\tau}))$ is finite. The only difficulty in the proof is now to handle the possibly infinite dimension. To this end, it is enough to take a look at the approximations

using the d first dimension of the sequence for each d. We note $Q_d, M_{\lambda_\star,\tau}, \Phi_{\tau,d}$ and $V_{\tau,d}$ the restriction of the corresponding quantities to the components $\{1,\dots,d\}$. Thus Q_d is Gaussian $\mathcal{N}(0, I_d)$. Following the steps from Abbasi-Yadkori et al. (2011), we obtain that

$$M_{m,d,\lambda_\star} = \frac{1}{\det(V_{\lambda_\star,m,d})^{1/2}} \exp\left(\frac{1}{2\sigma^2\lambda_\star}\|\mathbf{\Phi}_{m,d}^\top E_m\|^2_{V^{-1}_{\lambda_\star,m,d}}\right).$$

Note also that $\mathbb{E}[M_{\tau,d,\lambda_\star}] \leqslant 1$ for all $d \in \mathbb{N}$. Thus, we obtain by an application of Fatou's lemma that

$$\begin{aligned}\mathbb{P}\bigg(\lim_{d\to\infty} \frac{\|\mathbf{\Phi}_{\tau,d}^\top E_\tau\|^2_{V^{-1}_{\lambda_\star,\tau,d}}}{2\sigma^2\lambda_\star \log\Big(\det(V_{\lambda_\star,\tau,d})^{1/2}/\delta\Big)} > 1\bigg) &\leqslant \mathbb{E}\bigg[\lim_{d\to\infty} \frac{\delta \exp\left(\frac{1}{2\lambda_\star\sigma^2}\|\mathbf{\Phi}_{\tau,d}^\top E_\tau\|^2_{V^{-1}_{\tau,d}}\right)}{\det(V_{\lambda_\star,\tau,d})^{1/2}}\bigg] \\ &\leqslant \delta \lim_{d\to\infty} \mathbb{E}[M_{\tau,d,\lambda_\star}] \leqslant \delta\,.\end{aligned}$$

We conclude by defining τ following Abbasi-Yadkori et al. (2011), by

$$\tau(\omega) = \min\left\{t \geqslant 0; \omega \in \Omega \text{ s.t. } \|\mathbf{\Phi}_t^\top E_t\|^2_{V^{-1}_{\lambda_\star,t}} > 2\sigma^2\lambda_\star \log\Big(\det(V_{\lambda_\star,t})^{1/2}/\delta\Big)\right\}.$$

Then τ is a random stopping time and

$$\mathbb{P}\bigg(\exists t, \|\mathbf{\Phi}_t^\top E_t\|^2_{V^{-1}_{\lambda_\star,t}} > 2\sigma^2\lambda_\star \log\Big(\det(V_{\lambda_\star,t})^{1/2}/\delta\Big)\bigg) = \mathbb{P}(\tau < \infty) \leqslant \delta.$$

Finally, combining this result with the previous remarks we obtain that with probability higher than $1-\delta$, uniformly over $x \in \mathcal{X}$ and $t \leqslant T$, it holds that

$$|f_{\boldsymbol{\lambda},t} - f_\star(x)| \leqslant \frac{k^{1/2}_{\boldsymbol{\lambda},t}(x,x)}{\sqrt{\lambda_{t+1}}}\left[\sqrt{2\sigma^2 \ln\left(\frac{\det(I + \frac{1}{\lambda_\star}\Phi_t^\top\Phi_t)^{1/2}}{\delta}\right)} + \sqrt{\lambda_{t+1}}\|f_\star\|_{\mathcal{K}}\right].$$

■

Lemma 5 (Technical lemma) *The function $f : \lambda \mapsto u^\top(\lambda I + A)^{-1}u$, where A is a semi-definite positive matrix and u is any vector, is non-decreasing on $\lambda \in \mathbb{R}^+$.*

Proof Indeed, let $h > 0$. By the Sherman-Morrison formula, we obtain

$$f(\lambda+h) = f(\lambda) - hu^\top(\lambda I + A)^{-1}(I + h(\lambda I + A)^{-1})^{-1}(\lambda I + A)^{-1}u\,.$$

Thus, since $\lambda I + A$ is also semi-definite positive, we have

$$\lim_{h\to 0} \frac{f(\lambda+h) - f(\lambda)}{h} = -u^\top(\lambda I + A)^{-1}(\lambda I + A)^{-1}u \leqslant 0\,.$$

■

Appendix B. Variance estimation

In this section, we give the proof of Theorem 3. To this end, we proceed in two steps. First, we provide an upper bound and lower bound on the variance estimate in the next theorem. Then, we use these bounds in order to derive the final statement.

Theorem 6 (Regularized variance estimate) *Under the second-order sub-Gaussian predictable assumption, for any random stopping time τ for the filtration of the past, with probability higher than $1-3\delta$, it holds*

$$\begin{aligned}\sqrt{\widehat{\sigma}^2_{k,\lambda,\tau}} &\leqslant \sigma\left[1 + \sqrt{\frac{2C_\tau(\delta)}{\tau}}\right] + \|f^\star\|_{\mathcal{K}}\sqrt{\frac{\lambda}{\tau}}\sqrt{1 - \frac{1}{\max_{t\leqslant\tau}(1 + k_{\lambda,t-1}(x_t,x_t))}} \\ \sqrt{\widehat{\sigma}^2_{k,\lambda,\tau}} &\geqslant \sigma\left[1 - \sqrt{\frac{C_\tau(\delta)}{\tau}} - \sqrt{\frac{C_\tau(\delta) + 2D_{\lambda_\star,\tau}(\delta)}{\tau}}\right] - \sqrt{\frac{2\sigma\lambda^{1/2}\|f^\star\|_{\mathcal{K}}\sqrt{D_{\lambda_\star,\tau}(\delta)}}{\tau}}\,.\end{aligned}$$

where we introduced for convenience the constants $C_\tau(\delta) = \ln(e/\delta)\big[1+\ln(\pi^2\ln(\tau)/6)/\ln(1/\delta)\big]$ and $D_{\lambda_\star,\tau}(\delta) = 2\ln(1/\delta) + \sum_{t=1}^\tau \ln(1+\frac{1}{\lambda_\star}k_{\lambda_\star,t-1}(x_t,x_t))$.

Proof We use the feature maps and start with the following decomposition

$$\begin{aligned}\tau\widehat{\sigma}^2_{k,\lambda,\tau} &= \sum_{t=1}^\tau (y_t - f_{\lambda,\tau}(x_t))^2 = \sum_{t=1}^\tau (y_t - \langle\theta_{\lambda,\tau}, \varphi(x_t)\rangle)^2 \\ &= (\theta^\star - \theta_{\lambda,\tau})^\top G_\tau(\theta^\star - \theta_{\lambda,\tau}) + \|E_\tau\|^2 + 2(\theta^\star - \theta_{\lambda,\tau})^\top\Phi_\tau^\top E_\tau\,. \qquad (6)\end{aligned}$$

where $\theta^\star - \theta_{\lambda,\tau} = (I - G^{-1}_{\lambda,\tau}G_\tau)\theta^\star - G^{-1}_{\lambda,\tau}\mathbf{\Phi}_\tau^\top E_\tau$ with $G_{\lambda,\tau} = \lambda I + G_\tau$ and $G_\tau = \Phi_\tau^\top\Phi_\tau$.

On the one hand, we can control the first term in (6) via

$$\begin{aligned}&(\theta^\star - \theta_{\lambda,\tau})^\top G_\tau(\theta^\star - \theta_{\lambda,\tau}) \\ &= [(I - G^{-1}_{\lambda,\tau}G_\tau)\theta^\star - G^{-1}_{\lambda,\tau}\Phi_\tau^\top E_\tau]^\top G_\tau[(I - G^{-1}_{\lambda,\tau}G_\tau)\theta^\star - G^{-1}_{\lambda,\tau}\Phi_\tau^\top E_\tau] \\ &= [\lambda\theta^\star - \Phi_\tau^\top E_\tau]^\top G^{-1}_{\lambda,\tau}G_\tau G^{-1}_{\lambda,\tau}[\lambda\theta^\star - \Phi_\tau^\top E_\tau] \\ &= [\lambda\theta^\star - \Phi_\tau^\top E_\tau]^\top [G^{-1}_{\lambda,\tau} - \lambda G^{-2}_{\lambda,\tau}][\lambda\theta^\star - \Phi_\tau^\top E_\tau] \\ &= \|\Phi_\tau^\top E_\tau\|^2_{G^{-1}_{\lambda,\tau}} - \lambda\|\Phi_\tau^\top E_\tau\|^2_{G^{-2}_{\lambda,\tau}} + \lambda^2\|\theta^\star\|^2_{G^{-1}_{\lambda,\tau}} - \lambda^3\|\theta^\star\|^2_{G^{-2}_{\lambda,\tau}} \\ &\quad -2\lambda\theta^{\star\top}[G^{-1}_{\lambda,\tau} - \lambda G^{-2}_{\lambda,\tau}]\Phi_\tau^\top E_\tau\end{aligned}$$

where we used the fact that $I - G^{-1}_{\lambda,\tau}G_\tau = \lambda G^{-1}_{\lambda,\tau}$ and then that $G^{-1}_{\lambda,\tau}G_\tau G^{-1}_{\lambda,\tau} = G^{-1}_{\lambda,\tau} - \lambda G^{-2}_{\lambda,\tau}$. Likewise, we control the third term in (6) via

$$\begin{aligned}2(\theta^\star - \theta_{\lambda,\tau})^\top\Phi_\tau^\top E_\tau &= 2[(I - G^{-1}_{\lambda,\tau}G_\tau)\theta^\star - G^{-1}_{\lambda,\tau}\Phi_\tau^\top E_\tau]^\top\Phi_\tau^\top E_\tau \\ &= 2[\lambda\theta^\star - \Phi_\tau^\top E_\tau]^\top G^{-1}_{\lambda,\tau}\Phi_\tau^\top E_\tau \\ &= 2\lambda\theta^{\star\top}G^{-1}_{\lambda,\tau}\Phi_\tau^\top E_\tau - 2\|\Phi_\tau^\top E_\tau\|^2_{G^{-1}_{\lambda,\tau}}\,.\end{aligned}$$

Combining these two bounds, we have

$$\begin{aligned}
&\sum_{t=1}^{\tau}(y_t-\langle\theta_{\lambda,\tau},\varphi(x_t)\rangle)^2\\
&= \|E_\tau\|^2-\|\Phi_\tau^\top E_\tau\|^2_{G_{\lambda,\tau}^{-1}}-\lambda\|\Phi_\tau^\top E_\tau\|^2_{G_{\lambda,\tau}^{-2}}\\
&\quad+\lambda^2\|\theta^\star\|^2_{G_{\lambda,\tau}^{-1}}-\lambda^3\|\theta^\star\|^2_{G_{\lambda,\tau}^{-2}}+2\lambda^2\theta^{\star\top}G_{\lambda,\tau}^{-2}\Phi_\tau^\top E_\tau\\
&\leqslant \|E_\tau\|^2+\frac{\lambda^2}{\lambda_{\min(G_{\lambda,\tau})}}\|\theta^\star\|_2^2\Big(1-\frac{\lambda}{\lambda_{\max}(G_{\lambda,\tau})}\Big)+2\frac{\lambda^2}{\lambda_{\min^{3/2}(G_{\lambda,\tau})}}\|\theta^\star\|_2\|\Phi_\tau^\top E_\tau\|_{G_{\lambda,\tau}^{-1}}\\
&\geqslant \|E_\tau\|^2+\frac{\lambda^2}{\lambda_{\max}(G_{\lambda,\tau})}\|\theta^\star\|_2^2\Big(1-\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big)-2\frac{\lambda^2}{\lambda_{\min^{3/2}(G_{\lambda,\tau})}}\|\theta^\star\|_2\|\Phi_\tau^\top E_\tau\|_{G_{\lambda,\tau}^{-1}}\\
&\quad-\|\Phi_\tau^\top E_\tau\|^2_{G_{\lambda,\tau}^{-1}}\Big(1+\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big).
\end{aligned}$$

Now, from Lemma 6, it holds on an event Ω_1 of probability higher than $1-\delta$,

$$0\leqslant\|\Phi_\tau^\top E_\tau\|^2_{G_{\lambda,\tau}^{-1}}=\frac{1}{\lambda}\|\Phi_\tau^\top E_\tau\|^2_{V_{\lambda,\tau}^{-1}}\leqslant\frac{1}{\lambda_\star}\|\Phi_\tau^\top E_\tau\|^2_{V_{\lambda_\star,\tau}^{-1}}\leqslant\sigma^2 D_{\lambda_\star,\tau}(\delta)\,.$$

On the other hand, we control the second term $\|E_\tau\|^2$ by Lemma 6 below, and obtain that with probability higher than $1-2\delta$,

$$\begin{aligned}
\|E_\tau\|^2 &\leqslant \tau\sigma^2+2\sigma^2\sqrt{2\tau C_\tau(\delta)}+2\sigma^2C_\tau(\delta)\\
\|E_\tau\|^2 &\geqslant \tau\sigma^2-2\sigma^2\sqrt{\tau C_\tau(\delta)}\,,
\end{aligned}$$

where $C_\tau(\delta)=\ln(e/\delta)(1+c_\tau/\ln(1/\delta))$.

Thus, combining these two results with a union bound, we deduce that with probability higher than $1-3\delta$ it holds that

$$\begin{aligned}
\widehat{\sigma}^2_{\lambda,\tau} &\leqslant \sigma^2+2\sigma^2\sqrt{\frac{2C_\tau(\delta)}{\tau}}+\frac{2\sigma^2C_\tau(\delta)}{\tau}\\
&\quad+\frac{\lambda^2}{\tau\lambda_{\min(G_{\lambda,\tau})}}\|\theta^\star\|_2^2\Big(1-\frac{\lambda}{\lambda_{\max}(G_{\lambda,\tau})}\Big)-2\frac{\sigma\lambda^2}{\tau\lambda_{\min^{3/2}(G_{\lambda,\tau})}}\|\theta^\star\|_2\sqrt{D_{\lambda_\star,\tau}(\delta)}\\
\widehat{\sigma}^2_{\lambda,\tau} &\geqslant \sigma^2-2\sigma^2\sqrt{\frac{C_\tau(\delta)}{\tau}}+\frac{\lambda^2}{\tau\lambda_{\max}(G_{\lambda,\tau})}\|\theta^\star\|_2^2\Big(1-\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big)\\
&\quad-2\frac{\lambda^2\sigma}{\tau\lambda_{\min^{3/2}(G_{\lambda,\tau})}}\|\theta^\star\|_2\sqrt{D_{\lambda_\star,\tau}(\delta)}-\frac{\sigma^2D_{\lambda_\star,\tau}(\delta)}{\tau}\Big(1+\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big).
\end{aligned}$$

We can now derive a bound on $\sqrt{\widehat{\sigma}^2_{\lambda,\tau}}$. Indeed,

$$\begin{aligned}
\widehat{\sigma}^2_{\lambda,\tau} &\leqslant \Big(\sigma+\sqrt{\frac{2\sigma^2C_\tau(\delta)}{\tau}}\Big)^2+\frac{\lambda^2}{\tau\lambda_{\min(G_{\lambda,\tau})}}\|\theta^\star\|_2^2\Big(1-\frac{\lambda}{\lambda_{\max}(G_{\lambda,\tau})}\Big)\\
\widehat{\sigma}^2_{\lambda,\tau} &\geqslant \Big(\sigma-\sqrt{\frac{\sigma^2C_\tau(\delta)}{\tau}}\Big)^2-\frac{\sigma^2}{\tau}\Big(C_\tau(\delta)+D_{\lambda_\star,\tau}(\delta)\Big(1+\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big)\Big)\\
&\quad-\frac{2\lambda^2\sigma}{\tau\lambda_{\min^{3/2}(G_{\lambda,\tau})}}\|\theta^\star\|_2\sqrt{D_{\lambda_\star,\tau}(\delta)}\,.
\end{aligned}$$

Thus, using the inequality $\sqrt{a+b}\leqslant\sqrt{a}+\sqrt{b}$, on both inequalities, we get

$$\begin{aligned}
\sqrt{\widehat{\sigma}^2_{\lambda,\tau}} &\leqslant \sigma+\sigma\sqrt{\frac{2C_\tau(\delta)}{\tau}}+\frac{\lambda\|\theta^\star\|_2}{\sqrt{\tau\lambda_{\min(G_{\lambda,\tau})}}}\sqrt{1-\frac{\lambda}{\lambda_{\max}(G_{\lambda,\tau})}}\\
\sqrt{\widehat{\sigma}^2_{\lambda,\tau}} &\geqslant \sigma-\sigma\sqrt{\frac{C_\tau(\delta)}{\tau}}-\sigma\sqrt{\frac{C_\tau(\delta)+D_{\lambda_\star,\tau}(\delta)\Big(1+\frac{\lambda}{\lambda_{\min(G_{\lambda,\tau})}}\Big)}{\tau}}\\
&\quad-\lambda\sqrt{\frac{2\sigma\|\theta^\star\|_2\sqrt{D_{\lambda_\star,\tau}(\delta)}}{\tau\lambda_{\min^{3/2}(G_{\lambda,\tau})}}}\,.
\end{aligned}$$

■

Corollary 1 (Extension of Corollary 3.13 in Maillard (2016)) *With probability higher than $1-3\delta'$, it holds simultaneously over all $t\geqslant 0$,*

$$\begin{aligned}
\sigma &\leqslant \frac{1}{\alpha^2}\left(\sqrt{\frac{\sqrt{\lambda}\|f_\star\|_{\mathcal{K}}\sqrt{D_{t,\lambda_\star}(\delta')}}{2t}}+\sqrt{\frac{\sqrt{\lambda}\|f_\star\|_{\mathcal{K}}\sqrt{D_{\lambda_\star,t}(\delta')}}{2t}+\widehat{\sigma}_{\lambda,t}\alpha}\right)^2\\
\sigma &\geqslant \left[\widehat{\sigma}_{\lambda,t}-\|f_\star\|_{\mathcal{K}}\sqrt{\frac{\lambda}{t}\Big(1-\frac{1}{\max_{t'\leqslant t}(1+k_{\lambda,t'-1}(x_{t'},x_{t'}))}\Big)}\right]\Big(1+\sqrt{\frac{2C_t(\delta')}{t}}\Big)^{-1},
\end{aligned}$$

where $\alpha=\max\Big(1-\sqrt{\frac{C_t(\delta')}{t}}-\sqrt{\frac{C_t(\delta')+2D_{\lambda_\star,t}(\delta')}{t}},0\Big)$. Further, if an upper bound $\sigma^+\geqslant\sigma$ is known, one can derive the following inequalities that hold with probability higher than $1-3\delta'$,

$$\begin{aligned}
\sigma &\leqslant \widehat{\sigma}_{\lambda,t}+\sigma^+\left(\sqrt{\frac{C_t(\delta')}{t}}+\sqrt{\frac{C_t(\delta')+2D_{\lambda_\star,t}(\delta')}{t}}\right)+\sqrt{\frac{2\sigma^+\lambda^{1/2}\|f_\star\|_{\mathcal{K}}\sqrt{D_{t,\lambda_\star}(\delta')}}{t}}\\
\sigma &\geqslant \widehat{\sigma}_{\lambda,t}-\sigma^+\sqrt{\frac{2C_t(\delta')}{t}}-\|f_\star\|_{\mathcal{K}}\sqrt{\frac{\lambda}{t}\Big(1-\frac{1}{\max_{t'\leqslant t}(1+k_{\lambda,t'-1}(x_{t'},x_{t'}))}\Big)}.
\end{aligned}$$

Proof Using Theorem 6, it holds with high probability that

$$\underbrace{\widehat{\sigma}_{\lambda,\tau}}_{A} \geqslant \sigma\Big[\underbrace{1-\sqrt{\frac{C_\tau(\delta')}{\tau}}-\sqrt{\frac{C_\tau(\delta')+2D_{\lambda_\star,\tau}(\delta')}{\tau}}}_{C}\Big]-\sqrt{\sigma}\underbrace{\sqrt{\frac{2\sqrt{\lambda}\|f_\star\|_{\mathcal{K}}\sqrt{D_{\lambda_\star,\tau}(\delta')}}{\tau}}}_{B}.$$

The inequality rewrites $A \geqslant \sigma C - \sqrt{\sigma}B$. Now, let $y^2 = \sigma$. If $C > 0$, the inequality holds provided that $y \geqslant 0$ and $A + yB - Cy^2 \geqslant 0$, that is when $0 \leqslant y \leqslant \frac{B+\sqrt{B^2+4AC}}{2C}$. We conclude by choosing the stopping time τ corresponding to the probability of *bad events*, as in the proof of Theorem 2, then by remarking that $t \mapsto C_t(\delta')$ is an increasing function. ■

Lemma 6 (Lemma 5.10 from Maillard (2016)) *Assume that T_n is a random stopping time that satisfies $T_n \leqslant n$ almost surely, then*

$$\mathbb{P}\Big[\frac{1}{T_n}\sum_{i=1}^{T_n}\xi_i^2 \geqslant \sigma^2 + 2\sigma^2\sqrt{\frac{2\ln(e/\delta)}{T_n}} + 2\sigma^2\frac{\ln(e/\delta)}{T_n}\Big] \leqslant \Big(\lceil\ln(n)\ln(e/\delta)\rceil\Big)\delta\,,$$

$$\mathbb{P}\Big[\frac{1}{T_n}\sum_{i=1}^{T_n}\xi_i^2 \leqslant \sigma^2 - 2\sigma^2\sqrt{\frac{\ln(e/\delta)}{T_n}}\Big] \leqslant \Big(\lceil\ln(n)\ln(e/\delta)\rceil\Big)\delta\,.$$

Further, for a random stopping time T, and if we introduce $c_T = \ln(\pi^2\ln^2(T)/6)$, then

$$\mathbb{P}\Big[\frac{1}{T}\sum_{i=1}^{T}\xi_i^2 \geqslant \sigma^2 + 2\sigma^2\sqrt{\frac{2\ln(e/\delta)(1+c_T/\ln(1/\delta))}{T}} + 2\sigma^2\frac{\ln(e/\delta)(1+c_T/\ln(1/\delta))}{T}\Big] \leqslant \delta\,,$$

$$\mathbb{P}\Big[\frac{1}{T}\sum_{i=1}^{T}\xi_i^2 \leqslant \sigma^2 - 2\sigma^2\sqrt{\frac{\ln(e/\delta)(1+c_T/\ln(1/\delta))}{T}}\Big] \leqslant \delta\,.$$

Appendix C. Application to stochastic multi-armed bandits

Proof of Lemma 1 Using the facts that $\min\{r,\alpha\} \leqslant (\alpha/\ln(1+\alpha))\ln(1+r)$ and $\min_{\lambda\in\boldsymbol{\lambda}}\lambda \geqslant \sigma^2/C^2$:

$$\begin{aligned}\sum_{t=1}^{T} s^2_{\boldsymbol{\lambda},t-1}(x_t) &= \sigma^2\sum_{t=1}^{T}\frac{1}{\lambda_t}k_{\lambda_t,t-1}(x_t,x_t)\\ &\leqslant \sigma^2\sum_{t=1}^{T}\frac{C^2}{\sigma^2}k_{\sigma^2/C^2,t-1}(x_t,x_t)\\ &= \sigma^2\sum_{t=1}^{T}\min\Big\{\frac{C^2}{\sigma^2}k_{\sigma^2/C^2,t-1}(x_t,x_t),\frac{C^2}{\sigma^2}\Big\}\\ &\leqslant \frac{2C^2}{\ln(1+C^2/\sigma^2)}\gamma_T(\sigma^2/C^2).\end{aligned}$$

In particular, we obtain by a Cauchy-Schwarz inequality,

$$\sum_{t=1}^{T}\sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}} \leqslant \sqrt{T\frac{2C^2/\sigma^2}{\ln(1+C^2/\sigma^2)}\gamma_T(\sigma^2/C^2)}\,.$$

■

Proof of Lemma 2 We want to control the quantity $B_{\lambda_t,t}(\delta)$. First of all, recall from Equation 3 that

$$\begin{aligned}B_{\lambda_t,t}(\delta) &= \sqrt{\lambda_t}C + \sigma_{+,t}\sqrt{2\ln(1/\delta)+2\gamma_t(\lambda_-)}\\ &\leqslant \sigma_+ + \sigma_+\sqrt{2\ln(1/\delta)+2\gamma_t(\sigma^2_{t,-}/C^2)}\,,\end{aligned}$$

where we use the facts that $\lambda_t \leqslant \sigma_+^2/C^2$ and $\lambda_- \geqslant \sigma^2_{t,-}/C^2$. Then, using that $\sigma^2_{t,-} \geqslant \sigma_-$, that $\gamma_t(\cdot)$ is non-increasing and non-decreasing with t, it comes

$$B_{\lambda_t,t}(\delta) \leqslant \sigma_+ + \sigma_+\sqrt{2\ln(1/\delta)+2\gamma_T(\sigma^2_-/C^2)}\,.$$

Alternatively one may use Theorem 6 in order to control the random variables $\sigma_{t,+}$ and $\sigma_{t,-}$ in a tighter way. For instance, by Theorem 6, we easily obtain that with high probability, for all t,

$$\begin{aligned}\sigma \geqslant \sigma_{t,-} \;\geqslant\; \sigma &- \frac{\sigma}{\sqrt{t}}\frac{\Big[(\sqrt{2}+1)\sqrt{C_t(\delta)}-\sqrt{C_t(\delta)+2D_{\lambda_\star,t}(\delta)}\Big]}{1+\sqrt{2C_t(\delta)/t}}\\ &- \frac{\sqrt{2\sigma\lambda^{1/2}\|f^\star\|_{\mathcal{K}}\sqrt{D_{\lambda_\star,t}(\delta)}}+C\sqrt{\lambda}\sqrt{1-\frac{1}{\max_{t\leqslant t}(1+k_{\lambda,t-1}(x_t,x_t))}}}{\sqrt{t}(1+\sqrt{2C_t(\delta)/t})}\,,\end{aligned}$$

that is the estimate satisfies $\sigma \geqslant \sigma_{t,-} \geqslant \sigma - O(1/\sqrt{t})$. This in turns implies that $\gamma_t(\sigma^2_{-,t}/C^2) \leqslant \gamma_t(\sigma^2/C^2) + O(1/\sqrt{t})$. Likewise, it can be shown that $\sigma \leqslant \sigma_{t,+} \leqslant \sigma + O(1/\sqrt{t})$, which yields

$$B_{\lambda_t,t}(\delta) \leqslant \sigma\Big(1+\sqrt{2\ln(1/\delta)+2\gamma_T(\sigma^2_-/C^2)}\Big)+o(1)\,.$$

■

Proof of Theorem 4 (UCB algorithm for kernel bandits) Let r_t denote the instantaneous regret at time t and $f^+(x_t)$ denote the optimistic value at the chosen point x_t, built from the confidence set used by the UCB algorithm. The following holds with probability higher than $1-4\delta$ for each time-step t

$$\begin{aligned}r_t(\lambda_t) &= f_\star(x_\star) - f_\star(x_t) \leqslant f^+_{t-1}(x_t) - f_\star(x_t)\\ &\leqslant |f^+_{t-1}(x_t) - f_{\lambda_t,t-1}(x_t)| + |f_{\lambda_t,t-1}(x_t) - f_\star(x_t)|\\ &\leqslant 2\sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}}B_{\lambda_t,t-1}(\delta)\,.\end{aligned}$$

Thus, we deduce that with probability higher than $1-4\delta$:

$$\mathfrak{R}_T \;=\; \sum_{t=1}^{T} r_t(\lambda) \leqslant 2\sum_{t=1}^{T}\sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}}B_{\lambda_t,t-1}(\delta)\,.$$

We then use Lemma 2 in order to control the term $B_{\lambda_t,t-1}(\delta)$, and Lemma 1 in order to control the sum of $\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}$. This yields the following bound on the regret:

$$\mathfrak{R}_T \;\leqslant\; 2\sigma_+\Big(1+\sqrt{2\ln(1/\delta)+2\gamma_T(\sigma_-^2/C^2)}\Big)\sqrt{T\frac{2C^2/\sigma^2}{\ln(1+C^2/\sigma^2)}\gamma_T(\sigma^2/C^2)}\,.$$

■

Proof of Theorem 5 (TS algorithm for kernel bandits) We closely follow the proof technique of Agrawal and Goyal (2013), while clarifying and simplifying some steps. The general idea is to split the arms into two groups: *saturated arms* and *unsaturated arms*. The former designates arms where samples $\tilde{f}_t$ have low probability of dominating $f_\star(x_\star)$ while the latter designates the other case. This is related to the *optimism* (Abeille and Lazaric, 2017), that is the possibility of sampling a value that is higher than the optimum. Let $\widehat{E}_t$ and $\tilde{E}_t$ be the events that $\tilde{f}_t$ and $\tilde{f}_t$ are concentrated around their respective means. More precisely, for a given confidence level δ, we introduce

$$\begin{aligned}\widehat{E}_{t,\delta} &= \{\forall x\in\mathcal{X}, |f_\star(x)-f_{\lambda_t,t-1}(x)|\leqslant \widehat{C}_{t,\delta}(x)\}\\ \tilde{E}_{t,\delta} &= \{\forall x\in\mathcal{X}, |f_{\lambda_t,t-1}(x)-\tilde{f}_t(x)|\leqslant \tilde{C}_{t,\delta}(x)\}\,,\end{aligned}$$

for some quantities $\widehat{C}_{t,\delta}(x), \tilde{C}_{t,\delta}(x)$ to be defined.

Controlling the event $\widehat{E}_{t,\delta}$ Choosing the confidence bound to be

$$\widehat{C}_{t,\delta}(x) = \sqrt{\frac{k_{\lambda_t,t-1}(x,x)}{\lambda_t}}B_{\lambda_t,t-1}(\delta/4)\,,$$

then the event $\widehat{E}_{t,\delta}$ is controlled as $\mathbb{P}\Big(\forall t\geqslant 0,\ \widehat{E}_{t,\delta}\Big)\geqslant 1-\delta$.

Controlling the event $\tilde{E}_{t,\delta}$ On the other hand, since $\tilde{f}_t(x)|\mathcal{H}_{t-1}=\mathcal{N}(f_{\lambda_t,t-1}(x),\mathbf{V}_t)$ where we introduced the notation $\mathbf{V}_t = v_t^2\frac{\sigma_{+,t-1}^2}{\lambda_t}(k_{\lambda_t,t-1}(x,x'))_{x,x'\in\mathbb{X}}$, then we have by a simple union bound over $x\in\mathbb{X}$,

$$\mathbb{P}(\tilde{E}_{t,\delta}^c|\mathcal{H}_{t-1}) \leqslant \sum_{x\in\mathbb{X}}\frac{1}{\sqrt{\pi}z_x}e^{-z_x^2/2}$$

provided that $z_x = \frac{\tilde{C}_{t,\delta}(x)}{v_t\sqrt{\frac{\sigma_{+,t-1}^2}{\lambda_t}k_{\lambda_t,t-1}(x,x)}} \geqslant 1$ for all $x\in\mathbb{X}$. This motivates the following definition,

$$\tilde{C}_{t,\delta}(x) = c_{t,\delta}v_t\sqrt{\frac{\sigma_{+,t-1}^2}{\lambda_t}k_{\lambda_t,t-1}(x,x)}\,,$$

for a well-chosen sequence $(c_{t,\delta})_t$. The choice $c_{t,\delta}=\max\{\sqrt{2\ln(t(t+1)|\mathbb{X}|/\sqrt{\pi}\delta)},1\}$ ensures that

$$\begin{aligned}\mathbb{P}(\exists t\geqslant 0\ \tilde{E}_{t,\delta}^c|\mathcal{H}_{t-1}) &\leqslant \sum_{t\geqslant 0}\frac{|\mathbb{X}|}{\sqrt{\pi}c_{t,\delta}}e^{-c_{t,\delta}^2/2} = \sum_{t\geqslant 0}\frac{\delta}{c_{t,\delta}t(t+1)}\\ &\leqslant \sum_{t\geqslant 0}\frac{\delta}{t(t+1)} = \delta,\end{aligned}$$

from which we obtain $\mathbb{P}\Big(\forall t\geqslant 0,\ \tilde{E}_{t,\delta}\Big)\geqslant 1-\delta$.

Summary By definition of the events, under $\widehat{E}_{t,\delta}$ and $\tilde{E}_{t,\delta}$, it thus holds that

$$\begin{aligned}\forall x\in\mathcal{X},\ \Big|f_\star(x)-\tilde{f}_t(x)\Big| &\leqslant \Big|f_\star(x)-f_{\lambda_t,t-1}(x)\Big|+\Big|f_{\lambda_t,t-1}(x)-\tilde{f}_t(x)\Big|\\ &\leqslant \widehat{C}_{t,\delta}(x)+\tilde{C}_{t,\delta}(x)\\ &= \sqrt{\frac{k_{\lambda_t,t-1}(x,x)}{\lambda_t}}\Big(B_{\lambda_t,t-1}(\delta/4)+c_{t,\delta}v_t\sigma_{+,t-1}\Big)\\ &= s_{\boldsymbol{\lambda},t-1}(x)\Big(\underbrace{\frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma}+c_{t,\delta}v_t\frac{\sigma_{+,t-1}}{\sigma}}_{g_t(\delta)}\Big)\,.\end{aligned}$$

Saturated arms It is now convenient to introduce the set of saturated times a time t

$$\mathcal{S}_{t,\delta} = \Big\{x\in\mathbb{X} : f_\star(x_\star)-f_\star(x) > s_{\boldsymbol{\lambda},t-1}(x)g_t(\delta)\Big\} \quad\text{together with}\quad x_{\mathcal{S},t} = \operatorname*{argmin}_{x\notin\mathcal{S}_{t,\delta}} s_{\boldsymbol{\lambda},t-1}(x)\,.$$

We remark that by construction $\star\notin\mathcal{S}_{t,\delta}$ for all t. Now, by the strategy of the Kernel TS algorithm, $x_t=\operatorname{argmax}_{x\in\mathbb{X}}\tilde{f}_t(x)$. Thus, we deduce that on the event $\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}$

$$\begin{aligned}f_\star(x_\star)-f_\star(x_t) &= f_\star(x_\star)-f_\star(x_{\mathcal{S},t})+f_\star(x_{\mathcal{S},t})-f_\star(x_t)\\ &\leqslant s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})g_t(\delta)+\Big(f_\star(x_{\mathcal{S},t})-\tilde{f}_t(x_{\mathcal{S},t})\Big)\\ &\quad+\Big(\underbrace{\tilde{f}_t(x_{\mathcal{S},t})-\tilde{f}_t(x_t)}_{\leqslant 0}\Big)+\Big(\tilde{f}_t(x_t)-f_\star(x_t)\Big)\\ &\leqslant 2s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})g_t(\delta)+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\,.\end{aligned}$$

Also, $f_\star(x_\star)-f_\star(x_t)\leqslant R$, where $R=\max_{x\in\mathbb{X}}f_\star(x_\star)-f_\star(x)<\infty$. We then remark that by definition of $x_{\mathcal{S},t}$, we have

$$\begin{aligned}\mathbb{E}[s_{\boldsymbol{\lambda},t-1}(x_t)|\mathcal{H}_{t-1}] &\geqslant \mathbb{E}[s_{\boldsymbol{\lambda},t-1}(x_t)\mathbb{I}\{x_t\notin\mathcal{S}_{t,\delta}\}|\mathcal{H}_{t-1}]\\ &\geqslant \mathbb{E}[s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})\mathbb{I}\{x_t\notin\mathcal{S}_{t,\delta}\}|\mathcal{H}_{t-1}]\\ &= s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})\mathbb{P}\Big(x_t\notin\mathcal{S}_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)\,.\end{aligned}$$

Likewise,

$$\min\{s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})g_t(\delta),R\} \leqslant \frac{\mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]}{\mathbb{P}\Big(x_t \notin \mathcal{S}_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)}\,.$$

Since on the other hand, $(f_\star(x_\star)-f_\star(x_t))\mathbb{I}\{x_t\notin\mathcal{S}_{t,\delta}\}\leqslant s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\mathbb{I}\{x_t\notin\mathcal{S}_{t,\delta}\}$, we deduce that on the event $\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}$ we have

$$\begin{aligned}
f_\star(x_\star)-f_\star(x_t) &\leqslant \min\Big\{2s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})g_t(\delta)+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\Big\}\mathbb{I}\{x_t\in\mathcal{S}_{t,\delta}\}\\
&\quad +s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\mathbb{I}\{x_t\notin\mathcal{S}_{t,\delta}\}\\
&\leqslant \min\Big\{2s_{\boldsymbol{\lambda},t-1}(x_{\mathcal{S},t})g_t(\delta),R\Big\}\mathbb{I}\{x_t\in\mathcal{S}_{t,\delta}\}+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\\
&\leqslant \frac{\mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]}{\mathbb{P}\Big(x_t\notin\mathcal{S}_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)}\mathbb{I}\{x_t\in\mathcal{S}_{t,\delta}\}+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\,.
\end{aligned}$$

Lower bounding the denominator At this point, we note that on the event $\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}$, for all $x\in\mathcal{S}_{t,\delta}$,

$$\tilde{f}_t(x)\leqslant f_\star(x)+s_{\boldsymbol{\lambda},t-1}(x)g_t(\delta)\leqslant f_\star(x_\star)\,,$$

while on the other hand we have the inclusion $\quad\{\forall x\in\mathcal{S}_{t,\delta},\ \tilde{f}_t(x_\star)>\tilde{f}_t(x)\}\subset\{x_t\notin\mathcal{S}_{t,\delta}\}$. Thus, combining these two properties, we deduce that

$$\begin{aligned}
&\{x_t\in\mathcal{S}_{t,\delta}\}\cap\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\\
&\quad\subset\ \Big\{\exists x\in\mathcal{S}_{t,\delta},\tilde{f}_t(x_\star)\leqslant\tilde{f}_t(x)\Big\}\cap\Big\{\forall x\in\mathcal{S}_{t,\delta},\tilde{f}_t(x)\leqslant f_\star(x_\star)\Big\}\\
&\quad\subset\ \Big\{\tilde{f}_t(x_\star)\leqslant f_\star(x_\star)\Big\}\,.
\end{aligned}$$

Further, using that $\tilde{f}_t(x)|\mathcal{H}_{t-1}=\mathcal{N}(f_{\lambda_t,t-1}(x),\mathbf{V}_t)$ yields

$$\begin{aligned}
&\{x_t\in\mathcal{S}_{t,\delta}\}\cap\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\\
&\quad\subset\ \Big\{\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\leqslant f_\star(x_\star)-f_{\lambda_t,t-1}(x_\star)\Big\}\cap\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\\
&\quad\subset\ \Big\{\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\leqslant\widehat{C}_{t,\delta}(x_\star)\Big\}\subset\Big\{\big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\big|\leqslant\widehat{C}_{t,\delta}(x_\star)\Big\}\,,
\end{aligned}$$

from which we obtain

$$\Big\{\big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\big|>\widehat{C}_{t,\delta}(x_\star)\Big\}\cap\widehat{E}_{t,\delta}\subset\{x_t\notin\mathcal{S}_{t,\delta}\}\cup\tilde{E}^c_{t,\delta}\,.$$

Thus, we have proved that

$$\begin{aligned}
\mathbb{P}\Big(x_t\notin\mathcal{S}_{t,\delta}\Big|\mathcal{H}_{t-1}\Big) &\geqslant \mathbb{P}\Big(\big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\big|>\widehat{C}_{t,\delta}(x_\star),\widehat{E}_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)-\mathbb{P}\Big(\tilde{E}^c_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)\\
&= \mathbb{P}\Big(\big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\big|>\widehat{C}_{t,\delta}(x_\star)\Big|\mathcal{H}_{t-1}\Big)\mathbb{I}\{\widehat{E}_{t,\delta}\}-\mathbb{P}\Big(\tilde{E}^c_{t,\delta}\Big|\mathcal{H}_{t-1}\Big)\,.
\end{aligned}$$

Anti-concentration We now resort to an anti-concentration result for Gaussian variables (Abramowitz and Stegun, 1964). More precisely, the following inequality holds

$$\mathbb{P}\Big(\Big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\Big|>\widehat{C}_{t,\delta}(x_\star)\Big|\mathcal{H}_{t-1}\Big)\geqslant\frac{1}{2\sqrt{\pi}z}e^{-z^2/2}$$

where we introduced the $\mathcal{H}_{t-1}$-measurable random variable

$$z \;=\; \frac{\widehat{C}_{t,\delta}(x_\star)}{v_t\sigma_{+,t-1}\sqrt{\frac{k_{\lambda_t,t-1}(\star,\star)}{\lambda_t}}}=\frac{B_{\lambda_t,t-1}(\delta/4)}{v_t\sigma_{+,t-1}}\,,\qquad\text{provided that } z\geqslant 1\,.$$

Taking $v_t=\frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma_{+,t-1}\sqrt{2\alpha_t\ln(\beta_t)}}$ for constants α_t,β_t such that $2\alpha_t\ln(\beta_t)\geqslant 1$ thus yields

$$\mathbb{P}\Big(\Big|\tilde{f}_t(x_\star)-f_{\lambda_t,t-1}(x_\star)\Big|>\widehat{C}_{t,\delta}(x_\star)\Big|\mathcal{H}_{t-1}\Big)\geqslant p_t\overset{\text{def}}{=}\frac{\beta_t^{-\alpha_t}}{2\sqrt{\pi}\sqrt{2\alpha_t\ln(\beta_t)}}\,.$$

Summary At this point of the proof, we have proved that

$$\begin{aligned}
&(f_\star(x_\star)-f_\star(x_t))\mathbb{I}\{\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\}\\
&\leqslant \frac{\mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]\mathbb{I}\{x_t\in\mathcal{S}_{t,\delta}\}}{p_t\mathbb{I}\{\widehat{E}_{t,\delta}\}-\mathbb{P}(\tilde{E}^c_{t,\delta}\,|\mathcal{H}_{t-1})}\mathbb{I}\{\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\}\\
&\qquad +s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\mathbb{I}\{\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\}\\
&\leqslant \mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]\Big(\frac{1}{p_t}+\frac{\mathbb{P}(\tilde{E}^c_{t,\delta}|\mathcal{H}_{t-1})}{p_t^2}\Big)+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\,,
\end{aligned}$$

where in the second inequality, we used the property $\frac{1}{p-q}=\frac{1}{p}+\frac{q}{p(p-q)}\leqslant\frac{1}{p}+\frac{q}{p^2}$, for $p>q$. Combining the bound on $\mathbb{P}(\tilde{E}^c_{t,\delta}|\mathcal{H}_{t-1})$ and the definition of p_t, we obtain

$$\begin{aligned}
&(f_\star(x_\star)-f_\star(x_t))\mathbb{I}\{\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}\}\\
&\leqslant \mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]\Big(\sqrt{8\pi\alpha_t\ln(\beta_t)}\beta_t^{\alpha_t}+\delta\frac{8\pi\alpha_t\ln(\beta_t)\beta_t^{2\alpha_t}}{c_{t,\delta}t(t+1)}\Big)\\
&\qquad +s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\,.
\end{aligned}$$

Pseudo-regret Summing-up the previous terms over $t\geqslant 1$, we obtain that the pseudo-regret of the Kernel TS strategy satisfies, on the event $\bigcap_{t\geqslant1}\widehat{E}_{t,\delta}\cap\tilde{E}_{t,\delta}$ that holds with probability higher than $1-2\delta$,

$$\begin{aligned}
\mathfrak{R}_T \;\leqslant\; \sum_{t=1}^T\Big[&\mathbb{E}[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta),R\}|\mathcal{H}_{t-1}]\sqrt{8\pi\alpha_t\ln(\beta_t)}\beta_t^{\alpha_t}\Big(1+\delta\frac{\sqrt{8\pi\alpha_t\ln(\beta_t)}\beta_t^{\alpha_t}}{c_{t,\delta}t(t+1)}\Big)\\
&+s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\Big]\,,
\end{aligned}$$

where $c_{t,\delta} = \max\{\sqrt{2\ln(t(t+1)|\mathbb{X}|/\sqrt{\pi}\delta)}, 1\}$, and the constants α_t, β_t must be such that $2\alpha_t \ln(\beta_t) \geqslant 1$ and $\sqrt{8\pi\alpha_t \ln(\beta_t)}\beta_t^{\alpha_t} \geqslant 1$. Also, let us recall that

$$\begin{aligned} g_t(\delta) &= \frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma} + c_{t,\delta} v_t \frac{\sigma_{+,t-1}}{\sigma} \\ &= \frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma}\Big(1 + \frac{c_{t,\delta}}{\sqrt{2\alpha_t \ln(\beta_t)}}\Big)\,. \end{aligned}$$

In particular, the specific choice $\alpha_t = 1/2\ln(\beta_t)$ where $\beta_t > 1$ (which satisfies $1 \geqslant 1$ and $\sqrt{4\pi e} \geqslant 1$) yields

$$\begin{aligned} \mathfrak{R}_T &\leqslant \sum_{t=1}^T \mathbb{E}\Big[\min\{2s_{\boldsymbol{\lambda},t-1}(x_t)\frac{B_{\lambda_t,t-1}(\delta/4)}{\sigma}\Big(1+c_{t,\delta}\Big), R\}\Big|\mathcal{H}_{t-1}\Big]\eta_t + s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta) \\ &= \sum_{t=1}^T \mathbb{E}\Big[\min\big\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta), R\big\}\Big|\mathcal{H}_{t-1}\Big]\eta_t + s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\,, \end{aligned}$$

where we introduced the deterministic quantity $\eta_t = \sqrt{4\pi e}\Big(1 + \delta\frac{\sqrt{4\pi e}}{c_{t,\delta}t(t+1)}\Big)$.

Concentration In order to finish the proof, we now relate the sum of the terms $\mathbb{E}[s_{\boldsymbol{\lambda},t-1}(x_t)|\mathcal{H}_{t-1}]$ to the sum of the terms $s_{\boldsymbol{\lambda},t-1}(x_t)$, for $t \geqslant 1$. More precisely, let us introduce the following random variable

$$X_t = \mathbb{E}\Big[\min\big\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta), R\big\}\Big|\mathcal{H}_{t-1}\Big]\eta_t - \min\big\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta), R\big\}\eta_t\,.$$

By construction, $\mathbb{E}[X_t|\mathcal{H}_{t-1}] = 0$ and $|X_t| \leqslant R\eta_t$. Thus, by an application of Azuma-hoeffding's inequality for martingales, we obtain that for all $\delta \in (0,1)$, with probability higher than $1-\delta$,

$$\sum_{t=1}^T X_t \leqslant \sqrt{2\sum_{t=1}^T R^2\eta_t^2 \ln(1/\delta)}\,,$$

and thus that on an event of probability higher than $1-3\delta$,

$$\mathfrak{R}_T \leqslant \sum_{t=1}^T \min\big\{2s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta), R\big\}\eta_t + s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta) + \sqrt{2\sum_{t=1}^T R^2\eta_t^2\ln(1/\delta)}\,.$$

Replacing η_t with its expression, that is

$$\begin{aligned} \eta_t &= \sqrt{4\pi e}\Big(1 + \delta\frac{\sqrt{4\pi e}}{\max\{\sqrt{2\ln(t(t+1)|\mathbb{X}|/\sqrt{\pi}\delta)}, 1\}t(t+1)}\Big) \\ &\leqslant \sqrt{4\pi e}\Big(1 + \delta\frac{\sqrt{4\pi e}}{t(t+1)}\Big)\,, \end{aligned}$$

we deduce that with probability higher than $1-3\delta$,

$$\begin{aligned} \mathfrak{R}_T &\leqslant (4\sqrt{\pi e}+1)\Big(\sum_{t=1}^T s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\Big) + R\delta 4\pi e + R\sqrt{8\pi e\sum_{t=1}^T(1+\delta\frac{\sqrt{4\pi e}}{t(t+1)})^2\ln(1/\delta)} \\ &\leqslant (4\sqrt{\pi e}+1)\Big(\sum_{t=1}^T s_{\boldsymbol{\lambda},t-1}(x_t)g_t(\delta)\Big) + R\delta 4\pi e + \sqrt{8\pi e(1+\delta\sqrt{4\pi e})^2}R\sqrt{T\ln(1/\delta)} \\ &= (4\sqrt{\pi e}+1)\Big(\sum_{t=1}^T \sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}}B_{\lambda_t,t-1}(\delta/4)(1+c_{t,\delta})\Big) \\ &\quad + R\delta 4\pi e + \sqrt{8\pi e(1+\delta\sqrt{4\pi e})^2}R\sqrt{T\ln(1/\delta)}\,. \end{aligned}$$

This concludes the proof of the main result, since $c_{t,\delta} \leqslant c_{T,\delta}$.

Final bound Then, using Lemma 2 we can rewrite the regret as

$$\begin{aligned} \mathfrak{R}_T &= (4\sqrt{\pi e}+1)(1+c_{T,\delta})\sigma_+\Big(1+\sqrt{2\ln(4/\delta)+2\gamma_T(\sigma_-^2/C^2)}\Big)\sum_{t=1}^T\sqrt{\frac{k_{\lambda_t,t-1}(x_t,x_t)}{\lambda_t}} \\ &\quad + R\delta 4\pi e + \sqrt{8\pi e(1+\delta\sqrt{4\pi e})^2}R\sqrt{T\ln(1/\delta)}\,. \end{aligned}$$

Using Lemma 1 together with a Cauchy-Schwarz inequality, we finally obtain

$$\begin{aligned} \mathfrak{R}_T &= (4\sqrt{\pi e}+1)(1+c_{T,\delta})\sigma_+\Big(1+\sqrt{2\ln(4/\delta)+2\gamma_T(\sigma_-^2/C^2)}\Big)\sqrt{\frac{2TC^2/\sigma^2}{\ln\Big(1+\frac{C^2}{\sigma^2}\Big)}\gamma_T(\sigma^2/C^2)} \\ &\quad + R\delta 4\pi e + \sqrt{8\pi e(1+\delta\sqrt{4\pi e})^2}R\sqrt{T\ln(1/\delta)}\,. \end{aligned}$$

■

References

Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2312–2320, 2011.

M. Abeille and A. Lazaric. Linear Thompson sampling revisited. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 176–184, 2017.

M. Abramowitz and I. A. Stegun. *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, volume 55. Courier Corporation, 1964.

S. Agrawal and N. Goyal. Thompson sampling for contextual bandits with linear payoffs. In *International Conference on Machine Learning (ICML)*, pages 127–135, 2013.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

E. Brochu, N. De Freitas, and A. Ghosh. Active preference learning with discrete choice data. In *Advances in Neural Information Processing Systems (NIPS)*, pages 409–416, 2008.

S. Canu, X. Mary, and A. Rakotomamonjy. Functional learning through kernels. *arXiv preprint arXiv:0910.1013*, 2009.

W. Chu, L. Li, L. Reyzin, and R. E. Schapire. Contextual bandits with linear payoff functions. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 15, pages 208–214, 2011.

A. Cotter, J. Keshet, and N. Srebro. Explicit approximations of the Gaussian kernel. *arXiv preprint arXiv:1109.4603*, 2011.

T. M Cover and J. A. Thomas. Elements of information theory. 1991.

A. Krause and C. S. Ong. Contextual Gaussian process bandit optimization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2447–2455, 2011.

L. Li, W. Chu, J. Langford, and R. E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *International Conference on World Wide Web (WWW)*, pages 661–0670, 2010.

S. Loustau. Penalized empirical risk minimization over besov spaces. *Electronic Journal of Statistics*, 3:824–850, 2009.

O.-A. Maillard. Self-normalization techniques for streaming confident regression. working paper or preprint, May 2016. URL `https://hal.archives-ouvertes.fr/hal-01349727`.

R. Marchant and F. Ramos. Bayesian optimisation for informative continuous path planning. In *International Conference on Robotics and Automation (ICRA)*, pages 6136–6143. IEEE, 2014.

A. Maurer and M Pontil. Empirical Bernstein bounds and sample variance penalization. In *Annual Conference on Learning Theory (COLT)*, 2009.

C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. MIT Press, 2006.

J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2951–2959, 2012.

N. Srinivas, A. Krause, S. M. Kakade, and M. Seeger. Gaussian process optimization in the bandit setting: No regret and experimental design. In *International Conference on Machine Learning (ICML)*, 2010.

N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger. Information-theoretic regret bounds for Gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.

M. Valko, N. Korda, R. Munos, I. Flaounas, and N. Cristianini. Finite-time analysis of kernelised contextual bandits. In *Conference on Uncertainty In Artificial Intelligence (UAI)*, pages 654–665, 2013.

Z. Wang and N. de Freitas. Theoretical analysis of Bayesian optimisation with unknown Gaussian process hyper-parameters. *arXiv preprint arXiv:1406.7758*, 2014.

L. Wasserman. 10-702 Statistical Machine Learning, Function Spaces, Spring 2017 (Carnegie Mellon University). `http://www.stat.cmu.edu/~larry/=sml/functionspaces.pdf`, 2017. (Accessed June 26, 2018).

A. Wilson, A. Fern, and P. Tadepalli. Using trajectory data to improve Bayesian optimization for reinforcement learning. *Journal of Machine Learning Research*, 15:253–282, 2014.

Journal of Machine Learning Research 19 (2018) 1-49 Submitted 7/17; Revised 8/18; Published 8/18

Dual Principal Component Pursuit

Manolis C. Tsakiris MTSAKIRIS@SHANGHAITECH.EDU.CN
SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
SHANGHAITECH UNIVERSITY
PUDONG, SHANGHAI, CHINA

René Vidal RVIDAL@JHU.EDU
MATHEMATICAL INSTITUTE FOR DATA SCIENCE
JOHNS HOPKINS UNIVERSITY
BALTIMORE, MD, 21218, USA

Editor: Michael Mahoney

Abstract

We consider the problem of learning a linear subspace from data corrupted by outliers. Classical approaches are typically designed for the case in which the subspace dimension is small relative to the ambient dimension. Our approach works with a dual representation of the subspace and hence aims to find its orthogonal complement; as such, it is particularly suitable for subspaces whose dimension is close to the ambient dimension (subspaces of high relative dimension). We pose the problem of computing normal vectors to the inlier subspace as a non-convex ℓ_1 minimization problem on the sphere, which we call Dual Principal Component Pursuit (DPCP) problem. We provide theoretical guarantees under which every global solution to DPCP is a vector in the orthogonal complement of the inlier subspace. Moreover, we relax the non-convex DPCP problem to a recursion of linear programs whose solutions are shown to converge in a finite number of steps to a vector orthogonal to the subspace. In particular, when the inlier subspace is a hyperplane, the solutions to the recursion of linear programs converge to the global minimum of the non-convex DPCP problem in a finite number of steps. We also propose algorithms based on alternating minimization and iteratively re-weighted least squares, which are suitable for dealing with large-scale data. Experiments on synthetic data show that the proposed methods are able to handle more outliers and higher relative dimensions than current state-of-the-art methods, while experiments in the context of the three-view geometry problem in computer vision suggest that the proposed methods can be a useful or even superior alternative to traditional RANSAC-based approaches for computer vision and other applications.

Keywords: Outliers, Robust Principal Component Analysis, High Relative Dimension, ℓ_1 Minimization, Non-Convex Optimization, Linear Programming, Trifocal Tensor

1. Introduction

Principal Component Analysis (PCA) is one of the oldest (Pearson, 1901; Hotelling, 1933) and most fundamental techniques in data analysis, with ubiquitous applications in engineering (Moore, 1981), economics and sociology (Vyas and Kumaranayake, 2006), chemistry (Ku et al., 1995), physics (Loyd et al., 2014) and genetics (Price et al., 2006) to name a few; see Jolliffe (2002) for more applications. Given a data matrix $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D\times L}$ of L data points of coordinate dimension D, PCA gives a closed form solution to the problem of finding a d-dimensional linear subspace $\hat{\mathcal{S}}$ that is closest, in the Euclidean sense, to the columns of $\boldsymbol{\mathcal{X}}$. Although the optimization problem associated with PCA is non-convex, it does admit a simple solution by means of the Singular Value Decomposition (SVD) of $\boldsymbol{\mathcal{X}}$. In fact, $\hat{\mathcal{S}}$ is the subspace spanned by the first d left singular vectors of $\boldsymbol{\mathcal{X}}$.

Using $\hat{\mathcal{S}}$ as a model for the data $\boldsymbol{\mathcal{X}}$ is meaningful when the data are known to have an approximately linear structure of underlying dimension d, i.e., they lie close to a d-dimensional subspace $\mathcal{S}$. In practice, the principal components of $\boldsymbol{\mathcal{X}}$ are known to be well-behaved under mild levels of noise, i.e., the principal angles between $\hat{\mathcal{S}}$ and $\mathcal{S}$ are relatively small, and in fact, $\hat{\mathcal{S}}$ is optimal when the noise is Gaussian (Jolliffe, 2002). However, very often in applications the data are corrupted by outliers, i.e., the data matrix has the form $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{\mathcal{O}}]\,\boldsymbol{\Gamma}$, where the M columns of $\boldsymbol{\mathcal{O}} \in \mathbb{R}^{D\times M}$ are points of $\mathbb{R}^D$ whose angles from the underlying ground truth subspace $\mathcal{S}$ associated with the inlier points $\boldsymbol{\mathcal{X}}$ are large, and $\boldsymbol{\Gamma}$ is an unknown permutation. In such cases, the principal angles between $\mathcal{S}$ and its PCA estimate $\hat{\mathcal{S}}$ will in general be large, even when M is small. This is to be expected since, by definition, the principal components of $\tilde{\boldsymbol{\mathcal{X}}}$ are orthogonal directions of maximal correlation with *all* the points of $\tilde{\boldsymbol{\mathcal{X}}}$. This phenomenon, together with the fact that outliers are almost always present in real datasets, has given rise to the important problem of outlier detection in PCA.

Traditionally, outlier detection has been a major area of study in robust statistics with notable methods being *Influence-based Detection*, *Multivariate Trimming*, M*-Estimators*, *Iteratively Reweighted Least Squares* (IRLS) and *Random Sampling Consensus* (RANSAC) (Huber, 1981; Jolliffe, 2002). These methods are usually based on non-convex optimization problems and in practice converge only to a local minimum. In addition, their theoretical analysis is usually limited and their computational complexity may be large (e.g., in the case of RANSAC). Recently, two attractive methods have appeared (Xu et al., 2012; Soltanolkotabi and Candès, 2012) that are directly based on convex optimization and are inspired by *low-rank representation* (Liu et al., 2010) and *compressed sensing* (Candès and Wakin, 2008). Even though both of these methods admit theoretical guarantees and efficient implementations, they are in principle applicable only in the case of subspaces of small relative dimension (i.e., $d/D \ll 1$). On the other hand, the theoretical guarantees of the recent REAPER method of Lerman et al. (2015) seem to suggest that the method is able to handle any subspace dimension.

In this paper we adopt a *dual* approach to the problem of robust PCA in the presence of outliers, which allows us to explicitly transcend the low relative dimension regime of modern methods such as Xu et al. (2012) or Soltanolkotabi and Candès (2012), and even be able to handle as many as 70% outliers for hyperplanes (subspaces of maximal relative dimension $(D-1)/D$), a regime where other modern (Lerman et al., 2015) or classic (Huber, 1981) methods fail. The key idea of our approach comes from the fact that, in the absence of noise, the inliers $\boldsymbol{\mathcal{X}}$ lie inside any hyperplane $\mathcal{H}_1 = \mathrm{Span}(\boldsymbol{b}_1)^\perp$ that contains the underlying linear subspace $\mathcal{S}$ associated with the inliers. This suggests that, instead of attempting to fit directly a low-dimensional linear subspace to the entire dataset $\tilde{\boldsymbol{\mathcal{X}}}$, as done e.g. in Xu et al. (2012), we can search for a *maximal hyperplane* $\mathcal{H}_1$ that contains as many points of the dataset as possible. When the inliers $\boldsymbol{\mathcal{X}}$ are in general position (to be made precise shortly) inside $\mathcal{S}$, and the outliers $\boldsymbol{\mathcal{O}}$ are in general position in $\mathbb{R}^D$, such a maximal hyperplane will contain the entire set of inliers together with possibly a few outliers. Then one may remove all points that lie outside this hyperplane and be left with an easier robust PCA problem that could potentially be addressed by existing methods. Alternatively, one can continue by finding a second maximal hyperplane $\mathcal{H}_2 = \mathrm{Span}(\boldsymbol{b}_2)^\perp$, with the new dual principal component $\boldsymbol{b}_2$ perpendicular to the first one, i.e., $\boldsymbol{b}_2 \perp \boldsymbol{b}_1$, and so on, until $c := D-d$ such maximal hyperplanes $\mathcal{H}_1, \ldots, \mathcal{H}_c$ have been found, leading to a *Dual Principal Component Analysis (DPCA)* of $\tilde{\boldsymbol{\mathcal{X}}}$. In

such a case, the inlier subspace is precisely equal to $\bigcap_{i=1}^{c} \mathcal{H}_i$, and a point is an outlier if and only if it lies outside this intersection.

We formalize the problem of searching for maximal hyperplanes with respect to $\tilde{\boldsymbol{\mathcal{X}}}$ as an ℓ_0 cosparsity-type problem (Nam et al., 2013), which we relax to a non-convex ℓ_1 problem on the sphere, referred to as the Dual Principal Component Pursuit (DPCP) problem. We provide theoretical guarantees under which every global solution of the DPCP problem is a vector orthogonal to the linear subspace associated with the inliers, i.e., it is a dual principal component. Moreover, we relax the non-convex DPCP problem to a recursion of linear programming problems and we show that, under mild conditions, their solutions converge to a dual principal component in a finite number of steps. In particular, when the inlier subspace is a hyperplane, then the solutions of the linear programming recursion converge to the global minimum of the non-convex problem in a finite number of steps. Furthermore, we propose algorithms based on alternating minimization and IRLS that are suitable for dealing with large-scale data. Extensive experiments on synthetic data show that the proposed methods are able to handle more outliers and subspaces of higher relative dimension d/D than state-of-the-art methods (Fischler and Bolles, 1981; Xu et al., 2012; Soltanolkotabi and Candès, 2012; Lerman et al., 2015), while experiments with real face and object images show that our DPCP-based methods perform on par with state-of-the-art methods.

Notation The shorthand RHS stands for *Right-Hand-Side* and similarly for LHS. The notation $\cong$ stands for *isomorphism* in whatever category the objects lying to the LHS and RHS of the symbol belong to. The notation $\simeq$ denotes approximation. For any positive integer n let $[n] := \{1, 2, \ldots, n\}$. For any positive number α let $\lceil \alpha \rceil$ denote the smallest integer that is greater than α. For sets $\mathcal{A}, \mathcal{B}$, the set $\mathcal{A} \setminus \mathcal{B}$ is the set of all elements of $\mathcal{A}$ that do not belong to $\mathcal{B}$. If $\mathcal{S}$ is a subspace of $\mathbb{R}^D$, then $\dim(\mathcal{S})$ denotes the dimension of $\mathcal{S}$ and $\pi_{\mathcal{S}} : \mathbb{R}^D \to \mathcal{S}$ is the orthogonal projection of $\mathbb{R}^D$ onto $\mathcal{S}$. For vectors $\boldsymbol{b}, \boldsymbol{b}' \in \mathbb{R}^D$ we let $\angle \boldsymbol{b}, \boldsymbol{b}'$ be the acute angle between $\boldsymbol{b}$ and $\boldsymbol{b}'$, defined as the unique angle $\theta \in [0\ 90^\circ]$ such that $\cos\theta = \left|\boldsymbol{b}^\top \boldsymbol{b}'\right|$. If $\boldsymbol{b}$ is a vector of $\mathbb{R}^D$ and $\mathcal{S}$ a linear subspace of $\mathbb{R}^D$, the principal angle of $\boldsymbol{b}$ from $\mathcal{S}$ is $\angle \boldsymbol{b}, \pi_{\mathcal{S}}(\boldsymbol{b})$. The symbol $\oplus$ denotes direct sum of subspaces. The orthogonal complement of a subspace $\mathcal{S}$ in $\mathbb{R}^D$ is $\mathcal{S}^\perp$. If $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s$ are elements of $\mathbb{R}^D$, we denote by $\mathrm{Span}(\boldsymbol{y}_1, \ldots, \boldsymbol{y}_s)$ the subspace of $\mathbb{R}^D$ spanned by these elements. $\mathbb{S}^{D-1}$ denotes the unit sphere of $\mathbb{R}^D$. For a vector $\boldsymbol{w} \in \mathbb{R}^D$ we define $\hat{\boldsymbol{w}} := \boldsymbol{w} / \left\|\boldsymbol{w}\right\|_2$, if $\boldsymbol{w} \neq \boldsymbol{0}$, and $\hat{\boldsymbol{w}} := 0$ otherwise. Given a square matrix $\boldsymbol{C}$, $\mathrm{Diag}(\boldsymbol{C})$ denotes the vector of diagonal elements of $\boldsymbol{C}$. Given a square matrix $\boldsymbol{P}$, the notation $\boldsymbol{0} \leq \boldsymbol{P} \leq \boldsymbol{I}$ indicates that $\boldsymbol{P}, \boldsymbol{I} - \boldsymbol{P}$ are positive semi-definite matrices. With a mild abuse of notation we will be treating on several occasions matrices as sets, i.e., if $\boldsymbol{\mathcal{X}}$ is $D \times N$ and $\boldsymbol{x}$ a point of $\mathbb{R}^D$, the notation $\boldsymbol{x} \in \boldsymbol{\mathcal{X}}$ signifies that $\boldsymbol{x}$ is a column of $\boldsymbol{\mathcal{X}}$. Similarly, if $\boldsymbol{\mathcal{O}}$ is a $D \times M$ matrix, the notation $\boldsymbol{\mathcal{X}} \cap \boldsymbol{\mathcal{O}}$ signifies the points of $\mathbb{R}^D$ that are common columns of $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{O}}$. The notation Sign denotes the sign function $\mathrm{Sign} : \mathbb{R} \to \{-1, 0, 1\}$ defined as

$$\mathrm{Sign}(x) = \begin{cases} x/|x| & \text{if } x \neq 0, \\ 0 & \text{if } x = 0. \end{cases} \tag{1}$$

Finally, we note that the ith entry of the subdifferential of the ℓ_1-norm $\|\boldsymbol{z}\|_1 = \sum_{i=1}^{D} |z_i|$ of a vector $\boldsymbol{z} = (z_1, \ldots, z_D)^\top$ is a set-valued function on $\mathbb{R}^D$ defined as

$$\mathrm{Sgn}(z_i) = \begin{cases} \mathrm{Sign}(z_i) & \text{if } z_i \neq 0, \\ [-1, 1] & \text{if } z_i = 0. \end{cases} \tag{2}$$

2. Prior Art

We begin by briefly reviewing some state-of-the-art methods for learning a linear subspace from data $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{\mathcal{O}}]\,\boldsymbol{\Gamma}$ in the presence of outliers. The literature on this subject is vast and our account is far from exhaustive; with a few exceptions, we mainly focus on modern methods based on convex optimization. For methods from robust statistics see Huber (1981); Jolliffe (2002), for online subspace learning methods see Balzano et al. (2010); Feng et al. (2013), for regression-type methods see Wang et al. (2015), while for fast and other methods the reader is referred to the excellent literature review of Lerman and Zhang (2014) or the recent survey by Lerman and Maunu (2018). Finally, we note that preliminary results associated with the present work have been published in the form of a conference paper[1] (Tsakiris and Vidal, 2015). While the present paper was under review, we extended our approach to clustering data from multiple subspaces (Tsakiris and Vidal, 2017), which can also be thought of as a robust PCA problem but with structured outliers. This is a continuation of the present work, which certainly builds on the concepts and algorithms presented here, yet requires sufficiently distinct machinery to be fully established.

RANSAC One of the oldest and most popular outlier detection methods for PCA is *Random Sampling Consensus (RANSAC)* (Fischler and Bolles, 1981). The idea behind RANSAC is simple: alternate between randomly sampling a subset of cardinality d from the dataset and computing a d-dimensional subspace from this subset, until a subspace $\hat{\mathcal{S}}$ is found that maximizes the number of points in the entire dataset that approximately lie in $\hat{\mathcal{S}}$ within some error. RANSAC is typically used when the ambient dimension D is small (say $D \leq 20$), yielding high quality subspace estimates regardless of the subspace relative dimension d/D. However, as D increases, RANSAC becomes inefficient for large values of d/D, except when the outlier ratio is very small, as otherwise a prohibitive number of trials may be required in order to obtain outlier-free samples and thus furnish reliable models. Additionally, RANSAC requires as input an estimate for the dimension of the subspace as well as a thresholding parameter, which is used to distinguish outliers from inliers; naturally the performance of RANSAC is very sensitive to these two parameters.

$\ell_{2,1}$-RPCA Unlike RANSAC, modern methods for outlier detection in PCA are primarily based on convex optimization. One of the earliest and most important such methods is the $\ell_{2,1}$-RPCA method of Xu et al. (2012), which is in turn inspired by the Robust Principal Component Analysis (RPCA) algorithm of Candès et al. (2011). $\ell_{2,1}$-RPCA computes a $(\ell_* + \ell_{2,1})$-norm decomposition[2] of the data matrix, instead of the $(\ell_* + \ell_1)$-decomposition in Candès et al. (2011). More specifically, $\ell_{2,1}$-RPCA solves the optimization problem

$$\min_{\boldsymbol{L}, \boldsymbol{E}:\, \tilde{\boldsymbol{\mathcal{X}}} = \boldsymbol{L} + \boldsymbol{E}} \ \left\|\boldsymbol{L}\right\|_* + \lambda \left\|\boldsymbol{E}\right\|_{2,1}, \tag{3}$$

which attempts to decompose the data matrix $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma}$ into the sum of a low-rank matrix $\boldsymbol{L}$, and a matrix $\boldsymbol{E}$ that has only a few non-zero columns. The idea is that $\boldsymbol{L}$ is associated with the inliers, having the form $\boldsymbol{L} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{0}_{D \times M}]\boldsymbol{\Gamma}$, and $\boldsymbol{E}$ is associated with the outliers, having the form $\boldsymbol{E} = [\boldsymbol{0}_{D \times N}\ \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma}$. The optimization problem (3) is convex and admits theoretical guarantees

1. We note that the proof of Theorem 2 in Tsakiris and Vidal (2015) contained an inaccuracy which makes its statement incomplete. The complete statement is Theorem 11 in the present paper.
2. Here ℓ_* denotes the nuclear norm, which is the sum of the singular values of the matrix. Also, $\ell_{2,1}$ is defined as the sum of the Euclidean norms of the columns of a matrix.

and efficient algorithms based on the alternating direction method of multipliers (ADMM) (Gabay and Mercier, 1976). However, it is expected to succeed only when the intrinsic dimension d of the inliers is small enough (otherwise $[\boldsymbol{\mathcal{X}}\ \mathbf{0}_{D\times M}]$ will not be low-rank), and the outlier ratio is not too large (otherwise $[\mathbf{0}_{D\times N}\ \boldsymbol{\mathcal{O}}]$ will not be column-sparse). Finally, notice that $\ell_{2,1}$-RPCA does not require as input the subspace dimension d, because it does not directly compute an estimate for the subspace. Rather, the subspace can be obtained subsequently by applying classic PCA on $\boldsymbol{L}$, and now one does need an estimate for d.

SE-RPCA Separating outliers from low-rank inlier points can also be achieved by exploiting the *self-expressiveness* (SE) property of the data matrix, a notion popularized by the work of Elhamifar and Vidal (2011, 2013) in the area of subspace clustering (Vidal, 2011). Specifically, if a column $\tilde{\boldsymbol{x}}$ of $\tilde{\boldsymbol{\mathcal{X}}}$ is an inlier, then it can be expressed as a linear combination of d other inliers in $\tilde{\boldsymbol{\mathcal{X}}}$, while if $\tilde{\boldsymbol{x}}$ is an outlier, then in principle it requires D columns of $\tilde{\boldsymbol{\mathcal{X}}}$. The coefficient matrix $\boldsymbol{C}$ can be obtained as the solution to the convex optimization problem

$$\min_{\boldsymbol{C}} \ \|\boldsymbol{C}\|_1 \ \text{ s.t. } \ \tilde{\boldsymbol{\mathcal{X}}} = \tilde{\boldsymbol{\mathcal{X}}}\boldsymbol{C}, \ \ \text{Diag}(\boldsymbol{C}) = \mathbf{0}, \tag{4}$$

where the extra constraint prevents the trivial solution $\boldsymbol{C} = I$. Given $\boldsymbol{C}$, and under the hypothesis that d/D is small, a column of $\tilde{\boldsymbol{\mathcal{X}}}$ is declared as an outlier, if the ℓ_1 norm of the corresponding column of $\boldsymbol{C}$ is large; see Soltanolkotabi and Candès (2012) for an explicit formula. SE-RPCA admits theoretical guarantees (Soltanolkotabi and Candès, 2012) and efficient ADMM implementations (Elhamifar and Vidal, 2013). Moreover, the recent work of You et al. (2017) has demonstrated that the information contained in the self-expressive matrix $\boldsymbol{C}$ can be further exploited to identify the outliers by means of a random walk on the directed affinity graph defined by $\boldsymbol{C}$, thus yielding superior results than the simple thresholding of the norms of the columns of $\boldsymbol{C}$. In contrast to $\ell_{2,1}$-RPCA, which in principle fails in the presence of a very large number of outliers, SE-RPCA is still expected to perform well, since the existence of sparse *subspace-preserving* self-expressive patterns does not depend on the number of outliers present. Also, similarly to $\ell_{2,1}$-RPCA, SE-RPCA does not directly require an estimate for the subspace dimension d. Nevertheless, knowledge of d is necessary if one wants to furnish an actual subspace estimate, which entails removing the outliers (a judiciously chosen threshold would also be necessary here) and applying PCA.

REAPER Another robust subspace learning method that admits an interesting theoretical analysis is *REAPER* (Lerman et al., 2015), which is conceptually associated with the optimization problem

$$\min_{\boldsymbol{P}} \sum_{j=1}^{L} \left\|(\boldsymbol{I}_D - \boldsymbol{P})\tilde{\boldsymbol{x}}_j\right\|_2 \ \text{ s.t. } \ \boldsymbol{P} \ \text{ is an orthogonal projection} \quad \text{and} \quad \text{Trace}(\boldsymbol{P}) = d. \tag{5}$$

Here the vector $\tilde{\boldsymbol{x}}_j$ denotes the j-th column of $\tilde{\boldsymbol{\mathcal{X}}}$ and the matrix $\boldsymbol{P}$ denotes the orthogonal projection onto a d-dimensional linear subspace $\mathcal{S}$. Notice that $\boldsymbol{P}$ can be thought of as the product $\boldsymbol{P} = \boldsymbol{U}\boldsymbol{U}^\top$, where the columns of $\boldsymbol{U} \in \mathbb{R}^{D\times d}$ form an orthonormal basis for $\mathcal{S}$. Since the problem in (5) is non-convex, Lerman et al. (2015) relaxed it to the convex semi-definite program

$$\min_{\boldsymbol{P}} \sum_{j=1}^{L} \left\|(\boldsymbol{I}_D - \boldsymbol{P})\tilde{\boldsymbol{x}}_j\right\|_2 \ \text{ s.t. } \ \mathbf{0} \le \boldsymbol{P} \le \boldsymbol{I}_D, \ \ \text{Trace}\,(\boldsymbol{P}) = d, \tag{6}$$

and obtained an approximate solution $\boldsymbol{P}^*$ to (5) as the rank-d orthogonal projector that is closest to the global solution of (6), as measured by the nuclear norm. It was shown by Lerman et al. (2015) that the orthoprojector $\boldsymbol{P}^*$ obtained in this way is within a neighborhood of the orthoprojector corresponding to the true underlying inlier subspace. In practice, the semi-definite program (6) may become prohibitively expensive to solve even for moderate values of the ambient dimension D. As a consequence, the authors proposed an *Iteratively Reweighted Least Squares (IRLS)* scheme to obtain a numerical solution of (6), whose objective value was shown to converge to a neighborhood of the optimal objective value of problem (6).

One advantage of REAPER with respect to $\ell_{2,1}$-RPCA and SE-RPCA, is that its theoretical conditions allow for the subspace to have arbitrarily large relative dimension, providing that the outlier ratio is sufficiently small. It is interesting to note here that this is precisely the condition under which RANSAC (Huber, 1981) can handle large relative dimensions; the main difference though between RANSAC and REAPER is that the latter employs convex optimization, and for a fixed relative dimension and computational budget REAPER can tolerate considerably higher outlier ratios than RANSAC (see Fig. 6, §7).

COHERENCE PURSUIT (CoP) The recent work of Rahmani and Atia (2017) analyzes a simple yet efficient algorithm for detecting the inlier space from pairwise point coherences, hence called *Coherence Pursuit (CoP)*. The main insight behind CoP is that, under the hypothesis that the inliers lie in a low-dimensional subspace, inlier points tend to have significantly higher coherence with the rest of the points in the dataset, than outlier points. Hence the columns of the pairwise coherence matrix $\tilde{\boldsymbol{\mathcal{X}}}^\top\tilde{\boldsymbol{\mathcal{X}}}$ that have large norm are expected to correspond to inliers. Indeed, CoP orders the columns of $\tilde{\boldsymbol{\mathcal{X}}}^\top\tilde{\boldsymbol{\mathcal{X}}}$ in descending values for a given norm and selects sufficiently many of them from the top, until their span yields a d-dimensional subspace. Similarly to SE-RPCA, the performance of CoP is not expected to be significantly degraded as the number of outliers increases, as long as each outlier remains sufficiently incoherent with the rest of the dataset. As demonstrated by Rahmani and Atia (2017), CoP has a competitive performance and admits an extensive theoretical analysis.

L1-PCA* Finally, we mention the method L1-PCA* of Brooks et al. (2013), since it works with the orthogonal complement of the subspace, similarly to the proposed method of the present paper. Nevertheless, L1-PCA* is slightly unusual in that it learns ℓ_1 hyperplanes, i.e., hyperplanes that minimize the ℓ_1 distance to the points, as opposed to the Euclidean distance used by methods such as PCA and REAPER. Overall, no theoretical guarantees seem to be known for L1-PCA*, as far as the subspace learning problem is concerned. In addition, L1-PCA* requires solving $\mathcal{O}(D^2)$ linear programs, where D is the ambient dimension, which makes it computationally expensive.

3. Problem Formulation

In this section we formulate the problem addressed in this paper. We describe our data model (§3.1), and motivate the problem at a conceptual (§3.2) and computational level (§3.3).

3.1 Data model

We employ a deterministic noise-free data model, under which the given data is

$$\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \ \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma} = [\tilde{\boldsymbol{x}}_1, \ldots, \tilde{\boldsymbol{x}}_L] \in \mathbb{R}^{D\times L}, \tag{7}$$

where the N inliers $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \in \mathbb{R}^{D\times N}$ lie in the intersection of the unit sphere $\mathbb{S}^{D-1}$ with an unknown proper subspace $\mathcal{S}$ of $\mathbb{R}^D$ of unknown dimension $1 \leq d \leq D-1$, and the M outliers $\boldsymbol{\mathcal{O}} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_M] \in \mathbb{R}^{D\times M}$ lie on the sphere $\mathbb{S}^{D-1}$. The unknown permutation $\boldsymbol{\Gamma}$ indicates that we do not know which point is an inlier and which point is an outlier. Finally, we assume that the points $\tilde{\boldsymbol{\mathcal{X}}}$ are in *general position*, in the sense that there are no relations among the columns of $\tilde{\boldsymbol{\mathcal{X}}}$ except for those implied by the inclusions $\boldsymbol{\mathcal{X}} \subset \mathcal{S}$ and $\tilde{\boldsymbol{\mathcal{X}}} \subset \mathbb{R}^D$. In particular, every D-tuple of columns of $\tilde{\boldsymbol{\mathcal{X}}}$ such that at most d points come from $\boldsymbol{\mathcal{X}}$ is linearly independent. Notice that as a consequence every d-tuple of inliers and every D-tuple of outliers are linearly independent, and also $\boldsymbol{\mathcal{X}} \cap \boldsymbol{\mathcal{O}} = \emptyset$. Finally, to avoid degenerate situations we will assume that $N \geq d+1$ and $M \geq D-d$.[3]

3.2 Conceptual formulation

Given $\tilde{\boldsymbol{\mathcal{X}}}$, we consider the problem of partitioning its columns into those that lie in $\mathcal{S}$ and those that don't. Since we have made no assumption about the dimension of $\mathcal{S}$, this problem is however not well posed because $\mathcal{S}$ can be anything from a line to a $(D-1)$-dimensional hyperplane, and hence $\boldsymbol{\mathcal{X}}$ lies inside every subspace that contains $\mathcal{S}$, which in turn may contain some elements of $\boldsymbol{\mathcal{O}}$. Instead, it is meaningful to search for a linear subspace of $\mathbb{R}^D$ that contains all of the inliers and perhaps a few outliers. Since we do not know the intrinsic dimension d of the inliers, a natural choice is to search for a hyperplane of $\mathbb{R}^D$ that contains all the inliers.

Problem 1 *Given the dataset $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}} \;\; \boldsymbol{\mathcal{O}}]\,\boldsymbol{\Gamma}$, find a hyperplane $\mathcal{H}$ that contains all the inliers $\boldsymbol{\mathcal{X}}$.*

Notice that hyperplanes that contain all the inliers always exist: any non-zero vector $\boldsymbol{b}$ in the orthogonal complement $\mathcal{S}^\perp$ of the linear subspace $\mathcal{S}$ associated with the inliers defines a hyperplane (with normal vector $\boldsymbol{b}$) that contains all inliers $\boldsymbol{\mathcal{X}}$. Having such a hyperplane $\mathcal{H}_1$ at our disposal, we can partition our dataset as $\tilde{\boldsymbol{\mathcal{X}}} = \tilde{\boldsymbol{\mathcal{X}}}_1 \cup \tilde{\boldsymbol{\mathcal{X}}}_2$, where $\tilde{\boldsymbol{\mathcal{X}}}_1$ are the points of $\tilde{\boldsymbol{\mathcal{X}}}$ that lie in $\mathcal{H}_1$ and $\tilde{\boldsymbol{\mathcal{X}}}_2$ are the remaining points. Then by definition of $\mathcal{H}_1$, we know that $\tilde{\boldsymbol{\mathcal{X}}}_2$ will consist purely of outliers, in which case we can safely replace our original dataset $\tilde{\boldsymbol{\mathcal{X}}}$ with $\tilde{\boldsymbol{\mathcal{X}}}_1$ and reconsider the problem of robust PCA on $\tilde{\boldsymbol{\mathcal{X}}}_1$. We emphasize that $\tilde{\boldsymbol{\mathcal{X}}}_1$ will contain all the inliers $\boldsymbol{\mathcal{X}}$ together with at most $D-d-1$ outliers,[4] a number which may be dramatically smaller than the original number of outliers. Then one may apply existing methods such as Xu et al. (2012), Soltanolkotabi and Candès (2012) or Fischler and Bolles (1981) to finish the task of identifying the remaining outliers, as the following example demonstrates.

Example 1 *Suppose we have $N = 1000$ inliers lying in general position in a linear subspace of $\mathbb{R}^{100}$ of dimension $d = 90$. Suppose that the dataset is corrupted by $M = 1000$ outliers lying in general position in $\mathbb{R}^{100}$. Let $\mathcal{H}$ be a hyperplane that contains all 1000 inliers. Since the dimensionality of the inliers is 90 and the dimensionality of the hyperplane is 99, there are only $99 - 90 = 9$ linearly independent directions left for the hyperplane to fit, i.e., $\mathcal{H}$ will contain at most 9 outliers (it can not contain more outliers since this would violate the general position hypothesis). If we remove the points of the dataset that do not lie in $\mathcal{H}$, then we are left with 1000 inliers and at most 9 outliers. A simple application of RANSAC is expected to identify the remaining outliers in only a few trials.*

3. If the number of outliers is less than $D-d$, then the entire dataset is degenerate because it lies in a proper hyperplane of the ambient space, hence we can reduce the coordinate representation of the data and eventually satisfy the stated condition.
4. This comes from the assumption of general position.

Alternatively, if the true dimension d is known, one may keep working with the entire dataset $\tilde{\boldsymbol{\mathcal{X}}}$ (i.e., no point removal takes place) and search for a second hyperplane $\mathcal{H}_2$ that contains all the inliers, such that its normal vector $\boldsymbol{b}_2$ is linearly independent (e.g., orthogonal) from the normal vector $\boldsymbol{b}_1$ of $\mathcal{H}_1$. Then $\mathcal{H}_1 \cap \mathcal{H}_2$ is a linear subspace of dimension $D-2$ that contains all the inliers $\boldsymbol{\mathcal{X}}$, and as a consequence $\mathrm{Span}(\boldsymbol{\mathcal{X}}) = \mathcal{S} \subset \mathcal{H}_1 \cap \mathcal{H}_2$. Then a third hyperplane $\mathcal{H}_3 \supset \boldsymbol{\mathcal{X}}$ may be sought for, such that its normal vector $\boldsymbol{b}_3$ is not in $\mathrm{Span}(\boldsymbol{b}_1, \boldsymbol{b}_2)$, and so on. Repeating this process $c = \mathrm{codim}\,\mathcal{S} = D-d$ times, until c linearly independent hyperplanes[5] $\mathcal{H}_1, \ldots, \mathcal{H}_c$ have been found, each containing $\boldsymbol{\mathcal{X}}$, we arrive at a situation where $\bigcap_{k=1}^{c} \mathcal{H}_k$ is a subspace of dimension $d = \dim \mathcal{S}$ that contains $\mathcal{S}$ and thus it must be the case that $\bigcap_{k=1}^{c} \mathcal{H}_k = \mathcal{S}$. Hence we may declare a point to be an inlier if and only if the point lies in the intersection of these c hyperplanes.

3.3 Hyperplane pursuit by ℓ_1 minimization

In this section we propose an optimization framework for the computation of a hyperplane that solves Problem 1, i.e., a hyperplane that contains all the inliers. To proceed, we need a definition.

Definition 2 *A hyperplane $\mathcal{H}$ of $\mathbb{R}^D$ is called maximal with respect to the dataset $\tilde{\boldsymbol{\mathcal{X}}}$, if it contains a maximal number of data points in $\tilde{\boldsymbol{\mathcal{X}}}$, i.e., if for any other hyperplane $\mathcal{H}^\dagger$ of $\mathbb{R}^D$ we have that $\mathrm{Card}(\tilde{\boldsymbol{\mathcal{X}}} \cap \mathcal{H}) \geq \mathrm{Card}(\tilde{\boldsymbol{\mathcal{X}}} \cap \mathcal{H}^\dagger)$.*

In principle, hyperplanes that are maximal with respect to $\tilde{\boldsymbol{\mathcal{X}}}$, always solve Problem 1, as the next proposition shows (see §5.1 for the proof).

Proposition 3 *Suppose that $N \geq d+1$ and $M \geq D-d$, and let $\mathcal{H}$ be a hyperplane that is maximal with respect to the dataset $\tilde{\boldsymbol{\mathcal{X}}}$. Then $\mathcal{H}$ contains all the inliers $\boldsymbol{\mathcal{X}}$.*

In view of Proposition 3, we may restrict our search for hyperplanes that contain all the inliers $\boldsymbol{\mathcal{X}}$ to the subset of hyperplanes that are maximal with respect to the dataset $\tilde{\boldsymbol{\mathcal{X}}}$. The advantage of this approach is immediate: the set of hyperplanes that are maximal with respect to $\tilde{\boldsymbol{\mathcal{X}}}$ is in principle computable, since it is precisely the set of solutions of the following optimization problem

$$\min_{\boldsymbol{b}} \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\right\|_0 \;\text{ s.t. }\; \boldsymbol{b} \neq 0. \tag{8}$$

The idea behind (8) is that a hyperplane $\mathcal{H} = \mathrm{Span}(\boldsymbol{b})^\perp$ contains a maximal number of columns of $\tilde{\boldsymbol{\mathcal{X}}}$ if and only if its normal vector $\boldsymbol{b}$ has a maximal *cosparsity* level with respect to the matrix $\tilde{\boldsymbol{\mathcal{X}}}^\top$, i.e., the number of non-zero entries of $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}$ is minimal. Since (8) is a combinatorial problem admitting no efficient solution, we consider its natural relaxation

$$\min_{\boldsymbol{b}} \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\right\|_1 \;\text{ s.t. }\; \left\|\boldsymbol{b}\right\|_2 = 1, \tag{9}$$

which in our context we will be referring to as *Dual Principal Component Pursuit* (DPCP). A major question that arises, to be answered in Theorem 11, is under what conditions every global solution of (9) is orthogonal to the inlier subspace $\mathrm{Span}(\boldsymbol{\mathcal{X}})$. A second major question, raised by the non-convexity of the constraint $\boldsymbol{b} \in \mathbb{S}^{D-1}$, is how to efficiently solve (9) with theoretical guarantees.

5. By the hyperplanes being linearly independent we mean that their normal vectors are linearly independent.

We emphasize here that the optimization problem (9) is far from new; interestingly, its earliest appearance in the literature that we are aware of is in Späth and Watson (1987), where the authors proposed to solve it by means of the recursion of convex problems given by[6]

$$\boldsymbol{n}_{k+1} := \operatorname*{argmin}_{\boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1} \left\| \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b} \right\|_1. \tag{10}$$

Notice that at each iteration of (10) the problem that is solved is computationally equivalent to a linear program; this makes the recursion (10) a very appealing candidate for solving the non-convex (9). Even though Späth and Watson (1987) proved the very interesting result that (10) converges to a critical point of (9) in a finite number of steps (see Appendix A), there is no reason to believe that in general (10) converges to a global minimum of (9).

Other works in which optimization problem (9) appears are Spielman et al. (2013); Qu et al. (2014); Sun et al. (2015c,d,b,a). More specifically, Spielman et al. (2013) propose to solve (9) by replacing the quadratic constraint $\boldsymbol{b}^\top \boldsymbol{b} = 1$ with a linear constraint $\boldsymbol{b}^\top \boldsymbol{w} = 1$ for some vector $\boldsymbol{w}$. In Qu et al. (2014); Sun et al. (2015b) (9) is approximately solved by alternating minimization, while a Riemannian trust-region approach is employed in Sun et al. (2015a). Finally, we note that problem (9) is closely related to the non-convex problem (5) associated with REAPER. To see this, suppose that the REAPER orthoprojector $\boldsymbol{\Pi}$ appearing in (5), represents the orthogonal projection to a hyperplane $\mathcal{U}$ with unit-ℓ_2 normal vector $\boldsymbol{b}$. In such a case $\boldsymbol{I}_D - \boldsymbol{\Pi} = \boldsymbol{b}\boldsymbol{b}^\top$ and it readily follows that problem (5) becomes identical to problem (9).

4. Dual Principal Component Pursuit Theory

In this section we establish our analysis framework and discuss our main theoretical results regarding the global optimum of the non-convex problem (9) as well as the recursion of convex relaxations in (10). We begin our theoretical investigation in §4.1 by establishing a connection between the *discrete* problems (9) and (10) and certain underlying *continuous* problems. The continuous problems do not depend on a finite set of inliers and outliers, rather on uniform distributions on the respective inlier and outlier spaces, and as such, are easier to analyze. The analysis of Theorems 5 and 6 reveals that the optimal solutions of the continuous analogue of (9) are orthogonal to the inlier space, and that the solutions of the continuous recursion corresponding to (10) converge to a normal vector to the inlier space, respectively. This suggests that under certain conditions on the distribution of the data, the same must be true for the *discrete* problem (9) and the *discrete* recursion (10), where the adjective *discrete* refers to the fact that these problems depend on a finite set of points. Our analysis of the discrete problems is inspired by the analysis of their continuous counterparts and the link between the two is formally captured through certain *discrepancy* bounds that we introduce in §4.2. In turn, these allow us to prove conditions under which we can characterize the global optimal of problem (9) as well as the convergence of recursion (10); this is done in §4.3 and the main results are Theorems 11 and 12, which are analogues of Theorems 5 and 6. These theorems suggest that both (9) and (10) are natural formulations for computing the orthogonal complement of a linear subspace in the presence of outliers. The proofs of all theorems as well as intermediate results are deferred to §5.

6. Being unaware of the work of Späth and Watson (1987), we independently proposed the same recursion in (Tsakiris and Vidal, 2015).

4.1 Formulation and theoretical analysis of the underlying continuous problems

In this section we show that the problems of interest (9) and (10) can be viewed as discrete versions of certain continuous problems, which are easier to analyze. To begin with, consider given outliers $\boldsymbol{\mathcal{O}} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_M] \subset \mathbb{S}^{D-1}$ and inliers $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N] \subset \mathcal{S} \cap \mathbb{S}^{D-1}$, and recall the notation $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\, \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma}$, where $\boldsymbol{\Gamma}$ is an unknown permutation. Next, for any $\boldsymbol{b} \in \mathbb{S}^{D-1}$ define the function $f_{\boldsymbol{b}} : \mathbb{S}^{D-1} \to \mathbb{R}$ by $f_{\boldsymbol{b}}(\boldsymbol{z}) = \left|\boldsymbol{b}^\top \boldsymbol{z}\right|$. Define also *discrete* measures $\mu_{\boldsymbol{\mathcal{O}}}$ and $\mu_{\boldsymbol{\mathcal{X}}}$ on $\mathbb{S}^{D-1}$ associated with the outliers and inliers respectively, as

$$\mu_{\boldsymbol{\mathcal{O}}}(\boldsymbol{z}) = \frac{1}{M}\sum_{j=1}^{M} \delta(\boldsymbol{z} - \boldsymbol{o}_j) \;\text{ and }\; \mu_{\boldsymbol{\mathcal{X}}}(\boldsymbol{z}) = \frac{1}{N}\sum_{j=1}^{N} \delta(\boldsymbol{z} - \boldsymbol{x}_j), \tag{11}$$

where $\delta(\cdot)$ is the Dirac function on $\mathbb{S}^{D-1}$, satisfying

$$\int_{\boldsymbol{z} \in \mathbb{S}^{D-1}} g(\boldsymbol{z}) \delta(\boldsymbol{z} - \boldsymbol{z}_0) d\mu_{\mathbb{S}^{D-1}} = g(\boldsymbol{z}_0), \tag{12}$$

for every $g : \mathbb{S}^{D-1} \to \mathbb{R}$ and every $\boldsymbol{z}_0 \in \mathbb{S}^{D-1}$; $\mu_{\mathbb{S}^{D-1}}$ is the uniform measure on $\mathbb{S}^{D-1}$.

With these definitions, we have that the objective function $\left\| \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b} \right\|_1$ appearing in (9) and (10) is the sum of the weighted expectations of the function $f_{\boldsymbol{b}}$ under the measures $\mu_{\boldsymbol{\mathcal{O}}}$ and $\mu_{\boldsymbol{\mathcal{X}}}$, i.e.,

$$\left\| \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b} \right\|_1 = \left\| \boldsymbol{\mathcal{O}}^\top \boldsymbol{b} \right\|_1 + \left\| \boldsymbol{\mathcal{X}}^\top \boldsymbol{b} \right\|_1 = \sum_{j=1}^{M} \left|\boldsymbol{b}^\top \boldsymbol{o}_j\right| + \sum_{j=1}^{N} \left|\boldsymbol{b}^\top \boldsymbol{x}_j\right| \tag{13}$$

$$= \sum_{j=1}^{M} \int_{\boldsymbol{z} \in \mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| \delta(\boldsymbol{z} - \boldsymbol{o}_j) d\mu_{\mathbb{S}^{D-1}} + \sum_{j=1}^{N} \int_{\boldsymbol{z} \in \mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| \delta(\boldsymbol{z} - \boldsymbol{x}_j) d\mu_{\mathbb{S}^{D-1}} \tag{14}$$

$$= \int_{\boldsymbol{z} \in \mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| \sum_{j=1}^{M} \delta(\boldsymbol{z} - \boldsymbol{o}_j) d\mu_{\mathbb{S}^{D-1}} + \int_{\boldsymbol{z} \in \mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| \sum_{j=1}^{N} \delta(\boldsymbol{z} - \boldsymbol{x}_j) d\mu_{\mathbb{S}^{D-1}} \tag{15}$$

$$= M\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{O}}}}(f_{\boldsymbol{b}}) + N\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{X}}}}(f_{\boldsymbol{b}}). \tag{16}$$

Hence, the optimization problem (9), which we repeat here for convenience,

$$\min_{\boldsymbol{b}} \left\| \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b} \right\|_1 \;\text{ s.t. }\; \boldsymbol{b}^\top \boldsymbol{b} = 1, \tag{17}$$

is equivalent to the problem

$$\min_{\boldsymbol{b}} \left[M\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{O}}}}(f_{\boldsymbol{b}}) + N\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{X}}}}(f_{\boldsymbol{b}}) \right] \;\text{ s.t. }\; \boldsymbol{b}^\top \boldsymbol{b} = 1. \tag{18}$$

Similarly, the recursion (10), repeated here for convenience,

$$\boldsymbol{n}_{k+1} = \operatorname*{argmin}_{\boldsymbol{b}} \left\| \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b} \right\|_1 \;\text{ s.t. }\; \boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1, \tag{19}$$

is equivalent to the recursion

$$\boldsymbol{n}_{k+1} = \operatorname*{argmin}_{\boldsymbol{b}} \left[M\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{O}}}}(f_{\boldsymbol{b}}) + N\, \mathbb{E}_{\mu_{\boldsymbol{\mathcal{X}}}}(f_{\boldsymbol{b}}) \right] \;\text{ s.t. }\; \boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1. \tag{20}$$

Now, the discrete measures $\mu_{\boldsymbol{\mathcal{O}}}, \mu_{\boldsymbol{\mathcal{X}}}$ of (11), are discretizations of the continuous measures $\mu_{\mathbb{S}^{D-1}}$, and $\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}$ respectively, where the latter is the uniform measure on $\mathbb{S}^{D-1}\cap\mathcal{S}$. Hence, for the purpose of understanding the properties of the global minimizer of (18) and the limiting point of (20), it is meaningful to replace in (18) and (20) the discrete measures $\mu_{\boldsymbol{\mathcal{O}}}$ and $\mu_{\boldsymbol{\mathcal{X}}}$ by their continuous counterparts $\mu_{\mathbb{S}^{D-1}}$ and $\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}$, and study the resulting *continuous* problems

$$\min_{\boldsymbol{b}} \left[M\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}}}(f_{\boldsymbol{b}}) + N\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}}(f_{\boldsymbol{b}}) \right] \;\text{ s.t. }\; \boldsymbol{b}^\top\boldsymbol{b} = 1, \tag{21}$$

$$\boldsymbol{n}_{k+1} = \operatorname*{argmin}_{\boldsymbol{b}} \left[M\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}}}(f_{\boldsymbol{b}}) + N\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}}(f_{\boldsymbol{b}}) \right] \;\text{ s.t. }\; \boldsymbol{b}^\top\hat{\boldsymbol{n}}_k = 1. \tag{22}$$

It is important to note that if these two continuous problems have the geometric properties of interest, i.e., if every global solution of (21) is a vector orthogonal to the inlier subspace, and similarly, if the sequence of vectors $\{\boldsymbol{n}_k\}$ produced by (22) converges to a vector $\boldsymbol{n}_{k^*}$ orthogonal to the inlier subspace, then this *correctness* of the continuous problems can be viewed as a first theoretical verification of the correctness of the *discrete* formulations (9) and (10). The objective of the rest of this section is to establish that this is precisely the case.

Before discussing our main two results in this direction, we note that the continuous objective function appearing in (21) and (22) can be re-written in a more suggestive form. To see what that is, define c_D as the average height of the unit hemisphere of $\mathbb{R}^D$, directly computed as

$$c_D := \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} |z_1| d\mu_{\mathbb{S}^{D-1}} = \frac{(D-2)!!}{(D-1)!!} \cdot \begin{cases} \frac{2}{\pi} & \text{if } D \text{ even}, \\ 1 & \text{if } D \text{ odd}, \end{cases} \tag{23}$$

where z_1 is the first coordinate of the vector $\boldsymbol{z}$, and the double factorial is defined as

$$k!! := \begin{cases} k(k-2)(k-4)\cdots 4\cdot 2 & \text{if } k \text{ even}, \\ k(k-2)(k-4)\cdots 3\cdot 1 & \text{if } k \text{ odd}. \end{cases} \tag{24}$$

Then we have the following result, whose proof can be found in §5.2.

Proposition 4 *The objective function of the continuous problem* (21) *can be rewritten as:*

$$M\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}}}(f_{\boldsymbol{b}}) + N\,\mathbb{E}_{\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}}(f_{\boldsymbol{b}}) = \|\boldsymbol{b}\|_2 \left(Mc_D + Nc_d\cos(\phi)\right), \tag{25}$$

where ϕ is the principal angle between $\boldsymbol{b}$ and the subspace $\mathcal{S}$.

As a consequence of this result, when $\boldsymbol{b}\in\mathbb{S}^{D-1}$, the first term (the outlier term) of the objective function becomes a constant (Mc_D) and hence the outliers do not affect the optimal solution of (21). Moreover, the second term (the inlier term) of the objective function depends only on the cosine of the principal angle between $\boldsymbol{b}$ and the subspace, which is minimized when $\boldsymbol{b}$ is orthogonal to the subspace ($\phi = \pi/2$). This leads to the following result about the continuous problem, whose proof can be found in §5.3.

Theorem 5 *Any global solution to problem* (21) *must be orthogonal to $\mathcal{S}$.*

Observe that this result is true irrespective of the weight M of the outlier term or the weight N of the inlier term in the continuous objective function (25). Similarly, the next result, whose proof can be found in §5.4, shows that the solutions to the continuous recursion in (22) converge to a vector orthogonal to the inlier subspace in a finite number of steps, regardless of the outlier and inlier weights M, N.

Theorem 6 *Consider the sequence $\{\boldsymbol{n}_k\}_{k\geq 0}$ generated by recursion* (22)*, with $\hat{\boldsymbol{n}}_0 \in \mathbb{S}^{D-1}$. Let ϕ_0 be the principal angle of $\boldsymbol{n}_0$ from $\mathcal{S}$, and define $\alpha := Nc_d/Mc_D$. Then, as long as $\phi_0 > 0$, the sequence $\{\boldsymbol{n}_k\}_{k\geq 0}$ converges to a unit ℓ_2-norm element of $\mathcal{S}^\perp$ in a finite number k^* of iterations, where $k^* = 0$ if $\phi_0 = \pi/2$, $k^* = 1$ if $\tan(\phi_0) \geq 1/\alpha$, and $k^* \leq \left\lceil \frac{\tan^{-1}(1/\alpha)-\phi_0}{\sin^{-1}(\alpha\sin(\phi_0))} \right\rceil + 1$ otherwise.*

Notice the remarkable fact that according to Theorem 6, the continuous recursion (22) converges to a vector orthogonal to the inlier subspace $\mathcal{S}$ in a *finite* number of steps. Moreover, if the relation

$$\tan(\phi_0) \geq 1/\alpha = \frac{M}{N}\frac{c_D}{c_d}, \tag{26}$$

holds true, then this convergence occurs in a single step. One way to interpret (26) is to notice that as long as the angle ϕ_0 of the initial estimate $\hat{\boldsymbol{n}}_0$ from the inlier subspace is positive, and for any arbitrary but fixed number of outliers M, there is always a sufficiently large number N of inliers, such that (26) is satisfied and thus convergence occurs in one step. Likewise, condition (26) can also be satisfied if d/D is sufficiently small (so that c_D/c_d is small). Conversely, for any fixed number of inliers N and outliers M, there is always a sufficiently large angle ϕ_0 such that (26) is true, and thus (22) again converges in a single step. More generally, even when (26) is not true, the larger ϕ_0, N are, the smaller the quantity

$$\left\lceil \frac{\tan^{-1}(1/\alpha)-\phi_0}{\sin^{-1}(\alpha\sin(\phi_0))} \right\rceil \tag{27}$$

is, and thus according to Theorem 5 the faster (22) converges.

4.2 Discrepancy bounds between the continuous and discrete problems

The heart of our analysis framework is to bound the deviation of some underlying geometric quantities, which we call the *average outlier* and the *average inlier with respect to* $\boldsymbol{b}$, from their continuous counterparts. To begin with, recall our discrete objective function

$$\mathcal{J}_{\text{discrete}}(\boldsymbol{b}) = \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top\boldsymbol{b}\right\|_1 = \left\|\boldsymbol{\mathcal{O}}^\top\boldsymbol{b}\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top\boldsymbol{b}\right\|_1 \tag{28}$$

and its continuous counterpart

$$\mathcal{J}_{\text{continuous}}(\boldsymbol{b}) = \|\boldsymbol{b}\|_2 \left(Mc_D + Nc_d\cos(\phi)\right). \tag{29}$$

Now, notice that the term of the discrete objective that depends on the outliers $\boldsymbol{\mathcal{O}}$ can be written as

$$\left\|\boldsymbol{\mathcal{O}}^\top\boldsymbol{b}\right\|_1 = \sum_{j=1}^{M} |\boldsymbol{o}_j^\top\boldsymbol{b}| = \sum_{j=1}^{M} \boldsymbol{b}^\top \operatorname{Sign}(\boldsymbol{o}_j^\top\boldsymbol{b})\boldsymbol{o}_j = M\,\boldsymbol{b}^\top\boldsymbol{o}_{\boldsymbol{b}}, \tag{30}$$

where $\operatorname{Sign}(\cdot)$ is the sign function and $\boldsymbol{o}_b$ is the *average outlier with respect to* $\boldsymbol{b}$, defined as

$$\boldsymbol{o}_{\boldsymbol{b}} := \frac{1}{M}\sum_{j=1}^{M} \operatorname{Sign}(\boldsymbol{b}^\top\boldsymbol{o}_j)\boldsymbol{o}_j. \tag{31}$$

Defining a vector valued function $\boldsymbol{f}_{\boldsymbol{b}} : \mathbb{S}^{D-1} \to \mathbb{R}^D$ by $\boldsymbol{z} \in \mathbb{S}^{D-1} \overset{\boldsymbol{f}_{\boldsymbol{b}}}{\longmapsto} \text{Sign}(\boldsymbol{b}^\top \boldsymbol{z})\boldsymbol{z}$, we notice that

$$\boldsymbol{o}_{\boldsymbol{b}} = \frac{1}{M}\sum_{j=1}^{M} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{o}_j) = \frac{1}{M}\sum_{j=1}^{M} \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z})\delta(\boldsymbol{z}-\boldsymbol{o}_j)d\mu_{S^{D-1}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z})d\mu_{\boldsymbol{\mathcal{O}}}(\boldsymbol{z}), \quad (32)$$

where $\mu_{\boldsymbol{\mathcal{O}}}(\boldsymbol{z})$ is defined in (11), and so $\boldsymbol{o}_{\boldsymbol{b}}$ is a discrete approximation to the continuous integral $\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{z})d\mu_{\mathbb{S}^{D-1}}$, whose value is given by the next Lemma (see §5.5 for the proof).

Lemma 7 *Recall the definition of c_D in* (23)*. For any $\boldsymbol{b} \in \mathbb{S}^{D-1}$ we have*

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{z})d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \text{Sign}(\boldsymbol{b}^\top\boldsymbol{z})\boldsymbol{z}d\mu_{\mathbb{S}^{D-1}} = c_D\,\boldsymbol{b}. \quad (33)$$

In other words, the continuous average outlier with respect to $\boldsymbol{b}$ is $c_D\boldsymbol{b}$. We define $\epsilon_{\boldsymbol{\mathcal{O}},M}$ to be the maximum error between the discrete and continuous average outliers as $\boldsymbol{b}$ varies on $\mathbb{S}^{D-1}$, i.e.,

$$\epsilon_{\boldsymbol{\mathcal{O}},M} \coloneqq \max_{\boldsymbol{b}\in\mathbb{S}^{D-1}} \left\|c_D\,\boldsymbol{b} - \boldsymbol{o}_{\boldsymbol{b}}\right\|_2, \quad (34)$$

and we establish that the more *uniformly distributed* $\boldsymbol{\mathcal{O}} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_M] \subset \mathbb{S}^{D-1}$ is the smaller $\epsilon_{\boldsymbol{\mathcal{O}},M}$ becomes. The notion of uniformity of $\boldsymbol{\mathcal{O}}$ that we use here is a deterministic one and is captured by the *spherical cap discrepancy* of the set $\boldsymbol{\mathcal{O}}$, defined as (Grabner et al., 1997; Grabner and Tichy, 1993)

$$\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}}) := \sup_{\mathcal{C}} \left|\frac{1}{M}\sum_{j=1}^{M} \mathbb{I}_{\mathcal{C}}(\boldsymbol{o}_j) - \mu_{\mathbb{S}^{D-1}}(\mathcal{C})\right|. \quad (35)$$

In (35) the supremum is taken over all spherical caps $\mathcal{C}$ of the sphere $\mathbb{S}^{D-1}$, where a spherical cap is the intersection of $\mathbb{S}^{D-1}$ with a half-space of $\mathbb{R}^D$, and $\mathbb{I}_{\mathcal{C}}(\cdot)$ is the indicator function of $\mathcal{C}$, which takes the value 1 inside $\mathcal{C}$ and zero otherwise. The spherical cap discrepancy $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$ is precisely the supremum among all errors in approximating integrals of indicator functions of spherical caps via averages of such indicator functions on the point set $\boldsymbol{\mathcal{O}}$. Intuitively, $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$ captures how close the discrete measure $\mu_{\boldsymbol{\mathcal{O}}}$ (see equation (11)) associated with $\boldsymbol{\mathcal{O}}$ is to the measure $\mu_{\mathbb{S}^{D-1}}$. We will say that $\boldsymbol{\mathcal{O}}$ is uniformly distributed on $\mathbb{S}^{D-1}$ if $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$ is small. We note here that as a function of the number of points M, $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$ decreases with a rate of (Dick, 2014; Beck, 1984)

$$\sqrt{\log(M)}M^{-\frac{1}{2}-\frac{1}{2(D-1)}}. \quad (36)$$

As a consequence, to show that uniformly distributed points $\boldsymbol{\mathcal{O}}$ correspond to small $\epsilon_{\boldsymbol{\mathcal{O}},M}$, it suffices to bound the maximum integration error $\epsilon_{\boldsymbol{\mathcal{O}},M}$ from above by a quantity proportional to the spherical cap discrepancy $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$. Inequalities that bound from above the approximation error of the integral of a function in terms of the variation of the function and the discrepancy of a finite set of points (not necessarily the spherical cap discrepancy; there are several types of discrepancies) are widely known as *Koksma-Hlawka inequalites* (Kuipers and Niederreiter, 2012; Hlawka, 1971). Even though such inequalities exist and are well-known for integration of functions on the unit hypercube $[0,1]^D$ (Kuipers and Niederreiter, 2012; Hlawka, 1971; Harman, 2010), similar inequalities for integration of functions on the unit sphere $\mathbb{S}^{D-1}$ seem not to be known in general

(Grabner and Tichy, 1993), except if one makes additional assumptions on the distribution of the finite set of points (Grabner et al., 1997; Brauchart and Grabner, 2015). Nevertheless, the function $f_{\boldsymbol{b}} : \boldsymbol{z} \longmapsto |\boldsymbol{b}^\top\boldsymbol{z}|$ that is associated with $\epsilon_{\boldsymbol{\mathcal{O}},M}$ is simple enough to allow for a Koksma-Hlawka inequality of its own, as described in the next lemma, whose proof can be found in §5.6.[7]

Lemma 8 *Let $\boldsymbol{\mathcal{O}} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_M]$ be a finite subset of $\mathbb{S}^{D-1}$. Then*

$$\epsilon_{\boldsymbol{\mathcal{O}},M} = \max_{\boldsymbol{b}\in\mathbb{S}^{D-1}} \left\|c_D\boldsymbol{b} - \boldsymbol{o}_{\boldsymbol{b}}\right\|_2 \leq \sqrt{5}\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}}), \quad (37)$$

where $c_D, \boldsymbol{o}_{\boldsymbol{b}}$ and $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$ are defined in (23)*,* (31) *and* (35) *respectively.*

We now turn our attention to the inlier term $\left\|\tilde{\boldsymbol{\mathcal{X}}}^\top\boldsymbol{b}\right\|_1$ of the discrete objective function (28), which is slightly more complicated than the outlier term. We have

$$\left\|\boldsymbol{\mathcal{X}}^\top\boldsymbol{b}\right\|_1 = \sum_{j=1}^{N} |\boldsymbol{x}_j^\top\boldsymbol{b}| = \sum_{j=1}^{N} \boldsymbol{b}^\top \text{Sign}(\boldsymbol{x}_j^\top\boldsymbol{b})\boldsymbol{x}_j = N\,\boldsymbol{b}^\top\boldsymbol{x}_{\boldsymbol{b}}, \quad (38)$$

where

$$\boldsymbol{x}_{\boldsymbol{b}} \coloneqq \frac{1}{N}\sum_{j=1}^{N} \text{Sign}(\boldsymbol{b}^\top\boldsymbol{x}_j)\boldsymbol{x}_j = \frac{1}{N}\sum_{j=1}^{N} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{x}_j) = \int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} f_{\boldsymbol{b}}(\boldsymbol{x})d\mu_{\boldsymbol{\mathcal{X}}}(\boldsymbol{x}) \quad (39)$$

is the *average inlier with respect to $\boldsymbol{b}$*. Thus, $\boldsymbol{x}_{\boldsymbol{b}}$ is a discrete approximation of the integral

$$\int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{x})d\mu_{\mathbb{S}^{D-1}}, \quad (40)$$

whose value is given by the next lemma (see §5.7 for the proof).

Lemma 9 *For any $\boldsymbol{b} \in \mathbb{S}^{D-1}$ we have*

$$\int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{x})d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \text{Sign}(\boldsymbol{b}^\top\boldsymbol{x})\boldsymbol{x}d\mu_{\mathbb{S}^{D-1}} = c_d\,\hat{\boldsymbol{v}}, \quad (41)$$

where c_d is given by (23) *after replacing D with d, and $\boldsymbol{v}$ is the orthogonal projection of $\boldsymbol{b}$ onto $\mathcal{S}$.*

In other words, the continuous average inlier with respect to $\boldsymbol{b}$ is $c_d\hat{\boldsymbol{v}}$. We define $\epsilon_{\boldsymbol{\mathcal{X}}}$ to be the maximum error between the discrete and continuous average inliers as $\boldsymbol{b}$ varies on $\mathbb{S}^{D-1}$, which is the same as the maximum error as $\boldsymbol{b}$ varies on $\mathbb{S}^{D-1}\cap\mathcal{S}$, i.e.,

$$\epsilon_{\boldsymbol{\mathcal{X}},N} \coloneqq \max_{\boldsymbol{b}\in\mathbb{S}^{D-1}} \left\|c_d\,\widehat{\pi_{\mathcal{S}}(\boldsymbol{b})} - \boldsymbol{x}_{\boldsymbol{b}}\right\|_2 = \max_{\boldsymbol{b}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \left\|c_d\,\boldsymbol{b} - \boldsymbol{x}_{\boldsymbol{b}}\right\|_2. \quad (42)$$

Then an almost identical argument as the one that established Lemma 8 gives that

$$\epsilon_{\boldsymbol{\mathcal{X}},N} \leq \sqrt{5}\mathfrak{S}_{d,N}(\boldsymbol{\mathcal{X}}), \quad (43)$$

where now the discrepancy $\mathfrak{S}_{d,N}(\boldsymbol{\mathcal{X}})$ of the inliers $\boldsymbol{\mathcal{X}}$ is defined exactly as in (35) except that M is replaced by N and the supremum is taken over all spherical caps of $\mathbb{S}^{D-1}\cap\mathcal{S} \cong \mathbb{S}^{d-1}$.

7. The authors are grateful to Prof. Glyn Harman for pointing out that such a result is possible as well as suggesting how to prove it.

4.3 Conditions for global optimality and convergence of the discrete problems

In this section we analyze the discrete problem (9) and the associated discrete recursion (10), where the adjective *discrete* refers to the fact that (9) and (10) depend on a finite set of points $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma}$ sampled from the union of the space of outliers $\mathbb{S}^{D-1}$ and the space of inliers $\mathbb{S}^{D-1} \cap \mathcal{S}$. In §4.1 we showed that these two problems are discrete versions of the continuous problems (21) and (22), respectively. We further showed that the continuous problems possess the geometric property of interest, i.e., every global minimizer of (21) must be an element of $\mathcal{S}^{\perp} \cap \mathbb{S}^{D-1}$ (Theorem 5) and the recursion (22) produces a sequence of vectors that converges in a finite number of steps to an element of $\mathcal{S}^{\perp} \cap \mathbb{S}^{D-1}$ (Theorem 6). In this section we use the discrepancy bounds of §4.2 to show that under some conditions on the uniformity of $\boldsymbol{\mathcal{X}} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N]$ and $\boldsymbol{\mathcal{O}} = [\boldsymbol{o}_1, \ldots, \boldsymbol{o}_M]$, a similar statement holds for problems (9) and (10). We start with a definition.

Definition 10 *Given a set $\boldsymbol{\mathcal{Y}} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L] \subset \mathbb{S}^{D-1}$ and an integer $K \leq L$, define $\mathcal{R}_{\boldsymbol{\mathcal{Y}},K}$ to be the maximum circumradius among all polytopes of the form*

$$\left\{ \sum_{i=1}^{K} \alpha_{j_i} \boldsymbol{y}_{j_i} \,:\, \alpha_{j_i} \in [-1,1] \right\}, \tag{44}$$

where $j_1, \ldots, j_K$ are distinct integers in $[L]$, and the circumradius of a bounded subset of $\mathbb{R}^D$ is the infimum over the radii of all Euclidean balls of $\mathbb{R}^D$ that contain that subset. With that, define

$$\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}} := \max_{K_1+K_2<D,\, K_2<d} (\mathcal{R}_{\boldsymbol{\mathcal{O}},K_1} + \mathcal{R}_{\boldsymbol{\mathcal{X}},K_2}). \tag{45}$$

The next theorem, proved in §5.8, states that if both inliers and outliers are sufficiently uniformly distributed, i.e., if the uniformity parameters $\epsilon_{\boldsymbol{\mathcal{X}},N}$ and $\epsilon_{\boldsymbol{\mathcal{O}},M}$ are sufficiently small, then every global solution of (9) must be orthogonal to the inlier subspace $\mathcal{S}$. More precisely,

Theorem 11 *Suppose that the ratio γ of outliers to inliers satisfies*

$$\gamma := \frac{M}{N} < \frac{1}{2\epsilon_{\boldsymbol{\mathcal{O}},M}} \min \left\{ c_d - \epsilon_{\boldsymbol{\mathcal{X}},N}, 2\left(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N} - \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} \right), \Gamma \right\}, \quad \textit{where} \tag{46}$$

$$\Gamma := \frac{c_d - \epsilon_{\boldsymbol{\mathcal{X}},N}}{2(3c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})} \left[\sqrt{\mathscr{P}^2 + 8(3c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N} - \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N})} - \mathscr{P} \right], \tag{47}$$

$$\mathscr{P} := 2\frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} + \epsilon_{\boldsymbol{\mathcal{X}},N} + c_d. \tag{48}$$

Then any global solution $\boldsymbol{b}^*$ to (9) must be orthogonal to $\mathrm{Span}(\boldsymbol{\mathcal{X}})$.

Towards interpreting Theorem 11, consider first the asymptotic case where we allow N and M to go to infinity, while keeping the ratio γ constant. Assuming that both inliers and outliers are perfectly well distributed in the limit, i.e., under the hypothesis that $\lim_{N\to\infty} \mathfrak{S}_{d,N}(\boldsymbol{\mathcal{X}}) = 0$ and $\lim_{M\to\infty} \mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}}) = 0$, Lemma 8 and inequality (43) give that $\lim_{N\to\infty} \epsilon_{\boldsymbol{\mathcal{X}},N} = 0$ and $\lim_{M\to\infty} \epsilon_{\boldsymbol{\mathcal{O}},M} = 0$, in which case (46) is satisfied. This suggests the interesting fact that (9) can possibly give a normal to the inliers even for arbitrarily many outliers, and irrespectively of the subspace dimension d. Along the same lines, for a given γ and under the point set uniformity hypothesis, we can always increase the number of inliers and outliers (thus decreasing $\epsilon_{\boldsymbol{\mathcal{X}},N}$ and

$\epsilon_{\boldsymbol{\mathcal{O}},M}$), while keeping γ constant, until (46) is satisfied, once again indicating that (9) can possibly yield a normal to the space of inliers irrespectively of their intrinsic dimension; this becomes evident in the numerical evaluation of Figs. 4(a)-4(c). Notice that the intrinsic dimension d of the inliers manifests itself through the quantity c_d, which we recall is a decreasing function of d. Consequently, the smaller d is the larger the RHS of (46) becomes, and so the easier it is to satisfy (46).

More explicitly (and less formally), because of (36) the quantities $\epsilon_{\boldsymbol{\mathcal{O}},M}, \epsilon_{\boldsymbol{\mathcal{X}},N}$ decay at an approximate rate of $1/\sqrt{M}, 1/\sqrt{N}$ respectively. In turn, this shows that the conditions (46) are satisfied if roughly $M < \mathcal{O}(N^2)$. To see this, note, e.g., that the first inequality in (46) reads

$$\frac{M}{N} < \frac{c_d - \epsilon_{\boldsymbol{\mathcal{X}},N}}{2\epsilon_{\boldsymbol{\mathcal{O}},M}}, \tag{49}$$

which roughly says that

$$\text{constant} \cdot \sqrt{M} \leq \text{constant} \cdot N - \text{constant} \cdot \sqrt{N}, \tag{50}$$

where by *constant* here we mean independent of M, N. A similar conclusion can be drawn from the rest inequalities in (46) [8]. In contrast, the analysis of the *haystack model* of REAPER (Lerman et al., 2015) gives $M < \mathcal{O}(N)$.

A similar phenomenon holds for the recursion of convex relaxations (10). Notice that according to Theorem 5, the continuous recursion converges in a finite number of iterations to a vector that is orthogonal to $\mathrm{Span}(\boldsymbol{\mathcal{X}}) = \mathcal{S}$, as long as the initialization $\hat{\boldsymbol{n}}_0$ does not lie in $\mathcal{S}$ (equivalently $\phi_0 > 0$). Intuitively, one should expect that in the discrete case, the conditions for the *discrete* recursion (10) to be successful, should be at least as strong as the conditions of Theorem 11, and strictly stronger than the condition $\phi_0 > 0$ of Theorem 6. Our next result, whose proof can be found in §5.9, formalizes this intuition.

Theorem 12 *Suppose that condition* (46) *holds true and consider the sequence $\{\boldsymbol{n}_k\}_{k\geq 0}$ generated by the recursion* (10)*. Let ϕ_0 be the principal angle of $\hat{\boldsymbol{n}}_0$ from $\mathrm{Span}(\boldsymbol{\mathcal{X}})$ and suppose that*

$$\cos(\phi_0) < \frac{c_d - \epsilon_{\boldsymbol{\mathcal{X}},N}}{2c_d(c_d + \epsilon_{\boldsymbol{\mathcal{X}},N})} \left[-\mathscr{Q} + \sqrt{\mathscr{Q}^2 + 4c_d(c_d - \mathscr{Q})} \right] - \frac{2\epsilon_{\boldsymbol{\mathcal{O}},M}}{c_d + \epsilon_{\boldsymbol{\mathcal{X}},N}} \frac{M}{N}, \tag{51}$$

$$\mathscr{Q} := \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} + \epsilon_{\boldsymbol{\mathcal{O}},M} \frac{M}{N} + \epsilon_{\boldsymbol{\mathcal{X}},N}. \tag{52}$$

Then after a finite number of iterations the sequence $\{\boldsymbol{n}_k\}_{k\geq 0}$ converges to a unit ℓ_2-norm vector that is orthogonal to $\mathrm{Span}(\boldsymbol{\mathcal{X}})$.

First note that if (46) is true, then the expression of (51) always defines an angle between 0 and $\pi/2$. Moreover, Theorem 12 can be interpreted using the same asymptotic arguments as Theorem 11; notice in particular that the lower bound on the angle ϕ_0 tends to zero as M, N go to infinity with γ constant, i.e., the more uniformly distributed inliers and outliers are, the closer $\boldsymbol{n}_0$ is allowed to be to $\mathrm{Span}(\boldsymbol{\mathcal{X}})$. We also emphasize that Theorem 12 asserts the correctness of the linear programming recursions (10) as far as recovering a vector $\boldsymbol{n}_{k^*}$ orthogonal to $\mathcal{S} := \mathrm{Span}(\boldsymbol{\mathcal{X}})$ is concerned. Even though this was our initial motivation for posing problem (9), Theorem 12 does not assert in general that $\boldsymbol{n}_{k^*}$ is a global minimizer of problem (9). However, this is indeed the case, when the inlier subspace $\mathcal{S}$ is a hyperplane, i.e., $d = D - 1$. This is because, up to a sign, there is a unique vector $\boldsymbol{b} \in \mathbb{S}^{D-1}$ that is orthogonal to $\mathcal{S}$ (the normal vector to the hyperplane), which, under conditions (46) and (51), is the unique global minimizer of (9), as well as the limit point $\boldsymbol{n}_{k^*}$ of Theorem 12.

8. It is the subject of ongoing research to arrive at this conclusion by more formal means.

5. Proofs

In this section we provide the proofs of all claims stated in earlier sections.

5.1 Proof of Proposition 3

By the general position hypothesis on $\boldsymbol{\mathcal{X}}$ and $\boldsymbol{\mathcal{O}}$, any hyperplane that does not contain $\boldsymbol{\mathcal{X}}$ can contain at most $D-1$ points from $\tilde{\boldsymbol{\mathcal{X}}}$. We will show that there exists a hyperplane that contains more than $D-1$ points of $\tilde{\boldsymbol{\mathcal{X}}}$. Indeed, take d inliers and $D-d-1$ outliers and let $\mathcal{H}$ be the hyperplane generated by these $D-1$ points. Denote the normal vector to that hyperplane by $\boldsymbol{b}$. Since $\mathcal{H}$ contains d inliers, $\boldsymbol{b}$ will be orthogonal to these inliers. Since $\boldsymbol{\mathcal{X}}$ is in general position, every d-tuple of inliers is a basis for $\mathrm{Span}(\boldsymbol{\mathcal{X}})$. As a consequence, $\boldsymbol{b}$ will be orthogonal to $\mathrm{Span}(\boldsymbol{\mathcal{X}})$, and in particular $\boldsymbol{b} \perp \boldsymbol{\mathcal{X}}$. This implies that $\boldsymbol{\mathcal{X}} \subset \mathcal{H}$ and so $\mathcal{H}$ will contain $N + D - d - 1 \geq d + 1 + D - d - 1 > D - 1$ points of $\tilde{\boldsymbol{\mathcal{X}}}$.

5.2 Proof of Proposition 4

Writing $\boldsymbol{b} = \|\boldsymbol{b}\|_2 \hat{\boldsymbol{b}}$, and letting $\boldsymbol{R}$ be a rotation that takes $\hat{\boldsymbol{b}}$ to the first standard basis vector $\boldsymbol{e}_1$, we see that the first expectation in the LHS of (25) becomes equal to

$$\begin{aligned}
\mathbb{E}_{\mu_{\mathbb{S}^{D-1}}}(f_{\boldsymbol{b}}) &= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z}) d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}} && (53)\\
&= \|\boldsymbol{b}\|_2 \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \left|\hat{\boldsymbol{b}}^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}} = \|\boldsymbol{b}\|_2 \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \left|\boldsymbol{z}^\top \boldsymbol{R}^{-1}\boldsymbol{R}\hat{\boldsymbol{b}}\right| d\mu_{\mathbb{S}^{D-1}} && (54)\\
&= \|\boldsymbol{b}\|_2 \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} |\boldsymbol{z}^\top \boldsymbol{e}_1| d\mu_{\mathbb{S}^{D-1}} = \|\boldsymbol{b}\|_2 \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} |z_1| d\mu_{\mathbb{S}^{D-1}} = \|\boldsymbol{b}\|_2 c_D, && (55)
\end{aligned}$$

where $\boldsymbol{z} = (z_1, \ldots, z_D)^\top$ is the coordinate representation of $\boldsymbol{z}$. To see what the second expectation in the LHS of (25) evaluates to, decompose $\boldsymbol{b}$ as $\boldsymbol{b} = \pi_{\mathcal{S}}(\boldsymbol{b}) + \pi_{\mathcal{S}^\perp}(\boldsymbol{b})$, and note that because the support of the measure $\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}$ is contained in $\mathcal{S}$, we must have that

$$\begin{aligned}
\mathbb{E}_{\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}}(f_{\boldsymbol{b}}) &= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \left|\boldsymbol{b}^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}} && (56)\\
&= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \left|(\pi_{\mathcal{S}}(\boldsymbol{b}))^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}} && (57)\\
&= \|\pi_{\mathcal{S}}(\boldsymbol{b})\|_2 \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \left|\left(\widehat{\pi_{\mathcal{S}}(\boldsymbol{b})}\right)^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}. && (58)
\end{aligned}$$

Writing $\boldsymbol{z}'$ and $\boldsymbol{b}'$ for the coordinate representation of $\boldsymbol{z}$ and $\widehat{\pi_{\mathcal{S}}(\boldsymbol{b})}$ with respect to a basis of $\mathcal{S}$, and noting that $\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}} \cong \mu_{\mathbb{S}^{d-1}}$, we have that

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \left|\left(\widehat{\pi_{\mathcal{S}}(\boldsymbol{b})}\right)^\top \boldsymbol{z}\right| d\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}} = \int_{\boldsymbol{z}'\in\mathbb{S}^{d-1}} \left|\boldsymbol{z}'^\top \boldsymbol{b}'\right| d\mu_{\mathbb{S}^{d-1}} = c_d, \tag{59}$$

where now c_d is the average height of the unit hemisphere of $\mathbb{R}^d$. Finally, noting that

$$\|\pi_{\mathcal{S}}(\boldsymbol{b})\|_2 = \|\boldsymbol{b}\|_2 \cos(\phi), \tag{60}$$

where ϕ is the principal angle of $\boldsymbol{b}$ from the subspace $\mathcal{S}$, we have that

$$\mathbb{E}_{\mu_{\mathbb{S}^{D-1}\cap\mathcal{S}}}(f_{\boldsymbol{b}}) = \|\boldsymbol{b}\|_2 c_d \cos(\phi). \tag{61}$$

5.3 Proof of Theorem 5

Because of the constraint $\boldsymbol{b}^\top \boldsymbol{b} = 1$ in (21), and using (25), problem (21) can be written as

$$\min_{\boldsymbol{b}} \ \left[Mc_D + Nc_d \cos(\phi)\right] \ \text{ s.t. } \ \boldsymbol{b}^\top \boldsymbol{b} = 1. \tag{62}$$

It is then immediate that the global minimum is equal to Mc_D and it is attained if and only if $\phi = \pi/2$, which corresponds to $\boldsymbol{b} \perp \mathcal{S}$.

5.4 Proof of Theorem 6

At iteration k the optimization problem associated with (22) is

$$\min_{\boldsymbol{b}\in\mathbb{R}^D} \ \mathcal{J}(\boldsymbol{b}) = \|\boldsymbol{b}\|_2 \left(Mc_D + Nc_d \cos(\phi)\right) \ \text{ s.t. } \ \boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1, \tag{63}$$

where ϕ is the principal angle of $\boldsymbol{b}$ from the subspace $\mathcal{S}$.

Let ϕ_k be the principal angle of $\hat{\boldsymbol{n}}_k$ from $\mathcal{S}$, and let $\boldsymbol{n}_{k+1}$ be a global minimizer of (63), with principal angle from $\mathcal{S}$ equal to ϕ_{k+1}. We show that $\phi_{k+1} \geq \phi_k$. To see this, note that the decrease in the objective function at iteration k is

$$\begin{aligned}
\mathcal{J}(\hat{\boldsymbol{n}}_k) - \mathcal{J}(\boldsymbol{n}_{k+1}) :=& M\, c_D \left\|\hat{\boldsymbol{n}}_k\right\|_2 + N\, c_d \left\|\hat{\boldsymbol{n}}_k\right\|_2 \cos(\phi_k)\\
&- M\, c_D \left\|\boldsymbol{n}_{k+1}\right\|_2 - N\, c_d \left\|\boldsymbol{n}_{k+1}\right\|_2 \cos(\phi_{k+1}).
\end{aligned} \tag{64}$$

Since $\boldsymbol{n}_{k+1}^\top \hat{\boldsymbol{n}}_k = 1$, we must have that $\left\|\boldsymbol{n}_{k+1}\right\|_2 \geq 1 = \left\|\hat{\boldsymbol{n}}_k\right\|_2$. Now if $\phi_{k+1} < \phi_k$, then $\cos(\phi_{k+1}) > \cos(\phi_k)$. But then (64) implies that $\mathcal{J}(\boldsymbol{n}_{k+1}) > \mathcal{J}(\hat{\boldsymbol{n}}_k)$, which is a contradiction on the optimality of $\boldsymbol{n}_{k+1}$. Hence it must be the case that $\phi_{k+1} \geq \phi_k$, and so the sequence $\{\phi_k\}_k$ is non-decreasing. In particular, since $\phi_0 > 0$ by hypothesis, we must also have $\phi_k > 0$, i.e., $\hat{\boldsymbol{n}}_k \notin \mathcal{S}$, $\forall k \geq 0$.

Letting ψ_k be the angle of $\boldsymbol{b}$ from $\hat{\boldsymbol{n}}_k$, the constraint $\boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1$ gives $0 \leq \psi_k < \pi/2$ and $\|\boldsymbol{b}\|_2 = 1/\cos(\psi_k)$, and so we can write the optimization problem (63) equivalently as

$$\min_{\boldsymbol{b}\in\mathbb{R}^D} \ \frac{Mc_D + Nc_d \cos(\phi)}{\cos(\psi_k)} \ \text{ s.t. } \ \boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1. \tag{65}$$

If $\hat{\boldsymbol{n}}_k$ is orthogonal to $\mathcal{S}$, i.e., $\phi_k = \pi/2$, then $\mathcal{J}(\hat{\boldsymbol{n}}_k) = Mc_D \leq \mathcal{J}(\boldsymbol{b})$, $\forall \boldsymbol{b} : \boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1$, with equality only if $\boldsymbol{b} = \hat{\boldsymbol{n}}_k$. As a consequence, $\boldsymbol{n}_{k'} = \hat{\boldsymbol{n}}_k$, $\forall k' > k$, and in particular if $\phi_0 = \pi/2$, then $k^* = 0$.

So suppose that $\phi_k < \pi/2$ and let $\hat{\boldsymbol{n}}_k^\perp$ be the normalized orthogonal projection of $\hat{\boldsymbol{n}}_k$ onto $\mathcal{S}^\perp$. We will prove that every global minimizer of problem (65) must lie in the two-dimensional plane $\mathcal{H} := \mathrm{Span}(\hat{\boldsymbol{n}}_k, \hat{\boldsymbol{n}}_k^\perp)$. To see this, let $\boldsymbol{b}$ have norm $1/\cos(\psi_k)$ for some $\psi_k < \pi/2$. If $\psi_k > \pi/2 - \phi_k$, then such a $\boldsymbol{b}$ can not be a global minimizer of (65), as the feasible vector $\hat{\boldsymbol{n}}_k^\perp / \sin(\phi_k) \in \mathcal{H}$ already gives a smaller objective, since

$$\mathcal{J}(\hat{\boldsymbol{n}}_k^\perp / \sin(\phi_k)) = \frac{Mc_D}{\sin(\phi_k)} = \frac{Mc_D}{\cos(\pi/2 - \phi_k)} < \frac{Mc_D + Nc_d \cos(\phi)}{\cos(\psi_k)} = \mathcal{J}(\boldsymbol{b}). \tag{66}$$

Thus, without loss of generality, we may restrict to the case where $\psi_k \leq \pi/2 - \phi_k$. Denote by $\hat{\boldsymbol{h}}_k$ the normalized projection of $\hat{\boldsymbol{n}}_k$ onto $\mathcal{S}$ and by $\hat{\boldsymbol{n}}^\dagger$ the vector that is obtained from $\hat{\boldsymbol{n}}_k$ by rotating it

towards $\hat{\boldsymbol{n}}_k^\perp$ by ψ_k. Note that both $\hat{\boldsymbol{h}}_k$ and $\hat{\boldsymbol{n}}_k^\dagger$ lie in $\mathcal{H}$. Letting $\Psi_k \in [0,\pi]$ be the spherical angle between the spherical arc formed by $\hat{\boldsymbol{n}}_k, \hat{\boldsymbol{b}}$ and the spherical arc formed by $\hat{\boldsymbol{n}}_k, \hat{\boldsymbol{h}}_k$, the spherical law of cosines gives

$$\cos(\angle \boldsymbol{b}, \hat{\boldsymbol{h}}_k) = \cos(\phi_k)\cos(\psi_k) + \sin(\phi_k)\sin(\psi_k)\cos(\Psi_k). \tag{67}$$

Now, Ψ_k is equal to π if and only if $\hat{\boldsymbol{n}}_k, \hat{\boldsymbol{h}}_k, \boldsymbol{b}$ are coplanar, i.e., if and only if $\boldsymbol{b} \in \mathcal{H}$. Suppose that $\boldsymbol{b} \notin \mathcal{H}$. Then $\Psi_k < \pi$, and so $\cos(\Psi_k) > -1$, which implies that

$$\cos(\angle \boldsymbol{b}, \hat{\boldsymbol{h}}_k) > \cos(\phi_k)\cos(\psi_k) - \sin(\phi_k)\sin(\psi_k) = \cos(\phi_k + \psi_k). \tag{68}$$

This in turn implies that the principal angle ϕ of $\boldsymbol{b}$ from $\mathcal{S}$ is strictly smaller than $\phi_k + \psi_k$, and so

$$\mathcal{J}(\boldsymbol{b}) = \frac{Mc_D + Nc_d\cos(\phi)}{\cos(\psi_k)} > \frac{Mc_D + Nc_d\cos(\phi_k+\psi_k)}{\cos(\psi_k)} = \mathcal{J}(\hat{\boldsymbol{n}}_k^\dagger/\cos(\psi_k)), \tag{69}$$

i.e., the feasible vector $\hat{\boldsymbol{n}}_k^\dagger/\cos(\psi_k) \in \mathcal{H}$ gives strictly smaller objective than $\boldsymbol{b}$.

To summarize, for the case where $\phi_k < \pi/2$, we have shown that any global minimizer $\boldsymbol{b}$ of (65) must i) have angle ψ_k from $\hat{\boldsymbol{n}}_k$ less or equal to $\pi/2 - \phi_k$, and ii) it must lie in $\mathrm{Span}(\hat{\boldsymbol{n}}_k, \hat{\boldsymbol{n}}_k^\perp)$. Hence, we can rewrite (65) in the equivalent form

$$\min_{\psi \in [-\pi/2+\phi_k, \pi/2-\phi_k]} \mathcal{J}_k(\psi) := \frac{Mc_D + Nc_d\cos(\phi_k+\psi)}{\cos(\psi_k)}, \tag{70}$$

where now ψ_k takes positive values as $\boldsymbol{b}$ approaches $\hat{\boldsymbol{n}}_k^\perp$ and negative values as it approaches $\hat{\boldsymbol{h}}_k$. The function $\mathcal{J}_k$ is continuous and differentiable in the interval $[-\pi/2+\phi_k, \pi/2-\phi_k]$, with derivative given by

$$\frac{\partial \mathcal{J}_k}{\partial \psi} = \frac{Mc_D\sin(\psi) - Nc_d\sin(\phi_k)}{\cos^2(\psi)}. \tag{71}$$

Setting the derivative to zero gives

$$\sin(\psi) = \alpha\sin(\phi_k). \tag{72}$$

If $\alpha\sin(\phi_k) \geq \sin(\pi/2 - \phi_k) = \cos(\phi_k)$, or equivalently $\tan(\phi_k) \geq 1/\alpha$, then $\mathcal{J}_k$ is strictly decreasing in the interval $[-\pi/2+\phi_k, \pi/2-\phi_k]$, and so it must attain its minimum precisely at $\psi = \pi/2 - \phi_k$, which corresponds to the choice $\boldsymbol{n}_{k+1} = \hat{\boldsymbol{n}}_k^\perp/\sin(\phi_k)$. Then by an earlier argument we must have that $\hat{\boldsymbol{n}}_{k'} \perp \mathcal{S}$, $\forall k' \geq k+1$. If, on the other hand, $\tan(\phi_k) < 1/\alpha$, then the equation (72) defines an angle

$$\psi_k^* := \sin^{-1}(\alpha\sin(\phi_k)) \in (0, \pi/2 - \phi_k), \tag{73}$$

at which $\mathcal{J}_k$ must attain its global minimum, since

$$\frac{\partial^2 \mathcal{J}_k}{\partial \psi^2}(\psi_k^*) = \frac{1}{\cos(\psi_k^*)} > 0. \tag{74}$$

As a consequence, if $\tan(\phi_k) < 1/\alpha$, then

$$\phi_{k+1} = \phi_k + \sin^{-1}(\alpha\sin(\phi_k)) < \pi/2. \tag{75}$$

We then see inductively that as long as $\tan(\phi_k) < 1/\alpha$, ϕ_k increases by a quantity which is bounded from below by $\sin^{-1}(\alpha\sin(\phi_0))$. Thus, ϕ_k will keep increasing until it becomes greater than the solution to the equation $\tan(\phi) = 1/\alpha$, at which point the global minimizer will be the vector $\boldsymbol{n}_{k+1} = \hat{\boldsymbol{n}}_k^\perp/\sin(\phi_k)$, and so $\hat{\boldsymbol{n}}_{k'} = \hat{\boldsymbol{n}}_{k+1}$, $\forall k' \geq k+1$. Finally, under the hypothesis that $\phi_k < \tan^{-1}(1/\alpha)$, we have

$$\phi_k = \phi_0 + \sum_{j=0}^{k-1} \sin^{-1}(\alpha\sin(\phi_j)) \geq \phi_0 + k\sin^{-1}(\alpha\sin(\phi_0)), \tag{76}$$

from where it follows that the maximal number of iterations needed for ϕ_k to become larger than $\tan^{-1}(1/\alpha)$ is $\left\lceil \frac{\tan^{-1}(1/\alpha)-\phi_0}{\sin^{-1}(\alpha\sin(\phi_0))} \right\rceil$, at which point at most one more iteration will be needed to achieve orthogonality to $\mathcal{S}$.

5.5 Proof of Lemma 7

Letting $\boldsymbol{R}$ be a rotation that takes $\boldsymbol{b}$ to the first canonical vector $\boldsymbol{e}_1$, i.e., $\boldsymbol{R}\boldsymbol{b} = \boldsymbol{e}_1$, we have that

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(\boldsymbol{b}^\top\boldsymbol{z})\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(\boldsymbol{b}^\top\boldsymbol{R}^\top\boldsymbol{R}\boldsymbol{z})\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}} \tag{77}$$

$$= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(\boldsymbol{e}_1^\top\boldsymbol{z})\boldsymbol{R}^\top\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}} \tag{78}$$

$$= \boldsymbol{R}^\top \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(z_1)\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}}, \tag{79}$$

where z_1 is the first cartesian coordinate of $\boldsymbol{z}$. Recalling the definition of c_D in equation (23), we see that

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(z_1)z_1\,d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} |z_1|\,d\mu_{\mathbb{S}^{D-1}} = c_D. \tag{80}$$

Moreover, for any $i > 1$, we have

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(z_1)z_i\,d\mu_{\mathbb{S}^{D-1}} = 0. \tag{81}$$

Consequently, the integral in (79) becomes

$$\int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(\boldsymbol{b}^\top\boldsymbol{z})\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}} = \boldsymbol{R}^\top \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}(z_1)\boldsymbol{z}\,d\mu_{\mathbb{S}^{D-1}} = \boldsymbol{R}^\top(c_D\boldsymbol{e}_1) = c_D\boldsymbol{b}. \tag{82}$$

5.6 Proof of Lemma 8

For any $\boldsymbol{b} \in \mathbb{S}^{D-1}$ we can write

$$c_D\boldsymbol{b} - \boldsymbol{o}_{\boldsymbol{b}} = \rho_1\boldsymbol{b} + \rho_2\boldsymbol{\zeta}, \tag{83}$$

for some vector $\boldsymbol{\zeta} \in \mathbb{S}^{D-1}$ orthogonal to $\boldsymbol{b}$, and so it is enough to show that $\sqrt{\rho_1^2 + \rho_2^2} \leq \sqrt{5}\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$. Let us first bound from above $|\rho_1|$ in terms of $\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$. Towards that end, observe that

$$\rho_1 = \boldsymbol{b}^\top(c_D\boldsymbol{b} - \boldsymbol{o}_{\boldsymbol{b}}) = c_D - \frac{1}{M}\sum_{j=1}^{M}\left|\boldsymbol{b}^\top\boldsymbol{o}_j\right| = \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z})d\mu_{\mathbb{S}^{D-1}} - \frac{1}{M}\sum_{j=1}^{M} f_{\boldsymbol{b}}(\boldsymbol{o}_j), \quad (84)$$

where the equality follows from the definition of c_D in (23) and recalling that $f_{\boldsymbol{b}}(\boldsymbol{z}) = \left|\boldsymbol{b}^\top\boldsymbol{z}\right|$. In other words, ρ_1 is the error in approximating the integral of $f_{\boldsymbol{b}}$ on $\mathbb{S}^{D-1}$ by the average of $f_{\boldsymbol{b}}$ on the point set $\boldsymbol{\mathcal{O}}$.

Now, notice that each *super-level set* $\left\{\boldsymbol{z} \in \mathbb{S}^{D-1} : f_{\boldsymbol{b}}(\boldsymbol{z}) \geq \alpha\right\}$ for $\alpha \in [0,1]$, is the union of two spherical caps, and also that

$$\sup_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z}) - \inf_{\boldsymbol{z}\in\mathbb{S}^{D-1}} f_{\boldsymbol{b}}(\boldsymbol{z}) = 1 - 0 = 1. \quad (85)$$

We these in mind, repeating the entire argument of the proof of Theorem 1 in (Harman, 2010) that lead to inequality (9) in (Harman, 2010), but now for a measurable function with respect to $\mu_{\mathbb{S}^{D-1}}$ (that would be $f_{\boldsymbol{b}}$), leads directly to

$$|\rho_1| \leq \mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}}). \quad (86)$$

For ρ_2 we have that

$$\rho_2 = \boldsymbol{\zeta}^\top(c_D\boldsymbol{b}) - \boldsymbol{\zeta}^\top\boldsymbol{o}_{\boldsymbol{b}} \quad (87)$$

$$= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} \mathrm{Sign}\left(\boldsymbol{b}^\top\boldsymbol{z}\right)\boldsymbol{\zeta}^\top\boldsymbol{z}d\mu_{\mathbb{S}^{D-1}} - \frac{1}{M}\sum_{j=1}^{M}\mathrm{Sign}\left(\boldsymbol{b}^\top\boldsymbol{o}_j\right)\boldsymbol{\zeta}^\top\boldsymbol{o}_j \quad (88)$$

$$= \int_{\boldsymbol{z}\in\mathbb{S}^{D-1}} g_{\boldsymbol{b},\boldsymbol{\zeta}}(\boldsymbol{z})d\mu_{\mathbb{S}^{D-1}} - \frac{1}{M}\sum_{j=1}^{M} g_{\boldsymbol{b},\boldsymbol{\zeta}}(\boldsymbol{o}_j), \quad (89)$$

where $g_{\boldsymbol{b},\boldsymbol{\zeta}} : \mathbb{S}^{D-1} \to \mathbb{R}$ is defined as $g_{\boldsymbol{b},\boldsymbol{\zeta}}(\boldsymbol{z}) = \mathrm{Sign}\left(\boldsymbol{b}^\top\boldsymbol{z}\right)\boldsymbol{\zeta}^\top\boldsymbol{z}$. Then a similar argument as for ρ_1, with the difference that now

$$\sup_{\boldsymbol{z}\in\mathbb{S}^{D-1}} g_{\boldsymbol{b},\boldsymbol{\zeta}}(\boldsymbol{z}) - \inf_{\boldsymbol{z}\in\mathbb{S}^{D-1}} g_{\boldsymbol{b},\boldsymbol{\zeta}}(\boldsymbol{z}) = 1 - (-1) = 2, \quad (90)$$

leads to

$$|\rho_2| \leq 2\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}}). \quad (91)$$

In view of (86), inequality (91) establishes that $\sqrt{\rho_1^2 + \rho_2^2} \leq \sqrt{5}\mathfrak{S}_{D,M}(\boldsymbol{\mathcal{O}})$, which concludes the proof of the lemma.

5.7 Proof of Lemma 9

Since $\boldsymbol{x}$ lies in $\mathcal{S}$, we have $\boldsymbol{f}_{\boldsymbol{b}}(\boldsymbol{x}) = \boldsymbol{f}_{\boldsymbol{v}}(\boldsymbol{x}) = \boldsymbol{f}_{\hat{\boldsymbol{v}}}(\boldsymbol{x})$, so that

$$\int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \mathrm{Sign}(\boldsymbol{b}^\top\boldsymbol{x})\boldsymbol{x}d\mu_{\mathbb{S}^{D-1}} = \int_{\boldsymbol{x}\in\mathbb{S}^{D-1}\cap\mathcal{S}} \mathrm{Sign}(\hat{\boldsymbol{v}}^\top\boldsymbol{x})\boldsymbol{x}d\mu_{\mathbb{S}^{D-1}}. \quad (92)$$

Now express $\boldsymbol{x}$ and $\hat{\boldsymbol{v}}$ on a basis of $\mathcal{S}$, use Lemma 7 replacing D with d, and then switch back to the standard basis of $\mathbb{R}^D$.

5.8 Proof of Theorem 11

To prove the theorem we need the following lemma.

Lemma 13 *For any $\boldsymbol{b} \in \mathbb{S}^{D-1}$ we have that*

$$M(c_D + \epsilon_{\boldsymbol{\mathcal{O}},M}) \geq \left\|\boldsymbol{\mathcal{O}}^\top\boldsymbol{b}\right\|_1 \geq M(c_D - \epsilon_{\boldsymbol{\mathcal{O}},M}) \quad (93)$$

$$N(c_d + \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi) \geq \left\|\boldsymbol{\mathcal{X}}^\top\boldsymbol{b}\right\|_1 \geq N(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi). \quad (94)$$

Proof We only prove the second inequality as the first is even simpler. Let $\boldsymbol{v} \neq \boldsymbol{0}$ be the orthogonal projection of $\boldsymbol{b}$ onto $\mathcal{S}$. By definition of $\epsilon_{\boldsymbol{\mathcal{X}},N}$, there exists a vector $\boldsymbol{\xi} \in \mathcal{S}$ of ℓ_2 norm less or equal to $\epsilon_{\boldsymbol{\mathcal{X}},N}$, such that

$$\boldsymbol{x}_{\boldsymbol{v}} = \boldsymbol{x}_{\boldsymbol{b}} = \frac{1}{N}\sum_{j=1}^{N}\mathrm{Sign}(\boldsymbol{b}^\top\boldsymbol{x}_j)\boldsymbol{x}_j = c_d\hat{\boldsymbol{v}} + \boldsymbol{\xi}. \quad (95)$$

Taking inner product of both sides with $\boldsymbol{b}$ gives

$$\frac{1}{N}\left\|\boldsymbol{\mathcal{X}}^\top\boldsymbol{b}\right\|_1 = c_d\cos(\phi) + \boldsymbol{b}^\top\boldsymbol{\xi}. \quad (96)$$

Now, the result follows by noting that $\left|\boldsymbol{b}^\top\boldsymbol{\xi}\right| \leq \epsilon_{\boldsymbol{\mathcal{X}},N}\cos(\phi)$, since the principal angle of $\boldsymbol{b}$ from $\mathrm{Span}(\boldsymbol{\xi})$ can not be less then ϕ. ■

Now, let $\boldsymbol{b}^*$ be an optimal solution of (9). Then $\boldsymbol{b}^*$ must satisfy the first order optimality relation

$$\boldsymbol{0} \in \lambda\boldsymbol{b}^* + \tilde{\boldsymbol{\mathcal{X}}}\,\mathrm{Sgn}(\tilde{\boldsymbol{\mathcal{X}}}^\top\boldsymbol{b}^*), \quad (97)$$

where λ is a scalar Lagrange multiplier parameter, and Sgn is the sub-differential of the ℓ_1 norm. For the sake of contradiction, suppose that $\boldsymbol{b}^* \not\perp \mathcal{S}$. If $\boldsymbol{b}^* \in \mathcal{S}$, then using Lemma 13 we have

$$M\,c_D + M\,\epsilon_{\boldsymbol{\mathcal{O}}} \geq \min_{\boldsymbol{b}\perp\mathcal{S},\boldsymbol{b}^\top\boldsymbol{b}=1}\left\|\boldsymbol{\mathcal{O}}^\top\boldsymbol{b}\right\|_1 \geq \left\|\boldsymbol{\mathcal{O}}^\top\boldsymbol{b}^*\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top\boldsymbol{b}^*\right\|_1$$

$$\geq M\,c_D - M\,\epsilon_{\boldsymbol{\mathcal{O}}} + N\,c_d - N\,\epsilon_{\boldsymbol{\mathcal{X}}}, \quad (98)$$

which violates the first inequality of hypothesis (46). Hence, we can assume that $\boldsymbol{b}^* \notin \mathcal{S}$.

By the general position hypothesis as well as Proposition 14, $\boldsymbol{b}^*$ will be orthogonal to precisely $D-1$ points, among which K_1 points belong to $\boldsymbol{\mathcal{O}}$, say, without loss of generality, $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{K_1}$, and $0 \leq K_2 \leq d-1$ points belong to $\boldsymbol{\mathcal{X}}$, say $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K_2}$. Then there must exist real numbers $-1 \leq \alpha_j, \beta_j \leq 1$, such that

$$\lambda\boldsymbol{b}^* + \sum_{j=1}^{K_1}\alpha_j\boldsymbol{o}_j + \sum_{j=K_1+1}^{M}\mathrm{Sign}(\boldsymbol{o}_j^\top\boldsymbol{b}^*)\boldsymbol{o}_j + \sum_{j=1}^{K_2}\beta_j\boldsymbol{x}_j + \sum_{j=K_2+1}^{N}\mathrm{Sign}(\boldsymbol{x}_j^\top\boldsymbol{b}^*)\boldsymbol{x}_j = \boldsymbol{0}. \quad (99)$$

Since $\mathrm{Sign}(\boldsymbol{o}_j^\top\boldsymbol{b}^*) = 0, \forall j \leq K_1$ and similarly $\mathrm{Sign}(\boldsymbol{x}_j^\top\boldsymbol{b}^*) = 0, \forall j \leq K_2$, we can equivalently write

$$\lambda\boldsymbol{b}^* + \sum_{j=1}^{K_1}\alpha_j\boldsymbol{o}_j + \sum_{j=1}^{M}\mathrm{Sign}(\boldsymbol{o}_j^\top\boldsymbol{b}^*)\boldsymbol{o}_j + \sum_{j=1}^{K_2}\beta_j\boldsymbol{x}_j + \sum_{j=1}^{N}\mathrm{Sign}(\boldsymbol{x}_j^\top\boldsymbol{b}^*)\boldsymbol{x}_j = \boldsymbol{0} \quad (100)$$

or more compactly

$$\lambda \boldsymbol{b}^* + \boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}} + M\, \boldsymbol{o}_{\boldsymbol{b}^*} + \boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}} + N\, \boldsymbol{x}_{\hat{\boldsymbol{v}}^*} = \boldsymbol{0}, \tag{101}$$

where $\hat{\boldsymbol{v}}^*$ is the normalized projection of $\boldsymbol{b}^*$ onto $\mathcal{S}$ (nonzero since $\boldsymbol{b}^* \not\perp \mathcal{S}$ by hypothesis), and

$$\boldsymbol{o}_{\boldsymbol{b}^*} := \frac{1}{M}\sum_{j=1}^{M} \mathrm{Sign}(\boldsymbol{o}_j^\top \boldsymbol{b}^*)\boldsymbol{o}_j, \qquad \boldsymbol{x}_{\hat{\boldsymbol{v}}^*} := \frac{1}{N}\sum_{j=1}^{N} \mathrm{Sign}(\boldsymbol{x}_j^\top \hat{\boldsymbol{v}}^*)\boldsymbol{x}_j, \tag{102}$$

$$\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}} := \sum_{j=1}^{K_1} \alpha_j \boldsymbol{o}_j, \qquad \boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}} := \sum_{j=1}^{K_2} \beta_j \boldsymbol{x}_j. \tag{103}$$

From the definitions of $\epsilon_{\boldsymbol{\mathcal{O}},M}$ and $\epsilon_{\boldsymbol{\mathcal{X}},N}$ in (34) and (42) respectively, we have that

$$\boldsymbol{o}_{\boldsymbol{b}^*} = c_D\, \boldsymbol{b}^* + \boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}}, \;\; ||\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}}||_2 \leq \epsilon_{\boldsymbol{\mathcal{O}},M} \tag{104}$$

$$\boldsymbol{x}_{\hat{\boldsymbol{v}}^*} = c_d\, \hat{\boldsymbol{v}}^* + \boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}}, \;\; ||\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}}||_2 \leq \epsilon_{\boldsymbol{\mathcal{X}},N}, \tag{105}$$

and so (101) becomes

$$\lambda \boldsymbol{b}^* + \boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}} + M\, c_D\, \boldsymbol{b}^* + M\, \boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}} + \boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}} + N\, c_d\, \hat{\boldsymbol{v}}^* + N\, \boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}} = \boldsymbol{0}. \tag{106}$$

Since $\boldsymbol{b}^* \notin \mathcal{S}$, we have that $\boldsymbol{b}^*, \hat{\boldsymbol{v}}^*$ are linearly independent. Define the two-dimensional subspace $\mathcal{U} := \mathrm{Span}\,(\boldsymbol{b}^*, \hat{\boldsymbol{v}}^*)$ and project (106) onto $\mathcal{U}$ to get

$$\lambda \boldsymbol{b}^* + \pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}}) + M\, c_D\, \boldsymbol{b}^* + M\, \pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}}) + \pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}}) + N\, c_d\, \hat{\boldsymbol{v}}^* + N\, \pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}}) = \boldsymbol{0}. \tag{107}$$

Now, very vector $\boldsymbol{u}$ in the image of $\pi_{\mathcal{U}}$ can be written as a linear combination of $\boldsymbol{b}^*$ and $\hat{\boldsymbol{v}}^*$:

$$\boldsymbol{u} = [\boldsymbol{u}]_{\boldsymbol{b}^*}\, \boldsymbol{b}^* + [\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*}\, \hat{\boldsymbol{v}}^*, \;\; \text{with} \;\; [\boldsymbol{u}]_{\boldsymbol{b}^*}, [\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*} \in \mathbb{R}. \tag{108}$$

Taking inner product of $\boldsymbol{u}$ with $\boldsymbol{b}^*$ and $\hat{\boldsymbol{v}}^*$, we get respectively

$$\boldsymbol{u}^\top \boldsymbol{b}^* = [\boldsymbol{u}]_{\boldsymbol{b}^*} + [\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*} \cos(\phi^*) \tag{109}$$

$$\boldsymbol{u}^\top \hat{\boldsymbol{v}}^* = [\boldsymbol{u}]_{\boldsymbol{b}^*} \cos(\phi^*) + [\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*}, \tag{110}$$

where ϕ^* is the angle between $\boldsymbol{b}^*$ and $\hat{\boldsymbol{v}}^*$, i.e., the angle of $\boldsymbol{b}^*$ from $\mathcal{S}$. Solving with respect to $[\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*}$, we obtain

$$[\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*} = \frac{\boldsymbol{u}^\top \hat{\boldsymbol{v}}^* - \boldsymbol{u}^\top \boldsymbol{b}^* \cos(\phi^*)}{1 - \cos^2(\phi^*)}, \tag{111}$$

which in turn gives an upper bound on the magnitude of $[\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*}$:

$$|[\boldsymbol{u}]_{\hat{\boldsymbol{v}}^*}| \leq \frac{1 + \cos(\phi^*)}{1 - \cos^2(\phi^*)}\, \|\boldsymbol{u}\|_2\,. \tag{112}$$

Going back to (107) and writing each vector as a linear combination of $\boldsymbol{b}^*$ and $\hat{\boldsymbol{v}}^*$, we obtain

$$\begin{aligned}&\lambda \boldsymbol{b}^* + [\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}})]_{\boldsymbol{b}^*}\, \boldsymbol{b}^* + [\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*}\, \hat{\boldsymbol{v}}^* + M\, c_D\, \boldsymbol{b}^* + M\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}})]_{\boldsymbol{b}^*}\, \boldsymbol{b}^* + M\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*}\, \hat{\boldsymbol{v}}^* + \\ &[\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}})]_{\boldsymbol{b}^*}\, \boldsymbol{b}^* + [\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*}\, \hat{\boldsymbol{v}}^* + N\, c_d\, \hat{\boldsymbol{v}}^* + N\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}})]_{\boldsymbol{b}^*}\, \boldsymbol{b}^* + N\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*}\, \hat{\boldsymbol{v}}^* = \boldsymbol{0}.\end{aligned} \tag{113}$$

Since $\mathcal{U}$ is a two-dimensional space, there exists a vector $\hat{\boldsymbol{\zeta}} \in \mathcal{U}$ that is orthogonal to $\boldsymbol{b}^*$ but not orthogonal to $\hat{\boldsymbol{v}}^*$. Projecting the above equation onto the line spanned by $\hat{\boldsymbol{\zeta}}$, we obtain the one-dimensional equation

$$([\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*} + M\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*} + [\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*} + N\, c_d + N\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*}) \cdot \hat{\boldsymbol{\zeta}}^\top \hat{\boldsymbol{v}}^* = 0. \tag{114}$$

Since $\hat{\boldsymbol{\zeta}}$ is not orthogonal to $\hat{\boldsymbol{v}}^*$, the above equation implies that

$$[\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*} + M\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*} + [\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*} + N\, c_d + N\, [\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*} = 0, \tag{115}$$

which, in turn, implies that

$$N\, c_d \leq |[\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*}| + M\, |[\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}})]_{\hat{\boldsymbol{v}}^*}| + |[\pi_{\mathcal{U}}(\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*}| + N\, |[\pi_{\mathcal{U}}(\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}})]_{\hat{\boldsymbol{v}}^*}|\,. \tag{116}$$

Invoking the upper bound of (112) together with

$$\left\|\boldsymbol{\xi}_{\boldsymbol{\mathcal{O}}}\right\|_2 \leq \mathcal{R}_{\boldsymbol{\mathcal{O}},K_1},\; \left\|\boldsymbol{\xi}_{\boldsymbol{\mathcal{X}}}\right\|_2 \leq \mathcal{R}_{\boldsymbol{\mathcal{X}},K_2},\; \left\|\boldsymbol{\eta}_{\boldsymbol{\mathcal{O}}}\right\|_2 \leq \epsilon_{\boldsymbol{\mathcal{O}},M},\; \left\|\boldsymbol{\eta}_{\boldsymbol{\mathcal{X}}}\right\|_2 \leq \epsilon_{\boldsymbol{\mathcal{X}},N}, \tag{117}$$

and the definition of $\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}$ (Definition 10), we get

$$N\, c_d \leq \frac{1 + \cos(\phi^*)}{1 - \cos^2(\phi^*)}\, (\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}} + M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + N\, \epsilon_{\boldsymbol{\mathcal{X}},N})\,, \tag{118}$$

or equivalently

$$\begin{aligned}&N\, c_d\, \cos^2(\phi^*) + (\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}} + M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + N\, \epsilon_{\boldsymbol{\mathcal{X}},N})\, \cos(\phi^*) \\ &+ (\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}} + M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + N\, \epsilon_{\boldsymbol{\mathcal{X}},N} - N\, c_d) \geq 0.\end{aligned} \tag{119}$$

This is a quadratic polynomial in $\cos(\phi^*)$, whose constant term is negative by the second inequality of hypothesis (46), and thus has exactly one positive and one negative root. As consequence, this polynomial being non-negative together with the fact that $\cos(\phi^*) > 0$, implies that $\cos(\phi^*)$ must be greater than the positive root of the polynomial, i.e.,

$$\cos(\phi^*) \geq \frac{-\mathscr{Q} + \sqrt{\mathscr{Q}^2 + 4c_d(c_d - \mathscr{Q})}}{2c_d}, \;\; \mathscr{Q} := \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} + \epsilon_{\boldsymbol{\mathcal{O}},M}\frac{M}{N} + \epsilon_{\boldsymbol{\mathcal{X}},N}. \tag{120}$$

On the other hand, by Lemma 13 we have

$$M(c_D + \epsilon_{\boldsymbol{\mathcal{O}},M}) \geq \min_{\hat{\boldsymbol{b}} \perp \mathcal{S}} \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{b}}\right\|_1 \geq \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}^*\right\|_1 \geq M(c_D - \epsilon_{\boldsymbol{\mathcal{O}},M}) + N(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi^*), \tag{121}$$

which implies that

$$2M\epsilon_{\boldsymbol{\mathcal{O}},M} \geq N(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\frac{-\mathscr{Q} + \sqrt{\mathscr{Q}^2 + 4c_d(c_d - \mathscr{Q})}}{2c_d}. \tag{122}$$

This latter inequality is equivalent to the inequality

$$\begin{aligned}&2\epsilon_{\boldsymbol{\mathcal{O}},M}^2(3c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\left(\frac{M}{N}\right)^2 + \epsilon_{\boldsymbol{\mathcal{O}},M}(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\left(2\frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} + \epsilon_{\boldsymbol{\mathcal{X}},N} + c_d\right)\frac{M}{N} \\ &- (c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})^2\left(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N} - \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N}\right) \geq 0,\end{aligned} \tag{123}$$

whose left-hand-side we view as quadratic polynomial in M/N. By the first two inequalities of hypothesis (46), the second term of this polynomial is positive, while the constant term is negative, and so this inequality is equivalent to M/N being greater or equal than the unique positive root of that polynomial. But this contradicts the third inequality of hypothesis (46). Consequently, the initial hypothesis of the proof that $\boldsymbol{b}^* \not\perp \mathcal{S}$ can not be true, and the theorem is proved.

A Geometric View of the Proof of Theorem 11. Let us provide some geometric intuition that underlies the proof of Theorem 11. It is instructive to begin our discussion by considering the case $d = 1, D = 2$, i.e. the inlier space is simply a line and the ambient space is a 2-dimensional plane. Since all points have unit ℓ_2-norm, every column of $\boldsymbol{\mathcal{X}}$ will be of the form $\hat{\boldsymbol{x}}$ or $-\hat{\boldsymbol{x}}$ for a fixed vector $\hat{\boldsymbol{x}} \in \mathcal{S}^1$ that spans the inlier space $\mathcal{S}$. In this setting, let us examine a global solution $\boldsymbol{b}^*$ of the optimization problem (9). We will start by assuming that such a $\boldsymbol{b}^*$ is not orthogonal to $\mathcal{S}$, and intuitively arrive at the conclusion that this can not be the case as long as there are *sufficiently many* inliers.

We will argue on an intuitive level that if $\boldsymbol{b}^* \not\perp \mathcal{S}$, then the principal angle ϕ^* of $\boldsymbol{b}^*$ from $\mathcal{S}$ needs to be small; this is captured precisely by (120) in the proof of the theorem. To see this, suppose $\boldsymbol{b}^* \not\perp \mathcal{S}$; then $\boldsymbol{b}^*$ will be non-orthogonal to every inlier, and by Proposition14 orthogonal to $D - 1 = 1$ outlier, say $\boldsymbol{o}_1$. The optimality condition (97) specializes to

$$\alpha_1 \boldsymbol{o}_1 + \underbrace{\sum_{j=1}^{M} \text{Sign}(\boldsymbol{o}_j^\top \boldsymbol{b}^*)\boldsymbol{o}_j}_{M\boldsymbol{o}_{\boldsymbol{b}^*}} + \sum_{j=1}^{N} \text{Sign}(\boldsymbol{x}_j^\top \boldsymbol{b}^*)\boldsymbol{x}_j + \lambda \boldsymbol{b}^* = \boldsymbol{0}, \tag{124}$$

where $-1 \leq \alpha_1 \leq 1$. Notice that the third term is simply $N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$, and so

$$\alpha_1 \boldsymbol{o}_1 + M\ \boldsymbol{o}_{\boldsymbol{b}^*} + \lambda \boldsymbol{b}^* = -N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}. \tag{125}$$

Now, what (125) is saying is that the point $-N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$ must lie inside the set

$$\text{Conv}(\pm \boldsymbol{o}_1) + \{M\boldsymbol{o}_{\boldsymbol{b}^*}\} + \text{Span}(\boldsymbol{b}^*) = \{\alpha_1 \boldsymbol{o}_1 + M\boldsymbol{o}_{\boldsymbol{b}^*} + \lambda \boldsymbol{b}^* : |\alpha_1| \leq 1, \lambda \in \mathbb{R}\}, \tag{126}$$

where the $+$ operator on sets is the Minkowski sum. Notice that the set $\text{Conv}(\pm \boldsymbol{o}_1) + M\boldsymbol{o}_{\boldsymbol{b}^*}$ is the translation of the line segment (polytope) $\text{Conv}(\pm \boldsymbol{o}_1)$ by $M\boldsymbol{o}_{\boldsymbol{b}^*}$. Then (125) says that if we draw all affine lines that originate from every point of $\text{Conv}(\pm \boldsymbol{o}_1) + M\boldsymbol{o}_{\boldsymbol{b}^*}$ and have direction $\boldsymbol{b}^*$, then one of these lines must meet the point $-N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$. Let us illustrate this for the case where $M = N = 5$ and say it so happens that $\boldsymbol{b}^*$ has a rather large angle ϕ^* from $\mathcal{S}$, say $\phi^* = 45°$. Recall that $\boldsymbol{o}_{\boldsymbol{b}^*}$ is concentrated around $\boldsymbol{c}_D\, \boldsymbol{b}^*$ and for the case $D = 2$ we have $c_D = \frac{2}{\pi}$. As illustrated in Figure 1, because ϕ^* is large, the unbounded polytope $M\boldsymbol{o}_{\boldsymbol{b}^*} + \text{Conv}(\pm \boldsymbol{o}_1) + \text{Span}(\boldsymbol{b}^*)$ misses the point $-N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$ thus making the optimality equation (125) infeasible. This indicates that critical vectors $\boldsymbol{b}^* \not\perp \mathcal{S}$ having large angles from $\mathcal{S}$ are unlikely to exist.

On the other hand, critical points $\boldsymbol{b}^* \not\perp \mathcal{S}$ may exist, but their angle ϕ^* from $\mathcal{S}$ needs to be small, as illustrated in Figure 2. However, such critical points can not be global minimizers, because small angles from $\mathcal{S}$ yield large objective values; this is captured precisely by equation (121) in the proof of the theorem. Hence the only possibility that critical points $\boldsymbol{b}^* \not\perp \mathcal{S}$ that are also global minimizers do exist is that the number of inliers is significantly less than the number of outliers, i.e. $N << M$, as illustrated in Figure 3. The precise notion of how many inliers should exist with respect to outliers is captured by condition (46) of Theorem 11.

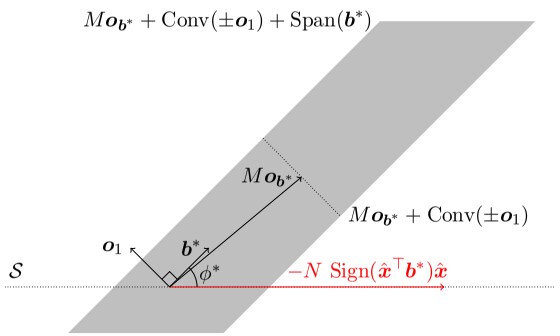


Figure 1: Geometry of the optimality condition (97) and (125) for the case $d = 1, D = 2, M = N = 5$. The polytope $M\boldsymbol{o}_{\boldsymbol{b}^*} + \text{Conv}(\pm \boldsymbol{o}_1) + \text{Span}(\boldsymbol{b}^*)$ misses the point $-N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$ and so the optimality condition can not be true for both $\boldsymbol{b}^* \not\perp \mathcal{S} = \text{Span}(\hat{\boldsymbol{x}})$ and ϕ^* large.

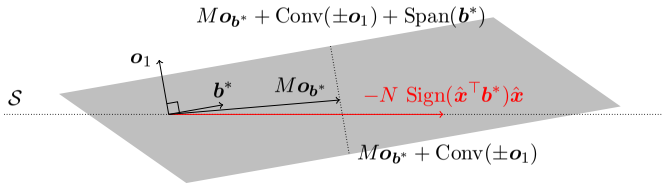


Figure 2: Geometry of the optimality condition (97) and (125) for the case $d = 1, D = 2, M = N = 5$. A critical $\boldsymbol{b}^* \not\perp \mathcal{S}$ exists, but its angle from $\mathcal{S}$ is small, so that the polytope $M\boldsymbol{o}_{\boldsymbol{b}^*} + \text{Conv}(\pm \boldsymbol{o}_1) + \text{Span}(\boldsymbol{b}^*)$ can contain the point $-N\ \text{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$. However, $\boldsymbol{b}^*$ can not be a global minimizer, since small angles from $\mathcal{S}$ yield large objective values.

We should note here that the picture for the general setting is analogous to what we described above, albeit harder to visualize: with reference to equation (100), the optimality condition says that every feasible point $\boldsymbol{b}^* \not\perp \mathcal{S}$ must have the following property: there exist $0 \leq K_2 \leq d - 1$ inliers $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{K_2}$ and $0 \leq K_1 \leq D - 1 - K_2$ outliers $\boldsymbol{o}_1, \ldots, \boldsymbol{o}_{K_1}$ to which $\boldsymbol{b}^*$ is orthogonal, and two points $\boldsymbol{\xi}_{\mathcal{O}} \in \text{Conv}(\pm \boldsymbol{o}_1 \pm \cdots \pm \boldsymbol{o}_{K_1}) + \boldsymbol{o}_{\boldsymbol{b}^*}$ and $\boldsymbol{\xi}_{\mathcal{X}} \in \text{Conv}(\pm \boldsymbol{x}_1 \pm \cdots \pm \boldsymbol{x}_{K_2}) + \boldsymbol{x}_{\boldsymbol{b}^*}$ that are joined by an affine line that is parallel to the line spanned by $\boldsymbol{b}^*$. In fact in our proof of Theorem 11 we reduced this general case to the case $d = 1, D = 2$ described above: this reduction is precisely taking place in equation (107), where we project the optimality equation onto the 2-dimensional subspace $\mathcal{U}$. The arguments that follow this projection consist of nothing more than a technical treatment of the intuition given above.

5.9 Proof of Theorem 12

First note that if (46) is true, then the expression of (51) always defines an angle between 0 and $\pi/2$. We start by establishing that $\hat{\boldsymbol{n}}_k$ does not lie in the inlier space $\mathcal{S}$. For $k = 0$ this is true by the

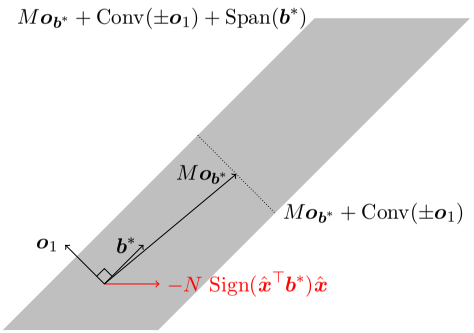


Figure 3: Geometry of the optimality condition (97) and (125) for the case $d = 1, D = 2, N << M$. Critical points $\boldsymbol{b}^* \not\perp \mathcal{S}$ do exist and moreover they can have large angle from $\mathcal{S}$. This is because N is small and so the polytope $M\boldsymbol{o}_{\boldsymbol{b}^*} + \mathrm{Conv}(\pm\boldsymbol{o}_1) + \mathrm{Span}(\boldsymbol{b}^*)$ contains the point $-N\ \mathrm{Sign}(\hat{\boldsymbol{x}}^\top \boldsymbol{b}^*)\hat{\boldsymbol{x}}$. Moreover, such critical points can be global minimizers. Condition (46) of Theorem 11 prevents such cases from occuring.

hypothesis (51). For the sake of contradiction suppose that $\hat{\boldsymbol{n}}_k \in \mathcal{S}$ for some $k > 0$. Note that

$$\left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_0\right\|_1 \geq \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{n}_1\right\|_1 \geq \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_1\right\|_1 \geq \cdots \geq \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_k\right\|_1. \tag{127}$$

Suppose first that $\hat{\boldsymbol{n}}_0 \perp \mathcal{S}$. Then (127) gives

$$\left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_0\right\|_1 \geq \left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_k\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top \hat{\boldsymbol{v}}_k\right\|_1, \tag{128}$$

where $\hat{\boldsymbol{v}}_k$ is the normalized projection of $\hat{\boldsymbol{n}}_k$ onto $\mathcal{S}$ (and since $\hat{\boldsymbol{n}}_k \in \mathcal{S}$, these two are equal). Using Lemma 13, we take an upper bound of the LHS and a lower bound of the RHS of (128), and obtain

$$M\, c_D + M\, \epsilon_{\boldsymbol{\mathcal{O}},M} \geq M\, c_D - M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + N\, c_d - N\, \epsilon_{\boldsymbol{\mathcal{X}},N}, \tag{129}$$

or equivalently

$$\frac{M}{N} \geq \frac{c_d - \epsilon_{\boldsymbol{\mathcal{X}},N}}{2\,\epsilon_{\boldsymbol{\mathcal{O}},M}}, \tag{130}$$

which contradicts the first inequality of (46). Consequently, $\hat{\boldsymbol{n}}_0 \not\perp \mathcal{S}$. Then (127) implies that

$$\left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_0\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top \hat{\boldsymbol{n}}_0\right\|_1 \geq \left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_k\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top \hat{\boldsymbol{n}}_k\right\|_1, \tag{131}$$

or equivalently

$$\left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_0\right\|_1 + \cos(\phi_0)\left\|\boldsymbol{\mathcal{X}}^\top \hat{\boldsymbol{v}}_0\right\|_1 \geq \left\|\boldsymbol{\mathcal{O}}^\top \hat{\boldsymbol{n}}_k\right\|_1 + \left\|\boldsymbol{\mathcal{X}}^\top \hat{\boldsymbol{v}}_k\right\|_1, \tag{132}$$

where $\hat{\boldsymbol{v}}_0$ is the normalized projection of $\hat{\boldsymbol{n}}_0$ onto $\mathcal{S}$. Once again, using Lemma 13 we obtain the following contradiction to (51):

$$M\, c_D + M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + (N\, c_d + N\, \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi_0) \geq M\, c_D - M\, \epsilon_{\boldsymbol{\mathcal{O}},M} + N\, c_d - N\, \epsilon_{\boldsymbol{\mathcal{X}},N}. \tag{133}$$

Now let us complete the proof of the theorem. We know by Proposition 16 that the sequence $\{\boldsymbol{n}_k\}$ converges to a critical point $\boldsymbol{n}_{k^*}$ of problem (9) in a finite number of steps k^*, and we have already shown that $\boldsymbol{n}_{k^*} \notin \mathcal{S}$. If $\boldsymbol{n}_{k^*}$ is not orthogonal to $\mathcal{S}$, an identical argument as in the proof of Theorem 11 (with $\boldsymbol{n}_{k^*}$ in place of $\boldsymbol{b}^*$) shows that the principal angle ϕ_{k^*} of $\boldsymbol{n}_{k^*}$ from $\mathcal{S}$ satisfies

$$\cos(\phi_{k^*}) \geq \frac{-\mathscr{Q} + \sqrt{\mathscr{Q}^2 + 4c_d(c_d - \mathscr{Q})}}{2c_d}, \quad \mathscr{Q} := \frac{\mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}}{N} + \epsilon_{\boldsymbol{\mathcal{O}},M}\frac{M}{N} + \epsilon_{\boldsymbol{\mathcal{X}},N}. \tag{134}$$

However, due to (127) and Lemma 13, we have that

$$M(c_D + \epsilon_{\boldsymbol{\mathcal{O}},M}) + N(c_d + \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi_0) \geq M(c_D - \epsilon_{\boldsymbol{\mathcal{O}},M}) + N(c_d - \epsilon_{\boldsymbol{\mathcal{X}},N})\cos(\phi_{k^*}), \tag{135}$$

which, after substituting the lower bound (134), contradicts (51). Thus $\boldsymbol{n}_{k^*} \perp \mathcal{S}$.

6. Dual Principal Component Pursuit Algorithms

We present algorithms based on the ideas discussed so far, for estimating the inlier linear subspace in the presence of outliers. Specifically, in §6.1 we describe the main algorithmic contribution of this paper, which is based on the implementation of the recursion (10) via linear programming. In §6.2 we propose an alternative way of computing dual principal components based on Iteratively Reweighted Least-Squares, which, as will be seen in §7 performs almost as well as recursion (6.1), yet it is significantly more efficient. Finally, in §6.3 we present a variation of the DPCP optimization problem (9) suitable for noisy data and propose a heuristic method for solving it.

6.1 DPCP via Linear Programming (DPCP-LP)

For the sake of an argument, suppose that there is no noise in the inliers, i.e., the inliers $\boldsymbol{\mathcal{X}}$ span a linear subspace $\mathcal{S}$ of dimension d. Then Theorem 12 suggests a mechanism for obtaining an element $\boldsymbol{b}_1$ of $\mathcal{S}^\perp$: run the recursion of linear programs (10) until the sequence $\hat{\boldsymbol{n}}_k$ converges and identify the limit point with $\boldsymbol{b}_1$. Due to computational constraints, in practice one usually terminates the recursion when the objective value $\left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_k\right\|_1$ converges within some small ε, or a maximal number $T_{\max}$ of iterations is reached, and obtains a normal vector $\boldsymbol{b}_1$. Having computed a vector $\boldsymbol{b}_1$, there are two possibilities: either $\mathcal{S}$ is a hyperplane of dimension $D-1$ or $\dim\mathcal{S} < D-1$. In the first case we can identify our subspace model with the hyperplane defined by the normal $\boldsymbol{b}_1$. If on the other hand $\dim\mathcal{S} < D-1$, we can proceed to find a second vector $\boldsymbol{b}_2 \perp \boldsymbol{b}_1$ that is approximately orthogonal to $\mathcal{S}$, and so on, until we have computed an orthogonal basis for the orthogonal complement of $\mathcal{S}$; this process naturally leads to Algorithm 1, in which c is an estimate for the codimension $D-d$ of the inlier subspace $\mathrm{Span}(\boldsymbol{\mathcal{X}})$.

Notice how the algorithm initializes $\boldsymbol{n}_0$: This is precisely the right singular vector of $\tilde{\boldsymbol{\mathcal{X}}}^\top$ that corresponds to the smallest singular value, after projection of $\tilde{\boldsymbol{\mathcal{X}}}$ onto $\mathrm{Span}(\boldsymbol{b}_1,\ldots,\boldsymbol{b}_{i-1})^\perp$. As it will be demonstrated in §7, this is a key choice, since it has the effect that the angle of $\boldsymbol{n}_0$ from the inlier subspace is typically large, a desirable property for the success of recursion (10) (see Theorem 12). We refer to Algorithm 1 as DPCP-LP, to emphasize that the optimization problem associated with each iteration of the recursion (10) is a linear program. In fact, at iteration k the optimization problem is

$$\min_{\boldsymbol{b}} \ \left\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\right\|_1 \ \text{s.t.}\ \boldsymbol{b}^\top \hat{\boldsymbol{n}}_{k-1} = 1, \tag{136}$$

Algorithm 1 Dual Principal Component Pursuit via Linear Programming

procedure DPCP-LP($\tilde{\boldsymbol{\mathcal{X}}}, c, \varepsilon, T_{\max}$)
  $\mathcal{B} \leftarrow \emptyset$;
  for $i = 1 : c$ **do**
    $k \leftarrow 0; \mathcal{J} \leftarrow 0; \Delta\mathcal{J} \leftarrow \infty$;
    $\hat{\boldsymbol{n}}_0 \leftarrow \boldsymbol{w} \in \operatorname{argmin}_{\|\boldsymbol{b}\|_2=1, \boldsymbol{b}\perp\mathcal{B}} \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_2$;
    while $k < T_{\max}$ and $\Delta\mathcal{J} > \varepsilon\mathcal{J}$ **do**
      $\mathcal{J} \leftarrow \|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_k\|_1$;
      $k \leftarrow k + 1$;
      $\boldsymbol{n}_k \leftarrow \boldsymbol{w} \in \operatorname{argmin}_{\boldsymbol{b}^\top \hat{\boldsymbol{n}}_{k-1}=1, \boldsymbol{b}\perp\mathcal{B}} \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_1$;
      $\hat{\boldsymbol{n}}_k \leftarrow \boldsymbol{n}_k / \|\boldsymbol{n}_k\|_2$;
      $\Delta\mathcal{J} \leftarrow \left(\mathcal{J} - \|\tilde{\boldsymbol{\mathcal{X}}}^\top \hat{\boldsymbol{n}}_k\|_1\right)$;
    end while
    $\boldsymbol{b}_i \leftarrow \hat{\boldsymbol{n}}_k$;
    $\mathcal{B} \leftarrow \mathcal{B} \cup \{\boldsymbol{b}_i\}$;
  end for
  return $\mathcal{B}$;
end procedure

which can equivalently be written as a standard linear program,

$$\min_{\boldsymbol{b}, \boldsymbol{u}^+, \boldsymbol{u}^-} \quad \begin{bmatrix} \mathbf{1}_{1\times N} & \mathbf{1}_{1\times N} \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^+ \\ \boldsymbol{u}^- \end{bmatrix} \tag{137}$$

$$\text{s.t.} \quad \begin{bmatrix} \boldsymbol{I}_N & -\boldsymbol{I}_N & -\tilde{\boldsymbol{\mathcal{X}}}^\top \\ \mathbf{0}_{1\times N} & \mathbf{0}_{1\times N} & \hat{\boldsymbol{n}}_k^\top \end{bmatrix} \begin{bmatrix} \boldsymbol{u}^+ \\ \boldsymbol{u}^- \\ \boldsymbol{b} \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{N\times 1} \\ 1 \end{bmatrix}, \ \boldsymbol{u}^+, \boldsymbol{u}^- \geq 0, \tag{138}$$

and can be solved efficiently with an optimized general purpose linear programming solver, such as Gurobi (Gurobi Optimization, 2015).

6.2 DPCP via Iteratively Reweighted Least-Squares (DPCP-IRLS)

Even though DPCP-LP (Algorithm 1) comes with theoretical guarantees as per Theorem 12, and moreover will be shown to have a rather remarkable performance (at least for synthetic data, see Fig. 6), it has the weakness that the linear programs (which are non-sparse) may become inefficient to solve in high dimensions and for a large number of data points. Moreover, even though DPCP-LP is theoretically applicable regardless of the subspace relative dimension d/D, its running time increases with the subspace codimension $c = D - d$, since the c basis elements of $\mathcal{S}^\perp$ are computed sequentially. This motivates us to generalize the DPCP problem (9) to an optimization problem that targets the entire orthogonal basis of $\mathcal{S}^\perp$:

$$\min_{\boldsymbol{B}\in\mathbb{R}^{D\times c}} \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}\|_{1,2} \ \text{s.t.} \ \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}_c. \tag{139}$$

Algorithm 2 Dual Principal Component Pursuit via Iteratively Reweighted Least Squares

procedure DPCP-IRLS($\tilde{\boldsymbol{\mathcal{X}}}, c, \varepsilon, T_{\max}, \delta$)
  $k \leftarrow 0; \mathcal{J} \leftarrow 0; \Delta\mathcal{J} \leftarrow \infty$;
  $\boldsymbol{B}_0 \leftarrow \boldsymbol{W} \in \operatorname{argmin}_{\boldsymbol{B}\in\mathbb{R}^{D\times c}, \boldsymbol{B}^\top\boldsymbol{B}=\boldsymbol{I}_c} \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}\|_F$;
  while $k < T_{\max}$ and $\Delta\mathcal{J} > \varepsilon\mathcal{J}$ **do**
    $\mathcal{J} \leftarrow \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}_k\|_{1,2}$;
    $k \leftarrow k + 1$;
    $\boldsymbol{B}_k \leftarrow \operatorname{argmin}_{\boldsymbol{B}\in\mathbb{R}^{D\times c}, \boldsymbol{B}^\top\boldsymbol{B}=\boldsymbol{I}_c} \sum_{\tilde{\boldsymbol{x}}\in\tilde{\boldsymbol{\mathcal{X}}}} \|\boldsymbol{B}^\top\tilde{\boldsymbol{x}}\|_2^2 / \max\{\delta, \|\boldsymbol{B}_{k-1}^\top\tilde{\boldsymbol{x}}\|_2\}$;
    $\Delta\mathcal{J} \leftarrow \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}_{k-1}\|_{1,2} - \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}_k\|_{1,2}$;
  end while
  return $\boldsymbol{B}_k$;
end procedure

Notice that in (139), the $\ell_{1,2}$ matrix norm $\|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}\|_{1,2}$ of $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}$ is defined as the sum of the Euclidean norms of the rows of $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}$, and as such, favors a solution $\boldsymbol{B}$ that results in a matrix $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{B}$ that is row-wise sparse (notice that for $c = 1$ (139) reduces precisely to the DPCP problem (9)). In fact, Lerman et al. (2015) consider exactly the same problem (139), and proceed to relax it to a semi-definite convex program, which they solve via an *Iteratively Reweighted Least-Squares (IRLS)* scheme (Candès et al., 2008; Daubechies et al., 2010; Chartrand and Yin, 2008); while similar IRLS schemes appear in Zhang and Lerman (2014) and Lerman and Maunu (2017). Instead, we propose to solve (139) directly via IRLS (and not a convex relaxation of it as Lerman et al. (2015)): Given a $D \times c$ orthonormal matrix $\boldsymbol{B}_{k-1}$, we define for each point $\tilde{\boldsymbol{x}}_j$ a weight

$$w_{j,k} := \frac{1}{\max\{\delta, \|\boldsymbol{B}_{k-1}^\top \tilde{\boldsymbol{x}}_j\|_2\}}, \tag{140}$$

where $\delta > 0$ is a small constant that prevents division by zero. Then we obtain $\boldsymbol{B}_k$ as the solution to the quadratic problem

$$\min_{\boldsymbol{B}\in\mathbb{R}^{D\times c}} \sum_{j=1}^{L} w_{j,k} \|\boldsymbol{B}^\top \tilde{\boldsymbol{x}}_j\|_2^2 \ \text{s.t.} \ \boldsymbol{B}^\top\boldsymbol{B} = \boldsymbol{I}_c, \tag{141}$$

which is readily seen to be the c right singular vectors corresponding to the c smallest singular values of the weighted data matrix $\boldsymbol{W}_k \tilde{\boldsymbol{\mathcal{X}}}^\top$, where $\boldsymbol{W}_k$ is a diagonal matrix with $\sqrt{w_{j,k}}$ at position (j, j). We refer to the resulting Algorithm 2 as DPCP-IRLS; a study of its theoretical properties is deferred to future work.

6.3 Denoised DPCP (DPCP-d)

Clearly, problem (9) (and (139)) is tailored for noise-free inliers, since, when the inliers $\boldsymbol{\mathcal{X}}$ are contaminated by noise, the vector $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}$ is no longer sparse, even if $\boldsymbol{b}$ is a true normal to the inlier subspace. As a consequence, it is natural to propose the following *DPCP-denoised* (DPCP-d)

Algorithm 3 Denoised Dual Principal Component Pursuit

procedure DPCP-d($\tilde{\boldsymbol{\mathcal{X}}}, \varepsilon, T_{\max}, \delta, \tau$)
  Compute a Cholesky factorization $\boldsymbol{L}\boldsymbol{L}^\top = \tilde{\boldsymbol{\mathcal{X}}}\tilde{\boldsymbol{\mathcal{X}}}^\top + \delta \boldsymbol{I}_D$;
  $k \leftarrow 0$; $\boldsymbol{y}_0 \leftarrow \boldsymbol{0}$; $\mathcal{J} \leftarrow 0$; $\Delta\mathcal{J} \leftarrow \infty$;
  $\boldsymbol{b}_0 \leftarrow \operatorname{argmin}_{\boldsymbol{b}\in\mathbb{R}^D:\, \|\boldsymbol{b}\|_2=1} \ \|\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_2$;
  while $k < T_{\max}$ and $\Delta\mathcal{J} > \varepsilon\mathcal{J}$ **do**
    $\mathcal{J} \leftarrow \tau\|\boldsymbol{y}_k\|_1 + \frac{1}{2}\|\boldsymbol{y}_k - \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}_k\|_2^2$
    $\boldsymbol{y}_{k+1} \leftarrow \mathcal{S}_\tau\left(\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}_k\right)$;
    $\boldsymbol{b}_{k+1} \leftarrow$ solution of $\boldsymbol{L}\boldsymbol{L}^\top \boldsymbol{\xi} = \tilde{\boldsymbol{\mathcal{X}}}\boldsymbol{y}_{k+1}$ by backward/forward propagation;
    $k \leftarrow k+1$;
    $\boldsymbol{b}_k \leftarrow \boldsymbol{b}_k / \|\boldsymbol{b}_k\|_2$;
    $\Delta\mathcal{J} \leftarrow \mathcal{J} - \left(\tau\|\boldsymbol{y}_k\|_1 + \frac{1}{2}\|\boldsymbol{y}_k - \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}_k\|_2^2\right)$;
  end while
  return $(\boldsymbol{y}_k, \boldsymbol{b}_k)$;
end procedure

problem

$$\min_{\boldsymbol{b},\boldsymbol{y}:\, \|\boldsymbol{b}\|_2=1} \left[\tau\|\boldsymbol{y}\|_1 + \frac{1}{2}\|\boldsymbol{y} - \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_2^2\right], \tag{142}$$

where now the vector variable $\boldsymbol{y} \in \mathbb{R}^{N+M}$ is to be interpreted as a denoised vesion of the vector $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}$. Interestingly, both problems (9) and (142) appear in Qu et al. (2014), in the quite different context of dictionary learning, where the authors propose to solve (142) via alternating minimization, in order to obtain an approximate solution to (9). Given $\boldsymbol{b}$, the optimal $\boldsymbol{y}$ is given by $\mathcal{S}_\tau(\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b})$, where $\mathcal{S}_\tau$ is the soft-thresholding operator applied element-wise on the vector $\tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}$. Given $\boldsymbol{y}$ the optimal $\boldsymbol{b}$ is a solution to the quadratically constrained least-squares problem

$$\min_{\boldsymbol{b}\in\mathbb{R}^D} \ \|\boldsymbol{y} - \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_2^2 \ \text{ s.t. } \ \|\boldsymbol{b}\|_2 = 1. \tag{143}$$

In the context of Qu et al. (2014), the coefficient matrix of the least-squares problem ($\tilde{\boldsymbol{\mathcal{X}}}^\top$ in our notation) has orthonormal columns. As a consequence, the solution to (143) is obtained in closed form by projecting the solution of the unconstrained least-squares problem $\min_{\boldsymbol{b}\in\mathbb{R}^D} \ \|\boldsymbol{y} - \tilde{\boldsymbol{\mathcal{X}}}^\top \boldsymbol{b}\|_2$ onto the unit sphere. However, in our context the assumption that $\tilde{\boldsymbol{\mathcal{X}}}^\top$ has orthonormal columns is in principle violated, so that the optimal $\boldsymbol{b}$ is no longer available in closed form. Even though using Lagrange multipliers one ends up with a polynomial equation for the Lagrange multiplier, it is known that computing the optimal value of the multiplier is a numerically challenging problem (Elden, 2002; Golub and Von Matt, 1991; Gander, 1980). For this reason we leave exact approaches for solving (143) to future investigations, and we instead propose to obtain a suboptimal $\boldsymbol{b}$ as Qu et al. (2014) do, i.e., by projecting onto the unit sphere the solution of the unconstrained least-squares problem. The resulting Algorithm 3 is very efficient, since the least-squares problems that

appear in the various iterations have the same coefficient matrix $\tilde{\boldsymbol{\mathcal{X}}}\tilde{\boldsymbol{\mathcal{X}}}^\top$, a factorization of which can be precomputed.[9] Moreover, Algorithm 3 can trivially be extended to compute multiple normal vectors, just as in Algorithm 1.

7. Experiments

In this section we evaluate the proposed algorithms experimentally. In §7.1 we investigate numerically the theoretical regime of success of recursion (10) predicted by Theorems 11 and 12. We also show that even when these sufficient conditions are violated, (10) can still converge to a normal vector to the subspace if initialized properly. Finally, in §7.2 we compare DPCP variants with state-of-the-art robust PCA algorithms for the purpose of outlier detection using synthetic data, and similarly in §7.3 using real images.

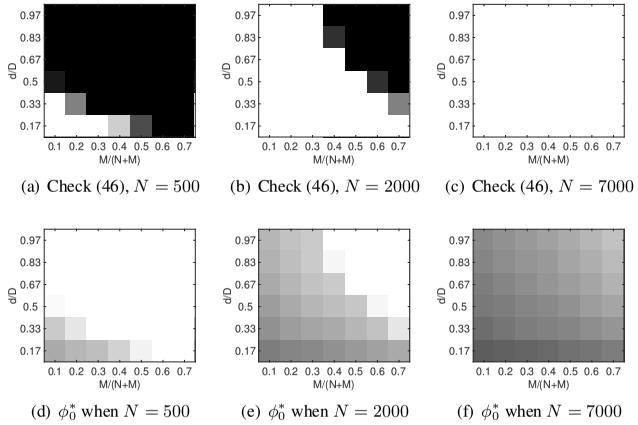
7.1 Numerical evaluation of the theoretical conditions of Theorems 11 and 12

We begin with a numerical evaluation of the theoretical condition (46) of Theorem 11, under which every global minimizer of the DPCP problem (9) is orthogonal to the inlier subspace $\mathcal{S}$. We also evaluate the initial minimal angle ϕ_0^* from $\mathcal{S}$ given in (51) of Theorem 12, which together with (46) guarantee the convergence of the linear programming recursion (10) to an element of $\mathcal{S}^\perp$. As explained in the discussion of Theorems 11 and 12, for any fixed outlier ratio, condition (46) will eventually be satisfied and also the angle ϕ_0^* will become arbitrarily small regardless of the subspace relative dimension d/D, provided that N is sufficiently large and that both inliers and outliers are uniformly distributed. Hence, we check whether (46) is true and also plot ϕ_0^* as we vary N for uniformly distributed inliers and outliers. Towards that end, we fix the ambient dimension as $D = 30$ and randomly sample a subspace $\mathcal{S}$ of varying dimension $d = [5 : 5 : 25\ 29]$ so that the relative subspace dimension d/D varies as $[5/30 : 5/30 : 25/30\ 29/30]$. We sample N inliers uniformly at random from $\mathcal{S} \cap \mathbb{S}^{D-1}$ for different values $N = 500, 2000, 7000$. For each value of N we also sample M outliers uniformly at random from $\mathbb{S}^{D-1}$ so that the percentage of outliers varies as $R := M/(N+M) = [0.1 : 0.1 : 0.7]$. For each dataset instance as above, we estimate the parameters $\epsilon_{\boldsymbol{\mathcal{X}},N}, \epsilon_{\boldsymbol{\mathcal{O}},M}, \mathcal{R}_{\boldsymbol{\mathcal{O}},\boldsymbol{\mathcal{X}}}$ appearing in (46) and (51) by Monte-Carlo simulation.

The top row of Fig. 4 shows whether condition (46) is true (white) or not (black) as we vary N. Notice that for $N = 500$, Fig. 4(a) shows a poor success regime. However, as we increase N to 2000, Fig. 4(b) shows that the success regime improves dramatically. Finally, as expected from our earlier theoretical arguments, for sufficiently large N, in particular for $N = 7000$, Fig. 4(c) shows that the sufficient condition (46) is satisfied regardless of outlier ratio or subspace relative dimension. Similarly, notice how the angle ϕ_0^*, plotted in the bottom row of Fig. 4 (black for 0° white for 90°), uniformly decreases as we increase N across all outlier ratios and relative dimensions.

Next, we show that the recursion (10), if initialized properly, is in fact able to converge in just a few iterations to a vector normal to the inlier subspace, even when the sufficient conditions of Theorem 12 are not satisfied. Towards that end, we maintain the same experimental setting as above using $N = 500$ and run (10) with a maximal number of iterations set to $T_{\max} = 10$ and a convergence accuracy set to 10^{-3}. Fig. 5(a) is a replicate of Fig. 4(a) and serves as a reminder that $N = 500$ results in a limited success regime as predicted by the theory; in particular the sufficient

9. The parameter δ in Algorithm 3 is a small positive number, typically 10^{-6}, which helps avoiding solving ill-conditioned linear systems.



(a) Check (46), $N = 500$ (b) Check (46), $N = 2000$ (c) Check (46), $N = 7000$

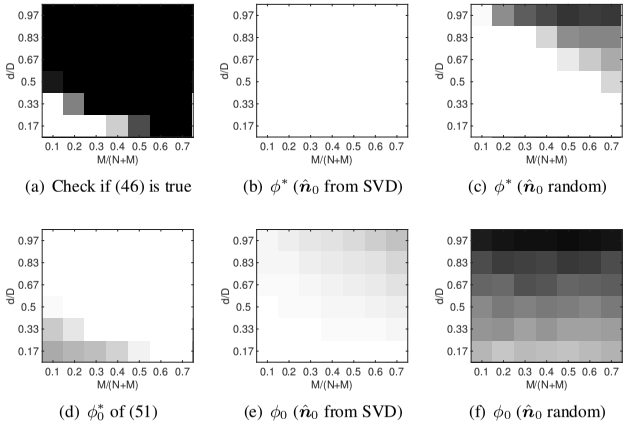
(d) ϕ_0^* when $N = 500$ (e) ϕ_0^* when $N = 2000$ (f) ϕ_0^* when $N = 7000$

Figure 4: Figs. 4(a)-4(c) check whether the condition (46) is satisfied (white) or not (black) for a fixed number N of inliers while varying the outlier ratio $M/(N+M)$ and the subspace relative dimension d/D. Figs. 4(d)-4(f) plot the minimum initial angle ϕ_0^* needed for convergence of the recursion of linear programs (10) as per Theorem 12 (0° corresponds to black and 90° corresponds to white). Results are averaged over 10 independent trials.

condition (46) is satisfied only for a small outlier ratio or small subspace relative dimensions. Even so, Fig. 5(b) shows that, when the recursion (10) is initialized using $\hat{\boldsymbol{n}}_0$ as the left singular vector of $\tilde{\boldsymbol{\mathcal{X}}}$ corresponding to the smallest singular value, then (10) converges in at most 10 iterations to a vector $\hat{\boldsymbol{n}}^*$ whose angle ϕ^* from the subspace is precisely 90°. This suggests that the sufficient condition (46) is much stronger than necessary, leaving room for future theoretical improvements. On the other hand, Fig. 5(c) shows that when $\hat{\boldsymbol{n}}_0$ is initialized uniformly at random, the recursion (10) does not always converge to a normal vector, particularly for high outlier ratios and relative dimensions. This reveals that initializing (10) from the SVD of the data is indeed a good strategy, which is further supported by Fig. 5(e), which plots the angle of the initialization from the subspace (contrast this to Fig. 5(f), which shows the angle of a random initialization from the subspace).

7.2 Comparative analysis using synthetic data

In this section we use the same synthetic experimental set-up as in §7.1 (with $N = 500$) to demonstrate the behavior of several methods relative to each other under uniform conditions, in the context of outlier rejection in single subspace learning. In particular, we test DPCP-LP (Algorithm 1), DPCP-IRLS (Algorithm 2), DPCP-d (Algorithm 3), RANSAC (Fischler and Bolles, 1981), SE-RPCA (Soltanolkotabi and Candès, 2012), ℓ_{21}-RPCA (Xu et al., 2012), the IRLS version of REAPER (Lerman et al., 2015), as well as Coherence Pursuit (CoP) (Rahmani and Atia, 2017); see §2 for details on these last five existing methods.

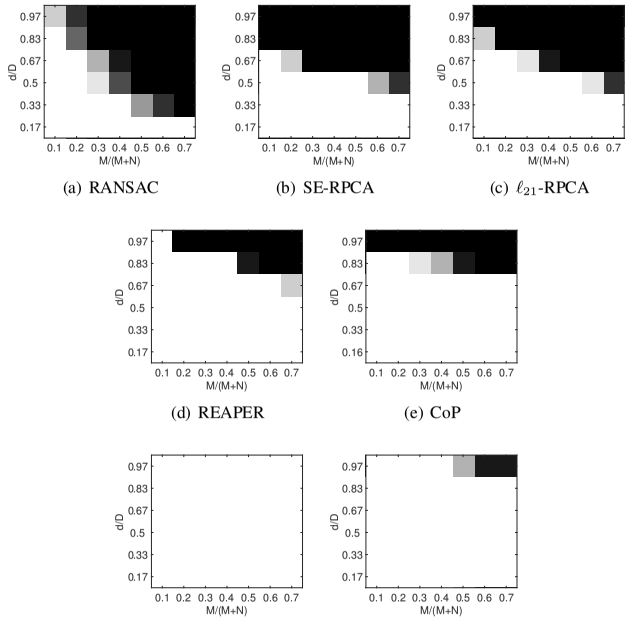


(a) Check if (46) is true (b) ϕ^* ($\hat{\boldsymbol{n}}_0$ from SVD) (c) ϕ^* ($\hat{\boldsymbol{n}}_0$ random)

(d) ϕ_0^* of (51) (e) ϕ_0 ($\hat{\boldsymbol{n}}_0$ from SVD) (f) ϕ_0 ($\hat{\boldsymbol{n}}_0$ random)

Figure 5: Convergence of recursion (10) in a regime of limited theoretical guarantees ($N = 500$) as concluded from Fig. 4. The number of inliers is fixed at $N = 500$. Fig. 5(a) plots whether the sufficient condition (46) is satisfied (white) or not (black) for varying outlier ratios $M/(N+M)$ and relative dimensions d/D. Figs. 5(b)-5(c) plot the angle ϕ^* (0° corresponds to black and 90° corresponds to white) from the inlier subspace of the vector $\hat{\boldsymbol{n}}^*$ that recursion (10) converges to, when $\hat{\boldsymbol{n}}_0$ is initialized from the SVD of the data or uniformly at random, respectively. Fig. 5(d) plots the minimum angle ϕ_0^* needed for the convergence of (10) to a normal vector in the inlier subspace as per Theorem 12, while Figs. 5(e)-5(f) plot the angle ϕ_0 of $\hat{\boldsymbol{n}}_0$ from the subspace, when it is initialized from the SVD of the data or uniformly at random, respectively. The results are averages over 10 independent trials.

For the methods that require an estimate of the subspace dimension d, such as RANSAC, REAPER, CoP, and all DPCP variants, we provide as input the true subspace dimension. The convergence accuracy of all methods is set to 10^{-3}. For REAPER we set the regularization parameter as $\delta = 10^{-6}$ and the maximum number of iterations equal to 100. For DPCP-d we set $\tau = 1/\sqrt{N+M}$ as suggested in Qu et al. (2014) and the maximum number of iterations to 1000. For RANSAC we set its thresholding parameter to 10^{-3}, and for fairness, we do not let it terminate earlier than the running time of DPCP-LP, unless the theoretically required number of iterations for a success probability 0.99 is reached (here we are using the ground truth outlier ratio). Both SE-RPCA and ℓ_{21}-RPCA are implemented with ADMM, with augmented Lagrange parameters 1000 and 100 respectively. For ℓ_{21}-RPCA λ is set to $3/(7\sqrt{M})$, as suggested in Xu et al. (2012). DPCP variants are initialized via the SVD of the data as in Algorithm 1. CoP is implemented using the code provided by its authors, and selects $3d$ points upon classic PCA gives the subspace estimate. Finally, the linear programs in DPCP-LP are solved via the generic LP solver Gurobi (Gurobi Optimization, 2015), while the maximum number of iterations for DPCP-LP is set to $T_{\max} = 10$.



(a) RANSAC (b) SE-RPCA (c) ℓ_{21}-RPCA

(d) REAPER (e) CoP

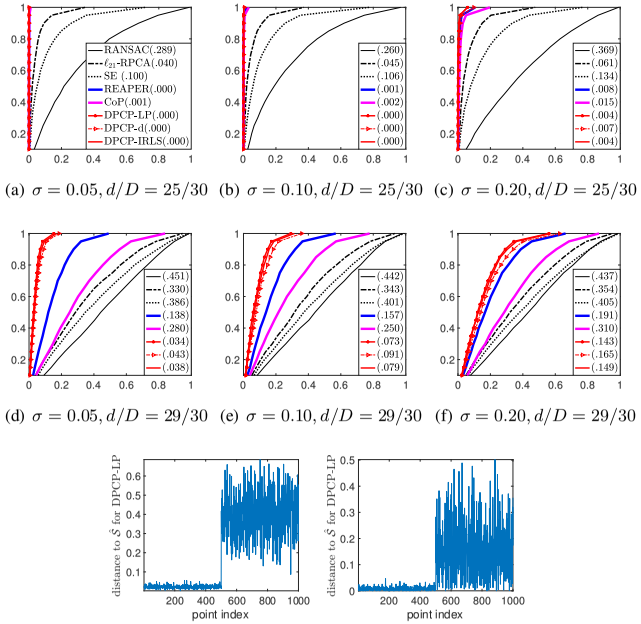
(f) DPCP-LP (g) DPCP-IRLS

Figure 6: Outlier/Inlier separation in the absence of noise over 10 independent trials. The horizontal axis is the outlier ratio defined as $M/(N+M)$, where M is the number of outliers and N is the number of inliers. The vertical axis is the relative inlier subspace dimension d/D; the dimension of the ambient space is $D = 30$. Success (white) is declared by the existence of a threshold that, when applied to the output of each method, perfectly separates inliers from outliers.

Absence of Noise We first investigate the potential of each of the above methods to perfectly distinguish outliers from inliers in the absence of noise.[10] Note that each method returns a *signal* $\boldsymbol{\alpha} \in \mathbb{R}_+^{N+M}$, which can be thresholded for the purpose of declaring outliers and inliers. For SE-RPCA, $\boldsymbol{\alpha}$ is the ℓ_1-norm of the columns of the coefficient matrix $\boldsymbol{C}$, while for ℓ_{21}-RPCA $\boldsymbol{\alpha}$ is the ℓ_2-norm of the columns of $\boldsymbol{E}$. Since RANSAC, REAPER, CoP, and DPCP variants directly return subspace models, for these methods $\boldsymbol{\alpha}$ is simply the distances of all points to the estimated subspace.

In Fig. 6 we depict success (white) versus failure (black), where success is interpreted as the existence of a threshold on $\boldsymbol{\alpha}$ that perfectly separates outliers and inliers. First observe that, as expected, RANSAC succeeds when there are very few outliers (10%) regardless of the inlier relative dimension d/D, since in such a case the probability of sampling outlier-free points is high. Similarly, RANSAC succeeds when d/D is small regardless of the outlier ratio, since in that case one needs only sample d points, and for a sufficient budget (say, the running time of DPCP-LP), the probability of one of these samples being outlier-free is again high. Moving on, and again as expected, both SE-RPCA and ℓ_{21}-RPCA succeed only for low to medium relative dimensions, since both methods are meant to exploit the low-rank structure of the inlier data, and thus fail when such a structure does not exist. Remarkably, even though CoP is a low-rank method in spirit, it performs surprisingly better than its low-rank alternatives SE-RPCA and ℓ_{21}-RPCA, giving perfect inlier/outlier separation regardless of the outlier ratio for relative dimensions $d/D \leq 2/3$. Further improvement is achieved by REAPER, which succeeds for as many as 40% outliers and as high a

10. We do not include DPCP-d for this experiment, since it only approximately solves the DPCP optimization problem, and hence it can not be expected to perfectly separate inliers from outliers, even when there is no noise (we have confirmed this experimentally).



(a) $\sigma = 0.05, d/D = 25/30$ (b) $\sigma = 0.10, d/D = 25/30$ (c) $\sigma = 0.20, d/D = 25/30$

(d) $\sigma = 0.05, d/D = 29/30$ (e) $\sigma = 0.10, d/D = 29/30$ (f) $\sigma = 0.20, d/D = 29/30$

(g) $\sigma = 0.05, d/D = 25/30$ (h) $\sigma = 0.05, d/D = 29/30$

Figure 7: Figs. 7(a)-7(f) show ROC curves for varying noise standard deviation $\sigma = 0.05, 0.1, 0.2$, and subspace relative dimension $d/D = 25/30, 29/30$ (number of inliers is $N = 500$ and outlier ratio is $M/(N+M) = 0.5$). The horizontal axis is False Positives ratio and the vertical axis is True Positives ratio. The number associated with each curve is the area above the curve; smaller numbers reflect more accurate performance. Figs. 7(g)-7(h) show the distance of the noisy points to the subspace estimated by DPCP-LP for noise $\sigma = 0.05$ for both relative dimensions under consideration.

Table 1: Mean running time of each method in seconds over 10 independent trials for the experimental setting of §7.2. We report only the extreme regimes corresponding to $d/D = 5/30, 29/30$ and $M/(N+M) = 0.1, 0.7$. The experiment is run in MATLAB on a standard Macbook-Pro with a dual core 2.5GHz Processor and a total of 4GB Cache memory.

$d/D : M/(N+M)$	$5/30 : 0.1$	$29/30 : 0.1$	$5/30 : 0.7$	$29/30 : 0.7$
RANSAC	0.097	0.410	23.31	2.83
SE-RPCA	4.485	4.519	58.94	79.07
ℓ_{21}-RPCA	0.048	0.014	0.185	0.180
REAPER	0.050	0.058	0.153	0.042
CoP	0.014	0.014	0.062	0.061
DPCP-LP	16.87	0.407	95.35	2.822
DPCP-IRLS	0.046	0.038	0.121	0.415

relative dimension as $25/30 \approx 0.83$. Yet, REAPER fails in the challenging case $d/D = 29/30$ as soon as there are more than 20% outliers. Remarkably, DPCP-LP allows for perfect outlier rejection across all outlier ratios and all relative dimensions, thus clearly improving the state-of-art in the high relative dimension regime. Moreover, DPCP-IRLS does almost as well as DPCP-LP thus being the second best method, except that it only fails in the hardest of regimes, i.e., for relative dimension $29/30 \approx 0.97$ and for more than 50% outliers.

Finally, it is important to comment on the running time of the methods. As Table 1 shows, DPCP-LP is admittedly the slowest among the methods, particularly for low relative dimensions, since in that case many dual principal components need to be computed. Indeed, for 10% outliers and $d/D = 5/30$ DPCP-LP computes $D - d = 25$ dual components and thus it takes about 17 seconds, as opposed to 0.4 seconds for the same amount of outliers but $d/D = 29/30$, since a single dual component is computed in this latter case. Similarly, for 70% outliers DPCP-LP takes about 95 seconds when $d/D = 5/30$ as opposed to about 3 seconds for $d/D = 29/30$. On the other hand, DPCP-IRLS is one order of magnitude faster than DPCP-LP and comparable to ℓ_{21}-RPCA and REAPER, which overall are the second fastest methods, with CoP being the fastest of all.

Presence of Noise Next, we fix $D = 30, N = 500, M/(N+M) = 0.5$, and investigate the performance of the methods, adding DPCP-d to the mix, in the presence of varying levels of noise for two cases of high relative dimension, i.e., $d/D = 25/30$ and $d/D = 29/30$. The inliers are corrupted by additive white Gaussian noise of zero mean and standard deviation $\sigma = 0.05, 0.1, 0.2$, with support in the orthogonal complement of the inlier subspace. The parameters of all methods are the same as earlier except for DPCP-d we set $\tau = \max\left\{\sigma, 1/\sqrt{N+M}\right\}$, while for RANSAC we set its threshold parameter equal to σ.

We evaluate the performance of each method by its corresponding ROC curve. Each point of an ROC curve corresponds to a certain value of a threshold, with the vertical coordinate of the point giving the percentage of inliers being correctly identified as inliers (True Positives), and the horizontal coordinate giving the number of outliers erroneously identified as inliers (False Positives). As a consequence, an ideal ROC curve should be concentrated to the top left of the plot, i.e., the area above the curve should be zero. The ROC curves[11] as well as the area above each curve are shown in Fig. 7. As expected, the low-rank methods RANSAC, SE-RPCA and ℓ_{21}-RPCA perform poorly for either relative dimension with performance being close to that of a random guess (inlier vs. outlier) for relative dimension $29/30$. On the other hand REAPER, CoP, DPCP-LP, DPCP-IRLS and DPCP-d perform almost perfectly well for $d/D = 25/30$, while REAPER starts failing for very high relative dimension $29/30$, even for as low noise standard devision as $\sigma = 0.05$ (of course this is to be expected because we already know from Fig. 6 that REAPER fails at this regime even in the absence of noise), and CoP fails completely. In contrast, the DPCP variants remain robust to noise in this challenging regime for as high noise as $\sigma = 0.1$. What is remarkable, is that both DPCP-LP and DPCP-IRLS, which are designed for noiseless data, are surprisingly robust to noise, and slightly outperform DPCP-d, the latter meant to handle noisy data. We attribute this fact to the suboptimal approach we followed in solving the DPCP-d problem, as well as to the lack of a suitable mechanism for optimally tuning its thresholding parameter.

11. We note that the vertical axis of all ROC curves in this paper starts from a ratio of 0.1 True Positives.

7.3 Comparative analysis using real data: Three-view geometry

In this section we perform an experimental evaluation of the proposed methods in the context of the three-view problem in computer vision using real data. In that setting one is given three images of the same static scene taken from different views, and the goal is to estimate the relative view poses, i.e., the rotations and translations that relate, say, view 1 to view 2 and 3 (e.g., see Figs 8(a)-8(b)). This task is of fundamental importance in many computer vision applications, such as 3D reconstruction, where a 3D model of a real-world scene is constructed from 2D images of the scene.

The trifocal tensor. The three images of the static scene may have been taken from three different cameras, or from a single moving camera. Regardless, the underlying three-view geometry is characterized by the constraints satisfied by any points lying in views $1, 2$ and 3 respectively, that correspond to the same 3D point (e.g., see Fig. 8(b)). To describe the nature of these constraints, we fix a coordinate system (x, y, z) for the 3D space, and identify view 1 with the plane $z = 1$ and its optical center with the origin $\mathbf{0}$ of the coordinate system. We refer to this view as *canonical view* $\mathcal{V}$. Then the projection $\mathcal{X}$ of a 3D point $\Xi = (\xi_1, \xi_2, \xi_3)^\top$ onto $\mathcal{V}$ is the intersection point of the plane $z = 1$ with the line that passes through Ξ and the origin, i.e., $\mathcal{X} = \lambda\Xi$ with $\lambda = 1/\xi_3$. For simplicity, we assume that views 2 ($\mathcal{V}'$) and 3 ($\mathcal{V}''$) are rotated and translated versions of the canonical view $\mathcal{V}$, i.e., there exist rotations $\boldsymbol{R}', \boldsymbol{R}'' \in \mathrm{SO}(3)$ and translations $\boldsymbol{t}', \boldsymbol{t}'' \in \mathbb{R}^3$, such that[12]

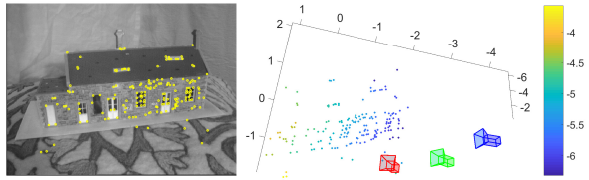
$$\mathcal{V} = \boldsymbol{R}'(\mathcal{V}') + \boldsymbol{t}' = \boldsymbol{R}''(\mathcal{V}'') + \boldsymbol{t}''. \tag{144}$$

The projection $\mathcal{X}'$ of the 3D point Ξ onto $\mathcal{V}'$ is the intersection of $\mathcal{V}'$ with the line that passes from Ξ and the optical center $-\boldsymbol{t}'$ of view 2. However, in practice $\mathcal{X}'$ is only known up to local pixel coordinates with respect to view 2. That is, we can only know the representation $\boldsymbol{x}'$ of $\mathcal{X}'$ with respect to a coordinate system where view 2 is the canonical view. In such a system of coordinates the point Ξ is represented as $\boldsymbol{R}'\Xi + \boldsymbol{t}'$ and hence $\boldsymbol{x}' = \lambda'(\boldsymbol{R}'\Xi + \boldsymbol{t}')$, where λ' is the inverse of the third coordinate of the vector $\boldsymbol{R}'\Xi + \boldsymbol{t}'$. Substituting $\Xi = (1/\lambda)\boldsymbol{x}$ in this equation yields a relation between the local representations[13] $\boldsymbol{x}, \boldsymbol{x}'$ of $\mathcal{X}, \mathcal{X}'$ in $\mathcal{V}$ and $\mathcal{V}'$ respectively as follows:

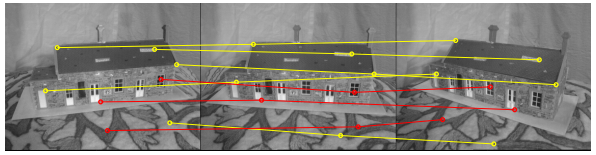
$$\frac{1}{\lambda'}\boldsymbol{x}' = \frac{1}{\lambda}\boldsymbol{R}'\boldsymbol{x} + \boldsymbol{t}'. \tag{145}$$

12. This geometry corresponds to *calibrated cameras*, where the camera projection parameters are known.
13. Without loss of generality we take the local system of coordinates of view 1 to be the same as the global system of coordinates.

(a) Three views of the same scene.



(b) Left: Example of points viewed by all three cameras. Right: Configuration of cameras and the same points depicted in 3D space. Color represents height.



(c) Examples of correct (yellow) and incorrect (red) point correspondences between the three views.

Figure 8: An example of three views of a static scene along with camera configurations and point correspondences (views $2, 4, 6$ of the Model House dataset, provided by the Visual Geometry Group, University of Oxford).

Now, for a vector $\boldsymbol{v} = (\alpha, \beta, \gamma)^\top \in \mathbb{R}^3$, denote by $[\boldsymbol{v}]$ the skew-symmetric matrix

$$[\boldsymbol{v}] = \begin{bmatrix} 0 & \gamma & -\beta \\ -\gamma & 0 & \alpha \\ \beta & -\alpha & 0 \end{bmatrix}, \tag{146}$$

and note that $[\boldsymbol{v}]\boldsymbol{v} = \boldsymbol{0}$. Multiplying equation (145) from the left by $[\boldsymbol{x}']$ gives

$$\frac{1}{\lambda}[\boldsymbol{x}']\boldsymbol{R}'\boldsymbol{x} + [\boldsymbol{x}']\boldsymbol{t}' = \boldsymbol{0}. \tag{147}$$

In exactly the same way, and letting $\boldsymbol{x}''$ be the representation of $\mathcal{X}''$ in the canonical coordinate system for view 3, we have a relationship

$$\frac{1}{\lambda}[\boldsymbol{x}'']\boldsymbol{R}''\boldsymbol{x} + [\boldsymbol{x}'']\boldsymbol{t}'' = \boldsymbol{0}. \tag{148}$$

Degenerate cases aside, equations (147)-(148) are equivalent to the condition

$$\text{Rank}\left(\begin{bmatrix} [\boldsymbol{x}']\boldsymbol{R}'\boldsymbol{x} & [\boldsymbol{x}']\boldsymbol{t}' \\ [\boldsymbol{x}'']\boldsymbol{R}''\boldsymbol{x} & [\boldsymbol{x}'']\boldsymbol{t}'' \end{bmatrix}\right) \leq 1, \tag{149}$$

which in turn is equivalent to the matrix equation[14]

$$[\boldsymbol{x}']\boldsymbol{R}'\boldsymbol{x}\,\boldsymbol{t}''^\top[\boldsymbol{x}'']^\top - [\boldsymbol{x}']\boldsymbol{t}'\boldsymbol{x}^\top\boldsymbol{R}''^\top[\boldsymbol{x}'']^\top = \boldsymbol{0}_{3\times 3}, \tag{150}$$

or more elegantly written as

$$[\boldsymbol{x}']\Big(\sum_{i=1}^{3} x_i \boldsymbol{T}_i\Big)[\boldsymbol{x}'']^\top = \boldsymbol{0}_{3\times 3}, \;\; \boldsymbol{x} = (x_1, x_2, x_3)^\top, \;\; \boldsymbol{T}_i := \boldsymbol{r}_i'\boldsymbol{t}''^\top - \boldsymbol{t}'\boldsymbol{r}_i''^\top, \; i = 1, 2, 3, \tag{151}$$

where $\boldsymbol{r}_i', \boldsymbol{r}_i''$ is the ith column of $\boldsymbol{R}', \boldsymbol{R}''$ respectively. Equation (151) consists of 9 *trilinear* constraints on the local representations $\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}''$ of the imaged 3D point Ξ, among which a maximal number of four are linearly independent (Hartley and Zisserman, 2004). The matrices $(\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3)$ are the slices of the so-called *trifocal tensor*[15] $\mathcal{T} \in \mathbb{R}^{3\times 3} \times \mathbb{R}^{3\times 3} \times \mathbb{R}^{3\times 3}$, which is the mathematical object that encodes the relative geometry of the calibrated three views: indeed, up to a change of coordinates there is a $1-1$ correspondence between trifocal tensors and camera views $\mathcal{V}, \mathcal{V}', \mathcal{V}''$; see Proposition 15 and Theorem 16 in Kileel (2017).

Trifocal tensor estimation as a hyperplane learning problem. Notice that the trilinear constraints (151) are linear in the entries of the tensor $\mathscr{T} = (\boldsymbol{T}_1, \boldsymbol{T}_2, \boldsymbol{T}_3)$, which in its unfolded form can be regarded as a vector $\mathbf{t} \in \mathbb{R}^{27}$. In fact, the space of (uncalibrated) trifocal tensors is an algebraic variety of $\mathbb{R}^{27}$ of dimension 18 (Alzati and Tortora, 2010; Aholt and Oeding, 2014). As already noted, every point correspondence $(\boldsymbol{x}, \boldsymbol{x}', \boldsymbol{x}'')$ contributes four linearly independent equations in $\mathbf{t}$; equivalently, every point correspondence cuts the variety with four hyperplanes. As it turns out though, only 3 of these hyperplanes are algebraically independent with respect to the variety[16], i.e., every generic point correspondence reduces the dimension of the variety by three (Kileel, 2017). As a result, one needs 6 correspondences to get a finite number of candidate trifocal tensors that agree with them. Adding a 7th correspondence allows us to uniquely determine $\mathbf{t}$ via solving a 28×27 homogeneous linear system of equations, while the relative poses $(\boldsymbol{R}', \boldsymbol{t}')$ and $(\boldsymbol{R}'', \boldsymbol{t}'')$ can be extracted from $\mathbf{t}$ by, e.g., the procedure described by Hartley and Zisserman (2004).

The above discussion suggests that given a set of $N' \geq 7$ generic and exact point correspondences $\{(\boldsymbol{x}_j, \boldsymbol{x}_j', \boldsymbol{x}_j'')\}_{j=1}^{N'}$, the coefficient vectors

$$\boldsymbol{c}_1^{(1)}, \boldsymbol{c}_2^{(1)}, \boldsymbol{c}_3^{(1)}, \boldsymbol{c}_4^{(1)}, \ldots, \boldsymbol{c}_1^{(j)}, \boldsymbol{c}_2^{(j)}, \boldsymbol{c}_3^{(j)}, \boldsymbol{c}_4^{(j)} \ldots, \boldsymbol{c}_1^{(N')}, \boldsymbol{c}_2^{(N')}, \boldsymbol{c}_3^{(N')}, \boldsymbol{c}_4^{(N')} \in \mathbb{R}^{27}, \tag{152}$$

of the resulting $N = 4N'$ linear equations span a hyperplane in $\mathbb{R}^{27}$ with normal vector $\mathbf{t}$. We will be referring to the vectors (152) as the *trilinear embeddings* of the point correspondences.

14. Here we have used the fact that for vectors $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}, \boldsymbol{d} \in \mathbb{R}^n$ the $2n \times 2$ matrix $\begin{bmatrix} \boldsymbol{a} & \boldsymbol{b} \\ \boldsymbol{c} & \boldsymbol{d} \end{bmatrix}$ has rank at most 1 if and only if $\boldsymbol{a}\boldsymbol{d}^\top - \boldsymbol{b}\boldsymbol{c}^\top = \boldsymbol{0}_{n\times n}$; thanks to Tianjiao Ding for this observation.
15. In the uncalibrated case, which is more relevant in computer vision applications, the trifocal tensor has exactly the same structure as in (151), with the only difference being that the matrices $\boldsymbol{R}', \boldsymbol{R}''$ are no longer rotations.
16. A more precise way to state this in algebraic-geometric language is that the ideal generated by these four equations has depth 3 in the quotient ring of the trifocal variety.

However, in practice one obtains such correspondences by matching points across images based on the similarity of some local features, such as *SIFT* (Lowe, 1999). As a result, it is typically the case that many of the produced correspondences are incorrect (see Fig. 8(c)), and the problem then becomes that of detecting inliers lying close to a hyperplane of $\mathbb{R}^{27}$, from a dataset corrupted by outliers. Equivalently, one is presented with a codimension 1 subspace learning problem, for which the proposed Dual Principal Component Pursuit (DPCP) is ideally suited; as we show next, the method can achieve superior performance than RANSAC, the latter being the traditional and up to date one of the most popular options in the computer vision community for such problems.

Data. We use the first three views of the datasets *Model House*, *Corridor* and *Merton College III*, provided by the Visual Geometry Group at Oxford University. Each dataset contains different views of the same static scene, together with the projection matrices of each view and high-quality (inlier) point correspondences. From each dataset we randomly pick $N' = 125$ inlier correspondences $\{(\boldsymbol{x}_j, \boldsymbol{x}'_j, \boldsymbol{x}''_j)\}_{j=1}^{N'}$. We further generate $100 \cdot M'/(N'+M')\% = 30\%, 40\%, 50\%$ outlier correspondences $\{(\boldsymbol{o}_j, \boldsymbol{o}'_j, \boldsymbol{o}''_j)\}_{j=1}^{M'}$ as follows: For each triplet $(\mathcal{V}, \mathcal{V}', \mathcal{V}'')$ of views we sample uniformly at random M' points from each view, and randomly match them in M' triplets. We normalize all data according to Hartley (1997) and form a unit ℓ_2-norm dataset $\tilde{\boldsymbol{\mathcal{X}}} = [\boldsymbol{\mathcal{X}}\ \boldsymbol{\mathcal{O}}]\boldsymbol{\Gamma} \in \mathbb{R}^{27\times 4(M'+N')}$ that consists of the trilinear embeddings (see (152)) of all inlier/outlier correspondences.

Algorithms. We compare REAPER and two variants of RANSAC (see §2) with the proposed fast DPCP variants DPCP-IRLS (Algorithm 2) and DPCP-d (Algorithm 3) in the context of outlier detection for trifocal tensor estimation[17]. REAPER, DPCP-IRLS and DPCP-d receive as input the dataset $\tilde{\boldsymbol{\mathcal{X}}}$ and are configured to learn a subspace of dimension 26 in $\mathbb{R}^{27}$ (a hyperplane) that fits $\tilde{\boldsymbol{\mathcal{X}}}$. So does the first RANSAC variant, called H-RANSAC (*Hyperplane*-RANSAC), which is the standard RANSAC that randomly samples 26 trilinear embeddings at each trial. The second variant, called H-G-RANSAC (*Hyperplane-Group*-RANSAC), is exactly as H-RANSAC except that it samples trilinear embeddings in groups of four (instead of individually), where each group is associated with a point correspondence[18]. For H-RANSAC we use as threshold the maximal distance among inlier trilinear embeddings to the hyperplane associated with the ground-truth trifocal tensor. For H-G-RANSAC we use the maximal average such distance among inlier trilinear embeddings in the same group. Regarding time budget, we note that the fastest method is DPCP-d, and then follows REAPER and DPCP-IRLS. For example, for the experiment we are considering and for 40% outliers, DPCP-d needs an average of about 1msec to converge as opposed to 16msec for DPCP-IRLS. Since RANSAC's performance is sensitive to the allocated time budget, we explore a high time budget regime (running time of DPCP-IRLS), as well as a low time budget regime (running time of DPCP-d), and for fairness, we also restrict the running time of rest of the methods accordingly.

Results. We use as a metric the *group precision* of the algorithms that corresponds to a *recall* value equal to 1: given a hyperplane estimate $\hat{\mathcal{H}}$, this induces an ordering of all the inlier/outlier point correspondences based on increasing average distance of the corresponding 4-tuples of trilinear embeddings to $\hat{\mathcal{H}}$. Letting α be the maximal such average distance that corresponds to an inlier point correspondence, our metric is the percentage of the inliers among all correspondences with average distance to $\hat{\mathcal{H}}$ less or equal than α. Tables 2, 3 and 4 report the precision of the algorithms for the three different scenes, *Corridor, Model House* and *Merton College III* respectively, for different outlier ratios and different time budgets.

17. The purely low-rank methods SE-RPCA and ℓ_2-RPCA are unsuitable for learning a hyperplane; since Coherence Pursuit (CoP) performed much better than them with synthetic data, we also included it in our experiments, but do not report its performance as it was not competitive with the other tested methods, i.e., RANSAC, REAPER and DPCP. Notice from Fig. 7 that with synthetic data CoP fails precisely at the case of a hyperplane.

18. Since the trilinear embeddings also lie in a union of coordinate hyperplanes irrelevant to the trifocal tensor, taking this grouping into consideration prevents the method from learning one of these hyperplanes. This is implicitly achieved by the RANSAC variant penalizing the *re-projection error*, most commonly used for trifocal tensor estimation. We have also tested this variant, however due to its higher complexity and the running time restrictions enforced in this experiment, it did not perform on par with the rest of the methods and hence we do not report its performance further. A group-based DPCP approach is the subject of current research.

Table 2: Algorithm precision when recall value is 1 for the first three views of dataset *Corridor*.

Algorithm vs.	30% outliers		40% outliers		50% outliers	
% of outliers & time budget	high t.b.	low t.b.	high t.b.	low t.b.	high t.b.	low t.b.
H-RANSAC	0.698	0.735	0.601	0.625	0.502	0.522
H-G-RANSAC	**1.000**	0.992	0.992	0.977	0.992	0.665
REAPER-IRLS	**1.000**	**1.000**	**1.000**	0.992	0.984	0.954
DPCP-IRLS	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	0.977
DPCP-d	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**	**1.000**

Table 3: Algorithm precision when recall value is 1 for the first three views of dataset *Model House*.

Algorithm vs.	30% outliers		40% outliers		50% outliers	
% of outliers & time budget	high t.b.	low t.b.	high t.b.	low t.b.	high t.b.	low t.b.
H-RANSAC	0.702	0.725	0.601	0.619	0.502	0.510
H-G-RANSAC	**0.996**	**0.984**	**0.992**	0.893	**0.984**	0.587
REAPER-IRLS	0.992	0.977	0.977	0.943	0.880	0.820
DPCP-IRLS	0.969	0.980	0.969	0.965	0.947	0.856
DPCP-d	0.984	**0.984**	0.977	**0.977**	0.954	**0.954**

Table 4: Algorithm precision when recall value is 1 for the three views of dataset *Merton College 3*.

Algorithm vs.	30% outliers		40% outliers		50% outliers	
% of outliers & time budget	high t.b.	low t.b.	high t.b.	low t.b.	high t.b.	low t.b.
H-RANSAC	0.698	0.723	0.607	0.620	0.505	0.515
H-G-RANSAC	**1.000**	0.980	0.992	0.906	**0.992**	0.546
REAPER-IRLS	**1.000**	0.992	0.984	0.919	0.839	0.786
DPCP-IRLS	**1.000**	0.992	**1.000**	0.936	**0.992**	0.794
DPCP-d	**1.000**	**1.000**	**1.000**	**1.000**	**0.992**	**0.992**

As a first observation note that H-RANSAC essentially fails for all three datasets, with a precision not exceeding 73.5 %, even for the case of 30% outliers. Moreover, its precision is higher for low time budget, which at first sight seems contradictory. Both phenomena are attributed to

a combination of insufficient time budget together with the fact that the dataset $\tilde{\mathcal{X}}$ also lies in a union of coordinate hyperplanes irrelevant to the trifocal tensor; this is evident by inspecting the zero structure of the trilinear embeddings, not shown here. As a result, given different time budgets H-RANSAC identifies different hyperplanes that fit a significant portion of the data, but with none coinciding with the trifocal tensor.

The aforementioned issue is remedied by H-G-RANSAC, which forces the estimated hyperplane to fit groups of trilinear embeddings respecting the underlying point-point-point correspondences. Indeed, this dramatically improves its performance and for the dataset *Model House* in Table 3 it is the best performing method for high time budget. However, H-G-RANSAC is not able to cope with 50% outliers at low time budget: its highest precision in that regime is only 66.5% for the dataset *Corridor* in Table 2.

On the other hand, DPCP-d is not only fast, but also very robust: in the challenging regime of 50% outliers it is the only method that gives 100% precision for the dataset *Corridor*, 95.4% for *Model House*, and 99.2% for *Merton College III*, while in this low time budget regime the second best method is DPCP-IRLS with precision 97.7%, 85.6% and 79.4% respectively. Interestingly, DPCP-d is performing uniformly better than DPCP-IRLS even if the latter is allowed to run to convergence (high time budget); we attribute this to the fact that DPCP-d is designed to explicitly handle noise. Overall, DPCP-d is the best performing method in low time budget across all datasets, and the best performing method in high budget for datasets *Corridor* and *Merton College III*. Finally, REAPER performs somewhere between DPCP-IRLS and H-G-RANSAC, outperforming DPCP-IRLS only on a few occasions. This is consistent with the experiment of Fig. 6 on synthetic data, according to which the advantage of DPCP over REAPER is precisely the codimension 1 case, where the latter fails.

In conclusion, even though RANSAC can have a very high precision given sufficient time budget, once the latter is restricted its performance can drop dramatically. This is particularly the case for large outlier ratios, a regime where an exponentially large time budget might be needed. Moreover, RANSAC is very sensitive to its thresholding parameter, which in the above experiment was set using knowledge of the ground truth. Clearly, such knowledge is not available in practice and different choices for this parameter are expected to only lead to performance degradation. On the other hand, DPCP-d was found to be the best method in the above experiment, combining low running time, high precision and robustness to its thresholding parameter, suggesting that the proposed Dual Principal Component Pursuit can be a useful or even superior alternative to popular approaches such as RANSAC, for three-view geometry or other computer vision applications.

8. Conclusions

We presented and studied a solution to the problem of robust principal component analysis in the presence of outliers, called *Dual Principal Component Pursuit (DPCP)*. The heart of the proposed method consisted of a non-convex ℓ_1 optimization problem on the sphere, for which a solution strategy based on a recursion of linear programs was analyzed. Rigorous mathematical analysis revealed that DPCP is a natural method for learning the inlier subspace in the presence of outliers, even in the challenging regime of large outlier ratios and high subspace relative dimensions. In fact, experiments on synthetic data showed that DPCP was the only method that could handle 70% outliers inside a 30-dimensional ambient space, irrespectively of the subspace dimension. Moreover, experiments with real images in the context of three-view geometry showed that DPCP can outperform popular alternatives such as RANSAC, suggesting its potential in computer vision applications.

Appendix A. Review of existing results on Problems (9) and (10)

In this appendix we state three results that are important for our mathematical analysis, already known in Späth and Watson (1987). For the sake of clarity and convenience, we have also taken the liberty of writing complete proofs of the statements, as not all of them can be found in Späth and Watson (1987).

Proposition 14 *Let $\boldsymbol{\mathcal{Y}} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L] \in D \times L$ be full rank. Then any global solution $\boldsymbol{b}^*$ to*

$$\min_{\boldsymbol{b}^\top \boldsymbol{b}=1} \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}\right\|_1, \tag{153}$$

must be orthogonal to $(D-1)$ linearly independent columns of $\boldsymbol{\mathcal{Y}}$.

Proof Let $\boldsymbol{b}^*$ be an optimal solution of (153). Then $\boldsymbol{b}^*$ must satisfy the first order optimality relation

$$\boldsymbol{0} \in \boldsymbol{\mathcal{Y}}\, \mathrm{Sgn}(\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}^*) + \lambda \boldsymbol{b}^*, \tag{154}$$

where λ is a scalar Lagrange multiplier parameter, and Sgn is the sub-differential of the ℓ_1 norm. Without loss of generality, let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K$ be the columns of $\boldsymbol{\mathcal{Y}}$ to which $\boldsymbol{b}^*$ is orthogonal. Then equation (154) implies that there exist real numbers $\alpha_1, \ldots, \alpha_K \in [-1, 1]$ such that

$$\sum_{j=1}^{K} \alpha_j \boldsymbol{y}_j + \sum_{j=K+1}^{L} \mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^*) \boldsymbol{y}_j + \lambda \boldsymbol{b}^* = \boldsymbol{0}. \tag{155}$$

Now, suppose that the span of $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K$ is of dimension less than $D-1$. Then there exists a unit norm vector $\boldsymbol{\zeta} \in \mathbb{S}^{D-1}$ that is orthogonal to all $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_K, \boldsymbol{b}^*$, and multiplication of (155) from the left by $\boldsymbol{\zeta}^\top$ gives

$$\sum_{j=K+1}^{L} \mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^*) \boldsymbol{\zeta}^\top \boldsymbol{y}_j = 0. \tag{156}$$

Furthermore, we can choose a sufficiently small $\varepsilon > 0$, such that

$$\mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^* + \varepsilon \boldsymbol{y}_j^\top \boldsymbol{\zeta}) = \mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^*), \;\; \forall j \in [L]. \tag{157}$$

The above equation is trivially true for all j such that $\boldsymbol{y}_j^\top \boldsymbol{b}^* = 0$, because in that case $\boldsymbol{y}_j^\top \boldsymbol{\zeta} = 0$ by the definition of $\boldsymbol{\zeta}$. On the other hand, if $\boldsymbol{y}_j^\top \boldsymbol{b}^* \neq 0$, then a small perturbation ϵ will not change the sign of $\boldsymbol{y}_j^\top \boldsymbol{b}^*$. Consequently, we can write

$$\left|\boldsymbol{y}_j^\top (\boldsymbol{b}^* + \varepsilon \boldsymbol{\zeta})\right| = \left|\boldsymbol{y}_j^\top \boldsymbol{b}^*\right| + \varepsilon\, \mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^*) \boldsymbol{y}_j^\top \boldsymbol{\zeta}, \; \forall j \in [L] \tag{158}$$

and so

$$\left\|\boldsymbol{\mathcal{Y}}^\top (\boldsymbol{b}^* + \varepsilon \boldsymbol{\zeta})\right\|_1 = \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}^*\right\|_1 + \varepsilon \sum_{j=K+1}^{L} \mathrm{Sign}(\boldsymbol{y}_j^\top \boldsymbol{b}^*) \boldsymbol{\zeta}^\top \boldsymbol{y}_j = \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}^*\right\|_1. \tag{159}$$

However,

$$\left\|\boldsymbol{b}^* + \varepsilon\boldsymbol{\zeta}\right\|_2 = \sqrt{1+\varepsilon^2} > 0, \tag{160}$$

and normalizing $\boldsymbol{b}^* + \varepsilon\boldsymbol{\zeta}$ to have unit ℓ_2 norm, we get a contradiction on $\boldsymbol{b}^*$ being a global solution. ■

Proposition 15 *Let $\boldsymbol{\mathcal{Y}} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]$ be a $D \times L$ matrix of rank D. Then problem*

$$\min_{\boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1} \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}\right\|_1 \tag{161}$$

admits a computable solution $\boldsymbol{n}_{k+1}$ that is orthogonal to $(D-1)$ linearly independent points of $\boldsymbol{\mathcal{Y}}$.

Proof Let $\boldsymbol{n}_{k+1}$ be a solution to $\min_{\boldsymbol{b}^\top \hat{\boldsymbol{n}}_k = 1} \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}\right\|_1$ that is orthogonal to less than $D-1$ linearly independent points of $\boldsymbol{\mathcal{Y}}$. Then we can find a unit norm vector $\boldsymbol{\zeta}$ that is orthogonal to the same points of $\boldsymbol{\mathcal{Y}}$ that $\boldsymbol{n}_{k+1}$ is orthogonal to, and moreover $\boldsymbol{\zeta} \perp \boldsymbol{n}_{k+1}$. In addition, we can find a sufficiently small $\varepsilon > 0$ such that

$$\left\|\boldsymbol{\mathcal{Y}}^\top(\boldsymbol{n}_{k+1} + \varepsilon\boldsymbol{\zeta})\right\|_1 = \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{n}_{k+1}\right\|_1 + \varepsilon \sum_{j:\, \boldsymbol{n}_{k+1} \not\perp \boldsymbol{y}_j} \operatorname{Sign}(\boldsymbol{y}_j^\top \boldsymbol{n}_{k+1}) \boldsymbol{\zeta}^\top \boldsymbol{y}_j, \tag{162}$$

where

$$\sum_{j:\, \boldsymbol{n}_{k+1} \not\perp \boldsymbol{y}_j} \operatorname{Sign}(\boldsymbol{y}_j^\top \boldsymbol{n}_{k+1}) \boldsymbol{\zeta}^\top \boldsymbol{y}_j \leq 0. \tag{163}$$

Since $\boldsymbol{n}_{k+1}$ is optimal, it must be the case that

$$\sum_{j:\, \boldsymbol{n}_{k+1} \not\perp \boldsymbol{y}_j} \operatorname{Sign}(\boldsymbol{y}_j^\top \boldsymbol{n}_{k+1}) \boldsymbol{\zeta}^\top \boldsymbol{y}_j = 0, \tag{164}$$

and so

$$\left\|\boldsymbol{\mathcal{Y}}^\top(\boldsymbol{n}_{k+1} + \varepsilon\boldsymbol{\zeta})\right\|_1 = \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{n}_{k+1}\right\|_1. \tag{165}$$

By (165) we see that as we vary ε the objective remains unchanged. Notice also that varying ε preserves all zero entries appearing in the vector $\boldsymbol{\mathcal{Y}}^\top \boldsymbol{n}_{k+1}$. Furthermore, because of (164), it is always possible to either decrease or increase ε until an additional zero is achieved, i.e., until $\boldsymbol{n}_{k+1} + \varepsilon\boldsymbol{\zeta}$ becomes orthogonal to a point of $\boldsymbol{\mathcal{Y}}$ that $\boldsymbol{n}_{k+1}$ is not orthogonal to. Then we can replace $\boldsymbol{n}_{k+1}$ with $\boldsymbol{n}_{k+1} + \varepsilon\boldsymbol{\zeta}$ and repeat the process, until we get some $\boldsymbol{n}_{k+1}$ that is orthogonal to $D-1$ linearly independent points of $\boldsymbol{\mathcal{Y}}$. ■

Proposition 16 *Let $\boldsymbol{\mathcal{Y}} = [\boldsymbol{y}_1, \ldots, \boldsymbol{y}_L]$ be a $D \times L$ matrix of rank D. Suppose that for each problem* (161) *a solution $\boldsymbol{n}_{k+1}$ is chosen such that $\boldsymbol{n}_{k+1}$ is orthogonal to $D-1$ linearly independent points of $\boldsymbol{\mathcal{Y}}$, in accordance with Proposition 15. Then the sequence $\{\boldsymbol{n}_k\}$ converges to a critical point of problem* (153) *in a finite number of steps.*

Proof If $\boldsymbol{n}_{k+1} = \hat{\boldsymbol{n}}_k$, then inspection of the first order optimality conditions of the two problems, reveals that $\hat{\boldsymbol{n}}_k$ is a critical point of $\min_{\boldsymbol{b}^\top \boldsymbol{b}=1} \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}\right\|_1$. If $\boldsymbol{n}_{k+1} \neq \hat{\boldsymbol{n}}_k$, then $\left\|\boldsymbol{n}_{k+1}\right\|_2 > 1$, and so $\left\|\boldsymbol{\mathcal{Y}}^\top \hat{\boldsymbol{n}}_{k+1}\right\|_1 < \left\|\boldsymbol{\mathcal{Y}}^\top \hat{\boldsymbol{n}}_k\right\|_1$. As a consequence, if $\boldsymbol{n}_{k+1} \neq \hat{\boldsymbol{n}}_k$, then $\hat{\boldsymbol{n}}_k$ can not arise as a solution for some $k' > k$. Now, because of Proposition 15, for each k, there is a finite number of candidate directions $\boldsymbol{n}_{k+1}$. These last two observations imply that the sequence $\{\boldsymbol{n}_k\}$ must converge in a finite number of steps to a critical point of $\min_{\boldsymbol{b}^\top \boldsymbol{b}=1} \left\|\boldsymbol{\mathcal{Y}}^\top \boldsymbol{b}\right\|_1$. ■

Acknowledgement

This work was supported by NSF grants 1447822, 1618637 and 1704458. The first author is thankful to Dr. Daniel P. Robinson for many comments that helped improve this manuscript, to Dr. Glyn Harman for indicating the proof of Lemma 8, to Dr. Zhihui Zhu for pointing out the approximate asymptotic behavior $M < \mathcal{O}(N^2)$, and to Tianjiao Ding for his help in producing the experiments on the trifocal tensor. The authors also thank Dr. Gilad Lerman and Dr. Laurent Kneip for insightful discussions, as well as the two anonymous reviewers for their constructive comments.

References

C. Aholt and L. Oeding. The ideal of the trifocal variety. *Math. Comp.*, 83:2553–2574, 2014.

A. Alzati and A. Tortora. A geometric approach to the trifocal tensor. *Mathematical Imaging and Vision*, 38:159–170, 2010.

L. Balzano, R. Nowak, and B. Recht. Online identification and tracking of subspaces from highly incomplete information. In *Communication, Control, and Computing (Allerton), 2010 48th Annual Allerton Conference on*, pages 704–711. IEEE, 2010.

J. Beck. Sums of distances between points on a sphere—an application of the theory of irregularities of distribution to discrete geometry. *Mathematika*, 31(01):33–41, 1984.

J. S. Brauchart and P. J. Grabner. Distributing many points on spheres: minimal energy and designs. *Journal of Complexity*, 31(3):293–326, 2015.

J.P. Brooks, J.H. Dulá, and E.L. Boone. A pure l1-norm principal component analysis. *Computational statistics & data analysis*, 61:83–98, 2013.

E. Candès and M. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008.

E. Candès, M. Wakin, and S. Boyd. Enhancing sparsity by reweighted ℓ_1 minimization. *Journal of Fourier Analysis and Applications*, 14(5):877–905, 2008.

E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3), 2011.

R. Chartrand and W. Yin. Iteratively reweighted algorithms for compressive sensing. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3869–3872. IEEE, 2008.

I. Daubechies, R. DeVore, M. Fornasier, and C. S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Communications on Pure and Applied Mathematics*, 63(1): 1–38, 2010.

J. Dick. Applications of geometric discrepancy in numerical analysis and statistics. *Applied Algebra and Number Theory*, 2014.

L. Elden. Solving quadratically constrained least squares problems using a differential-geometric approach. *BIT Numerical Mathematics*, 42(2):323–335, 2002.

E. Elhamifar and R. Vidal. Robust classification using structured sparse representation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

J. Feng, H. Xu, and S. Yan. Online robust pca via stochastic optimization. In *Advances in Neural Information Processing Systems*, pages 404–412, 2013.

M. A. Fischler and R. C. Bolles. RANSAC random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 26: 381–395, 1981.

D. Gabay and B. Mercier. A dual algorithm for the solution of nonlinear variational problems via finite-element approximations. *Comp. Math. Appl.*, 2:17–40, 1976.

W. Gander. Least squares with a quadratic constraint. *Numerische Mathematik*, 36(3):291–307, 1980.

G. H Golub and U. Von Matt. Quadratically constrained least squares and quadratic problems. *Numerische Mathematik*, 59(1):561–580, 1991.

P. J. Grabner and R.F. Tichy. Spherical designs, discrepancy and numerical integration. *Math. Comp.*, 60(201):327–336, 1993. ISSN 0025-5718. doi: 10.2307/2153170. URL `http://dx.doi.org/10.2307/2153170`.

P. J. Grabner, B. Klinger, and R.F. Tichy. Discrepancies of point sequences on the sphere and numerical integration. *Mathematical Research*, 101:95–112, 1997.

Inc. Gurobi Optimization. Gurobi optimizer reference manual, 2015. URL `http://www.gurobi.com`.

G. Harman. Variations on the koksma-hlawka inequality. *Uniform Distribution Theory*, 5(1):65–78, 2010.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge, 2nd edition, 2004.

R. I. Hartley. In defense of the 8-point algorithm. *IEEE Trans. Pattern Anal. Mach. Intell.*, 19(6): 580–593, 1997.

E. Hlawka. Discrepancy and riemann integration. *Studies in Pure Mathematics*, pages 121–129, 1971.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24:417–441, 1933.

P. Huber. *Robust Statistics*. John Wiley & Sons, New York, 1981.

I. Jolliffe. *Principal Component Analysis*. Springer-Verlag, 2nd edition, 2002.

J. Kileel. Minimal problems for the calibrated trifocal variety. *SIAM Journal on Applied Algebra and Geometry*, 1:575–598, 2017.

W. Ku, R. H. Storer, and C. Georgakis. Disturbance detection and isolation by dynamic principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, 30:179–196, 1995.

L. Kuipers and H. Niederreiter. Uniform distribution of sequences. *Courier Corporation*, 2012.

G. Lerman and T. Maunu. Fast, robust and non-convex subspace recovery. *Information and Inference: A Journal of the IMA*, 2017.

G. Lerman and T. Maunu. An overview of robust subspace recovery. *arXiv:1803.01013v1*, 2018.

G. Lerman and T. Zhang. ℓ_p-recovery of the most significant subspace among multiple subspaces with outliers. *Constructive Approximation*, 40(3):329–385, 2014.

G. Lerman, M. B. McCoy, J. A Tropp, and T. Zhang. Robust computation of linear models by convex relaxation. *Foundations of Computational Mathematics*, 15(2):363–410, 2015.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, pages 663–670, 2010.

David G. Lowe. Object recognition from local scale-invariant features. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1150–1157, 1999.

S. Loyd, M. Mohseni, and P. Rebentrost. Quantum principal component analysis. *Nature Physics*, 10(9):631–633, 2014.

B. C. Moore. Principal component analysis in linear systems: Controllability, observability, and model reduction. *IEEE Transactions on Automatic Control*, 26(1):17–32, 1981.

S. Nam, M.E. Davies, M. Elad, and R. Gribonval. The cosparse analysis model and algorithms. *Applied and Computational Harmonic Analysis*, 34(1):30–56, 2013.

K. Pearson. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh and Dublin Philosphical Magazine and Journal of Science*, 2:559–572, 1901.

A. Price, N. Patterson, R. M. Plenge, M. E. Weinblatt, N. A. Shadick, and D. Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature Genetics*, 38(8):904–909, 2006.

Q. Qu, J. Sun, and J. Wright. Finding a sparse vector in a subspace: Linear sparsity using alternating directions. In *Advances in Neural Information Processing Systems*, pages 3401–3409, 2014.

M. Rahmani and G. Atia. Coherence pursuit: Fast, simple, and robust principal component analysis. *arXiv:1609.04789v3*, 2017.

M. Soltanolkotabi and E. J. Candès. A geometric analysis of subspace clustering with outliers. *Annals of Statistics*, 40(4):2195–2238, 2012.

H. Späth and G.A. Watson. On orthogonal linear ℓ_1 approximation. *Numerische Mathematik*, 51 (5):531–543, 1987.

D.A. Spielman, H. Wang, and J. Wright. Exact recovery of sparsely-used dictionaries. In *Proceedings of the 23d international joint conference on Artificial Intelligence*, pages 3087–3090. AAAI Press, 2013.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere ii: Recovery by riemannian trust-region method. *arXiv preprint arXiv:1511.04777*, 2015a.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *arXiv preprint arXiv:1511.03607*, 2015b.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery using nonconvex optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2351–2360, 2015c.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere. In *Sampling Theory and Applications (SampTA), 2015 International Conference on*, pages 407–410. IEEE, 2015d.

M. C. Tsakiris and R. Vidal. Hyperplane clustering via dual principal component pursuit. In *International Conference on Machine Learning*, 2017.

M.C. Tsakiris and R. Vidal. Dual principal component pursuit. In *ICCV Workshop on Robust Subspace Learning and Computer Vision*, pages 10–18, 2015.

R. Vidal. Subspace clustering. *IEEE Signal Processing Magazine*, 28(3):52–68, March 2011.

S. Vyas and L. Kumaranayake. Constructing socio-economic status indices: how to use principal components analysis. *Health Policy and Planning*, 21:459–468, 2006.

Y. Wang, C. Dicle, M. Sznaier, and O. Camps. Self scaled regularized robust regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3261–3269, 2015.

H. Xu, C. Caramanis, and S. Sanghavi. Robust pca via outlier pursuit. *IEEE transactions on information theory*, 58(5):3047–3064, 2012.

C. You, D. Robinson, and R. Vidal. Provable self-representation based outlier detection in a union of subspaces. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Teng Zhang and Gilad Lerman. A novel m-estimator for robust pca. *The Journal of Machine Learning Research*, 15(1):749–808, 2014.

Journal of Machine Learning Research 19 (2018) 1-32 Submitted 07/17; Revised 06/18; Published 07/18

Distributed Proximal Gradient Algorithm for Partially Asynchronous Computer Clusters *

Yi Zhou ZHOU.1172@OSU.EDU
Yingbin Liang LIANG.889@OSU.EDU
Department of Electrical and Computer Engineering
The Ohio State University

Yaoliang Yu YAOLIANG.YU@UWATERLOO.CA
Department of Computer Science
University of Waterloo

Wei Dai WDAI@CS.CMU.EDU
Eric P. Xing EPXING@CS.CMU.EDU
Machine Learning Department
Carnegie Mellon University

Editor: Tong Zhang

Abstract

With ever growing data volume and model size, an error-tolerant, communication efficient, yet versatile distributed algorithm has become vital for the success of many large-scale machine learning applications. In this work we propose m-PAPG, an implementation of the flexible proximal gradient algorithm in model parallel systems equipped with the partially asynchronous communication protocol. The worker machines communicate asynchronously with a controlled staleness bound s and operate at different frequencies. We characterize various convergence properties of m-PAPG: 1) Under a general non-smooth and non-convex setting, we prove that every limit point of the sequence generated by m-PAPG is a critical point of the objective function; 2) Under an error bound condition of convex objective functions , we prove that the optimality gap decays linearly for every s steps; 3) Under the Kurdyka-Łojasiewicz inequality and a sufficient decrease assumption , we prove that the sequences generated by m-PAPG converge to the same critical point, provided that a proximal Lipschitz condition is satisfied.

Keywords: proximal gradient, distributed system, model parallel, partially asynchronous, machine learning

1. Introduction

The composite minimization problem

$$\min_{\mathbf{x}\in\mathbb{R}^d} \; f(\mathbf{x}) + g(\mathbf{x}) \tag{1}$$

has drawn a lot of recent attention due to its ubiquity in machine learning and statistical applications. Typically, the first term

$$f(\mathbf{x}) := \frac{1}{n}\sum_{i=1}^{n} f_i(\mathbf{x}) \tag{2}$$

is a smooth loss function over n training samples that describes the fitness to data, and the second term g is a nonsmooth regularization function that encodes *a priori* information. We list below some popular examples under this framework.

- Lasso: least squares loss $f_i(\mathbf{x}) = (y_i - \mathbf{a}_i^\top \mathbf{x})^2$ and ℓ_1 norm regularizer $g(\mathbf{x}) = \|\mathbf{x}\|_1$;
- Logistic regression: logistic loss $f_i = \log(1 + \exp(-y_i \mathbf{a}_i^\top \mathbf{x}_i))$;
- Boosting: exponential loss $f_i(\mathbf{x}) = \exp(-y_i \mathbf{a}_i^\top \mathbf{x})$;
- Support vector machines: hinge loss $f_i(\mathbf{x}) = \max\{0, 1 - y_i \mathbf{a}_i^\top \mathbf{x}\}$ and (squared) ℓ_2 norm regularizer $g(\mathbf{x}) = \|\mathbf{x}\|_2^2$.

Over the years there is also a rising interest in using nonconvex losses f (mainly for robustness against outlying observations) Collobert et al. (2006); Wu and Liu (2007); Xu et al. (2006); Yu et al. (2015) and nonconvex regularizers g (mainly for smaller bias in statistical estimation) Fan and Li (2001); Zhang and Zhang (2012).

Due to the apparent importance of the composite minimization framework and the rapidly growing size in both dimension (d) and volume (n) of data, there is a strong need to develop a practical *parallel* system that can solve the problem in (1) efficiently and in a scale that is impossible for a single machine Agarwal and Duchi (2011); Bertsekas and Tsitsiklis (1989); Dean and Ghemawat (2008); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Low et al. (2012); Zaharia et al. (2010). Existing systems can be categorized by how communication among worker machines is managed: bulk synchronous (also called fully synchronous) Dean and Ghemawat (2008); Valiant (1990); Zaharia et al. (2010); Lorenzo and Scutari (2016), totally asynchronous Baudet (1978); Bertsekas and Tsitsiklis (1989); Low et al. (2012), and partially asynchronous (a.k.a. stale synchronous or chaotic) Agarwal and Duchi (2011); Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Tseng (1991). Bulk synchronous parallel (BSP) systems explicitly force synchronization barriers so that the worker machines can stay on the same page to ensure correctness. However, in a real deployed parallel system, BSP usually suffers from the straggler problem, that is, the performance of the whole system is bottlenecked at the bandwidth of communication and the *slowest* worker machine. On the other hand, totally asynchronous systems do not put any constraint on synchronization, hence achieve much greater throughputs by potentially sacrificing the correctness of the algorithm. Partially asynchronous parallel (PAP) systems Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969) are a compromise between the previous two: it allows the worker machines to communicate asynchronously up to a controlled staleness and to perform updates at different paces. PAP is particularly suitable for machine learning applications, where iterative algorithms that are robust to small computational errors are usually favored for finding an appropriate solution. Due to its flexibility, the PAP mechanism has been the method of choice in many recent practical implementations Agarwal and Duchi (2011);

∗. The material in this paper is presented in part at the 19th International Conference on Artificial Intelligence and Statistics (AISTATS), Cadiz, Spain, 2016.

Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Liu and Wright (2015); Recht et al. (2011).

Existing parallel systems can also be categorized by how computation is divided among worker machines: data parallel and model parallel. Data parallel systems usually distribute the computation involving each component function f_i in (2) into different worker machines, which is suitable when $n \gg d$, i.e., large data volume but moderate model size. In this setting the stochastic proximal gradient algorithm, along with the PAP protocol, has been shown to be quite effective in solving the composite problem (1) Agarwal and Duchi (2011); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014). Some other works developed ADMM-based algorithms for data parallelism Hong et al. (2016) and stochastic variance-reduced gradient algorithms under the PAP protocol Huo and Huang (2017); Fang and Lin (2017), and proved their effectiveness both theoretically and empirically. In this work, we focus on the "dual" model parallel regime where $d \gg n$, i.e., large model size but moderate data volume. In modern machine learning and statistics applications, it is not uncommon that the dimensionality of data largely exceeds its volume, for example, in computational biology, conducting an experimental study that involves many patients can be very expensive but for each patient, technology (e.g. next-generation genome sequencing) has advanced to a stage where taking a large number of measurements (model parameters) is relatively cheap. Deep neural networks are another example that calls for model parallelism. Not surprisingly, the design of a model parallel system is fundamentally different from that of a data parallel system, and so is the subsequent analysis.

To achieve model parallelism, the model $\mathbf{x}$ is partitioned into different (disjoint) blocks and is distributed among many worker machines. In this setting, the block proximal gradient algorithm has been proposed to solve the composite problem (1) Fercoq and Richtárik (2015); Lu and Xiao (2015); Richtárik and Takáč (2014), although under the more restrictive BSP protocol. Other works proposed ADMM-based algorithm for model parallelism to solve the sparse PCA problem Hajinezhad and Hong (2015). Under the PAP protocol, the only work that we are aware of is Bertsekas and Tsitsiklis (1989) which focused on a special case of (1) where g is an indicator function of a convex set, and Tseng (1991) which established a periodic linear rate of convergence under an error bound condition. Our main goal in this work is to provide a formal convergence analysis of the model parallel proximal gradient algorithm under the more flexible PAP communication protocol, and our results naturally extend those in Bertsekas and Tsitsiklis (1989); Tseng (1991) to allow nonsmooth and nonconvex functions.

Our main contributions in this work are: 1). We propose m-PAPG, an extension of the proximal gradient algorithm to the model parallel and partially asynchronous setting. In specific, the worker machines in the system can communicate with each other to synchronize the model parameters with staleness. 2). We provide a rigorous analysis of the convergence properties of m-PAPG, allowing both *nonsmooth* and *nonconvex* functions. In particular, we prove in Theorem 7 that any limit point of the sequences generated by m-PAPG is a critical point. 3) Under an additional error bound condition of convex objective functions, we prove in Theorem 9 that the function values generated by m-PAPG decays periodically linearly. 4) Lastly, using the Kurdyka-Łojasiewicz (KŁ) inequality Bolte et al. (2014) and under a sufficient decrease assumption, we prove in Theorem 11 that for functions that

satisfy a proximal Lipschitz condition the whole sequences of m-PAPG converge to a single critical point.

This paper proceeds as follows: We first set up the notations and definitions in Section 2. The proposed algorithm m-PAPG is presented in Section 3, and convergence analysis are detailed in Sections 4 to 6. The implementation of m-PAPG on a distributed system is detailed in Section 7, and numerical experiments are reported in Section 8. Section 9 concludes our work.

2. Preliminaries

We first recall some fundamental definitions that will be needed in our analysis. Throughout, $h : \mathbb{R}^d \to (-\infty, +\infty]$ denotes an extended real-valued function that is proper and closed, i.e., its domain $\operatorname{dom} h := \{\mathbf{x} : h(\mathbf{x}) < +\infty\}$ is nonempty and its sublevel set $\{\mathbf{x} : h(\mathbf{x}) \le \alpha\}$ is closed for all $\alpha \in \mathbb{R}$. Since the function h may not be smooth or convex, we need the following generalized notion of "derivative."

Definition 1 (Subdifferential and critical point, e.g. Rockafellar and Wets (1997)) *The Frechét subdifferential $\hat{\partial} h$ of h at $\mathbf{x} \in \operatorname{dom} h$ is the set of $\mathbf{u}$ such that*

$$\liminf_{\mathbf{z} \neq \mathbf{x}, \mathbf{z} \to \mathbf{x}} \frac{h(\mathbf{z}) - h(\mathbf{x}) - \mathbf{u}^\top(\mathbf{z} - \mathbf{x})}{\|\mathbf{z} - \mathbf{x}\|} \ge 0, \tag{3}$$

while the (limiting) subdifferential ∂h at $\mathbf{x} \in \operatorname{dom} h$ is the "closure" of $\hat{\partial} h$:

$$\{\mathbf{u} : \exists \mathbf{x}^k \to \mathbf{x}, h(\mathbf{x}^k) \to h(\mathbf{x}), \mathbf{u}^k \in \hat{\partial} h(\mathbf{x}^k), \mathbf{u}^k \to \mathbf{u}\}. \tag{4}$$

The critical points of h are $\operatorname{crit} h := \{\mathbf{x} : \mathbf{0} \in \partial h(\mathbf{x})\}$.

When h is continuously differentiable or convex, the subdifferential ∂h and the set of critical points $\operatorname{crit} h$ coincide with the usual notions. For a closed function h, its subdifferential is either nonempty at any point in its domain or the subgradient diverges to some "direction" (Rockafellar and Wets, 1997, Corollary 8.10).

Definition 2 (Distance and projection) *The distance function w.r.t. a closed set $\Omega \subseteq \mathbb{R}^d$ is defined as:*

$$\operatorname{dist}_\Omega(\mathbf{x}) := \min_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|, \tag{5}$$

while the metric projection onto Ω is defined as:

$$\operatorname{proj}_\Omega(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{y} \in \Omega} \|\mathbf{y} - \mathbf{x}\|, \tag{6}$$

where $\|\cdot\|$ is the usual Euclidean norm.

Note that $\operatorname{proj}_\Omega(\mathbf{x})$ is single-valued for all $\mathbf{x} \in \mathbb{R}^d$ if and only if Ω is convex.

Definition 3 (Proximal map, e.g. Rockafellar and Wets (1997)) *The proximal map of a closed and proper function h is (with parameter $\eta > 0$):*

$$\operatorname{prox}_h^\eta(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{z} \in \mathbb{R}^d} h(\mathbf{z}) + \tfrac{1}{2\eta}\|\mathbf{z} - \mathbf{x}\|^2. \tag{7}$$

Occasionally, we will write prox_h *instead of* prox_h^1.

Clearly, for the indicator function $h(\mathbf{x}) = \iota_\Omega(\mathbf{x})$, which takes the value 0 for $\mathbf{x} \in \Omega$ and ∞ otherwise, its proximal map (with any $\eta > 0$) reduces to the metric projection proj_Ω. If h decreases slower than a quadratic function (in particular, when h is bounded below), then its proximal map is well-defined for all (small) η Rockafellar and Wets (1997). If h is convex, then its proximal map is always a singleton while for nonconvex h, the proximal map can be set-valued. In the latter case we will also abuse the notation $\mathrm{prox}_h^\eta(\mathbf{x})$ for an arbitrary element from that set. For convex functions, the proximal map is nonexpansive:

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ \|\mathrm{prox}_h^\eta(\mathbf{x}) - \mathrm{prox}_h^\eta(\mathbf{y})\| \le \|\mathbf{x} - \mathbf{y}\|, \tag{8}$$

while for nonconvex functions this may not hold everywhere.

The proximal map is the key component of the proximal gradient algorithm Fukushima and Mine (1981) (a.k.a. forward-backward splitting):

$$\forall\, t = 0, 1, \ldots, \quad \mathbf{x}(t+1) = \mathrm{prox}_g^\eta(\mathbf{x}(t) - \eta \nabla f(\mathbf{x}(t))), \tag{9}$$

where ∇f is the (sub)gradient of f, and η is a suitable step size (that may change with t). It is known that when f is convex with L-Lipschitz continuous gradient and $0 < \eta < 2/L$, then $F_t := f(\mathbf{x}(t)) + g(\mathbf{x}(t))$ converges to the minimum at the rate $O(1/t)$ and $\mathbf{x}(t)$ converges to some minimizer $\mathbf{x}^*$. Accelerated versions Beck and Teboulle (2009); Nesterov (2013) where F_t converges at the faster rate $O(1/t^2)$ are also well-known. Recently, Bolte et al. (2014) proved that $\mathbf{x}(t)$ converges to a critical point even for nonconvex f and nonconvex and nonsmooth g as long as together they satisfy a certain KŁ inequality.

3. Formulation of m-PAPG

Recall the composite minimization problem:

$$\min_{\mathbf{x} \in \mathbb{R}^d} F(\mathbf{x}), \quad \text{where} \quad F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}). \tag{P}$$

We are interested in the case where d is so large that implementing the proximal gradient algorithm (9) on a single machine is no longer feasible, hence distributed computation is necessary.

We consider a **model** parallel system with p machines in total. The machines are fully connected and can communicate with each other. Decompose the d model parameters into p disjoint groups. Formally, consider the decomposition $\mathbb{R}^d = \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \times \cdots \times \mathbb{R}^{d_p}$, and denote x_i and $\nabla_i f(\mathbf{x}) : \mathbb{R}^d \to \mathbb{R}^{d_i}$ as the i-th component of $\mathbf{x}$ and $\nabla f(\mathbf{x})$, respectively. Clearly, $\mathbf{x} = (x_1, x_2, \ldots, x_p)$ and $\nabla f = (\nabla_1 f, \nabla_2 f, \cdots, \nabla_p f)$. The i-th machine is responsible for updating the component $x_i \in \mathbb{R}^{d_i}$, and for the purpose of evaluating the partial gradient $\nabla_i f(\mathbf{x})$ we assume the i-th machine also has access to a local, full model parameter $\mathbf{x}^i \in \mathbb{R}^d$. The last assumption is made only to simplify our presentation; it can be removed for many machine learning problems, see for instance Richtárik and Takáč (2014); Zhou et al. (2016).

We make the following standard assumptions regarding problem (P):

Assumption 1 (Bounded Below) *The function $F = f + g$ is bounded below.*

Assumption 2 (Smooth) *The function f is L-smooth, i.e.,*

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ \|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\| \le L\|\mathbf{x} - \mathbf{y}\|. \tag{10}$$

Assumption 3 (Separable) *The function g is closed and separable, i.e., $g(\mathbf{x}) = \sum_{i=1}^p g_i(x_i)$.*

Assumption 1 simply allows us to have a finite minimum value and is usually satisfied in practice. The smoothness assumption is critical in two aspects: (1) It allows us to upper bound f by its quadratic expansion at the current iterate—a standard step in the convergence proof of gradient type algorithms:

$$\forall\, \mathbf{x}, \mathbf{y} \in \mathbb{R}^d,\ f(\mathbf{x}) \le f(\mathbf{y}) + \langle \mathbf{x} - \mathbf{y}, \nabla f(\mathbf{y}) \rangle + \tfrac{L}{2}\|\mathbf{x} - \mathbf{y}\|^2. \tag{11}$$

(2) It allows us to bound the inconsistencies in different machines due to asynchronous updates, see Theorem 4 below. The separable assumption makes model parallelism interesting and feasible, and is satisfied by many popular regularizers. Popular examples include vector norms such as ℓ_0, ℓ_1, $\ell_{1,2}$ (i.e., group norm), ℓ_2^2, elastic net, and matrix norms such as Frobenious norm, etc. We remark that both Assumption 2 and Assumption 3 can be relaxed using techniques in Beck and Teboulle (2012) and Yu et al. (2015), respectively. For brevity we do not pursue these extensions here. Note that we do *not* assume convexity on either f or g, and g need not even be continuous.

We now specify the m-PAPG algorithm for solving (P) under model parallelism and the PAP protocol. The separable assumption on g implies that

$$\mathrm{prox}_g^\eta(\mathbf{x}) = \big(\mathrm{prox}_{g_1}^\eta(x_1),\ \ldots\ ,\ \mathrm{prox}_{g_p}^\eta(x_p)\big). \tag{12}$$

Then, the update on machine i is defined as:

$$x_i \leftarrow \mathrm{prox}_{g_i}^\eta(x_i - \eta \nabla_i f(\mathbf{x}^i)). \tag{13}$$

That is, machine i computes a partial gradient mapping Nesterov (2013) w.r.t. the i-th component using the local component x_i and the local full model $\mathbf{x}^i$. To define the latter, consider a global clock shared by all machines and denote T_i as the set of active clocks when machine i performs an update. Note that the global clock is introduced solely for the purpose of our analysis, and the machines need not maintain it in a practical implementation. Denote $\tau_j^i(t)$ as the iteration of the block model x_j that is accessed by machine i at its t-th iteration. Then, the t-th iteration on machine i can be formally written as:

$$\begin{cases} \forall i,\ x_i(t+1) = \begin{cases} x_i(t), & t \notin T_i \\ \mathrm{prox}_{g_i}^\eta(x_i(t) - \eta \nabla_i f(\mathbf{x}^i(t))), & t \in T_i \end{cases}, \\ \text{(local)} \quad \mathbf{x}^i(t) = (x_1(\tau_1^i(t)),\ \ldots\ ,\ x_p(\tau_p^i(t))), \\ \text{(global)} \quad \mathbf{x}(t) = (x_1(t),\ \ldots\ ,\ x_p(t)). \end{cases} \tag{m-PAPG}$$

That is, machine i only performs its update operator at its active clocks. The local full model $\mathbf{x}^i(t)$ assembles all components from other machines, and is possibly a delayed version of the global model $\mathbf{x}(t)$, which assembles the most up-to-date component in each machine. Note that the global model is introduced for our analysis, and is not accessible in a real implementation. More specifically, $\tau_j^i(t) \le t$ models the communication delay among machines: when machine i conducts its t-th update it only has access to $x_j(\tau_j^i(t))$, a delayed version of the component $x_j(t)$ that is received by the i-th machine from the j-th machine.

We refer to the above algorithm as m-PAPG (for **m**odel parallel, **P**artially **A**synchronous, **P**roximal **G**radient).

In a practical distributed system, communication among machines is much slower than local computations, and the performance of a *synchronous* system is often bottlenecked at the *slowest* machine, due to the need of synchronization in every step. The delays $\tau_j^i(t)$ and active clocks T_i that we introduced in m-PAPG aim to address such issues. For our convergence proofs, we need the following assumptions:

Assumption 4 (Bounded Delay) $\exists s \in \mathbb{N},\ \forall i, \forall j, \forall t,\ 0 \le t - \tau_j^i(t) \le s,\ \tau_i^i(t) \equiv t.$

Assumption 5 (Frequent Update) $\exists s \in \mathbb{N},\ \forall i, \forall t, T_i \cap \{t, t+1, \cdots, t+s\} \neq \emptyset.$

Intuitively, Assumption 4 guarantees the information that machine i gathered from other machines at the t-th iteration are not too obsolete (bounded by at most s clocks apart). The assumption $\tau_i^i(t) \equiv t$ is natural since the i-th worker machine is maintaining x_i hence would always have the latest copy. Assumption 5 requires each machine to update at least once in every $s+1$ iterations, for otherwise some component x_i may not be updated at all. We remark that Assumption 4 and Assumption 5 are very natural and have been widely adopted in previous works Baudet (1978); Bertsekas and Tsitsiklis (1989); Chazan and Miranker (1969); Feyzmahdavian et al. (2014); Tseng (1991). Clearly, when $s = 0$ (i.e., no delay), m-PAPG reduces to the fully synchronous, model parallel proximal gradient algorithm.

Before closing this section, we provide a technical tool to control the inconsistency between the local models $\mathbf{x}^i(t)$ and the global model $\mathbf{x}(t)$. Recall that $(t)_+ = \max\{t, 0\}$ is the positive part of t.

Lemma 4 *Let Assumption 4 hold, then the global model* $\mathbf{x}(t)$ *and the local models* $\{\mathbf{x}^i(t)\}_{i=1}^p$ *satisfy:*

$$\forall i = 1, \cdots, p, \quad \|\mathbf{x}(t) - \mathbf{x}^i(t)\| \le \sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|, \tag{14}$$

$$\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| \le \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|. \tag{15}$$

Proof Indeed, by the definitions in (m-PAPG):

$$\begin{aligned}
\|\mathbf{x}(t) - \mathbf{x}^i(t)\|^2 &= \sum_{j=1}^p \|x_j(t) - x_j(\tau_j^i(t))\|^2 \\
&\le \sum_{j=1}^p \left(\sum_{k=\tau_j^i(t)}^{t-1} \|x_j(k+1) - x_j(k)\| \right)^2 \\
&\le \sum_{j=1}^p \left(\sum_{k=(t-s)_+}^{t-1} \|x_j(k+1) - x_j(k)\| \right)^2 \\
&= \sum_{j=1}^p \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \|x_j(k+1) - x_j(k)\| \|x_j(k'+1) - x_j(k')\| \\
&= \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \sum_{j=1}^p \|x_j(k+1) - x_j(k)\| \|x_j(k'+1) - x_j(k')\| \\
&\le \sum_{k=(t-s)_+}^{t-1} \sum_{k'=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \|\mathbf{x}(k'+1) - \mathbf{x}(k')\| \\
&= \left(\sum_{k=(t-s)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \right)^2,
\end{aligned}$$

where the first inequality is due to the triangle inequality; the second inequality is due to Assumption 4; and the last inequality follows from the Cauchy-Schwarz inequality.

Similarly,

$$\begin{aligned}
\|\mathbf{x}^i(t) - \mathbf{x}^i(t+1)\|^2 &= \sum_{j=1}^p \|x_j(\tau_j^i(t)) - x_j(\tau_j^i(t+1))\|^2 \\
&\le \sum_{j=1}^p \left(\sum_{k=\tau_j^i(t)}^{\tau_j^i(t+1)-1} \|x_j(k+1) - x_j(k)\| \right)^2 \\
&\le \sum_{j=1}^p \left(\sum_{k=(t-s)_+}^{t} \|x_j(k+1) - x_j(k)\| \right)^2,
\end{aligned}$$

and the rest of the proof is completely similar to the previous case. ■

4. Characterizing the limit points

In this section, we characterize the convergence property of the sequences generated by m-PAPG under very general conditions. Recall from Assumption 2 that ∇f is L-Lipschitz continuous. Our first result is as follows:

Theorem 5 *Let Assumptions 1 to 5 hold. If the step size* $\eta \in \left(0, \frac{1}{L(1+2\sqrt{ps})}\right)$*, then the sequence generated by m-PAPG is square summable, i.e.*

$$\sum_{t=0}^{\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 < \infty. \tag{16}$$

In particular, $\lim_{t\to\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| = 0$ *and* $\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^i(t)\| = 0.$

Remark 6 *Our bound on the step size* η *is natural: If* $s = 0$*, i.e., there is no asynchronism then we recover the standard step size rule* $\eta < 1/L$ *(we can increase* η *by another factor of 2, had convexity on* g *been assumed). As staleness* s *increases, we need a smaller step size to*

"damp" the system to still ensure convergence. The factor $\sqrt{p}$ is another measurement of the degree of "dependency" among worker machines: Indeed, we can reduce $\sqrt{p}$ to $\sqrt{\sum_i L_i^2}/L$, where L_i is the Lipschitz constant of $\nabla_i f$ (cf. (21)).

Proof The last claim follows immediately from eq. (16) and eq. (14), so we only need to prove (16).

Consider machine i and any $t \in T_i$. Combining eq. (13) with eq. (m-PAPG) gives

$$x_i(t+1) = \operatorname{prox}_{g_i}^{\eta}(x_i(t) - \eta \nabla_i f(\mathbf{x}^i(t))). \tag{17}$$

Then, from Definition 3 of the proximal map we have for all $z \in \mathbb{R}^{d_i}$:

$$\begin{aligned} g_i(x_i(t+1)) + \frac{1}{2\eta}\|x_i(t+1) - x_i(t) + \eta \nabla_i f(\mathbf{x}^i(t))\|^2 \\ \leq g_i(z) + \frac{1}{2\eta}\left\|z - x_i(t) + \eta \nabla_i f(\mathbf{x}^i(t))\right\|^2. \end{aligned} \tag{18}$$

Set $z = x_i(t)$ and simplify, we obtain:

$$\begin{aligned} g_i(x_i(t+1)) - g_i(x_i(t)) \\ \leq -\frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 - \left\langle \nabla_i f(\mathbf{x}^i(t)), x_i(t+1) - x_i(t)\right\rangle. \end{aligned} \tag{19}$$

Note that if $t \notin T_i$, then $x_i(t+1) = x_i(t)$ and eq. (19) still holds. On the other hand, Assumption 2 implies that for all t (cf. (11)):

$$f(\mathbf{x}(t+1)) - f(\mathbf{x}(t)) \leq \langle \mathbf{x}(t+1) - \mathbf{x}(t), \nabla f(\mathbf{x}(t))\rangle + \frac{L}{2}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2. \tag{20}$$

Adding up eq. (20) and eq. (19) (for all i) and recall $F = f + \sum_i g_i$, we have

$$\begin{aligned} &F(\mathbf{x}(t+1)) - F(\mathbf{x}(t)) - \tfrac{1}{2}(L - 1/\eta)\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \\ &\leq \sum_{i=1}^{p}\left\langle x_i(t+1) - x_i(t), \nabla_i f(\mathbf{x}(t)) - \nabla_i f(\mathbf{x}^i(t))\right\rangle \\ &\leq \sum_{i=1}^{p}\|x_i(t+1) - x_i(t)\| \cdot \|\nabla_i f(\mathbf{x}(t)) - \nabla_i f(\mathbf{x}^i(t))\| \\ &\overset{\text{(i)}}{\leq} \sum_{i=1}^{p}\|x_i(t+1) - x_i(t)\| \cdot L\|\mathbf{x}(t) - \mathbf{x}^i(t)\| \qquad (21) \\ &\overset{\text{(ii)}}{\leq} L \cdot \sum_{i=1}^{p}\|x_i(t+1) - x_i(t)\| \cdot \sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\overset{\text{(iii)}}{\leq} \sqrt{p}L\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1) - \mathbf{x}(k)\| \qquad (22) \\ &\overset{\text{(iv)}}{\leq} \frac{\sqrt{p}L}{2}\sum_{k=(t-s)_+}^{t-1}\left[\|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2 + \|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2\right] \\ &\leq \frac{\sqrt{p}Ls}{2}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 + \frac{\sqrt{p}L}{2}\sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2, \qquad (23) \end{aligned}$$

where (i) is due to the L-Lipschitz continuity of ∇f, (ii) follows from eq. (14), (iii) is the Cauchy-Schwarz inequality, and (iv) follows from the elementary inequality $ab \leq \frac{a^2+b^2}{2}$. Summing the above inequality over t from 0 to $m-1$ and rearranging we obtain

$$\begin{aligned} F(\mathbf{x}(m)) - F(\mathbf{x}(0)) &\leq \frac{1}{2}(L + \sqrt{p}Ls - 1/\eta)\sum_{t=0}^{m-1}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \\ &\quad + \frac{L}{2}\sum_{t=0}^{m-1}\sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2 \\ &\leq \frac{1}{2}(L + 2\sqrt{p}Ls - 1/\eta)\sum_{t=0}^{m-1}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2. \end{aligned}$$

Therefore, if we choose $0 < \eta < \frac{1}{L(1+2\sqrt{p}s)}$, then let $m \to \infty$ we deduce

$$\sum_{t=0}^{\infty}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 \leq \frac{2}{1/\eta - L - 2\sqrt{p}Ls}[F(\mathbf{x}(0)) - \inf_{\mathbf{z}} F(\mathbf{z})]. \tag{24}$$

By Assumption 1, F is bounded from below, hence the right-hand side is finite. ■

The first assertion of the above theorem states that the global sequence $\mathbf{x}(t)$ has square summable successive differences, while the second assertion implies that both the successive difference of the global sequence and the inconsistency between the local sequences and the global sequence diminish as the number of iterations grows. These two conclusions provide a prelimenary stability guarantee for m-PAPG.

Next, we prove that the limit points (if exist) of the sequences $\mathbf{x}(t)$ and $\mathbf{x}^i(t), i = 1, \ldots, p$ coincide, and they are critical points of F. Recall that the set of critical points of the function F is denoted as $\operatorname{crit} F$.

Theorem 7 *Consider the same setting as in Theorem 5. Then, the sequences $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}, i = 1, \ldots, p$, generated by m-PAPG share the same set of limit points, which is a subset of $\operatorname{crit} F$.*

Proof It is clear from Theorem 5 that $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}, i = 1, \ldots, p$, share the same set of limit points, and we need to show that any limit point of $\{\mathbf{x}(t)\}$ is also a critical point of F.

Let $\mathbf{x}^*$ be a limit point of $\{\mathbf{x}(t)\}$. By Theorem 1 it suffices to exhibit a sequence $\mathbf{x}(k)$ satisfying[1]

$$\mathbf{x}(k) \to \mathbf{x}^*,\ F(\mathbf{x}(k)) \to F(\mathbf{x}^*),\ \mathbf{0} \leftarrow \mathbf{u}(k) \in \partial F(\mathbf{x}(k)). \tag{25}$$

1. Technically, from Theorem 1 we should have the Frechét subdifferential $\hat{\partial} F$ in eq. (25), however, a standard argument allows us to use the more convenient subdifferential (Rockafellar and Wets, 1997, Proposition 8.7).

Let us first construct the subgradient sequence $\mathbf{u}(k)$. Consider machine i and any $\hat{t} \in T_i$, the optimality condition of eq. (17) gives

$$u_i(\hat{t}+1) := -\tfrac{1}{\eta}\left[x_i(\hat{t}+1) - x_i(\hat{t}) + \eta\nabla_i f(\mathbf{x}^i(\hat{t}))\right] \in \partial g_i(x_i(\hat{t}+1)). \tag{26}$$

It then follows that

$$\begin{aligned}
&\|u_i(\hat{t}+1) + \nabla_i f(\mathbf{x}(\hat{t}+1))\| \\
&\quad\le \|u_i(\hat{t}+1) + \nabla_i f(\mathbf{x}(\hat{t}))\| + \|\nabla_i f(\mathbf{x}(\hat{t}+1)) - \nabla_i f(\mathbf{x}(\hat{t}))\| \\
&\quad\overset{\text{(i)}}{\le} \left\|\tfrac{1}{\eta}\left[x_i(\hat{t}+1) - x_i(\hat{t})\right] + \nabla_i f(\mathbf{x}^i(\hat{t})) - \nabla_i f(\mathbf{x}(\hat{t}))\right\| + L\|\mathbf{x}(\hat{t}+1) - \mathbf{x}(\hat{t})\| \\
&\quad\overset{\text{(ii)}}{\le} \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\|\mathbf{x}^i(\hat{t}) - \mathbf{x}(\hat{t})\| + L\|\mathbf{x}(\hat{t}+1) - \mathbf{x}(\hat{t})\| \\
&\quad\overset{\text{(iii)}}{\le} \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\sum_{k=(\hat{t}-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|,
\end{aligned} \tag{27}$$

where (i) and (ii) are due to the L-Lipschitz continuity of ∇f, and (iii) follows from eq. (14). Next, consider any other $t \notin T_i$ and $t \ge s$, we denote $\hat{t}$ as the *largest* element in the set $\{k \le t : k \in T_i\}$. By Assumption 5 $\hat{t}$ always exists and $t - \hat{t} \le s$. Since no update is performed on machine i at any clock in $[\hat{t}+1, t]$, we have $x_i(t+1) = x_i(\hat{t}+1)$. Thus, we can choose $u_i(t+1) = u_i(\hat{t}+1) \in \partial g_i(x_i(\hat{t}+1)) = \partial g_i(x_i(t+1))$, and obtain

$$\begin{aligned}
&\|u_i(t+1) + \nabla_i f(\mathbf{x}(t+1)) - u_i(\hat{t}+1) - \nabla_i f(\mathbf{x}(\hat{t}+1))\| \\
&\qquad= \|\nabla_i f(\mathbf{x}(t+1)) - \nabla_i f(\mathbf{x}(\hat{t}+1))\| \\
&\qquad\le \sum_{k=\hat{t}+1}^{t} \|\nabla_i f(\mathbf{x}(k+1)) - \nabla_i f(\mathbf{x}(k))\| \\
&\qquad\le \sum_{k=(t-s+1)_+}^{t} \|\nabla_i f(\mathbf{x}(k+1)) - \nabla_i f(\mathbf{x}(k))\| \\
&\qquad\le \sum_{k=(t-s+1)_+}^{t} L\|\mathbf{x}(k+1) - \mathbf{x}(k)\|.
\end{aligned} \tag{28}$$

$$\tag{29}$$

Combining the two cases in eq. (27) and eq. (29), we have for all t and all i:

$$\begin{aligned}
\|u_i(t+1) + \nabla_i f(\mathbf{x}(t+1))\| &\le \tfrac{1}{\eta}\|x_i(\hat{t}+1) - x_i(\hat{t})\| + L\sum_{k=(\hat{t}-s)_+}^{\hat{t}} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\
&\quad + L\sum_{k=(t-s+1)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\
&\le (\tfrac{1}{\eta} + 2L)\sum_{k=(t-2s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|,
\end{aligned}$$

where the last inequality uses the fact that $t - s \le \hat{t} \le t$. Observing that the right hand side of the above inequality does not depend on i, we can sum the square of the above inequality over i and further conclude that

$$\|\mathbf{u}(t+1) + \nabla f(\mathbf{x}(t+1))\| \le \sqrt{p}(\tfrac{1}{\eta} + 2L)\sum_{k=(t-2s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|, \tag{30}$$

where $\mathbf{u}(t+1) = \big(u_1(t+1),\ \ldots\ ,\ u_p(t+1)\big) \in \partial g(\mathbf{x}(t+1))$. Therefore, by eq. (30) and Theorem 5 we deduce

$$\lim_{t\to\infty} \operatorname{dist}_{\partial F(\mathbf{x}(t+1))}(\mathbf{0}) \le \lim_{t\to\infty} \|\mathbf{u}(t+1) + \nabla f(\mathbf{x}(t+1))\| = 0. \tag{31}$$

Recall that $\mathbf{x}^*$ is a limit point of $\{\mathbf{x}(t)\}$, thus there exists a subsequence $\mathbf{x}(t_m) \to \mathbf{x}^*$. Next we verify the function value convergence in eq. (25). The challenge here is that the component function g is only closed, hence may not be continuous. For any $t \in T_i$, applying eq. (18) with $z = x_i^*$ and rearranging gives

$$\begin{aligned}
g_i(x_i(t+1)) &\le g_i(x_i^*) + \frac{1}{2\eta}\|x_i^* - x_i(t)\|^2 - \frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 \\
&\quad + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^i(t))\rangle \\
&= g_i(x_i^*) + \frac{1}{2\eta}\|x_i^* - x_i(t)\|^2 - \frac{1}{2\eta}\|x_i(t+1) - x_i(t)\|^2 \\
&\quad + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^*)\rangle + \langle x_i^* - x_i(t+1), \nabla_i f(\mathbf{x}^i(t)) - \nabla_i f(\mathbf{x}^*)\rangle.
\end{aligned} \tag{32}$$

We note that the above inequality holds only for the iterations $t \in T_i$. Next, observe that $\lim_{m\to\infty} \|\mathbf{x}(t_m) - \mathbf{x}^*\| = 0$. Since $\lim_{t\to\infty} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| = 0$, we further conclude that

$$\lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\mathbf{x}(t) - \mathbf{x}^*\| = 0. \tag{33}$$

Moreover, note that $\lim_{t\to\infty} \|\mathbf{x}(t) - \mathbf{x}^i(t)\| = 0$. Then, the above equation further implies that

$$\begin{aligned}
&\lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\nabla_i f(\mathbf{x}^i(t)) - \nabla_i f(\mathbf{x}^*)\| \\
&\le L \lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \|\mathbf{x}^* - \mathbf{x}^i(t)\| \\
&\le L \lim_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} \left[\|\mathbf{x}^* - \mathbf{x}(t)\| + \|\mathbf{x}(t) - \mathbf{x}^i(t)\|\right] = 0.
\end{aligned} \tag{34}$$

By Assumption 5, $[t_m - s, t_m + s] \cap T_i \ne \emptyset$ for all i. We can now take the limsup on both sides of eq. (32) and utilize eqs. (33) and (34) to obtain that

$$\limsup_{m\to\infty} \max_{t\in[t_m - s, t_m + s]\cap T_i} g_i(x_i(t+1)) \le g_i(x_i^*). \tag{35}$$

Denote $\hat{t}_m \in T_i$ as the largest element such that $\hat{t}_m \le t_m$. Note that $t_m - s \le \hat{t}_m$ due to the constraint on the maximum delay. It then follows that

$$\max_{t\in[t_m, t_m + s]} g_i(x_i(t+1)) = \max_{t\in[\hat{t}_m, t_m + s]\cap T_i} g_i(x_i(t+1)) \le \max_{t\in[t_m - s, t_m + s]\cap T_i} g_i(x_i(t+1)),$$

where the first equality is due to the fact that no update is performed during $[\hat{t}_m, t_m]$ and machine i updates only at its active clocks T_i, and the second inequality uses the fact that $\hat{t}_m \geq t_m - s$. Hence, we further obtain from (35) that

$$\limsup_{m\to\infty} \max_{t\in[t_m, t_m+s]} g_i(x_i(t+1)) \leq g_i(x_i^*). \tag{36}$$

To complete the proof, choose any $k_m \in [t_m, t_m + s]$. Since $\mathbf{x}(t_m) \to \mathbf{x}^*$, Theorem 5 implies that

$$\mathbf{x}(k_m) \to \mathbf{x}^*. \tag{37}$$

From eq. (36) we know for all i, $\limsup_{m\to\infty} g_i(x_i(k_m)) \leq g_i(x_i^*)$. On the other hand, it follows from the closedness of the function g_i (cf. Assumption 3) that $\liminf_{m\to\infty} g_i(x_i(k_m)) \geq g_i(x_i^*)$, thus in fact $\lim_{m\to\infty} g_i(x_i(k_m)) = g_i(x_i^*)$. Since f is continuous, we know

$$\lim_{m\to\infty} F(\mathbf{x}(k_m)) = \lim_{m\to\infty} f(\mathbf{x}(k_m)) + \sum_i g_i(x_i(k_m)) = F(\mathbf{x}^*). \tag{38}$$

Combining eq. (31), eq. (37) and eq. (38) we know from Theorem 1 that $\mathbf{x}^* \in \operatorname{crit} F$. ■

Theorem 7 further justifies m-PAPG by showing that any limit point it produces is necessarily a critical point. Of course, for convex functions any critical point is a global minimizer. The closest result to Theorem 5 and Theorem 7 we are aware of is (Bertsekas and Tsitsiklis, 1989, Proposition 7.5.3), where essentially the same conclusion was reached but under the much more restrictive assumption that g is an indicator function of a product *convex* set. Thus, our result is new even when g is a convex function such as the ℓ_1 norm that is widely used to promote sparsity. Furthermore, we allow g to be any closed separable function (convex or not), covering the many recent nonconvex regularization functions in machine learning and statistics (see e.g. Fan and Li (2001); Mazumder et al. (2011); Zhang (2010); Zhang and Zhang (2012)). We also note that the proof of Theorem 7 (for nonconvex g) involves significantly new ideas beyond those of Bertsekas and Tsitsiklis (1989).

We note that the existence of limit points can be guaranteed, for instance, if $\{\mathbf{x}(t)\}$ is bounded or the sublevel set $\{\mathbf{x} \mid F(\mathbf{x}) \leq \alpha\}$ is bounded for all $\alpha \in \mathbb{R}$. However, we have yet to prove that the sequence $\{\mathbf{x}(t)\}$ generated by m-PAPG does converge to one of the critical points, and we fill this gap under two complementary sets of assumptions on the objective function in Sections 5 and 6, respectively.

5. Convergence under Error Bound

In this section we prove that the global sequence $\{\mathbf{x}(t)\}$ produced by m-PAPG converges periodically linearly to a global minimizer, by assuming an error bound condition on the objective function in (P) and a convexity assumption that serves to simplify the presentation:

Assumption 6 (Convex) *The functions f and g in* (P) *are convex.*

Note that for convex functions g the proximal mapping $\operatorname{prox}_g^\eta$ is single valued for any $\eta > 0$. The error bound condition we need is as follows:

Assumption 7 (Error Bound) *For every $\alpha > 0$, there exist $\delta, \kappa > 0$ such that for all $\mathbf{x} \in \mathbb{R}^d$ with $f(\mathbf{x}) \leq \alpha$ and $\|\mathbf{x} - \operatorname{prox}_g(\mathbf{x} - \nabla f(\mathbf{x}))\| \leq \delta$,*

$$\operatorname{dist}_{\operatorname{crit} F}(\mathbf{x}) \leq \kappa \|\mathbf{x} - \operatorname{prox}_g(\mathbf{x} - \nabla f(\mathbf{x}))\|, \tag{39}$$

where recall that $\operatorname{crit} F$ *is the set of critical points of F.*

Equation (39) is a proximal extension of the Luo-Tseng error bound Luo and Tseng (1993) where g is the indicator function of a closed convex set. A prototypic convex function F satisfying (39) is the following:

$$F(\mathbf{x}) = f(A\mathbf{x}) + g(\mathbf{x}), \tag{40}$$

where f is strongly convex (i.e., $f - \frac{\mu}{2}\|\cdot\|^2$ is convex for some $\mu > 0$), A is a linear map, and g is either an indicator function of a convex set Luo and Tseng (1993) or the ℓ_p norm for $p \in [1, 2] \cup \{\infty\}$ Zhou et al. (2015). Many machine learning formulations such as Lasso and sparse logistic regression fit into this form. In fact, for convex functions F taking such form, the error bound condition in eq. (39) is recently shown to be equivalent to the following conditions Drusvyatskiy and Lewis (2016); Zhang (2016):

$$\begin{aligned} \text{Restricted strong convexity}: &\ \langle \mathbf{x} - \operatorname{prox}_g(\mathbf{x}), \mathbf{x} - \operatorname{proj}_{\operatorname{crit} F}(\mathbf{x})\rangle \geq \mu \cdot \operatorname{dist}^2_{\operatorname{crit} F}(\mathbf{x}), \\ \text{Quadratic growth}: &\ F(\mathbf{x}) - F^* \geq \mu \cdot \operatorname{dist}^2_{\operatorname{crit} F}(\mathbf{x}), \end{aligned}$$

where F^* is the minimum value of F and $\mu > 0$ is a constant. In general, the error bound condition in eq. (39) is not exclusive to convex functions. For instance, it holds for $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{x}\|^2$ and any function g that has a unique global minimizer at 0 (such as the cardinality function $g(\mathbf{x}) = \|\mathbf{x}\|_0$). However, it is often quite challenging to establish the error bound condition for a large family of nonconvex functions.

We define the following nonnegative quantities that measure the progress of m-PAPG:

$$A(t) := F(\mathbf{x}(t)) - F^*, \ F^* := \inf_{\mathbf{x}} F(\mathbf{x}), \tag{41}$$

$$B(t) := \sum_{k=(t-s-1)_+}^{t-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2, \tag{42}$$

In the following key lemma we relate the gap quantities defined above inductively.

Lemma 8 *Let Assumptions 1 to 7 hold. Then, we have*

$$\begin{aligned} &A(t+s+1) \leq A(t) - \tfrac{1}{2}(\tfrac{1}{\eta} - L - 2sL\sqrt{p})B(t+s+1) + \tfrac{1}{2}sL\sqrt{p}B(t) \\ &0 \leq A(t+s+1) \leq a_\eta B(t+s+1) + bB(t), \end{aligned}$$

where a_η and b are given in (53) *below.*

Proof The first inequality is obtained by summing the inequality eq. (23) over $t, t+1, \cdots, t+s$. So we need only prove the second inequality.

Let us introduce some notations to simplify the proof. For each machine i let t_i be the largest clock in $[t, t+s] \cap T_i$, and denote

$$\mathbf{z} = (x_1(t_1), \ \ldots \ , x_p(t_p)) \tag{43}$$

$$\mathbf{z}^+ = (x_1(t_1+1), \ \ldots \ , x_p(t_p+1)) = (x_1(t+s+1), \ \ldots \ , x_p(t+s+1)), \tag{44}$$

where the last equality is due to the maximality of each t_i. From the optimality condition of the proximal map $z_i^+ = \mathrm{prox}_{g_i}^{\eta}(z_i - \eta \nabla_i f(\mathbf{x}^i(t_i)))$ we deduce

$$\eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \in \partial g_i(z_i^+). \tag{45}$$

Since the gradient of f is L-Lipschitz continuous and the function g is convex, we obtain

$$f(\mathbf{z}^+) - f(\bar{\mathbf{z}}) \le \sum_{i=1}^p \langle z_i^+ - \bar{z}_i, \nabla_i f(\bar{\mathbf{z}}) \rangle + \frac{L}{2}\|\mathbf{z}^+ - \bar{\mathbf{z}}\|^2,$$

$$g(\mathbf{z}^+) - g(\bar{\mathbf{z}}) \le \sum_{i=1}^p \langle z_i^+ - \bar{z}_i, \eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \rangle,$$

where we define $\bar{\mathbf{z}} := \mathrm{proj}_{\mathrm{crit}\, F}(\mathbf{z})$, i.e., the projection of $\mathbf{z}$ onto the set of critical points of F, and the last inequality follows from eq. (45). Adding up the above two inequalities we obtain

$$\begin{aligned}
F(\mathbf{z}^+) - F^* - \tfrac{L}{2}\|\mathbf{z}^+ - \bar{\mathbf{z}}\|^2 &\le \sum_{i=1}^p \langle z_i^+ - \bar{z}_i, \nabla_i f(\bar{\mathbf{z}}) + \eta^{-1}(z_i - z_i^+) - \nabla_i f(\mathbf{x}^i(t_i)) \rangle \\
&\overset{\text{(i)}}{\le} \sum_{i=1}^p [\|z_i^+ - z_i\| + \|z_i - \bar{z}_i\|][\|\nabla_i f(\mathbf{x}^i(t_i)) - \nabla_i f(\bar{\mathbf{z}})\| + \eta^{-1}\|z_i - z_i^+\|] \\
&\overset{\text{(ii)}}{\le} \sum_{i=1}^p 4\left[\|z_i^+ - z_i\|^2 + \|z_i - \bar{z}_i\|^2 + \eta^{-2}\|z_i^+ - z_i\|^2 + \|\nabla_i f(\mathbf{x}^i(t_i)) - \nabla_i f(\bar{\mathbf{z}})\|^2\right] \\
&\le 4\left[\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (1+\eta^{-2})\|\mathbf{z}^+ - \mathbf{z}\|^2 + \sum_{i=1}^p L^2\|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2\right],
\end{aligned}$$

where (i) is due to the Cauchy-Schwarz inequality and the triangle inequality, (ii) is due to the elementary inequality $(a+b)(c+d) \le 4(a^2+b^2+c^2+d^2)$, and the last inequality is due to the L-Lipschitz continuity of ∇f. Using again the triangle inequality we obtain from the above inequality that

$$\begin{aligned}
F(\mathbf{z}^+) - F^* &\le (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})\|\mathbf{z}^+ - \mathbf{z}\|^2 + 4L^2\sum_{i=1}^p \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2 \\
&\overset{\text{(i)}}{=} (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + \sum_{i=1}^p [(L+4+\tfrac{4}{\eta^2})\|x_i(t_i+1) - x_i(t_i)\|^2 + 4L^2\|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2], \\
&\overset{\text{(ii)}}{\le} (L+4)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 4L^2\sum_{i=1}^p \|\mathbf{x}^i(t_i) - \bar{\mathbf{z}}\|^2,
\end{aligned} \tag{46}$$

$$\le (L+4+8L^2p)\|\bar{\mathbf{z}} - \mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 8L^2\sum_{i=1}^p \|\mathbf{x}^i(t_i) - \mathbf{z}\|^2, \tag{47}$$

where (i) is due to our definition of $\mathbf{z}$ and $\mathbf{z}^+$ in (43) and (44), and (ii) is due to the fact that $t_i \in [t, t+s]$ for all i.

We next bound the terms $\|\bar{\mathbf{z}} - \mathbf{z}\|^2$ and $\|\mathbf{x}^i(t_i) - \mathbf{z}\|^2$. We recall that $\mathbf{x}^i(t_i)$ corresponds to the local model on machine i at the iteration t_i. Since $t_i \in T_i$, the update rule for the i-th machine implies that

$$\begin{aligned}
\|x_i(t_i+1) - x_i(t_i)\| &= \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{x}^i(t_i))) - x_i(t_i)\| \\
&\ge \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z})) - x_i(t_i)\| \\
&\quad - \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{x}^i(t_i))) - \mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z}))\| \\
&\overset{\text{(i)}}{\ge} \|\mathrm{prox}_{g_i}^{\eta}(x_i(t_i) - \eta\nabla_i f(\mathbf{z})) - x_i(t_i)\| - \eta L\|\mathbf{z} - \mathbf{x}^i(t_i)\|,
\end{aligned}$$

where (i) follows from the non-expansiveness of prox_g^{η} (recall that g is convex) and the L-Lipschitz continuity of ∇f. Rearranging the above inequality and summing over all i, we obtain

$$\begin{aligned}
\|\mathrm{prox}_g^{\eta}(\mathbf{z} - \eta\nabla f(\mathbf{z})) - \mathbf{z}\|^2 &\le \sum_{i=1}^p \Big[\|x_i(t_i+1) - x_i(t_i)\| + \eta L\|\mathbf{z} - \mathbf{x}^i(t_i)\|\Big]^2 \\
&\le 2\sum_{i=1}^p \Big[\|x_i(t_i+1) - x_i(t_i)\|^2 + \eta^2 L^2\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2\Big].
\end{aligned} \tag{48}$$

The last term $\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2$ can be further bounded as follows:

$$\begin{aligned}
\|\mathbf{z} - \mathbf{x}^i(t_i)\|^2 &= \sum_{j=1}^p \|x_j(t_j) - x_j(\tau_j^i(t_i))\|^2 \\
&= \sum_{j=1}^p \Big\| \sum_{k=\min\{t_j, \tau_j^i(t_i)\}}^{\max\{t_j, \tau_j^i(t_i)\}-1} x_j(k+1) - x_j(k) \Big\|^2 \\
&\le \sum_{j=1}^p \Big[\sum_{k=\min\{t_j, \tau_j^i(t_i)\}}^{\max\{t_j, \tau_j^i(t_i)\}-1} \|x_j(k+1) - x_j(k)\| \Big]^2 \\
&\overset{\text{(i)}}{\le} \sum_{j=1}^p 2s \sum_{k=t-s}^{t+s-1} \|x_j(k+1) - x_j(k)\|^2 \\
&= 2s \sum_{k=t-s}^{t+s-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|^2 \\
&\le 2s[B(t) + B(t+s+1)],
\end{aligned} \tag{49}$$

where (i) is due to the fact that $t_j \in [t, t+s]$ and $\tau_j^i(t_i) \in [t-s, t+s]$. Combining (48) and (49) we obtain

$$\|\mathrm{prox}_g^{\eta}(\mathbf{z} - \eta\nabla f(\mathbf{z})) - \mathbf{z}\|^2 \le 2B(t+s+1) + 4ps\eta^2 L^2[B(t) + B(t+s+1)]. \tag{50}$$

Thanks to Theorem 5, we know for t sufficiently large, $\|\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))-\mathbf{z}\| \le \eta\delta$. Since the function $\eta \mapsto \frac{1}{\eta}\|\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))-\mathbf{z}\|$ is monotonically decreasing Sra (2012), we can apply the error bound condition in Assumption 7 for $\eta < 1$ and t sufficiently large, and obtain

$$\|\bar{\mathbf{z}}-\mathbf{z}\|^2 \le \kappa\|\mathbf{z}-\text{prox}_g(\mathbf{z}-\nabla f(\mathbf{z}))\|^2 \le \kappa\eta^{-2}\|\mathbf{z}-\text{prox}_g^\eta(\mathbf{z}-\eta\nabla f(\mathbf{z}))\|^2. \tag{51}$$

Finally, combining (46), (49), (50) and (51) we arrive at:

$$\begin{aligned} F(\mathbf{x}(t+s+1)) - F^* &= F(\mathbf{z}^+) - F^* \\ &\le (L+4+8L^2p)\|\bar{\mathbf{z}}-\mathbf{z}\|^2 + (L+4+\tfrac{4}{\eta^2})B(t+s+1) + 8L^2\sum_{i=1}^p \|\mathbf{x}^i(t_i)-\mathbf{z}\|^2, \\ &\le a_\eta B(t+s+1) + bB(t), \end{aligned} \tag{52}$$

where the coefficients are

$$a_\eta = L + 4 + 16psL^2 + 4ps\kappa L^2(L+4+8L^2p) + \tfrac{2}{\eta^2}(2+4\kappa+\kappa L), \tag{53}$$

$$b = 16psL^2 + 4ps\kappa L^2(L+4+8L^2p). \tag{54}$$

■

Theorem 8 improves the analysis of Tseng (1991) in three aspects: (1) it is shorter and simpler; (2) it allows any convex function g; and (3) the leading coefficient for $B(t)$ is reduced from $O(1/\eta)$ to $O(1)$. The two recursive relations in Lemma 8, as shown in (Tseng, 1991, Lemma 4.5), easily imply the following convergence guarantee.

Theorem 9 *Let Assumptions 1 to 7 hold. Then, there exists some $\eta_0 > 0$ such that if $0 < \eta < \eta_0$, then the sequences $\{A(t), B(t)\}$ generated by m-PAPG satisfy for all $r = 0, 1, 2, \cdots$*

$$A(r(s+1)) \le C_1(1-\gamma\eta)^r, \;\; B(r(s+1)) \le C_2(1-\gamma\eta)^r, \tag{55}$$

where $C_1, C_2, \gamma < 1/\eta$ are positive constants.

Hence, the gaps $A(t)$ and $B(t)$ that measure the progress of m-PAPG decrease by a constant factor $(1-\gamma\eta)$ for every $s+1$ steps, which makes intuitive sense since in the worst case each worker machine only performs one update in every $s+1$ steps. In other words, $(s+1)$ is the natural time scale for measuring progress here. Note that since $\|\mathbf{x}(t+s+1)-\mathbf{x}(t)\|^2 \le (s+1)B(t+s+1)$, it follows easily that the global sequence $\mathbf{x}(t)$ and consequently also the local sequences $\{\mathbf{x}^i(t)\}$ all converge to the same limit point in crit F at a $(s+1)$-periodically linear rate.

6. Convergence with KŁ inequality

The error bound condition considered in the previous section is not easy to verify in general. It has been discovered recently that the error bound condition is equivalent to other notions in optimization that can be verified in alternative ways Drusvyatskiy and Lewis (2016);

Zhang (2016), see e.g. (40). However, for nonconvex functions, sometimes even the simple ones, it remains a challenging task to verify if the error bound condition holds. This failure motivates us to investigate another property, the Kurdyka-Łojasiewicz (KŁ) inequality, that has been shown to be quite effective in dealing with nonconvex functions.

Definition 10 (KŁ property, (Bolte et al., 2014, Lemma 6)) *Let $\Omega \subset \text{dom}h$ be a compact set on which the function h is a constant. We say that h satisfies the KŁ property if there exist $\varepsilon, \lambda > 0$ such that for all $\bar{\mathbf{x}} \in \Omega$ and all $\mathbf{x} \in \{\mathbf{z} \in \mathbb{R}^d : \text{dist}_\Omega(\mathbf{z}) < \varepsilon\} \cap [\mathbf{z} : h(\bar{\mathbf{x}}) < h(\mathbf{z}) < h(\bar{\mathbf{x}}) + \lambda]$, it holds that*

$$\varphi'(h(\mathbf{x}) - h(\bar{\mathbf{x}})) \cdot \text{dist}_{\partial h(\mathbf{x})}(\mathbf{0}) \ge 1, \tag{56}$$

where the function $\varphi : [0, \lambda) \to \mathbb{R}_+, 0 \mapsto 0$, is continuous, concave, and has continuous and positive derivative φ' on $(0, \lambda)$.

The KŁ inequality in eq. (56) is an important tool to bound the trajectory length of a dynamical system (see Bolte et al. (2010); Kurdyka (1998) and the references therein for some historic developments). It has recently been used to analyze discrete-time algorithms in Absil et al. (2005) and proximal algorithms in Attouch and Bolte (2009); Attouch et al. (2010); Bolte et al. (2014). As we shall see, the function φ will serve as a Lyapunov potential function. Quite conveniently, most practical functions, in particular, the quasi-norm $\|\cdot\|_p$ for positive rational p, as well as convex functions with certain growth conditions, are KŁ. For a more detailed discussion of KŁ functions, including many familiar examples, see (Bolte et al., 2014, Section 5) and (Attouch et al., 2010, Section 4).

Following the recipe in Bolte et al. (2014), we need the following assumption to guarantee the algorithm is making *sufficient* progress:

Assumption 8 (Sufficient decrease) *There exists $\alpha > 0$ such that for all large t,*

$$F(\mathbf{x}(t+1)) \le F(\mathbf{x}(t)) - \alpha\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2. \tag{57}$$

The sufficient decrease assumption is automatically satisfied in many descent algorithms, e.g., the proximal gradient algorithm. However, in the partially asynchronous parallel (PAP) setting, it is highly nontrivial to satisfy the sufficient decrease assumption because of the complication due to communication delays and update skips. Note also that none of the worker machines actually has access to the global sequence $\mathbf{x}(t)$, so even verifying the sufficient decrease property is not trivial. To simplify the presentation, we first analyze the performance of m-PAPG using the KŁ inequality and taking the sufficient decrease property for granted, and later we we will give some verifiable conditions to justify this simplification.

Our first result in this section strengthens the convergence properties in Theorems 5 and 7 for m-PAPG:

Theorem 11 (Finite Length) *Let Assumptions 1 to 5 and 8 hold for m-PAPG, and let F satisfy the KL property in Theorem 10. Then, with step size $\eta \in \left(0, \frac{1}{L(1+2\sqrt{ps})}\right)$, every bounded sequence $\{\mathbf{x}(t)\}$ generated by m-PAPG satisfies*

$$\sum_{t=0}^{\infty} \|\mathbf{x}(t+1)-\mathbf{x}(t)\| < \infty, \tag{58}$$

$$\forall i = 1, \dots, p, \ \sum_{t=0}^{\infty} \|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| < \infty. \tag{59}$$

Furthermore, $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}_{i=1}^p$ converge to the same critical point of F.

Proof We first show that eq. (58) implies eq. (59). Indeed, recall from (15):

$$\|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| \le \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|.$$

Therefore, summing for $t = 0, 1, \cdots, n$ gives

$$\begin{aligned}\sum_{t=0}^{n} \|\mathbf{x}^i(t+1) - \mathbf{x}^i(t)\| &\le \sum_{t=0}^{n} \sum_{k=(t-s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\le (2s+1) \sum_{t=0}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|.\end{aligned}$$

The claim then follows by letting n tend to infinity.

By Theorem 5, the limit points of $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}_{i=1}^p$ coincide and are critical points of F. Thus, the only thing left to prove is the finite length property in eq. (58). By Assumption 8 and Assumption 1, the objective value $F(\mathbf{x}(t))$ decreases to a finite limit F^*. Since $\{\mathbf{x}(t)\}$ is assumed to be bounded, the set of its limit points Ω is nonempty and compact. Summing eq. (18) over all i and set $\mathbf{z} \in \Omega$, we obtain

$$g(\mathbf{x}(t+1)) \le g(\mathbf{z}) - \frac{1}{2\eta}\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 - \sum_{i=1}^{p} \langle \nabla_i f(\mathbf{x}^i(t)), \mathbf{x}(t+1) - \mathbf{x}(t) \rangle.$$

Note that $\mathbf{x}(t+1) - \mathbf{x}(t) \to 0$. Also, since $\{\mathbf{x}(t)\}$ is bounded and $\mathbf{x}(t) - \mathbf{x}^i(t) \to 0$ for all i, $\{\mathbf{x}^i(t)\}_{i=1}^p$ are all bounded. we then take limsup on both sides and obtain that $\limsup_{t\to\infty} g(\mathbf{x}(t+1)) \le g(\mathbf{z})$. Together with the closedness of g we further obtain that $\lim_{t\to\infty} g(\mathbf{x}(t+1)) = g(\mathbf{z})$. Note that f is continuous, we thus conclude that $\lim_{t\to\infty} F(\mathbf{x}(t+1)) = F(\mathbf{z})$ for all $\mathbf{z} \in \Omega$. Note that $F(\mathbf{x}(t)) \downarrow F^*$. Thus for all $\mathbf{x}^* \in \Omega$, we have $F(\mathbf{x}^*) \equiv F^*$. Now fix $\varepsilon > 0$. Since Ω is compact, for t sufficiently large we have $\mathrm{dist}_\Omega(\mathbf{x}(t)) \le \varepsilon$. We now have all ingredients to apply the KŁ inequality in Theorem 10: for all sufficiently large t,

$$\varphi'(F(\mathbf{x}(t)) - F^*) \cdot \mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0}) \ge 1. \tag{60}$$

Since φ is concave, we obtain

$$\begin{aligned}\Delta_{t,t+1} &:= \varphi(F(\mathbf{x}(t)) - F^*) - \varphi(F(\mathbf{x}(t+1)) - F^*) \\ &\ge \varphi'(F(\mathbf{x}(t)) - F^*)(F(\mathbf{x}(t)) - F(\mathbf{x}(t+1))) \\ &\overset{\text{(i)}}{\ge} \frac{\alpha\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2}{\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0})},\end{aligned} \tag{61}$$

where (i) follows from Assumption 8 and eq. (60). It is clear that the function φ (composed with F) serves as a Lyapunov function. Using the elementary inequality $2\sqrt{ab} \le a + b$ we obtain from eq. (61) that for t sufficiently large,

$$2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \le \tfrac{\delta}{\alpha}\Delta_{t,t+1} + \tfrac{1}{\delta}\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0}),$$

where $\delta > 0$ will be specified later. Recalling the bound for $\partial F(\mathbf{x}(t))$ in eq. (30), and summing over t from m (sufficiently large) to n gives:

$$\begin{aligned}2\sum_{t=m}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| &\le \sum_{t=m}^{n} \frac{\delta}{\alpha}\Delta_{t,t+1} + \sum_{t=m}^{n} \frac{1}{\delta}\mathrm{dist}_{\partial F(\mathbf{x}(t))}(\mathbf{0}) \\ &\overset{\text{(i)}}{\le} \frac{\delta}{\alpha}\varphi(F(\mathbf{x}(m)) - F^*) + \sum_{t=m}^{n} \frac{\sqrt{p}(1/\eta + 2L)}{\delta} \sum_{k=(t-2s)_+}^{t} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\le \frac{\delta}{\alpha}\varphi(F(\mathbf{x}(m)) - F^*) + \frac{(2s+1)\sqrt{p}(1/\eta + 2L)}{\delta} \sum_{k=(m-2s)_+}^{m-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \\ &\quad + \frac{(2s+1)\sqrt{p}(1/\eta + 2L)}{\delta} \sum_{t=m}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\|,\end{aligned}$$

where (i) is due to eq. (30). Setting $\delta = (2s+1)\sqrt{p}(1/\eta + 2L)$ and rearranging gives

$$\begin{aligned}\sum_{t=m}^{n} \|\mathbf{x}(t+1) - \mathbf{x}(t)\| &\le \frac{(2s+1)\sqrt{p}(1/\eta + 2L)}{\alpha}\varphi(F(\mathbf{x}(m)) - F^*) \\ &\quad + \sum_{k=(m-2s)_+}^{m-1} \|\mathbf{x}(k+1) - \mathbf{x}(k)\|.\end{aligned}$$

Since the right-hand side is finite, let n tend to infinity completes the proof for eq. (58). ■

Compared with (16) in Theorem 5, we now have the successive differences to be absolutely summable (instead of square summable). This is a significantly stronger result as it immediately implies that the whole sequence is Cauchy and hence convergent, whereas we cannot get the same conclusion from the square summable property in Theorem 5. We note that local maxima are excluded from being the limit in Theorem 11, due to Assumption 8. Also, the boundedness assumption on the trajectory $\{\mathbf{x}(t)\}$ is easy to satisfy, for instance, when F has bounded sublevel sets. We refer to (Attouch et al., 2010, Remark 3.3) for more conditions that imply the boundedness condition. Moreover, following similar arguments in Attouch et al. (2010) we can also determine the local convergence rates of the sequences generated by m-PAPG.

In the remaining part of this section we provide some justifications for the sufficient decrease property in Assumption 8. For simplicity we assume all worker machines perform updates in each time step t:

Assumption 9 $\forall i = 1, \cdots, p, \forall t, t \in T_i$.

Note that Assumption 9 is commonly adopted in the analysis of many recent parallel systems Agarwal and Duchi (2011); Feyzmahdavian et al. (2014); Ho et al. (2013); Li et al. (2014); Liu and Wright (2015); Recht et al. (2011). In fact, Assumption 9 is somewhat necessary to justify Assumption 8. This is because Assumption 8 requires a sufficient decrease of the function value at every iteration k, which may not hold under the PAP as all machines can be idle for s iterations in the worst case. In other words, to achieve convergence of $\{\mathbf{x}_k\}_k$

to a critical point in nonconvex optimization under the KL inequality, the parallel system should make a steady progress per-iteration. As we show next, this is guaranteed under Assumptions 9 and 10.

We will replace the sufficient decrease property in Assumption 8 with the following key property that turns out to be easier to verify:

Assumption 10 (Proximal Lipschitz) *We say a pair of functions f and g satisfy the proximal Lipschitz property on a sequence $\{\mathbf{x}(t)\}$ if for all η sufficiently small, there exists $L_\eta \in o(1)$, i.e. $L_\eta \to 0$ as $\eta \to 0$, such that for all large t,*

$$\|\Delta_\eta(\mathbf{x}(t)) - \Delta_\eta(\mathbf{x}(t+1))\| \leq L_\eta \|\mathbf{x}(t) - \mathbf{x}(t+1)\|, \tag{62}$$

where[2] $\Delta_\eta(\mathbf{x}) \in \operatorname{prox}_g^\eta(\mathbf{x} - \eta\nabla f(\mathbf{x})) - \mathbf{x}$.

The proximal Lipschitz assumption is motivated by the special case where $g \equiv 0$ and hence $\Delta_\eta(\mathbf{x}) = -\eta\nabla f(\mathbf{x})$ is η-Lipschitz, thanks to Assumption 2. As we have seen in previous sections, Lipschitz continuity plays a crucial role in our proof where a major difficulty is to control the inconsistencies among different worker machines due to communication delays. Similarly here, the proximal Lipschitz property, as we show next, allows us to remove the sufficient decrease property in Assumption 8—the seemingly strong assumption that we needed in proving our main result Theorem 11.

Let us first present a quick justification for Assumption 10.

Lemma 12 *Suppose the functions f and g both have Lipschitz continuous gradient, then Assumption 10 holds for any sequence $\{\mathbf{x}(t)\}$.*

Proof Let us denote L_f and L_g as the Lipschitz constant of the gradient ∇f and ∇g, respectively. Since $\Delta_\eta(\mathbf{x}) \in \operatorname{prox}_g^\eta(\mathbf{x} - \eta\nabla f(\mathbf{x})) - \mathbf{x}$, using the optimality condition for the proximal map, see for instance (Yu et al., 2015, Proposition 7(iii)), we have

$$\mathbf{x} + \Delta_\eta(\mathbf{x}) + \eta\nabla g\big(\mathbf{x} + \Delta_\eta(\mathbf{x})\big) = \mathbf{x} - \eta\nabla f(\mathbf{x}),$$

and similarly

$$\mathbf{z} + \Delta_\eta(\mathbf{z}) + \eta\nabla g\big(\mathbf{z} + \Delta_\eta(\mathbf{z})\big) = \mathbf{z} - \eta\nabla f(\mathbf{z}).$$

Subtracting one inequality from another, we obtain

$$\begin{aligned}\|\Delta_\eta(\mathbf{x}) - \Delta_\eta(\mathbf{z})\| &= \|\eta\nabla g(\mathbf{z} + \Delta_\eta(\mathbf{z})) - \eta\nabla g(\mathbf{x} + \Delta_\eta(\mathbf{x})) + \eta\nabla f(\mathbf{z}) - \eta\nabla f(\mathbf{x})\| \\ &\leq \eta L_g\|\mathbf{z} - \mathbf{x} + \Delta_\eta(\mathbf{z}) - \Delta_\eta(\mathbf{x})\| + \eta L_f\|\mathbf{z} - \mathbf{x}\| \\ &\leq \eta L_g\|\Delta_\eta(\mathbf{z}) - \Delta_\eta(\mathbf{x})\| + \eta(L_f + L_g)\|\mathbf{z} - \mathbf{x}\|.\end{aligned}$$

Rearranging we obtain

$$\|\Delta_\eta(\mathbf{x}) - \Delta_\eta(\mathbf{z})\| \leq \frac{\eta(L_f + L_g)}{1 - \eta L_g}\|\mathbf{z} - \mathbf{x}\|,$$

when $0 < \eta < 1/L_g$. Clearly, when η is mall, the leading coefficient $\frac{\eta(L_f+L_g)}{1-\eta L_g} \in \mathcal{O}(\eta) \subseteq o(1)$, and our proof is complete. ■

2. Should the proximal map be multi-valued, we contend with any single-valued selection.

It is clear that Lemma 12 captures the motivating case $g \equiv 0$, but also many other important functions, such as the widely-used regularization function $g = \|\cdot\|_p^p$ for any $p > 1$. We can now continue with our next result in this section.

Theorem 13 *Let Assumptions 1 to 4 and 9 hold for m-PAPG, and let F satisfy the KL property in Theorem 10. Fix any $r > 1$ with $C = \frac{r^{s+1}-1}{r-1}$ and step size η such that $\eta < \frac{1}{L(1+2\sqrt{p}C+2\sqrt{p}s)}$. If for each local sequence $\{\mathbf{x}^i(t)\}$ generated by m-PAPG, Assumption 10 holds with $L_\eta \leq \frac{r^2-1}{2pr^2C^2}$, and the global sequence $\{\mathbf{x}(t)\}$ is bounded, then the finite length properties in (58) and (59) hold. Then, Assumption 8 holds, and consequently, $\{\mathbf{x}(t)\}$ and $\{\mathbf{x}^i(t)\}_{i=1}^p$ converge to the same critical point of F based on Theorem 11.*

Theorem 13 assumes that $L_\eta \leq \frac{r^2-1}{2pr^2C^2}$. We note that L_η implicitly depends on the stepsize η, i.e., $L_\eta \to 0$ as $\eta \to 0$ (see Assumption 10). Thus, one can tune the stepsize η to be small enough such that L_η satisfies the requirement. As an example, if $g = \|\mathbf{x}\|_2^2$, then one can calculate that $L_\eta = \mathcal{O}(\eta)$. In this case, we should choose the stepsize to be roughly $\eta \leq \frac{r^2-1}{2pr^2C^2}$.

Proof Using the elementary inequality $\|a\|^2 - \|b\|^2 \leq 2\|a\|\|a-b\|$, we have for all t:

$$\begin{aligned}&\|\mathbf{x}(t+1) - \mathbf{x}(t)\|^2 - \|\mathbf{x}(t+2) - \mathbf{x}(t+1)\|^2 \\ &\quad\leq 2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \|(\mathbf{x}(t+1) - \mathbf{x}(t)) - (\mathbf{x}(t+2) - \mathbf{x}(t+1))\| \\ &\quad\leq 2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \sum_{i=1}^p \|(x_i(t+1) - x_i(t)) - (x_i(t+2) - x_i(t+1))\| \\ &\quad\overset{\text{(i)}}{\leq} 2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \sum_{i=1}^p \left\|\Delta_\eta(\mathbf{x}^i(t)) - \Delta_\eta(\mathbf{x}^i(t+1))\right\| \\ &\quad\overset{\text{(ii)}}{\leq} 2\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \left(\sum_{i=1}^p L_\eta\|\mathbf{x}^i(t) - \mathbf{x}^i(t+1)\|\right) \\ &\quad\overset{\text{(iii)}}{\leq} 2pL_\eta\|\mathbf{x}(t+1) - \mathbf{x}(t)\| \cdot \sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1) - \mathbf{x}(k)\|,\end{aligned} \tag{63}$$

where (i) is due to Assumption 9 hence $t \in T_i$ for all t, (ii) follows from Assumption 10, and (iii) is due to (15).

If for some $r > 1$ there exists some T such that for all $t \geq T$,

$$\sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1) - \mathbf{x}(k)\| \geq C\|\mathbf{x}(t+1) - \mathbf{x}(t)\|, \tag{64}$$

where $C = \frac{r^{s+1}-1}{r-1} > s+1$ (since $r > 1$ and w.l.o.g. $s > 0$). Summing the index t from T to n yields

$$C\sum_{t=T}^n \|\mathbf{x}(t+1) - \mathbf{x}(t)\| \leq \sum_{t=T}^n \sum_{k=(t-s)_+}^t \|\mathbf{x}(k+1) - \mathbf{x}(k)\|$$

$$\leq (s+1)\sum_{t=(T-s)_+}^{n}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|,$$

which after rearranging terms becomes

$$(C-s-1)\sum_{t=T}^{n}\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \leq (s+1)\sum_{t=(T-s)_+}^{T-1}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|.$$

Since the right hand side does not depend on n, letting n tend to infinity we conclude

$$\sum_{t=0}^{\infty}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|<\infty, \tag{65}$$

and the proof of the finite length property would be complete.

Therefore, in the remaining part of the proof, we can assume (64) fails for infinitely many t. Take any such $t=\hat{t}$, we have

$$\sum_{k=(t-s)_+}^{t}\|\mathbf{x}(k+1)-\mathbf{x}(k)\| \leq C\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \leq C^2\|\mathbf{x}(t+1)-\mathbf{x}(t)\|, \tag{66}$$

since $C>1$. Combining (63) and (66) we have for $t=\hat{t}$:

$$\begin{aligned}\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2-\|\mathbf{x}(t+2)-\mathbf{x}(t+1)\|^2 &\leq 2pL_\eta C^2\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2\\ &\leq \left(1-\frac{1}{r^2}\right)\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2,\end{aligned}$$

if η is small enough (recall that $L_\eta=o(1)$). After rearranging terms we conclude that for $t=\hat{t}$:

$$\|\mathbf{x}(t+1)-\mathbf{x}(t)\| \leq r\|\mathbf{x}(t+2)-\mathbf{x}(t+1)\|. \tag{67}$$

Using induction we can continue the same process for any $t\geq\hat{t}$. Indeed, suppose (67) is true for any $t\leq m-1$, then (63) holds (for any t), and (66) also holds: If $m\leq\hat{t}+s$, then

$$\begin{aligned}\sum_{k=(m-s)_+}^{m}\|\mathbf{x}(k+1)-\mathbf{x}(k)\| &= \sum_{k=(m-s)_+}^{\hat{t}}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|+\sum_{k=\hat{t}+1}^{m}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|\\ &\overset{\text{(i)}}{\leq} \sum_{k=(\hat{t}-s)_+}^{\hat{t}}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|+\sum_{k=\hat{t}+1}^{m}r^{m-k}\|\mathbf{x}(m+1)-\mathbf{x}(m)\|\\ &\overset{\text{(ii)}}{\leq} C\left[\|\mathbf{x}(\hat{t}+1)-\mathbf{x}(\hat{t})\|+\sum_{k=\hat{t}+1}^{m}r^{m-k}\|\mathbf{x}(m+1)-\mathbf{x}(m)\|\right]\\ &\overset{\text{(iii)}}{\leq} C\sum_{k=\hat{t}}^{m}r^{m-k}\|\mathbf{x}(m+1)-\mathbf{x}(m)\|\\ &\overset{\text{(iv)}}{\leq} C^2\|\mathbf{x}(m+1)-\mathbf{x}(m)\|,\end{aligned}$$

where (i) is due to the induction hypothesis, (ii) is due to the definition of $\hat{t}$ and the fact that $C>1$, (iii) is due to again the induction hypothesis, and finally (iv) is due to the definition of C (recall $m\leq\hat{t}+s$). If $m>\hat{t}+s$, the same inequality, with C^2 replaced by C, would still hold (essentially dropping all the first terms on the right hand side of the above inequalities). Thus, (63) and (66) would imply again (67) for $t=m$.

Lastly, we recall from eq. (22) that for large t,

$$\begin{aligned}F(\mathbf{x}(t+1))-F(\mathbf{x}(t)) &\leq \tfrac{1}{2}(L-1/\eta)\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2\\ &\quad+\sqrt{p}L\|\mathbf{x}(t+1)-\mathbf{x}(t)\|\sum_{k=(t-s)_+}^{t-1}\|\mathbf{x}(k+1)-\mathbf{x}(k)\|.\\ &\leq \tfrac{1}{2}(L-1/\eta)\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2+\sqrt{p}CL\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2.\\ &\leq -\alpha\|\mathbf{x}(t+1)-\mathbf{x}(t)\|^2,\end{aligned}$$

where $\alpha=\frac{1}{2}(1/\eta-L-2\sqrt{p}CL)>0$ if η is small. Hence, the sufficient decrease property in Assumption 8 is verified and the finite length properties follow from Theorem 11. ■

Lastly, we show that Assumption 10 also holds for the important cardinality function $\|\mathbf{x}\|_0$ (number of nonzero entries).

Lemma 14 *Consider the same setting as in Theorem 5, then Assumption 10 holds for any function f and $g=\|\cdot\|_0$ on all local sequences $\{\mathbf{x}^i(t)\}$ of m-PAPG.*

Proof The crucial observation here is that for the cardinality function $g=\|\cdot\|_0$, its proximal map on the j-th entry can be chosen as:

$$\text{prox}_{g_j}^{\eta}(z_j)=\begin{cases} z_j, \text{ if } |z_j|>\sqrt{2\eta}\\ 0, \text{ otherwise}\end{cases}. \tag{68}$$

However, Theorem 5 implies that $\lim_{t\to\infty}\|\mathbf{x}^i(t+1)-\mathbf{x}^i(t)\|=0$. Thus, for t sufficiently large, the sequence $\{\mathbf{x}^i(t)\}$ will have the same support Ω (indices that have nonzero entries), for otherwise $\|\mathbf{x}^i(t+1)-\mathbf{x}^i(t)\|\geq\sqrt{2\eta}$ even if one index in the support changes. Therefore,

$$\begin{aligned}\|\Delta_\eta(\mathbf{x}^i(t+1))-\Delta_\eta(\mathbf{x}^i(t))\| &\overset{\text{(i)}}{\leq} \sum_{j\in\Omega}\|\text{prox}_{g_j}^{\eta}(x_j^i(t+1)-\eta\nabla_j f(\mathbf{x}^i(t+1)))-x_j^i(t+1)\\ &\qquad-\text{prox}_{g_j}^{\eta}(x_j^i(t)-\eta\nabla_j f(\mathbf{x}^i(t)))-x_j^i(t)\|\\ &\overset{\text{(ii)}}{\leq} \sum_{j\in\Omega}\|\eta\nabla_j f(\mathbf{x}^i(t+1))-\eta\nabla_j f(\mathbf{x}^i(t))\|\\ &\overset{\text{(iii)}}{\leq} \eta pL\|\mathbf{x}^i(t+1)-\mathbf{x}^i(t)\|,\end{aligned}$$

where (i) is the triangle inequality, (ii) uses the property of the proximal map (68), and (iii) is due to Assumption 2. ■

Note that similar results as Theorem 14 can be derived for the rank function, and more generally for functions whose proximal map is discontinuous with pieces satisfying Theorem 12 (for instance, the group cardinality norm $\|\cdot\|_{0,2}$).

7. Economical Implementation for Linear Models

In this section, we provide an economical implementation of m-PAPG on a distributed system for the widely used linear models:

$$\min_{\mathbf{x}\in\mathbb{R}^d} f(A\mathbf{x}) + g(\mathbf{x}), \tag{69}$$

where $A \in \mathbb{R}^{n\times d}$ corresponds to the data matrix. Typically $f : \mathbb{R}^n \to \mathbb{R}$ is the likelihood function and $g : \mathbb{R}^d \to \mathbb{R}$ is the regularizer. The data matrix A consists of n sample points and we have suppressed the labels in classification or the responses in regression. Support vector machines (SVM), Lasso, logistic regression, boosting, etc., all fit under this framework. Our interest here is when the model dimension d is much higher than the number of samples n (d can be up to hundreds of millions and n can be up to millions). This is also the usual setup in many computational biology and health care problems.

A direct implementation of m-PAPG can be inefficient in terms of both network communication and parameter storage. First, each machine needs to communicate with every other machine to synchronize the model blocks. This leads to a peer-to-peer network topology and result in a dense connection when the system holds hundreds of machines. Second, each machine needs to keep a local copy of the full model (i.e. $\mathbf{x}^i(t)$), which incurs a high storage cost when the dimension is high. Note that the local models $\mathbf{x}^i(t)$ are kept solely for the convenience of evaluating the partial gradient $\nabla_i f : \mathbb{R}^d \to \mathbb{R}^{d_i}$. For some problems such as the Lasso, a seemingly workaround is to pre-compute the Hessian $H = A^\top A$ and distribute the corresponding row blocks of H to each worker machine. This scheme, however, is problematic in the high dimensional setting: the pre-computation of the Hessian can be very costly, and each row block of H has a very large size ($d_i \times d$).

The above issues can be avoided by exploiting the structure of the linear model in eq. (69) and adopting the parameter server distributed system Ho et al. (2013); Li et al. (2014). The system dedicates a central server to store the key parameters, and let each worker machine to communicate only with the server. To be specific, we partition the data matrix A into p *column* blocks $A = [A_1, \ldots, A_p]$ and distribute the block $A_i \in \mathbb{R}^{n\times d_i}$ to machine i Boyd et al. (2010); Richtárik and Takáč (to appear) . Note the local update computed by machine i at the t-th iteration is

$$U_i(\mathbf{x}^i(t)) = \mathrm{prox}^\eta_{g_i}\big(x_i(t) - \eta A_i^\top f'(A\mathbf{x}^i(t))\big) - x_i(t). \tag{70}$$

Since machine i is in charge of updating the i-th block $x_i(t)$ of the global model, it suffices to have the matrix-vector product $A\mathbf{x}^i(t)$ to compute the local update in eq. (70) . If we initialize $\forall i,\ \mathbf{x}^i(0) \equiv \mathbf{0}$, then $A\mathbf{x}^i(t)$ can be written in a cumulative form as

$$A\mathbf{x}^i(t) = \sum_{j=1}^p A_j[\mathbf{x}^i(t)]_j = \sum_{j=1}^p \sum_{k=0}^{\tau_j^i(t)} \underbrace{A_j \mathbb{I}_{\{k\in T_j\}} U_j(\mathbf{x}^j(k))}_{\Delta_j(k)},$$

where recall that machine i only has access to a delayed copy $x_j(\tau_j^i(t))$ of the parameters in machine j. Hence, to evaluate the matrix-vector product, every machine needs to accumulate $\Delta_j(k)$ over all machines upto a delayed clock. Thus, we aggregate $\Delta_j(t) \in \mathbb{R}^n$ on the parameter server whenever it is generated and sent by the worker machines. In details,

Algorithm 1 Economic Implementation of m-PAPG

For the server:
  while recieves update Δ_i from machine i **do**
    $\blacktriangle \leftarrow \blacktriangle + \Delta_i$
  end while
  while machine i sends a pull request **do**
    send $\blacktriangle$ to machine i
  end while
For machine i at active clock $t \in T_i$:
  pull $\blacktriangle$ from the server
  $U_i \leftarrow \mathrm{prox}^\eta_{g_i}(x_i - \eta A_i^\top f'(\blacktriangle)) - x_i$
  send $\Delta_i = A_i U_i$ to the server
  update $x_i \leftarrow x_i + U_i$

the worker machines first pull this matrix-vector product (denoted as $\blacktriangle$) from the server to conduct the local computation in eq. (70). Then machine i performs the local update:

$$x_i(t+1) = x_i(t) + U_i(\mathbf{x}^i(t)). \tag{71}$$

Note that machine i does not maintain or update other blocks of parameters $x_j(t), j \neq i$. Lastly, machine i computes and sends the vector $\Delta_i(t) = A_i U_i(\mathbf{x}^i(t)) \in \mathbb{R}^n$ to the server, and the server immediately performs the aggregation:

$$\blacktriangle \leftarrow \blacktriangle + \Delta_i(t). \tag{72}$$

We summarize the above economical implementation in Algorithm 1, where $\blacktriangle$ denotes the aggregated matrix-vector product. The storage cost for each worker machine is $O(nd_i)$ (for storing A_i only). Each iteration requires two matrix-vector products that cost $O(nd_i)$ in the dense case, and the communication of a length n vector between the server and the worker machines. Note that the cost is significantly lower than the direct implementation.

8. Experiments

In this section, we empirically verify the convergence properties and time efficiency of m-PAPG. All data are generated via normal distribution with the columns being normalized to have unit norm. We first test the convergence properties of m-PAPG via a non-convex Lasso problem with the group regularizer $\|\cdot\|_{0,2}$, which takes the form

$$\min_{\mathbf{x}\in\mathbb{R}^d}\ \tfrac{1}{2}\|A\mathbf{x} - \mathbf{b}\|^2 + \lambda\|\mathbf{x}\|_{0,2}, \tag{73}$$

where we set sample size $n = 1000$ and dimension size $d = 2000$, and the group norm divides the whole model into 20 groups with equal dimension. We use 4 machines (cores) with each handling five groups of coordinates, and consider maximal staleness $s = 0, 10, 20, 30$, respectively. To better demonstrate the effect of staleness, we let machines only communicate when exceed the maximum staleness. This can be viewed as the worst case communication scheme and a larger s brings more staleness into the system. We set the learning rate to

have the form $\eta(\alpha s) = 1/(L_f + 2L\alpha s), \alpha > 0$, that is, a linear dependency on staleness s as suggested by Theorem 5. Then we run Algorithm 1 with different staleness and use $\eta(0), \eta(10), \eta^*(\alpha s)$, respectively, where $\eta^*(\alpha s)$ is the largest step size we tuned for each s that achieves a stable convergence. We track the global model $\mathbf{x}(t)$ and plot the results in Figure 1. Note that with the large step size $\eta(0)$ all instances (with nonzero staleness) diverge hence are not presented. With $\eta(10)$ (Figure 1, left), the staleness does not substantially affect the convergence in terms of the objective value. We note that the objective curves converge to slightly different minimal values due to the non-convexity of problem (73). With $\eta^*(\alpha s)$ (Figure 1, middle), it can be observed that adding a slight penalty αs on the learning rate suffices to achieve a stable convergence, and the penalty grows as s increases, which is intuitive since a larger staleness requires a smaller step size to cancel the inconsistency. In particular, for $s = 10$ the best convergence is comparable to the bulk synchronized case $s = 0$. (Figure 1, right) further shows the asymptotic convergence behavior of the global model $\mathbf{x}(t)$ under the step size $\eta^*(\alpha s)$. It is clear that a linear convergence is eventually attained, which confirms the finite length property in Theorem 11.

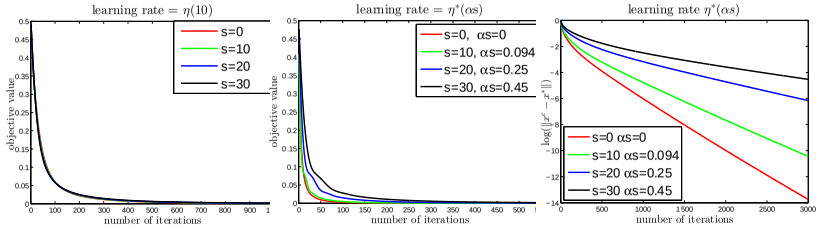


Figure 1: Convergence curves of m-PAPG under different staleness parameter s and step size η.

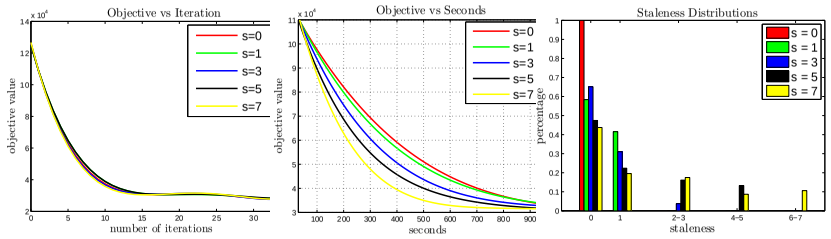


Figure 2: Efficiency of m-PAPG on a large scale Lasso problem.

Next, we verify the time and communication efficiency of m-PAPG via an l_1 regularized quadratic programming problem with very high dimensions, taking the form

$$\min_{\mathbf{x}} \ \tfrac{1}{2}\mathbf{x}^\top A^\top A\mathbf{x} + \lambda\|\mathbf{x}\|_1. \tag{74}$$

We generate samples of size $n = 1$Million and dimension $d = 100$Millions. We implement Algorithm 1 on Petuum Ho et al. (2013); Dai et al. (2014) — a stale synchronous parallel system which updates the local parameter caches via stale synchronous communications. The system contains 100 computing nodes and each is equipped with 16 AMD Opteron processors and 16GB RAM linked by 1Gbps ethernet. We fix the learning rate $\eta = 10^{-3}$ and consider maximum staleness $s = 0, 1, 3, 5, 7$, respectively. (Figure 2, left) shows that per-iteration progress is virtually indistinguishable among various staleness settings, which is consistent with our previous experiment. (Figure 2, middle) shows that system throughput is significantly higher when we introduce staleness. This is due to lower synchronization overheads, which offsets any potential loss due to staleness in progress per iteration. We also track the distributions of staleness during the experiments, where we record in ▲ the clocks of the freshest updates that accumulate from all the machines. Then whenever a machine pulls ▲ from the server, it compares its local clock with these clocks and records the clock differences. (Figure 2, right) shows the distributions of staleness under different maximal staleness settings. Observe that bulk synchronous ($s = 0$) peaks at staleness 0 by design, and the distribution concentrates in small staleness area due to the eager communication mechanism of Petuum. It can be seen that a small amount of staleness is sufficient to relax the communication bottlenecks without affecting the iterative convergence rate much.

9. Conclusion

We have proposed m-PAPG as an extension of the proximal gradient algorithm to the model parallel and partially asynchronous setting. m-PAPG allows worker machines to operate asynchronously as long as they are not too far apart, hence greatly improves the system throughput. The convergence properties of m-PAPG are thoroughly analyzed. In particular, we proved that: 1) every limit point of the sequences generated by m-PAPG is a critical point of the objective function; 2) under an additional error bound condition, the function values decay periodically linearly; 3) under the additional Kurdyka-Łojasiewicz inequality, the sequences generated by m-PAPG converge to the same critical point, provided that a proximal Lipschitz condition is satisfied. In the future we plan to further weaken the proximal Lipschitz condition so that our analysis can handle many more nonsmooth functions.

Acknowledgment

This work of Y. Zhou and Y. Liang is supported in part by the grants AFOSR FA9550-16-1-0077, NSF ECCS-1818904 and CCF-1761506.

References

P.-A. Absil, R. Mahony, and B. Andrews. Convergence of the iterates of descent methods for analytic cost functions. *SIAM Journal on Optimization*, 16(2):531–547, 2005.

Alekh Agarwal and John C. Duchi. Distributed delayed stochastic optimization. In *Advances in Neural Information Processing Systems 24*, pages 873–881. 2011.

Hedy Attouch and Jerome Bolte. On the convergence of the proximal algorithm for nonsmooth functions involving analytic features. *Mathematical Programming*, 116(1-2):5–16, 2009. ISSN 0025-5610.

Hedy Attouch, Jerome Bolte, Patrick Redont, and Antoine Soubeyran. Proximal alternating minimization and projection methods for nonconvex problems: An approach based on the Kurdyka-Łojasiewicz inequality. *Mathematics of Operations Research*, 35(2):438–457, 2010.

Gérard M. Baudet. Asynchronous iterative methods for multiprocessors. *Journal of the Association for Computing Machinery*, 25(2):226–244, 1978.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Img. Sci.*, 2(1):183–202, 2009.

Amir Beck and Marc Teboulle. Smoothing and first order methods: A unified framework. *SIAM Journal on Optimization*, 22(2):557–580, 2012.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1989.

Jéróme Bolte, Aris Danilidis, Olivier Ley, and Laurent Mazet. Characterizations of Łojasiewicz inequalities and applications: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

Jerome Bolte, Shoham Sabach, and Marc Teboulle. Proximal alternating linearized minimization for nonconvex and nonsmooth problems. *Mathematical Programming*, 146(1-2): 459–494, 2014.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

D. Chazan and W. Miranker. Chaotic relaxation. *Linear Algebra and Its Applications*, 2: 199–222, 1969.

Ronan Collobert, Fabian Sinz, Jason Weston, and Léon Bottou. Trading convexity for scalability. pages 201–208, 2006.

Wei Dai, Abhimanu Kumar, Jinliang Wei, Qirong Ho, Garth Gibson, and Eric P. Xing. High-performance distributed ml at scale through parameterserver consistency models. In *AAAI*, 2014.

Jeffrey Dean and Sanjay Ghemawat. Mapreduce: Simplified data processing on large clusters. *Communications of ACM*, 51(1):107–113, 2008.

Dmitriy Drusvyatskiy and Adrian S. Lewis. Error bounds, quadratic growth, and linear convergence of proximal methods, 2016.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96(456):1348–1360, 2001.

Cong Fang and Zhouchen Lin. Parallel asynchronous stochastic variance reduction for nonconvex optimization, 2017.

Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

H.R. Feyzmahdavian, A. Aytekin, and M. Johansson. A delayed proximal gradient method with linear convergence rate. In *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

Masao Fukushima and Hisashi Mine. A generalized proximal point algorithm for certain non-convex minimization problems. *International Journal of Systems Science*, 12(8): 989–1000, 1981.

D. Hajinezhad and M. Hong. Nonconvex alternating direction method of multipliers for distributed sparse principal component analysis. In *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, pages 255–259, Dec 2015.

Qirong Ho, James Cipar, Henggang Cui, Seunghak Lee, Jin Kyu Kim, Phillip B. Gibbons, Garth A Gibson, Greg Ganger, and Eric P Xing. More effective distributed ml via a stale synchronous parallel parameter server. In *Advances in Neural Information Processing Systems 26*, pages 1223–1231. 2013.

Mingyi Hong, Zhi Quan Luo, and Meisam Razaviyayn. Convergence analysis of alternating direction method of multipliers for a family of nonconvex problems. *SIAM Journal on Optimization*, 26(1):337–364, 1 2016.

Zhouyuan Huo and Heng Huang. Asynchronous mini-batch gradient descent with variance reduction for non-convex optimization, 2017.

Krzysztof Kurdyka. On gradients of functions definable in o-minimal structures. *Annales de l'institut Fourier*, 48(3):769–783, 1998.

Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. Scaling distributed machine learning with the parameter server. In *11th USENIX Symposium on Operating Systems Design and Implementation (OSDI 14)*, pages 583–598, 2014.

Ji Liu and Stephen J. Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

P. D. Lorenzo and G. Scutari. NEXT: In-network nonconvex optimization. *IEEE Transactions on Signal and Information Processing over Networks*, 2(2):120–136, June 2016.

Yucheng Low, Danny Bickson, Joseph Gonzalez, Carlos Guestrin, Aapo Kyrola, and Joseph M. Hellerstein. Distributed graphlab: A framework for machine learning and data mining in the cloud. *Proc. VLDB Endow.*, 5(8):716–727, 2012.

Zhaosong Lu and Lin Xiao. On the complexity analysis of randomized block-coordinate descent methods. *Mathematical Programming*, 152:615–642, 2015.

Zhi-Quan Luo and Paul Tseng. Error bounds and convergence analysis of feasible descent methods: A general approach. *Annals of Operations Research*, 46(1):157–178, 1993.

Rahul Mazumder, Jerome H. Friedman, and Trevor Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495): 1125–1138, 2011.

Yurii Nesterov. Gradient methods for minimizing composite functions. *Mathematical Programming, Series B*, 140:125–161, 2013.

Benjamin Recht, Christopher Re, Stephen Wright, and Feng Niu. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24*, pages 693–701. 2011.

P. Richtárik and M. Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(1-2):1–38, 2014.

P. Richtárik and M. Takáč. Distributed Coordinate Descent Method for Learning with Big Data. *Journal of Machine Learning Research*, to appear.

R.T. Rockafellar and R.J.B. Wets. *Variational Analysis*. Springer, 1997.

Suvrit Sra. Scalable nonconvex inexact proximal splitting. In *Advances of Neural Information Processing Systems*, 2012.

Paul Tseng. On the rate of convergence of a partially asynchronous gradient projection algorithm. *SIAM Journal on Optimization*, 1(4):603–619, 1991.

Leslie G. Valiant. A bridging model for parallel computation. *Communications of ACM*, 33(8):103–111, 1990.

Yichao Wu and Yufeng Liu. Robust truncated hinge loss support vector machines. *Journal of the American Statistical Association*, 102(479):974–983, 2007.

Linli Xu, Koby Crammer, and Dale Schuurmans. Robust support vector machine training via convex outlier ablation. 2006.

Yaoliang Yu, Xun Zheng, Micol Marchetti-Bowick, and Eric P. Xing. Minimizing nonconvex non-separable functions. In *The* 17^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.

Matei Zaharia, Mosharaf Chowdhury, Michael J. Franklin, Scott Shenker, and Ion Stoica. Spark: Cluster computing with working sets. pages 10–10, 2010.

Cun-Hui Zhang. Nearly unbaised variable selection under minimax concave penalty. *Annals of Statistics*, 38(2):894–942, 2010.

Cun-Hui Zhang and Tong Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, 27(4):576–593, 2012.

Hui Zhang. The restricted strong convexity revisited: analysis of equivalence to error bound and quadratic growth. *Optimization Letters*, pages 1–17, 2016.

Y. Zhou, Y.L. Yu, W. Dai, Y.B. Liang, and E.P. Xing. On convergence of model parallel proximal gradient algorithm for stale synchronous parallel system. In *The* 19^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2016.

Zirui Zhou, Qi Zhang, and Anthony Man-Cho So. $\ell_{1,p}$-norm regularization: Error bounds and convergence rate analysis of first-order methods. In *Proceedings of the 32nd International Conference on Machine Learning*, pages 1501–1510. JMLR Workshop and Conference Proceedings, 2015.

Journal of Machine Learning Research 19 (2018) 1-32 Submitted 9/17; Revised 7/18; Published 7/18

Refining the Confidence Level for Optimistic Bandit Strategies

Tor Lattimore TOR.LATTIMORE@GMAIL.COM
DeepMind
5 New Street
London, United Kingdom

Editor: Peter Auer

Abstract

This paper introduces the first strategy for stochastic bandits with unit variance Gaussian noise that is simultaneously minimax optimal up to constant factors, asymptotically optimal, and never worse than the classical upper confidence bound strategy up to universal constant factors. Preliminary empirical evidence is also promising. Besides this, a conjecture on the optimal form of the regret is shown to be false and a finite-time lower bound on the regret of any strategy is presented that very nearly matches the finite-time upper bound of the newly proposed strategy.

Keywords Stochastic bandits, sequential decision making, regret minimisation.

1. Introduction

Let $k > 1$ be the number of bandits (or arms) and $\mu \in \mathbb{R}^k$ be the unknown vector of mean payoffs so that $\mu_i \in \mathbb{R}$ is the expected payoff when playing the ith bandit (or arm). In each round $t \in [n] = \{1, 2, \ldots, n\}$ the player chooses an arm $A_t \in [k]$ based on past observations and (optionally) an independent source of randomness. After making her choice, the player observes a payoff $X_t = \mu_{A_t} + \eta_t$ where $\eta_1, \eta_2, \ldots, \eta_n$ is a sequence of independent standard Gaussian random variables. It is standard to minimise the expected pseudo-regret (from now on, just the regret). Let $\Delta_i(\mu) = \max_j \mu_j - \mu_i$ be the suboptimality gap for the ith arm. The regret over n rounds is

$$\mathcal{R}_n(\mu) = \mathbb{E}\left[\sum_{t=1}^{n} \Delta_{A_t}(\mu)\right] .$$

Because the regret depends on the unknown payoff vector, no strategy can hope to make the regret small for all μ simultaneously. There are a number of performance metrics in the literature, two of which are described below along with a new one. To spoil the surprise, the strategy introduced in the present article is simultaneously optimal with respect to all of them.

Worst-case optimality The *worst-case* regret of a strategy is the value of the regret it suffers when faced with the worst possible μ.

$$\mathcal{R}_n^{\text{WC}} = \sup_{\mu \in \mathbb{R}^k : \Delta(\mu) \in [0,1]^k} \mathcal{R}_n(\mu) .$$

The restriction to bounded suboptimality gaps is necessary to allow an algorithm to choose each arm at least once without suffering arbitrarily large regret. Generally problems with small suboptimality gaps are the most interesting. Provided that $n \geq k$ it is known that all algorithms suffer $\mathcal{R}_n^{\text{WC}} = \Omega(\sqrt{kn})$ (Auer et al., 1995).

Asymptotic optimality The worst-case regret obscures interesting structure in the problem that becomes relevant in practice. This motivates the study of a problem-dependent metric, which demands that strategies have smaller regret on 'easier' bandit instances. A strategy is called *asymptotically optimal* if

$$\lim_{n\to\infty} \frac{\mathcal{R}_n(\mu)}{\log(n)} = \sum_{i:\Delta_i(\mu)>0} \frac{2}{\Delta_i(\mu)} \qquad \text{for all } \mu \in \mathbb{R}^d .$$

The name is justified by the existence of policies satisfying the definition and lower bounds by Lai and Robbins (1985) and Burnetas and Katehakis (1996) showing that consistent policies (those with sub-polynomial regret on all μ) cannot do better.

The sub-UCB criteria While asymptotic analysis is quite insightful, the ultimate quantity of interest is the finite-time regret. To make a stab at quantifying this I say an algorithm is sub-UCB if there exist universal constants $C_1, C_2 > 0$ such that for all k, n and μ it holds that

$$\mathcal{R}_n(\mu) \leq C_1 \sum_{i=1}^{k} \Delta_i(\mu) + C_2 \sum_{i:\Delta_i(\mu)>0} \frac{\log(n)}{\Delta_i(\mu)} . \tag{1}$$

Of course UCB (Auer et al., 2002) satisfies Eq. (1), along with many other policies as shown in Table 2 in Appendix E, which outlines the long history of algorithms for stochastic finite-armed bandits. The study of this new metric can be justified in several ways. First, it provides a forgiving finite-time analogue of asymptotic optimality. Lai and Robbins (1985) derived asymptotic optimality by making a restriction on policies (the consistent ones). Consistency is an asymptotic notion, so it is not surprising that the resulting lower bound is also asymptotic. The sub-UCB notion is suggested by making a finite-time restriction on the worst case regret. Precisely, for any strategy the finite-time instance-dependent regret can be bounded in terms of the worst case regret by

$$\mathcal{R}_n(\mu) \geq \sup_{\varepsilon \in (0,1]} \sum_{i:\Delta_i(\mu)>0} \max\left\{0, \frac{2\log\left(\frac{n}{\mathcal{R}_n^{\text{WC}}}\right) + 2\log\left(\frac{\varepsilon\Delta_i(\mu)}{8}\right)}{(1+\varepsilon)\Delta_i(\mu)}\right\} \tag{2}$$

for all μ with $\max_i \Delta_i(\mu) \leq 1/2$ (Lattimore and Szepesvári, 2018). This means that if you demand a reasonable worst case bound, then the instance-dependent regret cannot be *much* better than sub-UCB. Note that the first sum in Eq. (1) is unavoidable for policies that always choose each arm at least once, which is also necessary for any algorithm to have reasonable worst case regret. The finite-time world is not as clean as the asymptotic and it is not easy to decide how tight Eq. (2) might be, which justifies the additional constant-factor allowance in Eq. (1) and the removal of the (typically negative) second logarithm term. The second justification for using Eq. (1) as a yardstick is that it is forgiving and yet recent policies that are minimax optimal up to constant factors do not satisfy it. One of the core contributions of this article is to correct these deficiencies. Note that Eq. (2) depends quite weakly on the worst case regret and is meaningful as long as $\mathcal{R}_n^{\text{WC}} = O(n^p)$ for p not too close to 1.

None of these criteria are perfect by themselves. Asymptotic optimality is achievable by policies with outrageous burn-in time and/or large minimax regret, minimax optimal policies may be unreasonably conservative on easy problems and sub-UCB policies may be far from asymptotically optimal.

Contributions The main contribution is a new strategy called ADA-UCB ('adaptive UCB') and analysis showing it is asymptotically optimal, minimax optimal and sub-UCB. No other algorithm is simultaneously minimax optimal and sub-UCB (see Table 2). Results are specialised to the Gaussian case with unit variance, but upper bounds can be generalised to subgaussian noise with known subgaussian constant at the price of increased constants (without losing asymptotic optimality) and longer proofs. The latter justifies the specialisation because it allows for an elegant concentration analysis via an embedding of Gaussian random walks into Brownian motion. Also included:

(a) Finite-time lower bounds showing the new strategy is close to optimal.

(b) A conjecture by Bubeck and Cesa-Bianchi (2012) is proven false.

(c) A generic analysis for a large class of strategies simplifying the analysis for existing strategies.

Beyond the concrete results, the approach used for deriving ADA-UCB by examining lower bounds will likely generalise to other noise models, and indeed, other sequential optimisation problems with an exploration/exploitation flavour. The contents of this article combines the best parts of two technical reports with improved results, intuition and analysis (Lattimore, 2015a, 2016b).

Notation For natural number n let $[n] = \{1, 2, \ldots, n\}$. Binary minimums and maximums are abbreviated by $\wedge$ and $\vee$ respectively. The complement of event A is A^c. Except where otherwise stated, it is assumed without loss of generality that $\mu_1 \geq \mu_2 \geq \ldots \geq \mu_k$. None of the proposed strategies depend on the labelling of the arms, so if this is not the case the indices can simply be re-ordered. The dependence of the suboptimality gap on the mean vector will usually be omitted when the context is clear: $\Delta_i = \Delta_i(\mu) = \max_j \mu_j - \mu_i$. Occasionally it is convenient to define $\mu_{k+1} = -\infty$ and $\Delta_{k+1} = \infty$. Let $T_i(t) = \sum_{s=1}^{t} \mathbb{1}_{\{A_s = i\}}$ be the number of times arm i has been chosen after round t and $\hat{\mu}_i(t) = \sum_{s=1}^{t} \mathbb{1}_{\{A_s=i\}} X_s / T_i(t)$ be the corresponding empirical estimate of its return. Let $\hat{\mu}_{i,s}$ be the empirical estimate of the mean of arm i after s samples from that arm so that $\hat{\mu}_{i,T_i(t)} = \hat{\mu}_i(t)$. Define σ-algebra $\mathcal{F}_t = \sigma(\xi, X_1, \ldots, X_t)$ to contain the information available to the strategy after round t, where ξ is an independent source of randomness that allows for randomness in the strategy. This means that formally a strategy is a sequence of random variables $(A_t)_t$ such that A_t is $\mathcal{F}_{t-1}$-measurable. It is assumed throughout that $n \geq k$. Finally, let $\overline{\log}(x) = \log((x+e)\log^{1/2}(x+e))$. A table of notation is available in Table 3 in the appendix.

2. The strategy

The ADA-UCB strategy chooses each arm once in arbitrary order for the first k rounds and subsequently $A_t = \arg\max_{i \in [k]} \gamma_i(t)$ where the index of arm i in round t is:

$$\gamma_i(t) = \hat{\mu}_i(t-1) + \sqrt{\frac{2}{T_i(t-1)} \overline{\log}\left(\frac{n}{H_i(t-1)}\right)},$$

$$\text{with} \quad H_i(t) = T_i(t) K_i(t) \quad \text{and} \quad K_i(t) = \sum_{j=1}^{k} \min\left\{1, \sqrt{\frac{T_j(t)}{T_i(t)}}\right\}.$$

At first sight the new index seems overly complicated. After the statement of the main regret guarantee I show how the strategy is derived in a principled fashion from *lower bounds* obtained

from information theoretic limits of the problem. Similar approaches have been used before, for example, by Agrawal et al. (1989) for reinforcement learning, and by Garivier and Kaufmann (2016) for pure exploration in bandits. One interesting consequence of this approach is that ADA-UCB is not a true index strategy in the sense that $\gamma_i(t)$ depends on random variables associated with other arms. An intriguing open question is whether or not there exists an index strategy for which all three performance criteria are met. A minor observation is that it is not clear whether or not a true index strategy should be allowed to depend on t or just on $n - t$ and the samples from the given arm.

Relation to other algorithms The index is the same as that used by Katehakis and Robbins (1995) except that $\log(t)$ has been replaced by $\overline{\log}(n/H_i(t-1))$. The change from $\log(\cdot)$ to $\overline{\log}(\cdot)$ is quite minor. Such inflations of the logarithmic term are typical for algorithms with finite-time guarantees. The main difference between ADA-UCB and previous work is the term inside the logarithm, often called the confidence level. The most common choice is the current round t, which is used by various versions of UCB, KL-UCB and BAYES-UCB (Katehakis and Robbins, 1995; Burnetas and Katehakis, 1996; Agrawal, 1995; Auer et al., 2002; Kaufmann et al., 2012; Cappé et al., 2013). Already in the early work by Lai (1987) there appeared an unnamed variant of UCB for which the confidence level was $n/T_i(t-1)$. Due to its similarity to KL-UCB (Cappé et al., 2013) this algorithm will be called KL-UCB* from now on. A variety of other choices have been used as shown in Table 1.

majority	t
Lai (1987)	n/T_i
Honda and Takemura (2010)	t/T_i
Audibert and Bubeck (2009)	$n/(kT_i)$
Degenne and Perchet (2016)	$t/(kT_i)$
Lattimore (2015a)	n/t

Table 1: Confidence levels

Computation A naive implementation of ADA-UCB requires a computation time that is quadratic in the number of arms in each round. Fortunately an incremental implementation leads to an algorithm with linear computation time by noting that:

(a) If $T_i(t-1) \leq T_{A_t}(t-1)$, then $\gamma_i(t+1) = \gamma_i(t)$.

(b) For arms i with $T_i(t-1) > T_{A_t}(t-1)$ the value of $K_i(t-1)$ may be computed incrementally by $K_i(t) = K_i(t-1) - \sqrt{(T_{A_t}(t)-1)/T_i(t)} + \sqrt{T_{A_t}(t)/T_i(t)}$.

(c) The index of A_t can be computed trivially in order k time.

The algorithm follows by maintaining a list of arms sorted by $T_i(t)$ and applying the above observations to incrementally update the indices. If all details are addressed carefully, then the computation required in round t is $O(\sum_{i=1}^{k} \mathbb{1}\{T_i(t-1) \geq T_{A_t}(t-1)\})$, which in the worst case is $O(k)$, but can be much smaller when a single arm is played significantly more often than any other.

Regret bound The theorem statement has a more complicated form than previous regret bounds for finite-armed bandits, mainly because it correctly deals with the case where there are many near-optimal arms that cannot be statistically identified within the time horizon. Define k_i and λ_i by

$$k_i = \sum_{j=1}^{k} \min\left\{1, \frac{\Delta_i}{\Delta_j}\right\} \qquad \text{and} \qquad \lambda_i = 1 + \frac{1}{\Delta_i^2} \overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right).$$

The main theorem is below, which gives the best known finite-time guarantee for any strategy, as well as all three optimality criteria defined in the introduction.

Theorem 1 *Assume that $\mu_1 \geq \mu_2 \geq \cdots \geq \mu_k$. Then there exists a universal constant $C > 0$ such that the regret of* ADA-UCB *is bounded by*

$$\mathcal{R}_n(\mu) \leq C \min_{i \in [k]} \left(n\bar{\Delta}_i + \sum_{m>i} \Delta_m \lambda_m \right), \qquad \text{where } \bar{\Delta}_i = \frac{1}{i} \sum_{m=1}^{i} \Delta_m . \tag{3}$$

Furthermore:

(a) ADA-UCB *is minimax optimal up to constant factors: $\mathcal{R}_n^{\text{wc}} \leq C' \sqrt{kn}$.*

(b) ADA-UCB *is sub-UCB: $\mathcal{R}_n(\mu) \leq C \sum_{m:\Delta_m>0} \left(\Delta_m + \frac{\log(n)}{\Delta_m} \right)$.*

(c) ADA-UCB *is asymptotically optimal: $\lim_{n\to\infty} \frac{\mathcal{R}_n(\mu)}{\log(n)} = \sum_{i:\Delta_i>0} \frac{2}{\Delta_i}$.*

Remark 2 *The assumption on the order of the arms is purely for cosmetic purposes. The algorithm does not need this ordering and treats all arms symmetrically.*

Intuition for bound and strategy Let $\mu \in [0,1]^k$ and i be a suboptimal arm so that $\Delta_i = \Delta_i(\mu) > 0$. Set $\mu' \in [0,2]^k$ equal to μ except for $\mu'_i = \mu_i + 2\Delta_i$, which means that arm i has the largest mean for the bandit determined by μ'. Provided that n is sufficiently large, a sub-UCB strategy should play arm i logarithmically often if the mean payoff vector is μ, and linearly often for μ'. Let $\mathbb{E}$ and $\mathbb{E}'$ and $\mathbb{P}$ and $\mathbb{P}'$ denote the measures on the outcomes $A_1, X_1, \ldots, A_n, X_n$ when the strategy interacts with the bandits determined by μ and μ' respectively. Let $\delta \in (0,1]$ be such that

$$\mathbb{E}[T_i(n)] = \frac{2\log(1/\delta)}{(2\Delta_i)^2} . \tag{4}$$

Since μ and μ' are only different in the ith coordinate, the problem of minimising the regret is essentially equivalent to a hypothesis test on the mean of the ith arm, which satisfies $|\mu'_i - \mu_i| = 2\Delta_i$. Using this idea, a standard information-theoretic argument (see the section on lower bounds for formal details) shows that $\mathbb{P}'(T_i(n) \leq n/2) \gtrsim \delta$. Abbreviate $\Delta'_i = \Delta_i(\mu')$. Since $\Delta'_j = \mu'_i - \mu'_j \geq \Delta_i$ for all $j \neq i$ it holds that

$$\mathcal{R}_n(\mu') \geq \frac{n\Delta_i}{2} \mathbb{P}'(T_i(n) \leq n/2) \gtrsim n\Delta_i \delta/2 . \tag{5}$$

Assuming the strategy is sub-UCB, then there exists a (hopefully small) constant $C > 0$ such that

$$\mathcal{R}_n(\mu') \leq C \sum_{j:\Delta'_j>0} \left(\Delta'_j + \frac{\log(n)}{\Delta'_j} \right) \approx C \sum_{j:\Delta'_j>0} \frac{\log(n)}{\Delta'_j} , \tag{6}$$

where the approximation follows because $\Delta'_j \in [0,2]$ has been assumed. By Eqs. (5) and (6):

$$\delta \lesssim \frac{2C\log(n)}{n\Delta_i} \sum_{j:\Delta'_j>0} \frac{1}{\Delta'_j} = \frac{2C\log(n)}{n\Delta_i^2} \sum_{j\neq i} \frac{\Delta_i}{\Delta_j + \Delta_i} \leq \frac{2Ck_i \log(n)}{n\Delta_i^2} .$$

The regret guarantee given in Theorem 1 is now justified up to constant factors and an extraneous additive $\log\log(\cdot)$ term by substituting the above display into Eq. (4) and writing the regret as $\mathcal{R}_n(\mu) = \sum_{i=1}^k \Delta_i \mathbb{E}[T_i(n)]$. The idea behind ADA-UCB is to use the approximation $\Delta_i^{-2} \approx T_i(t-1)$. The approximation is poor when t is small, but becomes reasonable at the critical time when arm i should no longer be played. Specifically, if $T_i(t-1) \approx \Delta_i^{-2}$, then we should expect $T_j(t-1) \approx \min\{T_i(t-1), \Delta_j^{-2}\} \approx \min\{\Delta_i^{-2}, \Delta_j^{-2}\}$. Then

$$\frac{n}{H_i(t-1)} = \frac{n}{\sum_{j=1}^k \min\left\{T_i(t-1), \sqrt{T_i(t-1)T_j(t-1)}\right\}} \approx \frac{n}{\sum_{j=1}^k \min\left\{\frac{1}{\Delta_i^2}, \frac{1}{\Delta_i\Delta_j}\right\}} = \frac{n\Delta_i^2}{k_i} .$$

The implication is that the index dynamically tunes its confidence level using the pull counts to loosely estimate the gaps. The ideas in this section are made formal in the proof of Theorem 1 or the lower bound (§5).

3. Asymptotic analysis

The primary purpose of this section is to prove part (c) of Theorem 1. Along the way, a finite-time regret bound for a whole class of strategies is derived, including slightly modified versions of KL-UCB* (Lai, 1987) and the MOSS (**M**inimax **O**ptimal in the **S**tochastic **S**etting, Audibert and Bubeck 2009). The analysis leads to an optimal worst-case analysis of MOSS and KL-UCB*, but not ADA-UCB. The following theorem holds for the class of index strategies that choose $A_t = t$ for $1 \leq t \leq k$ and subsequently maximise

$$\gamma_i(t) = \hat{\mu}_i(t-1) + \sqrt{\frac{2}{T_i(t-1)} \overline{\log}\left(\frac{n}{J_i(t-1)T_i(t-1)} \right)} , \tag{7}$$

where $J_i(t-1)$ is $\mathcal{F}_{t-1}$-measurable and $J_i(t-1) \in [a,b]$ almost surely for constants $0 < a \leq b$. Except for minor differences in the leading constant and the logarithmic term, this index is the same as MOSS if $J_i(t-1) = k$, KL-UCB* if $J_i(t-1) = 1$ and ADA-UCB if $J_i(t-1) = K_i(t-1)$.

Theorem 3 *For any $\varepsilon \in (0, 1/2)$ and $1 \leq \ell \leq k$, the regret of the strategy in Eq. (7) is at most*

$$\mathcal{R}_n(\mu) \leq n\Delta_\ell + \frac{2c_1 b}{\varepsilon^2 \Delta_{\ell+1}} + \sum_{i>\ell} \left(2\Delta_i + \frac{1}{\Delta_i} \left(1 + \frac{1}{\varepsilon^2} + \frac{2\,\overline{\log}\left(\frac{n\Delta_i^2}{a}\right)}{(1-2\varepsilon)^2} \right) \right) .$$

Furthermore:

(a) $\lim_{n\to\infty} \mathcal{R}_n(\mu)/\log(n) = \sum_{i:\Delta_i>0} \frac{2}{\Delta_i}$.

(b) If $b \leq k$, then $\mathcal{R}_n^{\text{wc}} \leq C\sqrt{nk\left(1 + \log\left(\frac{k}{a}\right)\right)}$ where $C > 0$ is a universal constant.

Before the proof a little more notation is required. For each i and $\Delta > 0$ let $\zeta_i(\Delta)$ be a random variable given by

$$\zeta_i(\Delta) = 1 + \max\{s : \hat{\mu}_{i,s} > \mu_i + \Delta\} . \tag{8}$$

Clearly, $\zeta_i(\Delta)$ is surely monotone non-increasing in Δ and may be upper bounded in expectation using Lemma 13 in the appendix.

Proof [of Theorem 3] Let $\Delta \in \mathbb{R}$ be the smallest value such that

$$\hat{\mu}_{1,s} + \sqrt{\frac{2}{s}\overline{\log}\left(\frac{n}{bs}\right)} \leq \mu_1 - \Delta \qquad \text{for all } 1 \leq s \leq n\,,$$

which is chosen so that $\gamma_1(t) \geq \mu_1 - \Delta$ for all t. By part (a) of Lemma 12 with $\alpha = n/b$ and $d = 1$ and $\lambda_1 = \infty$ we have for any $x > 0$ that

$$\mathbb{P}\left(\Delta \geq x\right) \leq \frac{c_1 b h_\lambda(1/x^2)}{n} = \frac{c_1 b}{n x^2}\,. \tag{9}$$

Define random variable

$$\Lambda_i = 1 + \max\left\{\frac{1}{\Delta_i^2},\, \zeta_i(\varepsilon\Delta_i),\, \frac{2}{(1-2\varepsilon)^2\Delta_i^2}\overline{\log}\left(\frac{n\Delta_i^2}{a}\right)\right\}\,.$$

The definitions of the policy, Λ_i and Δ ensure that if $\Delta_i > \Delta/\varepsilon$, then $T_i(n) \leq \Lambda_i$ and by Lemma 13,

$$\mathbb{E}[\Lambda_i] \leq 2 + \frac{1}{\Delta_i^2}\left(1 + \frac{1}{\varepsilon^2} + \frac{2\overline{\log}\left(\frac{n\Delta_i^2}{a}\right)}{(1-2\varepsilon)^2}\right)\,. \tag{10}$$

Hence the regret of the strategy maximising the index in Eq. (7) is

$$\begin{aligned}
\mathcal{R}_n(\mu) &= \mathbb{E}\left[\sum_{i=1}^k \Delta_i T_i(n)\right] \\
&= \mathbb{E}\left[\sum_{i=1}^{\ell} \Delta_i T_i(n)\right] + \mathbb{E}\left[\sum_{i=\ell+1}^{k} \mathbb{1}\left\{\Delta_i \leq \frac{\Delta}{\varepsilon}\right\}\Delta_i T_i(n)\right] + \mathbb{E}\left[\sum_{i=\ell+1}^{k} \mathbb{1}\left\{\Delta_i > \frac{\Delta}{\varepsilon}\right\}\Delta_i T_i(n)\right] \\
&\leq n\Delta_\ell + \mathbb{E}\left[\frac{n\Delta}{\varepsilon}\mathbb{1}\left\{\Delta \geq \varepsilon\Delta_{\ell+1}\right\}\right] + \sum_{i>\ell}\Delta_i\mathbb{E}\left[\Lambda_i\right]\,.
\end{aligned} \tag{11}$$

The last expectation in Eq. (11) is bounded using Eq. (10),

$$\sum_{i>\ell}\Delta_i\mathbb{E}\left[\Lambda_i\right] \leq \sum_{i>\ell}\left(2\Delta_i + \frac{1}{\Delta_i}\left(1 + \frac{1}{\varepsilon^2} + \frac{2\overline{\log}\left(\frac{n\Delta_i^2}{a}\right)}{(1-2\varepsilon)^2}\right)\right)\,. \tag{12}$$

Given an arbitrary random variable X and constant $b \in \mathbb{R}$ it holds that

$$\mathbb{E}[X] \leq b\mathbb{P}\left(X \geq b\right) + \int_b^\infty \mathbb{P}\left(X \geq x\right)dx\,.$$

The first expectation in Eq. (11) is bounded by combining the above display with Eq. (9),

$$\begin{aligned}
\frac{n}{\varepsilon}\mathbb{E}\left[\Delta\mathbb{1}\left\{\Delta \geq \varepsilon\Delta_{\ell+1}\right\}\right] &\leq n\Delta_{\ell+1}\mathbb{P}\left(\Delta \geq \varepsilon\Delta_{\ell+1}\right) + \frac{n}{\varepsilon}\int_{\varepsilon\Delta_{\ell+1}}^\infty \mathbb{P}\left(\Delta \geq x\right)dx \\
&\leq \frac{c_1 b}{\varepsilon^2\Delta_{\ell+1}} + \frac{c_1 b}{\varepsilon}\int_{\varepsilon\Delta_{\ell+1}}^\infty \frac{dx}{x^2} = \frac{2c_1 b}{\varepsilon^2\Delta_{\ell+1}}\,,
\end{aligned}$$

which together with Eq. (12) and Eq. (11) completes the proof of the first part. The asymptotic result follows by choosing $\ell = \max\{i : \Delta_i = 0\}$ and $\varepsilon = \log^{-1/4}(n)$. The equality follows by the lower bound of Lai and Robbins (1985). The worst-case bound follows by choosing $\varepsilon = 1/4$ and tuning the cut-off ℓ. ■

The failure of MOSS Notice that if $J_i(t-1) = k$, then $a = b = k$ and except for a smaller inflated logarithm the resulting policy is the same as the variant proposed by Ménard and Garivier (2017). The troublesome term in the regret is the second term, which when $\ell = 1$ is approximately k/Δ_2. By contrast, for more conservative strategies this term is approximately $1/\Delta_2$, which is negligible. An especially challenging regime is when $\Delta_2 = 1/k$ and $\Delta_i = 1$ for $i > 2$. Suppose now that $n = k^3$ and k is large, then the regret of UCB on this problem should be

$$\mathcal{R}_n = O\left(k\log(k)\right)\,.$$

For MOSS, however, the regret on this problem is $\Omega(\sqrt{nk}) = \Omega(k^2)$, which for large k is arbitrarily worse than UCB. A vague explanation of the poor performance is that although there are k arms, all but two of them are so suboptimal that *effectively* it is a two-armed bandit. And yet MOSS is heavily tuned for the k-armed case and suffers as a consequence. A more precise argument is that distinguishing the first and second arms requires $T_2(t) \approx T_1(t) \approx 1/\Delta_2^2 = k^2$. But after playing these arms roughly k^2 times each the confidence level is $n/(kT_i(t)) \approx 1$ and the likelihood of misidentification is large enough that the regret is $\Omega(n\Delta_2) = \Omega(k^2) = \Omega(\sqrt{nk})$. Note that for ADA-UCB we would expect $K_i(t) \approx 2$ and the confidence level is approximately $n/k^2 = k$, which is exactly as large as necessary.

Remark 4 *An empirical study in this problem was given in a previous technical report (Lattimore, 2015a). In practice the failure does not become extreme until k is very large (approximately 1000).*

4. Finite-time analysis

In this section the remainder of Theorem 1 is proven. The argument is quite long, but never terribly complicated. The main novel challenge is to deal with the dependence of the index of one arm on the number of plays of other arms. The usual program for analysing strategies based on upper confidence bounds has two parts:

(a) Show that with high probability the index of the optimal arm is never much smaller than its mean.

(b) Show that the index of each suboptimal arm drops below the mean of the optimal arm after not too many plays with high probability.

The proof starts with (b), the main component of which is showing for suboptimal arms i that $H_i(t)$ grows at a reasonable rate. As discussed, this means showing that other arms are played sufficiently often. The following definitions spell out *which* arms will be played reasonably often. For each arm i

define a deterministic set of arms $V_i \subset [k]$ and random subset $W_i \subseteq V_i$ by

$$V_i = \{j \in [k] : j \text{ is even or } j \geq i\} \ . \tag{13}$$

$$W_i = \{i\} \cup \left\{ j \in V_i : \min_{1 \leq s \leq \Delta_i^{-2}} \hat{\mu}_{j,s} + \sqrt{\frac{2}{s} \log\left(\frac{n\Delta_i}{k_i \sqrt{s}}\right)} - \sqrt{\frac{2}{s}} \geq \mu_j \right\} \ . \tag{14}$$

The set W_i is a subset of V_i that includes arm i and those arms for which the empirical mean is always nearly as large as the true mean. We'll soon see that arms $j \in W_i$ will be played sufficiently often to ensure that $H_i(t)$ grows at the right rate. The exclusion of odd arms with $j < i$ from V_i is for technical reasons. In order to show that arm i is not played too often we need to show that its index drops sufficiently fast and that the index of some other arm is sufficiently large. The separation of the arms allows us to exploit the independence between the arms. Those arms not in V_i will be used to show that some index is large enough. Define

$$\delta_i = c_2 \sqrt{\frac{k_i}{n\Delta_i^2}} \qquad \text{and} \qquad \ell = \max\{i : \delta_i > 1/4\} \ . \tag{15}$$

It is shown in Lemma 21 in the appendix that there exists a universal constant $C > 0$ such that ℓ satisfies

$$n\bar{\Delta}_\ell + \sum_{m>\ell} \lambda_m \Delta_m \leq C \min_{i \in [k]} \left(n\bar{\Delta}_i + \sum_{m>i} \Delta_m \lambda_m \right) \ , \tag{16}$$

and so the rest of the proof is devoted to bounding the regret of ADA-UCB in terms of the left-hand-side of Eq. (16). Define F_i to be the event that the 'mass' of W_i is sufficiently large.

$$F_i = \mathbb{1}\left\{ \sum_{m \in W_i} \min\{1, \Delta_i/\Delta_m\} \geq k_i/8 \right\} \ . \tag{17}$$

Recall that $k_i = \sum_{m=1}^k \min\{1, \Delta_i/\Delta_m\}$. So F_i holds if the sum over the restricted set W_i is at most a factor of 8 smaller. The point is that if $j \in V_i$, then Lemma 12 implies $\mathbb{P}(j \notin W_i) \leq \delta_i \leq 1/4$. Later this will be combined with Hoeffding's bound to show that F_i occurs with high probability. And now the lemmas begin. First up is to show that arms $j \in W_i$ are played sufficiently often relative to arm i. This will then be used to show that $H_i(t)$ grows at the right rate, which leads to the conclusion of the proof of part (b) in the outline in Lemma 7.

Lemma 5 *If $A_t = i$ and $H_i(t-1) \leq k_i/\Delta_i^2$ and $T_i(t-1) \geq \zeta_i(\Delta_i) \vee 1/\Delta_i^2$, then for all $j \in W_i$,*

$$T_j(t-1) \geq \min\left\{ \frac{1}{2\Delta_j^2}, \frac{T_i(t-1)}{4\log\left(\frac{n}{H_i(t-1)}\right)} \right\} \ .$$

Proof If $i = j$ or $T_j(t-1) \geq T_i(t-1)$ or $T_j(t-1) \geq 1/(2\Delta_j^2)$, then we are done, so assume from now on that none of these are true.

$$\begin{aligned} \frac{k_i}{\Delta_i^2} \geq H_i(t-1) &= \sqrt{T_j(t-1)} \sum_{m=1}^k \min\left\{ \frac{T_i(t-1)}{\sqrt{T_j(t-1)}}, \sqrt{\frac{T_i(t-1)T_m(t-1)}{T_j(t-1)}} \right\} \\ &\geq \frac{\sqrt{T_j(t-1)}}{\Delta_i} \sum_{m=1}^k \min\left\{1, \sqrt{\frac{T_m(t-1)}{T_j(t-1)}}\right\} = \sqrt{T_j(t-1)} \frac{K_j(t-1)}{\Delta_i} \ , \end{aligned} \tag{18}$$

where the first inequality is assumed in the lemma statement and the second because $T_i(t-1) \geq T_j(t-1) \vee (1/\Delta_i^2)$. Therefore if $j \in W_i$, then

$$\begin{aligned} \mu_1 + \sqrt{\frac{2}{T_i(t-1)} \log\left(\frac{n}{H_i(t-1)}\right)} &\geq \hat{\mu}_i(t-1) + \sqrt{\frac{2}{T_i(t-1)} \log\left(\frac{n}{H_i(t-1)}\right)} \\ &= \gamma_i(t) \geq \gamma_j(t) = \hat{\mu}_j(t-1) + \sqrt{\frac{2}{T_j(t-1)} \log\left(\frac{n}{T_j(t-1)K_j(t-1)}\right)} \\ &\geq \hat{\mu}_j(t-1) + \sqrt{\frac{2}{T_j(t-1)} \log\left(\frac{n\Delta_i}{k_i\sqrt{T_j(t-1)}}\right)} \\ &\geq \mu_j + \sqrt{\frac{2}{T_j(t-1)}} \geq \mu_1 + \sqrt{\frac{1}{2T_j(t-1)}} \end{aligned} \tag{19}$$

where the first inequality follows from the assumption that $T_i(t-1) \geq \zeta_i(\Delta_i)$, which ensures that $\mu_1 = \mu_i + \Delta_i \geq \hat{\mu}_i(t-1)$. The second follows from Eq. (18). The inequalities in Eq. (19) from the definition of $j \in W_i$ and the assumption that $T_j(t-1) \geq 1/(2\Delta_j^2)$. Therefore

$$T_j(t-1) \geq \frac{T_i(t-1)}{4\log\left(\frac{n}{H_i(t-1)}\right)} \ . \qquad \blacksquare$$

The next lemma uses the previous result to show that if W_i is large enough (F_i holds), then $H_i(t-1)$ is reasonably large at the critical point when $T_i(t-1) \approx 1/\Delta_i^2$.

Lemma 6 *If F_i holds and $T_i(t-1) \geq \zeta_i(\Delta_i) \vee 128/\Delta_i^2$ and $A_t = i$, then*

$$\log\left(\frac{n}{H_i(t-1)}\right) \leq 2\log\left(\frac{n\Delta_i^2}{k_i}\right) \ .$$

Proof If $H_i(t-1) > k_i/\Delta_i^2$, then there is nothing more to do. Otherwise, by Lemma 5

$$\begin{aligned} H_i(t-1) &= \sum_{m=1}^k \min\left\{T_i(t-1), \sqrt{T_i(t-1)T_m(t-1)}\right\} \\ &\geq \sum_{m \in W_i} \min\left\{T_i(t-1), \sqrt{T_i(t-1)T_m(t-1)}\right\} \\ &\geq \sum_{m \in W_i} \min\left\{ T_i(t-1), \frac{T_i(t-1)}{2\log^{\frac{1}{2}}\left(\frac{n}{H_i(t-1)}\right)}, \frac{\sqrt{T_i(t-1)/2}}{\Delta_m} \right\} \qquad (20) \\ &\geq \frac{8\sum_{m \in W_i} \min\left\{\frac{1}{\Delta_i^2}, \frac{1}{\Delta_i\Delta_m}\right\}}{\log^{\frac{1}{2}}\left(\frac{n}{H_i(t-1)}\right)} \geq \frac{k_i}{\Delta_i^2 \log^{\frac{1}{2}}\left(\frac{n}{H_i(t-1)}\right)} \ , \qquad (21) \end{aligned}$$

where Eq. (20) follows from Lemma 5. The first inequality in Eq. (21) follows because $\log(x) \geq 1$ and the assumption $T_i(t-1) \geq 128/\Delta_i^2$, and the second because F_i holds. The result follows via

re-arrangement and some algebraic trickery with the $\overline{\log}(\cdot)$ function using Part (v) of Lemma 20 in the appendix. ■

For each arm i define random variable Λ_i that will be shown to be a high probability bound on $T_i(n)$ and approximately equal to λ_i in expectation.

$$\Lambda_i = 1 + \max\left\{\frac{128}{\Delta_i^2},\ \zeta_i(\Delta_i/3),\ \frac{36}{\Delta_i^2}\,\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right),\ \frac{18\mathbb{1}_{\{F_i^c\}}}{\Delta_i^2}\,\overline{\log}\left(n\Delta_i^2\right)\right\},$$

where $\zeta_i(\cdot)$ is defined in Eq. (8). The next lemma is a simple consequence of the previous two and shows that if $T_i(t-1)+1 \geq \Lambda_i$, then either arm i is not played or its index is smaller than the mean of the optimal arm by a margin of at least $\Delta_i/3$.

Lemma 7 *If $T_i(t-1)+1 \geq \Lambda_i$, then either $A_t \neq i$ or $\gamma_i(t) \leq \mu_1 - \Delta_i/3$.*

Proof If F_i does not occur, then $H_i(t-1) \geq T_i(t-1) \geq 1/\Delta_i^2$ and so

$$\gamma_i(t) = \hat{\mu}_i(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}{T_i(t-1)}} \leq \mu_i + \frac{\Delta_i}{3} + \sqrt{\frac{2\,\overline{\log}\left(n\Delta_i^2\right)}{T_i(t-1)}} \leq \mu_1 - \frac{\Delta_i}{3},$$

where the first inequality follows because $T_i(t-1)+1 \geq \Lambda_i \geq 1 + \zeta_i(\Delta_i/3)$ and the second because $H_i(t-1) = T_i(t-1)K_i(t-1) \geq T_i(t-1) \geq 1/\Delta_i^2$ and the definition of Λ_i and because $\mu_i + \Delta_i/3 = \mu_1 - 2\Delta_i/3$. Next suppose that F_i occurs, then either $A_t \neq i$ and the result is true, or $A_t = i$ and so by Lemma 6

$$\gamma_i(t) = \hat{\mu}_i(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}{T_i(t-1)}} \leq \mu_i + \frac{\Delta_i}{3} + \sqrt{\frac{4\,\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right)}{T_i(t-1)}} \leq \mu_1 - \frac{\Delta_i}{3}. \qquad ■$$

This essentially completes the first part of the proof by showing that if $T_i(t-1)+1 \geq \Lambda_i$, then arm i is either not played or its index is not too large. The next step is to show that with reasonable probability there is some near-optimal arm for which the index is big enough to prevent arm i from being played. The value of Λ_i has been carefully chosen to bound the number of times arm i can be played provided the index of some other arm is always larger than $\mu_1 - \Delta_i/3$. But more than this, Λ_i does not depend on the rewards for odd-index arms $j < i$ so is measurable with respect to the σ-algebra $\sigma(\hat{\mu}_{j,s} : j \in V_i, 1 \leq s \leq n)$. Define random variable $I \in [k]$ to be the arm with the largest mean such that there exists an odd $j < I+1$ with $\Delta_j \leq \Delta_{I+1}/6$ and

$$\min_{1\leq s\leq n} \hat{\mu}_{j,s} + \sqrt{\frac{2}{s}\,\overline{\log}\left(\frac{n}{sI + \sum_{m>I}\min\left\{s, \sqrt{s\Lambda_m}\right\}}\right)} > \mu_j - \frac{\Delta_{I+1}}{6}. \tag{22}$$

The definition of I implies that arms $i > I$ will not be played once their index drops far enough below the mean of the optimal arm. We should hope that I is small with reasonably high probability, with the best case being $I = 1$, which occurs when the optimal arm is always optimistic. Notice that I is well-defined because of the convention that $\Delta_{k+1} = \infty$. Let E_1 be the event that $I \leq \ell$, where ℓ is given in Eq. (15).

Lemma 8 *The regret of* ADA-UCB *is bounded by*

$$\mathcal{R}_n(\mu) \leq \mathbb{E}\left[\sum_{i>\ell}\Delta_i\Lambda_i\right] + n\mathbb{E}\left[\mathbb{1}_{\{E_1^c\}}\Delta_I\right] + \mathbb{E}\left[\mathbb{1}_{\{E_1\}}\sum_{i=1}^{\ell}\Delta_i T_i(n)\right].$$

Proof The first task is to show that $T_i(n) < \Lambda_i$ for all $i > I$, which follows by induction over rounds $k+1 \leq t \leq n$. Starting with the base case, note that when $t = k+1$ we have $T_i(t-1) = 1 < \Lambda_i$ for all i. Now suppose for $t \geq k+1$ that $T_i(t-1) < \Lambda_i$ for all $i > I$. By the definition of I there must exist an odd j with $\Delta_j \leq \Delta_{I+1}/6$ that satisfies Eq. (22). For this arm we have

$$\begin{aligned} H_j(t-1) &= \sum_{m=1}^{k}\min\left\{T_j(t-1), \sqrt{T_j(t-1)T_m(t-1)}\right\} \\ &< T_j(t-1)I + \sum_{m>I}\min\left\{T_j(t-1), \sqrt{T_j(t-1)\Lambda_m}\right\}. \end{aligned}$$

Therefore

$$\gamma_j(t) = \hat{\mu}_j(t-1) + \sqrt{\frac{2}{T_j(t-1)}\,\overline{\log}\left(\frac{n}{H_j(t-1)}\right)} > \mu_j - \Delta_{I+1}/6 \geq \mu_1 - \Delta_{I+1}/3.$$

It follows that if $i > I$ and $T_i(t-1)+1 \geq \Lambda_i$, then Lemma 7 implies $A_t \neq i$. Therefore $T_i(t) < \Lambda_i$ for all $i > I$, which completes the induction. Finally, since $\sum_{i=1}^k T_i(n) = n$ and using the definition of E_1 as the event that $I \leq \ell$, the regret may be bounded by

$$\mathcal{R}_n(\mu) = \mathbb{E}\left[\sum_{i=1}^{k}\Delta_i T_i(n)\right] \leq \mathbb{E}\left[\sum_{i>\ell}\Delta_i\Lambda_i\right] + n\mathbb{E}[\mathbb{1}_{\{E_1^c\}}\Delta_I] + \mathbb{E}\left[\mathbb{1}_{\{E_1\}}\sum_{i=1}^{\ell}\Delta_i T_i(n)\right]. \qquad ■$$

The last step in the proof of Theorem 1 is to bound each of the expectations in Lemma 8.

Lemma 9 *There exists a universal $C > 0$ such that:*

$$\textit{(i)}\ \ \mathbb{E}\left[\sum_{i>\ell}\Delta_i\Lambda_i\right] \leq C\sum_{i>\ell}\Delta_i\lambda_i \qquad \textit{(ii)}\ \ \mathbb{E}[\mathbb{1}_{\{E_1^c\}}\Delta_I] \leq C\left(n\bar{\Delta}_\ell + \sum_{i>\ell}\Delta_i\lambda_i\right)$$

$$\textit{(iii)}\ \ \mathbb{E}\left[\mathbb{1}_{\{E_1\}}\sum_{i=1}^{\ell}\Delta_i T_i(n)\right] \leq C\left(n\bar{\Delta}_\ell + \sum_{i>\ell}\Delta_i\lambda_i\right).$$

The proof of part *(i)* follows shortly. The proof of part *(ii)* is deferred to Appendix B. Very briefly, it follows somewhat directly from part (a) of Lemma 12 (notice the similarity between the lemma and Eq. (22) that defines Δ_I). Part *(iii)* would be trivial if one assumed that $\mathbb{E}[T_i(n)]$ is monotone non-increasing in i. This seems likely, but despite significant effort I was only able to show that this is approximately true using a complicated proof (details are in Appendix C).

Proof [of Lemma 9 (*i*)] The proof follows by bounding $\mathbb{E}[\Lambda_i]$ for each arm $i > \ell$. Naively bounding the max in the definition of Λ_i by a sum shows that

$$\mathbb{E}[\Lambda_i] \leq 1 + \frac{128}{\Delta_i^2} + \frac{36}{\Delta_i^2}\,\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right) + \mathbb{E}\left[\zeta_i\left(\frac{\Delta_i}{3}\right)\right] + \frac{18\mathbb{P}\left(F_i^c\right)}{\Delta_i^2}\,\overline{\log}\left(n\Delta_i^2\right). \tag{23}$$

The first three terms are non-random. The second last term is bounded using Lemma 13 by $\mathbb{E}[\zeta_i(\Delta_i/3)] \leq 1 + 18/\Delta_i^2$. For the last term we need to upper bound $\mathbb{P}(F_i^c)$, where F_i is the event defined in Eq. (17). In order to do this we need to show that W_i is reasonably large with high probability. Let $\chi_j = \mathbb{1}\{j \notin W_i\}$, which for $j \in V_i$ satisfies $\mathbb{E}[\chi_j] \leq \delta_i \leq 1/4$. Then

$$\mathbb{P}(F_i^c) = \mathbb{P}\left(\sum_{j \in W_i} \min\left\{1, \frac{\Delta_i}{\Delta_j}\right\} < \frac{k_i}{8}\right) \leq \mathbb{P}\left(\sum_{j \in V_i} (\chi_j - \mathbb{E}[\chi_j]) \min\left\{1, \frac{\Delta_i}{\Delta_j}\right\} > \frac{k_i}{4}\right)$$
$$\leq \exp\left(-\frac{k_i^2}{8\sum_{j \in V_i} \min\left\{1, \frac{\Delta_i}{\Delta_j}\right\}^2}\right) \leq \exp\left(-\frac{k_i}{8}\right),$$

where the first inequality follows from the facts that $\mathbb{E}[\chi_j] \leq 1/4$ and $\sum_{j \in V_i} \min\{1, \Delta_i/\Delta_j\} \geq k_i/2$ and the second inequality follows from Hoeffding's bound and the fact that χ_j are independent for $j \in V_i$. Therefore

$$\mathbb{P}(F_i^c)\,\overline{\log}\left(n\Delta_i^2\right) \leq \overline{\log}\left(n\Delta_i^2\right)\exp\left(-k_i/8\right) \leq 4\,\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right),$$

which holds because $k_i \geq 2$ is guaranteed for all suboptimal arms i. The proof is completed by substituting the above display into Eq. (23) and using the definition of $\lambda_i = 1 + \frac{1}{\Delta_i^2}\,\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right)$. ■

Proof [of Theorem 1] The finite-time bound Eq. (3) follows by substituting the bounds given in Lemma 9 into Lemma 8 and Eq. (16). Minimax and sub-UCB results are derived as corollaries of the finite-time bound Eq. (3) via Parts (iii) and (iv) of Lemma 21 in the appendix. The asymptotic analysis has been given already in §3. ■

5. A lower bound

I now formalise the intuitive argument for the regret guarantee given in §2. The results show that in a certain sense the upper bound in Theorem 1 is very close to optimal. The following lower bound holds for all strategies, but does not give a lower bound for all μ simultaneously. Related results have been proven by a variety of authors (Kulkarni and Lugosi, 2000; Bubeck et al., 2013; Salomon et al., 2013; Lattimore, 2015b), with the most related by Garivier et al. (2016b). The most significant difference between that work and the present article is that the lower-order terms are more carefully considered here, and besides this, the assumptions, and so also results, are different.

Theorem 10 *Fix a strategy and let $\mu \in \mathbb{R}^k$ be such that $n\Delta_i^2 \geq 2k_i \log(n)$ and $\Delta_i \leq 1$ for all i with $\Delta_i > 0$. Then one of the following holds:*

(a) $\mathcal{R}_n(\mu) \geq \frac{1}{2}\sum_{i:\Delta_i>0} \frac{1}{\Delta_i}\log\left(\frac{n\Delta_i^2}{2k_i\log(n)}\right)$.

(b) There exists a $\mu' \in \mathbb{R}^k$ and i with $\Delta_i > 0$ such that $\mathcal{R}_n(\mu') \geq \frac{1}{4}\sum_{i:\Delta_i'>0} \frac{1}{\Delta_i'}\log(n)$,

where $\mu_i' = \mu_i + 2\Delta_i$ and $\mu_j' = \mu_j$ for $j \neq i$ and $\Delta_i' = \Delta_i(\mu')$.

Proof Suppose that (a) does not hold, then there exists a suboptimal arm i such that

$$\mathbb{E}[T_i(n)] \leq \frac{1}{2\Delta_i^2}\log\left(\frac{n\Delta_i^2}{2k_i\log(n)}\right). \tag{24}$$

Let μ' be as defined in the second part of the lemma and write $\mathbb{P}'$ and $\mathbb{E}'$ for expectation when rewards are sampled from μ'. Then by Lemmas 18 and 19 in Appendix D we have

$$\mathbb{P}(T_i(n) \geq n/2) + \mathbb{P}'(T_i(n) < n/2) \geq \frac{k_i\log(n)}{n\Delta_i^2} \triangleq 2\delta.$$

By Markov's inequality and Eq. (24) and the fact that $k_i \geq 2$,

$$\mathbb{P}(T_i(n) \geq n/2) \leq \frac{2\mathbb{E}[T_i(n)]}{n} \leq \frac{1}{n\Delta_i^2}\log\left(\frac{n\Delta_i^2}{2k_i\log(n)}\right) \leq \frac{k_i\log(n)}{2n\Delta_i^2} = \delta.$$

Therefore $\mathbb{P}'(T_i(n) < n/2) \geq \delta$, which implies that

$$\mathcal{R}_n(\mu') \geq \frac{\delta n\Delta_i}{2} = \frac{1}{4}\sum_{j=1}^{k} \min\left\{\frac{1}{\Delta_i}, \frac{1}{\Delta_j}\right\}\log(n) \geq \frac{1}{4}\sum_{j:\Delta_j'>0}\frac{1}{\Delta_j'}\log(n).$$ ■

A conjecture is false It was conjectured by Bubeck and Cesa-Bianchi (2012) that the optimal regret might have approximately the following form.

$$\mathcal{R}_n(\mu) \leq C\sum_{i:\Delta_i>0}\left(\Delta_i + \frac{1}{\Delta_i}\,\overline{\log}\left(\frac{n}{H}\right)\right) \qquad \text{for all } \mu \text{ and } n \text{ and } k, \tag{25}$$

where $C > 0$ is a universal constant and $H = \sum_{i:\Delta_i>0}\Delta_i^{-2}$ is a quantity that appears in the best-arm identification literature (Bubeck et al., 2009; Audibert and Bubeck, 2010; Jamieson et al., 2014).

Theorem 11 *There does not exist a strategy for which Eq.* (25) *holds.*

Proof Let $k \geq 2$ and $\mu_1 = 0$ and $\mu_2 = -1/k$ and $\mu_i = -1$ for $i > 2$, which implies that $H = k^2 + k - 2 \geq n$. For the rest of the proof we view the horizon $n = k^2$ to be a function of k. Suppose that $\mathcal{R}_n(\mu) = o(k\log k)$, which must be true for any strategy witnessing Eq. (25). Then $\min_{i>2}\mathbb{E}[T_i(n)] = o(\log k)$. Let $i = \arg\min_{i>2}\mathbb{E}[T_i(n)]$ and define μ' to be equal to μ except for the ith coordinate, which has $\mu_i' = 1$. Let A be the event that $T_i(n) \geq n/2$ and let $\mathbb{P}$ and $\mathbb{P}'$ be measures on the space of outcomes induced by the interaction between the fixed strategy and the bandits determined by μ and μ' respectively. Then for all $\varepsilon > 0$,

$$\mathcal{R}_n(\mu) + \mathcal{R}_n(\mu') \geq \frac{n}{2}\left(\mathbb{P}(A) + \mathbb{P}'(A^c)\right) \geq \frac{n}{4}\exp\left(-\operatorname{KL}(\mathbb{P}, \mathbb{P}')\right)$$
$$= \frac{k^2}{4}\exp\left(-2\mathbb{E}[T_i(n)]\right) = \omega(k^{2-\varepsilon}),$$

By the assumption on $\mathcal{R}_n(\mu)$ and for suitably small ε we have $\mathcal{R}_n(\mu') = \omega(k^{2-\varepsilon})$. But as the number of arms $k \to \infty$ (and so also the horizon), this cannot be true for any policy satisfying Eq. (25), or even Eq. (1). Therefore the conjecture is not true. ■

6. Empirical evaluation

ADA-UCB is compared to UCB (Katehakis and Robbins, 1995), MOSS (Ménard and Garivier, 2017), THOMPSON SAMPLING (Agrawal and Goyal, 2012) and IMED (Honda and Takemura, 2015),[1] where the reference indicates the source of the algorithm. All algorithms choose each arm once and subsequently:

$$A_t^{\text{IMED}} = \arg\min_{i\in[k]} \frac{T_i(t-1)}{2}(\hat{\mu}_i(t-1) - \max_{j\in[k]} \hat{\mu}_j(t-1))^2 + \log(T_i(t-1))\,.$$

$$A_t^{\text{UCB}} = \arg\max_{i\in[k]} \hat{\mu}_i(t-1) + \sqrt{\frac{2\log(t)}{T_i(t-1)}}\,.$$

$$A_t^{\text{MOSS}} = \arg\max_{i\in[k]} \hat{\mu}_i(t-1) + \sqrt{\frac{2}{T_i(t-1)}\log_+\left(\frac{n}{kT_i(t-1)}\log_+^2\left(1+\frac{n}{kT_i(t-1)}\right)\right)}\,.$$

$$A_t^{\text{TS}} = \arg\max_{i\in[k]} \theta_i(t) \qquad \text{with } \theta_i(t) \sim \mathcal{N}(\hat{\mu}_i(t-1), 1/T_i(t-1))\,.$$

The logarithmic term used by Ménard and Garivier (2017) in their version of MOSS is larger than $\overline{\log}(\cdot)$ and this negatively affects its performance. If the variant proposed in Section 3 is used instead, then it becomes comparable to ADA-UCB on the experiments described below, but still fails on the computationally expensive experiment given in the previous technical report (Lattimore, 2015a). For all other algorithms I have chosen the variant for which (a) guarantees exist and (b) the empirical performance is best. In all plots N indicates the number of independent samples per data point and confidence bands are calculated at a 95% level. The first three plots in Figure 1 show the regret in the worst case regime where all suboptimal arms have the same suboptimality gap. Unsurprisingly the relative advantage of policies with well-tuned confidence levels increases with the number of arms. At its worst, UCB suffers about three times the regret of ADA-UCB. Coincidentally, $\sqrt{\log(10^4)} \approx 3.03$. Figure 2 shows the regret as a function of the horizon n on a fixed bandit with $k = 20$ arms (see caption of figure for means). The regret of ADA-UCB is again a little better than the alternatives.

7. Discussion

Anytime strategies The ADA-UCB strategy depends on the horizon n, which may sometimes be unknown. The natural idea is to replace n by t, which indeed leads to a reasonable strategy that enjoys the same guarantees as ADA-UCB provided the $\overline{\log}(\cdot)$ function is replaced by something fractionally larger. The analysis, however, is significantly longer and is not included. Interested readers may refer to the technical report (Lattimore, 2016b) for the core ideas, but more work is required to find a clean proof.

Multiple optimal arms The finite-time bound in Theorem 1 is the first that demonstrates an improvement when there are multiple (near-)optimal arms. The gain in terms of the expected regret is not very large because k_i (which grows as optimal arms are added) appears only in the denominator of the logarithm. There is, however, a more significant advantage when there are many optimal arms, which (up to a point) is an exponential decrease in the variance of most strategies. This can be

1. IMED is usually defined for bandits where the rewards have (semi-)bounded support, but Junya Honda kindly provided unpublished details of the adaptation to the Gaussian case.

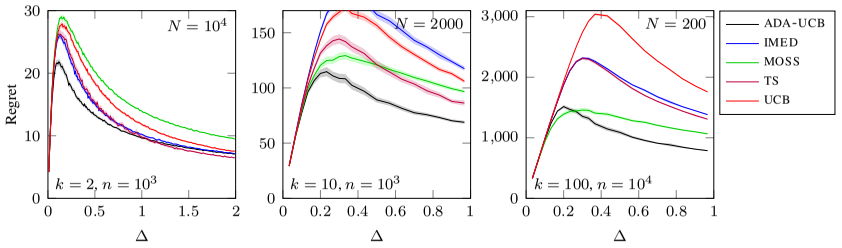


Figure 1: The regret of various algorithms as a function of Δ when $\mu = (\Delta, 0, \ldots, 0)$ and the number of arms is 2, 10 and 100 respectively. The y-axis shows the regret averaged over N independent samples for each data point.

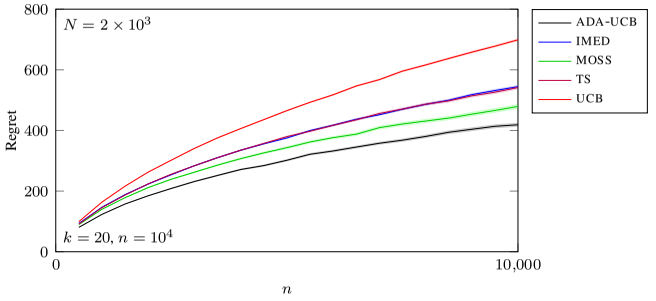


Figure 2: The regret of various algorithms as a function of the horizon for the Gaussian bandit with $k = 20$ arms and payoff vector $\mu = (0, -0.03, -0.03, -0.07, -0.07, -0.07, -0.15, -0.15, -0.15, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -0.5, -1, -1)$. ADA-UCB is again outperforming the competitors. Note that IMED and THOMPSON SAMPLING are so similar they cannot be distinguished in the plot.

extracted from the analysis by observing that the high variance is caused by the possibility that an optimal arm is not sufficiently optimistic, but the probability of this occurring drops exponentially as the number of optimal arms increases.

Alternative noise models and other extensions The most obvious open question is how to generalise the results to a broader class of noise models and setups. I am quite hopeful that this is possible for noise from exponential families, though the analysis will necessarily become more complicated because the divergences become more cumbersome to work with than the squared distance that is the divergence in the Gaussian case. An alternative direction is to consider the

situation where the variance is also unknown, which has seen surprisingly little attention, but is now understood reasonably well (Honda and Takemura, 2014; Cowan et al., 2015). At the very least, the concentration analysis using Brownian motion could be applied, but I expect an adaptive confidence level will also yield theoretical and practical improvements. Another potential application of the ideas presented here would be to try and port them into other bandit strategies that depend on a confidence level such as BAYES-UCB (Kaufmann, 2016), or even linear bandits (Dani et al., 2008; Abbasi-Yadkori et al., 2011; Lattimore and Szepesvari, 2017).

References

Yasin Abbasi-Yadkori, Csaba Szepesvári, and David Tax. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2312–2320, 2011.

Rajeev Agrawal. Sample mean based index policies with O(log n) regret for the multi-armed bandit problem. *Advances in Applied Probability*, pages 1054–1078, 1995.

Rajeev Agrawal, Demosthenis Teneketzis, and Venkatachalam Anantharam. Asymptotically efficient adaptive allocation schemes for controlled i.i.d. processes: Finite parameter space. *IEEE Transaction on Automatic Control*, 34:258–267, 1989.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *Proceedings of Conference on Learning Theory (COLT)*, 2012.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *Proceedings of Conference on Learning Theory (COLT)*, pages 217–226, 2009.

Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *Proceedings of Conference on Learning Theory (COLT)*, 2010.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Tuning bandit algorithms in stochastic environments. In *Algorithmic Learning Theory (ALT)*, pages 150–165. Springer, 2007.

Peter Auer and Ronald Ortner. UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1-2):55–65, 2010.

Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Foundations of Computer Science, 1995. Proceedings., 36th Annual Symposium on*, pages 322–331. IEEE, 1995.

Peter Auer, Nicoló Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. OUP Oxford, 2013.

Sébastien Bubeck and Nicolò Cesa-Bianchi. *Regret Analysis of Stochastic and Nonstochastic Multi-armed Bandit Problems*. Foundations and Trends in Machine Learning. Now Publishers Incorporated, 2012. ISBN 9781601986269.

Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in multi-armed bandits problems. In *Algorithmic Learning Theory (ALT)*, pages 23–37. Springer, 2009.

Sébastien Bubeck, Vianney Perchet, and Philippe Rigollet. Bounded regret in stochastic multi-armed bandits. *arXiv preprint arXiv:1302.1611*, 2013.

Apostolos N Burnetas and Michael N Katehakis. Optimal adaptive policies for sequential allocation problems. *Advances in Applied Mathematics*, 17(2):122–142, 1996.

Olivier Cappé, Aurélien Garivier, Odalric-Ambrym Maillard, Rémi Munos, and Gilles Stoltz. Kullback–Leibler upper confidence bounds for optimal sequential allocation. *The Annals of Statistics*, 41(3):1516–1541, 2013.

Nicolo Cesa-Bianchi and Paul Fischer. Finite-time regret bounds for the multiarmed bandit problem. In *ICML*, pages 100–108. Citeseer, 1998.

Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. Boltzmann exploration done right. In *Advances in Neural Information Processing Systems*, pages 6284–6293, 2017.

Wesley Cowan, Junya Honda, and Michael N Katehakis. Normal bandits of unknown means and variances: Asymptotic optimality, finite horizon regret bounds, and a solution to an open problem. *arXiv preprint arXiv:1504.05823*, 2015.

Varsha Dani, Thomas P Hayes, and Sham M Kakade. Stochastic linear optimization under bandit feedback. In *Proceedings of Conference on Learning Theory (COLT)*, pages 355–366, 2008.

Rémy Degenne and Vianney Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In *Proceedings of International Conference on Machine Learning (ICML)*, 2016.

Aurélien Garivier. Informational confidence bounds for self-normalized averages and applications. *arXiv preprint arXiv:1309.3376*, 2013.

Aurélien Garivier and Emilie Kaufmann. Optimal best arm identification with fixed confidence. In *Conference on Learning Theory (COLT)*, 2016.

Aurélien Garivier, Emilie Kaufmann, and Tor Lattimore. On explore-then-commit strategies. In *Neural Information Processing Systems (NIPS)*, 2016a.

Aurélien Garivier, Pierre Ménard, and Gilles Stoltz. Explore first, exploit next: The true shape of regret in bandit problems. *arXiv preprint arXiv:1602.07182*, 2016b.

Sébastien Gerchinovitz and Tor Lattimore. Refined lower bounds for adversarial bandits. *arXiv preprint arXiv:1605.07416*, 2016.

John Gittins, Kevin Glazebrook, and Richard Weber. *Multi-armed bandit allocation indices*. John Wiley & Sons, 2011.

Eli Gutin and Vivek Farias. Optimistic gittins indices. In *Advances in Neural Information Processing Systems*, pages 3153–3161, 2016.

Takeyuki Hida. Brownian motion. In *Brownian Motion*, pages 44–113. Springer, 1980.

Junya Honda and Akimichi Takemura. An asymptotically optimal bandit algorithm for bounded support models. In *Proceedings of Conference on Learning Theory (COLT)*, pages 67–79, 2010.

Junya Honda and Akimichi Takemura. An asymptotically optimal policy for finite support models in the multiarmed bandit problem. *Machine Learning*, 85(3):361–391, 2011.

Junya Honda and Akimichi Takemura. Optimality of thompson sampling for gaussian bandits depends on priors. In *Artificial Intelligence and Statistics*, pages 375–383, 2014.

Junya Honda and Akimichi Takemura. Non-asymptotic analysis of a new bandit algorithm for semi-bounded rewards. *The Journal of Machine Learning Research*, 16(1):3721–3756, 2015.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'UCB: An optimal exploration algorithm for multi-armed bandits. In *Proceedings of Conference on Learning Theory (COLT)*, 2014.

Michael N Katehakis and Herbert Robbins. Sequential choice from several populations. *Proceedings of the National Academy of Sciences of the United States of America*, 92(19):8584, 1995.

Emilie Kaufmann. On Bayesian index policies for sequential resource allocation. *arXiv preprint arXiv:1601.01190*, 2016.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On Bayesian upper confidence bounds for bandit problems. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 592–600, 2012.

Nathaniel Korda, Emilie Kaufmann, and Rémi Munos. Thompson sampling for 1-dimensional exponential family bandits. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1448–1456, 2013.

Sanjeev R Kulkarni and Gábor Lugosi. Finite-time lower bounds for the two-armed bandit problem. *Automatic Control, IEEE Transactions on*, 45(4):711–714, 2000.

Tze Leung Lai. Adaptive treatment allocation and the multi-armed bandit problem. *The Annals of Statistics*, pages 1091–1114, 1987.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

Tor Lattimore. Optimally confident UCB: Improved regret for finite-armed bandits. *arXiv preprint arXiv:1507.07880*, 2015a.

Tor Lattimore. The pareto regret frontier for bandits. In *Proceedings of the 28th Conference on Neural Information Processing Systems (NIPS)*, 2015b.

Tor Lattimore. Regret analysis of the finite-horizon Gittins index strategy for multi-armed bandits. In *Proceedings of Conference on Learning Theory (COLT)*, 2016a.

Tor Lattimore. Regret analysis of the anytime optimally confident UCB algorithm. Technical report, 2016b.

Tor Lattimore and Csaba Szepesvari. The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 728–737, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR.

Tor Lattimore and Csaba Szepesvári. *Bandit Algorithms*. 2018.

Hans R Lerche. *Boundary crossing of Brownian motion: Its relation to the law of the iterated logarithm and to sequential analysis*. Springer, 1986.

Pierre Ménard and Aurélien Garivier. A minimax and asymptotically optimal algorithm for stochastic bandits. *arXiv preprint arXiv:1702.07211*, 2017.

Dan Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. In *Advances in Neural Information Processing Systems*, pages 1583–1591, 2014.

Antoine Salomon, Jean-Yves Audibert, and Issam El Alaoui. Lower bounds and selectivity of weak-consistent policies in stochastic multi-armed bandit problem. *Journal of Machine Learning Research*, 14(Jan):187–207, 2013.

Alexandre B Tsybakov. *Introduction to nonparametric estimation*. Springer Science & Business Media, 2008.

Walter Vogel. An asymptotic minimax theorem for the two armed bandit problem. *The Annals of Mathematical Statistics*, 31(2):444–451, 1960.

Appendix A. Boundary crossing for Gaussian random walks

Let $Z_1, Z_2, \ldots$ be an infinite sequence of independent standard Gaussian random variables and $S_n = \sum_{t=1}^{n} Z_t$. The proof of Theorem 1 relies on a precise understanding of the behaviour of the random walk $(S_n)_n$. More specifically, what is the hitting probability that S_n ever crosses above a carefully chosen concave boundary. The following lemma is an easy consequence of the elegant analysis of boundary crossing probabilities for Brownian motion by Lerche (1986).

Lemma 12 *Let $d \in \{1, 2, \ldots\}$ and $\Delta > 0$, $\alpha > 0$, $\lambda \in [0, \infty]^d$ and $h_\lambda(t) = \sum_{i=1}^{d} \min\{t, \sqrt{t\lambda_i}\}$, then there exist constants $c_1 = 4$ and $c_2 = 12$ such that*

(a) $\mathbb{P}\left(\text{exists } t \geq 0 : S_t \geq \sqrt{2t \log\left(\frac{\alpha}{h_\lambda(t)}\right)} + t\Delta\right) \leq \frac{c_1 h_\lambda(1/\Delta^2)}{\alpha}$.

(b) $\mathbb{P}\left(\text{exists } t \leq \frac{1}{\Delta^2} : S_t \geq \sqrt{2t \log\left(\frac{\alpha}{t^{1/2}}\right)} - \sqrt{2t}\right) \leq \frac{c_2}{\sqrt{\alpha}\Delta}$.

The second lemma is a bound on the expected number of samples required before the empirical mean after t samples is close to its true value. The result is relatively standard in the literature, except that here the proofs are simplified by using the properties of Brownian motion.

Lemma 13 *If $\Delta > 0$ and $\zeta = 1 + \max\left\{t : \frac{S_t}{t} \geq \Delta\right\}$, then $\mathbb{E}[\zeta] \leq 1 + \frac{2}{\Delta^2}$.*

Subgaussian case A common relaxation of the Gaussian noise assumption is to assume the noise is 1-subgaussian, which means the reward X_t is chosen so that $\mathbb{E}[\exp(c(X_t - \mu_{A_t})) \mid \mathcal{F}_{t-1}] \leq \exp(c^2/2)$ almost surely for all $c \in \mathbb{R}$. Brownian motion cannot be used to analyse this situation, but Lemma 12 can still be proven for martingale subgaussian noise using the peeling trick on a carefully optimised grid (as used by Garivier (2013) and others). Besides a messier proof, the price is that the $\overline{\log}$ function must be increased slightly (but not so much that Theorem 1 needs to change). Lemma 13 is also easily adapted to the subgaussian setting. The only other lemma that needs modification is Lemma 15 in the appendix, which has the same flavour as Lemma 12 and is adaptable via a peeling trick.

The tangent approximation The connection to Brownian motion is made by noting the discrete time random walk S_t can be embedded in Brownian motion, which means that if B_t is a standard Brownian motion, then for any function $f : \mathbb{R} \to \mathbb{R}$ we have $\mathbb{P}(\text{exists } n \geq 0 : S_n \geq f(n)) \leq \mathbb{P}(\text{exists } n \geq 0 : B_n \geq f(n))$. The main tool of the analysis is called the tangent approximation, which was developed in a beautiful book by Lerche (1986) and is summarised in the following lemma.

Lemma 14 (§3 of Lerche (1986)) *Let $f : \mathbb{R} \to \mathbb{R}$ be a concave function with $f(x) \geq 0$ for all $x \geq 0$ and $\Lambda(t) = f(t) - tf'(t)$ be the intersection of the tangent to f at t with the y-axis, where if f is non-differentiable, then f' denotes any 'super-derivative' (gradient such that the tangent does not intersect the curve). Then*

$$\mathbb{P}\left(\textit{exists } t \geq 0 : B_t \geq f(t)\right) \leq \int_0^\infty \frac{\Lambda(t)}{\sqrt{2\pi t^3}} \exp\left(-\frac{f(t)^2}{2t}\right) dt\,.$$

Proof [of Lemma 12] Let B_t be a standard Brownian motion. Each part of the lemma will follow by analysing the probability that the Brownian motion hits the relevant boundary. For the first part let $f(t) = \sqrt{2t\,\overline{\log}(\alpha/h_\lambda(t))}$, which by simple calculus is monotone non-decreasing. Therefore the intersection of the tangent to $f(t) + t\Delta$ at t with the y-axis is $\Lambda(t) = f(t) - tf'(t) \leq f(t)$. By the tangent approximation:

$$\mathbb{P}(\exists t \geq 0 : B_t \geq f(t)) \leq \int_0^\infty \frac{f(t)}{\sqrt{2\pi t^3}} \exp\left(-\frac{(f(t)+t\Delta)^2}{2t}\right) dt \leq \int_0^\infty \frac{3h_\lambda(t)}{2\alpha\sqrt{\pi t}} \exp\left(-\frac{t\Delta^2}{2}\right) dt\,,$$

where the first inequality follows from Lemma 14 and the second from (iv) of Lemma 20 and since for positive $x, y \geq 0$ it holds that $(x+y)^2 \geq x^2 + y^2$.

$$\begin{aligned}
\int_0^\infty \frac{h_\lambda(t)}{t} \exp\left(-\frac{t\Delta^2}{2}\right) dt &= \sum_{i=1}^d \int_0^\infty \min\left\{1, \sqrt{\lambda_i/t}\right\} \exp\left(-\frac{t\Delta^2}{2}\right) dt \\
&= \sum_{i=1}^d \left(\frac{2 - 2\exp\left(-\frac{\lambda_i\Delta^2}{2}\right)}{\Delta^2} + \frac{\sqrt{2\pi\lambda_i}}{\Delta} \operatorname{erfc}\left(\Delta\sqrt{\frac{\lambda_i}{2}}\right)\right) \\
&\leq \sum_{i=1}^d \min\left\{ \frac{2+\sqrt{\pi}}{\Delta^2}, (1+\sqrt{2\pi})\frac{\sqrt{\lambda_i}}{\Delta} \right\} \leq (2+\sqrt{\pi}) h_\lambda(1/\Delta^2)\,,
\end{aligned}$$

where first inequality follows from the fact that $1 - e^{-x} \leq x$ and $\operatorname{erfc}(x) \leq \min\{1, 1/(2x)\}$. The result is completed by naively bounding the constants. For the second part, let $f(t) = \sqrt{2t\,\overline{\log}(\alpha/t^{1/2})} - \sqrt{2t}$, which is concave and monotone non-decreasing so that $\Lambda(t) = f(t) - tf'(t) \leq f(t)$. By Lemma 14,

$$\begin{aligned}
\mathbb{P}\left(\exists t \leq \frac{1}{\Delta^2} : B_t \geq \sqrt{2t\,\overline{\log}\left(\frac{\alpha}{s^{1/2}}\right)} - \sqrt{2t}\right) &\leq \int_0^{\frac{1}{\Delta^2}} \frac{f(t)}{\sqrt{2\pi t^3}} \exp\left(-\frac{f(t)^2}{2t}\right) dt \\
&\leq \int_0^{\frac{1}{\Delta^2}} \sqrt{\frac{\overline{\log}(\alpha/t^{1/2})}{\pi t^2}} \exp\left(-1 - \overline{\log}\left(\frac{\alpha}{t^{1/2}}\right) + 2\sqrt{\overline{\log}\left(\frac{\alpha}{t^{1/2}}\right)}\right) dt \\
&\leq \int_0^{\frac{1}{\Delta^2}} \frac{5dt}{\sqrt{\pi}\alpha^{1/2}t^{3/4}} \leq \frac{12}{\sqrt{\alpha\Delta}}\,,
\end{aligned}$$

where the second last inequality follows from part (vi) of Lemma 20. ■

Proof [of Lemma 13] The time inversion formula and reflection principle will imply the result (Hida, 1980, for example). Let B_s be a standard Brownian motion. Then for $t > 0$ we have

$$\mathbb{P}(\text{exists } s \geq t : B_s/s \geq \Delta) = \mathbb{P}(\text{exists } s \leq 1/t : B_s \geq \Delta) = 2\mathbb{P}\left(B_{1/t} \geq \Delta\right) \leq \exp\left(-\frac{t\Delta^2}{2}\right)\,,$$

where the first equality follows from the time inversion formula and the second from the reflection principle. The inequality is a standard Gaussian tail bound (Boucheron et al., 2013, Chap. 2). Therefore $\mathbb{E}[\zeta] \leq 1 + \int_0^\infty \exp(-t\Delta^2/2)dt = 1 + \frac{2}{\Delta^2}$, where the additive constant is due to the embedding of the discrete random walk in the continuous Brownian motion. ■

Appendix B. Proof of Lemma 9 *(ii)*

Some of the steps in this proof are simplified by using $C > 0$ for a universal positive constant that occasionally has a different value from one equation to the next. When these changes occur they are indicated by the $\overset{\times}{\leq}$ symbol. Let $i \in [k]$ and $j \notin V_i$ be such that $\Delta_j \leq \Delta_i/6$. By part (a) of Lemma 12,

$$\mathbb{P}\left(\min_{1\leq t\leq n} \gamma_j(t) \leq \mu_j - \frac{\Delta_i}{6}\Big|\Lambda_i, \ldots, \Lambda_k\right) \leq \frac{36c_1}{n}\left(\frac{i-1}{\Delta_i^2} + \sum_{m\geq i}\frac{\sqrt{\Lambda_m}}{\Delta_i}\right)\,.$$

It is important to note here that Lemma 12 could only be applied to control the conditional probability above because the random variables $\Lambda_i, \ldots, \Lambda_k$ are independent of $\hat{\mu}_{j,s}$ for all $1 \leq s \leq n$. Let

$$\Psi_i = \min\left\{1,\ \frac{36c_1}{n}\left(\frac{i-1}{\Delta_i^2} + \sum_{m\geq i}\frac{\sqrt{\Lambda_m}}{\Delta_i}\right)\right\}\,.$$

Let $m_i = \sum_{j\notin V_i} \mathbb{1}\{\Delta_j \leq \Delta_i/6\}$ be the number of arms that might satisfy Eq. (22) in the definition of I. Then by the previous display and the definition of I, $\mathbb{P}(I \geq i|\Lambda_i, \ldots, \Lambda_k) \leq \Psi_i^{m_i}$. It would be tempting to try and bound the expectation of Δ_I by taking a union bound over all arms, but this is not tight when many arms have nearly the same mean. Let $\mathcal{I} \subset [k]$ be empty if $i_1 = \ell + 1 > k$.

Otherwise $\mathcal{I} = \{i_1, i_2, \ldots, i_b\}$ where and $i_{j+1} = \min\left\{i : \Delta_i > 6\Delta_{i_j}\right\}$ and b is as large as possible so that $i_b \leq k$.

$$\mathbb{E}[\Delta_I] \leq 6\sum_{i\in\mathcal{I}} \mathbb{P}\left(I \geq i\right)\Delta_i = 6\sum_{i\in\mathcal{I}} \mathbb{E}\left[\mathbb{P}\left(I \geq i \middle| \Lambda_i, \ldots, \Lambda_k\right)\Delta_i\right]$$
$$\leq 6\sum_{i\in\mathcal{I}} \mathbb{E}\left[\Psi_i^{m_i}\Delta_i\right] = 6\mathbb{E}\left[\sum_{i\in\mathcal{I}} \Psi_i^{m_i}\Delta_i\right] \overset{\times}{\leq} C\mathbb{E}\left[\max_{i\in\mathcal{I}} \Psi_i^{m_i}\Delta_i\right].$$

Only the last step above is non-trivial. It follows by choosing a as the smallest value such that $\Psi_{i_a} < 1/2$ (or $a = b$ if such a choice does not exist). Then the contribution of $\Psi_i^{m_i}\Delta_i$ is decreasing exponentially in both directions away from i_a by the definition of $\mathcal{I}$ and the fact that $m_{i_{a+2}} \geq m_{i_a} + 1$.

Case 1 ($\Psi_i \geq 1/2$) By the definition of ℓ and the fact that $i > \ell$ we have $\delta_i \leq 1/4$ and so by Eq. (15), $k_i \leq n\Delta_i^2/(16c_2^2) \leq n\Delta_i^2(144c_1)$. Therefore

$$\frac{1}{2} \leq \frac{36c_1}{n}\left(\frac{i-1}{\Delta_i^2} + \sum_{m\geq i}\frac{\sqrt{\Lambda_m}}{\Delta_i}\right) \leq \frac{36c_1}{n}\left(\frac{k_i}{\Delta_i^2} + \sum_{m\geq i}\frac{\sqrt{\Lambda_m}}{\Delta_i}\right) \leq \frac{1}{4} + \frac{36c_1}{n}\sum_{m\geq i}\frac{\sqrt{\Lambda_m}}{\Delta_i}.$$

Rearranging and using the fact that $\Psi_i \leq 1$ and $\sqrt{\Lambda_m} \leq \Delta_m\Lambda_m$ shows that

$$\Psi_i^{m_i}\Delta_i \leq \Delta_i \leq \frac{C}{n}\sum_{m\geq i}\sqrt{\Lambda_m} \leq \frac{C}{n}\sum_{m>\ell}\Delta_m\Lambda_m.$$

Case 2 ($\Psi_i < 1/2$) By the definition of Ψ_i and the assumption that $\Psi_i < 1/2$ and $i > \ell$,

$$\Psi_i^{m_i}\Delta_i \leq \frac{36c_1 2^{1-m_i}}{n}\left(\frac{i-1}{\Delta_i} + \sum_{m\geq i}\sqrt{\Lambda_m}\right) \overset{\times}{\leq} \frac{C\ell 2^{1-m_{\ell+1}}}{n\Delta_{\ell+1}} + \frac{C}{n}\sum_{m>\ell}\Delta_m\Lambda_m.$$

The second term is already in the right form. For the first,

$$\frac{\ell 2^{1-m_{\ell+1}}}{n\Delta_{\ell+1}} \overset{\times}{\leq} \frac{C}{n\Delta_{\ell+1}} + \frac{\ell}{n\Delta_{\ell+1}}\mathbb{1}\left\{m_{\ell+1} < \ell/4\right\}$$
$$\overset{\times}{\leq} \frac{C}{n}\sum_{m>\ell}\Delta_m\Lambda_m + C\Delta_{\ell+1}\mathbb{1}\left\{m_{\ell+1} < \ell/4\right\} \overset{\times}{\leq} C\left(\bar{\Delta}_\ell + \frac{1}{n}\sum_{m>\ell}\Delta_m\Lambda_m\right),$$

where the last inequality follows since if $m_{\ell+1} < \ell/4$, then many arms $i \leq \ell$ must have means nearly as small as $\ell + 1$. The result is completed by part *(i)* of the lemma.

Appendix C. Proof of Lemma 9 *(iii)*

The proof relies on another concentration result.

Lemma 15 *There exists an $\varepsilon > 0$ such that for any arm j*

$$\mathbb{P}\left(\textit{exists } s \leq \frac{8n}{\ell} : \hat{\mu}_{j,s} + \sqrt{\frac{2}{s}\,\overline{\log}\left(\sqrt{\frac{n}{2\ell s}}\right)} \leq \mu_j + 2\sqrt{\frac{\varepsilon}{s}}\right) \leq \frac{1}{2}.$$

Proof The result follows by rescaling the time horizon and noting that if B_s is a Brownian motion, then for sufficiently small ε (for example, $1/200$).

$$\mathbb{P}\left(\text{exists } s \leq 8 : B_s \geq \sqrt{2s\,\overline{\log}\left(\frac{1}{\sqrt{2s}}\right)} - 2\sqrt{\varepsilon s}\right) \leq \frac{1}{2}.$$

The above bound does not depend on any variables and can be verified numerically (either by simulating Brownian motion or numerically solving the heat equation that characterises the density of the paths of Brownian motion). An analytical proof is also possible, but requires a modest increase in the definition $\overline{\log}(\cdot)$ if the tangent approximation is to yield a sufficiently tight bound. ■

We need a little more notation. First up is another set of 'usually optimistic' arms, $U \subset [k]$ defined by

$$U = \left\{j : \Delta_j \leq 4\bar{\Delta}_\ell \text{ and } \hat{\mu}_{j,s} + \sqrt{\frac{2}{s}\,\overline{\log}\sqrt{\frac{n}{2\ell s}}} \geq \mu_j + 2\sqrt{\frac{\varepsilon}{s}} \text{ for all } s \leq \frac{8n}{\ell}\right\}.$$

Let $\underline{\Delta} = \sqrt{\varepsilon\ell/(8n)}$ with $\varepsilon > 0$ as given in Lemma 15. Finally we need two more events E_2 and E_3 given by

$$E_2 = \left\{\sqrt{n\ell} + \sum_{m>\ell}\sqrt{\Lambda_m} \leq \sqrt{2n\ell}\right\} \qquad \text{and} \qquad E_3 = \left\{|U| \geq \frac{\ell}{8}\right\}. \tag{26}$$

Lemma 16 *If E_1, E_2, E_3 and $\Delta_i > 4\bar{\Delta}_\ell$, then $T_i(n) \leq \left\lceil \zeta_i(\underline{\Delta}) + \frac{24n}{\varepsilon\ell}\right\rceil$.*

Proof Proceeding by contradiction. Suppose the claim is not true, then there exists a round $t-1 < n$ such that $A_t = i$ and

$$T_i(t-1) = \left\lceil \zeta_i(\underline{\Delta}) + \frac{24n}{\varepsilon\ell}\right\rceil. \tag{27}$$

By the definition of $j \in U$,

$$H_j(t-1) \leq \sum_{m=1}^{k}\sqrt{T_j(t-1)T_m(t-1)} \leq \sqrt{T_j(t-1)}\left(\sqrt{\ell n} + \sum_{m>\ell}\sqrt{\Lambda_m}\right) \leq \sqrt{2T_j(t-1)\ell n}, \tag{28}$$

where the first inequality follows from the definition of $H_j(t-1)$, the second by splitting the sum and because E_1 holds (so that $T_m(n) \leq \Lambda_m$ for $m > \ell$) and Cauchy-Schwarz, the third follows because E_2 holds. Therefore arms $j \in U$ with $T_j(t-1) \leq 8n/\ell$ satisfy

$$\gamma_j(t) = \hat{\mu}_j(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_j(t-1)}\right)}{T_j(t-1)}} \geq \mu_j + 2\sqrt{\frac{\varepsilon}{T_j(t-1)}}. \tag{29}$$

Furthermore, since $T_i(t-1) \geq \zeta_i(\underline{\Delta})$ and $A_t = i$ it holds that $\gamma_i(t) \geq \gamma_j(t)$ and so using Eq. (29) and the same argument as in Lemma 5 leads to

$$\begin{aligned}
\mu_i + \underline{\Delta} + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}{T_i(t-1)}} &\geq \hat{\mu}_i(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}{T_i(t-1)}} \\
&\geq \hat{\mu}_j(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\frac{n}{H_j(t-1)}\right)}{T_j(t-1)}} \geq \hat{\mu}_j(t-1) + \sqrt{\frac{2\,\overline{\log}\left(\sqrt{\frac{n}{2\ell T_j(t-1)}}\right)}{T_j(t-1)}} \\
&\geq \mu_j + 2\sqrt{\frac{\varepsilon}{T_j(t-1)}} \geq \mu_i + \underline{\Delta} + \sqrt{\frac{\varepsilon}{T_j(t-1)}}\,.
\end{aligned}$$

And by rearranging $\frac{2}{T_i(t-1)}\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right) \geq \frac{\varepsilon}{T_j(t-1)}$. Therefore for all $j \in U$ we have

$$T_j(t-1) \geq \min\left\{\frac{8n}{\ell},\, \frac{\varepsilon T_i(t-1)}{\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}\right\}. \tag{30}$$

Since $T_i(t-1) \geq 8n/(\varepsilon\ell)$, the definition of $H_i(t-1)$ implies that

$$\begin{aligned}
H_i(t-1) &= \sum_{j=1}^{k} \min\left\{T_i(t-1),\, \sqrt{T_i(t-1)T_j(t-1)}\right\} \\
&\geq \sum_{j\in U} \min\left\{T_i(t-1),\, T_j(t-1)\right\} \geq \frac{8|U|n}{\ell\,\overline{\log}\left(\frac{n}{H_i(t-1)}\right)} \geq \frac{n}{\overline{\log}\left(\frac{n}{H_i(t-1)}\right)}\,.
\end{aligned}$$

Then since E_3 holds, by Lemma 20(vii) we have $\overline{\log}(n/H_i(t-1)) \leq 3$. Therefore if $T_i(t-1) \geq 24n/(\varepsilon\ell)$, then another application of Eq. (30) shows that $T_j(t-1) \geq 8n/\ell$ and so $n > t-1 = \sum_{j=1}^{k} T_j(t-1) \geq \sum_{j\in U} T_j(t-1) \geq 8n|U|/\ell \geq n$, which is a contradiction. Therefore there does not exist a round $t-1$ where Eq. (27) holds and $A_t = i$ and the lemma follows. ■

Lemma 17 $\mathbb{P}\left(E_3^c\right) \leq 3/\ell$.

Proof By Markov's inequality, $m = \sum_{j\leq\ell} \mathbb{1}\left\{\Delta_j \leq 4\bar{\Delta}_\ell\right\} \geq \frac{3\ell}{4}$. Let $\chi_1, \ldots, \chi_m$ be a sequence of independent Bernoulli events given by $\chi_j = \mathbb{1}\left\{j \notin U\right\}$. Then by Lemma 15, $\mathbb{P}\left(\chi_j = 1\right) \leq 1/2$ and so Chebyshev's inequality implies that

$$\begin{aligned}
\mathbb{P}\left(E_3^c\right) &= \mathbb{P}\left(|U| < \frac{\ell}{8}\right) = \mathbb{P}\left(\sum_{j=1}^{m}(1-\chi_j) < \frac{\ell}{8}\right) = \mathbb{P}\left(\sum_{j=1}^{m}\chi_j > m - \frac{\ell}{8}\right) \\
&\leq \mathbb{P}\left(\sum_{j=1}^{m}(\chi_j - \mathbb{E}[\chi_j]) \geq \frac{m}{2} - \frac{\ell}{8}\right) \leq \frac{m/4}{\left(\frac{m}{2} - \frac{\ell}{8}\right)^2} \leq \frac{3}{\ell}\,.
\end{aligned}$$

■

At last all the tools are available to prove part (iii) of Lemma 9.

Proof [of Lemma 9 *(iii)*] The regret due to arms $i \leq \ell$ is decomposed

$$\mathbb{E}\left[\mathbb{1}_{\{E_1\}}\sum_{i\leq\ell}\Delta_i T_i(n)\right] \leq \mathbb{E}\left[\mathbb{1}_{\{E_2^c\}} n\Delta_\ell\right] + \mathbb{E}\left[\mathbb{1}_{\{E_1,E_2\}}\sum_{i\leq\ell}\Delta_i T_i(n)\right]. \tag{31}$$

The first term is bounded easily using the definition of E_2, Lemma 21(i) and Lemma 9(i).

$$\mathbb{E}\left[\mathbb{1}_{\{E_2^c\}} n\Delta_\ell\right] \leq C\sum_{m>\ell}\mathbb{E}\left[\sqrt{\Lambda_m}\right] \leq C\sum_{m>\ell}\Delta_m\lambda_m\,.$$

The second term in Eq. (31) is bounded using Lemmas 16 and 17. By noting that the contribution to the regret of arms i with $\Delta_i \leq 4\bar{\Delta}_\ell$ is at most $4n\bar{\Delta}_\ell$ it follows that

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}_{\{E_1,E_2\}}\sum_{i\leq\ell}\Delta_i T_i(n)\right] &\leq 4n\bar{\Delta}_\ell + n\Delta_\ell\mathbb{P}\left(E_3^c\right) \\
&\quad + \mathbb{E}\left[\mathbb{1}_{\{E_1,E_2,E_3\}}\sum_{i\leq\ell:\Delta_i>4\bar{\Delta}_\ell}\Delta_i\left(\zeta_i(\underline{\Delta}) + \left\lceil\frac{24n}{\varepsilon\ell}\right\rceil\right)\right].
\end{aligned}$$

The proof is completed since Lemma 17 implies that $n\Delta_\ell\mathbb{P}\left(E_3^c\right) \leq 3n\Delta_\ell/\ell \leq 3n\bar{\Delta}_\ell$ and Lemma 13 implies that

$$\begin{aligned}
\mathbb{E}\left[\mathbb{1}_{\{E_1,E_2,E_3\}}\sum_{i\leq\ell:\Delta_i>4\bar{\Delta}_\ell}\Delta_i\left(\zeta_i(\underline{\Delta}) + \left\lceil\frac{24n}{\varepsilon\ell}\right\rceil\right)\right] &\leq \mathbb{E}\left[\sum_{i\leq\ell}\Delta_i\left(\zeta_i(\underline{\Delta}) + \left\lceil\frac{24n}{\varepsilon\ell}\right\rceil\right)\right] \\
&\leq \sum_{i\leq\ell}\Delta_i\left(2 + \frac{40n}{\varepsilon\ell}\right) \leq Cn\bar{\Delta}_\ell\,.
\end{aligned}$$

■

Appendix D. Technical results

Here some lemmas that are either known or follow from uninteresting calculations. The first two are used for the lower bounds have have been seen before.

Lemma 18 (See Lemma 2.6 in Tsybakov 2008) *Let $\mathbb{P}$ and $\mathbb{P}'$ be measures on the same probability space and assume $\mathbb{P}'$ is absolutely continuous with respect to $\mathbb{P}$. Then for any event A, $\mathbb{P}\left(A\right) + \mathbb{P}'(A^c) \geq \exp(-\mathrm{KL}(\mathbb{P},\mathbb{P}'))/2$, where $\mathrm{KL}(\mathbb{P},\mathbb{P}')$ is the relative entropy between $\mathbb{P}$ and $\mathbb{P}'$.*

The next lemma has also been seen before. For example in the articles by Auer et al. (1995) or Gerchinovitz and Lattimore (2016) where the formalities are described in great detail.

Lemma 19 *Fix a strategy. Let $1 \leq i \leq k$ and $\mu \in \mathbb{R}^k$ and $\mu' \in \mathbb{R}^k$ be such that $\mu_j = \mu'_j$ for all $j \neq i$ and $\mu_i - \mu'_i = \Delta$. Then let $\mathbb{P}$ be the measure on $A_1, X_1, A_2, X_2, \ldots, A_n, X_n$ induced by the interaction of the strategy with rewards sampled using mean vector μ and $\mathbb{P}'$ be the same but with rewards sampled with means from μ'. Then $\mathrm{KL}(\mathbb{P},\mathbb{P}') = \mathbb{E}[T_i(n)]\Delta^2/2$, where the expectation is taken with respect to $\mathbb{P}$.*

Recall that $\overline{\log}(x) = \log((e+x)\log^{\frac{1}{2}}(e+x))$. Here are a few simple facts that make manipulating this unusual function a little easier.

Lemma 20 *The following hold:*

(i) $\overline{\log}$ *is concave and monotone increasing on* $[0, \infty)$.

(ii) $\overline{\log}(0) \geq 1$.

(iii) $\lim_{x\to\infty} \overline{\log}(x)/\log(x) = 1$.

(iv) $(\overline{\log}(x))^{1/2}\exp(-\overline{\log}(x)) \leq 3/(2x)$.

(v) If $x\overline{\log}^{\frac{1}{2}}(b/x) \geq a$, *then* $\overline{\log}(b/x) \leq 2\overline{\log}(b/a)$ *for all* $a, b > 0$.

(vi) $\sqrt{\overline{\log}(x)}\exp(-1-\overline{\log}(x)+2\sqrt{\overline{\log}(x)}) \leq 5\sqrt{1/x}$.

(vii) If $\overline{\log}(x) \geq x$, *then* $\overline{\log}(x) \leq 2$.

Proof (i) is proven by checking derivatives:

$$\frac{d}{dx}\overline{\log}(x) = \frac{\frac{1}{2\sqrt{\log(e+x)}} + \sqrt{\log(e+x)}}{(e+x)\sqrt{\log(e+x)}} \qquad \frac{d^2}{dx^2}\overline{\log}(x) = -\frac{1+\log(e+x)+2\log^2(e+x)}{2(e+x)^2\log^2(e+x)}\,.$$

The former is clearly positive and the latter negative, which shows that $\overline{\log}$ is monotone increasing and concave. Parts (ii) and (iii) are trivial. For (iv),

$$\begin{aligned}
&\overline{\log}^{\frac{1}{2}}(x)\exp(-\overline{\log}(x)) = \overline{\log}^{\frac{1}{2}}(x)\exp(-\log((e+x)\log^{\frac{1}{2}}(e+x))) \\
&= \frac{\overline{\log}^{\frac{1}{2}}(x)}{(e+x)\log^{\frac{1}{2}}(e+x)} = \frac{1}{e+x}\sqrt{\frac{\log(e+x)+\log\log^{\frac{1}{2}}(e+x)}{\log(e+x)}} \leq \frac{1}{e+x}\sqrt{1+\frac{1}{2e}} \leq \frac{3}{2x}\,.
\end{aligned}$$

For (v),

$$\overline{\log}\left(\frac{b}{x}\right) \leq \overline{\log}\left(\frac{b\,\overline{\log}^{\frac{1}{2}}(b/x)}{a}\right) \leq \overline{\log}\left(\frac{b}{a}\,\overline{\log}^{\frac{1}{2}}\left(\frac{b}{a}\,\overline{\log}^{\frac{1}{2}}\left(\frac{b}{a}\,\overline{\log}^{\frac{1}{2}}\left(\frac{b}{a}\cdots\right.\right.\right.\right. = z\,,$$

where the final equality serves as the definition of z. If $z \leq 2$, then $\overline{\log}(b/x) \leq z \leq 2 \leq 2\overline{\log}(b/a)$. Suppose now that $z \geq 2$. Let $u, v \geq 0$, then $\overline{\log}(uv) \leq \overline{\log}(u) + \overline{\log}(v)$ and if $u \geq 2$, then $u^2 - \overline{\log}(u) \geq u^2/2$. Therefore $z^2/2 \leq z^2 - \overline{\log}(z) \leq \overline{\log}(zb/a) - \overline{\log}(z) \leq \overline{\log}(b/a)$. Therefore $\overline{\log}(b/x) \leq z^2 \leq 2\overline{\log}(b/a)$ as required. For (vi), using a similar reasoning as (iv),

$$\begin{aligned}
\sqrt{x\,\overline{\log}(x)}\exp\left(-1-\overline{\log}(x)+2\,\overline{\log}^{\frac{1}{2}}(x)\right) &= \frac{\overline{\log}^{\frac{1}{2}}(x)\sqrt{x}\exp\left(2\,\overline{\log}^{\frac{1}{2}}(x)\right)}{e(e+x)\log^{\frac{1}{2}}(e+x)} \\
&\leq \sqrt{\frac{\overline{\log}(x)}{\log(e+x)}}(x+e)^{-\frac{1}{2}}\exp\left(2\,\overline{\log}^{\frac{1}{2}}(x)\right)\,.
\end{aligned}$$

Simple calculus shows that $\overline{\log}(x)/\log(e+x) \leq (1/2+e)/e$. Let $g(x) = (x+e)^{-\frac{1}{2}}\exp(2\,\overline{\log}^{\frac{1}{2}}(x))$. Then $\max_{x\geq 0} g(x) \leq 10.34 \leq 11$ by numerical calculation, which is valid by the following argument: First, the function g is twice differentiable, satisfies $g(0) > 0$, $g'(0) > 0$ and $\lim_{x\to\infty} g(x) = 0$. By taking the first derivative it is easy to see that g has a unique maximum in $x^* \in (0,\infty)$ with $g(x^*) > 0$. Therefore g is monotone increasing for $x < x^*$ and monotone decreasing afterwards for $x > x^*$. This means the maximiser may be found by a binary search with arbitrary precision. Therefore $\sqrt{x\,\overline{\log}(x)}\exp(-1-\overline{\log}(x)+2\,\overline{\log}^{\frac{1}{2}}(x)) \leq \frac{11}{e}\sqrt{(1/2+e)/e} \leq 5$. For (vii), if $x \geq 0$, then $\frac{d}{dx}\overline{\log}(x) \leq 3/(2e) \leq 1$. Therefore $\overline{\log}(x) - x$ is monotone non-increasing and the result follows by checking that $\overline{\log}(2) \leq 2$. ■

The second technical lemma provides some useful results relating to the optimisation problem appearing in Eq. (3) and the definition of ℓ in Eq. (15).

Lemma 21 *There exists a universal constant* $C > 0$ *such that:*

(i) $\Delta_\ell \leq C\sqrt{\frac{\ell}{n}}$ *or* $n\Delta_\ell \leq C\sum_{m>\ell}\frac{1}{\Delta_m}$.

(ii) $n\bar{\Delta}_\ell + \sum_{m>\ell}\Delta_m\lambda_m \leq C\min_{i\in[k]}\left(n\bar{\Delta}_i + \sum_{m>i}\Delta_m\lambda_m\right)$.

(iii) $\min_{i\in[k]}\left(n\bar{\Delta}_i + \sum_{m>i}\Delta_i\lambda_i\right) \leq C\sqrt{kn} + \sum_{m=1}^{k}\Delta_m$.

(iv) $\min_{i\in[k]}\left(n\bar{\Delta}_i + \sum_{m>i}\Delta_i\lambda_i\right) \leq C\sum_{m:\Delta_m>0}\left(\Delta_m + \frac{\log(n)}{\Delta_m}\right)$.

Proof For part *(i)*, by the definition of ℓ we have

$$\Delta_\ell < 4c_2\sqrt{\frac{k_\ell}{n}} = 4c_2\sqrt{\frac{\ell}{n} + \frac{\Delta_\ell}{n}\sum_{i>\ell}\frac{1}{\Delta_i}} \leq 4c_2\sqrt{\max\left\{\frac{\ell}{n}, \frac{\Delta_\ell}{n}\sum_{i>\ell}\frac{1}{\Delta_i}\right\}}\,.$$

The result follows by simplifying each of the two cases in the maximum. For part *(ii)*, let $i = \arg\min_j n\bar{\Delta}_j + \sum_{m>j}\Delta_m\lambda_m$.

Case 1 ($\ell > i$) Using the fact that for $m \leq \ell$ we have $\Delta_m \leq 16c_2^2 k_m/\Delta_m$ leads to

$$\begin{aligned}
n\bar{\Delta}_\ell + \sum_{m>\ell}\Delta_m\lambda_m &\leq n\bar{\Delta}_i + \frac{n}{\ell}\sum_{m=i+1}^{\ell}\Delta_m + \sum_{m>\ell}\Delta_m\lambda_m \\
&\leq n\bar{\Delta}_i + \frac{16c_2^2 n}{\ell}\sum_{m=i+1}^{\ell}\frac{k_m}{\Delta_m} + \sum_{m>\ell}\Delta_m\lambda_m \leq (1+32c_2^2)\left(n\bar{\Delta}_i + \sum_{m>i}\Delta_m\lambda_m\right)\,.
\end{aligned}$$

Case 2 ($\ell < i$) Using the fact that $\overline{\log}(x)/x \leq 3/2$ for $x \geq 1$ leads to

$$\sum_{m=\ell+1}^{i} \Delta_m \lambda_m = \sum_{m=\ell+1}^{i} \frac{n\Delta_m}{k_m} \cdot \frac{k_m}{n\Delta_m^2} \overline{\log}\left(\frac{n\Delta_m^2}{k_m}\right) + \sum_{m=\ell+1}^{i} \Delta_m$$
$$\leq \frac{3n}{2} \sum_{m=1}^{i} \frac{\Delta_m}{k_m} + \frac{n}{i} \sum_{m=\ell+1}^{i} \Delta_m \leq 7n\bar{\Delta}_i \,,$$

where the first inequality is true since $i/n \leq 1$ and by the definition of ℓ, if $m \geq \ell$, then $n\Delta_m^2/k_m \geq 16c_2^2 \geq 1$. The second inequality follows by letting $k(x) = \sum_{m=1}^{k} \min\{1, x/\Delta_m\}$ and noting that $x/k(x)$ is monotone increasing and by Markov's inequality $k(2\bar{\Delta}_i) \geq i/2$. Then for $\Delta_m \leq 2\bar{\Delta}_i$ it holds that $\Delta_m/k_m \leq 2\bar{\Delta}_i/k(2\bar{\Delta}_i) \leq 4\bar{\Delta}_i/i$ while for $\Delta_m > 2\bar{\Delta}_i$ we have $\Delta_m/k_m \leq 2\Delta_m/i$. Therefore

$$n\bar{\Delta}_\ell + \sum_{m>\ell} \Delta_m \lambda_m \leq n\bar{\Delta}_i + \sum_{m=\ell+1}^{i} \Delta_m \lambda_m + \sum_{m>i} \Delta_m \lambda_m \leq 8\left(n\bar{\Delta}_i + \sum_{m>i} \Delta_m \lambda_m\right)$$

The last two parts are straightforward. For *(iii)*, let $j = \max\{m : \Delta_m \leq 3\sqrt{k/n}\}$, which means that for $m > j$ it holds that $\overline{\log}(n\Delta_m^2/m) \leq 2\log(n\Delta_m^2/m)$. Then

$$\min_{i \in [k]} \left(n\bar{\Delta}_i + \sum_{m>i} \Delta_m \lambda_m\right) - \sum_{m=1}^{k} \Delta_m \leq n\bar{\Delta}_j + \sum_{m>j} \Delta_m(\lambda_m - 1)$$
$$\leq 3\sqrt{kn} + \sum_{m>j} \frac{1}{\Delta_m} \overline{\log}\left(\frac{n\Delta_m^2}{k_m}\right) \leq 3\sqrt{kn} + \sum_{m>j} \frac{2}{\Delta_m} \log\left(\frac{n\Delta_m^2}{m}\right)$$
$$\leq 3\sqrt{kn} + \frac{2}{3}\sqrt{\frac{n}{k}} \int_1^k \log\left(\frac{9k}{x}\right) dx = 3\sqrt{kn} + \frac{2}{3}\sqrt{nk}\,(1 + \log(9)) \leq 6\sqrt{kn}\,.$$

Rearranging completes the proof of *(iii)*. For part *(iv)*, first note that

$$\lambda_m = 1 + \frac{1}{\Delta_m^2} \overline{\log}\left(\frac{n\Delta_m^2}{k_m}\right) \leq 1 + \frac{\overline{\log}(n)}{\Delta_m^2} + \frac{\overline{\log}(\Delta_m^2)}{\Delta_m^2} \leq \frac{5}{2} + \frac{3/2 + \log(e+n)}{\Delta_m^2}\,.$$

The result follows by choosing $i = \max\{i : \Delta_i = 0\}$. ■

Appendix E. History

Table 2 outlines the long history of finite-armed stochastic bandits. It indicates which algorithms are asymptotically optimal and/or sub-UCB and the ratio (up to constant factors) by which they are minimax suboptimal. Empty cells represent results unknown at the time. Most papers do not provide minimax bounds, but they can be derived easily from finite-time bounds using the argument given by Bubeck and Cesa-Bianchi (2012), which I have done where possible. In some cases the finite-time bound cannot be used to derive the minimax bound and these results are marked as conjectures. Algorithms were omitted from the list if (a) I could not straightforwardly adapt their analysis to the

Gaussian noise model and/or frequentist regret (Honda and Takemura, 2011; Russo and Van Roy, 2014; Gutin and Farias, 2016), or (b) the algorithm depends on μ-dependent tuning such as SOFTMIX (Cesa-Bianchi and Fischer, 1998), ε-GREEDY (Auer et al., 2002), EXPLORE-THEN-COMMIT (Garivier et al., 2016a) and BOLTZMANN EXPLORATION (Cesa-Bianchi et al., 2017). Also omitted are algorithms designed for adversarial bandits, few of which are suitable for unbounded rewards and none are competitive with UCB for stochastic problems, but a nice survey of these algorithms is by Bubeck and Cesa-Bianchi (2012). The vast majority of the Bayesian literature is also omitted since it deals with discounted rewards. See the recent book by Gittins et al. (2011) for an overview of Bayesian algorithms.

Remark 22 *It must be emphasised that many of the algorithms in the table were designed for settings more general than Gaussian and the core contribution was actually this generality.*

Date	Algorithm	Sub-UCB	Asy. opt.	Minimax ratio	Anytime
1960	'explore-then-commit' Vogel (1960)			1*	no
1985	'forcing' Lai and Robbins (1985)		yes		yes
1987	KL-UCB* Lai (1987)		yes		no
1995	UCB Katehakis and Robbins (1995), Agrawal (1995)		yes		yes
2002	UCB† Auer et al. (2002)	yes	no	$\sqrt{\log(n)}$	yes
2002	UCB2† Auer et al. (2002)	yes	yes	$\sqrt{\log(n)}$	yes
2007	UCB-V Audibert et al. (2007)	yes	no	$\sqrt{\log(n)}$	yes
2009	MOSS† Audibert and Bubeck (2009)	no	no	1	no
2010	IMPROVED UCB† Auer and Ortner (2010)	yes	no	$\sqrt{\log(k)}$	yes
2010	DMED† Honda and Takemura (2010)		yes	$\sqrt{\log(n)}$ ♭	yes
2010	DMED+† Honda and Takemura (2010)		yes	$\sqrt{\log(k)}$ ♭	yes
2011	KL-UCB Cappé et al. (2013)	yes	yes	$\sqrt{\log(n)}$	yes
2011	KL-UCB+‡ Cappé et al. (2013)	yes	yes	$\sqrt{\log(k)}$	yes
2012	BAYES-UCB† Kaufmann et al. (2012)	yes	yes	$\sqrt{\log(n)}$	yes
2012	THOMPSON SAMPLING♯ Agrawal and Goyal (2012)	yes	yes	$\sqrt{\log(k)}$	yes
2015	IMED† Honda and Takemura (2015)	yes	yes	$\sqrt{\log(k)}$ ♭	yes
2016	BAYES-UCB+ Kaufmann (2016)	yes	yes	$\sqrt{\log(k)}$ ♭	yes
2016	FH-GITTINS Lattimore (2016a)	yes		$\sqrt{\log(n)}$	no
2016	MOSS-ANYTIME Degenne and Perchet (2016)	no	no	1	yes
2017	KL-UCB++ Ménard and Garivier (2017)	no	yes	1	no
2018	KL-UCB*	yes	yes	$\sqrt{\log(k)}$	no
2018	ADA-UCB	yes	yes	1	no

*Results given for two-armed Bernoulli bandits only.
†Results given for bounded and/or Bernoulli rewards, but algorithm/proof is easily adapted.
‡No known reference. Can be shown using tools of this paper combined with those by Kaufmann (2016); Lattimore (2016b).
♯Results are given for bounded rewards, but the same technique works for Gaussian rewards. See also the article by Korda et al. (2013).
♭A conjectured result.

Table 2: History of bandit algorithms

$\overline{\log}(\cdot)$	$\overline{\log}(x) = \log((e+x)\log^{\frac{1}{2}}(e+x))$
k	number of arms
n	time horizon
A_t	action chosen in round t
μ	k-dimensional vector of mean payoffs
$\Delta_i(\mu)$	suboptimality gap, $\Delta_i = \max_j \mu_j - \mu_i$
k_i	$\sum_{j=1}^{k} \min\left\{1, \frac{\Delta_i}{\Delta_j}\right\}$
λ_i	$1 + \frac{1}{\Delta_i}\overline{\log}\left(\frac{n\Delta_i^2}{k_i}\right)$
η_t	noise in round t
X_t	reward in the tth round, $X_t = \mu_{A_t} + \eta_t$
$T_i(t)$	number of plays of arm i after t rounds
$K_i(t)$	$\sum_{m=1}^{k} \min\left\{1, \sqrt{\frac{T_m(t-1)}{T_i(t-1)}}\right\}$
$H_i(t)$	$T_i(t)K_i(t)$
Λ_i	see display before Lemma 7
$\hat{\mu}_i(t)$	empirical mean of arm i after t rounds
$\hat{\mu}_{i,s}$	empirical mean of arm i after s plays
$\bar{\Delta}_i$	$\sum_{m=1}^{i} \Delta_m / i$
V_i, W_i	sets of arms defined in Eq. (14)
δ_i	see Eq. (15)
ℓ	see Eq. (15)
F_i	Event that enough arms are optimistic, see Eq. (17)
$\zeta_i(\Delta)$	$1 + \max_s \{s : \hat{\mu}_{i,s} > \mu_i + \Delta\}$
c_1, c_2	constants $c_1 = 4$ and $c_2 = 12$

Table 3: Table of notation

Journal of Machine Learning Research 19 (2018) 1-5 Submitted 12/17; Revised 6/18; Published 7/18

ThunderSVM: A Fast SVM Library on GPUs and CPUs

Zeyi Wen[†] WENZY@COMP.NUS.EDU.SG
Jiashuai Shi[†‡] SHIJIASHUAI@GMAIL.COM
Qinbin Li[†] LIQINBIN1998@GMAIL.COM
Bingsheng He[†] HEBS@COMP.NUS.EDU.SG
Jian Chen[‡] ELLACHEN@SCUT.EDU.CN
[†]*School of Computing, National University of Singapore, 117418, Singapore*
[‡]*School of Software Engineering, South China University of Technology, Guangzhou, 510006, China*

Editor: Alexandre Gramfort

Abstract

Support Vector Machines (SVMs) are classic supervised learning models for classification, regression and distribution estimation. A survey conducted by Kaggle in 2017 shows that 26% of the data mining and machine learning practitioners are users of SVMs. However, SVM training and prediction are very expensive computationally for large and complex problems. This paper presents an efficient and open source SVM software toolkit called ThunderSVM which exploits the high-performance of Graphics Processing Units (GPUs) and multi-core CPUs. ThunderSVM supports all the functionalities—including classification (SVC), regression (SVR) and one-class SVMs—of LibSVM and uses identical command line options, such that existing LibSVM users can easily apply our toolkit. ThunderSVM can be used through multiple language interfaces including C/C++, Python, R and MATLAB. Our experimental results show that ThunderSVM is generally an order of magnitude faster than LibSVM while producing identical SVMs. In addition to the high efficiency, we design our convex optimization solver in a general way such that SVC, SVR, and one-class SVMs share the same solver for the ease of maintenance. Documentation, examples, and more about ThunderSVM are available at `https://github.com/zeyiwen/thundersvm`.

Keywords: SVMs, GPUs, multi-core CPUs, efficiency, multiple interfaces

1. Introduction

Support Vector Machines (SVMs) have been widely used in many applications including document classification (D'Orazio et al., 2014), image classification (Pasolli et al., 2014), blood pressure estimation (Kachuee et al., 2015), disease detection (Bodnar and Salathé, 2013), and outlier detection (Roth, 2006). A survey conducted by Kaggle in 2017 shows that 26% of the data science practitioners use SVMs to solve their problems (Thomas, 2017). The open-source project LibSVM which supports classification (SVC), regression (SVR) and one-class SVMs has been widely used in many applications. LibSVM was developed in 2000 (Chang and Lin, 2011), and has been maintained since then. Despite the advantages of SVMs, SVM training and prediction are very expensive for large and complex problems.

Graphics Processing Units (GPUs) have been used to accelerate the solutions of many real-world applications (Dittamo and Cisternino, 2008), due to the abundant computing cores and high memory bandwidth of GPUs. In this paper, we introduce a toolkit named *ThunderSVM* which exploits GPUs and multi-core CPUs. The mission of our toolkit is to help users easily and efficiently apply SVMs to solve problems. It is worthy to point out that one way to train SVMs faster is to use kernel approximations. ThunderSVM aims to find an exact solution. ThunderSVM supports all the functionalities of LibSVM including SVC, SVR and one-class SVMs. We use the same command line input options as LibSVM, such that existing LibSVM users are able to easily switch to ThunderSVM. Moreover, ThunderSVM supports multiple interfaces such as C/C++, Python, R and MATLAB. ThunderSVM can run on Linux, Windows or Macintosh operating systems with or without GPUs. Empirical results show ThunderSVM is generally 10 times faster than LibSVM in all the functionalities. The full version of ThunderSVM, which is released under Apache License 2.0, can be found on GitHub at `https://github.com/zeyiwen/thundersvm`. The GitHub repository of ThunderSVM has attracted 700 stars and 85 forks as of July 24, 2018.

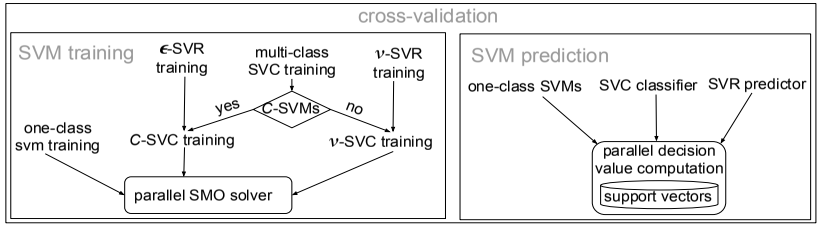


Figure 1: Overview of training and prediction

2. Overview and Design of ThunderSVM

Like LibSVM, ThunderSVM supports one-class SVMs, C-SVMs and ν-SVMs where C represents the regularization constant and ν represents the parameter controlling the training error. Both C-SVMs and ν-SVMs are used for classification and regression.

Figure 1 shows the overview of ThunderSVM which has many functionalities: one-class SVMs for distribution estimation, C-SVC and ν-SVC for SVM classification, and ϵ-SVR and ν-SVR for SVM regression. The training algorithms for those SVMs are built on top of a generic parallel SMO solver which is for solving quadratic optimization problems. Notably, the SVM training for regression (such as ϵ-SVR and ν-SVR) and the multi-class SVM training can be converted into the training of an SVM classifier. The prediction module is relatively simple, because the prediction is the same for one-class SVMs, C-SVMs and ν-SVMs. The prediction is essentially computing predicted values based on support vectors. ThunderSVM also contains the cross-validation functionality.

2.1 Design of Parallel SVM Training Algorithms

We have developed a series of optimizations for the training. First, ThunderSVM computes a number of rows of the kernel matrix in a batch, reuses the rows that are stored in the GPU

memory buffer, and solves multiple subproblems in that batch. Thus, ThunderSVM avoids performing a large number of small read/write operations to the high latency memory and reduces repeated kernel value computation. Moreover, we apply GPU shared memory to accelerate parallel reduction, and use the massive parallelism to update elements of arrays.

For solving each subproblem in the training, we use the SMO algorithm which consists of three key steps. Step (i): Find two extreme training instances which can potentially improve the currently trained SVM the most. Step (ii): Improve the two Lagrange multipliers of the two instances. Step (iii): Update the optimality indicators of all the training instances. We parallelize Step (i) and (iii). Step (ii) is computationally inexpensive, and we simply execute it sequentially. In Step (i), our key idea is to apply the parallel reduction (Merrill, 2015) twice for finding the two extreme training instances. In the parallel reduction, we first load the whole array from the GPU global memory to shared memory in a coalescent way, and then reduce the array size by two at each iteration until only one element left. In Step (iii), we dedicate one thread to update one optimality indicator to use the massive parallelism mechanism of the GPU. The training is terminated when the optimality condition is met or the SVMs cannot be further improved. More details about the training and the termination condition can be found in our supplementary material (Wen et al., 2017).

For solving the batch of subproblems, we propose techniques to exploit the properties of the batch of subproblems. First, to reduce access to the high latency memory, the kernel values needed for the batch are organized together and computed through matrix multiplication. As a result, we reduce a large number of small read/write operations to the high latency memory during kernel value computation. We use matrix operations from the high-performance library cuSparse (Nvidia, 2008) provided by NVIDIA. Second, to reduce repeated kernel value computation, we store the kernel values in a GPU buffer for efficient reuse during the optimization for the batch. Regarding the selection of the batch, we make use of the set of instances with the deepest gradients.

Training SVMs for regression (SVR) or for multi-class classification can be reduced to training SVMs for binary classification (SVC), as discussed in the previous study (Shevade et al., 2000). Two SVC training algorithms (C-SVC and ν-SVC) and training algorithm for one-class SVMs are essentially solving optimization problems using SMO. More details about the relationship of SVR and SVC, and SMO can be found in our supplementary material (Wen et al., 2017). The key task in the SVM training is to parallelize SMO, and the insight has been discussed above. The parallelism principles are applicable to CPUs.

2.2 Design of the Parallel Prediction Algorithm

Although ThunderSVM supports several algorithms (such as one-class SVMs, classification and regression), their underlying prediction algorithm is identical: a function based on the support vectors and their Lagrange multipliers. The function is $v = \sum_{i=1}^{n} y_i \alpha_i K(\boldsymbol{x}_i, \boldsymbol{x}_j) + b$ where $\boldsymbol{x}_j$ is the instance of interest for prediction; y_i and α_i are the label and Lagrange multiplier of the support vector $\boldsymbol{x}_i$, respectively; b is the bias of the SVM hyperplane; $K(\cdot, \cdot)$ is the kernel function. In ThunderSVM, we perform the prediction by evaluating the equation in parallel. First, we conduct a vector to matrix multiplication in parallel to obtain all the needed kernel values, where the vector is $\boldsymbol{x}_j$ and the matrix consists of all the support vectors. Then, the sum of the equation can be performed using a parallel reduction.

data set			elapsed time (sec)			speedup	
name	cardinality	dimension	ThunderSVM		LibSVM		
			gpu	cpu		gpu	cpu
mnist8m (svc)	8.1x10^6	784	7.1x10^3	3.2x10^4	8.1x10^5	114	39.9
rcv1_test (svc)	677399	47236	621	20633	8.3x10^5	1337	40
epsilon (svc)	400000	2000	1251	26042	1 week+	483+	23+
e2006-tfidf (svr)	16087	150360	13.25	343.5	9161	691	25.3
webdata (ocsvm)	49749	300	4.66	16.5	1493	320	90.5

Table 1: Comparison between ThunderSVM with LibSVM

3. Experimental Studies

We compare the efficiency on training SVMs for classification, regression and one-class SVMs (denoted by "OCSVM"). Five representative data sets are listed in Table 1. We conducted our experiments on a workstation running Linux with two Xeon E5-2640 v4 10 core CPUs, 256GB main memory and a Tesla P100 GPU of 12GB memory. ThunderSVM are implemented in CUDA-C and C++ with OpenMP. We used the Gaussian kernel. Five pairs of hyper-parameters (C, γ) for the data sets are (10, 0.125), (100, 0.125), (0.01, 1), (256, 0.125), and (64,7.8125) and are the same as the existing studies (Wen et al., 2014, 2018). More experimental evaluation can be found in our supplementary material. ThunderSVM when using GPUs is over 100 times faster than LibSVM. When running on CPUs, it is over 10 times faster than LibSVM. For prediction, ThunderSVM is also 10 to 100 times faster than LibSVM (Wen et al., 2017). We varied the hyper-parameters C from 0.01 to 100 and γ from 0.03 to 10, and ThunderSVM is 10 to 100 times faster than LibSVM.

4. Conclusion

In this paper, we present our software tool called "ThunderSVM" which supports all the functionalities of LibSVM. For ease of usage, ThunderSVM uses identical input command line options as LibSVM, and supports Python, R and Matlab. Empirical results show that ThunderSVM is generally 100 times faster than LibSVM in all the functionalities when GPUs are used. When running purely on CPUs, ThunderSVM is often 10 times faster than LibSVM. We hope this significant efficiency improvement would help practitioners in the community quickly solve their problems and enable SVMs to solve more complex problems.

Acknowledgments

This work is supported by a MoE AcRF Tier 1 grant (T1 251RES1610) and Tier 2 grant (MOE2017-T2-1-122) in Singapore. Prof. Chen is supported by the Guangdong special branch plans young talent with scientific and technological innovation (No. 2016TQ03X445), Guangzhou science and technology planning project (No. 2019-03-01-06-3002-0003) and Guangzhou Tianhe District science and technology planning project (No. 201702YH112). Bingsheng He and Jian Chen are corresponding authors. We acknowledge NVIDIA for the hardware donations and thank the anonymous reviewers for their insightful comments.

References

Todd Bodnar and Marcel Salathé. Validating models for disease detection using Twitter. In *International Conference on World Wide Web (WWW)*, pages 699–702. ACM, 2013.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: a library for Support Vector Machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

Cristian Dittamo and Antonio Cisternino. GPU White paper, 2008.

Vito D'Orazio, Steven T Landis, Glenn Palmer, and Philip Schrodt. Separating the wheat from the chaff: applications of automated document classification using Support Vector Machines. *Political Analysis*, 22(2):224–242, 2014.

Mohamad Kachuee, Mohammad Mahdi Kiani, Hoda Mohammadzade, and Mahdi Shabany. Cuff-less high-accuracy calibration-free blood pressure estimation using pulse transit time. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1006–1009. IEEE, 2015.

Duane Merrill. CUB v1. 5.3: CUDA unbound, a library of warp-wide, block-wide, and device-wide GPU parallel primitives, 2015.

CUDA Nvidia. Cublas library. *NVIDIA Corporation, Santa Clara, California*, 15(27):31, 2008.

Edoardo Pasolli, Farid Melgani, Devis Tuia, Fabio Pacifici, and William J Emery. SVM active learning approach for image classification using spatial information. *IEEE Transactions on Geoscience and Remote Sensing*, 52(4):2217–2233, 2014.

Volker Roth. Kernel fisher discriminants for outlier detection. *Neural Computation*, 18(4): 942–960, 2006.

Shirish Krishnaj Shevade, S Sathiya Keerthi, Chiranjib Bhattacharyya, and Karaturi Radha Krishna Murthy. Improvements to the SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks*, 11(5):1188–1193, 2000.

Amber Thomas. Kaggle 2017 survey results: `https://www.kaggle.com/amberthomas/kaggle-2017-survey-results`, 2017.

Zeyi Wen, Rui Zhang, Kotagiri Ramamohanarao, Jianzhong Qi, and Kerry Taylor. MASCOT: fast and highly scalable SVM cross-validation using GPUs and SSDs. In *IEEE International Conference on Data Mining (ICDM)*, pages 580–589. IEEE, 2014.

Zeyi Wen, Jiashuai Shi, Qinbin Li, Bingsheng He, and Jian Chen. Supplementary material of ThunderSVM: `https://github.com/zeyiwen/thundersvm/blob/master/thundersvm-full.pdf`, 2017.

Zeyi Wen, Rui Zhang, Kotagiri Ramamohanarao, and Li Yang. Scalable and fast SVM regression using modern hardware. *World Wide Web*, 21(2):261–287, 2018.

Journal of Machine Learning Research 19 (2018) 1-51 Submitted 12/17; Revised 5/18; Published 8/18

Robust Synthetic Control

Muhammad Amjad MAMJAD@MIT.EDU
Operations Research Center
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Devavrat Shah DEVAVRAT@MIT.EDU
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Dennis Shen DESHEN@MIT.EDU
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
Cambridge, MA 02139, USA

Editor: Peter Spirtes

Abstract

We present a robust generalization of the synthetic control method for comparative case studies. Like the classical method cf. Abadie and Gardeazabal (2003), we present an algorithm to estimate the unobservable counterfactual of a treatment unit. A distinguishing feature of our algorithm is that of de-noising the data matrix via singular value thresholding, which renders our approach robust in multiple facets: it automatically identifies a good subset of donors for the synthetic control, overcomes the challenges of missing data, and continues to work well in settings where covariate information may not be provided. We posit that the setting can be viewed as an instance of the Latent Variable Model and provide the first finite sample analysis (coupled with asymptotic results) for the estimation of the counterfactual. Our algorithm accurately imputes missing entries and filters corrupted observations in producing a consistent estimator of the underlying signal matrix, provided $p = \Omega(T^{-1+\zeta})$ for some $\zeta > 0$; here, p is the fraction of observed data and T is the time interval of interest. Under the same proportion of observations, we demonstrate that the mean-squared error in our counterfactual estimation scales as $\mathcal{O}(\sigma^2/p + 1/\sqrt{T})$, where σ^2 is the variance of the inherent noise. Additionally, we introduce a Bayesian framework to quantify the estimation uncertainty. Our experiments, using both synthetic and real-world datasets, demonstrate that our robust generalization yields an improvement over the classical synthetic control method.

Keywords: Observation Studies, Causal Inference, Matrix Estimation

1. Introduction

On November 8, 2016 in the aftermath of several high profile mass-shootings, voters in California passed Proposition 63 in to law BallotPedia (2016). Prop. 63 "outlaw[ed] the possession of ammunition magazines that [held] more than 10 rounds, requir[ed] background checks for people buying bullets," and was proclaimed as an initiative for "historic progress to reduce gun violence" McGreevy (2016). Imagine that we wanted to study the impact of Prop. 63 on the rates of violent crime in California.

Randomized control trials, such as A/B testings, have been successful in establishing effects of interventions by randomly exposing segments of the population to various types of interventions. Unfortunately, a randomized control trial is not applicable in this scenario since only one California exists. Instead, a statistical comparative study could be conducted where the rates of violent crime in California are compared to a "control" state after November 2016, which we refer to as the post-intervention period. To reach a statistically valid conclusion, however, the control state must be demonstrably similar to California sans the passage of a Prop. 63 style legislation. In general, there may not exist a natural control state for California, and subject-matter experts tend to disagree on the most appropriate state for comparison.

As a suggested remedy to overcome the limitations of a classical comparative study outlined above, Abadie et al. proposed a powerful, data-driven approach to construct a "synthetic" control unit absent of intervention Abadie et al. (2010); Abadie and Gardeazabal (2003); Abadie et al. (2011). In the example above, the synthetic control method would construct a "synthetic" state of California such that the rates of violent crime of that hypothetical state would best match the rates in California before the passage of Prop. 63. This synthetic California can then serve as a data-driven counterfactual for the period after the passage of Prop. 63. Abadie et al. propose to construct such a synthetic California by choosing a convex combination of other states (donors) in the United States. For instance, synthetic California might be 80% like New York and 20% like Massachusetts. This approach is nearly entirely data-driven and appeals to intuition. For optimal results, however, the method still relies on subjective covariate information, such as employment rates, and the presence of domain "experts" to help identify a useful subset of donors. The approach may also perform poorly in the presence of non-negligible levels of noise and missing data.

1.1 Overview of main contributions.

As the main result, we propose a simple, two-step robust synthetic control algorithm, wherein the first step de-noises the data and the second step learns a linear relationship between the treated unit and the donor pool under the de-noised setting. The algorithm is robust in two senses: first, it is fully data-driven in that it is able to find a good donor subset even in the absence of helpful domain knowledge or supplementary covariate information; and second, it provides the means to overcome the challenges presented by missing and/or noisy observations. As another important contribution, we establish analytic guarantees (finite sample analysis and asymptotic consistency) – that are missing from the literature – for a broader class of models.

Robust algorithm. A distinguishing feature of our work is that of de-noising the observation data via singular value thresholding. Although this spectral procedure is commonplace in the matrix completion arena, it is novel in the realm of synthetic control. Despite its simplicity, however, thresholding brings a myriad of benefits and resolves points of concern that have not been previously addressed. For instance, while classical methods have not even tackled the obstacle of missing data, our approach is well equipped to impute missing values as a consequence of the thresholding procedure. Additionally, thresholding can help prevent the model from overfitting to the idiosyncrasies of the data, providing a knob for practitioners to tune the "bias-variance" trade-off of their model and, thus, reduce their mean-squared error (MSE). From empirical studies, we hypothesize that thresholding may possibly render auxiliary covariate information (vital to several existing methods) a luxury as opposed to a necessity. However, as one would expect, the algorithm can only benefit from useful covariate and/or "expert" information and we do not advocate ignoring such helpful information, if available.

In the spirit of combatting overfitting, we extend our algorithm to include regularization techniques such as ridge regression and LASSO. We also move beyond point estimates in establishing a Bayesian framework, which allows one to quantitatively compute the uncertainty of the results

through posterior probabilities.

Theoretical performance. To the best of our knowledge, ours is the first to provide finite sample analysis of the MSE for the synthetic control method, in addition to guarantees in the presence of missing data. Previously, the main theoretical result from the synthetic control literature (cf. Abadie et al. (2010); Abadie and Gardeazabal (2003); Abadie et al. (2011)) pertained to bounding the bias of the synthetic control estimator; however, the proof of the result assumed that the latent parameters, which live in the simplex, have a perfect pre-treatment match in the noisy predictor variables – our analysis, on the other hand, removes this assumption. We begin by demonstrating that our de-noising procedure produces a consistent estimator of the latent signal matrix (Theorems 1, 2), proving that our thresholding method accurately imputes and filters missing and noisy observations, respectively. We then provide finite sample analysis that not only highlights the value of thresholding in balancing the inherent "bias-variance" trade-off of forecasting, but also proves that the prediction efficacy of our algorithm degrades gracefully with an increasing number of randomly missing data (Theorems 3, 7, and Corollary 4). Further, we show that a computationally beneficial pre-processing data aggregation step allows us to establish the asymptotic consistency of our estimator in generality (Theorem 5).

Additionally, we prove a simple linear algebraic fact that justifies the basic premise of synthetic control, which has not been formally established in literature, i.e. the linear relationship between the treatment and donor units that exists in the pre-intervention continues to hold in post-intervention period (Theorem 6). We introduce a latent variable model, which subsumes many of the models previously used in literature (e.g. econometric factor models). Despite this generality, a unifying theme that connects these models is that they all induce (approximately) low rank matrices, which is well suited for our method.

Experimental results. We conduct two sets of experiments: (a) on existing case studies from real world datasets referenced in Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003), and (b) on synthetically generated data. Remarkably, while Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003) use numerous covariates and employ expert knowledge in selecting their donor pool, our algorithm achieves similar results without any such assistance; additionally, our algorithm detects subtle effects of the intervention that were overlooked by the original synthetic control approach. Since it is impossible to simultaneously observe the evolution of a treated unit and its counterfactual, we employ synthetic data to validate the efficacy of our method. Using the MSE as our evaluation metric, we demonstrate that our algorithm is robust to varying levels of noise and missing data, reinforcing the importance of de-noising.

1.2 Related work.

Synthetic control has received widespread attention since its conception by Abadie and Gardeazabal in their pioneering work Abadie and Gardeazabal (2003); Abadie et al. (2010). It has been employed in numerous case studies, ranging from criminology Saunders et al. (2014) to health policy Kreif et al. (2015) to online advertisement to retail; other notable studies include Abadie et al. (2014); Billmeier and Nannicini (2013); Adhikari and Alm (2016); Aytug et al. (2016). In their paper on the state of applied econometrics for causality and policy evaluation, Athey and Imbens assert that synthetic control is "one of the most important development[s] in program evaluation in the past decade" and "arguably the most important innovation in the evaluation literature in the last fifteen years" Athey and Imbens (2016). In a somewhat different direction, Hsiao et al. introduce the panel data method Hsiao (2014); Hsiao et al. (2011), which seems to have a close bearing with some of the approaches of this work. In particular, to learn the weights of the synthetic control, Hsiao (2014); Hsiao et al. (2011) solve an ordinary least squares problem of Y on $\tilde{X}$, where Y is the data for the outcome variable of the treatment unit and $\tilde{X}$ includes other variables, e.g. covariates, and the outcome variable data from the donor units. However, Hsiao (2014); Hsiao et al. (2011) restrict the

subset of possible controls to units that are within the geographical or economic proximity of the treated unit. Therefore, there is still some degree of subjectivity in the choice of the donor pool. In addition, Hsiao (2014); Hsiao et al. (2011) do not include a "de-noising" step, which is a key feature of our approach. For an empirical comparison between the synthetic control and panel data methods, see Gardeazabal and Vega-Bayo (2016). It should be noted that Gardeazabal and Vega-Bayo (2016) also adapts the panel data method to automate the donor selection process. In this comparison study, the authors conclude that neither the synthetic control method nor the panel data method is vastly superior to the other. They suggest that the synthetic control method may be more useful when there are more time periods and covariates. However, when there is a poor pre-treatment match, the synthetical control method is not feasible while the panel data method can still be used, even though it may suffer from some extrapolation bias. But when a good pre-intervention match is found, the authors conclude that the synthetic control method tends to produce lower MSE, MAPE and mean-error. However, in another comparison study, Wan et al. (2018) compare and contrast the assumptions of both methods and note that the panel data method appears to outperform the synthetic control method in a majority of the simulations they conducted.

Among other notable bodies of work, Doudchenko and Imbens (2016) allows for an additive difference between the treated unit and donor pool, similar to the difference-in-differences (DID) method. Moreover, similar to our exposition, Doudchenko and Imbens (2016) relaxes the convexity aspect of synthetic control and proposes an algorithm that allows for unrestricted linearity as well as regularization. In an effort to infer the causal impact of market interventions, Brodersen et al. (2015) introduce yet another evaluation methodology based on a diffusion-regression state-space model that is fully Bayesian; similar to Abadie et al. (2010); Abadie and Gardeazabal (2003); Hsiao (2014); Hsiao et al. (2011), their model also generalizes the DID procedure. Due to the subjectivity in the choice of covariates and predictor variables, Ferman et al. (2016) provides recommendations for specification-searching opportunities in synthetic control applications. The recent work of Xu (2017) extends the synthetic control method to allow for multiple treated units and variable treatment periods as well as the treatment being correlated with unobserved units. Similar to our work, Xu (2017) computes uncertainty estimates; however, while Xu (2017) obtains these measurements via a parametric bootstrap procedure, we obtain uncertainty estimates under a Bayesian framework.

Matrix completion and factorization approaches are well-studied problems with broad applications (e.g. recommendation systems, graphon estimation, etc.). As shown profusely in the literature, spectral methods, such as singular value decomposition and thresholding, provide a procedure to estimate the entries of a matrix from partial and/or noisy observations Candès and Recht (2008). With our eyes set on achieving "robustness", spectral methods become particularly appealing since they de-noise random effects and impute missing information within the data matrix Jha et al. (2010). For a detailed discussion on the topic, see Chatterjee (2015); for algorithmic implementations, see Mazumder et al. (2010) and references there in. We note that our goal differs from traditional matrix completion applications in that we are using spectral methods to estimate a low-rank matrix, allowing us to determine a linear relationship between the rows of the mean matrix. This relationship is then projected into the future to determine the counterfactual evolution of a row in the matrix (treated unit), which is traditionally not the goal in matrix completion applications. Another line of work within this arena is to impute the missing entries via a nearest neighbor based estimation algorithm under a latent variable model framework Lee et al. (2016); Borgs et al. (2017).

There has been some recent work in using matrix norm methods in relation to causal inference, including for synthetic control. In Athey et al. (2017), the authors use matrix norm regularization techniques to estimate counterfactuals for panel data under settings that rely on the availability of a large number of units relative to the number of factors or characteristics, and under settings that involve limited number of units but plenty of history (synthetic control). This is different from our approach, which increases robustness by "de-noising" using matrix completion methods, and then using linear regression on the de-noised matrix, instead of relying on matrix norm regularizations.

Despite its popularity, there has been less theoretical work in establishing the consistency of the synthetic control method or its variants. Abadie et al. (2010) demonstrates that the bias of the synthetic control estimator can be bounded by a function that is close to zero when the pre-intervention period is large in relation to the scale of the transitory shocks, but under the additional condition that a perfect convex match between the pre-treatment noisy outcome and covariate variables for the treated unit and donor pool exists. Ferman and Pinto (2016) relaxes the assumption in Abadie et al. (2010), and derives conditions under which the synthetic control estimator is asymptotically unbiased under non-stationarity conditions. To our knowledge, however, no prior work has provided finite-sample analysis, analyzed the performance of these estimators with respect to the mean-squared error (MSE), established asymptotic consistency, or addressed the possibility of missing data, a common handicap in practice.

2. Background

2.1 Notation.

We will denote $\mathbb{R}$ as the field of real numbers. For any positive integer N, let $[N] = \{1, \dots, N\}$. For any vector $v \in \mathbb{R}^n$, we denote its Euclidean (ℓ_2) norm by $\|v\|_2$, and define $\|v\|_2^2 = \sum_{i=1}^n v_i^2$. We define its infinity norm as $\|v\|_\infty = \max_i |v_i|$. In general, the ℓ_p norm for a vector v is defined as $\|v\|_p = \left(\sum_{i=1}^n |v_i|^p\right)^{1/p}$. Similarly, for an $m \times n$ real-valued matrix $\boldsymbol{A} = [A_{ij}]$, its spectral/operator norm, denoted by $\|\boldsymbol{A}\|_2$, is defined as $\|\boldsymbol{A}\|_2 = \max_{1 \le i \le k} |\sigma_i|$, where $k = \min\{m, n\}$ and σ_i are the singular values of $\boldsymbol{A}$. The Moore-Penrose pseudoinverse $\boldsymbol{A}^\dagger$ of $\boldsymbol{A}$ is defined as

$$\boldsymbol{A}^\dagger = \sum_{i=1}^k (1/\sigma_i) y_i x_i^T, \quad \text{where} \quad \boldsymbol{A} = \sum_{i=1}^k \sigma_i x_i y_i^T, \tag{1}$$

with x_i and y_i being the left and right singular vectors of $\boldsymbol{A}$, respectively. We will adopt the shorthand notation of $\|\cdot\| \equiv \|\cdot\|_2$. To avoid any confusions between scalars/vectors and matrices, we will represent all matrices in bold, e.g. $\boldsymbol{A}$.

Let f and g be two functions defined on the same space. We say that $f(x) = \mathcal{O}(g(x))$ and $f(x) = \Omega(g(x))$ if and only if there exists a positive real number M and a real number x_0 such that for all $x \ge x_0$,

$$|f(x)| \le M|g(x)| \quad \text{and} \quad |f(x)| \ge M|g(x)|, \tag{2}$$

respectively.

2.2 Model.

The data at hand is a collection of time series with respect to an aggregated metric of interest (e.g. violent crime rates) comprised of both the treated unit and the donor pool outcomes. Suppose we observe $N \ge 2$ units across $T \ge 2$ time periods. We denote T_0 as the number of pre-intervention periods with $1 \le T_0 < T$, rendering $T - T_0$ as the length of the post-intervention stage. Without loss of generality, let the first unit represent the treatment unit – exposed to the intervention of interest at time $t = T_0 + 1$. The remaining donor units, $2 \le i \le N$, are unaffected by the intervention for the entire time period $[T] = \{1, \dots, T\}$.

Let X_{it} denote the measured value of metric for unit i at time t. We posit

$$X_{it} = M_{it} + \epsilon_{it}, \tag{3}$$

where M_{it} is the deterministic mean while the random variables ϵ_{it} represent zero-mean noise that are independent across i, t. Following the philosophy of latent variable models Chatterjee (2015); Lee et al. (2016); Aldous (1981); Hoover (1979, 1981), we further posit that for all $2 \le i \le N$, $t \in [T]$

$$M_{it} = f(\theta_i, \rho_t), \tag{4}$$

where $\theta_i \in \mathbb{R}^{d_1}$ and $\rho_t \in \mathbb{R}^{d_2}$ are latent feature vectors capturing unit and time specific information, respectively, for some $d_1, d_2 \ge 1$; the latent function $f : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$ captures the model relationship. We note that this formulation subsumes popular econometric factor models, such as the one presented in Abadie et al. (2010), as a special case with (small) constants $d_1 = d_2$ and f as a bilinear function.

The treatment unit obeys the same model relationship during the pre-intervention period. That is, for $t \le T_0$

$$X_{1t} = M_{1t} + \epsilon_{1t}, \tag{5}$$

where $M_{1t} = f(\theta_1, \rho_t)$ for some latent parameter $\theta_1 \in \mathbb{R}^{d_1}$. If unit one was never exposed to the intervention, then the same relationship as (5) would continue to hold during the post-intervention period as well. In essence, we are assuming that the outcome random variables for *all* unaffected units follow the model relationship defined by (5) and (3). Therefore, the "synthetic control" would ideally help estimate the underlying counterfactual means $M_{1t} = f(\theta_1, \rho_t)$ for $T_0 < t \le T$ by using an appropriate combination of the post-intervention observations from the donor pool since the donor units are immune to the treatment.

To render this feasible, we make the key operating assumption (as done similarly in literature cf. Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003)) that the mean vector of the treatment unit over the pre-intervention period, i.e. the vector $M_1^- = [M_{1t}]_{t \le T_0}$, lies within the span of the mean vectors within the donor pool over the pre-intervention period, i.e. the span of the donor mean vectors $M_i^- = [M_{it}]_{2 \le i \le N, t \le T_0}$ [1]. More precisely, we assume there exists a set of weights $\beta^* \in \mathbb{R}^{N-1}$ such that for all $t \le T_0$,

$$M_{1t} = \sum_{i=2}^N \beta_i^* M_{it}. \tag{6}$$

This is a reasonable and intuitive assumption, utilized in literature, hypothesizing that the treatment unit can be modeled as some combination of the donor pool. In fact, the set of weights β^* are the very definition of a synthetic control. Note, however, that in contrast to Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003), we do not constrain the weights to be non-negative and sum to 1. This may reduce the interpretability of the synthetic control produced. We discuss ways to increase interpretability using our method in Section 3.4.2.

In order to distinguish the pre- and post-intervention periods, we use the following notation for all (donor) matrices: $\boldsymbol{A} = [\boldsymbol{A}^-, \boldsymbol{A}^+]$, where $\boldsymbol{A}^- = [A_{ij}]_{2 \le i \le N, j \in [T_0]}$ and $\boldsymbol{A}^+ = [A_{ij}]_{2 \le i \le N, T_0 < j \le T}$ denote the pre- and post-intervention submatrices, respectively; vectors will be defined in the same manner, i.e. $A_i = [A_i^-, A_i^+]$, where $A_i^- = [A_{it}]_{t \in [T_0]}$ and $A_i^+ = [A_{it}]_{T_0 < t \le T}$ denote the pre- and post-intervention subvectors, respectively, for the ith donor. Moreover, we will denote all vectors related to the treatment unit with the subscript "1", e.g. $A_1 = [A_1^-, A_1^+]$.

In contrast with the classical synthetic control work, we allow our model to be robust to incomplete observations. To model randomly missing data, the algorithm observes each data point X_{it} in the donor pool with probability $p \in (0, 1]$, independently of all other entries. While the assumption that p is constant across all rows and columns of our observation matrix is standard in literature, our results remain valid even in situations where the probability of observation is dependent on the row

1. We note that this is a minor departure from the literature on synthetic control starting in Abadie and Gardeazabal (2003) – in literature, the pre-intervention *noisy* observation (rather than the mean) vector X_1, is assumed to be a *convex* (rather than linear) combination of the noisy donor observations. We believe our setup is more reasonable since we do not want to fit noise.

and column latent parameters, i.e. $p_{ij} = g(\theta_i, \rho_j) \in (0, 1]$. In such situations, p_{ij} can be estimated as $\hat{p}_{ij}$ using consistent graphon estimation techniques described in a growing body of literature, e.g. see Borgs et al. (2017); Chatterjee (2015); Wolfe and Olhede; Yang et al.. These estimates can then be used in our analysis presented in Section 4.

3. Algorithm

3.1 Intuition.

We begin by exploring the intuition behind our proposed two-step algorithm: (1) *de-noising the data:* since the singular values of our observation matrix, $\boldsymbol{X} = [X_{it}]_{2 \le i \le N, t \in [T]}$, encode both signal and noise, we aim to discover a low rank approximation of $\boldsymbol{X}$ that only incorporates the singular values associated with useful information; simultaneously, this procedure will naturally impute any missing observations. We note that this procedure is similar to the algorithm proposed in Chatterjee (2015). (2) *learning β^*:* using the pre-intervention portion of the de-noised matrix, we learn the linear relationship between the treatment unit and the donor pool prior to estimating the post-intervention counterfactual outcomes. Since our objective is to produce accurate predictions, it is not obvious why the synthetic treatment unit should be a convex combination of its donor pool as assumed in Abadie et al. (2010); Abadie and Gardeazabal (2003); Abadie et al. (2014). In fact, one can reasonably expect that the treatment unit and some of the donor units may exhibit negative correlations with one another. In light of this intuition, we learn the optimal set of weights via linear regression, allowing for both positive and negative elements.

3.2 Robust algorithm (algorithm 1).

We present the details of our robust method in Algorithm 1 below. The algorithm utilizes two hyperparameters: (1) a thresholding hyperparameter $\mu \ge 0$, which serves as a knob to effectively trade-off between the bias and variance of the estimator, and (2) a regularization hyperameter $\eta \ge 0$ that controls the model complexity. We discuss the procedure for determining the hyperparameters in Section 3.4. To simplify the exposition, we assume the entries of $\boldsymbol{X}$ are bounded by one in absolute value, i.e. $|X_{it}| \le 1$.

3.3 Bayesian algorithm: measuring uncertainty (algorithm 2).

In order to quantitatively assess the uncertainty of our model, we will transition from a frequentist perspective to a Bayesian viewpoint. As commonly assumed in literature, we consider a zero-mean, isotropic Gaussian noise model (i.e. $\epsilon \sim \mathcal{N}(0, \sigma^2 \boldsymbol{I})$) and use the square loss for our cost function. We present the Bayesian method as Algorithm 2. Note that we perform step one of our robust algorithm exactly as in Algorithm 1; as a result, we only detail the alterations of step two in the Bayesian version (Algorithm 2).

3.4 Algorithmic features: the fine print.

3.4.1 Bounded entries transformation.

Several of our results, as well as the algorithm we propose, assume that the observation matrix is bounded such that $|X_{it}| \le 1$. For any data matrix, we can achieve this by using the following pre-processing transformation: suppose the entries of $\boldsymbol{X}$ belong to an interval $[a, b]$. Then, one can first pre-process the matrix $\boldsymbol{X}$ by subtracting $(a+b)/2$ from each entry, and dividing by $(b-a)/2$ to enforce that the entries lie in the range $[-1, 1]$. The reverse transformation, which can be applied at the end of the algorithm description above, returns a matrix with values contained in the original

Algorithm 1 Robust synthetic control

Step 1. De-noising the data: singular value thresholding (inspired by Chatterjee (2015)).

1. Define $\boldsymbol{Y} = [Y_{it}]_{2 \le i \le N, t \in [T]}$ with

$$Y_{it} = \begin{cases} X_{it} & \text{if } X_{it} \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

2. Compute the singular value decomposition of $\boldsymbol{Y}$:

$$\boldsymbol{Y} = \sum_{i=1}^{N-1} s_i u_i v_i^T. \tag{8}$$

3. Let $S = \{i : s_i \ge \mu\}$ be the set of singular values above the threshold μ.

4. Define the estimator of $\boldsymbol{M}$ as

$$\hat{\boldsymbol{M}} = \frac{1}{\hat{p}} \sum_{i \in S} s_i u_i v_i^T, \tag{9}$$

where $\hat{p}$ is the maximum of the fraction of observed entries in $\boldsymbol{X}$ and $\frac{1}{(N-1)T}$.

Step 2. Learning and projecting

1. For any $\eta \ge 0$, let

$$\hat{\beta}(\eta) = \underset{v \in \mathbb{R}^{N-1}}{\arg\min} \left\| Y_1^- - (\hat{\boldsymbol{M}}^-)^T v \right\|^2 + \eta \|v\|^2. \tag{10}$$

2. Define the counterfactual means for the treatment unit as

$$\hat{M}_1 = \hat{\boldsymbol{M}}^T \hat{\beta}(\eta). \tag{11}$$

Algorithm 2 Bayesian robust synthetic control

Step 2. Learning and projecting

1. Estimate the noise variance via (bias-corrected) maximum likelihood, i.e.

$$\hat{\sigma}^2 = \frac{1}{T_0 - 1} \sum_{t=1}^{T_0} (Y_{1t} - \bar{Y})^2, \tag{12}$$

   where $\bar{Y}$ denotes the pre-intervention sample mean.

2. Compute posterior distribution parameters for an appropriate choice of the prior α:

$$\boldsymbol{\Sigma}_D = \left(\frac{1}{\hat{\sigma}^2} \hat{\boldsymbol{M}}^- (\hat{\boldsymbol{M}}^-)^T + \alpha \boldsymbol{I} \right)^{-1} \tag{13}$$

$$\beta_D = \frac{1}{\hat{\sigma}^2} \boldsymbol{\Sigma}_D \hat{\boldsymbol{M}}^- Y_1^-. \tag{14}$$

3. Define the counterfactual means for the treatment unit as

$$\hat{M}_1 = \hat{\boldsymbol{M}}^T \beta_D. \tag{15}$$

4. For each time instance $t \in [T]$, compute the model uncertainty (variance) as

$$\sigma_D^2(\hat{M}_{\cdot,t}) = \hat{\sigma}^2 + \hat{M}_{\cdot,t}^T \boldsymbol{\Sigma}_D \hat{M}_{\cdot,t}, \tag{16}$$

   where $\hat{M}_{\cdot,t} = [\hat{M}_{it}]_{2 \le i \le N}$ is the de-noised vector of donor outcomes at time t.

range. Specifically, the reverse transformation equates to multiplying the end result by $(b-a)/2$ and adding by $(a+b)/2$.

3.4.2 Solution interpretability.

For the practitioner who seeks a more interpretable solution, e.g. a convex combination of donors as per the original synthetic control estimator of Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003), we recommend using an ℓ_1-regularization penalty in the learning procedure of step 2. Due to the geometry of LASSO, the resulting estimator will often be a sparse vector; in other words, LASSO effectively performs model selection and selects the most important donors that comprise the synthetic control. LASSO can also be beneficial if the number of donors exceeds the number of pre-treatment time periods. Specifically, for any $\eta > 0$, we define the LASSO estimator to be

$$\hat{\beta}(\eta) = \underset{v \in \mathbb{R}^{N-1}}{\arg\min} \left\| Y_1^- - (\hat{\boldsymbol{M}}^-)^T v \right\|^2 + \eta \|v\|_1.$$

For the purposes of this paper, we focus our attention on ridge regression and provide theoretical results in Section 4 under the ℓ_2-regularization setting. However, a closer examination of the LASSO estimator for synthetic control methods can be found in Li and Bell (2017).

3.4.3 Choosing the hyperparameters.

Here, we discuss several approaches to choosing the hyperparameter μ for the singular values; note that μ defines the set S that includes the singular values we wish to include in the imputation procedure. If it is known a priori that the underlying model is low rank with rank at most k, then it may make sense to choose μ such that $|S| = k$. A data driven approach, however, could be implemented based on cross-validation. Precisely, reserve a portion of the pre-intervention period for validation, and use the rest of the pre-intervention data to produce an estimate $\hat{\beta}(\eta)$ for each of the finitely many choices of μ $(s_1, \ldots, s_{N-1})$. Using each estimate $\hat{\beta}(\eta)$, produce its corresponding treatment unit mean vector over the validation period. Then, select the μ that achieves the minimum MSE with respect to the observed data. Finally, Chatterjee (2015) provides a universal approach to picking a threshold; similarly, we also propose another such universal threshold, (20), in Section 4.1. We utilize the data driven approach in our experiments in this work.

The regularization parameter, η, also plays a crucial role in learning the synthetic control and influences both the training and generalization errors. As is often the case in model selection, a popular strategy in estimating the ideal η is to employ cross-validation as described above. However, since time-series data often have a natural temporal ordering with causal effects, we also recommend employing the forward chaining strategy. Although the forward chaining strategy is similar to leave-one-out (LOO) cross-validation, an important distinction is that forward chaining does not break the temporal ordering in the training data. More specifically, for a particular candidate of η at every iteration $t \in [T_0]$, the learning process uses $[Y_{11}, \ldots, Y_{1,t-1}]$ as the training portion while reserving Y_{1t} as the validation point. As before, the average error is then computed and used to evaluate the model (characterized by the choice of η). The forward chaining strategy can also be used to learn the optimal μ.

3.4.4 Scalability.

In terms of scalability, the most computationally demanding procedure is that of evaluating the singular value decomposition (SVD) of the observation matrix. Given the ubiquity of SVD methods in the realm of machine learning, there are well-known techniques that enable computational and storage scaling for SVD algorithms. For instance, both Spark (through alternative least squares) and Tensor-Flow come with built-in SVD implementations. As a result, by utilizing the appropriate

computational infrastructure, our de-noising procedure, and algorithm in generality, can scale quite well. Also note that for a low rank structure, we typically only need to compute the top few singular values and vectors. Various truncated-SVD algorithms provide resource-efficient implementations to compute the top k singular values and vectors instead of the complete-SVD.

3.4.5 Low rank hypothesis.

The factor models that are commonly used in the Econometrics literature, cf. Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003), often lead to a low rank structure for the underlying mean matrix $\boldsymbol{M}$. When f is nonlinear, $\boldsymbol{M}$ can still be well approximated by a low rank matrix for a large class of functions. For instance, if the latent parameters assumed values from a bounded, compact set, and if f was Lipschitz continuous, then it can be argued that $\boldsymbol{M}$ is well approximated by a low rank matrix, cf. see Chatterjee (2015) for a very simple proof. As the reader will notice, while we establish results for low rank matrix, the results of this work are robust to low rank approximations whereby the approximation error can be viewed as "noise". Lastly, as shown in Udell and Townsend (2017), many latent variable models can be well approximated (up to arbitrary accuracy ϵ) by low rank matrices. Specifically, Udell and Townsend (2017) shows that the corresponding low rank approximation matrices associated with "nice" functions (e.g. linear functions, polynomials, kernels, etc.) are of log-rank.

3.4.6 Covariate information.

Although the algorithm does not appear to rely on any helpful covariate information and the experimental results, presented in Section 5, suggest that it performs on par with that of the original synthetic control algorithm, we want to emphasize that we are not suggesting that practitioners should abandon the use of any additional covariate information or the application of domain knowledge. Rather, we believe that our key algorithmic feature – the de-noising step – may render covariates and domain expertise as luxuries as opposed to necessities for many practical applications. If the practitioner has access to supplementary predictor variables, we propose that step one of our algorithm be used as a pre-processing routine for de-noising the data before incorporating additional information. Moreover, other than the obvious benefit of narrowing the donor pool, domain expertise can also come in handy in various settings, such as determining the appropriate method for imputing the missing entries in the data. For instance, if it is known a priori that there is a trend or periodicity in the time series evolution for the units, it may behoove the practitioner to impute the missing entries using "nearest-neighbors" or linear interpolation.

4. Theoretical Results

In this section, we derive the finite sample and asymptotic properties of the estimators $\hat{\boldsymbol{M}}$ and $\hat{M}_1$. We begin by defining necessary notations and recalling a few operating assumptions prior to presenting the results, with the corresponding proofs relegated to the Appendix. To that end, we re-write (3) in matrix form as $\boldsymbol{X} = \boldsymbol{M} + \boldsymbol{E}$, where $\boldsymbol{E} = [\epsilon_{it}]_{2 \le i \le N, t \in [T]}$ denotes the noise matrix. We shall assume that the noise parameters ϵ_{it} are independent zero-mean random variables with bounded second moments. Specifically, for all $2 \le i \le N, t \in [T]$,

$$\mathbb{E}[\epsilon_{it}] = 0, \quad \text{and} \quad \text{Var}(\epsilon_{it}) = \sigma^2. \tag{17}$$

We shall also assume that the treatment unit noise in (5) obeys (17). Further, we assume the relationship in (6) holds. To simplify the following exposition, we assume that $|M_{ij}| \le 1$ and $|X_{ij}| \le 1$.

As previously discussed, we evaluate the accuracy of our estimated means for the treatment unit with respect to the deviation between $\hat{M}_1$ and M_1 measured in ℓ_2-norm, and similarly between $\hat{\boldsymbol{M}}$ and $\boldsymbol{M}$. Additionally, we aim to establish the validity of our pre-intervention linear model assumption (cf. (6)) and investigate how the linear relationship translates over to the post-intervention regime, i.e. if $M_1^- = (\boldsymbol{M}^-)^T \beta^*$ for some β^*, does M_1^+ (approximately) equal to $(\boldsymbol{M}^+)^T \beta^*$? If so, under what conditions? We present our results for the above aspects after a brief motivation of ℓ_2 regularization.

Combatting overfitting. One weapon to combat overfitting is to constrain the learning algorithm to limit the effective model complexity by fitting the data under a simpler hypothesis. This technique is known as regularization, and it has been widely used in practice. To employ regularization, we introduce a complexity penalty term into the objective function (10). For a general regularizer, the objective function takes the form

$$\hat{\beta}(\eta) = \underset{v \in \mathbb{R}^{N-1}}{\arg\min} \left\| Y_1^- - (\hat{\boldsymbol{M}}^-)^T v \right\|^2 + \eta \sum_{j=1}^{N-1} |v_j|^q, \tag{18}$$

for some choice of positive constants η and q. The first term measures the empirical error of the model on the given dataset, while the second term penalizes models that are too "complex" by controlling the "smoothness" of the model in order to avoid overfitting. In general, the impact/trade-off of regularization can be controlled by the value of the regularization parameter η. Via the use of Lagrange multipliers, we note that minimizing (18) is equivalent to minimizing (10) subject to the constraint that

$$\sum_{j=1}^{N-1} |v_j|^q \le c,$$

for some appropriate value of c. When $q = 2$, (18) corresponds to the classical setup known as *ridge regression* or *weight decay*. The case of $q = 1$ is known as the LASSO in the statistics literature; the ℓ_1-norm regularization of LASSO is a popular heuristic for finding a sparse solution. In either case, incorporating an additional regularization term encourages the learning algorithm to output a simpler model with respect to some measure of complexity, which helps the algorithm avoid overfitting to the idiosyncrasies within the observed dataset. Although the training error may suffer from the simpler model, empirical studies have demonstrated that the generalization error can be greatly improved under this new setting. Throughout this section, we will primarily focus our attention on the case of $q = 2$, which maintains our learning objective to be (convex) quadratic in the parameter v so that its exact minimizer can be found in closed form:

$$\hat{\beta}(\eta) = \left(\hat{\boldsymbol{M}}^- (\hat{\boldsymbol{M}}^-)^T + \eta \boldsymbol{I} \right)^{-1} \hat{\boldsymbol{M}}^- Y_1^-. \tag{19}$$

4.1 Imputation analysis.

In this section, we highlight the importance of our de-noising procedure and prescribe a universal threshold (similar to that of Chatterjee (2015)) that dexterously distinguishes signal from noise, enabling the algorithm to capture the appropriate amount of useful information (encoded in the singular values of $\boldsymbol{Y}$) while discarding out the randomness. Due to its universality, the threshold naturally adapts to the amount of structure within $\boldsymbol{M}$ in a purely data-driven manner. Specifically, for any choice of $\omega \in (0.1, 1)$, we find that choosing

$$\mu = (2 + \omega) \sqrt{T(\hat{\sigma}^2 \hat{p} + \hat{p}(1 - \hat{p}))}, \tag{20}$$

results in an estimator with strong theoretical properties for both interpolation and extrapolation (discussed in Section 4.2). Here, $\hat{p}$ and $\hat{\sigma}^2$ denote the unbiased maximum likelihood estimates of p and σ^2, respectively, and can be computed via (9) and (12).

The following Theorems (adapted from Theorems 2.1 and 2.7 of Chatterjee (2015)) demonstrate that Step 1 of our algorithm (detailed in Section 3.2) accurately imputes missing entries within our data matrix $\boldsymbol{X}$ when the signal matrix $\boldsymbol{M}$ is either low rank or generated by an $\mathcal{L}$-Lipschitz function. In particular, Theorems 1 and 2 state that Step 1 produces a consistent estimator of the underlying mean matrix $\boldsymbol{M}$ with respect to the (matrix) mean-squared-error, which is defined as

$$\text{MSE}(\hat{\boldsymbol{M}}) = \mathbb{E}\left[\frac{1}{(N-1)T}\sum_{i=2}^{N}\sum_{j=1}^{T}(\hat{M}_{ij} - M_{ij})^2\right]. \tag{21}$$

We say that $\hat{\boldsymbol{M}}$ is a consistent estimator of $\boldsymbol{M}$ if the right-hand side of (21) converges to zero as N and T grow without bound.

The following theorem demonstrates that $\hat{\boldsymbol{M}}$ is a good estimate of $\boldsymbol{M}$ when $\boldsymbol{M}$ is a low rank matrix, particularly when the rank of $\boldsymbol{M}$ is small compared to $(N-1)p$.

Theorem 1 (**Theorem 2.1 of Chatterjee (2015)**) *Suppose that $\boldsymbol{M}$ is rank k. Suppose that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Then using μ as defined in* (20)*,*

$$\text{MSE}(\hat{\boldsymbol{M}}) \leq C_1\sqrt{\frac{k}{(N-1)p}} + \mathcal{O}\Big(\frac{1}{(N-1)T}\Big), \tag{22}$$

where C_1 is a universal positive constant.

Suppose that the latent row and column feature vectors, $\{\theta_i\}$ and $\{\rho_j\}$, belong to some bounded, closed sets $K \subset \mathbb{R}^d$, where d is some arbitrary but fixed dimension. If we assume $f : K \times K \to [-1, 1]$ possesses desirable smoothness properties such as Lipschitzness, then $\hat{\boldsymbol{M}}$ is again a good estimate of $\boldsymbol{M}$.

Theorem 2 (**Theorem 2.7 of Chatterjee (2015)**) *Suppose f is a $\mathcal{L}$-Lipschitz function. Suppose that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Then using μ as defined in* (20)*,*

$$\text{MSE}(\hat{\boldsymbol{M}}) \leq C(K, d, \mathcal{L})\frac{(N-1)^{-\frac{1}{d+2}}}{\sqrt{p}} + \mathcal{O}\Big(\frac{1}{(N-1)T}\Big), \tag{23}$$

where $C(K, d, \mathcal{L})$ is a constant depending on K, d, and $\mathcal{L}$.

It is important to observe that the models under consideration for both Theorems 1 and 2 encompass the mean matrices, $\boldsymbol{M}$, generated as per many of the popular Econometric factor models often considered in literature and assumed in practice. Therefore, de-noising the data serves as an important imputing and filtering procedure for a wide array of applications.

4.2 Forecasting analysis: pre-intervention regime.

Similar to the setting for interpolation, the prediction performance metric of interest is the average mean-squared-error in estimating M_1^- using $\hat{M}_1^-$. Precisely, we define

$$\text{MSE}(\hat{M}_1^-) = \mathbb{E}\left[\frac{1}{T_0}\sum_{t=1}^{T_0}(M_{1t} - \hat{M}_{1t})^2\right]. \tag{24}$$

If the right-hand side of (33) approaches zero in the limit as T_0 grows without bound, then we say that $\hat{M}_1^-$ is a consistent estimator of M_1^- (note that our analysis here assumes that only $T_0 \to \infty$).

In what follows, we first state the finite sample bound on the average MSE between $\hat{M}_1^-$ and M_1^- for the most generic setup (Theorem 3). As a main Corollary of the result, we specialize the bound in the case where we use our prescribed universal threshold. Finally, we discuss a minor variation of the algorithm where the data is pre-processed, and specialize the above result to establish the consistency of our estimator (Theorem 5).

4.2.1 General result.

We provide a finite sample error bound for the most generic setting, i.e. for any choice of the threshold, μ, and regularization hyperparameter, η.

Theorem 3 *Let S denote the set of singular values included in the imputation procedure, i.e., the set of singular values greater than μ. Then for any $\eta \geq 0$ and $\mu \geq 0$, the pre-intervention error of the algorithm can be bounded as*

$$\text{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p^2T_0}\mathbb{E}\Big(\lambda^* + \|\boldsymbol{Y} - p\boldsymbol{M}\| + \big\|(\hat{p} - p)\boldsymbol{M}^-\big\|\Big)^2 + \frac{2\sigma^2|S|}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_2e^{-cp(N-1)T}. \tag{25}$$

Here, $\lambda_1, \ldots, \lambda_{N-1}$ are the singular values of $p\boldsymbol{M}$ in decreasing order and repeated by multiplicities, with $\lambda^* = \max_{i \notin S}\lambda_i$; C_1, C_2 and c are universal positive constants.

Bias-variance tradeoff. Let us interpret the result by parsing the terms in the error bound. The last term decays exponentially with $(N-1)T$, as long as the fraction of observed entries is such that, on average, we see a super-constant number of entries, i.e. $p(N-1)T \gg 1$. More interestingly, the first two terms highlight the "bias-variance tradeoff" of the algorithm with respect to the singular value threshold μ. Precisely, the size of the set S increases with a decreasing value of the hyperparameter μ, causing the second error term to increase. Simultaneously, however, this leads to a decrease in λ^*. Note that λ^* denotes the aspect of the "signal" within the matrix $\boldsymbol{M}$ that is not captured due to the thresholding through S. On the other hand, the second term, $|S|\sigma^2/T_0$, represents the amount of "noise" captured by the algorithm, but wrongfully interpreted as a signal, during the thresholding process. In other words, if we use a large threshold, then our model may fail to capture pertinent information encoded in $\boldsymbol{M}$; if we use a small threshold, then the algorithm may overfit the spurious patterns in the data. Thus, the hyperparameter μ provides a way to trade-off "bias" (first term) and "variance" (second term).

4.2.2 Goldilocks principle: a universal threshold.

Using the universal threshold defined in (20), we now highlight the prediction power of our estimator for any choice of η, the regularization hyperparameter. As described in Section 4.1, the prescribed threshold automatically captures the "correct" level of information encoded in the (noisy) singular values of $\boldsymbol{Y}$ in a data-driven manner, dependent on the structure of $\boldsymbol{M}$. However, unlike the statements in Theorems 1 and 2, the following bound does not require $\boldsymbol{M}$ to be low rank or f to be Lipschitz.

Corollary 4 *Suppose $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Let $T \leq \alpha T_0$ for some constant $\alpha > 1$. Then for any $\eta \geq 0$ and using μ as defined in* (20)*, the pre-intervention error is bounded above by*

$$\text{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p}(\sigma^2 + (1-p)) + \mathcal{O}(1/\sqrt{T_0}), \tag{26}$$

where C_1 is a universal positive constant.

As an implication, if $p = (1+\vartheta)\sqrt{T_0}/(1+\sqrt{T_0})$ and $\sigma^2 \leq \vartheta$, we have that $\text{MSE}(\hat{M}_1^-) = \mathcal{O}(1/\sqrt{T_0})$. More generally, Corollary 4 shows that by adroitly capturing the signal, the resulting error bound simply depends on the variance of the noise terms, σ^2, and the error introduced due to missing data. Ideally, one would hope to overcome the error term when T_0 is sufficiently large. This motivates the following setup.

4.2.3 Consistency.

We present a straightforward pre-processing step that leads to the consistency of our algorithm. The pre-processing step simply involves replacing the columns of $\boldsymbol{X}$ by the averages of subsets of its columns. This admits the same setup as before, but with the variance for each noise term reduced. An implicit side benefit of this approach is that required SVD step in the algorithm is now applied to a matrix of smaller dimensions.

To begin, partition the T_0 columns of the pre-intervention data matrix $\boldsymbol{X}^-$ into Δ blocks, each of size $\tau = \lfloor T_0/\Delta \rfloor$ except potentially the last block, which we shall ignore for theoretical purposes; in practice, however, the remaining columns can be placed into the last block. Let $B_j = \{(j-1)\tau + \ell : 1 \le \ell \le \tau\}$ denote the column indices of $\boldsymbol{X}^-$ within partition $j \in [\Delta]$. Next, we replace the τ columns within each partition by their average, and thus create a new matrix, $\bar{\boldsymbol{X}}^-$, with Δ columns and $N-1$ rows. Precisely, $\bar{\boldsymbol{X}}^- = [\bar{X}_{ij}]_{2 \le i \le N, j \in [\Delta]}$ with

$$\bar{X}_{ij} = \frac{1}{\tau} \sum_{t \in B_j} X_{it} \cdot D_{it}, \tag{27}$$

where

$$D_{it} = \begin{cases} 1 & \text{if } X_{it} \text{ is observed,} \\ 0 & \text{otherwise.} \end{cases}$$

For the treatment row, let $\bar{X}_{1j} = \frac{\hat{p}}{\tau} \sum_{t \in B_j} X_{1t}$ for all $j \in [\Delta]$[2]. Let $\bar{\boldsymbol{M}}^- = [\bar{M}_{ij}]_{2 \le i \le N, j \in [\Delta]}$ with

$$\bar{M}_{ij} = \mathbb{E}[\bar{X}_{ij}] \ = \frac{p}{\tau} \sum_{t \in B_j} M_{it}. \tag{28}$$

We apply the algorithm to $\bar{\boldsymbol{X}}^-$ to produce the estimate $\hat{\bar{\boldsymbol{M}}}^-$ of $\bar{\boldsymbol{M}}^-$, which is sufficient to produce $\hat{\beta}(\eta)$. This $\hat{\beta}(\eta)$ can be used to produce the post-intervention synthetic control means $\hat{M}_1^+ = [\hat{M}_{1t}]_{T_0 < t \le T}$ in a similar manner as before [3]: for $T_0 < t \le T$,

$$\hat{M}_{1t} = \sum_{i=2}^{N} \hat{\beta}_i(\eta) X_{it}. \tag{29}$$

For the pre-intervention period, we produce the estimator $\hat{\bar{M}}_1^- = [\hat{\bar{M}}_{1j}]_{j \in [\Delta]}$: for $j \in [\Delta]$,

$$\hat{\bar{M}}_{1j} = \sum_{i=2}^{N} \hat{\beta}_i(\eta) \hat{\bar{M}}_{ij}. \tag{30}$$

Our measure of estimation error is defined as

$$\mathrm{MSE}(\hat{\bar{M}}_1^-) = \mathbb{E}\left[\frac{1}{\Delta} \sum_{j=1}^{\Delta} (\bar{M}_{1j} - \hat{\bar{M}}_{1j})^2\right]. \tag{31}$$

For simplicity, we will analyze the case where each block contains at least one entry such that $\bar{\boldsymbol{X}}^-$ is completely observed. We now state the following result.

2. Although the statement in Theorem 5 assumes that an oracle provides the true p, we prescribe practitioners to use $\hat{p}$ since $\hat{p}$ converges to p almost surely by the Strong Law of Large Numbers.
3. In practice, one can first de-noise $\boldsymbol{X}^+$ via step one of Section 3, and use the entries of $\hat{\boldsymbol{M}}^+$ in (29).

Theorem 5 *Fix any $\gamma \in (0, 1/2)$ and $\omega \in (0.1, 1)$. Let $\Delta = T_0^{\frac{1}{2}+\gamma}$ and $\mu = (2+\omega)\sqrt{T_0^{2\gamma}(\hat{\sigma}^2\hat{p} + \hat{p}(1-\hat{p}))}$. Suppose $p \ge \frac{T_0^{-2\gamma}}{\sigma^2+1}$ is known. Then for any $\eta \ge 0$,*

$$\mathrm{MSE}(\hat{\bar{M}}_1^-) = \mathcal{O}(T_0^{-1/2+\gamma}). \tag{32}$$

We note that the method of (Abadie and Gardeazabal, 2003, Sec 2.3) learns the weights (here $\hat{\beta}(0)$) by pre-processing the data. One common pre-processing proposal is to also aggregate the columns, but the aggregation parameters are chosen by solving an optimization problem to minimize the resulting prediction error of the observations. In that sense, the above averaging of column is a simple, data agnostic approach to achieve a similar effect, and potentially more effectively.

4.3 Forecasting analysis: post-intervention regime.

For the post-intervention regime, we consider the average root-mean-squared-error in measuring the performance of our algorithm. Precisely, we define

$$\mathrm{RMSE}(\hat{M}_1^+) = \mathbb{E}\left[\frac{1}{\sqrt{T - T_0}} \left(\sum_{t > T_0}^{T} (M_{1t} - \hat{M}_{1t})^2\right)^{1/2}\right]. \tag{33}$$

The key assumption of our analysis is that the treatment unit signal can be written as a linear combination of donor pool signals. Specifically, we assume that this relationship holds in the pre-intervention regime, i.e. $M_1^- = (\boldsymbol{M}^-)^T \beta^*$ for some $\beta^* \in \mathbb{R}^{N-1}$ as stated in (6). However, the question still remains: does the same relationship hold for the post-intervention regime and if so, under what conditions does it hold? We state a simple linear algebraic fact to this effect, justifying the approach of synthetic control. It is worth noting that this important aspect has been amiss in the literature, potentially implicitly believed or assumed starting in the work by Abadie and Gardeazabal (2003).

Theorem 6 *Let Equation (6) hold for some β^*. Let $\mathrm{rank}(\boldsymbol{M}^-) = \mathrm{rank}(\boldsymbol{M})$. Then $M_1^+ = (\boldsymbol{M}^+)^T \beta^*$.*

If we assume that the linear relationship prevails in the post-intervention period, then we arrive at the following error bound.

Theorem 7 *Suppose $p \ge \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Suppose $\left\|\hat{\beta}(\eta)\right\|_\infty \le \psi$ for some $\psi > 0$. Let $\alpha' T_0 \le T \le \alpha T_0$ for some constants $\alpha', \alpha > 1$. Then for any $\eta \ge 0$ and using μ as defined in (20), the post-intervention error is bounded above by*

$$\mathrm{RMSE}(\hat{M}_1^+) \le \frac{C_1}{\sqrt{p}}(\sigma^2 + (1-p))^{1/2} + \frac{C_2 \|\boldsymbol{M}\|}{\sqrt{T_0}} \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| + \mathcal{O}(1/\sqrt{T_0}),$$

where C_1 and C_2 are universal positive constants.

Benefits of regularization. In order to motivate the use of regularization, we analyze the error bounds of Theorems 3 and 7 to observe how the pre- and post-intervention errors react to regularization. As seen from Theorem 3, the pre-intervention error *increases* linearly with respect to the choice of η. Intuitively, this increase in pre-intervention error derives from the fact that regularization reduces the model complexity, which biases the model and handicaps its ability to fit the data. At the same time, by restricting the hypothesis space and controlling the "smoothness" of the model, regularization prevents the model from overfitting to the data, which better equips the model to generalize to unseen

data. Therefore, a larger value of η *reduces* the post-intervention error. This can be seen by observing the second error term of Theorem 7, which is controlled by the expression $\left\|\hat{\beta}(\eta) - \beta^*\right\|$. In words, this error is a function of the learning algorithm used to estimate β^*. Interestingly, Farebrother (1976) demonstrates that there exists an $\eta > 0$ such that

$$\text{MSE}(\hat{\beta}(\eta)) \leq \text{MSE}(\hat{\beta}(0)),$$

without any assumptions on the rank of $\hat{\boldsymbol{M}}^-$. In other words, Farebrother (1976) demonstrates that regularization can decrease the MSE between $\hat{\beta}(\eta)$ and the true β^*, thus reducing the overall error. Ultimately, employing ridge regression introduces extraneous bias into our model, yielding a higher pre-intervention error. In exchange, regularization reduces the post-intervention error (due to smaller variance).

4.4 Bayesian analysis.

We now present a Bayesian treatment of synthetic control. By operating under a Bayesian framework, we allow practitioners to naturally encode domain knowledge into prior distributions while simultaneously avoiding the problem of overfitting. In addition, rather than making point estimates, we can now quantitatively express the uncertainty in our estimates with posterior probability distributions.

We begin by treating β^* as a random variable as opposed to an unknown constant. In this approach, we specify a prior distribution, $p(\beta)$, that expresses our apriori beliefs and preferences about the underlying parameter (synthetic control). Given some new observation for the donor units, our goal is to make predictions for the counterfactual treatment unit on the basis of a set of pre-intervention (training) data. For the moment, let us assume that the noise parameter σ^2 is a known quantity and that the noise is drawn from a Gaussian distribution with zero-mean; similarly, we temporarily assume $\boldsymbol{M}^-$ is also given. Let us denote the vector of donor estimates as $M_{\cdot t} = [M_{it}]_{2 \leq i \leq N}$; we define $X_{\cdot t}$ similarly. Denoting the pre-intervention data as $D = \{(Y_{1t}, M_{\cdot t}) : t \in [T_0]\}$, the likelihood function $p(Y_1^- \mid \beta, \boldsymbol{M}^-)$ is expressed as

$$p(Y_1^- \mid \beta, \hat{\boldsymbol{M}}^-) = \mathcal{N}((\boldsymbol{M}^-)^T\beta, \sigma^2\boldsymbol{I}), \tag{34}$$

an exponential of a quadratic function of β. The corresponding conjugate prior, $p(\beta)$, is therefore given by a Gaussian distribution, i.e. $\beta \sim \mathcal{N}(\beta \mid \beta_0, \boldsymbol{\Sigma}_0)$ with mean β_0 and covariance Σ_0. By using a conjugate Gaussian prior, the posterior distribution, which is proportional to the product of the likelihood and the prior, will also be Gaussian. Applying Bayes' Theorem (derivation unveiled below in Section G in the Appendix, we have that the posterior distribution is $p(\beta \mid D) = \mathcal{N}(\beta_D, \boldsymbol{\Sigma}_D)$ where

$$\boldsymbol{\Sigma}_D = \left(\boldsymbol{\Sigma}_0^{-1} + \frac{1}{\sigma^2}\boldsymbol{M}^-(\boldsymbol{M}^-)^T\right)^{-1} \tag{35}$$

$$\beta_D = \boldsymbol{\Sigma}_D\left(\frac{1}{\sigma^2}\boldsymbol{M}^-Y_1^- + \boldsymbol{\Sigma}_0^{-1}\beta_0\right). \tag{36}$$

For the remainder of this section, we shall consider a popular form of the Gaussian prior. In particular, we consider a zero-mean isotropic Gaussian with the following parameters: $\beta_0 = 0$ and $\boldsymbol{\Sigma}_0 = \alpha^{-1}\boldsymbol{I}$ for some choice of $\alpha > 0$. Since $\boldsymbol{M}^-$ is unobserved by the algorithm, we use the estimated $\hat{\boldsymbol{M}}^-$, computed as per step one of Section 3, as a proxy; therefore, we redefine our data as $D = \{(Y_{1t}, \hat{M}_{\cdot t}) : t \in [T_0]\}$.

Putting everything together, we have that $p(\beta \mid D) = \mathcal{N}(\beta_D, \boldsymbol{\Sigma}_D)$ whereby

$$\boldsymbol{\Sigma}_D = \left(\alpha\boldsymbol{I} + \frac{1}{\sigma^2}\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^T\right)^{-1} \tag{37}$$

$$\beta_D = \frac{1}{\sigma^2}\boldsymbol{\Sigma}_D\hat{\boldsymbol{M}}^-Y_1^- \tag{38}$$

$$= \frac{1}{\sigma^2}\left(\frac{1}{\sigma^2}\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^T + \alpha\boldsymbol{I}\right)^{-1}\hat{\boldsymbol{M}}^-Y_1^-. \tag{39}$$

4.4.1 Maximum a posteriori (MAP) estimation.

By using the zero-mean, isotropic Gaussian conjugate prior, we can derive a point estimate of β^* by maximizing the log posterior distribution, which we will show is equivalent to minimizing the regularized objective function of (10) for a particular choice of η. In essence, we are determining the optimal $\hat{\beta}$ by finding the most probable value of β^* given the data and under the influence of our prior beliefs. The resulting estimate is known as the maximum a posteriori (MAP) estimate.

We begin by taking the log of the posterior distribution, which gives the form

$$\ln p(\beta \mid D) = -\frac{1}{2\sigma^2}\left\|Y_1^- - (\hat{\boldsymbol{M}}^-)^T\beta\right\|^2 - \frac{\alpha}{2}\|\beta\|^2 + \text{const.}$$

Maximizing the above log posterior then equates to minimizing the quadratic regularized error (10) with $\eta = \alpha\sigma^2$. We define the MAP estimate, $\hat{\beta}_{\text{MAP}}$, as

$$\begin{aligned}\hat{\beta}_{\text{MAP}} &= \arg\max_{\beta \in \mathbb{R}^{N-1}} \ln p(\beta \mid D) \\ &= \arg\min_{\beta \in \mathbb{R}^{N-1}} \frac{1}{2}\left\|Y_1^- - (\hat{\boldsymbol{M}}^-)^T\beta\right\|^2 + \frac{\alpha\sigma^2}{2}\|\beta\|^2 \\ &= \left(\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^T + \alpha\sigma^2\boldsymbol{I}\right)^{-1}\hat{\boldsymbol{M}}^-Y_1^-. \end{aligned} \tag{40}$$

With the MAP estimate at hand, we then make predictions of the counterfactual as

$$\hat{M}_1 = \hat{\boldsymbol{M}}^T\hat{\beta}_{\text{MAP}}. \tag{41}$$

Therefore, we have seen that the MAP estimation is equivalent to ridge regression since the introduction of an appropriate prior naturally induces the additional complexity penalty term.

4.4.2 Fully Bayesian treatment.

Although we have treated β^* as a random variable attached with a prior distribution, we can venture beyond point estimates to be fully Bayesian. In particular, we will make use of the posterior distribution over β^* to marginalize over all possible values of β^* in evaluating the predictive distribution over Y_1^-. We will decompose the regression problem of predicting the counterfactual into two separate stages: the *inference* stage in which we use the pre-intervention data to learn the predictive distribution (defined shortly), and the subsequent *decision* stage in which we use the predictive distribution to make estimates. By separating the inference and decision stages, we can readily develop new estimators for different loss functions without having to relearn the predictive distribution, providing practitioners tremendous flexibility with respect to decision making.

Let us begin with a study of the inference stage. We evaluate the predictive distribution over Y_{1t}, which is defined as

$$\begin{aligned}p(Y_{1t} \mid \hat{M}_{\cdot t}, D) &= \int p(Y_{1t} \mid \hat{M}_{\cdot t}, \beta)\; p(\beta \mid D)\, d\beta \\ &= \mathcal{N}(\hat{M}_{\cdot t}^T\,\beta_D, \sigma_D^2), \end{aligned} \tag{42}$$

where

$$\sigma_D^2 = \sigma^2 + \hat{M}_{\cdot,t}^T \mathbf{\Sigma}_D \hat{M}_{\cdot,t}. \tag{43}$$

Note that $p(\beta \mid D)$ is the posterior distribution over the synthetic control parameter and is governed by (37) and (39). With access to the predictive distribution, we move on towards the decision stage, which consists of determining a particular estimate $\hat{M}_{1t}$ given a new observation vector $X_{\cdot t}$ (used to determine $\hat{M}_{\cdot t}$). Consider an arbitrary loss function $L(Y_{1t}, g(\hat{M}_{\cdot t}))$ for some function g. The expected loss is then given by

$$\begin{aligned}\mathbb{E}[L] &= \int \int L(Y_{1t}, g(\hat{M}_{\cdot t})) \cdot p(Y_{1t}, \hat{M}_{\cdot t})\, dY_{1t}\, d\hat{M}_{\cdot t} \\ &= \int \left(\int L(Y_{1t}, g(\hat{M}_{\cdot t})) \cdot p(Y_{1t} \mid \hat{M}_{\cdot t})\, dY_{1t} \right) p(\hat{M}_{\cdot t})\, d\hat{M}_{\cdot t},\end{aligned} \tag{44}$$

and we choose our estimator $\hat{g}(\cdot)$ as the function that minimizes the average cost, i.e.,

$$\hat{g}(\cdot) = \arg\min_{g(\cdot)} \mathbb{E}[L(Y_{1t}, g(\hat{M}_{\cdot t}))]. \tag{45}$$

Since $p(\hat{M}_{\cdot t}) \geq 0$, we can minimize (44) by selecting $\hat{g}(\hat{M}_{\cdot t})$ to minimize the term within the parenthesis for each individual value of Y_{1t}, i.e.,

$$\begin{aligned}\hat{M}_{1t} &= \hat{g}(\hat{M}_{\cdot t}) \\ &= \arg\min_{g(\cdot)} \int L(Y_{1t}, g(\hat{M}_{\cdot t})) \cdot p(Y_{1t} \mid \hat{M}_{\cdot t})\, dY_{1t}.\end{aligned} \tag{46}$$

As suggested by (46), the optimal estimate $\hat{M}_{1t}$ for a particular loss function depends on the model only through the predictive distribution $p(Y_{1t} \mid \hat{M}_{\cdot t}, D)$. Therefore, the predictive distribution summarizes all of the necessary information to construct the desired Bayesian estimator for any given loss function L.

4.4.3 Bayesian least-squares estimate.

We analyze the case for the squared loss function (MSE), a common cost criterion for regression problems. In this case, we write the expected loss as

$$\mathbb{E}[L] = \int \left(\int (Y_{1t} - g(\hat{M}_{\cdot t}))^2 \cdot p(Y_{1t} \mid \hat{M}_{\cdot t})\, dY_{1t} \right) p(\hat{M}_{\cdot t})\, d\hat{M}_{\cdot t}.$$

Under the MSE cost criterion, the optimal estimate is the mean of the predictive distribution, also known as the Bayes' least-squares (BLS) estimate:

$$\begin{aligned}\hat{M}_{1t} &= \mathbb{E}[Y_{1t} \mid \hat{M}_{\cdot t}, D] \\ &= \int Y_{1t}\ p(Y_{1t} \mid \hat{M}_{\cdot t}, D) dY_{1t} \\ &= \hat{M}_t^T\, \beta_D.\end{aligned} \tag{47}$$

Remark 8 *Since the noise variance σ^2 is usually unknown in practice, we can introduce another conjugate prior distribution $p(\beta, 1/\sigma^2)$ given by the Gaussian-gamma distribution. This prior yields a Student's t-distribution for the predictive probability distribution. Alternatively, one can estimate σ^2 via* (12).

5. Experiments

We begin by exploring two real-world case studies discussed in Abadie et al. (2010, 2011); Abadie and Gardeazabal (2003) that demonstrate the ability of the original synthetic control's algorithm to produce a reliable counterfactual reality. We use the same case-studies to showcase the "robustness" property of our proposed algorithm. Specifically, we demonstrate that our algorithm reproduces similar results even in presence of missing data, and without knowledge of the extra covariates utilized by prior works. We find that our approach, surprisingly, also discovers a few subtle effects that seem to have been overlooked in prior studies. In the following empirical studies, we will employ three different learning procedures as described in the robust synthetic control algorithm: (1) linear regression ($\eta = 0$), (2) ridge regression ($\eta > 0$), and (3) LASSO ($\zeta > 0$).

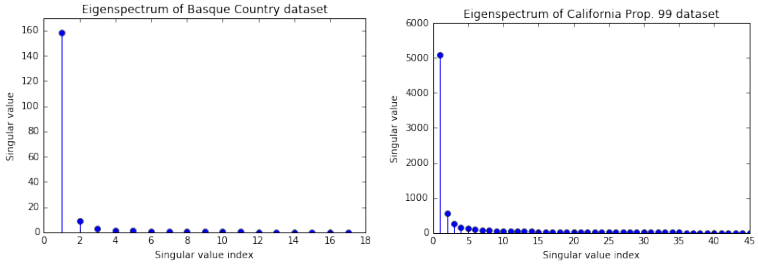
As described in Abadie et al. (2010, 2011, 2014), the synthetic control method allows a practitioner to evaluate the reliability of his or her case study results by running placebo tests. One such placebo test is to apply the synthetic control method to a donor unit. Since the control units within the donor pool are assumed to be unaffected by the intervention of interest (or at least much less affected in comparison), one would expect that the estimated effects of intervention for the placebo unit should be less drastic and divergent compared to that of the treated unit. Ideally, the counterfactuals for the placebo units would show negligible effects of intervention. Similarly, one can also perform exact inferential techniques that are similar to permutation tests. This can be done by applying the synthetic control method to every control unit within the donor pool and analyzing the gaps for every simulation, and thus providing a distribution of estimated gaps. In that spirit, we present the resulting placebo tests (for only the case of linear regression) for the Basque Country and California Prop. 99 case studies below to assess the significance of our estimates.

We will also analyze both case studies under a Bayesian setting. From our results, we see that our predictive uncertainty, captured by the standard deviation of the predictive distribution, is influenced by the number of singular values used in the de-noising process. Therefore, we have plotted the eigenspectrum of the two case study datasets below. Clearly, the bulk of the signal contained within the datasets is encoded into the top few singular values – in particular, the top two singular values. Given that the validation errors computed via forward chaining are nearly identical for low-rank settings (with the exception of a rank-1 approximation), we shall use a rank-2 approximation of the data matrix. In order to exhibit the role of thresholding in the interplay between bias and variance, we also plot the cases where we use threshold values that are too high (bias) or too low (variance). For each figure, the dotted blue line will represent our posterior predictive means while the shaded light blue region spans one standard deviation on both sides of the mean. As we shall see, our predictive uncertainty is smallest in the neighborhood around the pre-intervention period. However, the level of uncertainty increases as we deviate from the the intervention point, which appeals to our intuition.

In order to choose an appropriate choice of the prior parameter α, we first use forward-chaining for the ridge regression setting to find the optimal regularization hyperparameter η. By observing the expressions of (19) and (40), we see that $\eta = \alpha\sigma^2$ since ridge regression is closely related to MAP estimation for a zero-mean, isotropic Gaussian prior. Consequently, we choose $\alpha = \eta/\hat{\sigma}^2$ where η is the value obtained via forward chaining.

5.1 Basque Country

The goal of this case-study is to investigate the effects of terrorism on the economy of Basque Country using the neighboring Spanish regions as the control group. In 1968, the first Basque Country victim of terrorism was claimed; however, it was not until the mid-1970s did the terrorist activity become more rampant Abadie and Gardeazabal (2003). To study the economic ramifications of terrorism on Basque Country, we only use as data the per-capita GDP (outcome variable) of 17 Spanish regions from 1955-1997. We note that in Abadie and Gardeazabal (2003), 13 additional predictor variables



(a) Eigenspectrum of Basque data. **(b)** Eigenspectrum of California data.

Figure 1: Eigenspectrum

for each region were used including demographic information pertaining to one's educational status, and average shares for six industrial sectors.

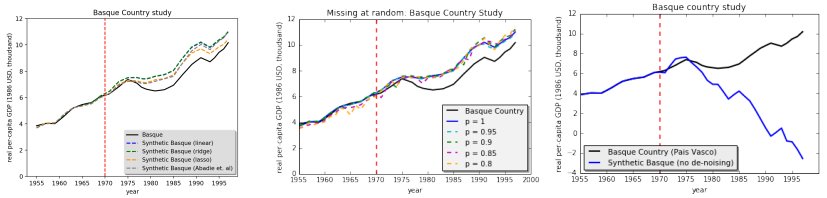
Results. Figure 2a shows that our method (for all three estimators) produces a very similar qualitative synthetic control to the original method even though we do not utilize additional predictor variables. Specifically, the synthetic control resembles the observed GDP in the pre-treatment period between 1955-1970. However, due to the large-scale terrorist activity in the mid-70s, there is a noticeable economic divergence between the synthetic and observed trajectories beginning around 1975. This deviation suggests that terrorist activity negatively impacted the economic growth of Basque Country.

One subtle difference between our synthetic control – for the case of linear and ridge regression – and that of Abadie and Gardeazabal (2003) is between 1970-75: our approach suggests that there was a small, but noticeable economic impact starting just prior to 1970, potentially due to first terrorist attack in 1968. Notice, however, that the original synthetic control of Abadie and Gardeazabal (2003) diverges only after 1975. Our LASSO estimator's trajectory also agrees with that of the original synthetic control method's, which is intuitive since both estimators seek sparse solutions.

To study the robustness of our approach with respect to missing entries, we discard each data point uniformly at random with probability $1 - p$. The resulting control for different values of p is presented in Figure 2b suggesting the robustness of our (linear) algorithm. Finally, we produce Figure 2c by applying our algorithm without the de-noising step. As evident from the Figure, the resulting predictions suffer drastically, reinforcing the value of de-noising. Intuitively, using an appropriate threshold μ equates to selecting the correct model complexity, which helps safeguard the algorithm from potentially overfitting to the training data.

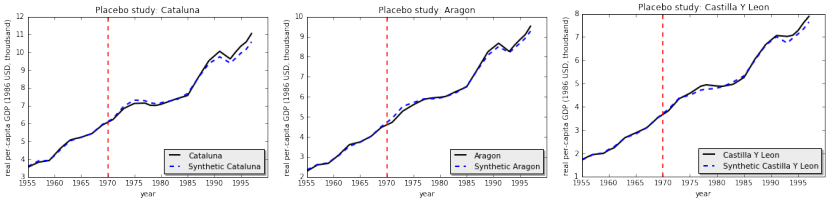
Placebo tests. We begin by applying our robust algorithm to the Spanish region of Cataluna, a control unit that is not only similar to Basque Country, but also exposed to a much lower level of terrorism Abadie et al. (2011). Observing both the synthetic and observed economic evolutions of Cataluna in Figure 3a, we see that there is no identifiable treatment effect, especially compared to the divergence between the synthetic and observed Basque trajectories. We provide the results for the regions of Aragon and Castilla Y Leon in Figures 3b and 3c.

Finally, similar to Abadie et al. (2011), we plot the differences between our estimates and the observations for Basque Country and all other regionals, individually, as placebos. Note that Abadie et al. (2011) excluded five regions that had a poor pre-intervention fit but we keep all regions. Figure 4a shows the resulting plot for all regions with the solid black line being Basque Country. This



(a) Comparison of methods. **(b)** Missing data. **(c)** Impact of de-noising.

Figure 2: Trends in per-capita GDP between Basque Country vs. synthetic Basque Country.

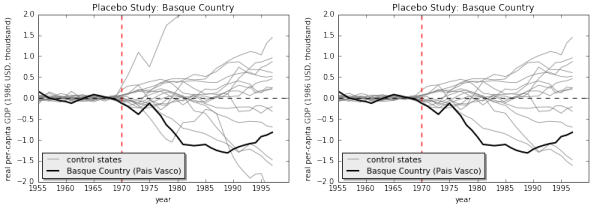


(a) Cataluna. **(b)** Aragon. **(c)** Castilla Y Leon.

Figure 3: Trends in per-capita GDP for placebo regions.

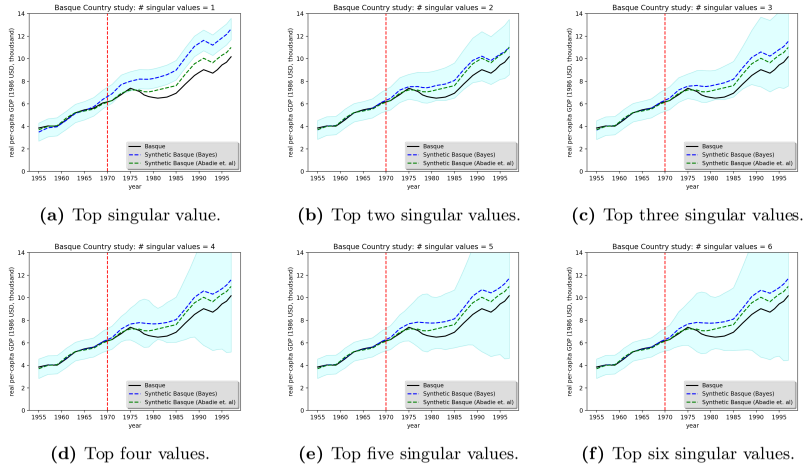
plot helps visualize the extreme post-intervention divergence between the predicted means and the observed values for Basque. Up until about 1990, the divergence for Basque Country is the most extreme compared to all other regions (placebo studies) lending credence to the belief that the effects of terrorism on Basque Country were indeed significant. Refer to Figure 4b for the same test but with Madrid and Balearic Islands excluded, as per Abadie et al. (2011). The conclusions drawn should remain the same, pointing to the robustness of our approach.

Bayesian approach. We plot the resulting Bayesian estimates in the figures below under varying thresholding conditions. It is interesting to note that our uncertainty grows dramatically once we include more than two singular values in the thresholding process. This confirms what our theoretical



(a) Includes all control regions. **(b)** Excludes 2 regions.

Figure 4: Per-capita GDP gaps for Basque Country and control regions.



(a) Top singular value. **(b)** Top two singular values. **(c)** Top three singular values.

(d) Top four values. **(e)** Top five singular values. **(f)** Top six singular values.

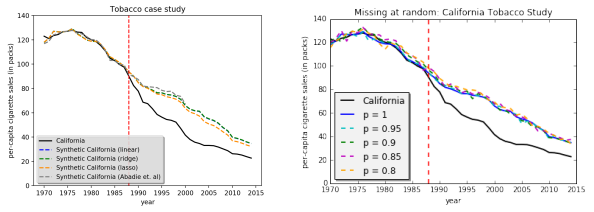
Figure 5: Trends in per-capita GDP between Basque Country vs. synthetic Basque Country.

results indicated earlier: choosing a smaller threshold, μ, would lead to a greater number of singular values retained which results in higher variance. On the other hand, notice that just selecting 1 singular value results in an apparently biased estimate which is overestimating the synthetic control. It appears that selecting the top two singular values balance the bias-variance tradeoff the best and is also agrees with our earlier finding that the data matrix appears to be of rank 2 or 3. Note that in this setting, we would find it hard to reject the null-hypothesis because the observations for the treated unit lie within the uncertainty band of the estimated synthetic control.

5.2 California Anti-tobacco Legislation

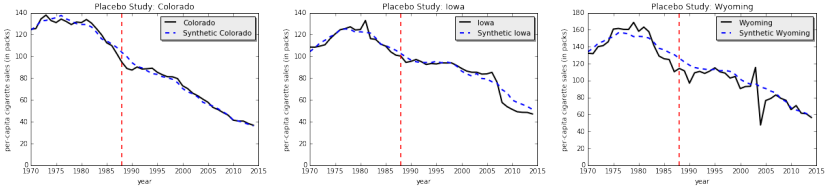
We study the impact of California's anti-tobacco legislation, Proposition 99, on the per-capita cigarette consumption of California. In 1988, California introduced the first modern-time large-scale anti-tobacco legislation in the United States Abadie et al. (2010). To analyze the effect of California's anti-tobacco legislation, we use the annual per-capita cigarette consumption at the state-level for all 50 states in the United States, as well as the District of Columbia, from 1970-2015. Similar to the previous case study, Abadie and Gardeazabal (2003) uses 6 additional observable covariates per state, e.g. retail price, beer consumption per capita, and percentage of individuals between ages of 15-24, to predict their synthetic California. Furthermore, Abadie and Gardeazabal (2003) discarded 12 states from the donor pool since some of these states also adopted anti-tobacco legislation programs or raised their state cigarette taxes, and discarded data after the year 2000 since many of the control units had implemented anti-tobacco measures by this point in time.

Results. As shown in Figure 6a, in the pre-intervention period of 1970-88, our control matches the observed trajectory. Post 1988, however, there is a significant divergence suggesting that the passage of Prop. 99 helped reduce cigarette consumption. Similar to the Basque case-study, our estimated effect is similar to that of Abadie and Gardeazabal (2003). As seen in Figure 6b, our algorithm is



(a) Comparison of methods. **(b)** Missing data.

Figure 6: Trends in per-capita cigarette sales between California vs. synthetic California.



(a) Colorado. **(b)** Iowa. **(c)** Wyoming.

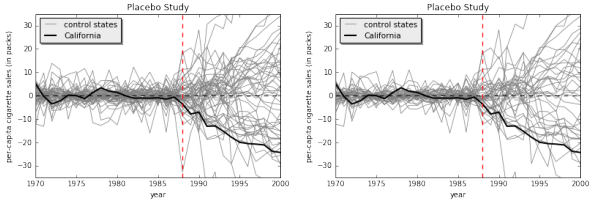
Figure 7: Placebo Study: trends in per-capita cigarette sales for Colorado, Iowa, and Wyoming.

again robust to randomly missing data.

Placebo tests. We now proceed to apply the same placebo tests to the California Prop 99 dataset. Figures 7a, 7b, and 7c are three examples of the applied placebo tests on the remaining states (including District of Columbia) within the United States.

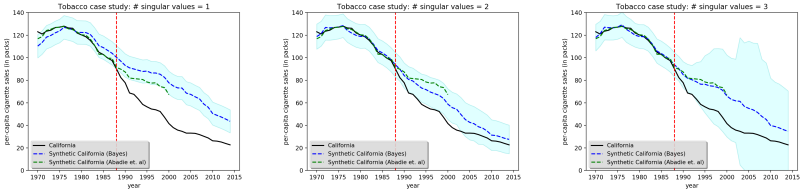
Finally, similar to Abadie et al. (2010), we plot the differences between our estimates and the actual observations for California and all other states, individually, as placebos. Note that Abadie et al. (2010) excluded twelve states but we keep all states. Figure 8a shows the resulting plot for all states with the solid black line being California. This plot helps visualize the extreme post-intervention divergence between the predicted means and the observed values for California. Up until about 1995, the divergence for California was clearly the most significant and consistent outlier compared to all other regions (placebo studies) lending credence to the belief that the effects of Proposition 99 were indeed significant. Refer to Figure 8b for the same test but with the same twelve states excludes as in Abadie et al. (2010). Just like the Basque Country case study, the exclusion of states should not affect the conclusions drawn.

Bayesian approach. Similar to the Basque Country case study, our predictive uncertainty increases as the number of singular values used in the learning process exceeds two. In order to gain some new insight, however, we will focus our attention to the resulting figure associated with three singular values, which is particularly interesting. Specifically, we observe that our predictive means closely match the counterfactual trajectory produced by the classical synthetic control method in both the pre- and post-intervention periods (up to year 2000), and yet our uncertainty for this estimate is significantly greater than our uncertainty associated with the estimate produced using two singular values. As a result, it may be possible that the classical synthetic control method overestimated the



(a) Includes all donors. **(b)** Excludes 12 states.

Figure 8: Per-capita cigarette sales gaps in California and control regions.



(a) Top singular value. **(b)** Top two singular values. **(c)** Top three singular values.

Figure 9: Trends in per-capita cigarette sales between California vs. synthetic California.

effect of Prop. 99, even though the legislation did probably discourage the consumption of cigarettes – a conclusion reached by both our robust approach and the classical approach.

Remark 9 *We note that in Abadie et al. (2014), the authors ran two robustness tests to examine the sensitivity of their results (produced via the original synthetic control method) to alterations in the estimated convex weights – recall that the original synthetic control estimator produces a β^* that lies within the simplex. In particular, the authors first iteratively reproduced a new synthetic West Germany by removing one of the countries that received a positive weight in each iteration, demonstrating that their synthetic model is fairly robust to the exclusion of any particular country with positive weight. Furthermore, Abadie et al. (2014) examined the trade-off between the original method's ability to produce a good estimate and the sparsity of the given donor pool. In order to examine this tension, the authors restricted their synthetic West Germany to be a convex combination of only four, three, two, and a single control country, respectively, and found that, relative to the baseline synthetic West Germany (composed of five countries), the degradation in their goodness of fit was moderate.*

5.3 Synthetic simulations

We conduct synthetic simulations to establish the various properties of the estimates in both the pre- and post-intervention stages.

Experimental setup. For each unit $i \in [N]$, we assign latent feature θ_i by drawing a number uniformly at random in $[0, 1]$. For each time $t \in [T]$, we assign latent variable $\rho_t = t$. The mean value

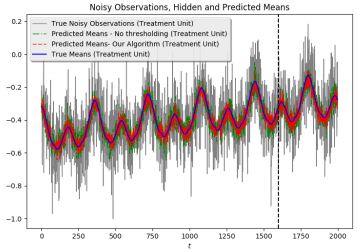


Figure 10: Treatment unit: noisy observations (gray) and true means (blue) and the estimates from our algorithm (red) and one where no singular value thresholding is performed (green). The plots show all entries normalized to lie in range $[-1, 1]$. Notice that the estimates in red generated by our model are much better at estimating the true underlying mean (blue) when compared to an algorithm which performs no singular value thresholding.

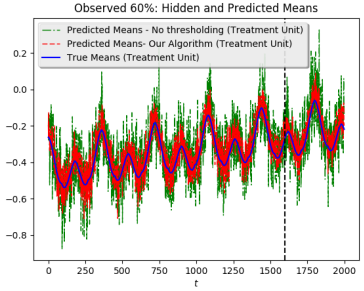


Figure 11: Same dataset as shown in Figure 10 but with 40% data missing at random. Treatment unit: not showing the noisy observations for clarity; plotting true means (blue) and the estimates from our algorithm (red) and one where no singular value thresholding is performed (green). The plots show all entries normalized to lie in range $[-1, 1]$.

$m_{it} = f(\theta_i, \rho_t)$. In the experiments described in this section, we use the following:

$$f(\theta_i, \rho_t) = \theta_i + (0.3 \cdot \theta_i \cdot \rho_t/T) * (\exp^{\rho_t/T}) + \cos(f_1\pi/180) + 0.5\sin(f_2\pi/180) + 1.5\cos(f_3\pi/180) - 0.5\sin(f_4 * \pi/180),$$

where f_1, f_2, f_3, f_4 define the periodicities: $f_1 = \rho_t \bmod (360)$, $f_2 = \rho_t \bmod (180)$, $f_3 = 2 \cdot \rho_t \bmod (360)$, $f_4 = 2.0 \cdot \rho_t \bmod (180)$. The observed value X_{it} is produced by adding i.i.d. Gaussian noise to mean with zero mean and variance σ^2. For this set of experiments, we use $N = 100, T = 2000$, while assuming the treatment was performed at $t = 1600$.

Training error approximates generalization error. For the first experimental study, we analyze the relationship between the pre-intervention MSE (training error) and the post-intervention MSE (generalization error). As seen in Table 1, the post-intervention MSE closely matches that of the

Table 1: Training vs. generalization error

Noise	Training error	Generalization error
3.1	0.48	0.53
2.5	0.31	0.34
1.9	0.19	0.22
1.3	0.09	0.1
0.7	0.027	0.03
0.4	0.008	0.009
0.1	0.0005	0.0006

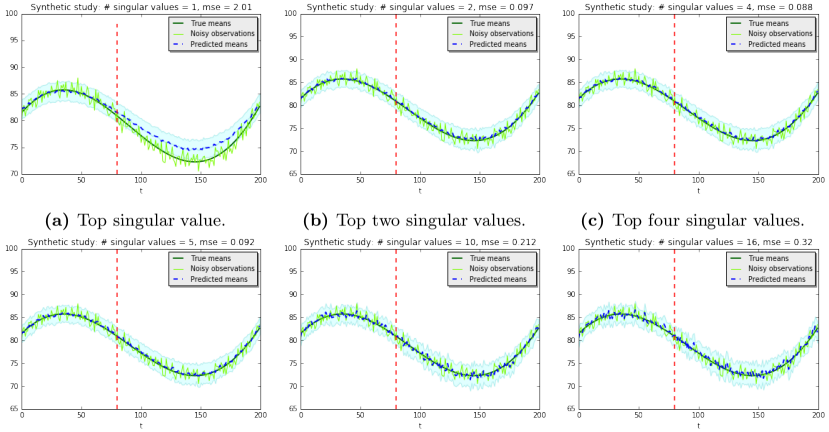
Table 2: Impact of thresholding

Noise	De-noising error	No De-noising error
3.1	0.122	0.365
2.5	0.079	0.238
1.9	0.046	0.138
1.6	0.032	0.098
1	0.013	0.038
0.7	0.006	0.018
0.4	0.002	0.005

pre-intervention MSE for varying noise levels, σ^2. Thus suggesting efficacy of our algorithm. Figures 10 and 11 plot the estimates of algorithm with no missing data (Figure 10) and with 40% randomly missing data (Figure 11) on the same underlying dataset. All entries in the plots were normalized to lie within $[-1, 1]$. These plots confirm the robustness of our algorithm. Our algorithm outperforms the algorithm with no singular value thresholding under all proportions of missing data. The estimates from the algorithm which performs no singular value thresholding (green) degrade significantly with missing data while our algorithm remains robust.

Benefits of de-noising. We now analyze the benefit of de-noising the data matrix, which is the main contribution of this work compared to the prior work. Specifically, we study the generalization error of method using de-noising via thresholding and without thresholding as in prior work. The results summarized in Table 2 show that for range of parameters the generalization error with de-noising is consistency better than that without de-noising.

Bayesian approach. From the synthetic simulations (figures below), we see that the number of singular values included in the thresholding process plays a crucial role in the model's prediction capabilities. If not enough singular values are used, then there is a significant loss of information (high bias) resulting in a higher MSE. On the other hand, if we include too many singular values, then the model begins to overfit to the dataset by misinterpreting noise for signal (high variance). As emphasized before, the goal is to find the simplest model that both fits the data and is also plausible, which is achieved when four singular values are employed.



(a) Top singular value. **(b)** Top two singular values. **(c)** Top four singular values.

(d) Top five singular values. **(e)** Top 10 singular values. **(f)** Top 16 singular values.

Figure 12: Bayesian Synthetic Control simulation.

6. Conclusion

The classical synthetic control method is recognized as a powerful and effective technique for causal inference for comparative case studies. In this work, we motivate a robust synthetic control algorithm, which attempts to improve on the classical method in the following regimes: (a) randomly missing data and (b) large levels of noise. We also demonstrate that the algorithm performs well even in the absence of covariate or expert information, but do *not* propose ignoring information which may eliminate "bad" donors. Our data-driven algorithm, and its Bayesian counterpart, uses singular value thresholding to impute missing data and "de-noise" the observations. Once "de-noised", we use regularized linear regression to determine the synthetic control. Motivating our algorithm is a modeling framework, specifically the Latent Variable Model, which is a generalization of the various factor models used in related work. We establish finite-sample bounds on the MSE between the estimated "synthetic" control and the latent *true* means of the treated unit of interest. In situations with plentiful data, we show that a simple data aggregation method can lead to an asymptotically consistent estimator. Experiments on synthetically generated data (where the *truth* is known) and on real-world case-studies allow us to demonstrate the promise of our approach, which is an improvement over the classical method.

Acknowledgments

We would like to thank Alberto Abadie for careful reading and comments that have helped in improving the manuscript.

References

A. Abadie and J. Gardeazabal. The economic costs of conflict: A case study of the basque country. *American Economic Review*, 2003.

A. Abadie, A. Diamond, and J. Hainmueller. Synthetic control methods for comparative case studies: Estimating the effect of californiaâs tobacco control program. *Journal of the American Statistical Association*, 2010.

A. Abadie, A. Diamond, and J. Hainmueller. Synth: An r package for synthetic control methods in comparative case studies. *Journal of Statistical Software*, 2011.

A. Abadie, A. Diamond, and J. Hainmueller. Comparative politics and the synthetic control method. *American Journal of Political Science*, 2014.

B. Adhikari and J. Alm. Evaluating the economic effects of flat tax reforms using synthetic control methods. *Southern Economic Association*, 2016.

D. J. Aldous. Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis*, 11(4):581–598, 1981.

S. Athey and G. Imbens. The state of applied econometrics - causality and policy evaluation. *The Journal of Economic Perspectives*, 31(2):3–32, 2016.

Susan Athey, Mohsen Bayati, Nikolay Doudchenko, and Guido Imbens. Matrix completion methods for causal panel data models. 2017.

H. Aytug, M. Kutuk, A. Oduncu, and S. Togan. Twenty years of the eu-turkey customs union: A synthetic control method analysis. *Journal of Common Market Studies*, 2016.

BallotPedia. California proposition 63, background checks for ammunition purchases and large-capacity ammunition magazine ban (2016). *www.ballotpedia.org*, 2016. URL `https://ballotpedia.org/California_Proposition_63,_Background_Checks_for_Ammunition_Purchases_and_Large-Capacity_Ammunition_Magazine_Ban_(2016)`.

A. Billmeier and T. Nannicini. Assessing economic liberalization episodes: A synthetic control approach. *The Review of Economics and Statistics*, 2013.

C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

C. Borgs, J. Chayes, C. E. Lee, and D. Shah. Thy friend is my friend: Iterative collaborative filtering for sparse matrix estimation. *Advances in Neural Information Processing Systems*, 2017.

K. Brodersen, F. Gallusser, J. Koehler, N. Remy, and S. Scott. Inferring causal impact using bayesian structural time-series models. *Annals of Applied Statistics*, 2015.

Emmanuel J. Candès and Benjamin Recht. Exact matrix completion via convex optimization. *CoRR*, abs/0805.4471, 2008. URL `http://arxiv.org/abs/0805.4471`.

Sourav Chatterjee. Matrix estimation by universal singular value thresholding. *Annals of Statistics*, 43:177–214, 2015.

N. Doudchenko and G. Imbens. Balancing, regression, difference-in-differences and synthetic control methods: A synthesis. *NBER Working Paper No. 22791*, 2016.

R. Farebrother. Further results on the mean square error of ridge regression. *Journal of the Royal Statistical Society. Series B (Methodological)*, 38(3):248–250, 1976.

B. Ferman and C. Pinto. Revisiting the synthetic control estimator. 2016.

B. Ferman, C. Pinto, and V. Possebom. Cherry picking with synthetic controls. 2016.

J. Gardeazabal and A. Vega-Bayo. An empirical comparison between the synthetic control method and hsiao et al.'s panel data approach to program evaluation. *Journal of Applied Econometrics*, 2016.

D. N. Hoover. Relations on probability spaces and arrays of random variables. 1979.

D. N. Hoover. Row-columns exchangeability and a generalized model for exchangeability. *Exchangeability in probability and statistics*, (281-291), 1981.

C. Hsiao. *Analysis of panel data*. Cambridge University Press, 2014.

C. Hsiao, H. Ching, and S. Wan. A panel data approach for program evaluation: Measuring the benefits of political and economic integration of hong kong with mainland china. *Journal of Applied Econometrics*, 2011.

Jha, Sunil K., and R. D. S. Yadava. Denoising by singular value decomposition and its application to electronic nose data processing. *IEEE Sensors Journal*, 11:35–44, June 2010.

N. Kreif, R. Grieve, D. Hangartner, A. Turner, S. Nikolova, and M. Sutton. Examination of the synthetic control method for evaluating health policies with multiple treated units. *Health Economics*, 2015.

C. E. Lee, Y. Li, D. Shah, and D. Song. Blind regression via nearest neighbors under latent variable models. *Advances in Neural Information Processing Systems*, 2016.

T. Kathleen Li and R. David Bell. Estimation of average treatment effects with panel data: Asymptotic theory and implementation. *Journal of Econometrics*, 197:65–75, 2017.

Rahul Mazumder, Trevor Hastie, and Robert Tibshirani. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug):2287–2322, 2010.

Patrick McGreevy. California voters approve gun control measure proposition 63. *Los Angeles Times*, Nov. 2016. URL `http://www.latimes.com/nation/politics/trailguide/la-na-election-day-2016-proposition-63-gun-control-1478280771-htmlstory.html`.

J. Saunders, R. Lundberg, A. Braga, G. Ridgeway, and J. Miles. A synthetic control approach to evaluating place-based crime interventions. *Journal of Quantitative Criminology*, 2014.

M. Udell and A. Townsend. Nice latent variable models have log-rank. 2017.

Shui-Ki Wan, Yimeng Xie, and Cheng Hsiao. Panel data approach vs synthetic control method. *Economics Letters*, 164:121–123, 2018.

P. J. Wolfe and S. C. Olhede. Nonparametric graphon estimation. *https://arxiv.org/abs/1309.5936*.

Y. Xu. Generalized synthetic control method: Causal inference with interactive fixed effects models. *Political Analysis*, 25(1), 2017.

J. Yang, Q. Han, and E. M. Airoldi. Nonparametric estimation and testing of exchangeable graph models. *Journal of Machine Learning Research, Conference and Workshop Proceedings*, 33: 1060–1067.

Appendix A. Useful Theorems

We present useful theorems that we will frequently employ in our proofs.

Theorem 10 Perturbation of singular values.
Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two $m \times n$ matrices. Let $k = \min\{m, n\}$. Let $\lambda_1, \ldots, \lambda_k$ be the singular values of $\boldsymbol{A}$ in decreasing order and repeated by multiplicities, and let $\tau_1, \ldots, \tau_k$ be the singular values of $\boldsymbol{B}$ in decreasing order and repeated by multiplicities. Let $\delta_1, \ldots, \delta_k$ be the singular values of $\boldsymbol{A} - \boldsymbol{B}$, in any order but still repeated by multiplicities. Then,

$$\max_{1 \le i \le k} |\lambda_i - \tau_i| \le \max_{1 \le i \le k} |\delta_i|.$$

References for the proof of the above result can be found in Chatterjee (2015), for example.

Theorem 11 Poincaré separation Theorem.
Let $\boldsymbol{A}$ be a symmetric $n \times n$ matrix. Let $\boldsymbol{B}$ be the $m \times m$ matrix with $m \le n$, where $\boldsymbol{B} = \boldsymbol{P}^T \boldsymbol{A} \boldsymbol{P}$ for some orthogonal projection matrix $\boldsymbol{P}$. If the eigenvalues of $\boldsymbol{A}$ are $\sigma_1 \le \ldots \le \sigma_n$, and those of $\boldsymbol{B}$ are $\tau_1 \le \ldots \le \tau_m$, then for all $j < m + 1$,

$$\sigma_j \le \tau_j \le \sigma_{n-m+j}.$$

Remark 12 *In the case where $\boldsymbol{B}$ is the principal submatrix of $\boldsymbol{A}$ with dimensions $(n-1) \times (n-1)$, the above Theorem is also known as Cauchy's interlacing law.*

Theorem 13 Bernstein's Inequality.
Suppose that $X_1, \ldots, X_n$ are independent random variables with zero mean, and M is a constant such that $|X_i| \le M$ with probability one for each i. Let $S := \sum_{i=1}^n X_i$ and $v := Var(S)$. Then for any $t \ge 0$,

$$\mathbb{P}(|S| \ge t) \le 2 \exp\left(-\frac{3t^2}{6v + 2Mt}\right).$$

Theorem 14 Hoeffding's Inequality.
Suppose that $X_1, \ldots, X_n$ are independent random variables that are strictly bounded by the intervals $[a_i, b_i]$. Let $S := \sum_{i=1}^n X_i$. Then for any $t > 0$,

$$\mathbb{P}(|S - \mathbb{E}[S]| \ge t) \le 2 \exp\left(-\frac{2t^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

Theorem 15 Theorem 3.4 of Chatterjee (2015).
Take any two numbers m and n such that $1 \le m \le n$. Suppose that $\boldsymbol{A} = [A_{ij}]_{1 \le i \le m, 1 \le j \le n}$ is a matrix whose entries are independent random variables that satisfy, for some $\delta^2 \in [0, 1]$,

$$\mathbb{E}[A_{ij}] = 0, \quad \mathbb{E}[A_{ij}^2] \le \delta^2, \quad \textit{and} \quad |A_{ij}| \le 1 \quad \textit{a.s.}$$

Suppose that $\delta^2 \ge n^{-1+\zeta}$ for some $\zeta > 0$. Then, for any $\omega \in (0, 1)$,

$$\mathbb{P}(\|\boldsymbol{A}\| \ge (2 + \omega)\delta\sqrt{n}) \le C(\zeta) e^{-c\delta^2 n},$$

where $C(\zeta)$ depends only on ω and ζ, and c depends only on ω. The same result is true when $m = n$ and A is symmetric or skew-symmetric, with independent entries on and above the diagonal, all other assumptions remaining the same. Lastly, all results remain true if the assumption $\delta^2 \ge n^{-1+\zeta}$ is changed to $\delta^2 \ge n^{-1}(\log n)^{6+\zeta}$.

Remark 16 *The proof of Theorem 15 can be found in Chatterjee (2015) under Theorem 3.4.*

Appendix B. Useful Lemmas

We begin by proving (and providing) a series of useful lemmas that we will frequently use to derive our desired results.

Lemma 17 *Suppose $\boldsymbol{C}$ is an $m \times n$ matrix composed of an $m \times p$ submatrix $\boldsymbol{A}$ and an $m \times (n-p)$ submatrix $\boldsymbol{B}$, i.e., $\boldsymbol{C} = \left[\begin{array}{c|c} \boldsymbol{A} & \boldsymbol{B} \end{array}\right]$. Then, the spectral (operator) norms of $\boldsymbol{A}$ and $\boldsymbol{B}$ are bounded above by the spectral norm of $\boldsymbol{C}$,*

$$\max\{\|\boldsymbol{A}\|, \|\boldsymbol{B}\|\} \le \|\boldsymbol{C}\|.$$

Proof Without loss of generality, we prove the case for $\|\boldsymbol{A}\| \le \|\boldsymbol{C}\|$, since the same argument applies for $\|\boldsymbol{B}\|$. By definition,

$$\boldsymbol{C}^T \boldsymbol{C} = \begin{bmatrix} \boldsymbol{A}^T \boldsymbol{A} & \boldsymbol{A}^T \boldsymbol{B} \\ \boldsymbol{B}^T \boldsymbol{A} & \boldsymbol{B}^T \boldsymbol{B} \end{bmatrix}.$$

Let $\sigma_1, \ldots, \sigma_n$ be the eigenvalues of $\boldsymbol{C}^T \boldsymbol{C}$ in increasing order and repeated by multiplicities. Let $\tau_1, \ldots, \tau_p$ be the eigenvalues of $\boldsymbol{A}^T \boldsymbol{A}$ in increasing order and repeated by multiplicities. By the Poincaré separation Theorem 11, we have for all $j < p + 1$,

$$\sigma_j \le \tau_j \le \sigma_{n-p+j}.$$

Thus, $\tau_p \le \sigma_n$. Since the eigenvalues of $\boldsymbol{C}^T \boldsymbol{C}$ and $\boldsymbol{A}^T \boldsymbol{A}$ are the squared singular values of $\boldsymbol{C}$ and $\boldsymbol{A}$ respectively, we have

$$\sqrt{\tau_p} = \|\boldsymbol{A}\| \le \|\boldsymbol{C}\| = \sqrt{\sigma_n}.$$

We complete the proof by applying an identical argument for the case of $\|\boldsymbol{B}\|$. ■

Lemma 18 *Let $\boldsymbol{A}$ be any m by n matrix, and let $\boldsymbol{A}^\dagger$ be its corresponding pseudoinverse. Then, the matrices $\boldsymbol{P}_1 = \boldsymbol{A}\boldsymbol{A}^\dagger$ and $\boldsymbol{P}_2 = \boldsymbol{A}^\dagger \boldsymbol{A}$ are projection matrices.*

Proof We first prove that $\boldsymbol{P}_1$ is a projection matrix. In order to show $\boldsymbol{P}_1$ is a projection matrix, we must demonstrate that $\boldsymbol{P}_1$ satisfies two properties: namely, (1) $\boldsymbol{P}_1$ is symmetric, i.e. $\boldsymbol{P}_1^T = \boldsymbol{P}_1$, and (2) $\boldsymbol{P}_1$ is idempotent, i.e. $\boldsymbol{P}_1^2 = \boldsymbol{P}_1$.

Let $\boldsymbol{A} = \boldsymbol{Q}_1 \boldsymbol{\Sigma} \boldsymbol{Q}_2^T$ represent the SVD of $\boldsymbol{A}$, with the pseudoinverse expressed as $\boldsymbol{A}^\dagger = \boldsymbol{Q}_2 \boldsymbol{\Sigma}^+ \boldsymbol{Q}_1^T$. As a result,

$$\begin{aligned} \boldsymbol{P}_1 &= \boldsymbol{A}\boldsymbol{A}^\dagger \\ &= \boldsymbol{Q}_1 \boldsymbol{\Sigma} \boldsymbol{Q}_2^T \boldsymbol{Q}_2 \boldsymbol{\Sigma}^+ \boldsymbol{Q}_1^T \\ &= \boldsymbol{Q}_1 \boldsymbol{\Sigma} \boldsymbol{\Sigma}^+ \boldsymbol{Q}_1^T. \end{aligned}$$

Note that

$$\begin{aligned} \boldsymbol{P}_1^T &= (\boldsymbol{Q}_1 \boldsymbol{\Sigma} \boldsymbol{\Sigma}^+ \boldsymbol{Q}_1^T)^T \\ &= \boldsymbol{Q}_1 \boldsymbol{\Sigma} \boldsymbol{\Sigma}^+ \boldsymbol{Q}_1^T \\ &= \boldsymbol{P}_1, \end{aligned}$$

which proves that $\boldsymbol{P}_1$ is symmetric. Furthermore,

$$\begin{aligned}\boldsymbol{P}_1^2 &= (\boldsymbol{Q}_1\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{Q}_1^T)\cdot(\boldsymbol{Q}_1\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{Q}_1^T)\\ &= \boldsymbol{Q}_1\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{Q}_1^T\\ &= \boldsymbol{Q}_1\boldsymbol{\Sigma}\boldsymbol{\Sigma}^+\boldsymbol{Q}_1^T\\ &= \boldsymbol{P}_1,\end{aligned}$$

which proves that $\boldsymbol{P}_1$ is idempotent. The same argument can be applied for $\boldsymbol{P}_2$. ■

Lemma 19 *The eigenvalues of a projection matrix are* 1 *or* 0.

Proof Let λ be an eigenvalue of the projection matrix $\boldsymbol{P}$ for some eigenvector v. Then, by definition of eigenvalues,

$$\boldsymbol{P}v = \lambda v.$$

However, by the idempotent property of projection matrices ($\boldsymbol{P}^2 = \boldsymbol{P}$), if we multiply the above equality by $\boldsymbol{P}$ on the left, then we have

$$\begin{aligned}\boldsymbol{P}(\boldsymbol{P}v) &= \boldsymbol{P}(\lambda v)\\ &= \lambda^2 v.\end{aligned}$$

Since $v \neq 0$, the eigenvalues of $\boldsymbol{P}$ can only be members $\mathbb{R}$ whereby $\lambda^2 = \lambda$. Ergo, we must have that $\lambda \in \{0, 1\}$. ■

Lemma 20 *Let $\boldsymbol{A} = \sum_{i=1}^m \sigma_i x_i y_i^T$ be the singular value decomposition of $\boldsymbol{A}$ with $\sigma_1, \ldots, \sigma_m$ in decreasing order and with repeated multiplicities. For any choice of $\mu \geq 0$, let $S = \{i : \sigma_i \geq \mu\}$. Define*

$$\hat{\boldsymbol{B}} = \sum_{i\in S} \sigma_i x_i y_i^T.$$

Let $\tau_1, \ldots, \tau_m$ be the singular values of $\boldsymbol{B}$ in decreasing order and repeated by multiplicities, with $\tau^* = \max_{i\notin S} \tau_i$. Then

$$\left\|\hat{\boldsymbol{B}} - \boldsymbol{B}\right\| \leq \tau^* + 2\|\boldsymbol{A} - \boldsymbol{B}\|.$$

Proof By Theorem 10, we have that $\sigma_i \leq \tau_i + \|\boldsymbol{A} - \boldsymbol{B}\|$ for all i. Applying triangle inequality, we obtain

$$\begin{aligned}\left\|\hat{\boldsymbol{B}} - \boldsymbol{B}\right\| &\leq \left\|\hat{\boldsymbol{B}} - \boldsymbol{A}\right\| + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &= \max_{i\notin S} \sigma_i + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &\leq \max_{i\notin S} \Big(\tau_i + \|\boldsymbol{A} - \boldsymbol{B}\|\Big) + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &= \tau^* + 2\|\boldsymbol{A} - \boldsymbol{B}\|.\end{aligned}$$

■

Lemma 21 *Let $\boldsymbol{A} = \sum_{i=1}^m \sigma_i x_i y_i^T$ be the singular value decomposition of $\boldsymbol{A}$. Fix any $\delta > 0$ such that $\mu = (1+\delta)\|\boldsymbol{A} - \boldsymbol{B}\|$, and let $S = \{i : \sigma_i \geq \mu\}$. Define*

$$\hat{\boldsymbol{B}} = \sum_{i\in S} \sigma_i x_i y_i^T.$$

Then

$$\left\|\hat{\boldsymbol{B}} - \boldsymbol{B}\right\| \leq (2+\delta)\|\boldsymbol{A} - \boldsymbol{B}\|.$$

Proof By the definition of μ and hence the set of singular values S, we have that

$$\begin{aligned}\left\|\hat{\boldsymbol{B}} - \boldsymbol{B}\right\| &\leq \left\|\hat{\boldsymbol{B}} - \boldsymbol{A}\right\| + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &= \max_{i\notin S} \sigma_i + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &\leq (1+\delta)\|\boldsymbol{A} - \boldsymbol{B}\| + \|\boldsymbol{A} - \boldsymbol{B}\|\\ &= (2+\delta)\|\boldsymbol{A} - \boldsymbol{B}\|.\end{aligned}$$

■

Lemma 22 ***(Lemma 3.5 of Chatterjee (2015))*** *Let $\boldsymbol{A} = \sum_{i=1}^m \sigma_i x_i y_i^T$ be the singular value decomposition of $\boldsymbol{A}$. Fix any $\delta > 0$ and define $S = \{i : \sigma_i \geq (1+\delta)\|\boldsymbol{A} - \boldsymbol{B}\|\}$ such that*

$$\hat{\boldsymbol{B}} = \sum_{i\in S} \sigma_i x_i y_i^T.$$

Then

$$\left\|\hat{\boldsymbol{B}} - \boldsymbol{B}\right\|_F \leq K(\delta)(\|\boldsymbol{A} - \boldsymbol{B}\|\|\boldsymbol{B}\|_*)^{1/2},$$

where $K(\delta) = (4+2\delta)\sqrt{2/\delta} + \sqrt{2+\delta}$.

Proof The proof can be found in Chatterjee (2015). ■

Appendix C. Preliminaries.

To simplify the following exposition, we assume that $|M_{ij}| \leq 1$ and $|X_{ij}| \leq 1$. Recall that all entries of the pre-intervention treatment row are observed such that $Y_1^- = X_1^- = M_1^- + \epsilon_1^-$. On the other hand, every entry within the pre- and post-intervention periods for the donor units are observed independently of the other entries with some arbitrary probability p. Specifically, for all $2 \leq i \leq N$ and $j \in [T]$, we define $Y_{ij} = X_{ij}$ if X_{ij} is observed, and $Y_{ij} = 0$ otherwise. Consequently, observe that for all $i > 1$ and j,

$$\mathbb{E}[Y_{ij}] = pM_{ij}$$

and

$$\begin{aligned}\mathrm{Var}(Y_{ij}) &= \mathbb{E}[Y_{ij}^2] - (\mathbb{E}[Y_{ij}])^2\\ &= p\mathbb{E}[X_{ij}^2] - (pM_{ij})^2\\ &\leq p(\sigma^2 + M_{ij}^2) - (pM_{ij})^2\\ &= p\sigma^2 + pM_{ij}^2(1-p)\\ &\leq p\sigma^2 + p(1-p).\end{aligned}$$

Recall that $\hat{p}$ denotes the proportion of observed entries in $\boldsymbol{X}$ and $\hat{\sigma}^2$ represents the (unbiased) sample variance computed from the pre-intervention treatment row (12). Given the information above, we define, for any $\omega \in (0.1, 1)$, three events E_1, E_2, and E_3 as

$$\begin{aligned} E_1 &:= \{|\hat{p} - p| \leq \omega p/z\}, \\ E_2 &:= \{|\hat{\sigma}^2 - \sigma^2| \leq \omega\sigma^2/z\}, \\ E_3 &:= \{\|\boldsymbol{Y} - p\boldsymbol{M}\| \leq (2 + \omega/2)\sqrt{Tq}\}, \end{aligned}$$

where $q = \sigma^2 p + p(1-p)$; for reasons that will be made clear later, we choose $z = 60(\frac{\sigma^2+1}{\sigma^2})$. By Bernstein's Inequality, we have that

$$\mathbb{P}(E_1) \geq 1 - 2e^{-c_1(N-1)Tp},$$

for appropriately defined constant c_1. By Hoeffding's Inequality, we obtain

$$\mathbb{P}(E_2) \geq 1 - 2e^{-c_2 T\sigma^2}$$

for some positive constant c_2. Moreover, by Theorem 15,

$$\mathbb{P}(E_3) \geq 1 - Ce^{-c_3 Tq}$$

as long as $q = \sigma^2 p + p(1-p) \geq T^{-1+\zeta}$ for some $\zeta > 0$. In other words,

$$\begin{aligned} p(\sigma^2 + 1) &\geq p(\sigma^2 + (1-p)) \\ &\geq T^{-1+\zeta}. \end{aligned}$$

Consequently, assuming the event E_3 occurs, we require that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$.

Finally, as previously discussed, we will assume that both N and T grow without bound in our imputation analysis. However, in our forecasting analysis, only $T_0 \to \infty$.

Appendix D. Imputation Analysis

In this section, we prove that our key de-noising procedure produces a consistent estimator of the underlying mean matrix, thereby adroitly imputing the missing entries and filtering corrupted observations within our data matrix.

Lemma 23 *Let $\boldsymbol{M} = [M_{ij}]$ be defined as before. Suppose f is a Lipschitz function with Lipschitz constant $\mathcal{L}$ and the latent row and column feature vectors come from a compact space K of dimension d. Then for any small enough $\delta > 0$,*

$$\|\boldsymbol{M}\|_* \leq \delta(N-1)\sqrt{T} + C(K, d, \mathcal{L})\sqrt{(N-1)T\delta^{-d}},$$

where $C(K, d, L)$ is a constant that depends on K, d, and $\mathcal{L}$.

Proof The proof is a straightforward adaptation of the arguments from [Chatterjee (2015), Lemma 3.6]; however, we provide it here for completeness. By the Lipschitzness assumption, every entry in $\boldsymbol{M} = [M_{ij}] = [f(\theta_i, \rho_j)]$ is Lipschitz in both its arguments, space (i) and time (j). For any $\delta > 0$, it is not hard to see that one can find a finite covering $P_1(\delta)$ and $P_2(\delta)$ of K so that for any $\theta, \rho \in K$, there exists $\theta' \in P_1(\delta)$ and $\rho' \in P_2(\delta)$ such that

$$|f(\theta, \rho) - f(\theta', \rho')| \leq \delta.$$

Without loss of generality, let us consider the case where $P(\delta) = P_1(\delta) = P_2(\delta)$. For every latent row feature θ_i, let $p_1(\theta_i)$ be the unique element in $P(\delta)$ that is closest to θ_i. Similarly, for the latent column feature ρ_j, find the corresponding closest element in $P(\delta)$ and denote it by $p_2(\rho_j)$. Let $\boldsymbol{B} = [B_{ij}]$ be the matrix where $B_{ij} = f(p_1(\theta_i), p_2(\rho_j))$. Using the arguments from above, we have that for all i and j,

$$\|\boldsymbol{M} - \boldsymbol{B}\|_F^2 = \sum_{i,j} (f(\theta_i, \rho_j) - f(p_1(\theta_i), p_2(\rho_j)))^2 \ \leq (N-1)T\delta^2.$$

Therefore,

$$\begin{aligned} \|\boldsymbol{M}\|_* &\leq \|\boldsymbol{M} - \boldsymbol{B}\|_* + \|\boldsymbol{B}\|_* \\ &\overset{(a)}{\leq} \sqrt{N-1}\|\boldsymbol{M} - \boldsymbol{B}\|_F + \|\boldsymbol{B}\|_* \\ &\leq \delta(N-1)\sqrt{T} + \|\boldsymbol{B}\|_*, \end{aligned}$$

where (a) follows from the fact that $\|\boldsymbol{Q}\|_* \leq \sqrt{\text{rank}(\boldsymbol{Q})}\|\boldsymbol{Q}\|_F$ for any real-valued matrix $\boldsymbol{Q}$. In order to bound the nuclear norm of $\boldsymbol{B}$, note that (by its construction) for any two columns, say $j, j' \in [N-1]$, if $p_2(\rho_j) = p_2(\rho_{j'})$ then it follows that the columns of j and j' of $\boldsymbol{B}$ are identical. Thus, there can be at most $|P(\delta)|$ distinct columns (and rows) of $\boldsymbol{B}$. In other words, $\text{rank}(\boldsymbol{B}) \leq |P(\delta)|$. Ergo,

$$\begin{aligned} \|\boldsymbol{B}\|_* &\leq \sqrt{|P(\delta)|}\|\boldsymbol{B}\|_F \\ &\leq \sqrt{|P(\delta)|}\sqrt{(N-1)T}. \end{aligned}$$

Due to the Lipschitzness property of f and the compactness of the latent space, it can be shown that $|P(\delta)| \leq C(K, d, \mathcal{L})\delta^{-d}$ where $C(K, d, \mathcal{L})$ is a constant that depends only on K, d, and $\mathcal{L}$ (the Lipschitz constant of f). ■

Lemma 24 (**Theorem 1.1 of Chatterjee (2015)**) *Let $\hat{\boldsymbol{M}}$ and $\boldsymbol{M}$ be defined as before. Suppose that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Then using μ as defined in* (20)

$$\text{MSE}(\hat{\boldsymbol{M}}) \leq \frac{C_1\|\boldsymbol{M}\|_*}{(N-1)\sqrt{Tp}} + \mathcal{O}\Big(\frac{1}{(N-1)T}\Big),$$

where C_1 is a universal positive constant.

Proof Let $\delta > 0$ be defined by the relation

$$(1+\delta)\|\boldsymbol{Y} - p\boldsymbol{M}\| = (2+\omega)\sqrt{T\hat{q}},$$

where $\hat{q} = \hat{\sigma}^2\hat{p} + \hat{p}(1-\hat{p})$. Observe that if E_1, E_2, and E_3 happen, then

$$\begin{aligned}
1+\delta &\geq \frac{(2+\omega)\sqrt{T(\hat{\sigma}^2\hat{p}+\hat{p}(1-\hat{p}))}}{(2+\omega/2)\sqrt{T(\sigma^2 p + p(1-p))}} \\
&\geq \frac{(2+\omega)\sqrt{1-\omega/z}\sqrt{(1-\omega/z)\sigma^2 p + p(1-p-\omega p/z)}}{(2+\omega/2)\sqrt{\sigma^2 p + p(1-p)}} \\
&= \frac{(2+\omega)\sqrt{1-\omega/z}}{2+\omega/2}\sqrt{1-\frac{\omega}{z}\Big(\frac{\sigma^2+p}{\sigma^2+1-p}\Big)} \\
&\geq \frac{(2+\omega)\sqrt{1-\omega/z}}{2+\omega/2}\sqrt{1-\frac{\omega}{z}\Big(\frac{\sigma^2+1}{\sigma^2}\Big)} \\
&= \frac{(2+\omega)\sqrt{1-\omega/z}}{2+\omega/2}\sqrt{1-\frac{\omega}{60}} \\
&\geq \frac{2+\omega}{2+\omega/2}\Big(1-\frac{\omega}{60}\Big) \\
&\geq \Big(1+\frac{\omega}{5}\Big)\Big(1-\frac{\omega}{60}\Big) \\
&\geq 1+\frac{\omega}{5}-\frac{1}{50}.
\end{aligned}$$

Let $K(\delta)$ be the constant defined in Lemma 22. Since $\omega \in (0.1, 1)$, $\delta \geq \frac{10\omega-1}{50} > 0$ and

$$\begin{aligned}
K(\delta) &= (4+2\delta)\sqrt{2/\delta} + \sqrt{2+\delta} \\
&\leq 4\sqrt{1+\delta}\sqrt{2/\delta} + 2\sqrt{2(1+\delta)} + \sqrt{2(1+\delta)} \\
&= (4\sqrt{2/\delta} + 3\sqrt{2})\sqrt{1+\delta} \\
&\leq C_1\sqrt{1+\delta}
\end{aligned}$$

where C_1 is a constant that depends only on the choice of ω. By Lemma 22, if E_1, E_2 and E_3 occur, then

$$\begin{aligned}
\Big\|\hat{p}\hat{\boldsymbol{M}} - p\boldsymbol{M}\Big\|_F^2 &\leq C_2(1+\delta)\|\boldsymbol{Y} - p\boldsymbol{M}\|\|p\boldsymbol{M}\|_* \\
&\leq C_3\sqrt{T\hat{q}}\|p\boldsymbol{M}\|_* \\
&\leq C_4\sqrt{Tq}\|p\boldsymbol{M}\|_*
\end{aligned}$$

for an appropriately defined constant C_4. Therefore,

$$\begin{aligned}
p^2\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F^2 &\leq C_5\hat{p}^2\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F^2 \\
&\leq C_5\Big\|\hat{p}\hat{\boldsymbol{M}} - p\boldsymbol{M}\Big\|_F^2 + C_5(\hat{p}-p)^2\|\boldsymbol{M}\|_F^2 \\
&\leq C_6\sqrt{Tq}\|p\boldsymbol{M}\|_* + C_5(\hat{p}-p)^2(N-1)T,
\end{aligned}$$

where the last inequality follows from the boundedness assumption of $\boldsymbol{M}$. In general, since $|M_{ij}|$ and $|Y_{ij}| \leq 1$,

$$\begin{aligned}
\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F &\leq \Big\|\hat{\boldsymbol{M}}\Big\|_F + \|\boldsymbol{M}\|_F \\
&\leq \sqrt{|S|}\Big\|\hat{\boldsymbol{M}}\Big\| + \|\boldsymbol{M}\|_F \\
&= \frac{\sqrt{|S|}}{\hat{p}}\|\boldsymbol{Y}\| + \|\boldsymbol{M}\|_F \\
&\leq (N-1)^{3/2}T\|\boldsymbol{Y}\| + \|\boldsymbol{M}\|_F \\
&\leq (N-1)^{3/2}T\sqrt{(N-1)T} + \sqrt{(N-1)T} \\
&\leq 2(N-1)^2T^{3/2}.
\end{aligned}$$

Let $E := E_1 \cap E_2 \cap E_3$. Applying DeMorgan's Law and the Union Bound,

$$\begin{aligned}
\mathbb{P}(E^c) &= \mathbb{P}(E_1^c \cup E_2^c \cup E_e^c) \\
&\leq \mathbb{P}(E_1^c) + \mathbb{P}(E_2^c) + \mathbb{P}(E_3^c) \\
&\leq C_7 e^{-c_8 T(p(N-1)+\sigma^2+q)} \\
&= C_7 e^{-c_8\phi T}, \qquad (48)
\end{aligned}$$

where we define $\phi := p(N-1) + \sigma^2 + q$ and C_7, c_8 are appropriately defined. Observe that $\mathbb{E}(\hat{p}-p)^2 = \frac{p(1-p)}{(N-1)T}$. Thus, by the law of total probability and noting that $\mathbb{P}(E) \leq 1$ (for appropriately defined constants),

$$\begin{aligned}
\mathbb{E}\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F^2 &\leq \mathbb{E}\Big[\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F^2 \mid E\Big] + \mathbb{E}\Big[\Big\|\hat{\boldsymbol{M}} - \boldsymbol{M}\Big\|_F^2 \mid E^c\Big]\mathbb{P}(E^c) \\
&\leq C_6 p^{-1}\sqrt{Tq}\|\boldsymbol{M}\|_* + C_5 p^{-1}(1-p) + C_9(N-1)^4T^3e^{-c_8\phi T} \\
&= C_6 p^{-1/2}T^{1/2}(\sigma^2+(1-p))^{1/2}\|\boldsymbol{M}\|_* + C_5 p^{-1}(1-p) + C_9(N-1)^4T^3e^{-c_8\phi T}.
\end{aligned}$$

Normalizing by $(N-1)T$, we obtain

$$\mathrm{MSE}(\hat{\boldsymbol{M}}) \leq \frac{C_{12}\|\boldsymbol{M}\|_*}{(N-1)\sqrt{Tp}} + \frac{C_5(1-p)}{(N-1)Tp} + C_{10}e^{-c_{11}\phi T}.$$

The proof is complete assuming constants are re-named. ■

D.1 Proof of Theorem 1

Theorem 1 (Theorem 2.1 of Chatterjee (2015)) *Suppose that $\boldsymbol{M}$ is rank k. Suppose that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Then using μ as defined in* (20)*,*

$$\mathrm{MSE}(\hat{\boldsymbol{M}}) \leq C_1\sqrt{\frac{k}{(N-1)p}} + \mathcal{O}\Big(\frac{1}{(N-1)T}\Big),$$

where C_1 is a universal positive constant.

Proof By the low rank assumption of $\boldsymbol{M}$, we have that

$$\begin{aligned}
\|\boldsymbol{M}\|_* &\leq \sqrt{\mathrm{rank}(\boldsymbol{M})}\|\boldsymbol{M}\|_F \\
&\leq \sqrt{k(N-1)T}.
\end{aligned}$$

The proof follows from a simple application of Lemma 24. ■

D.2 Proof of Theorem 2

Theorem 2 (Theorem 2.7 of Chatterjee (2015)) *Suppose f is a $\mathcal{L}$-Lipschitz function. Suppose that $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Then using μ as defined in (20),*

$$\text{MSE}(\hat{\boldsymbol{M}}) \leq C(K,d,\mathcal{L})\frac{(N-1)^{-\frac{1}{d+2}}}{\sqrt{p}} + \mathcal{O}\Big(\frac{1}{(N-1)T}\Big),$$

where $C(K,d,\mathcal{L})$ is a constant depending on K, d, and $\mathcal{L}$.

Proof Since f is Lipschitz, we invoke Lemmas 23 and 24 and choose $\delta = (N-1)^{-1/(d+2)}$. This completes the proof. ■

Appendix E. Forecasting Analysis: Pre-Intervention Regime

Here, we will bound the pre-intervention ℓ_2 error of our estimator in order to measure its prediction power.

E.1 Linear Regression

In this section, we will analyze the performance of our algorithm when learning β^* via linear regression, i.e. $\eta = 0$. As a result, we will temporarily drop the dependency on η in this subsection such that $\hat{\beta} = \hat{\beta}(0)$. To ease the notational complexity of the following Lemma 25 proof, we will make use of the following notations for **only** in this subsection:

$$\boldsymbol{Q} := (\boldsymbol{M}^-)^T \tag{49}$$
$$\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T \tag{50}$$

such that

$$M_1^- := \boldsymbol{Q}\beta^* \tag{51}$$
$$\hat{M}_1^- := \hat{\boldsymbol{Q}}\hat{\beta}. \tag{52}$$

Lemma 25 *Suppose $Y_1^- = M_1^- + \epsilon_1^-$ with $\mathbb{E}[\epsilon_{1j}] = 0$ and $Var(\epsilon_{1j}) \leq \sigma^2$ for all $j \in [T_0]$. Let β^* be defined as in (6) and let $\hat{\beta}$ be the minimizer of (10). Then for any $\mu \geq 0$ and $\eta = 0$,*

$$\mathbb{E}\Big\|M_1^- - \hat{M}_1^-\Big\|^2 \leq \mathbb{E}\Big\|(\boldsymbol{M}^- - \hat{\boldsymbol{M}}^-)^T\beta^*\Big\|^2 + 2\sigma^2|S|. \tag{53}$$

Proof Recall that for the treatment row, $Y_1^- = M_1^- + \epsilon_1^-$ with $M_1^- = \boldsymbol{Q}\beta^*$. Since $\hat{\beta}$, by definition, minimizes $\Big\|Y_1^- - \hat{\boldsymbol{Q}}v\Big\|$ for any $v \in \mathbb{R}^{N-1}$, we subsequently have

$$\begin{aligned}
\Big\|M_1^- - \hat{M}_1^-\Big\|^2 &= \Big\|(Y_1^- - \epsilon_1^-) - \hat{\boldsymbol{Q}}\hat{\beta}\Big\|^2 \\
&= \Big\|(Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}) + (-\epsilon_1^-)\Big\|^2 \\
&= \Big\|Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\Big\|^2 + \big\|\epsilon_1^-\big\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle \\
&\leq \Big\|Y_1^- - \hat{\boldsymbol{Q}}\beta^*\Big\|^2 + \big\|\epsilon_1^-\big\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle \\
&= \Big\|(\boldsymbol{Q}\beta^* + \epsilon_1^-) - \hat{\boldsymbol{Q}}\beta^*\Big\|^2 + \big\|\epsilon_1^-\big\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle \\
&= \Big\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^* + \epsilon_1^-\Big\|^2 + \big\|\epsilon_1^-\big\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle \\
&= \Big\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\Big\|^2 + 2\big\|\epsilon_1^-\big\|^2 + 2\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle.
\end{aligned}$$

Taking expectations, we arrive at the inequality

$$\mathbb{E}\Big\|\hat{M}_1^- - M_1^-\Big\|^2 \leq \mathbb{E}\Big\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\Big\|^2 + 2\mathbb{E}\big\|\epsilon_1^-\big\|^2 + 2\mathbb{E}[\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle] + 2\mathbb{E}[\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle]. \tag{54}$$

We will now deal with the two inner products on the right hand side of equation (54). First, observe that

$$\begin{aligned}
\mathbb{E}[\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle] &= \mathbb{E}[(\epsilon_1^-)^T]\boldsymbol{Q}\beta^* - \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}]\beta^* \\
&= -\mathbb{E}[(\epsilon_1^-)^T]\mathbb{E}[\hat{\boldsymbol{Q}}]\beta^* \\
&= 0,
\end{aligned}$$

since the additive noise terms are independent random variables that satisfy $\mathbb{E}[\epsilon_{ij}] = 0$ for all i and j by assumption, and $\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T$ depends only on the noise terms for $i \neq 1$; i.e., the construction of $\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T$ excludes the first row (treatment row), and thus depends solely on the donor pool.

For the other inner product term, we begin by recognizing that $(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-$ is a scalar random variable, which allows us to replace the random variable by its own trace. This is useful since the trace operator is a linear mapping and is invariant under cyclic permutations, i.e., $\text{tr}(\boldsymbol{AB}) = \text{tr}(\boldsymbol{BA})$. As a result,

$$\begin{aligned}
\mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-] &= \mathbb{E}[\text{tr}((\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-)] \\
&= \mathbb{E}[\text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-(\epsilon_1^-)^T)] \\
&= \text{tr}\Big(\mathbb{E}[\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-(\epsilon_1^-)^T]\Big) \\
&= \text{tr}\Big(\mathbb{E}[\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger]\mathbb{E}[\epsilon_1^-(\epsilon_1^-)^T]\Big) \\
&= \text{tr}\Big(\mathbb{E}[\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger]\sigma^2 I\Big) \\
&= \sigma^2\mathbb{E}[\text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger)] \\
&\overset{(a)}{=} \sigma^2\mathbb{E}[\text{rank}(\hat{\boldsymbol{Q}})] \\
&\leq \sigma^2|S|,
\end{aligned}$$

where (a) follows from the fact that $\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger$ is a projection matrix by Lemma 18. As a result, $\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger$ has $\text{rank}(\hat{\boldsymbol{Q}})$ eigenvalues equal to 1 and all other eigenvalues equal to 0 (by Lemma 19), and since the trace of a matrix is equal to the sum of its eigenvalues, $\text{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger) = \text{rank}(\hat{\boldsymbol{Q}})$. Simultaneously, by the definition of $\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T$, we have that the rank of $\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T$ is at most $|S|$. Returning to the second inner product and recalling $\hat{\beta} = \hat{\boldsymbol{Q}}^\dagger Y_1^-$,

$$\begin{aligned}
\mathbb{E}[\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}\rangle] &= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\beta}] - \mathbb{E}[(\epsilon_1^-)^T Y_1^-] \\
&= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger Y_1^-] - \mathbb{E}[(\epsilon_1^-)^T]M_1^- - \mathbb{E}[(\epsilon_1^-)^T\epsilon_1^-] \\
&= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger]M_1^- + \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-] - \mathbb{E}[(\epsilon_1^-)^T\epsilon_1^-] \\
&\overset{(a)}{=} \mathbb{E}[(\epsilon_1^-)^T]\mathbb{E}[\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger]M_1^- + \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-] - \mathbb{E}[(\epsilon_1^-)^T\epsilon_1^-] \\
&= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger\epsilon_1^-] - \mathbb{E}\|\epsilon_1^-\|^2 \\
&\leq \sigma^2|S| - \mathbb{E}\|\epsilon_1^-\|^2,
\end{aligned}$$

where (a) follows from the same independence argument used in evaluating the first inner product. Finally, we incorporate the above results to (54) to arrive at the inequality

$$\begin{aligned}
\mathbb{E}\left\|\hat{M}_1^- - M_1^-\right\|^2 &\leq \mathbb{E}\left\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\right\|^2 + 2\mathbb{E}\|\epsilon_1^-\|^2 + 2(\sigma^2|S| - \mathbb{E}\|\epsilon_1^-\|^2) \\
&= \mathbb{E}\left\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\right\|^2 + 2\sigma^2|S|.
\end{aligned}$$

■

Lemma 26 *For $\eta = 0$ and any $\mu \geq 0$, the pre-intervention error of the algorithm can be bounded as*

$$\text{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p^2T_0}\mathbb{E}\Big(\lambda^* + \|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big)^2 + \frac{2\sigma^2|S|}{T_0} + C_2e^{-cp(N-1)T}. \tag{55}$$

Here, $\lambda_1, \ldots, \lambda_{N-1}$ are the singular values of $p\boldsymbol{M}$ in decreasing order and repeated by multiplicities, with $\lambda^* = \max_{i \notin S}\lambda_i$; C_1, C_2 and c are universal positive constants.

Proof Recall that $E_1 := \{|\hat{p} - p| \leq \frac{\omega p}{z}\}$ for some choice of $\omega \in (0.1, 1)$. Thus, under the event E_1,

$$\begin{aligned}
p\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| &\leq C_1\hat{p}\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| \\
&\leq C_1\Big(\left\|\hat{p}\hat{\boldsymbol{M}}^- - p\boldsymbol{M}^-\right\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big) \\
&\overset{(a)}{\leq} C_1\Big(\left\|\hat{p}\hat{\boldsymbol{M}} - p\boldsymbol{M}\right\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big) \\
&\overset{(b)}{\leq} C_1\Big(\lambda^* + 2\|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big)
\end{aligned}$$

where (a) follows from Lemma 17 and (b) follows from Lemma 20. In general, since $|M_{ij}|$ and $|Y_{ij}| \leq 1$,

$$\begin{aligned}
\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| &\overset{(a)}{\leq} \left\|\hat{\boldsymbol{M}}\right\| + \|\boldsymbol{M}^-\| \\
&= \frac{1}{\hat{p}}\|\boldsymbol{Y}\| + \left\|\boldsymbol{M}^-\right\| \\
&\leq (N-1)T\|\boldsymbol{Y}\| + \left\|\boldsymbol{M}^-\right\| \\
&\leq (N-1)T\sqrt{(N-1)T} + \sqrt{(N-1)T_0} \\
&\leq 2((N-1)T)^{3/2}.
\end{aligned} \tag{56}$$

(a) follows from a simple application of Lemma 17 and the triangle inequality of operator norms. By the law of total probability and $\mathbb{P}(E_1) \leq 1$,

$$\begin{aligned}
\mathbb{E}\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 &\leq \mathbb{E}\Big[\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 \mid E_1\Big] + \mathbb{E}\Big[\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 \mid E_1^c\Big]\mathbb{P}(E_1^c) \\
&\overset{(a)}{\leq} \mathbb{E}\Big[\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\|^2 \mid E_1\Big]\|\beta^*\|^2 + \mathbb{E}\Big[\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\|^2 \mid E_1^c\Big]\|\beta^*\|^2\mathbb{P}(E_1^c) \\
&\leq \frac{C_2}{p^2}\mathbb{E}\Big[\Big(\lambda^* + 2\|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big)^2 \mid E_1\Big] + C_3((N-1)T)^{3/2}e^{-cp(N-1)T},
\end{aligned}$$

where (a) follows because the spectral norm is an induced norm, and the last inequality makes use of the results from above. Note that C_2 and C_3 are appropriately defined to depend on β^*. Moreover, for any non-negative valued random variable X and event E with $\mathbb{P}(E) \geq 1/2$,

$$\mathbb{E}[X \mid E] \leq \frac{\mathbb{E}[X]}{\mathbb{P}(E)} \leq 2\mathbb{E}[X]. \tag{57}$$

Using the fact that $\mathbb{P}(E_1) \geq 1/2$ for large enough T, N, we apply Lemma 25 to obtain (with appropriately defined constants C_4, C_5, c_6)

$$\begin{aligned}
\text{MSE}(\hat{M}_1^-) &\leq \frac{1}{T_0}\mathbb{E}\left\|(\boldsymbol{M}^- - \hat{\boldsymbol{M}}^-)^T\beta^*\right\|^2 + \frac{2\sigma^2|S|}{T_0} \\
&\leq \frac{C_4}{p^2T_0}\mathbb{E}\Big(\lambda^* + \|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\Big)^2 + \frac{2\sigma^2|S|}{T_0} + C_5e^{-c_6p(N-1)T}.
\end{aligned} \tag{58}$$

The proof is completed assuming we re-label constants C_4, C_5, c_6 as C_1, C_2, and c, respectively. ■

E.2 Ridge Regression

In this section, we will prove our results for the ridge regression setting, i.e. $\eta > 0$. Let us begin by deriving the closed form expression of $\hat{\beta}(\eta)$.

Derivation of $\hat{\beta}(\eta)$. We derive the closed form solution for $\hat{\beta}(\eta)$ under the new objective function with the additional complexity penalty term:

$$\left\|Y_1^- - (\hat{\boldsymbol{M}}^-)^Tv\right\|^2 + \eta\|v\|^2 = (Y_1^-)^TY_1^- - 2v^T\hat{\boldsymbol{M}}^-Y_1^- + v^T\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^Tv + \eta v^Tv.$$

Setting the gradient of the above expression to zero and solving for v, we obtain

$$\nabla_v\left\{\left\|Y_1^- - (\hat{\boldsymbol{M}}^-)^Tv\right\|^2 + \eta\|v\|^2\right\}_{v=\hat{\beta}(\eta)} = -2\hat{\boldsymbol{M}}^-Y_1^- + 2\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^Tv + 2\eta v = 0.$$

Therefore,

$$\hat{\beta}(\eta) = \Big(\hat{\boldsymbol{M}}^-(\hat{\boldsymbol{M}}^-)^T + \eta\boldsymbol{I}\Big)^{-1}\hat{\boldsymbol{M}}^-Y_1^-.$$

Remark 27 *To ease the notational complexity of the following Lemmas 28 and 30 proofs, we will make use of the following notations for* **only** *this derivation: Let*

$$\boldsymbol{Q} := (\boldsymbol{M}^-)^T \tag{59}$$

$$\hat{\boldsymbol{Q}} := (\hat{\boldsymbol{M}}^-)^T \tag{60}$$

such that

$$M_1^- := \boldsymbol{Q}\beta^* \tag{61}$$
$$\hat{M}_1^- := \hat{\boldsymbol{Q}}\hat{\beta}. \tag{62}$$

Lemma 28 *Let $\boldsymbol{P}_\eta = \hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T$ denote the "hat" matrix under the quadratic regularization setting. Then, the non-zero singular values of $\boldsymbol{P}_\eta$ are $s_i^2/(s_i^2+\eta)$ for all $i \in S$.*

Proof Recall that the singular values of $\boldsymbol{Y}$ are s_i, while the singular values of $\hat{\boldsymbol{Q}}$ are those $s_i \geq \mu$. Let $\hat{\boldsymbol{Q}} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T$ be the singular value decomposition of $\hat{\boldsymbol{Q}}$. Since $\boldsymbol{V}\boldsymbol{V}^T = \boldsymbol{I}$, we have that

$$\begin{aligned}
\boldsymbol{P}_\eta &= \hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T \\
&= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T+\eta\boldsymbol{I})^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T \\
&= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T(\boldsymbol{V}\boldsymbol{\Sigma}^2\boldsymbol{V}^T+\eta\boldsymbol{V}\boldsymbol{V}^T)^{-1}\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T \\
&= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T\boldsymbol{V}(\boldsymbol{\Sigma}^2+\eta\boldsymbol{I})^{-1}\boldsymbol{V}^T\boldsymbol{V}\boldsymbol{\Sigma}\boldsymbol{U}^T \\
&= \boldsymbol{U}\boldsymbol{\Sigma}(\boldsymbol{\Sigma}^2+\eta\boldsymbol{I})^{-1}\boldsymbol{\Sigma}\boldsymbol{U}^T \\
&= \boldsymbol{U}\boldsymbol{D}\boldsymbol{U}^T,
\end{aligned}$$

where

$$\boldsymbol{D} = \operatorname{diag}\left(\frac{s_1^2}{s_1^2+\eta}, \ldots, \frac{s_{|S|}^2}{s_{|S|}^2+\eta}, 0, \ldots, 0\right).$$

■

Lemma 29 *Suppose $Y_1^- = M_1^- + \epsilon_1^-$ with $\mathbb{E}[\epsilon_{1j}] = 0$ and $Var(\epsilon_{1j}) \leq \sigma^2$ for all $j \in [T_0]$. Let β^* be defined as in (6), i.e. $M_1^- = (\boldsymbol{M}^-)^T\beta^*$, and let $\hat{\beta}(\eta)$ be the minimizer of (10). Then for any $\mu \geq 0$ and $\eta > 0$,*

$$\mathbb{E}\left\|M_1^- - \hat{M}_1^-\right\|^2 \leq \mathbb{E}\left\|(\boldsymbol{M}^- - \hat{\boldsymbol{M}}^-)^T\beta^*\right\|^2 + \eta\|\beta^*\|^2 - \eta\mathbb{E}\left\|\hat{\beta}(\eta)\right\|^2 + 2\sigma^2|S|. \tag{63}$$

Proof The following proof is a slight modification for the proof of Lemmas 25. In particular, observe that $\hat{\beta}(\eta)$ minimizes $\left\|Y_1^- - \hat{\boldsymbol{Q}}v\right\| + \eta\|v\|^2$ for any $v \in \mathbb{R}^{N-1}$. As a result,

$$\begin{aligned}
&\left\|M_1^- - \hat{M}_1^-\right\|^2 + \eta\left\|\hat{\beta}(\eta)\right\|^2 \\
&= \left\|(Y_1^- - \epsilon_1^-) - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\right\|^2 + \eta\left\|\hat{\beta}(\eta)\right\|^2 \\
&= \left\|(Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)) + (-\epsilon_1^-)\right\|^2 + \eta\left\|\hat{\beta}(\eta)\right\|^2 \\
&= \left\|Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\right\|^2 + \eta\left\|\hat{\beta}(\eta)\right\|^2 + \left\|\epsilon_1^-\right\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle \\
&\leq \left\|Y_1^- - \hat{\boldsymbol{Q}}\beta^*\right\|^2 + \eta\|\beta^*\|^2 + \left\|\epsilon_1^-\right\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle \\
&= \left\|(\boldsymbol{Q}\beta^* + \epsilon_1^-) - \hat{\boldsymbol{Q}}\beta^*\right\|^2 + \eta\|\beta^*\|^2 + \left\|\epsilon_1^-\right\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle \\
&= \left\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^* + \epsilon_1^-\right\|^2 + \eta\|\beta^*\|^2 + \left\|\epsilon_1^-\right\|^2 + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle \\
&= \left\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\right\|^2 + \eta\|\beta^*\|^2 + 2\left\|\epsilon_1^-\right\|^2 + 2\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle + 2\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle
\end{aligned}$$

Taking expectations, we have

$$\begin{aligned}
&\mathbb{E}\left\|\hat{M}_1^- - M_1^-\right\|^2 \\
&\quad\leq \mathbb{E}\left\|(\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\right\|^2 + \eta\left(\|\beta^*\|^2 - \mathbb{E}\left\|\hat{\beta}(\eta)\right\|^2\right) + 2\mathbb{E}\left\|\epsilon_1^-\right\|^2 + 2\mathbb{E}\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle + 2\mathbb{E}\langle -\epsilon_1^-, Y_1^- - \hat{\boldsymbol{Q}}\hat{\beta}(\eta)\rangle.
\end{aligned}$$

As before, we have that $\mathbb{E}\langle \epsilon_1^-, (\boldsymbol{Q} - \hat{\boldsymbol{Q}})\beta^*\rangle = 0$ by the zero-mean and independence assumptions of the noise random variables. Similarly, note that

$$\begin{aligned}
\mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}\hat{\beta}(\eta)] &= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^TY_1^-] \\
&= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T]M_1^- + \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^T+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T\epsilon_1^-] \\
&= \mathbb{E}[(\epsilon_1^-)^T\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T\epsilon_1^-] \\
&= \mathbb{E}[\operatorname{tr}((\epsilon_1^-)^T\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T\epsilon_1^-)] \\
&= \mathbb{E}[\operatorname{tr}(\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T\epsilon_1^-(\epsilon_1^-)^T)] \\
&= \operatorname{tr}(\mathbb{E}[\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T\epsilon_1^-(\epsilon_1^-)^T]) \\
&= \operatorname{tr}(\mathbb{E}[\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T]\mathbb{E}[\epsilon_1^-(\epsilon_1^-)^T]) \\
&= \sigma^2\mathbb{E}[\operatorname{tr}(\hat{\boldsymbol{Q}}(\hat{\boldsymbol{Q}}^T\hat{\boldsymbol{Q}}+\eta\boldsymbol{I})^{-1}\hat{\boldsymbol{Q}}^T)] \\
&\overset{(a)}{\leq} \sigma^2\mathbb{E}[\operatorname{tr}(\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger)] \\
&\overset{(b)}{=} \sigma^2\operatorname{rank}(\hat{\boldsymbol{Q}}) \\
&\leq \sigma^2|S|,
\end{aligned}$$

where (a) follows from Lemma 28, and as before, (b) follows because $\hat{\boldsymbol{Q}}\hat{\boldsymbol{Q}}^\dagger$ is a projection matrix. ■

Lemma 30 *For any $\eta > 0$ and $\mu \geq 0$, the pre-intervention error of the regularized algorithm can be bounded as*

$$\operatorname{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p^2T_0}\mathbb{E}\left(\lambda^* + \|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\right)^2 + \frac{2\sigma^2|S|}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_2e^{-cp(N-1)T}.$$

Here, $\lambda_1, \ldots, \lambda_{N-1}$ are the singular values of $p\boldsymbol{M}$ in decreasing order and repeated by multiplicities, with $\lambda^* = \max_{i\notin S}\lambda_i$; C_1, C_2 and c are universal positive constants.

Proof The proof follows the same arguments as that of Lemma 26. ■

E.3 Combining linear and ridge regression.

E.3.1 Proof of Theorem 3

Theorem *3 For any $\eta \geq 0$ and $\mu \geq 0$, the pre-intervention error of the algorithm can be bounded as*

$$\operatorname{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p^2T_0}\mathbb{E}\left(\lambda^* + \|\boldsymbol{Y} - p\boldsymbol{M}\| + \left\|(\hat{p} - p)\boldsymbol{M}^-\right\|\right)^2 + \frac{2\sigma^2|S|}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_2e^{-cp(N-1)T}.$$

Here, $\lambda_1, \ldots, \lambda_{N-1}$ are the singular values of $p\boldsymbol{M}$ in decreasing order and repeated by multiplicities, with $\lambda^* = \max_{i\notin S}\lambda_i$; C_1, C_2 and c are universal positive constants.

Proof The proof follows from a simple amalgamation of Lemmas 26 and 30. ■

E.3.2 Proof of Corollary 4

Theorem 4 *Suppose $p \geq \frac{T^{-1+\zeta}}{\sigma^2+1}$ for some $\zeta > 0$. Let $T \leq \alpha T_0$ for some constant $\alpha > 1$. Then for any $\eta \geq 0$ and using μ as defined in (20), the pre-intervention error is bounded above by*

$$\text{MSE}(\hat{M}_1^-) \leq \frac{C_1}{p}(\sigma^2 + (1-p)) + \mathcal{O}(1/\sqrt{T_0}),$$

where C_1 is a universal positive constant.

Proof Since the singular value threshold $\mu = (2+\omega)\sqrt{T\hat{q}}$, let us define δ so that

$$(1+\delta)\|\boldsymbol{Y} - p\boldsymbol{M}\| = (2+\omega)\sqrt{T\hat{q}},$$

where $\hat{q} = \hat{\sigma}^2\hat{p} + \hat{p}(1-\hat{p})$; recall that $q = \sigma^2 p + p(1-p)$. If E_3 happens, then we know that $\delta \geq 0$. Therefore, assuming E_1, E_2, and E_3 happens, Lemma 21 states that

$$\begin{aligned}\left\|\hat{p}\hat{\boldsymbol{M}} - p\boldsymbol{M}\right\| &\leq (2+\delta)\|\boldsymbol{Y} - p\boldsymbol{M}\| \\ &\leq 2(1+\delta)\|\boldsymbol{Y} - p\boldsymbol{M}\| \\ &= (4+2\omega)\sqrt{T\hat{q}} \\ &\leq C_1\sqrt{Tq} \end{aligned} \tag{64}$$

for an appropriately defined constant C_1. Therefore,

$$\begin{aligned} p\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| &\leq C_2\hat{p}\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| \\ &\leq C_2\Big(\left\|\hat{p}\hat{\boldsymbol{M}}^- - p\boldsymbol{M}^-\right\| + \left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big) \\ &\overset{(a)}{\leq} C_2\Big(\left\|\hat{p}\hat{\boldsymbol{M}} - p\boldsymbol{M}\right\| + \left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big) \\ &\leq C_2\Big(C_1\sqrt{Tq} + \left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big) \end{aligned} \tag{65}$$

where (a) follows from Lemma 17. Applying the logic that led to (56), we have that, in general,

$$\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\| \leq 2((N-1)T)^{3/2}. \tag{66}$$

Let $E := E_1 \cap E_2 \cap E_3$. Further, using the same argument that led to (48), we have

$$\mathbb{P}(E^c) \leq C_3 e^{-c_4\phi T}$$

where we define $\phi := p(N-1) + \sigma^2 + q$ and C_3, c_4 are appropriately defined. Thus, by the law of total probability and noting that $\mathbb{P}(E) \leq 1$,

$$\begin{aligned}\mathbb{E}\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 &\leq \mathbb{E}\Big[\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 \mid E\Big] + \mathbb{E}\Big[\left\|(\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-)^T\beta^*\right\|^2 \mid E^c\Big]\mathbb{P}(E^c) \\ &\overset{(a)}{\leq} \mathbb{E}\Big[\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\|^2 \mid E\Big]\|\beta^*\|^2 + \mathbb{E}\Big[\left\|\hat{\boldsymbol{M}}^- - \boldsymbol{M}^-\right\|^2 \mid E^c\Big]\|\beta^*\|^2\mathbb{P}(E^c) \\ &\leq \frac{C_5}{p^2}\mathbb{E}\Big[\Big(\sqrt{Tq} + \left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big)^2 \mid E\Big] + C_6((N-1)T)^3 e^{-c_4\phi T}, \end{aligned} \tag{67}$$

where (a) follows because the spectral norm is an induced norm and the last inequality makes use of the results from above. Note that C_5 and C_6 are appropriately defined to depend on β^*. Using the

fact that $\mathbb{P}(E) \geq 1/2$ for large enough T, N, we apply Lemmas 25 and 30 as well as (57) to obtain (with appropriately defined constants C_7, C_8, c_9)

$$\begin{aligned}\text{MSE}(\hat{M}_1^-) &\leq \frac{1}{T_0}\mathbb{E}\left\|(\boldsymbol{M}^- - \hat{\boldsymbol{M}}^-)^T\beta^*\right\|^2 + \frac{2\sigma^2|S|}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} \\ &\leq \frac{C_7}{p^2T_0}\mathbb{E}\Big(\sqrt{Tq} + \left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big)^2 + \frac{2\sigma^2(N-1)}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_8 e^{-c_9\phi T}. \end{aligned} \tag{68}$$

From Jensen's Inequality, $\mathbb{E}|\hat{p}-p| \leq \sqrt{\text{Var}(\hat{p})}$ where $\text{Var}(\hat{p}) = \frac{p(1-p)}{(N-1)T}$. Therefore,

$$\begin{aligned}\mathbb{E}\Big(\sqrt{Tq}\left\|(\hat{p}-p)\boldsymbol{M}^-\right\|\Big) &\leq \frac{q^{1/2}\sqrt{p(1-p)}}{\sqrt{N-1}}\left\|\boldsymbol{M}^-\right\| \\ &\leq \sqrt{qp(1-p)T_0}. \end{aligned}$$

At the same time,

$$\begin{aligned}\mathbb{E}\left\|(\hat{p}-p)\boldsymbol{M}^-\right\|^2 &= \mathbb{E}(\hat{p}-p)^2 \cdot \left\|\boldsymbol{M}^-\right\|^2 \\ &\leq \frac{p(1-p)T_0}{T} \\ &\leq p(1-p). \end{aligned}$$

Putting everything together, we arrive at the inequality

$$\begin{aligned}\text{MSE}(\hat{M}_1^-) &\leq \frac{C_7}{p^2T_0}\Big(qT + p(1-p) + 2\sqrt{qp(1-p)T_0}\Big) + \frac{2\sigma^2(N-1)}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_8 e^{-c_9\phi T} \\ &= \frac{C_{10}q}{p^2} + \frac{C_7(1-p)}{pT_0} + \frac{C_{11}(q(1-p))^{1/2}}{p^{3/2}\sqrt{T_0}} + \frac{2\sigma^2(N-1)}{T_0} + \frac{\eta\|\beta^*\|^2}{T_0} + C_8 e^{-c_9\phi T} \\ &= \frac{C_{10}}{p}(\sigma^2 + (1-p)) + \mathcal{O}(1/\sqrt{T_0}). \end{aligned}$$

The proof is complete assuming we re-label C_{10} as C_1. ■

E.3.3 Proof of Theorem 5

Theorem 5 *Fix any $\gamma \in (0, 1/2)$ and $\omega \in (0.1, 1)$. Let $\Delta = T_0^{\frac{1}{2}+\gamma}$ and $\mu = (2+\omega)\sqrt{T_0^{2\gamma}(\hat{\sigma}^2\hat{p} + \hat{p}(1-\hat{p}))}$. Suppose $p \geq \frac{T_0^{-2\gamma}}{\sigma^2+1}$ is known. Then for any $\eta \geq 0$,*

$$\text{MSE}(\hat{\hat{M}}_1^-) = \mathcal{O}(T_0^{-1/2+\gamma}).$$

Proof To establish Theorem 5, we shall follow the proof of Corollary 4, using the block partitioned matrices instead. Recall that $\tau = T_0/\Delta$ where $\Delta = T_0^{1/2+\gamma}$. For analytical simplicity, we define the random variable

$$D_{it} = \begin{cases} 1 & \text{w.p. } p, \\ 0 & \text{otherwise,} \end{cases}$$

whose definition will soon prove to be useful. As previously described in Section 4, for all $i > 1$ and $j \in [\Delta]$, we define

$$\bar{X}_{ij} = \frac{1}{\tau}\sum_{t \in B_j} X_{it} \cdot D_{it}$$

and

$$\bar{M}_{ij} = \frac{p}{\tau} \sum_{t \in B_j} M_{it}.$$

Let us also define $\bar{\boldsymbol{E}}^- = [\bar{\epsilon}_{ij}]_{2 \le i \le N, j \le \Delta}$ with entries

$$\bar{\epsilon}_{ij} = \frac{1}{\tau} \sum_{t \in B_j} \epsilon_{it} \cdot D_{it}. \tag{69}$$

For the first row (treatment unit), since we know p by assumption, we define for all $j \in [\Delta]$

$$\begin{aligned} \bar{X}_{1j} &= \frac{p}{\tau} \sum_{t \in B_j} X_{1t} && (70) \\ &= \frac{p}{\tau} \sum_{t \in B_j} (M_{1t} + \epsilon_{1t}) \\ &= \frac{p}{\tau} \sum_{t \in B_j} M_{1t} + \frac{p}{\tau} \sum_{t \in B_j} \epsilon_{1t} \\ &= \bar{M}_{1j} + \bar{\epsilon}_{1j}, && (71) \end{aligned}$$

whereby $\bar{M}_{1j} = \frac{p}{\tau} \sum_{t \in B_j} M_{1t}$ and $\bar{\epsilon}_{1j} = \frac{p}{\tau} \sum_{t \in B_j} \epsilon_{1t}$. Under these constructions, the noise entries remain zero-mean random variables for all i, j, i.e. $\mathbb{E}[\bar{\epsilon}_{ij}] = 0$. However, the variance of each noise term is now rescaled, i.e. for $i = 1$

$$\begin{aligned} \mathrm{Var}(\bar{\epsilon}_{1j}) &= \frac{p^2}{\tau^2} \sum_{t \in B_j} \mathrm{Var}(\epsilon_{1t}) \\ &= \frac{\sigma^2}{\tau}, \end{aligned}$$

and for $i > 1$,

$$\begin{aligned} \mathrm{Var}(\bar{\epsilon}_{ij}) &= \frac{1}{\tau^2} \sum_{t \in B_j} \mathrm{Var}(\epsilon_{it} \cdot D_{it}) \\ &\overset{(a)}{\le} \frac{1}{\tau^2} \sum_{t \in B_j} (\sigma^2 p(1-p) + \sigma^2 p^2) \\ &\le \frac{\sigma^2}{\tau}. \end{aligned}$$

(a) used the fact that for any two independent random variables, X and Y, $\mathrm{Var}(XY) = \mathrm{Var}(X)\mathrm{Var}(Y) + \mathrm{Var}(X)(\mathbb{E}[Y])^2 + \mathrm{Var}(Y)(\mathbb{E}[X])^2$. Thus, for all i, j, $\mathrm{Var}(\bar{\epsilon}_{ij}) \le \sigma^2/\tau := \bar{\sigma}^2$.

We now show that the key assumption of (6) still holds under this setting with respect to the newly defined variables. In particular, for every partition $j \in [\Delta]$ of row one,

$$\begin{aligned} \bar{M}_{1j} &= \frac{p}{\tau} \sum_{t \in B_j} M_{1t} \\ &= \frac{p}{\tau} \sum_{t \in B_j} \Big(\sum_{k=2}^{N} \beta_k^* M_{kt} \Big) \\ &= \sum_{k=2}^{N} \beta_k^* \Big(\frac{p}{\tau} \sum_{t \in B_j} M_{kt} \Big) \\ &= \sum_{k=2}^{N} \beta_k^* \bar{M}_{kj}. \end{aligned}$$

As a result, we can express $\bar{M}_1^- = (\bar{\boldsymbol{M}}^-)^T \beta^*$ for the same β^* as in (6).

Following a similar setup as before, we define the matrix $\bar{\boldsymbol{Y}}^- = [\bar{Y}_{ij}]_{2 \le i \le N, j \le \Delta}$. Since we have assumed that each block contains at least one observed entry, we subsequently have that $\bar{Y}_{ij} = \bar{X}_{ij}$ for all i and j. We now proceed with our analysis in the exact same manner with the only difference being our newly defined set of variables and parameters. For completeness, we will highlight certain details below.

To begin, observe that $\mathbb{E}[\bar{Y}_{ij}] = \bar{M}_{ij}$ while

$$\mathrm{Var}(\bar{Y}_{ij}) \le \frac{\sigma^2 p + p(1-p)}{\tau}.$$

Consequently, we redefine the event $E_3 := \{ \|\bar{\boldsymbol{Y}}^- - \bar{\boldsymbol{M}}^-\| \le (2+\omega)\sqrt{\Delta \bar{q}} \}$ for some choice $\omega \in (0.1, 1)$ and for $\bar{q} = \frac{\sigma^2 p + p(1-p)}{\tau}$. By Theorem 15, it follows that $\mathbb{P}(E_3) \ge 1 - C' e^{-c\bar{q}\Delta}$.

Similar to before, let δ be defined by the relation

$$(1+\delta)\|\bar{\boldsymbol{Y}}^- - \bar{\boldsymbol{M}}^-\| = (2+\omega)\sqrt{\Delta \hat{\bar{q}}},$$

where $\hat{\bar{q}} = \frac{\hat{\sigma}^2 \hat{p} + \hat{p}(1-\hat{p})}{\tau}$. Letting $E = E_1 \cap E_2 \cap E_3$ and using arguments ((64), (65), (56)) that led us to (67), we obtain

$$\begin{aligned} \mathbb{E}\Big\| (\hat{\boldsymbol{M}}^- - \bar{\boldsymbol{M}}^-)^T \beta^* \Big\|^2 &\le \mathbb{E}\Big[\Big\| (\hat{\boldsymbol{M}}^- - \bar{\boldsymbol{M}}^-)^T \beta^* \Big\|^2 \mid E \Big] + \mathbb{E}\Big[\Big\| (\hat{\boldsymbol{M}}^- - \bar{\boldsymbol{M}}^-)^T \beta^* \Big\|^2 \mid E^c \Big] \mathbb{P}(E^c) \\ &\le C_1 \Delta \bar{q} + C_2 e^{-c_3 \phi \Delta}, \end{aligned}$$

where $\phi := p(N-1) + \sigma^2 + \bar{q}$. Utilizing Lemmas 25 and 30 gives us (for appropriately defined constants and defining $q = \sigma^2 p + p(1-p)$ as before such that $\bar{q} = q/\tau$)

$$\begin{aligned} \mathrm{MSE}(\hat{\bar{M}}_1^-) &\le C_1 \bar{q} + \frac{2\bar{\sigma}^2 k}{\Delta} + \frac{\eta \|\beta^*\|^2}{\Delta} + C_4 e^{-c_5 \phi \Delta}. \\ &= \frac{C_1 q}{\tau} + \frac{2\sigma^2 k}{\tau \Delta} + \frac{\eta \|\beta^*\|^2}{\Delta} + C_4 e^{-c_5 \frac{q}{\tau} \Delta} \\ &= \frac{C_1 q}{T_0^{1/2-\gamma}} + \frac{2\sigma^2 k}{T_0} + \frac{\eta \|\beta^*\|^2}{T_0^{1/2+\gamma}} + C_4 e^{-c_5 q T_0^{2\gamma}} \\ &= \mathcal{O}(T_0^{-1/2+\gamma}). \end{aligned}$$

This concludes the proof. ∎

Appendix F. Forecasting Analysis: Post-Intervention Regime

We now bound the post-intervention ℓ_2 error of our estimator.

F.1 Proof of Theorem 6

Theorem 6 *Let Equation* (6) *hold for some* $\beta^* \in \mathbb{R}^{N-1}$. *Let* $\text{rank}(\boldsymbol{M}^-) = \text{rank}(\boldsymbol{M})$. *Then* $M_1^+ = (\boldsymbol{M}^+)^T\beta^*$.

Proof Suppose we begin with only the matrix $\boldsymbol{M}^-$, i.e. $\boldsymbol{M} = \boldsymbol{M}^-$. From the assumption that $M_1^- = (\boldsymbol{M}^-)^T\beta^*$, we have for $t \le T_0$

$$M_{1t} = \sum_{j=2}^{N} \beta_j^* M_{jt}.$$

Suppose that we now add an extra column to $\boldsymbol{M}^-$ so that $\boldsymbol{M}$ is of dimension $N \times (T_0+1)$. Since $\text{rank}(\boldsymbol{M}^-) = \text{rank}(\boldsymbol{M})$, we have for $j \in [N]$

$$M_{j,T_0+1} = \sum_{t=1}^{T_0} \pi_t M_{jt},$$

for some set of weights $\pi \in \mathbb{R}^{T_0}$. In particular, for the first row we have

$$\begin{aligned} M_{1,T_0+1} &= \sum_{t=1}^{T_0} \pi_t M_{1t} \\ &= \sum_{t=1}^{T_0} \pi_t \Big(\sum_{j=2}^{N} \beta_j^* M_{jt} \Big) \\ &= \sum_{j=2}^{N} \beta_j^* \Big(\sum_{t=1}^{T_0} \pi_t M_{jt} \Big) \\ &= \sum_{j=2}^{N} \beta_j^* M_{j,T_0+1}. \end{aligned}$$

By induction, we observe that for any number of columns added to $\boldsymbol{M}^-$ such that $\text{rank}(\boldsymbol{M}^-) = \text{rank}(\boldsymbol{M})$, we must have $M_1^+ = (\boldsymbol{M}^+)^T\beta^*$ where $\boldsymbol{M}^+ = [M_{it}]_{2\le i\le N, T_0<t\le T}$. ■

F.2 Proof of Theorem 7

Theorem 7 *Suppose* $p \ge \frac{T^{-1+\zeta}}{\sigma^2+1}$ *for some* $\zeta > 0$. *Suppose* $\left\|\hat{\beta}(\eta)\right\|_\infty \le \psi$ *for some* $\psi > 0$. *Let* $\alpha' T_0 \le T \le \alpha T_0$ *for some constants* $\alpha', \alpha > 1$. *Then for any* $\eta \ge 0$ *and using* μ *as defined in* (20), *the post-intervention error is bounded above by*

$$\text{RMSE}(\hat{M}_1^+) \le \frac{C_1}{\sqrt{p}}(\sigma^2 + (1-p))^{1/2} + \frac{C_2\|\boldsymbol{M}\|}{\sqrt{T_0}} \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| + \mathcal{O}(1/\sqrt{T_0}),$$

where C_1 *and* C_2 *are universal positive constants.*

Proof We will prove Theorem 7 by drawing upon techniques and results from prior proofs. We begin by applying triangle inequality to obtain

$$\begin{aligned} \left\|\hat{M}_1^+ - M_1^+\right\| &= \left\|(\hat{\boldsymbol{M}}^+)^T\hat{\beta}(\eta) - (\boldsymbol{M}^+)^T\beta^*\right\| \\ &= \left\|(\hat{\boldsymbol{M}}^+)^T\hat{\beta}(\eta) - (\boldsymbol{M}^+)^T\beta^* + (\boldsymbol{M}^+)^T\hat{\beta}(\eta) - (\boldsymbol{M}^+)^T\hat{\beta}(\eta)\right\| \\ &\le \left\|(\hat{\boldsymbol{M}}^+ - \boldsymbol{M}^+)^T\hat{\beta}(\eta)\right\| + \left\|(\boldsymbol{M}^+)^T(\hat{\beta}(\eta) - \beta^*)\right\|. \end{aligned}$$

Taking expectations and using the property of induced norms gives

$$\begin{aligned} \mathbb{E}\left\|\hat{M}_1^+ - M_1^+\right\| &\le \mathbb{E}\Big[\left\|\hat{\boldsymbol{M}}^+ - \boldsymbol{M}^+\right\| \cdot \left\|\hat{\beta}(\eta)\right\|\Big] + \|\boldsymbol{M}^+\| \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| \\ &\le \sqrt{N}\psi \cdot \mathbb{E}\left\|\hat{\boldsymbol{M}}^+ - \boldsymbol{M}^+\right\| + \|\boldsymbol{M}^+\| \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\|, \end{aligned} \tag{72}$$

where the last inequality uses the boundedness assumption of $\hat{\beta}(\eta)$. Observe that the first term on the right-hand side of (72) is similar to that of (53) and (63) with the main difference being (72) uses the post-intervention submatrices, $\hat{\boldsymbol{M}}^+$ and $\boldsymbol{M}^+$, as opposed to the pre-intervention submatrices, $\hat{\boldsymbol{M}}^-$ and $\boldsymbol{M}^-$, in (53) and (63). Therefore, using (57) and the arguments that led to (67), it follows that (with appropriate constants C_1, C_2, c_3)

$$\mathbb{E}\left\|\hat{\boldsymbol{M}}^+ - \boldsymbol{M}^+\right\| \le \frac{C_1}{p}\mathbb{E}\Big(\sqrt{Tq} + \left\|(\hat{p}-p)\boldsymbol{M}^+\right\|\Big) + C_2((N-1)T)^{3/2}e^{-c_3\phi T},$$

where the slight modification arises due to the fact that we are now operating in the post-intervention regime. In particular, $\|\boldsymbol{M}^+\| \le \sqrt{(N-1)(T-T_0)}$ and $\left\|\hat{\boldsymbol{M}}^+\right\| \le ((N-1)T)^{3/2}$. Further, note that q and ϕ are defined exactly as before, i.e. $q = \sigma^2 p + p(1-p)$ and $\phi = p(N-1) + \sigma^2 + q$. Following the proof of Corollary 4, we apply Jensen's Inequality to obtain

$$\begin{aligned} \mathbb{E}\left\|(\hat{p}-p)\boldsymbol{M}^+\right\| &= \mathbb{E}|\hat{p}-p| \cdot \left\|\boldsymbol{M}^+\right\| \\ &\le \sqrt{\frac{p(1-p)}{(N-1)T}} \cdot \sqrt{(N-1)(T-T_0)} \\ &\le \sqrt{p(1-p)}. \end{aligned}$$

Putting the above results together, we have (for appropriately defined constants)

$$\begin{aligned} \text{RMSE}(\hat{M}_1^+) &\le \frac{C_1\sqrt{N}\psi}{p\sqrt{T-T_0}}\Big(\sqrt{Tq} + \sqrt{p(1-p)}\Big) + \frac{\|\boldsymbol{M}^+\|}{\sqrt{T-T_0}} \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| + C_4 e^{-c_5\phi T} \\ &\overset{(a)}{\le} \frac{C_6\sqrt{q}}{p} + \frac{C_7\sqrt{1-p}}{\sqrt{pT_0}} + \frac{C_8\|\boldsymbol{M}\|}{\sqrt{T_0}} \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| + C_4 e^{-c_5\phi T} \\ &= \frac{C_6}{\sqrt{p}}(\sigma^2 + (1-p))^{1/2} + \frac{C_8\|\boldsymbol{M}\|}{\sqrt{T_0}} \cdot \mathbb{E}\left\|\hat{\beta}(\eta) - \beta^*\right\| + \mathcal{O}(1/\sqrt{T_0}), \end{aligned}$$

where (a) follows from Lemma 17. Renaming constants would provide the desired result. ■

Appendix G. A Bayesian Perspective

Derivation of posterior parameters.

The following is based on the derivation presented in Section 2.2.3 of Bishop (2006), and is presented here for completeness. Suppose we are given a multivariate Gaussian marginal distribution $p(x)$ paired with a multivariate Gaussian conditional distribution $p(y \mid x)$ – where x and y may have differing dimensions – and we are interested in computing the posterior distribution over x, i.e. $p(x \mid y)$. We will derive the posterior parameters of $p(x \mid y)$ here. Without loss of generality, suppose

$$\begin{aligned} p(x) &= \mathcal{N}(x \mid \mu, \boldsymbol{\Lambda}^{-1}) \\ p(y \mid x) &= \mathcal{N}(y \mid \boldsymbol{A}x + b, \boldsymbol{\Sigma}^{-1}), \end{aligned}$$

where $\mu, \boldsymbol{A}$, and b are parameters that govern the means, while $\boldsymbol{\Lambda}$ and $\boldsymbol{\Sigma}$ are precision (inverse covariance) matrices.

We begin by finding the joint distribution over x and y. Ignoring the terms that are independent of x and y and encapsulating them into the "const." expression, we obtain

$$\begin{aligned} \ln p(x, y) &= \ln p(x) + \ln p(y \mid x) \\ &= -\frac{1}{2}(x-\mu)^T \boldsymbol{\Lambda}(x-\mu) - \frac{1}{2}(y - \boldsymbol{A}x - b)^T \boldsymbol{\Sigma}(y - \boldsymbol{A}x - b) + \text{const.} \\ &= -\frac{1}{2}x^T(\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})x - \frac{1}{2}y^T\boldsymbol{\Sigma}y + \frac{1}{2}x^T\boldsymbol{A}^T\boldsymbol{\Sigma}y + \text{const.} \\ &= -\frac{1}{2}\begin{bmatrix} x \\ y \end{bmatrix}^T \begin{bmatrix} \boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} & -\boldsymbol{A}^T\boldsymbol{\Sigma} \\ -\boldsymbol{\Sigma}\boldsymbol{A} & \boldsymbol{\Sigma} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \text{const.} \\ &= -\frac{1}{2}z^T\boldsymbol{Q}z + \text{const.}, \end{aligned}$$

where $z = [x, y]^T$, and

$$\boldsymbol{Q} = \begin{bmatrix} \boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} & -\boldsymbol{A}^T\boldsymbol{\Sigma} \\ -\boldsymbol{\Sigma}\boldsymbol{A} & \boldsymbol{\Sigma} \end{bmatrix}$$

is the precision matrix. Applying the matrix inversion formula, we have that the covariance matrix of z is

$$\text{Var}(z) = \boldsymbol{Q}^{-1} = \begin{bmatrix} \boldsymbol{\Lambda}^{-1} & \boldsymbol{\Lambda}^{-1}\boldsymbol{A}^T \\ \boldsymbol{A}\boldsymbol{\Lambda}^{-1} & \boldsymbol{\Sigma}^{-1} + \boldsymbol{A}\boldsymbol{\Lambda}^{-1}\boldsymbol{A}^T \end{bmatrix}.$$

After collecting the linear terms over z, we find that the mean of the Gaussian distribution over z is defined as

$$\mathbb{E}[z] = \boldsymbol{Q}^{-1}\begin{bmatrix} \boldsymbol{\Lambda}\mu - \boldsymbol{A}^T\boldsymbol{\Sigma}b \\ \boldsymbol{\Sigma}b \end{bmatrix}.$$

Now that we have the parameters over the joint distribution of x and y, we find that the posterior distribution parameters over x are

$$\begin{aligned} \mathbb{E}[x \mid y] &= (\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}\{\boldsymbol{A}^T\boldsymbol{\Sigma}(y-b) + \boldsymbol{\Lambda}\mu\} \\ \text{Var}(x \mid y) &= (\boldsymbol{\Lambda} + \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A})^{-1}. \end{aligned}$$

Journal of Machine Learning Research 19 (2018) 1-39 Submitted 12/17; Published 8/18

Reverse Iterative Volume Sampling for Linear Regression*

Michał Dereziński MDEREZIN@UCSC.EDU
Manfred K. Warmuth MANFRED@UCSC.EDU
Department of Computer Science
University of California Santa Cruz

Editor: Michael Mahoney

Abstract

We study the following basic machine learning task: Given a fixed set of input points in $\mathbb{R}^d$ for a linear regression problem, we wish to predict a hidden response value for each of the points. We can only afford to attain the responses for a small subset of the points that are then used to construct linear predictions for all points in the dataset. The performance of the predictions is evaluated by the total square loss on all responses (the attained as well as the remaining hidden ones). We show that a good approximate solution to this least squares problem can be obtained from just dimension d many responses by using a joint sampling technique called volume sampling. Moreover, the least squares solution obtained for the volume sampled subproblem is an unbiased estimator of optimal solution based on all n responses. This unbiasedness is a desirable property that is not shared by other common subset selection techniques.

Motivated by these basic properties, we develop a theoretical framework for studying volume sampling, resulting in a number of new matrix expectation equalities and statistical guarantees which are of importance not only to least squares regression but also to numerical linear algebra in general. Our methods also lead to a regularized variant of volume sampling, and we propose the first efficient algorithm for volume sampling which makes this technique a practical tool in the machine learning toolbox. Finally, we provide experimental evidence which confirms our theoretical findings.

Keywords: volume sampling, linear regression, row sampling, active learning, optimal design

1. Introduction

As an introductory case, consider linear regression in one dimension. We are given n non-zero points x_i. Each point has a hidden real response (or target value) y_i. Assume that obtaining the responses is expensive and the learner can afford to request the responses y_i for only a small number of indices i. After receiving the requested responses, the learner determines an approximate linear least squares solution. In the one dimensional homogeneous case, this is just a single weight. How many response values does the learner need to request so that the total square loss of its approximate solution on all n points is "close" to the total loss of the optimal linear least squares solution found with the knowledge of all responses? We will show here that just *one* response suffices if the index i is chosen proportional to x_i^2. When the learner uses the approximate solution $w_i^* = \frac{y_i}{x_i}$, then its expected loss equals 2 times the loss of the optimum w^* that is computed based on all responses (See Figure 1). Moreover, the approximate solution w_i^* is an unbiased estimator for the optimum w^*:

$$\mathbb{E}_i\left[\sum_j (x_j \frac{y_i}{x_i} - y_j)^2\right] = 2\sum_j (x_j w^* - y_j)^2 \quad \text{and} \quad \mathbb{E}_i\left[\frac{y_i}{x_i}\right] = w^*, \quad \text{when } P(i) \sim x_i^2.$$

*. This paper is an expanded version of two conference papers (Dereziński and Warmuth, 2017, 2018).

We will extend these formulas to higher dimensions and to sampling more responses by making use of a joint sampling distribution called *volume sampling*. We next break down our contributions into four parts.

Least squares with dimension many responses. Consider the case when the points $\mathbf{x}_i$ lie in $\mathbb{R}^d$. Let $\mathbf{X}$ denote a full rank $n \times d$ matrix that has the n transposed points $\mathbf{x}_i^\top$ as rows, and let $\mathbf{y} \in \mathbb{R}^n$ be the vector of responses. Now the goal is to minimize the (total) square loss,

$$L(\mathbf{w}) = \sum_{i=1}^n (\mathbf{x}_i^\top \mathbf{w} - y_i)^2 = \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2,$$

over all linear weight vectors $\mathbf{w} \in \mathbb{R}^d$. Let $\mathbf{w}^*$ denote the optimal such weight vector. We want to minimize the square loss based on a small number of responses we attained for a subset of rows. Again, the learner is initially given the fixed set of n rows (i.e. fixed design), but none of the responses. It is then allowed to choose a random subset of d indices, $S \subseteq \{1..n\}$, and obtains the responses for the corresponding d rows. The learner proceeds to find the optimal linear least squares solution $\mathbf{w}^*(S)$ for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. where $\mathbf{X}_S$ is the subset of d rows of $\mathbf{X}$ indexed by S and $\mathbf{y}_S$ the corresponding d responses from the response vector $\mathbf{y}$. As a generalization of the one-dimensional distribution that chooses an index based on the squared length, set S of size d is chosen proportional to the squared volume of the parallelepiped spanned by the rows of $\mathbf{X}_S$. This squared volume equals $\det(\mathbf{X}_S^\top \mathbf{X}_S)$. Using elementary linear algebra, we will show that *volume sampling* the set S assures that $\mathbf{w}^*(S)$ is a good approximation to $\mathbf{w}^*$ in the following sense: In expectation, the square loss (on all n row response pairs) of $\mathbf{w}^*(S)$ is equal $d+1$ times the square loss of $\mathbf{w}^*$ (when $\mathbf{X}$ is in general position):

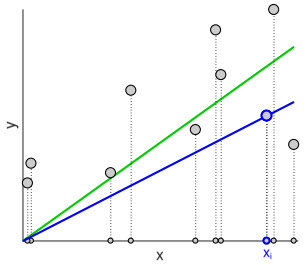


Figure 1: The expected loss of $w_i^* = \frac{y_i}{x_i}$ (blue line) based on one response y_i is twice the loss of the optimum w^* (green line).

$$\mathbb{E}[L(\mathbf{w}^*(S))] = (d+1)\, L(\mathbf{w}^*), \quad \text{when } P(S) \sim \det(\mathbf{X}_S^\top \mathbf{X}_S).$$

Furthermore, we will show that for any sampling procedure that attains less than d responses, the ratio between the expected loss and the loss of the optimum cannot be bounded by a constant.

Unbiased pseudoinverse estimator. There is a direct connection between solving linear least squares problems and the pseudoinverse $\mathbf{X}^+$ of matrix $\mathbf{X}$: For an $n-$dimensional response vector $\mathbf{y}$, the optimal solution is $\mathbf{w}^* = \text{argmin}_{\mathbf{w}} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 = \mathbf{X}^+\mathbf{y}$. Similarly $\mathbf{w}^*(S) = (\mathbf{X}_S)^+\mathbf{y}_S$ is the solution for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. We propose a new implementation of volume sampling called *reverse iterative sampling* which enables a novel proof technique for obtaining elementary expectation formulas for pseudoinverses based on volume sampling.

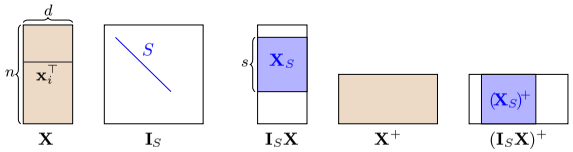


Figure 2: Shapes of the matrices. The indices of S may not be consecutive.

Suppose that our goal is to estimate the pseudoinverse $\mathbf{X}^+$ based on the pseudoinverse of a subset of rows. Recall that for a subset $S \subseteq \{1..n\}$ of s row indices (where the size s is fixed and $s \geq d$), we let $\mathbf{X}_S$ be the submatrix of the s rows indexed by S (see Figure 2). Consider a version of $\mathbf{X}$ in which all but the rows of S are zero. This matrix equals $\mathbf{I}_S\mathbf{X}$, where the selection matrix $\mathbf{I}_S$ is an n-dimensional diagonal matrix with $(\mathbf{I}_S)_{ii} = 1$ if $i \in S$ and 0 otherwise.

For the set S of fixed size $s \geq d$ row indices chosen proportional to $\det(\mathbf{X}_S^\top \mathbf{X}_S)$, we can prove the following two expectation formulas (for the second equality, $\mathbf{X}$ must be in general position):

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+ \quad \text{and} \quad \mathbb{E}\Big[\underbrace{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}_{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}\Big] = \frac{n-d+1}{s-d+1}\underbrace{(\mathbf{X}^\top\mathbf{X})^{-1}}_{\mathbf{X}^+\mathbf{X}^{+\top}}.$$

Note that $(\mathbf{I}_S\mathbf{X})^+$ has the $d \times n$ shape of $\mathbf{X}^+$ where the s columns indexed by S contain $(\mathbf{X}_S)^+$ and the remaining $n-s$ columns are zero. The expectation of this matrix is $\mathbf{X}^+$ even though $(\mathbf{X}_S)^+$ is clearly not a submatrix of $\mathbf{X}^+$. This expectation formula now implies that for any size $s \geq d$, if S of size s is drawn by volume sampling, then $\mathbf{w}^*(S)$ is an unbiased estimator[1] for $\mathbf{w}^*$, i.e.

$$\mathbb{E}[\mathbf{w}^*(S)] = \mathbb{E}[(\mathbf{X}_S)^+\mathbf{y}_S] = \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+\mathbf{y}] = \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\,\mathbf{y} = \mathbf{X}^+\mathbf{y} = \mathbf{w}^*.$$

The second expectation formula can be viewed as a second moment of the pseudoinverse estimator $(\mathbf{I}_S\mathbf{X})^+$, and it can be used to compute a useful notion of matrix variance with applications in random matrix theory:

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}] - \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]^\top = \frac{n-s}{s-d+1}\mathbf{X}^+\mathbf{X}^{+\top}.$$

Regularized volume sampling. We also develop a new regularized variant of volume sampling, which extends reverse iterative sampling to selecting subsets of size smaller than d, and leads to a useful extension of the above matrix variance formula. Namely, for any $\lambda \geq 0$, our λ-regularized procedure for sampling subsets S of size s satisfies

$$\mathbb{E}\big[(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}\big] \preceq \frac{n-d_\lambda+1}{s-d_\lambda+1}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1},$$

where $d_\lambda \overset{def}{=} \operatorname{tr}(\mathbf{X}(\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^\top) \leq d$ is a standard notion of statistical dimension. Crucially, the above bound holds for subset sizes $s \geq d_\lambda$, which can be much smaller than the dimension d.

Under the additional assumption that response vector $\mathbf{y}$ is generated by a linear transformation distorted with bounded white noise, the expected bound on $(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}$ leads to strong variance bounds for the ridge regression estimator. Specifically, we prove that when $\mathbf{y} = \mathbf{X}\widetilde{\mathbf{w}} + \boldsymbol{\xi}$, with $\boldsymbol{\xi}$ having mean zero and bounded variance $\operatorname{Var}[\boldsymbol{\xi}] \preceq \sigma^2\mathbf{I}$, then if S is sampled according to λ-regularized volume sampling with $\lambda \leq \frac{\sigma^2}{\|\widetilde{\mathbf{w}}\|^2}$, we can obtain the following bound on the mean squared prediction error (MSPE):

$$\mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\left[\frac{1}{n}\|\mathbf{X}(\mathbf{w}_\lambda^*(S) - \widetilde{\mathbf{w}})\|^2\right] \leq \frac{\sigma^2 d_\lambda}{s-d_\lambda+1},$$

1. For size $s = d$ volume sampling, the fact that $\mathbb{E}[\mathbf{w}^*(S)] = \mathbf{w}^*$ can be found in an early paper (Ben-Tal and Teboulle, 1990). They give a direct proof based on Cramer's rule.

where $\mathbf{w}_\lambda^*(S) = (\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}\mathbf{X}_S^\top\mathbf{y}_S$ is the ridge regression estimator for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. Our new lower bounds show that the above upper bound for regularized volume sampling is essentially optimal with respect to the choice of a subsampling procedure.

Algorithms and experiments. The only known polynomial time algorithm for size $s > d$ volume sampling was recently proposed by Li et al. (2017) with time complexity $O(n^4 s)$. In this paper we give two new algorithms using our general framework of reverse iterative sampling: one with deterministic runtime of $O((n-s+d)nd)$, and a second one that with high probability finishes in time $O(nd^2)$. Thus both algorithms improve on the state-of-the-art by a factor of at least n^2 and make volume sampling nearly as efficient as the comparable i.i.d. sampling technique called leverage score sampling. Our experiments on real datasets confirm the efficiency of our algorithms and show that for small sample sizes s, volume sampling is more effective than leverage score sampling for the task of subset selection for linear regression.

1.1 Related Work

Volume sampling is a type of determinantal point process (DPP) (Kulesza and Taskar, 2012). DPP's have been given a lot of attention in the literature with many applications to machine learning, including recommendation systems (Gartrell et al., 2016) and clustering (Kang, 2013). Many exact and approximate methods for efficiently generating samples from this distribution have been proposed (Deshpande and Rademacher, 2010; Kulesza and Taskar, 2011), making it a useful tool in the design of randomized algorithms. Most of those methods focus on sampling $s \leq d$ elements. In this paper, we study volume sampling sets of size $s \geq d$, which was proposed by Avron and Boutsidis (2013) and motivated with applications in graph theory, linear regression, matrix approximation and more.

The problem of selecting a subset of the rows of the input matrix for solving a linear regression task has been extensively studied in statistics literature under the terms *optimal design* (Fedorov, 1972) and *pool-based active learning* (Sugiyama and Nakajima, 2009). Various criteria for subset selection have been proposed, like A-optimality and D-optimality. For example, A-optimality seeks to minimize $\operatorname{tr}((\mathbf{X}_S^\top\mathbf{X}_S)^{-1})$, which is combinatorially hard to optimize exactly. We show that for size $s \geq d$ volume sampling, $\mathbb{E}[(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}] = \frac{n-d+1}{s-d+1}(\mathbf{X}^\top\mathbf{X})^{-1}$, which provides an approximate randomized solution of the sampled inverse covariance matrix rather than just its trace.

In the field of computational geometry a variant of volume sampling was used to obtain optimal bounds for low-rank matrix approximation. In this task, the goal is to select a small subset of rows of a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ (much fewer than the rank of $\mathbf{X}$, which is bounded by d), so that a good low-rank approximation of $\mathbf{X}$ can be constructed from those rows. Deshpande et al. (2006) showed that volume sampling of size $s < d$ index sets obtains optimal multiplicative bounds for this task and polynomial time algorithms for size $s < d$ volume sampling were given in Deshpande and Rademacher (2010) and Guruswami and Sinop (2012). We show in this paper that for linear regression, fewer than rank many rows do not suffice to obtain multiplicative bounds. This is why we focus on volume sampling sets of size $s \geq d$ (recall that, for simplicity, we assume that $\mathbf{X}$ is full rank).

Computing approximate solutions to linear regression has been explored in the domain of numerical linear algebra (see Mahoney (2011) for an overview). Here, multiplicative bounds on the loss of the approximate solution can be achieved via two approaches. The first approach relies on sketching the input matrix $\mathbf{X}$ and the response vector $\mathbf{y}$ by multiplying both by the same suitably chosen random matrix. Algorithms which use sketching to generate a smaller input matrix for a given linear regression problem are computationally efficient (Sarlos, 2006; Clarkson and Woodruff,

2013), but they require all of the responses from the original problem to generate the sketch and are thus not suitable for the goal of using as few response values as possible. The second approach is based on subsampling the rows of the input matrix and only asking for the responses of the sampled rows. The learner optimally solves the sampled subproblem[2] and then uses the obtained weight vector for its prediction on all rows. The selected subproblem is known under the term "$\mathbf{b}$-agnostic minimal coreset" in (Boutsidis et al., 2013; Drineas et al., 2008) since it is selected without knowing the response vector (denoted as the vector $\mathbf{b}$). The second approach coincides with the goals of this paper but the focus here is different in a number of ways. First, we focus on the smallest sample size for which a multiplicative loss bound is possible: Just d volume sampled rows are sufficient to achieve a multiplicative bound with a fixed factor, while $d-1$ are not sufficient. A second focus here is the efficiency and the combinatorics of volume sampling. The previous work is mostly based on i.i.d. sampling using the statistical leverage scores (Drineas et al., 2012). As we show in this paper, leverage scores are the marginals of volume sampling and any i.i.d. sampling method requires sample size $\Omega(d \log d)$ to achieve multiplicative loss bounds for linear regression. On the other hand, the rows obtained from volume sampling are *selected jointly* and this makes the chosen subset more informative and brings the required sample size down to d. Third, we focus on the fact that the estimators produced from volume sampling are unbiased and therefore can be averaged to get more accurate estimators. Using our methods, averaging immediately leads to an unbiased estimator with expected loss $1+\epsilon$ times the optimum based on sampling d^2/ϵ responses in total. We leave it as an open problem to construct a $1+\epsilon$ factor unbiased estimator from sampling only $O(d/\epsilon)$ responses. If unbiasedness is not a concern, then such an estimator has recently been found (Chen and Price, 2017).

1.2 Outline of the Paper

In the next section, we define volume sampling as an instance of a more general procedure we call reverse iterative sampling, and we use this methodology to prove closed form matrix expressions for the expectation of the pseudoinverse estimator $(\mathbf{I}_S\mathbf{X})^+$ and its square $(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}$, when S is sampled by volume sampling. Central to volume sampling is the Cauchy-Binet formula for determinants. As a side, we produce a number of short self-contained proofs for this formula and show that leverage scores are the marginals of volume sampling. Then in Section 3 we formulate the problem of solving linear regression from a small number of responses, and state the upper bound for the expected square loss of the volume sampled least squares estimator (Theorem 8), followed by a discussion and related lower bounds. In Section 3.3, we prove Theorem 8 and an additional related matrix expectation formula. We next discuss in Section 3.4 how unbiased estimators can easily be averaged for improving the expected loss and discuss open problems for constructing unbiased estimators. A new regularized variant of volume sampling is proposed in Section 4, along with the statistical guarantees it offers for computing subsampled ridge regression estimators. Next, we present efficient volume sampling algorithms in Section 5, based on the reverse iterative sampling paradigm, which are then experimentally evaluated in Section 6. Finally, Section 7 concludes the paper by suggesting a future research direction.

2. Note that those methods typically require additional rescaling of the subproblem, whereas the techniques proposed in this paper do not require any rescaling.

2. Reverse Iterative Sampling

Let n be an integer dimension. For each subset $S \subseteq \{1..n\}$ of size s we are given a matrix formula $\mathbf{F}(S)$. Our goal is to sample set S of size s using some sampling process and then develop concise expressions for $\mathbb{E}_{S:|S|=s}[\mathbf{F}(S)]$. Examples of formula classes $\mathbf{F}(S)$ will be given below.

We represent the sampling by a directed acyclic graph (DAG), with a single root node corresponding to the full set $\{1..n\}$. Starting from the root, we proceed along the edges of the graph, iteratively removing elements from the set S (see Figure 3). Concretely, consider a DAG with levels $s = n, n-1, ..., d$. Level s contains $\binom{n}{s}$ nodes for sets $S \subseteq \{1..n\}$ of size s. Every node S at level $s > d$ has s directed edges to the nodes $S - \{i\}$ (also denoted S_{-i}) at the next lower level. These edges are labeled with a conditional probability vector $P(S_{-i}|S)$, where the event S occurs if the sampling process visits node S as it traces a (directed) path in the DAG from the root node $\{1..n\}$ to a node at level d. Such paths have $n-d$ edges. It is natural to assign probabilities to shorter paths as well going from any node to a node at a lower level. The probability of such a path is again the product of its edge probabilities. It also follows that the probability $P(S)$ of visiting node S (via a path from the root) is the sum of the probabilities of all paths from root to S. Finally, the probability $P(\{1..n\})$ of the root node is 1 and more generally, the total probability of all nodes at each layer is 1.

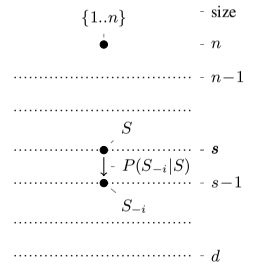


Figure 3: Reverse iterative sampling.

We associate a formula $\mathbf{F}(S)$ with each set node S in the DAG. The following key equality lets us compute expectations.

Lemma 1 *If for all $S \subseteq \{1..n\}$ of size greater than d we have*

$$\mathbf{F}(S) = \sum_{i \in S} P(S_{-i}|S)\mathbf{F}(S_{-i}),$$

then for any $s \in \{d..n\}$: $\mathbb{E}_{S:|S|=s}[\mathbf{F}(S)] = \sum_{S:|S|=s} P(S)\mathbf{F}(S) = \mathbf{F}(\{1..n\})$.

Proof Suffices to show that expectations at successive layers s and $s-1$ are equal for $s > d$:

$$\sum_{S:|S|=s} P(S)\,\mathbf{F}(S) = \sum_{S:|S|=s} P(S) \sum_{i \in S} P(S_{-i}|S)\,\mathbf{F}(S_{-i}) = \sum_{S:|S|=s} \sum_{i \in S} P(S)P(S_{-i}|S)\mathbf{F}(S_{-i})$$
$$= \sum_{T:|T|=s-1} \underbrace{\sum_{j \notin T} P(T_{+j})P(T|T_{+j})}_{P(T)}\,\mathbf{F}(T).$$

Note that the r.h.s. of the first line has one summand per edge leaving level s, and the r.h.s. of the second line has one summand per edge arriving at level $s-1$. Now the last equality holds because the edges leaving level s are exactly those arriving at level $s-1$, and the summand for each edge in both expressions is equivalent. ■

2.1 Volume Sampling

Given a tall full rank matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ and a sample size $s \in \{d..n\}$, volume sampling chooses subset $S \subseteq \{1..n\}$ of size s with probability proportional to squared volume spanned by the columns of submatrix[3] $\mathbf{X}_S$ and this squared volume equals $\det(\mathbf{X}_S^\top \mathbf{X}_S)$. The following theorem uses the above DAG setup to compute the normalization constant for this distribution. Note that all subsets S of volume 0 will be ignored, since they are unreachable in the proposed sampling procedure.

Theorem 2 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$, where $d \le n$ and $\det(\mathbf{X}^\top\mathbf{X}) > 0$. For any set S of size $s > d$ for which $\det(\mathbf{X}_S^\top\mathbf{X}_S) > 0$, define the probability of the edge from S to S_{-i} for $i \in S$ as:*

$$P(S_{-i}|S) \overset{def}{=} \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}})}{(s-d)\det(\mathbf{X}_S^\top\mathbf{X}_S)} = \frac{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d}, \quad \textbf{(reverse iterative volume sampling)}$$

where $\mathbf{x}_i^\top$ is the i-th row of $\mathbf{X}$. In this case $P(S_{-i}|S)$ is a proper probability distribution. If $\det(\mathbf{X}_S^\top\mathbf{X}_S) = 0$, then simply set $P(S_{-i}|S)$ to $\frac{1}{s}$. With these definitions, $\sum_{S:|S|=s} P(S) = 1$ for all $s \in \{d..n\}$ and the probability of all paths from the root to any subset S of size at least d is

$$P(S) = \frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X})}. \qquad \textbf{(volume sampling)}$$

The rewrite of the ratio $\frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}})}{\det(\mathbf{X}_S^\top\mathbf{X}_S)}$ as $1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i$ is Sylvester's Theorem for determinants. Incidentally, this is the only property of determinants used in this section.

The theorem also implies a generalization of the Cauchy-Binet formula to size $s \ge d$ sets:

$$\sum_{S:|S|=s} \det(\mathbf{X}_S^\top\mathbf{X}_S) = \binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X}). \tag{1}$$

When $s = d$, then the binomial coefficient is 1 and the above becomes the vanilla Cauchy-Binet formula. The below proof of the theorem thus results in a minimalist proof of this classical formula as well. The proof uses the reverse iterative sampling (Figure 3) and the fact that all paths from the root to node S have the same probability. For the sake of completeness we also give a more direct inductive proof of the above generalized Cauchy-Binet formula in Appendix A.

Proof First, for any node S s.t. $s > d$ and $\det(\mathbf{X}_S^\top\mathbf{X}_S) > 0$, the probabilities out of S sum to 1:

$$\sum_{i\in S} P(S_{-i}|S) = \sum_{i\in S} \frac{1-\mathrm{tr}((\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top)}{s-d} = \frac{s-\mathrm{tr}((\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{X}_S^\top\mathbf{X}_S)}{s-d} = \frac{s-d}{s-d} = 1.$$

It remains to show the formula for the probability $P(S)$ of all paths ending at node S. If $\det(\mathbf{X}_S^\top\mathbf{X}_S) = 0$, then one edge on any path from the root to S has probability 0. This edge goes from a superset of S with positive volume to a superset of S that has volume 0. Since all paths have probability 0, $P(S) = 0$ in this case.

Now assume $\det(\mathbf{X}_S^\top\mathbf{X}_S) > 0$ and consider any path from the root $\{1..n\}$ to S. There are $(n-s)!$ such paths all going through sets with positive volume. The fractions of determinants in the probabilities along each path telescope and the additional factors accumulate to the same product. So the probability of all paths from the root to S is the same and the total probability into S is

$$\frac{(n-s)!}{(n-d)\dots(s-d+1)}\frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\det(\mathbf{X}^\top\mathbf{X})} = \frac{1}{\binom{n-d}{s-d}}\frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\det(\mathbf{X}^\top\mathbf{X})}. \qquad \blacksquare$$

An immediate consequence of the above sampling procedure is the following composition property of volume sampling, which states that this distribution is closed under subsampling. We also give a direct proof to highlight the combinatorics of volume sampling.

Corollary 3 *For any $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $n \ge t > s \ge d$, the following hierarchical sampling procedure:*

$$T \overset{t}{\sim} \mathbf{X} \qquad \textit{(size } t \textit{ volume sampling from } \mathbf{X}\textit{)},$$
$$S \overset{s}{\sim} \mathbf{X}_T \qquad \textit{(size } s \textit{ volume sampling from } \mathbf{X}_T\textit{)}$$

returns a set S which is distributed according to size s volume sampling from $\mathbf{X}$.

Proof We start with the Law of Total Probability and then use the probability formula for volume sampling from the above theorem. Here $P(T \cap S)$ means the probability of all paths going through node T at level t and ending up at the final node S at level s. If $S \not\subseteq T$, then $P(T \cap S) = 0$.

$$\begin{aligned} P(S) &= \sum_{T:\, S\subseteq T} \overbrace{P(S\,|\,T)\qquad P(T)}^{P(T\cap S)} \\ &= \sum_{T:\, S\subseteq T} \frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\binom{t-d}{s-d}\cancel{\det(\mathbf{X}_T^\top\mathbf{X}_T)}} \quad \frac{\cancel{\det(\mathbf{X}_T^\top\mathbf{X}_T)}}{\binom{n-d}{t-d}\det(\mathbf{X}^\top\mathbf{X})} \\ &= \binom{n-s}{t-s}\frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\binom{t-d}{s-d}\binom{n-d}{t-d}\det(\mathbf{X}^\top\mathbf{X})} = \frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X})}. \end{aligned}$$

Note that for all sets T containing S, the probability $P(T \cap S)$ is the same, and there are $\binom{n-s}{t-s}$ such sets. $\blacksquare$

The main competitor of volume sampling is i.i.d. sampling of the rows of $\mathbf{X}$ w.r.t. the statistical leverage scores. For an input matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, the leverage score of the i-th row $\mathbf{x}_i^\top$ of $\mathbf{X}$ is defined as

$$l_i \overset{def}{=} \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i.$$

Recall that this quantity appeared in the definition of conditional probability $P(S_{-i}|S)$ in Theorem 2, where the leverage score was computed w.r.t. the submatrix $\mathbf{X}_S$. In fact, there is a more basic relationship between leverage scores and volume sampling: If set S is sampled according to size $s = d$ volume sampling, then the leverage score l_i of row i is the marginal probability $P(i \in S)$ of selecting i-th row into S. A general formula for the marginals of size s volume sampling is given in the following proposition:

3. For sample size $s = d$, the rows and columns of $\mathbf{X}_S$ have the same length and $\det(\mathbf{X}_S^\top\mathbf{X}_S)$ is also the squared volume spanned by the rows $\mathbf{X}_S$.

Proposition 4 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a full rank matrix and $s \in \{d..n\}$. If $S \subseteq \{1..n\}$ is sampled according to size s volume sampling, then for any $i \in \{1..n\}$,*

$$P(i \in S) = \frac{s-d}{n-d} + \frac{n-s}{n-d}\overbrace{\mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i}^{l_i}.$$

Proof Instead of $P(i \in S)$ we will first compute $P(i \notin S)$:

$$\begin{aligned}
P(i \notin S) &= \sum_{S:|S|=s, i\notin S} \frac{\det(\mathbf{X}_S^\top\mathbf{X}_S)}{\binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X})} \\
&= \sum_{S:|S|=s, i\notin S} \frac{\sum_{T\subseteq S:|T|=d}\det(\mathbf{X}_T^\top\mathbf{X}_T)}{\binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X})} \\
&= \frac{\overbrace{\sum_{T:|T|=d, i\notin T}\det((\mathbf{X}_{-i})_T^\top(\mathbf{X}_{-i})_T)}^{\det(\mathbf{X}_{-i}^\top\mathbf{X}_{-i})}}{\binom{n-d}{s-d}\det(\mathbf{X}^\top\mathbf{X})}\ \binom{n-d-1}{s-d} \\
&= \frac{n-s}{n-d}\left(1 - \mathbf{x}_i^\top(\mathbf{X}^\top\mathbf{X})^{-1}\mathbf{x}_i\right),
\end{aligned}$$

where we used Cauchy-Binet twice and the fact that every set $T : |T| = d,\ i \notin T$ appears in $\binom{n-d-1}{s-d}$ sets $S : |S| = s,\ i \notin S$. Now, the marginal probability follows from the fact that $P(i \in S) = 1 - P(i \notin S)$. ■

2.2 Expectation Formulas for Volume Sampling

All expectations in the remainder of the paper are w.r.t. volume sampling. We use the short-hand $\mathbb{E}[\mathbf{F}(S)]$ for expectation with volume sampling where the size of the sampled set is fixed to s. The expectation formulas for two choices of $\mathbf{F}(S)$ are proven in Theorems 5 and 6. By Lemma 1 it suffices to show $\mathbf{F}(S) = \sum_{i\in S} P(S_{-i}|S)\mathbf{F}(S_{-i})$ for volume sampling. We also present a related expectation formula (Theorem 7), which is proven later using different techniques.

Recall that $\mathbf{X}_S$ is the submatrix of rows indexed by $S \subseteq \{1..n\}$. We also use a version of $\mathbf{X}$ in which all but the rows of S are zeroed out. This matrix equals $\mathbf{I}_S\mathbf{X}$ where $\mathbf{I}_S$ is an n-dimensional diagonal matrix with $(\mathbf{I}_S)_{ii} = 1$ if $i \in S$ and 0 otherwise (see Figure 2).

Theorem 5 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a tall full rank matrix (i.e. $n \geq d$). For $s \in \{d..n\}$, let $S \subseteq \{1..n\}$ be a size s volume sampled set over $\mathbf{X}$. Then*

$$\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}^+.$$

For the special case of $s = d$, this fact was known in the linear algebra literature (Ben-Tal and Teboulle, 1990; Ben-Israel, 1992). It was shown there using elementary properties of the determinant such as Cramer's rule.[4] The proof methodology developed here based on reverse iterative volume sampling is very different. We believe that this fundamental formula lies at the core of why volume sampling is important in many applications. In this work, we focus on its application to linear regression. However, Avron and Boutsidis (2013) discuss many problems where controlling the pseudoinverse of a submatrix is essential. For those applications, it is important to establish variance bounds for the above expectation and volume sampling once again offers very concrete guarantees. We obtain them by showing the following formula, which can be viewed as a second moment for this estimator.

4. Using the composition property of volume sampling (Corollary 3), the $s > d$ case of the theorem can be reduced to the $s = d$ case. However, in this paper we give a different self-contained proof for Theorem 5.

Theorem 6 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a full rank matrix and $s \in \{d..n\}$. If size s volume sampling over $\mathbf{X}$ has full support, then*

$$\mathbb{E}[\ \underbrace{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}_{(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}}\ ] = \frac{n-d+1}{s-d+1}\underbrace{(\mathbf{X}^\top\mathbf{X})^{-1}}_{\mathbf{X}^+\mathbf{X}^{+\top}}.$$

In the case when volume sampling does not have full support, then the matrix equality "=" above is replaced by the positive-definite inequality "$\preceq$".

The condition that size s volume sampling over $\mathbf{X}$ has full support is equivalent to $\det(\mathbf{X}_S^\top\mathbf{X}_S) > 0$ for all $S \subseteq \{1..n\}$ of size s. Note that if size s volume sampling has full support, then size $t > s$ also has full support. So full support for the smallest size d (often phrased as $\mathbf{X}$ being *in general position*) implies that volume sampling w.r.t. any size $s \geq d$ has full support.

The above theorem immediately gives an expectation formula for the Frobenius norm $\|(\mathbf{I}_S\mathbf{X})^+\|_F$ of the estimator:

$$\mathbb{E}\left[\|(\mathbf{I}_S\mathbf{X})^+\|_F^2\right] = \mathbb{E}[\mathrm{tr}((\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top})] = \frac{n-d+1}{s-d+1}\|\mathbf{X}^+\|_F^2. \tag{2}$$

This norm formula was shown by Avron and Boutsidis (2013), with numerous applications. Theorem 6 can be viewed as a much stronger pre-trace version of the known norm formula. Also our proof techniques are quite different and much simpler. Note that if size s volume sampling for $\mathbf{X}$ does not have full support, then (2) becomes an inequality.

We now mention a second application of the above theorem in the context of linear regression for the case when the response vector $\mathbf{y}$ is modeled as a noisy linear transformation, i.e., $\mathbf{y} = \mathbf{X}\widetilde{\mathbf{w}} + \boldsymbol{\xi}$ for some $\widetilde{\mathbf{w}} \in \mathbb{R}^d$ and a random noise vector $\boldsymbol{\xi} \in \mathbb{R}^n$ (detailed discussion in Section 4). In this case the matrix $(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}$ can be interpreted as the covariance matrix of least-squares estimator $\mathbf{w}^*(S)$ (for a fixed set S) and Theorem 6 gives an exact formula for the covariance matrix of $\mathbf{w}^*(S)$ under volume sampling. In Section 4, we give an extended version of this result which provides even stronger guarantees for regularized least-squares estimators under this model (Theorem 16).

Note that except for the above application, all results in this paper hold for arbitrary response vectors $\mathbf{y}$. By combining Theorems 5 and 6, we can also obtain a covariance-type formula[5] for the pseudoinverse matrix estimator:

$$\begin{aligned}
&\mathbb{E}[((\mathbf{I}_S\mathbf{X})^+ - \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+])\ ((\mathbf{I}_S\mathbf{X})^+ - \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+])^\top] \\
&= \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+(\mathbf{I}_S\mathbf{X})^{+\top}] - \mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]\,\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+]^\top \\
&= \frac{n-d+1}{s-d+1}\,\mathbf{X}^+\mathbf{X}^{+\top} - \mathbf{X}^+\mathbf{X}^{+\top} = \frac{n-s}{s-d+1}\,\mathbf{X}^+\mathbf{X}^{+\top}.
\end{aligned} \tag{3}$$

5. This notion of "covariance" is used in random matrix theory, i.e. for a random matrix $\mathbf{M}$, we analyze $\mathbb{E}[(\mathbf{M} - \mathbb{E}[\mathbf{M}])(\mathbf{M} - \mathbb{E}[\mathbf{M}])^\top]$. See for example Tropp (2012).

We now give the background for a third matrix expectation formula for volume sampling. Pseudoinverses can be used to compute the projection matrix onto the span of columns of matrix $\mathbf{X}$, which is defined as follows:

$$\mathbf{P}_{\mathbf{X}} \stackrel{def}{=} \mathbf{X} \overbrace{(\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top}^{\mathbf{X}^+}.$$

Applying Theorem 5 leads us immediately to the following unbiased matrix estimator for the projection matrix:

$$\mathbb{E}[\mathbf{X}(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}\,\mathbb{E}[(\mathbf{I}_S\mathbf{X})^+] = \mathbf{X}\mathbf{X}^+ = \mathbf{P}_{\mathbf{X}}.$$

Note that this matrix estimator $\mathbf{X}(\mathbf{I}_S\mathbf{X})^+$ is closely connected to linear regression: It can be used to transform the response vector $\mathbf{y}$ into the prediction vector $\widehat{\mathbf{y}}(S)$ of subsampled least squares solution $\mathbf{w}^*(S)$ as follows:

$$\widehat{\mathbf{y}}(S) = \mathbf{X} \underbrace{(\mathbf{I}_S\mathbf{X})^+\mathbf{y}}_{\mathbf{w}^*(S)}.$$

In this case, volume sampling once again provides a covariance-type matrix expectation formula.

Theorem 7 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be a full rank matrix. If matrix $\mathbf{X}$ is in general position and $S \subseteq \{1..n\}$ is sampled according to size d volume sampling, then*

$$\mathbb{E}[\underbrace{(\mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2}_{(\mathbf{I}_S\mathbf{X})^{+\top}\mathbf{X}^\top\mathbf{X}(\mathbf{I}_S\mathbf{X})^+}] - \mathbf{P}_{\mathbf{X}} = d\,(\mathbf{I} - \mathbf{P}_{\mathbf{X}}).$$

If $\mathbf{X}$ is not in general position, then the matrix equality "=" is replaced by the positive-definite inequality "$\preceq$".

Note that this third expectation formula is limited to sample size $s = d$. It is a direct consequence of Theorem 8 given in the next section which relates the expected loss of a subsampled least squares estimator to the loss of the optimum least squares estimator. Unlike the first two formulas given in theorems 5 and 6, its proof does not rely on the methodology of Lemma 1, i.e., on showing that the expectations at all levels of a certain DAG associated with the sampling process are the same. We defer the proof of this third expectation formula to the end of Section 3.3. No extension of this third formula to sample size $s > d$ is known.

Proof of Theorem 5 We apply Lemma 1 with $\mathbf{F}(S) = (\mathbf{I}_S\mathbf{X})^+$. It suffices to show $\mathbf{F}(S) = \sum_{i\in S} P(S_{-i}|S)\mathbf{F}(S_{-i})$ for $P(S_{-i}|S) = \frac{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d}$, i.e.:

$$(\mathbf{I}_S\mathbf{X})^+ = \sum_{i\in S} \frac{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \underbrace{(\mathbf{I}_{S_{-i}}\mathbf{X})^+}_{(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}(\mathbf{I}_{S_{-i}}\mathbf{X})^\top}.$$

We first apply Sherman-Morrison to $(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1} = (\mathbf{X}_S^\top\mathbf{X}_S - \mathbf{x}_i\mathbf{x}_i^\top)^{-1}$ on the r.h.s. of the above:

$$\sum_i \frac{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{s-d} \left((\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \frac{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}\right)((\mathbf{I}_S\mathbf{X})^\top - \mathbf{x}_i\mathbf{e}_i^\top).$$

Next we expand the last two factors into 4 terms. The expectation of the first $(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}(\mathbf{I}_S\mathbf{X})^\top$ is $(\mathbf{I}_S\mathbf{X})^+$ (which is the l.h.s.) and the expectations of the remaining three terms times $s-d$ sum to 0:

$$-\sum_{i\in S}(1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i)\,(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{e}_i^\top + \cancel{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}(\mathbf{I}_S\mathbf{X})^\top$$
$$-\sum_{i\in S}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\,(\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i)\mathbf{e}_i^\top = 0. \quad \blacksquare$$

In Appendix B we give an alternate proof using a derivative argument.

Proof of Theorem 6 Choose $\mathbf{F}(S) = \frac{s-d+1}{n-d+1}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}$. By Lemma 1 it suffices to show $\mathbf{F}(S) = \sum_{i\in S} P(S_{-i}|S)\mathbf{F}(S_{-i})$ for volume sampling:

$$\frac{\cancel{s-d+1}}{\cancel{n-d+1}}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1} = \sum_{i\in S}\frac{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}{\cancel{s-d}}\frac{\cancel{s-d}}{\cancel{n-d+1}}(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}.$$

To show this we apply Sherman-Morrison to $(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}})^{-1}$ on the r.h.s.:

$$\sum_{i\in S}(1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i)\left((\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \frac{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}}{1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i}\right)$$
$$= (s-d)(\mathbf{X}_S^\top\mathbf{X}_S)^{-1} + \cancel{(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\sum_{i\in S}\mathbf{x}_i\mathbf{x}_i^\top}(\mathbf{X}_S^\top\mathbf{X}_S)^{-1} = (s-d+1)\,(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}.$$

If some denominators $1-\mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S)^{-1}\mathbf{x}_i$ are zero, then we only sum over i for which the denominators are positive. In this case the above matrix equality becomes a positive-definite inequality $\preceq$. $\blacksquare$

3. Linear Regression with Smallest Number of Responses

Our main motivation for studying volume sampling came from asking the following simple question. Suppose we want to solve a d-dimensional linear regression problem with an input matrix $\mathbf{X}$ of n rows in $\mathbb{R}^d$ and a response vector $\mathbf{y} \in \mathbb{R}^n$, i.e. find $\mathbf{w} \in \mathbb{R}^d$ that minimizes the least squares loss $\|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2$ on all n rows. We use $L(\mathbf{w})$ to denote this loss. The optimal weight vector minimizes $L(\mathbf{w})$, i.e.

$$\mathbf{w}^* \stackrel{def}{=} \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} L(\mathbf{w}) = \mathbf{X}^+\mathbf{y}.$$

Computing it requires access to the input matrix $\mathbf{X}$ and the response vector $\mathbf{y}$. Assume we are given $\mathbf{X}$ but the access to response vector $\mathbf{y}$ is restricted. We are allowed to pick a random subset $S \subseteq \{1..n\}$ of fixed size s for which the responses $\mathbf{y}_S$ for the submatrix $\mathbf{X}_S$ are revealed to us, and then must produce a weight vector $\mathbf{w}(\mathbf{X}, S, \mathbf{y}_S) \in \mathbb{R}^d$ from a subset of row indices S of the input matrix $\mathbf{X}$ and the corresponding responses $\mathbf{y}_S$. Our goal in this paper is to find a distribution on the subsets S of size s and a *weight function* $\mathbf{w}(\mathbf{X}, S, \mathbf{y}_S)$ s.t.[6]

$$\forall\,(\mathbf{X},\mathbf{y}) \in \mathbb{R}^{n\times d}\times\mathbb{R}^{n\times 1}: \quad \mathbb{E}\left[L(\mathbf{w}(\mathbf{X}, S, \mathbf{y}_S))\right] \le (1+c)\,L(\mathbf{w}^*),$$

6. Since the learner is given $\mathbf{X}$, it is natural to define the optimal multiplicative constant specialized for each $\mathbf{X}$: $c_{\mathbf{X},s} = \min_c \min_{P(\cdot),\mathbf{w}(\cdot)} \max_{\mathbf{y}} \mathbb{E}_P\left[L(\mathbf{w}(\mathbf{X}, S, \mathbf{y}_S))\right] \le (1+c)\,L(\mathbf{w}^*)$, where the domain for distribution $P(\cdot)$ and weight function $\mathbf{w}(\cdot)$ are sets of size s. Showing specialized bounds for $c_{\mathbf{X},s}$ is left for future research.

where c must be a fixed constant (that is independent of $\mathbf{X}$ and $\mathbf{y}$). Throughout the paper we use the one argument shorthand $\mathbf{w}(S)$ for the weight function $\mathbf{w}(\mathbf{X}, S, \mathbf{y}_S)$. We assume that attaining response values is expensive and ask the question: What is the smallest number of responses (i.e. smallest size of S) for which such a multiplicative bound is possible? We will use volume sampling to show that attaining d response values is sufficient and show that less than d responses is not.

Before we state our main upper bound based on volume sampling, we make the following key observation: If for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$ there is a weight vector $\mathbf{w}(S)$ that has loss zero, then the algorithm has to predict with such a consistent weight vector. This is because in that case the responses $\mathbf{y}_S$ can be extended to a response vector $\mathbf{y}$ for all of $\mathbf{X}$ s.t. $L(\mathbf{w}^*) = 0$. Thus since we aim for a multiplicative loss bound, we force the algorithm to predict with the optimum solution $\mathbf{w}^*(S) \stackrel{def}{=} (\mathbf{X}_S)^+ \mathbf{y}_S$ whenever the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$ has loss 0. In particular, when $|S| = d$ and $\mathbf{X}_S$ has full rank, then there is a unique consistent solution $\mathbf{w}^*(S)$ for the subproblem and the learner must use the weight function $\mathbf{w}(S) = \mathbf{w}^*(S)$.

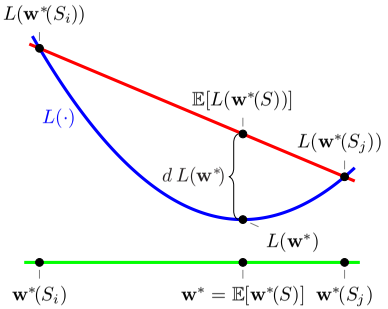


Figure 4: Unbiased estimator $\mathbf{w}^*(S)$ in expectation suffers loss $(d+1)\,L(\mathbf{w}^*)$.

Theorem 8 *If the input matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ is in general position, then for any response vector $\mathbf{y} \in \mathbb{R}^n$, the expected square loss (on all n rows of $\mathbf{X}$) of the optimal solution $\mathbf{w}^*(S)$ for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$, with the d-element set S obtained from volume sampling, is given by*

$$\mathbb{E}[L(\mathbf{w}^*(S))] = (d+1)\,L(\mathbf{w}^*).$$

If $\mathbf{X}$ is not in general position, then the expected loss is upper-bounded by $(d+1)\,L(\mathbf{w}^*)$.

There are no range restrictions on the n points and response values in this bound. Also, as discussed in the introduction, this bound is already non-obvious for dimension 1, when the multiplicative factor is 2 (See Figure 1 for a visualization). Note that if there is a bias term in dimension 1, then the factor becomes 3.

In dimension d, it is instructive to look at the case when the square loss of the optimum solution is zero, i.e. there is a weight vector $\mathbf{w}^* \in \mathbb{R}^d$ s.t. $\mathbf{X}\mathbf{w}^* = \mathbf{y}$. In this case the response values of any d linearly independent rows of $\mathbf{X}$ determine the optimum solution and the multiplicative loss formula of the theorem clearly holds. The formula specifies how noise-free case generalizes gracefully to the noisy case in that for volume sampling, the expected square loss of the solution obtained from d row response pairs is always by a factor of at most $d+1$ larger than the square loss of the optimum solution. Moreover, since $\mathbb{E}[\mathbf{w}^*(S)] = \mathbf{w}^*$ and the loss function $L(\cdot)$ is convex, we have by Jensen's inequality that

$$\mathbb{E}\big[L(\mathbf{w}^*(S))\big] \geq L\big(\mathbb{E}[\mathbf{w}^*(S)]\big) = L(\mathbf{w}^*).$$

The above theorem now states that the gap $\mathbb{E}[L(\mathbf{w}^*(S))] - L(\mathbf{w}^*)$ in Jensen's inequality (which coincides with the "regret" of the estimator) equals $d\,L(\mathbf{w}^*)$, when the expectation is w.r.t. size d volume sampling and $\mathbf{X}$ is in general position (See Figure 4 for a schematic). As we will show in Section 3.4, this gap also equals the variance $\mathbb{E}[\|\mathbf{X}\mathbf{w}^*(S) - \mathbf{X}\mathbf{w}^*\|^2]$ of the predictions since the estimator is unbiased. In summary:

$$\underbrace{\mathbb{E}\big[L(\mathbf{w}^*(S))\big] - L(\overbrace{\mathbf{w}^*}^{\mathbb{E}[\mathbf{w}^*(S)]})}_{\text{regret}} = \underbrace{d\,L(\mathbf{w}^*)}_{\text{gap in Jensen's}} = \underbrace{\mathbb{E}\big[\|\mathbf{X}\mathbf{w}^*(S) - \mathbf{X}\mathbf{w}^*\|^2\big]}_{\text{variance}}.$$

We now make a number of observations and present some lower bounds that highlight the upper bound of the above theorem. Then, in Section 3.3 we prove the theorem and a matrix expectation formula implied by it.

3.1 When X is not in General Position

The above theorem gives an equality for the expected loss of a volume-sampled solution. However, this equality is only guaranteed to hold when matrix $\mathbf{X}$ is in general position. We give a minimal example problem where the matrix $\mathbf{X}$ is not in general position and the equality of Theorem 8 turns into a strict inequality. This shows that for the equality, the general position assumption is necessary. If we apply even an infinitesimal additive perturbation to the matrix $\mathbf{X}$ of the example problem, then the resulting matrix $\mathbf{X}_\epsilon$ is in general position and the equality holds. Note that even though the optimum loss $L(\mathbf{w}^*)$ does not change significantly under such a perturbation, the expected sampling loss $\mathbb{E}[L(\mathbf{w}^*(S))]$ has to jump sufficiently to close the gap in the inequality. In our minimal example problem, $n = 3$ and $d = 2$, and

$$\mathbf{X} = \begin{pmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 0 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}.$$

We have three 2-element subsets to sample from: $S_1 = \{1,2\},\ S_2 = \{2,3\},\ S_3 = \{1,3\}$. Notice that the first two rows of $\mathbf{X}$ are identical, which means that the probability of sampling set S_1 is 0 in the volume sampling process. The other two subsets, S_2 and S_3, form identical submatrices $\mathbf{X}_{S_2} = \mathbf{X}_{S_3}$. Therefore they are equally probable. The optimal weight vectors for these sets are $\mathbf{w}^*(S_2) = (0,0)^\top$ and $\mathbf{w}^*(S_3) = (0,1)^\top$. Also $\mathbf{w}^* = (0, \frac{1}{2})^\top$ and the expected loss is bounded as:

$$\mathbb{E}[L(\mathbf{w}^*(S))] = \underbrace{\frac{1}{2}\overbrace{L(\mathbf{w}^*(S_2))}^{1} + \frac{1}{2}\overbrace{L(\mathbf{w}^*(S_3))}^{1}}_{1} \ < \ \underbrace{\overbrace{(d+1)}^{3}\overbrace{L(\mathbf{w}^*)}^{1/2}}_{3/2}.$$

Now consider a slightly perturbed input matrix

$$\mathbf{X}_\epsilon = \begin{pmatrix} 1 & 1+\epsilon \\ 1 & 1 \\ 1 & 0 \end{pmatrix},$$

where $\epsilon > 0$ is arbitrarily small (We keep the response vector $\mathbf{y}$ the same). Now, there is no $d \times d$ submatrix that is singular, so the upper bound from Theorem 8 must be tight. The reason is that even though subset S_1 still has very small probability, its loss is very large, so the expectation is significantly affected by this component, no matter how small ϵ is. We see this directly in the calculations. Let $\mathbf{w}^*$ and $\mathbf{w}^*(S_i)$ be the corresponding solutions for the perturbed problem and its subproblems. The volumes of the subproblems and their losses are:

$$\begin{array}{ll} \det(\mathbf{X}_{S_1}^\top \mathbf{X}_{S_1}) = \epsilon^2 & L(\mathbf{w}^*(S_1)) = \epsilon^{-2} \\ \det(\mathbf{X}_{S_2}^\top \mathbf{X}_{S_2}) = 1 & L(\mathbf{w}^*(S_2)) = 1 \\ \det(\mathbf{X}_{S_3}^\top \mathbf{X}_{S_3}) = (1+\epsilon)^2 & L(\mathbf{w}^*(S_3)) = (1+\epsilon)^{-2} \end{array} \qquad L(\mathbf{w}^*) = \frac{1}{2(1+\epsilon+\epsilon^2)}.$$

Note that for each subproblem, the product of volume times loss is equal to 1. Now the expected loss can be easily computed, and we can see that the gap in the bound disappears (the denominator is the normalizing constant for volume sampling):

$$\mathbb{E}[L(\mathbf{w}^*(S))] = \frac{1+1+1}{\epsilon^2+1+(1+\epsilon)^2} = (d+1)\ L(\mathbf{w}^*).$$

3.2 Lower Bounds and the Importance of Joint Sampling

The factor $d+1$ in Theorem 8 cannot, in general, be improved when selecting only d responses:

Proposition 9 *For any d, there exists a least squares problem $(\mathbf{X}, \mathbf{y})$ with $d+1$ rows in $\mathbb{R}^d$ such that for every d-element index set $S \subseteq \{1\,..\,d+1\}$, we have*

$$L(\mathbf{w}^*(S)) = (d+1)\ L(\mathbf{w}^*).$$

Proof Choose the input vectors $\mathbf{x}_i$ (and rows $\mathbf{x}_i^\top$) as the $d+1$ corners of any simplex in $\mathbb{R}^d$ centered at the origin and choose all $d+1$ responses as the same non-zero value α. For any α, the optimal solution $\mathbf{w}^*$ will be the all-zeros vector with loss

$$L(\mathbf{w}^*) = (d+1)\ \alpha^2.$$

On the other hand, taking any size d subset of indices $S \subseteq \{1\,..\,d+1\}$, the subproblem solution $\mathbf{w}^*(S)$ will only produce loss on the left out input vector $\mathbf{x}_i$, indexed with $i \notin S$. To obtain the prediction on x_i, we use a simple geometric argument. Observe that since the simplex is centered, we can write the origin of $\mathbb{R}^d$ in terms of the corners of the simplex as

$$\mathbf{0} = \sum_k \mathbf{x}_k = \mathbf{x}_i + d\,\bar{\mathbf{x}}_{-i}, \qquad \text{where } \bar{\mathbf{x}}_{-i} \stackrel{def}{=} \frac{1}{d}\sum_{k\neq i} \mathbf{x}_k.$$

Thus, the left out input vector $\mathbf{x}_i$ equals $-d\,\bar{\mathbf{x}}_{-i}$. The prediction of $\mathbf{w}^*(S)$ on this vector is

$$\widehat{y}_i = \mathbf{x}_i^\top \mathbf{w}^*(S) = -d\left(\frac{1}{d}\sum_{k\neq i} \mathbf{x}_k^\top\right)\mathbf{w}^*(S) = -\sum_{k\neq i} \mathbf{x}_k^\top \mathbf{w}^*(S) = -d\alpha.$$

It follows that the loss of $\mathbf{w}^*(S)$ equals

$$L(\mathbf{w}^*(S)) = (\widehat{y}_i - y_i)^2 = (-d\alpha - \alpha)^2 = (d+1)^2\alpha^2 = (d+1)\,L(\mathbf{w}^*). \qquad \blacksquare$$

Moreover, it is easy to show that no *deterministic* algorithm for selecting d rows (without knowing the responses) can guarantee a multiplicative loss bound with a factor less than n/d (Boutsidis et al., 2013). For the sake of completeness, we show this here for $d = 1$:

Proposition 10 *For any $n \times 1$ input matrix $\mathbf{X}$ of all 1's and any deterministic algorithm that chooses some singleton set $S = \{i\}$, there is a response vector $\mathbf{y}$ for which the loss of the subproblem and the optimal loss are related as follows:*

$$L(\mathbf{w}^*(S)) = n\,L(\mathbf{w}^*).$$

Proof If the response vector $\mathbf{y}$ is the vector of n 1's except for a single 0 at index i, then we have

$$\underbrace{L(\overbrace{\mathbf{w}^*(\{i\})}^{0})}_{n-1} = n\ \underbrace{L(\overbrace{\mathbf{w}^*}^{\frac{n-1}{n}})}_{\frac{n-1}{n}}. \qquad \blacksquare$$

Note that for the 1-dimensional example used in the proof, volume sampling would pick the set S uniformly. For this distribution, the multiplicative factor drops from n down to 2, that is $\mathbb{E}[L(\mathbf{w}^*(S))] = \frac{1}{n}(n-1) + \frac{n-1}{n}1 = 2\ L(\mathbf{w}^*)$.

The importance of joint sampling. Three properties of volume sampling play a crucial role in achieving a multiplicative loss bound:

a) *Randomness.* No deterministic algorithm guarantees such a bound (see Proposition 10).

b) *The chosen submatrices must have full rank.* Choosing any rank deficient submatrix with positive probability, does not allow for a multiplicative bound (see Propositions 11 and 12).

c) *Jointness.* No i.i.d. sampling procedure can achieve a multiplicative loss bound with $O(d)$ responses (see Corollary 13).

By jointly selecting subset S, volume sampling ensures that the corresponding input vectors $\mathbf{x}_i$ are well spread out in the input space $\mathbb{R}^d$. In particular, volume sampling does not put any probability mass on sets S such that the rank of submatrix $\mathbf{X}_S$ is less than d. Intuitively, selecting rank deficient row subsets should not be effective, since such a choice leads to an under-determined least squares problem. We make this simple statement more precise by showing that any randomized algorithm, that with positive probability selects a rank deficient row subset, cannot achieve a multiplicative loss bound. Intuitively if the algorithm picks a rank deficient subset then it is not clear how it should select the weight vector $\mathbf{w}(S)$ given input matrix $\mathbf{X}$, subset S and responses $\mathbf{y}_S$. We reasoned before that $\mathbf{w}(S)$ must have loss 0 on the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. However if $\mathrm{rank}(\mathbf{X}_S) < d$, then the choice of the weight vector $\mathbf{w}(S)$ with loss 0 is not unique and this causes positive loss for some response vector $\mathbf{y}$.

Proposition 11 *If for any input matrix $\mathbf{X}$, the algorithm samples a rank deficient subset S of rows with positive probability, then the expected loss of the algorithm cannot be bounded by a constant times the optimum loss for all response vectors $\mathbf{y}$.*

Note that this means in particular that if $\mathbf{X}$ has rank d, then sampling $d-1$ size subsets with positive probability does not allow for a constant factor approximation.

Proof Let S be a rank deficient subset chosen with probability $P(S) > 0$. Since in our setup the bound has to hold for all response vectors $\mathbf{y}$ we can imagine an adversary choosing a worst-case $\mathbf{y}$. This adversary gives all rows of $\mathbf{X}_S$ the response value zero. Let $\mathbf{w}(S)$ be the plane produced by the algorithm when choosing S and receiving the responses 0 for $\mathbf{X}_S$. Let $i \in \{1..n\}$ s.t. $\mathbf{x}_i^\top \notin$ row-span$(\mathbf{X}_S)$ and let $\mathbf{w}^*$ be any weight vector that gives response value 0 to all rows of $\mathbf{X}_S$ and response value $\mathbf{x}_i^\top \mathbf{w}(S) + Y$ to $\mathbf{x}_i$. The adversary chooses $\mathbf{y}$ as $\mathbf{X}\mathbf{w}^*$, i.e. it gives all points $\mathbf{x}_j$ not indexed by S and different from $\mathbf{x}_i$ the response values $\mathbf{x}_j^\top \mathbf{w}^*$ as well. Now $\mathbf{w}^*$ has total loss 0 but $\mathbf{w}(S)$ has loss Y^2 on $\mathbf{x}_i$ and the algorithm's expected total loss is $\geq P(S)\,Y^2$. $\blacksquare$

We now strengthen the above proposition in that whenever the sample S is rank deficient then the loss of the optimum is zero while the loss of the algorithm is positive. However note that this proposition is weaker than the above in that it only holds for specific input matrices.

Proposition 12 *Let $d \leq n$ and let $\mathbf{X}$ be any input matrix of rank d consisting of n standard basis row vectors in $\mathbb{R}^d$. Then for any randomized learning algorithm that with probability p selects a subset S s.t. $\mathrm{rank}(\mathbf{X}_S) < d$ and any weight function $\mathbf{w}(\cdot)$, there is a response vector $\mathbf{y}$, satisfying:*

$$L(\mathbf{w}^*) = 0, \quad \textit{and} \quad L(\mathbf{w}(S)) > 0 \quad \textit{with probability at least } p.$$

Proof Let $Q = \{1, 2, \ldots, 2^n\}$. The adversarial response vector $\mathbf{y}$ is constructed by carefully selecting one of the weight vectors $\mathbf{w}^* \in Q^d$, and setting the response vector $\mathbf{y}$ to $\mathbf{X}\mathbf{w}^*$. This ensures that $L(\mathbf{w}^*) = 0$ and since $\mathbf{X}$ consists of standard basis row vectors, the components of $\mathbf{y}$ lie in Q as well. Note that if the learner does not discover $\mathbf{w}^*$ exactly, it will incur positive loss. Let $\mathcal{H}$ be the set of all rank deficient sets in $\mathbf{X}$, i.e. those that lack at least one of the standard basis vectors:

$$\mathcal{H} = \{S \subseteq \{1..n\} \ : \ \mathrm{rank}(\mathbf{X}_S) < d\}.$$

Suppose that given matrix $\mathbf{X}$, the learner uses weight function $\mathbf{w}(S, \mathbf{y}_S)$. (Note that for the sake of concreteness we stopped using the single argument shorthand for the weight function during this proof.) We will count the number of possible inputs to this function, when S is a rank deficient index set of the rows of $\mathbf{X}$ and the response vector $\mathbf{y}_S$ is consistent with some $\mathbf{w}^* \in Q^d$. For any fixed rank deficient set S, let t be the number of distinct basis vectors appearing in $\mathbf{X}_S$. Clearly $t \leq d-1$. Fix a subset $T \subseteq S$ of size t s.t. $\mathbf{X}_T$ contains all t basis vectors of $\mathbf{X}_S$ exactly once (Thus the basis vectors in $\mathbf{X}_{S\setminus T}$ are all duplicates). Since $\mathbf{y} \in Q^n$, the components of $\mathbf{y}_S$ also lie in Q and $\mathbf{y}_S$ is determined by the responses of $\mathbf{y}_T$. Clearly there are at most $|Q|^{d-1}$ choices for $\mathbf{y}_T$. It follows that the number of possible input pairs $(S, \mathbf{y}_S)$ for function $\mathbf{w}(\cdot,\cdot)$ under the above restrictions can be bounded as

$$\left|\left\{(S, \mathbf{y}_S) \ : \ [S \in \mathcal{H}] \text{ and } [\mathbf{y}_S = \mathbf{X}_S\mathbf{w}^* \text{ for } \mathbf{w}^* \in Q^d]\right\}\right| \leq \underbrace{|\mathcal{H}|}_{<2^n} \underbrace{\max_{S\in\mathcal{H}} |\{\mathbf{X}_S\mathbf{w}^* \ : \ \mathbf{w}^* \in Q^d\}|}_{\leq |Q|^{d-1}}$$
$$< 2^n |Q|^{d-1} = |Q^d|.$$

So for every weight function $\mathbf{w}(\cdot,\cdot)$, there exists $\mathbf{w}^* \in Q^d$ that is not present in the set $\{\mathbf{w}(S, \mathbf{y}_S) : S \in \mathcal{H}\}$. Selecting $\mathbf{y} = \mathbf{X}\mathbf{w}^*$ for the adversarial response vector, we guarantee that the learner picks the wrong solution for every rank deficient set S and therefore receives positive loss w.p. at least p. ■

Using Proposition 12, we show that any i.i.d. row sampling distribution (like for example leverage score sampling) requires $\Omega(d \log d)$ samples to get any multiplicative loss bound, either with high probability or in expectation.

Corollary 13 *Let $d \leq n$ and let $\mathbf{X}$ be any input matrix of rank d consisting of n standard basis row vectors in $\mathbb{R}^d$. Then for any randomized learning algorithm which selects a random multiset $S \subseteq \{1..n\}$ of size $|S| \leq (d-1)\ln(d)$ via i.i.d. sampling from any distribution and uses any weight function $\mathbf{w}(S)$, there is a response vector $\mathbf{y}$ satisfying:*

$$L(\mathbf{w}^*) = 0, \quad \textit{and} \quad L(\mathbf{w}(S)) > 0 \quad \textit{with probability at least } 1/2.$$

Proof Any i.i.d. sample of size at most $(d-1)\ln(d)$ with probability at least $1/2$ does not contain all of the unique standard basis vectors (Coupon Collector Problem[7]). Thus, with probability at least $1/2$ submatrix $\mathbf{X}_S$ has rank less than d. Now, for any such algorithm we can use Proposition 12 to select a consistent adversarial response vector $\mathbf{y}$ such that with probability at least $1/2$ the loss $L(\mathbf{w}(S))$ is positive. ■

Note that the corollary requires $\mathbf{X}$ to be of a restricted form that contains a lot of duplicate rows. It is open whether this corollary still holds when $\mathbf{X}$ is an arbitrary full rank matrix.

3.3 Proof of the Loss Expectation Formula

First, we discuss several key connections between linear regression and volume, which are used in the proof. Note that the loss $L(\mathbf{w}^*)$ suffered by the optimum weight vector can be written as $\|\widehat{\mathbf{y}} - \mathbf{y}\|^2$, the squared Euclidean distance between prediction vector $\widehat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$ and the response vector $\mathbf{y}$. Since $\widehat{\mathbf{y}}$ is minimizing the distance from $\mathbf{y}$ to the subspace of $\mathbb{R}^n$ spanning the feature vectors $\{\mathbf{f}_1, \ldots, \mathbf{f}_d\}$ (columns of $\mathbf{X}$), it has to be the *projection* of $\mathbf{y}$ onto that subspace (see Figure 5). We denote this projection as $\mathbf{P}_\mathbf{X}\,\mathbf{y}$, as defined in Section 2.2. Note that $\mathbf{P}_\mathbf{X}$ is a linear mapping from $\mathbb{R}^n$ onto the column span of the matrix $\mathbf{X}$ such that

$$\text{for } \mathbf{u} \in \mathrm{span}(\mathbf{X}) \quad \mathbf{u} = \mathbf{P}_\mathbf{X}\,\mathbf{y} \ \Leftrightarrow \ \mathbf{P}_\mathbf{X}\,(\mathbf{u} - \mathbf{y}) = \mathbf{0} \ \Leftrightarrow \ \mathbf{X}^\top(\mathbf{u} - \mathbf{y}) = \mathbf{0}. \tag{4}$$

We next give a second geometric interpretation of the length $\|\widehat{\mathbf{y}} - \mathbf{y}\|^2$. Let $\mathcal{P}$ be the parallelepiped formed by the d column/feature vectors of the input matrix $\mathbf{X}$. Furthermore, consider the extended input matrix produced by adding the response vector $\mathbf{y}$ to $\mathbf{X}$ as an extra column:

$$\widetilde{\mathbf{X}} \stackrel{def}{=} (\mathbf{X}, \mathbf{y}) \in \mathbb{R}^{n\times(d+1)}. \tag{5}$$

Using the "base × height" formula we can relate the volume of $\mathcal{P}$ to the volume of $\widetilde{\mathcal{P}}$, the parallelepiped formed by the $d+1$ columns of $\widetilde{\mathbf{X}}$. Observe that $\widetilde{\mathcal{P}}$ has $\mathcal{P}$ as one of its faces, with the response vector $\mathbf{y}$ representing the edge that protrudes from that face. Hence the volume of $\widetilde{\mathcal{P}}$ is the product of the volume of $\mathcal{P}$ and the distance between $\mathbf{y}$ and $\mathrm{span}(\mathbf{X})$. This distance equals $\|\widehat{\mathbf{y}} - \mathbf{y}\|$, since as discussed above, $\widehat{\mathbf{y}}$ is the projection of $\mathbf{y}$ onto $\mathrm{span}(\mathbf{X})$. Thus we have

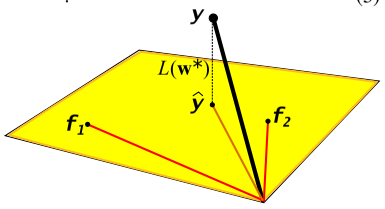


Figure 5: Prediction vector $\widehat{\mathbf{y}}$ is a projection of $\mathbf{y}$ onto the span of feature vectors $\mathbf{f}_i$.

$$\det(\widetilde{\mathbf{X}}^\top\widetilde{\mathbf{X}}) = \det(\mathbf{X}^\top\mathbf{X})\, L(\mathbf{w}^*). \tag{6}$$

Next, we present a proposition whose corollary is key to proving Theorem 8. Suppose that we select one test row from the input matrix and use the remaining $n-1$ row response pairs as the training set. The proposition relates the loss of the obtained solution on the test row to the total leave-one-out loss an all rows.

Proposition 14 *For any index $i \in \{1..n\}$, let $\mathbf{w}^*(-i)$ be the solution to the reduced linear regression problem $(\mathbf{X}_{-i}, \mathbf{y}_{-i})$. Then*

7. This was proven for uniform sampling in Theorem 1.24 of Auger and Doerr (2011). It can be shown that uniform sampling is the best case for Coupon Collector Problem (Holst, 2001), so the bound holds for any i.i.d. sampling.

$$L(\mathbf{w}^*(-i)) - L(\mathbf{w}^*) = \overbrace{\mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i}^{\frac{\det(\mathbf{X}^\top \mathbf{X}) - \det(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})}{\det(\mathbf{X}^\top \mathbf{X})}} \quad \ell_i(\mathbf{w}^*(-i)),$$

where $\ell_i(\mathbf{w}) \stackrel{def}{=} (\mathbf{x}_i^\top \mathbf{w} - y_i)^2$ *is the square loss of* $\mathbf{w}$ *on the* i*-th point.*

An algebraic proof of this proposition essentially appears in the proof of Theorem 11.7 in Cesa-Bianchi and Lugosi (2006). For the sake of completeness we give a new geometric proof of this proposition in Appendix C using basic properties of volume, thus stressing the connection to volume sampling.

Note that if matrix $\mathbf{X}$ has exactly $n = d + 1$ rows and the training matrix $\mathbf{X}_{-i}$ is full rank, then $\mathbf{w}^*(-i)$ has loss zero on all training rows. In this case we obtain a simpler relationship than the proposition.

Corollary 15 *If* $\mathbf{X}$ *has* $d + 1$ *rows and* $\operatorname{rank}(\mathbf{X}_{-i}) = d$*, then defining* $\widetilde{\mathbf{X}}$ *as in (5), we have*

$$\det(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}}) = \det(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})\ \ell_i(\mathbf{w}^*(-i)).$$

Proof By Proposition 14 and the fact that $L(\mathbf{w}^*(-i)) = \ell_i(\mathbf{w}^*(-i))$, we have

$$\det(\mathbf{X}^\top \mathbf{X})\ L(\mathbf{w}^*) = \det(\mathbf{X}_{-i}^\top \mathbf{X}_{-i})\ \ell_i(\mathbf{w}^*(-i)).$$

The corollary now follows from the "base $\times$ height" formula for volume. ■

We are now ready to present the proof of Theorem 8. Recall that our goal is to find the expected loss $\mathbb{E}[L(\mathbf{w}^*(S))]$, where S is a size d volume sampled set.

Proof of Theorem 8 First, we rewrite the expectation as follows:

$$\begin{aligned}\mathbb{E}[L(\mathbf{w}^*(S))] &= \sum_{S,|S|=d} P(S) L(\mathbf{w}^*(S)) = \sum_{S,|S|=d} P(S) \sum_{j=1}^{n} \ell_j(\mathbf{w}^*(S)) \\ &= \sum_{S,|S|=d} \sum_{j \notin S} P(S)\ \ell_j(\mathbf{w}^*(S)) = \sum_{T,|T|=d+1} \sum_{j \in T} P(T_{-j})\ \ell_j(\mathbf{w}^*(T_{-j})). \qquad (7)\end{aligned}$$

We now use Corollary 15 on the matrix $\mathbf{X}_T$ and test row $\mathbf{x}_j^\top$ (assuming $\operatorname{rank}(\mathbf{X}_{T_{-j}}) = d$):

$$P(T_{-j})\ \ell_j(\mathbf{w}^*(T_{-j})) = \frac{\det(\mathbf{X}_{T_{-j}}^\top \mathbf{X}_{T_{-j}})}{\det(\mathbf{X}^\top \mathbf{X})}\ \ell_j(\mathbf{w}^*(T_{-j})) = \frac{\det(\widetilde{\mathbf{X}}_T^\top \widetilde{\mathbf{X}}_T)}{\det(\mathbf{X}^\top \mathbf{X})}. \qquad (8)$$

Since the summand does not depend on the index $j \in T$, the inner summation in (7) becomes a multiplication by $d + 1$. This lets us write the expected loss as:

$$\mathbb{E}[L(\mathbf{w}^*(S))] = \frac{d+1}{\det(\mathbf{X}^\top \mathbf{X})} \sum_{T,|T|=d+1} \det(\widetilde{\mathbf{X}}_T^\top \widetilde{\mathbf{X}}_T) \stackrel{(1)}{=} (d+1) \frac{\det(\widetilde{\mathbf{X}}^\top \widetilde{\mathbf{X}})}{\det(\mathbf{X}^\top \mathbf{X})} \stackrel{(2)}{=} (d+1)\, L(\mathbf{w}^*), \qquad (9)$$

where (1) follows from the Cauchy-Binet formula and (2) is an application of the "base $\times$ height" formula. If $\mathbf{X}$ is not in general position, then for some summands in (8), $\operatorname{rank}(\mathbf{X}_{T_{-j}}) < d$ and $P(T_{-j}) = 0$. Thus the left-hand side of (8) is 0, while the right-hand side is non-negative, so (9) becomes an inequality, completing the proof of Theorem 8. ■

Lifting expectations to matrix form. We can now show the matrix expectation formula of Theorem 7 as a corollary to the loss expectation formula of Theorem 8. The key observation is that the loss formula holds for arbitrary response vector $\mathbf{y}$, which allows us to "lift" it to the matrix form.

Proof of Theorem 7 Note, that the loss of least squares estimator can be written in terms of the projection matrix $\mathbf{P}_\mathbf{X}$:

$$L(\mathbf{w}^*) = \|\mathbf{y} - \widehat{\mathbf{y}}\|^2 = \|(\mathbf{I} - \mathbf{P}_\mathbf{X})\mathbf{y}\|^2 = \mathbf{y}^\top (\mathbf{I} - \mathbf{P}_\mathbf{X})^2 \mathbf{y} \stackrel{(*)}{=} \mathbf{y}^\top (\mathbf{I} - \mathbf{P}_\mathbf{X})\, \mathbf{y},$$

where in $(*)$ we used the following property of a projection matrix: $\mathbf{P}_\mathbf{X}^2 = \mathbf{P}_\mathbf{X}$. Writing the loss expectation of the subsampled estimator in the same form, we obtain:

$$\begin{aligned}\mathbb{E}[L(\mathbf{w}^*(S))] &= \mathbb{E}[\|\mathbf{y} - \widehat{\mathbf{y}}(S)\|^2] = \mathbb{E}[\|(\mathbf{I} - \mathbf{X}(\mathbf{I}_S\mathbf{X})^+)\mathbf{y}\|^2] \\ &= \mathbb{E}[\mathbf{y}^\top (\mathbf{I} - \mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2\, \mathbf{y}] = \mathbf{y}^\top \mathbb{E}[(\mathbf{I} - \mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2]\, \mathbf{y}.\end{aligned}$$

Crucially, we are able to extract the response vector $\mathbf{y}$ out of the expectation formula, which allows us to write the formula from Theorem 8 as follows:

$$\mathbf{y}^\top\, \mathbb{E}[(\mathbf{I} - \mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2]\, \mathbf{y} = \mathbf{y}^\top (d+1)(\mathbf{I} - \mathbf{P}_\mathbf{X})\, \mathbf{y}, \quad \forall \mathbf{y} \in \mathbb{R}^n.$$

We now use the following elementary fact: If for two symmetric matrices $\mathbf{A}$ and $\mathbf{B}$, we have $\mathbf{y}^\top \mathbf{A}\mathbf{y} = \mathbf{y}^\top \mathbf{B}\mathbf{y},\ \forall \mathbf{y} \in \mathbb{R}^n$, then $\mathbf{A} = \mathbf{B}$.[8] This gives the matrix expectation formula:

$$\mathbb{E}[(\mathbf{I} - \mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2] = (d+1)(\mathbf{I} - \mathbf{P}_\mathbf{X}).$$

Expanding square on the l.h.s. of the above and applying Theorem 5, we obtain the covariance-type equivalent form stated in Theorem 7:

$$\begin{aligned}&\mathbf{I} - 2\overbrace{\mathbb{E}[\mathbf{X}(\mathbf{I}_S\mathbf{X})^+]}^{\mathbf{P}_\mathbf{X}} + \mathbb{E}[(\mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2] = (d+1)(\mathbf{I} - \mathbf{P}_\mathbf{X}) \\ \iff\quad &\mathbb{E}[(\mathbf{X}(\mathbf{I}_S\mathbf{X})^+)^2] - \mathbf{P}_\mathbf{X} = d\,(\mathbf{I} - \mathbf{P}_\mathbf{X}).\end{aligned}$$

■

3.4 Averaging Unbiased Estimators and the Open Problem for Worst-case Responses

As discussed at the beginning of Section 3, our goal is to find a way to sample a small index set S and construct a weight function $\mathbf{w}(S)$ which uses responses $\mathbf{y}_S$ so that $\mathbb{E}[L(\mathbf{w}(S))] \leq (1 + c)\ L(\mathbf{w}^*)$, where the multiplicative factor $1 + c$ is bounded for all input matrices $\mathbf{X}$ and all response vectors $\mathbf{y}$. Recall that $L(\cdot)$ denotes the square loss on all rows and $\mathbf{w}^*$ is the optimal solution based on all responses. We show in the previous subsections that the smallest size of S for which this goal can be achieved is d (There is no sampling procedure for sets of size less than d and weight function $\mathbf{w}(S)$ for which this factor is finite). We also prove that when sets S of size d are drawn proportional to

8. Similarly, if $\mathbf{y}^\top \mathbf{A}\mathbf{y} \leq \mathbf{y}^\top \mathbf{B}\mathbf{y},\ \forall \mathbf{y} \in \mathbb{R}^n$, then the positive-definite inequality $\mathbf{A} \preceq \mathbf{B}$ holds for the matrices.

the squared volume of $\mathbf{X}_S$ (i.e. $\det(\mathbf{X}_S^\top \mathbf{X}_S)$), then $\mathbb{E}[L(\mathbf{w}^*(S))] \leq (d+1)L(\mathbf{w}^*)$, where the factor $d+1$ is optimal for some $\mathbf{X}$ and $\mathbf{y}$. Here $\mathbf{w}^*(S)$ denotes the linear least squares solution for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$.

A natural more general goal is to get arbitrarily close to the optimum loss. That is, for any ϵ, what is the smallest sample size $|S| = s$ for which there is a sampling distribution over subsets S and a weight function $\mathbf{w}(S)$ built from $\mathbf{X}$ and $\mathbf{y}_S$, such that $\mathbb{E}[L(\mathbf{w}(S))] \leq (1+\epsilon)\, L(\mathbf{w}^*)$. A related bound for i.i.d. leverage score sampling states that a sample size of $O(d \log d + \frac{d}{\epsilon})$ suffices to achieve a $1+\epsilon$ factor *with high probability* (Hsu, 2017; Dereziński, 2018), however this does not imply multiplicative bounds *in expectation*.[9]

We conjecture that some form of volume sampling can be used to achieve the $1+\epsilon$ factor with sample size $O(\frac{d}{\epsilon})$, in expectation. How close can we get with the techniques presented in this paper? We showed that size d volume sampling achieves a factor of $1+d$, but we do not know how to generalize this proof to sample size larger than d. However, one unique property of the volume-sampled estimator $\mathbf{w}^*(S)$ that can be useful here is that it is an *unbiased estimator* of $\mathbf{w}^*$. As we shall see now, this basic property has many benefits. For any unbiased estimator (i.e. $\mathbb{E}[\mathbf{w}(S)] = \mathbf{w}^*$) and optimal prediction vector $\widehat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$, consider the following rudimentary version of a bias-variance decomposition:

$$\mathbb{E}\underbrace{\|\mathbf{X}\,\mathbf{w}(S) - \mathbf{y}\|^2}_{L(\mathbf{w}(S))} = \mathbb{E}\,\|\mathbf{X}\,\mathbf{w}(S) - \widehat{\mathbf{y}} + \widehat{\mathbf{y}} - \mathbf{y}\|^2 = \mathbb{E}\,\|\mathbf{X}\,\mathbf{w}(S) - \widehat{\mathbf{y}}\|^2 + \underbrace{\|\widehat{\mathbf{y}} - \mathbf{y}\|^2}_{L(\mathbf{w}^*)}. \tag{10}$$

The unbiasedness of the estimator assures that the cross term $(\mathbf{X}\overbrace{\mathbb{E}[\mathbf{w}(S)]}^{\mathbf{w}^*} - \widehat{\mathbf{y}})^\top(\widehat{\mathbf{y}} - \mathbf{y})$ is 0. Therefore a $1+c$ factor loss bound is equivalent to a c factor variance bound, i.e.

$$\overbrace{\mathbb{E}[L(\mathbf{w}(S))] \leq (1+c)\, L(\mathbf{w}^*)}^{\text{loss bound}} \quad \Longleftrightarrow \quad \overbrace{\mathbb{E}\,\|\mathbf{X}\,\mathbf{w}(S) - \widehat{\mathbf{y}}\|^2 \leq c\, L(\mathbf{w}^*)}^{\text{variance bound}}. \tag{11}$$

To reduce the variance of any unbiased estimator $\mathbf{w}(S)$ (i.e. $\mathbb{E}[\mathbf{w}(S)] = \mathbf{w}^*$) with sample size s, we can draw k independent samples $S_1, \ldots, S_k$ of size s each and predict with the average estimator $\frac{1}{k}\sum_{j=1}^k \mathbf{w}(S_j)$. If the loss bound from (11) holds for $\mathbf{w}(S)$, then the average estimator satisfies

$$\mathbb{E}\Big[L\Big(\frac{1}{k}\sum\nolimits_j \mathbf{w}(S_j)\Big)\Big] \leq \Big(1 + \frac{c}{k}\Big)\, L(\mathbf{w}^*).$$

Setting $k = c/\epsilon$, we need $t = s\,c/\epsilon$ responses to get a $1+\epsilon$ approximation. We showed that size $s = d$ volume sampling achieves factor $c = d$. So with our current proof techniques, we need $t = d^2/\epsilon$ responses to get a $1+\epsilon$ factor approximation, for $\epsilon \in (0, d]$.[10]

9. Also, the weight vectors produced from i.i.d. leverage score sampling are not unbiased.

10. Thus when averaging the estimators of $k = t/d$ independent volume sampled sets of size d,

$$\underbrace{\mathbb{E}\Big[L\Big(\frac{1}{k}\sum\nolimits_j \mathbf{w}^*(S_j)\Big)\Big] - L(\mathbf{w}^*)}_{\text{regret}} = \underbrace{\frac{d^2 L(\mathbf{w}^*)}{t}}_{\text{prediction variance}}, \quad \text{when } \mathbf{X} \text{ is in general position.}$$

The basic open problem for worst-case responses is the following: Is there a size $O(d/\epsilon)$ *unbiased estimator* that achieves a $1+\epsilon$ factor approximation?[11] By the above averaging method this is equivalent to the following question: Is there a size $O(d)$ unbiased estimator that achieves a constant factor? This is because once we have an unbiased estimator that achieves a constant factor, then by averaging $1/\epsilon$ copies, we get the $1+O(\epsilon)$ factor. Ideally the special unbiased estimators resulting from a version of volume sampling can achieve this feat. We conclude this section with our favorite open problem: Is there a version of $O(d)$ size volume sampling that achieves a constant factor approximation?

In the next section we make some minimal statistical assumptions on the response vector which let us prove much stronger bounds: We assume that the response vector is linear plus bounded noise of mean zero. In particular we show that with this noise model, $O(d)$ size volume sampling achieves a constant factor approximation.

4. Regularized Volume Sampling for Learning with Noisy Responses

Volume sampling, as defined in Section 2.1, has certain fundamental limitations. Namely, it is undefined whenever matrix $\mathbf{X}$ is not full rank or if we wish to sample a subset S of size smaller than the dimension d. Motivated by these limitations, we propose a regularized variant, called λ-regularized volume sampling, which we define through a generalization of the reverse iterative sampling procedure:

$$P(S_{-i} \,|\, S) \propto \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})}{\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})}. \tag{12}$$

Algorithm 1 λ-regularized v. sampling

$S \leftarrow \{1..n\}$
while $|S| > s$
  $\forall_{i \in S}\colon\ h_i \leftarrow \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})}{\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})}$
  Sample $i \propto h_i$ out of S
  $S \leftarrow S - \{i\}$
end
return S

The normalization factor of this conditional probability (i.e. the sum of (12) over $i \in S$) can be computed using Sylvester's theorem:

$$\begin{aligned}\sum_{i \in S} \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})}{\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})} &= \sum_{i \in S} \big(1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1} \mathbf{x}_i\big) \\ &= |S| - \operatorname{tr}\big(\mathbf{X}_S (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1} \mathbf{X}_S^\top\big) \\ &= |S| - d + \lambda \operatorname{tr}\big((\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}\big). \end{aligned} \tag{13}$$

Note that in the special case of no regularization (i.e. $\lambda = 0$) the last trace vanishes and (13) is equal to $|S| - d$, so we recover volume sampling from Section 2.1. However, when $\lambda > 0$, then the last term is non-zero and depends on the entire matrix $\mathbf{X}_S$. This makes regularized volume sampling more complicated and certain equalities proven in previous sections for $\lambda = 0$ no longer hold. In particular, the analogous closed form of the sampling probability $P(S)$ given in Theorem 2 is not recovered because the paths from node $\{1..n\}$ to node S in the graph of Figure 3 do not all have the same probability. However, the proof technique we developed for reverse iterative sampling can still be applied, resulting in the following extension of the variance formula of Theorem 6:

11. In a recent paper (Chen and Price, 2017) a $1+\epsilon$ factor approximation has been achieved with $O(d/\epsilon)$ examples (for $\epsilon \in (0, 1]$), but the guarantee holds with high probability (and not in expectation) and the estimator is not unbiased.

Theorem 16 *For any $\mathbf{X} \in \mathbb{R}^{n\times d}$, $\lambda \geq 0$, let S be sampled according to λ-regularized size s volume sampling from $\mathbf{X}$. Then,*

$$\mathbb{E}\big[(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}\big] \preceq \frac{n - d_\lambda + 1}{s - d_\lambda + 1}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}$$

for any $s \geq d_\lambda \overset{def}{=} \operatorname{tr}(\mathbf{X}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^\top)$.

Constant d_λ is a common notion of statistical dimension often referred to as the effective degrees of freedom. If λ_i are the eigenvalues of $\mathbf{X}^\top \mathbf{X}$, then $d_\lambda = \sum_{i=1}^d \frac{\lambda_i}{\lambda_i + \lambda}$. Note that d_λ is decreasing with λ and, when $\mathbf{X}$ is full rank, $d_0 = d$. Thus, unlike Theorem 6, the above result offers meaningful bounds for sampling sets S of size smaller than d.

Proof To obtain Theorem 16, we use essentially the same methodology as described in Lemma 1, except in the regularized case equality is replaced with inequality. Recall that using Sylvester's theorem we can compute the unnormalized conditional probability from (12) as:

$$h_i = \frac{\det(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})}{\det(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})} = 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}\mathbf{x}_i.$$

From now on, we will use $\mathbf{Z}_\lambda(S) = \mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I}$ as a shorthand in the proofs. Next, letting $M = \sum_{i\in S} h_i$, we compute unnormalized expectation by applying the Sherman-Morrison formula:

$$\begin{aligned}
M\,\mathbb{E}\big[(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})^{-1} \,|\, S\big] &= \sum_{i\in S} h_i \mathbf{Z}_\lambda(S_{-i})^{-1} = \sum_{i\in S} h_i \left(\mathbf{Z}_\lambda(S)^{-1} + \frac{\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top \mathbf{Z}_\lambda(S)^{-1}}{1 - \mathbf{x}_i^\top \mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i}\right) \\
&= M\,\mathbf{Z}_\lambda(S)^{-1} + \mathbf{Z}_\lambda(S)^{-1}\Big(\sum_{i\in S} \mathbf{x}_i\mathbf{x}_i^\top\Big)\mathbf{Z}_\lambda(S)^{-1} \\
&= M\,\mathbf{Z}_\lambda(S)^{-1} + \mathbf{Z}_\lambda(S)^{-1}(\mathbf{Z}_\lambda(S) - \lambda \mathbf{I})\mathbf{Z}_\lambda(S)^{-1} \\
&= M\,\mathbf{Z}_\lambda(S)^{-1} + \mathbf{Z}_\lambda(S)^{-1} - \lambda \mathbf{Z}_\lambda(S)^{-2} \\
&\preceq (M+1)\,\mathbf{Z}_\lambda(S)^{-1}.
\end{aligned}$$

Finally, the normalization factor M (which we already computed in (13)) can be lower-bounded using the λ-statistical dimension d_λ of matrix $\mathbf{X}$:

$$M = \sum_{i\in S}(1 - \mathbf{x}_i^\top \mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i) = s - d + \lambda \operatorname{tr}(\mathbf{Z}_\lambda(S)^{-1}) \geq s - \Big(\underbrace{d - \lambda \operatorname{tr}(\mathbf{Z}_\lambda(\{1..n\})^{-1})}_{d_\lambda}\Big).$$

Putting the bounds together, we obtain that:

$$\mathbb{E}\big[(\mathbf{X}_{S_{-i}}^\top \mathbf{X}_{S_{-i}} + \lambda \mathbf{I})^{-1} \,|\, S\big] \preceq \frac{s - d_\lambda + 1}{s - d_\lambda}(\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}.$$

To prove Theorem 16 it remains to chain the conditional expectations along the sequence of subsets obtained by λ-regularized volume sampling:

$$\mathbb{E}\big[\mathbf{Z}_\lambda(S)^{-1}\big] \preceq \left(\prod_{t=s+1}^{n} \frac{t - d_\lambda + 1}{t - d_\lambda}\right) \mathbf{Z}_\lambda(\{1..n\})^{-1} = \frac{n - d_\lambda + 1}{s - d_\lambda + 1}(\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}. \quad \blacksquare$$

4.1 Ridge Regression with Noisy Responses

We apply the above result to obtain statistical guarantees for subsampling with regularized estimators. Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, we consider the task of fitting a linear model to a vector of responses $\mathbf{y} = \mathbf{X}\widetilde{\mathbf{w}} + \boldsymbol{\xi}$, where $\widetilde{\mathbf{w}} \in \mathbb{R}^d$ and the noise $\boldsymbol{\xi} \in \mathbb{R}^n$ is a mean zero random vector with covariance matrix $\operatorname{Var}[\boldsymbol{\xi}] \preceq \sigma^2 \mathbf{I}$ for some $\sigma > 0$. A classical solution to this task is the ridge estimator:

$$\mathbf{w}_\lambda^* = \operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d} \|\mathbf{X}\mathbf{w} - \mathbf{y}\|^2 + \lambda\|\mathbf{w}\|^2 = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}\mathbf{X}^\top \mathbf{y}.$$

As a consequence of Theorem 16, we show that if S is sampled with λ-regularized volume sampling from $\mathbf{X}$, then the ridge estimator for the subproblem $(\mathbf{X}_S, \mathbf{y}_S)$

$$\mathbf{w}_\lambda^*(S) = (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}\mathbf{X}_S^\top \mathbf{y}_S$$

has strong generalization properties with respect to the full problem $(\mathbf{X}, \mathbf{y})$ in terms of the mean squared prediction error (MSPE) and mean squared error (MSE).

Theorem 17 *Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ and $\widetilde{\mathbf{w}} \in \mathbb{R}^d$, and suppose that $\mathbf{y} = \mathbf{X}\widetilde{\mathbf{w}} + \boldsymbol{\xi}$, where $\boldsymbol{\xi}$ is a mean zero vector with $\operatorname{Var}[\boldsymbol{\xi}] \preceq \sigma^2 \mathbf{I}$. Let S be sampled according to λ-regularized size $s \geq d_\lambda$ volume sampling from $\mathbf{X}$ and $\mathbf{w}_\lambda^*(S)$ be the λ-ridge estimator of $\widetilde{\mathbf{w}}$ computed from subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. Then, if $\lambda \leq \frac{\sigma^2}{\|\widetilde{\mathbf{w}}\|^2}$, we have*

$$\begin{aligned}
&\textit{(mean squared prediction error)} \quad \mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\Big[\frac{1}{n}\|\mathbf{X}(\mathbf{w}_\lambda^*(S) - \widetilde{\mathbf{w}})\|^2\Big] \leq \frac{\sigma^2 d_\lambda}{s - d_\lambda + 1}, \\
&\textit{(mean squared error)} \qquad\quad \mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\big[\|\mathbf{w}_\lambda^*(S) - \widetilde{\mathbf{w}}\|^2\big] \leq \frac{\sigma^2 n \operatorname{tr}((\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1})}{s - d_\lambda + 1}.
\end{aligned}$$

Next, we present two lower bounds for MSPE of a subsampled ridge estimator which show that the statistical guarantees achieved by regularized volume sampling are nearly optimal for $s \gg d_\lambda$ and better than standard approaches for $s = O(d_\lambda)$. In particular, we show that non-i.i.d. nature of volume sampling is essential if we want to achieve good generalization when the number of responses is close to d_λ. Namely for certain data matrices, any i.i.d. subsampling procedure (such as i.i.d. leverage score sampling) requires at least $d_\lambda \ln(d_\lambda)$ responses to achieve MSPE below σ^2. In contrast volume sampling obtains that bound for any matrix with $2d_\lambda$ responses.

Theorem 18 *For any $p \geq 1$ and $\sigma \geq 0$, there is $d \geq p$ such that for any sufficiently large n divisible by d, there exists a matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ such that*

$$d_\lambda(\mathbf{X}) \geq p \quad \textit{for any} \quad 0 \leq \lambda \leq \sigma^2,$$

and for each of the following two statements there is a vector $\widetilde{\mathbf{w}} \in \mathbb{R}^d$ for which the corresponding regression problem $\mathbf{y} = \mathbf{X}\widetilde{\mathbf{w}} + \boldsymbol{\xi}$ with $\operatorname{Var}[\boldsymbol{\xi}] = \sigma^2 \mathbf{I}$ satisfies that statement:

a) For any subset $S \subseteq \{1..n\}$ of size s,

$$\mathbb{E}_{\boldsymbol{\xi}}\Big[\frac{1}{n}\|\mathbf{X}(\mathbf{w}_\lambda^*(S) - \widetilde{\mathbf{w}})\|^2\Big] \geq \frac{\sigma^2 d_\lambda}{s + d_\lambda};$$

b) For multiset $S \subseteq \{1..n\}$ of size $s \leq (d_\lambda - 1)\ln(d_\lambda)$, sampled i.i.d. from any distribution over $\{1..n\}$,

$$\mathbb{E}_S \mathbb{E}_{\boldsymbol{\xi}}\Big[\frac{1}{n}\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\Big] \geq \sigma^2.$$

Proof of Theorem 17 Standard analysis for the ridge regression estimator follows by performing bias-variance decomposition of the error, and then selecting λ so that bias can be appropriately bounded. We will recall this calculation for a fixed subproblem $(\mathbf{X}_S, \mathbf{y}_S)$. First, we compute the bias of the ridge estimator for a fixed set S (recall the shorthand $\mathbf{Z}_\lambda(S) = \mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I}$):

$$\begin{aligned}
\text{Bias}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)] &= \mathbb{E}[\mathbf{w}^*_\lambda(S)] - \widetilde{\mathbf{w}} = \mathbb{E}_{\boldsymbol{\xi}}\left[\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}_S^\top \mathbf{y}_S\right] - \widetilde{\mathbf{w}} \\
&= \mathbf{Z}_\lambda(S)^{-1}\mathbf{X}_S^\top\left(\mathbf{X}_S\widetilde{\mathbf{w}} + \mathbb{E}_{\boldsymbol{\xi}}[\boldsymbol{\xi}_S]\right) - \widetilde{\mathbf{w}} \\
&= (\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}_S^\top\mathbf{X}_S - \mathbf{I})\widetilde{\mathbf{w}} = -\lambda\,\mathbf{Z}_\lambda(S)^{-1}\widetilde{\mathbf{w}}.
\end{aligned}$$

Similarly, the covariance matrix of $\mathbf{w}^*_\lambda(S)$ is given by:

$$\begin{aligned}
\text{Var}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)] &= \mathbf{Z}_\lambda(S)^{-1}\mathbf{X}_S^\top \text{Var}_{\boldsymbol{\xi}}[\boldsymbol{\xi}_S]\mathbf{X}_S\mathbf{Z}_\lambda(S)^{-1} \\
&\preceq \sigma^2\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}_S^\top\mathbf{X}_S\mathbf{Z}_\lambda(S)^{-1} = \sigma^2(\mathbf{Z}_\lambda(S)^{-1} - \lambda\,\mathbf{Z}_\lambda(S)^{-2}).
\end{aligned}$$

Mean squared error of the ridge estimator for a fixed subset S can now be bounded by:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[\|\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}}\|^2\right] &= \text{tr}(\text{Var}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)]) + \|\text{Bias}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)]\|^2 \\
&\leq \sigma^2\text{tr}(\mathbf{Z}_\lambda(S)^{-1} - \lambda\mathbf{Z}_\lambda(S)^{-2}) + \lambda^2\text{tr}(\mathbf{Z}_\lambda(S)^{-2}\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^\top) \\
&\leq \sigma^2\text{tr}(\mathbf{Z}_\lambda(S)^{-1}) + \lambda\text{tr}(\mathbf{Z}_\lambda(S)^{-2})(\lambda\|\widetilde{\mathbf{w}}\|^2 - \sigma^2) \qquad (14) \\
&\leq \sigma^2\text{tr}(\mathbf{Z}_\lambda(S)^{-1}), \qquad (15)
\end{aligned}$$

where in (14) we applied Cauchy-Schwartz inequality for matrix trace, and in (15) we used the assumption that $\lambda \leq \frac{\sigma^2}{\|\widetilde{\mathbf{w}}\|^2}$. Thus, taking expectation over the sampling of set S, we get

$$\begin{aligned}
\mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\left[\|\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}}\|^2\right] &\leq \sigma^2\mathbb{E}_S\left[\text{tr}(\mathbf{Z}_\lambda(S)^{-1})\right] \\
\text{(Theorem 16)} \quad &\leq \sigma^2\frac{n - d_\lambda + 1}{s - d_\lambda + 1}\text{tr}(\mathbf{Z}_\lambda(\{1..n\})^{-1}) \qquad (16) \\
&\leq \frac{\sigma^2\, n\,\text{tr}((\mathbf{X}^\top\mathbf{X} + \lambda\mathbf{I})^{-1})}{s - d_\lambda + 1}.
\end{aligned}$$

Next, we bound the mean squared prediction error. As before, we start with the standard bias-variance decomposition for fixed set S:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\right] &= \text{tr}(\text{Var}_{\boldsymbol{\xi}}[\mathbf{X}\mathbf{w}^*_\lambda(S)]) + \|\mathbf{X}(\mathbb{E}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)] - \widetilde{\mathbf{w}})\|^2 \\
&\leq \sigma^2\text{tr}(\mathbf{X}(\mathbf{Z}_\lambda(S)^{-1} - \lambda\,\mathbf{Z}_\lambda(S)^{-2})\mathbf{X}^\top) + \lambda^2\text{tr}(\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}^\top\mathbf{X}\mathbf{Z}_\lambda(S)^{-1}\widetilde{\mathbf{w}}\widetilde{\mathbf{w}}^\top) \\
&\leq \sigma^2\text{tr}(\mathbf{X}\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}^\top) + \lambda\,\text{tr}(\mathbf{X}\mathbf{Z}_\lambda(S)^{-2}\mathbf{X}^\top)(\lambda\|\widetilde{\mathbf{w}}\|^2 - \sigma^2) \\
&\leq \sigma^2\text{tr}(\mathbf{X}\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}^\top).
\end{aligned}$$

Once again, taking expectation over subset S, we have

$$\begin{aligned}
\mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\left[\frac{1}{n}\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\right] &\leq \frac{\sigma^2}{n}\mathbb{E}_S\left[\text{tr}(\mathbf{X}\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}^\top)\right] = \frac{\sigma^2}{n}\text{tr}(\mathbf{X}\,\mathbb{E}_S[\mathbf{Z}_\lambda(S)^{-1}]\,\mathbf{X}^\top) \\
\text{(Theorem 16)} \quad &\leq \frac{\sigma^2}{n}\frac{n - d_\lambda + 1}{s - d_\lambda + 1}\text{tr}(\mathbf{X}\mathbf{Z}_\lambda(\{1..n\})^{-1}\mathbf{X}^\top) \leq \frac{\sigma^2 d_\lambda}{s - d_\lambda + 1}. \qquad (17)
\end{aligned}$$

The key part of proving both bounds is the application of Theorem 16. For MSE, we only used the trace version of the inequality (see (16)), however to obtain the bound on MSPE we used the more general positive semi-definite inequality in (17). ■

Proof of Theorem 18 Let $d = \lceil p \rceil + 1$ and $n \geq \lceil \sigma^2 \rceil d(d-1)$ be divisible by d. We define

$$\mathbf{X} \overset{def}{=} [\mathbf{I}, ..., \mathbf{I}]^\top \in \mathbb{R}^{n \times d}, \qquad \widetilde{\mathbf{w}}^\top \overset{def}{=} [a\sigma, ..., a\sigma] \in \mathbb{R}^d$$

for some $a > 0$. For any $\lambda \leq \sigma^2$, the λ-statistical dimension of $\mathbf{X}$ is

$$d_\lambda = \text{tr}(\mathbf{X}\,\mathbf{Z}_\lambda(\{1..n\})^{-1}\mathbf{X}^\top) \geq \frac{\lceil \sigma^2 \rceil d(d-1)}{\lceil \sigma^2 \rceil (d-1) + \lambda} \geq \frac{d(d-1)}{d - 1 + 1} \geq p.$$

Let $S \subseteq \{1..n\}$ be any set of size s, and for $i \in \{1..d\}$ let $s_i \overset{def}{=} |\{i \in S : \mathbf{x}_i = \mathbf{e}_i\}|$. The prediction variance of estimator $\mathbf{w}^*_\lambda(S)$ is equal to

$$\begin{aligned}
\text{tr}\big(\text{Var}_{\boldsymbol{\xi}}[\mathbf{X}\mathbf{w}^*_\lambda(S)]\big) &= \sigma^2\text{tr}(\mathbf{X}(\mathbf{Z}_\lambda(S)^{-1} - \lambda\mathbf{Z}_\lambda(S)^{-2})\mathbf{X}^\top) \\
&= \frac{\sigma^2 n}{d}\sum_{i=1}^d\left(\frac{1}{s_i + \lambda} - \frac{\lambda}{(s_i + \lambda)^2}\right) = \frac{\sigma^2 n}{d}\sum_{i=1}^d\frac{s_i}{(s_i + \lambda)^2}.
\end{aligned}$$

The prediction bias of estimator $\mathbf{w}^*_\lambda(S)$ is equal to

$$\begin{aligned}
\|\mathbf{X}(\mathbb{E}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)] - \widetilde{\mathbf{w}})\|^2 &= \lambda^2\widetilde{\mathbf{w}}^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{X}^\top\mathbf{X}\mathbf{Z}_\lambda(S)^{-1}\widetilde{\mathbf{w}} \\
&= \frac{\lambda^2 a^2\sigma^2 n}{d}\,\text{tr}\big(\mathbf{Z}_\lambda(S)^{-2}\big) = \frac{\lambda^2 a^2\sigma^2 n}{d}\sum_{i=1}^d\frac{1}{(s_i + \lambda)^2}.
\end{aligned}$$

Thus, MSPE of estimator $\mathbf{w}^*_\lambda(S)$ is given by:

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\xi}}\left[\frac{1}{n}\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\right] &= \frac{1}{n}\text{tr}\big(\text{Var}_{\boldsymbol{\xi}}[\mathbf{X}\mathbf{w}^*_\lambda(S)]\big) + \frac{1}{n}\|\mathbf{X}(\mathbb{E}_{\boldsymbol{\xi}}[\mathbf{w}^*_\lambda(S)] - \widetilde{\mathbf{w}})\|^2 \\
&= \frac{\sigma^2}{d}\sum_{i=1}^d\left(\frac{s_i}{(s_i + \lambda)^2} + \frac{a^2\lambda^2}{(s_i + \lambda)^2}\right) = \frac{\sigma^2}{d}\sum_{i=1}^d\frac{s_i + a^2\lambda^2}{(s_i + \lambda)^2}.
\end{aligned}$$

Next, we find the λ that minimizes this expression. Taking the derivative with respect to λ we get:

$$\frac{\partial}{\partial\lambda}\left(\frac{\sigma^2}{d}\sum_{i=1}^d\frac{s_i + a^2\lambda^2}{(s_i + \lambda)^2}\right) = \frac{\sigma^2}{d}\sum_{i=1}^d\frac{2s_i(\lambda - a^{-2})}{(s_i + \lambda)^3}.$$

Thus, since at least one s_i has to be greater than 0, for any set S the derivative is negative for $\lambda < a^{-2}$ and positive for $\lambda > a^{-2}$, and the unique minimum of MSPE is achieved at $\lambda = a^{-2}$, regardless of which subset S is chosen. So, as we are seeking a lower bound, we can focus on the case of $\lambda = a^{-2}$.

Proof of part a. Let $a = 1$. As shown above, we can assume that $\lambda = 1$. In this case the formula simplifies to:

$$\mathbb{E}_{\boldsymbol{\xi}}\Big[\frac{1}{n}\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\Big] = \frac{\sigma^2}{d}\sum_{i=1}^{d}\frac{s_i+1}{(s_i+1)^2} = \frac{\sigma^2}{d}\sum_{i=1}^{d}\frac{1}{s_i+1}$$
$$\overset{(*)}{\geq} \frac{\sigma^2}{\frac{s}{d}+1} = \frac{\sigma^2 d}{s+d} \geq \frac{\sigma^2 d_\lambda}{s+d_\lambda},$$

where $(*)$ follows by applying Jensen's inequality to convex function $\phi(x) = \frac{1}{x+1}$.

Proof of part b. Let $a = \sqrt{2d}$. As shown above, we can assume that $\lambda = 1/(2d)$. Suppose that multiset S is sampled i.i.d. from some distribution over set $\{1..n\}$. Similarly as in Corollary 13, we exploit the Coupon Collector's problem, i.e. that if $|S| \leq (d-1)\ln(d)$, then with probability at least $1/2$ there is $i \in \{1..d\}$ such that $s_i = 0$ (i.e., one of the unit vectors $\mathbf{e}_i$ was never selected). Thus, MSPE can be lower-bounded as follows:

$$\mathbb{E}_S\mathbb{E}_{\boldsymbol{\xi}}\Big[\frac{1}{n}\|\mathbf{X}(\mathbf{w}^*_\lambda(S) - \widetilde{\mathbf{w}})\|^2\Big] \geq \frac{1}{2}\,\frac{\sigma^2}{d}\,\frac{s_i + a^2\lambda^2}{(s_i+\lambda)^2} = \frac{\sigma^2}{2d}\,\frac{2d\lambda^2}{\lambda^2} = \sigma^2. \qquad \blacksquare$$

5. Efficient Algorithms for Volume Sampling

In this section we propose algorithms for efficiently performing volume sampling. This addresses the question posed by Avron and Boutsidis (2013), asking for a polynomial-time algorithm for the case when the size of set S is $s > d$. Deshpande and Rademacher (2010) gave an algorithm for the case when $s = d$, which was later improved by Guruswami and Sinop (2012), running in time $O(nd^3)$. Recently, Li et al. (2017) offered an algorithm for arbitrary s, which has complexity $O(n^4 s)$. We propose two new methods, which use our reverse iterative sampling technique to achieve faster running times for volume sampling of any size s. Both algorithms apply to the more general setting of λ-regularized volume sampling (described in Section 4), and produce standard volume sampling as a special case for $\lambda = 0$ and $s \geq d$. The first algorithm has a deterministic runtime of $O((n-s+d)nd)$, whereas the second one is an accelerated version which with high probability finishes in time $O(nd^2)$. Thus, we obtain a direct improvement over Li et al. (2017) by a factor of at least n^2, and in the special case of $s = d$, by a factor of d over the algorithm of Guruswami and Sinop (2012).

Our algorithms implement reverse iterative sampling from Theorem 2. We start with the full index set $S = \{1..n\}$. In one step of the algorithm, we remove one row from set S. After removing q rows, we are left with the index set of size $n - q$ that is distributed according to volume sampling for row set size $n - q$, and we proceed until our set S has the desired size s. The primary cost of the procedure is updating the conditional distribution $P(S_{-i}|S)$ at every step. It is convenient to store it using the unnormalized weights defined in (12) which, via Sylvester's theorem, can be computed as $h_i = 1 - \mathbf{x}_i^\top(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}\mathbf{x}_i$ (For the sake of generality we state the methods for λ-regularized

volume sampling). Doing this naively, we would first compute $(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}$ which takes $O(nd^2)$ time[12]. After that for each i, we would multiply this matrix by $\mathbf{x}_i$ in time $O(d^2)$ to get the h_i's. The overall runtime of this naive method becomes:

$$\overbrace{\#\text{ of steps}}^{n-s} \times \Big(\overbrace{\text{compute } (\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}}^{O(nd^2)} + \overbrace{\#\text{ of weights}}^{\leq n} \times \overbrace{\text{compute } h_i}^{O(d^2)}\Big) \;=\; O((n-s)nd^2).$$

Both the computation of matrix inverse and the weights h_i can be made more efficient. First, the matrix $(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}$ can be computed from the one obtained in the previous step by using the Sherman-Morrison formula. This lets us update it in $O(d^2)$ time instead of $O(nd^2)$. Furthermore, we propose two strategies for dealing with the cost of maintaining the weights:

a) Update all h_i's at every step using Sherman-Morrison;

b) Use rejection sampling and only compute the h_i's needed for the rejection trials (This avoids computing all h_i's, but makes the computation of each needed h_i more expensive).

As we can see, there is a trade-off between those strategies. In the following lemma, we will show that updating the value of h_i, given its value in the previous step only costs $O(d)$ time as opposed to $O(d^2)$. However, the number of h_i's that need to be computed for rejection sampling (explained shortly) can be far smaller.

Lemma 19 *For any matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$, set $S \subseteq \{1..n\}$ and two distinct indices $i, j \in S$, we have*

$$1 - \mathbf{x}_j^\top(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}} + \lambda\mathbf{I})^{-1}\mathbf{x}_j = h_j - (\mathbf{x}_j^\top\mathbf{v})^2,$$

where $h_j = 1 - \mathbf{x}_j^\top(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}\mathbf{x}_j$ and $\mathbf{v} = \frac{1}{\sqrt{h_i}}(\mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I})^{-1}\mathbf{x}_i$.

Proof Letting $\mathbf{Z}_\lambda(S) = \mathbf{X}_S^\top\mathbf{X}_S + \lambda\mathbf{I}$, we have

$$\begin{aligned}
h_j - (\mathbf{x}_j^\top\mathbf{v})^2 &= 1 - \mathbf{x}_j^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_j - \frac{(\mathbf{x}_j^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i)^2}{1 - \mathbf{x}_i^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i} \\
&= 1 - \mathbf{x}_j^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_j - \frac{\mathbf{x}_j^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_j}{1 - \mathbf{x}_i^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i} \\
&= 1 - \mathbf{x}_j^\top\left(\mathbf{Z}_\lambda(S)^{-1} + \frac{\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i\mathbf{x}_i^\top\mathbf{Z}_\lambda(S)^{-1}}{1 - \mathbf{x}_i^\top\mathbf{Z}_\lambda(S)^{-1}\mathbf{x}_i}\right)\mathbf{x}_j \\
&\overset{(*)}{=} 1 - \mathbf{x}_j^\top(\mathbf{X}_{S_{-i}}^\top\mathbf{X}_{S_{-i}} + \lambda\mathbf{I})^{-1}\mathbf{x}_j,
\end{aligned}$$

where $(*)$ follows from the Sherman-Morrison formula. $\blacksquare$

Thus the overall time complexity of reverse iterative sampling when using the first strategy goes down by a factor of d compared to the naive version (except for an initialization cost which stays at $O(nd^2)$).

Theorem 20 *Algorithm RegVol produces an index set S of rows distributed according to λ-regularized size s volume sampling over $\mathbf{X}$ in time $O((n-s+d)nd)$.*

12. We are primarily interested in the case where $n \geq d$ and we state our time bounds under that assumption. However, when $\lambda > 0$, our techniques can be easily adapted to the case of $n < d$.

Proof Using Lemma 19 for h_i and the Sherman-Morrison formula for $\mathbf{Z}$, the following invariants hold at the beginning of the **while** loop:

$$h_i = 1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1} \mathbf{x}_i \qquad \text{and} \qquad \mathbf{Z} = (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}.$$

Runtime: Computing the initial $\mathbf{Z} = (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}$ takes $O(nd^2)$, as does computing the initial values of h_j's. Inside the **while** loop, updating h_j's takes $O(|S|d) = O(nd)$ and updating $\mathbf{Z}$ takes $O(d^2)$. The overall runtime becomes $O(nd^2 + (n-s)nd) = O((n-s+d)nd)$. ■

Algorithm 2 RegVol($\mathbf{X}, s, \lambda$)

$\mathbf{Z} \leftarrow (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}$
$\forall_{i \in \{1..n\}} \quad h_i \leftarrow 1 - \mathbf{x}_i^\top \mathbf{Z} \mathbf{x}_i$
$S \leftarrow \{1..n\}$
while $|S| > s$
  Sample $i \propto h_i$ out of S
  $S \leftarrow S - \{i\}$
  $\mathbf{v} \leftarrow \mathbf{Z}\mathbf{x}_i / \sqrt{h_i}$
  $\forall_{j \in S} \quad h_j \leftarrow h_j - (\mathbf{x}_j^\top \mathbf{v})^2$
  $\mathbf{Z} \leftarrow \mathbf{Z} + \mathbf{v}\mathbf{v}^\top$
end
return S

Algorithm 3 FastRegVol($\mathbf{X}, s, \lambda$)

$\mathbf{Z} \leftarrow (\mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I})^{-1}$
$S \leftarrow \{1..n\}$
while $|S| > \max\{s, 2d\}$
  repeat
    Sample i uniformly out of S
    $h_i \leftarrow 1 - \mathbf{x}_i^\top \mathbf{Z} \mathbf{x}_i$
    Sample $A \sim \text{Bernoulli}(h_i)$
  until $A = 1$
  $S \leftarrow S - \{i\}$
  $\mathbf{Z} \leftarrow \mathbf{Z} + h_i^{-1} \mathbf{Z}\mathbf{x}_i\mathbf{x}_i^\top \mathbf{Z}$
end
if $s < 2d$, $\quad S \leftarrow \text{RegVol}(\mathbf{X}_S, s, \lambda)$ $\quad$ **end**
return S

Next we present algorithm FastRegVol, which is based on the rejection sampling strategy. Our key observation is that updating the full conditional distribution $P(S_{-i}|S)$ is wasteful, since the distribution changes very slowly throughout the procedure. Moreover, the unnormalized weights h_i, which are computed in the process are all bounded by 1. Thus, to sample from the correct distribution at any given iteration, we can employ rejection sampling as follows:

1: Sample i uniformly from set S,
2: Compute h_i,
3: Accept with probability h_i,
4: Otherwise, draw another sample.

Note that this rejection sampling can be employed locally, within each iteration of the algorithm. Thus, one rejection does not revert us back to the beginning of the algorithm. Moreover, if the probability of acceptance is high, then this strategy requires computing only a small number of weights per iteration of the algorithm, as opposed to updating all of them. This turns out to be the case for a majority of the steps of the algorithm, except at the very end (for $s \leq 2d$), were the conditional probabilities start changing more drastically. At that point, it becomes more efficient to use the first algorithm, RegVol.

Theorem 21 *For any $\lambda, s \geq 0$, and $\delta \in (0,1)$, algorithm FastRegVol samples according to λ-regularized size s volume sampling, and with probability at least $1-\delta$ runs in time*

$$O\Big(\Big(n + \log\big(n/d\big)\log\big(1/\delta\big)\Big)d^2\Big).$$

Proof We analyze the efficiency of rejection sampling in FastRegVol. Let R_t be a random variable corresponding to the number of trials needed in the **repeat** loop from line 4 in FastRegVol at the point when $|S| = t$. Note that conditioning on the algorithm's history, R_t is distributed according to geometric distribution $\text{Ge}(q_t)$ with success probability:

$$q_t = \frac{1}{t}\sum_{i \in S}\big(1 - \mathbf{x}_i^\top (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1}\mathbf{x}_i\big) \geq \frac{t-d}{t} \geq \frac{1}{2}.$$

Thus, even though variables R_t are not themselves independent, they can be upper-bounded by a sequence of independent variables $\widehat{R}_t \sim \text{Ge}(\frac{t-d}{t})$. The expectation of the total number of trials in FastRegVol, $\bar{R} = \sum_t R_t$, can thus be bounded as follows:

$$\mathbb{E}[\bar{R}] \leq \sum_{t=2d}^{n} \mathbb{E}[\widehat{R}_t] = \sum_{t=2d}^{n} \frac{t}{t-d} \leq 2n.$$

Next, we will obtain a similar bound with high probability instead of in expectation. Here, we will have to use the fact that the variables $\widehat{R}_t$ are independent, which means that we can upper-bound their sum with high probability using standard concentration bounds for geometric distribution. For example, using Corollary 2.2 from Janson (2018) one can immediately show that with probability at least $1-\delta$ we have $\bar{R} = O(n \ln \delta^{-1})$. However, more careful analysis shows an even better dependence on δ.

Lemma 22 *Let $\widehat{R}_t \sim \text{Ge}(\frac{t-d}{t})$ be independent random variables. Then, w.p. at least $1-\delta$,*

$$\sum_{t=2d}^{n} \widehat{R}_t = O\Big(n + \log\big(n/d\big)\log\big(1/\delta\big)\Big).$$

Each trial of rejection sampling requires computing one weight h_i in time $O(d^2)$. The overall time complexity of FastRegVol thus includes computation and updating of matrix $\mathbf{Z}$ (in time $O(nd^2)$), rejection sampling which takes $O\left(\left(n + \log\left(\frac{n}{d}\right)\log\left(\frac{1}{\delta}\right)\right)d^2\right)$ time, and (if $s < 2d$) the RegVol portion, taking $O(d^3)$. ■

Proof of Lemma 22 As observed by Janson (2018), tail-bounds for the sum of geometric random variables depend on the minimum acceptance probability among those variables. Note that for the vast majority of $\widehat{R}_t$'s the acceptance probability is very close to 1, so intuitively we should be able to take advantage of this to improve our tail bounds. To that end, we partition the variables into groups of roughly similar acceptance probability and then separately bound the sum of variables in each group. Let $J = \log(\frac{n}{d})$ (w.l.o.g. assume that J is an integer). For $1 \leq j \leq J$, let $I_j = \{d2^j, d2^j + 1, .., d2^{j+1}\}$ represent the j-th partition. We use the following notation for each partition:

$$\bar{R}_j \stackrel{def}{=} \sum_{t \in I_j} R_t, \qquad \mu_j \stackrel{def}{=} \mathbb{E}[\bar{R}_j], \qquad r_j \stackrel{def}{=} \min_{t \in I_j} \frac{t-d}{t}, \qquad \gamma_j \stackrel{def}{=} \frac{\log(\delta^{-1})}{d2^{j-2}} + 3.$$

Now, we apply Theorem 2.3 of Janson (2018) to $\bar{R}_j$, obtaining

$$P(\bar{R}_j \geq \gamma_j \mu_j) \leq \gamma_j^{-1}(1-r_j)^{(\gamma_j - 1 - \ln \gamma_j)\mu_j} \overset{(1)}{\leq} (1-r_j)^{\gamma_j \mu_j / 4} \overset{(2)}{\leq} 2^{-j\gamma_j d 2^{j-2}},$$

where (1) follows since $\gamma_j \geq 3$, and (2) holds because $\mu_j \geq d2^j$ and $r_j \geq 1 - 2^{-j}$. Moreover, for the chosen γ_j we have

$$j\gamma_j d 2^{j-2} = j\log(\delta^{-1}) + 3jd2^{j-2} \geq \log(\delta^{-1}) + j = \log(2^j \delta^{-1}).$$

Let A denote the event that $\bar{R}_j \leq \gamma_j \mu_j$ for all $j \leq J$. Applying union bound, we get

$$P(A) \geq 1 - \sum_{j=1}^{J} P(\bar{R}_j \geq \gamma_j \mu_j) \geq 1 - \sum_{j=1}^{J} 2^{-\log(2^j \delta^{-1})} = 1 - \sum_{j=1}^{J} \frac{\delta}{2^j} \geq 1 - \delta.$$

If A holds, then we obtain the desired bound:

$$\begin{aligned} \sum_{t=2d}^{n} \widehat{R}_t &\leq \sum_{j=1}^{J} \gamma_j \mu_j \leq \sum_{j=1}^{J} \left(\frac{\log(\delta^{-1})}{d2^{j-2}} + 3 \right) d2^{j+1} = 8J\log(\delta^{-1}) + 6\sum_{j=1}^{J} d2^j \\ &= O\Big(\log\left(n/d\right) \log\left(1/\delta\right) + n \Big). \end{aligned}$$

■

6. Experiments

In this section we experimentally evaluate the proposed volume sampling algorithms in terms of runtime and in the task of subsampling for linear regression. We use regularization both for sampling and for prediction, as discussed in Section 4. The list of implemented algorithms is:

a) Regularized volume sampling (algorithms FastRegVol and RegVol),

b) Leverage score sampling[13] (LSS) – a popular i.i.d. sampling technique (Mahoney, 2011), where examples are selected w.p. $P(i) = (\mathbf{x}_i^\top (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{x}_i)/d$.

The experiments were performed on several benchmark linear regression datasets from the libsvm repository (Chang and Lin, 2011). Table 1 lists those datasets along with running times for sampling dimension many columns with each method. Dataset MSD was too big for RegVol to finish in reasonable time, however FastRegVol finished in less than 40 seconds. In Figure 6 we plot the runtime against varying values of n (using portions of the datasets), to compare how FastRegVol and RegVol scale with respect to the data size. We observe that FastRegVol exhibits linear dependence on n, thus it is much better suited for running on large datasets.

Dataset	$n \times d$	RegVol	FastRegVol	LSS
cadata	21k×8	33.5s	0.9s	0.1s
MSD	464k×90	>24hr	39s	12s
cpusmall	8k×12	1.7s	0.4s	0.07s
abalone	4k×8	0.5s	0.2s	0.03s

Table 1: List of regression datasets with runtime comparison between RegVol and FastRegVol. We also provide the runtime for i.i.d. sampling with exact leverage scores (LSS).

13. Regularized variants of leverage scores have also been considered in context of kernel ridge regression Alaoui and Mahoney (2015). However, in our experiments regularizing leverage scores did not provide any improvements.

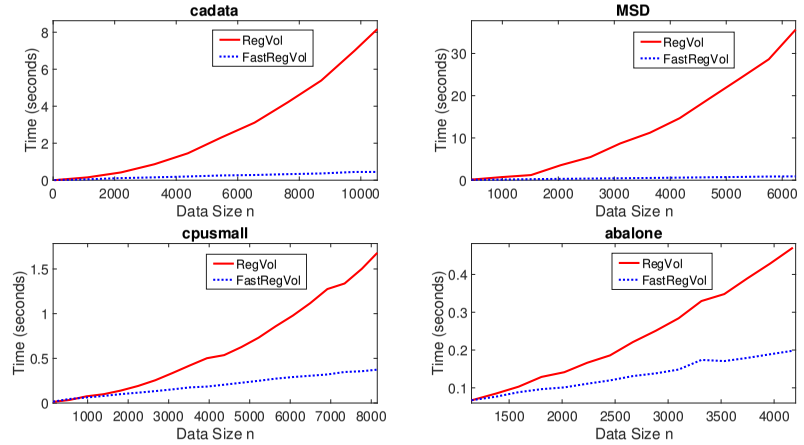


Figure 6: Comparison of runtime between FastRegVol and RegVol on four libsvm regression datasets (Chang and Lin, 2011), with the methods ran on data subsets of varying size (n).

6.1 Subset Selection for Ridge Regression

We applied volume sampling to the task of subset selection for linear regression, by evaluating the subsampled ridge estimator $\mathbf{w}_\lambda^*(S)$ using the average loss over the full dataset, i.e.,

$$\text{Average Loss:} \quad \frac{1}{n}\|\mathbf{X}\mathbf{w}_\lambda^*(S) - \mathbf{y}\|^2, \quad \text{where} \quad \mathbf{w}_\lambda^*(S) = (\mathbf{X}_S^\top \mathbf{X}_S + \lambda \mathbf{I})^{-1} \mathbf{X}_S^\top \mathbf{y}_S.$$

We evaluated the estimators for a range of subset sizes and values of λ, when the subsets are sampled according to λ-regularized volume sampling[14] and leverage score sampling. The results were averaged over 20 runs of each experiment. For clarity, Figure 7 shows the results only with one value of λ for each dataset, chosen so that the subsampled ridge estimator performed best (on average over all samples of preselected size s). Note that for leverage scores we did the appropriate rescaling of the instances before solving for $\mathbf{w}_\lambda^*(S)$ for the sampled subproblems (see Mahoney (2011) for details). Volume sampling does not require any rescaling. The results on all datasets show that when only a small number of responses s is obtainable, then regularized volume sampling offers better estimators than leverage score sampling (as predicted by Theorems 17 and 18). The lower bound from Theorem 18b can be observed for dataset cpusmall, where $d = 12$ and $d \log d \approx 30$.

7. Conclusions

Volume sampling is a joint sampling procedures that produces more diverse samples than i.i.d. sampling. We developed a method for proving exact matrix expectation formulas for volume sampling giving further credence to the fact that this is a fundamental sampling procedure. We also made significant progress on finding an efficient implementation of this sampling procedure: Our

14. Our experiments suggest that using the same λ for sampling and for computing the ridge estimator works best.

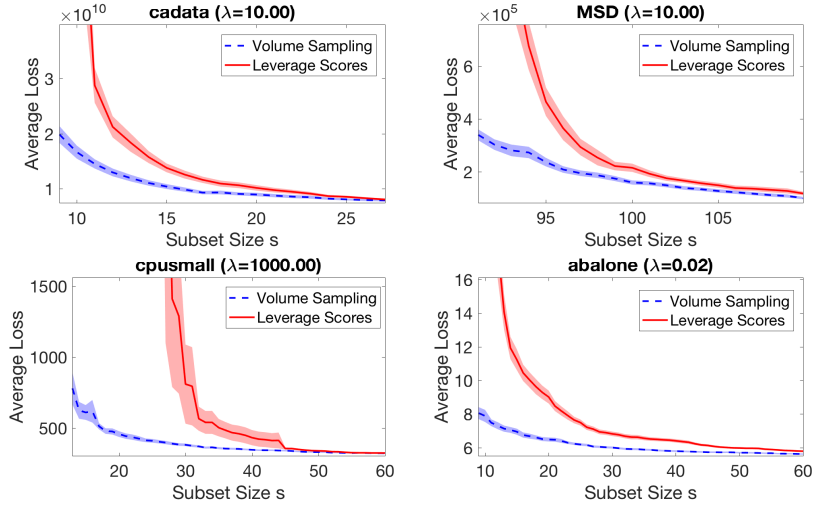


Figure 7: Comparison of loss of the subsampled ridge estimator when using regularized volume sampling vs using leverage score sampling on four datasets.

new reverse iterative volume sampling algorithm runs in time $O(nd^2)$. Note that this running time is within a constant factor of i.i.d. sampling with exact leverage scores and is a remarkable feat since volume sampling was only recently shown to be polynomial (that is $O(n^4 s)$ in Li et al. (2017)).

A final long ranging question is how to generalize volume sampling and the exact matrix expectation formulas to higher order tensors.

Acknowledgments

Thanks to Daniel Hsu and Wojciech Kotłowski for many valuable discussions. This research was supported by NSF grant IIS-1619271.

Appendix A. Inductive Proof of Cauchy-Binet

The most common form of the Cauchy-Binet equation deals with two real $n \times d$ matrices $\mathbf{A}, \mathbf{B}$: $\sum_{S:|S|=d} \det(\mathbf{A}_S^\top \mathbf{B}_S) = \det(\mathbf{A}^\top \mathbf{B})$. It is easy to generalize volume sampling and Theorem 2 to this "asymmetric" version. Here we give an alternate inductive proof.

For $i \in \{1..n\}$, let $\mathbf{a}_i, \mathbf{b}_i$ denote the i-th row of $\mathbf{A}, \mathbf{B}$, respectively. For $S \subseteq \{1..n\}$, $\mathbf{A}_S$ consists of all rows indexed by S, and $\mathbf{A}_{-i}$, all except for the i-th row.

Theorem 23 *For $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times d}$ and $n-1 \geq s \geq d$:*

$$\det(\mathbf{A}^\top \mathbf{B}) = \frac{1}{\binom{n-d}{s-d}} \sum_{S:|S|=s} \det(\mathbf{A}_S^\top \mathbf{B}_S).$$

Proof S is a size s subset of a set of size n. We rewrite the range restriction $n-1 \geq s \geq d$ for size s as $1 \leq n-s \leq n-d$ and induct on $n-s$. For the base case, $n-s=1$ or $s=n-1$, we need to show that

$$\det(\mathbf{A}^\top \mathbf{B}) = \frac{1}{n-d} \sum_{i=1}^{n} \det(\mathbf{A}_{-i}^\top \mathbf{B}_{-i}).$$

This clearly holds if $\det(\mathbf{A}^\top \mathbf{B}) = 0$. Otherwise, by Sylvester's Theorem

$$\sum_{i=1}^{n} \frac{\det(\overbrace{\mathbf{A}_{-i}^\top \mathbf{B}_{-i}}^{\mathbf{A}^\top \mathbf{B} - \mathbf{a}_i \mathbf{b}_i^\top})}{\det(\mathbf{A}^\top \mathbf{B})} = \sum_{i=1}^{n} (1 - \mathbf{a}_i^\top (\mathbf{A}^\top \mathbf{B})^{-1} \mathbf{b}_i) = n - \overbrace{\mathrm{tr}((\mathbf{A}^\top \mathbf{B})^{-1} \mathbf{A}^\top \mathbf{B})}^{d}.$$

Induction: Assume $2 \leq n-s \leq n-d$.

$$\begin{aligned}
\det(\mathbf{A}^\top \mathbf{B}) &\overset{\text{base case}}{=} \frac{1}{n-d} \sum_{i=1}^{n} \det(\mathbf{A}_{-i}^\top \mathbf{B}_{-i}) \\
&\overset{\text{ind. step}}{=} \frac{1}{n-d} \sum_{i=1}^{n} \sum_{S:|S|=s,\, i\notin S} \frac{1}{\binom{n-1-d}{s-d}} \det(\mathbf{A}_S^\top \mathbf{B}_S) \\
&= \underbrace{\frac{n-s}{n-d} \frac{1}{\binom{n-1-d}{s-d}}}_{\frac{1}{\binom{n-d}{s-d}}} \sum_{S:|S|=s} \det(\mathbf{A}_S^\top \mathbf{B}_S).
\end{aligned}$$

Note that for the induction step, S is a subset of size s from a set of size $n-1$ and we have the range restriction $1 \leq n-1-s \leq n-1-d$. Clearly, $n-1-s$ is one smaller than $n-s$. For the last equality, notice that each set $S : |S| = s$ is counted $n-s$ times in the double sum. ■

Appendix B. Alternate Proof of Theorem 5

We make use of the following derivative for determinants by Petersen and Pedersen (2012):

$$\text{For symmetric } \mathbf{C}: \quad \frac{\partial \det(\mathbf{X}^\top \mathbf{C} \mathbf{X})}{\partial \mathbf{X}} = 2 \det(\mathbf{X}^\top \mathbf{C} \mathbf{X}) \mathbf{C} \mathbf{X} (\mathbf{X}^\top \mathbf{C} \mathbf{X})^{-1}.$$

The proof begins with generalized Cauchy-Binet for size s volume sampling:

$$\sum_S \det(\mathbf{X}^\top \mathbf{I}_S \mathbf{X}) = \binom{n-d}{s-d} \det(\mathbf{X}^\top \mathbf{X}).$$

Now, we take a derivative w.r.t. $\mathbf{X}$ on both sides

$$\begin{aligned}
&\sum_S 2 \det(\mathbf{X}^\top \mathbf{I}_S \mathbf{X}) \ (\mathbf{I}_S \mathbf{X})^{+\top} = \binom{n-d}{s-d} \ 2 \det(\mathbf{X}^\top \mathbf{X}) \ \mathbf{X}^{+\top} \\
\iff \quad &\underbrace{\sum_S \frac{\det(\mathbf{X}^\top \mathbf{I}_S \mathbf{X})}{\binom{n-d}{s-d} \det(\mathbf{X}^\top \mathbf{X})} \ (\mathbf{I}_S \mathbf{X})^{+\top}}_{\mathbb{E}[(\mathbf{I}_S \mathbf{X})^{+\top}]} = \mathbf{X}^{+\top}.
\end{aligned}$$

Appendix C. Proof of Proposition 14

The main idea behind the proof is to construct variants of the input matrix $\mathbf{X}$ and relate their volumes. We use the following standard properties of the determinant:

Proposition 24 *For any matrix* $\mathbf{M}$, $\det(\mathbf{M}^\top\mathbf{M}) = \det(\widetilde{\mathbf{M}}^\top\widetilde{\mathbf{M}})$ *where* $\widetilde{\mathbf{M}}$ *is produced from* $\mathbf{M}$ *through the following operations:*

a) $\widetilde{\mathbf{M}}$ *equals* $\mathbf{M}$ *except that column* $\mathbf{m}_j$ *is replaced by* $\mathbf{m}_j + \alpha\mathbf{m}_i$*, where* $\mathbf{m}_i$ *is another column of* $\mathbf{M}$*;*

b) $\widetilde{\mathbf{M}}$ *equals* $\mathbf{M}$ *except that two rows are swapped.*

Recall that our goal is to prove the following formula for any $\mathbf{X}, \mathbf{y}$ and $i \in \{1..n\}$:

$$\det(\mathbf{X}^\top\mathbf{X})\left(L(\mathbf{w}^*(-i)) - L(\mathbf{w}^*)\right) = \left(\det(\mathbf{X}^\top\mathbf{X}) - \det(\mathbf{X}_{-i}^\top\mathbf{X}_{-i})\right)\ell_i(\mathbf{w}^*(-i)).$$

By Proposition 24b, we can assume w.l.o.g. that $i = n$, i.e. that the test row in Proposition 14 is the last row of $\mathbf{X}$. As discussed in Section 3.3, the columns of $\mathbf{X}$ are the feature vectors, denoted by $\mathbf{f}_1, \ldots, \mathbf{f}_d$. Moreover, the optimal prediction vector on the full dataset, $\widehat{\mathbf{y}} = \mathbf{X}\mathbf{w}^*$, is a projection of $\mathbf{y}$ onto the subspace spanned by the features/columns of $\mathbf{X}$, denoted as $\widehat{\mathbf{y}} = \mathbf{P}_\mathbf{X}\,\mathbf{y}$. Let us define a vector $\overline{\mathbf{y}}$ as

$$\overline{\mathbf{y}}^\top \stackrel{def}{=} (\text{——}\widehat{\mathbf{y}}_{-n}^\top\text{——}, y_n), \tag{18}$$

where $\widehat{\mathbf{y}}_{-n} \stackrel{def}{=} \mathbf{X}_{-n}\mathbf{w}^*(-n)$ is the optimal prediction vector for the training problem $(\mathbf{X}_{-n}, \mathbf{y}_{-n})$. Note, that if $\operatorname{rank}(\mathbf{X}_{-n}) < d$, then $\mathbf{w}^*(-n)$ may not be unique, but we can pick any weight vector as long as it minimizes the loss on the training set $\{1..n-1\}$. Next, we show the following lemma:

Lemma 25 *The best achievable loss for the problem* $(\mathbf{X}, \mathbf{y})$ *can be decomposed as follows:*

$$L(\mathbf{w}^*) = L(\mathbf{w}^*(-n)) - \ell_n(\mathbf{w}^*(-n)) + \|\overline{\mathbf{y}} - \widehat{\mathbf{y}}\|^2. \tag{19}$$

Proof First, we will show that $\overline{\mathbf{y}}$ is the projection of $\mathbf{y}$ onto the subspace spanned by all features and the unit vector $\mathbf{e}_n \in \mathbb{R}^n$ (where n corresponds to the test row). That is, we want to show that $\overline{\mathbf{y}} = \mathbf{P}_{(\mathbf{X},\mathbf{e}_n)}\,\mathbf{y}$. Denote $\widetilde{\mathbf{y}}$ as that projection. Observe that $\widetilde{y}_n = y_n$, because if this was not true, we could construct a vector $\widetilde{\mathbf{y}} + (y_n - \widetilde{y}_n)\mathbf{e}_n$ that is closer to $\mathbf{y}$ than $\widetilde{\mathbf{y}}$ and lies in $\operatorname{span}(\mathbf{X}, \mathbf{e}_n)$. Thus, the projection does not incur any loss along the n-th dimension and can be reduced to the remaining $n-1$ dimensions, which corresponds to solving the training problem $(\mathbf{X}_{-n}, \mathbf{y}_{-n})$. Using the definition of $\overline{\mathbf{y}}$ in (18), this shows that $\widetilde{\mathbf{y}} = \mathbf{P}_{(\mathbf{X},\mathbf{e}_n)}\,\mathbf{y}$ equals $\overline{\mathbf{y}}$.

Next, we will show that $\widehat{\mathbf{y}}$ is the projection of $\overline{\mathbf{y}}$ onto $\operatorname{span}(\mathbf{X})$, i.e. that $\mathbf{P}_\mathbf{X}\,\overline{\mathbf{y}} = \widehat{\mathbf{y}}$. By the linearity of projection, we have

$$\begin{aligned}\mathbf{P}_\mathbf{X}\,\overline{\mathbf{y}} &= \mathbf{P}_\mathbf{X}(\overline{\mathbf{y}} - \mathbf{y} + \mathbf{y}) \\ &= \mathbf{P}_\mathbf{X}(\overline{\mathbf{y}} - \mathbf{y}) + \mathbf{P}_\mathbf{X}\,\mathbf{y} \\ &= \mathbf{P}_\mathbf{X}(\overline{\mathbf{y}} - \mathbf{y}) + \widehat{\mathbf{y}}.\end{aligned}$$

We already showed that $\overline{\mathbf{y}} = \mathbf{P}_{(\mathbf{X},\mathbf{e}_n)}\,\mathbf{y}$. Therefore, the vector $\overline{\mathbf{y}} - \mathbf{y}$ is orthogonal to the column vectors of $\mathbf{X}$, and thus $\mathbf{P}_\mathbf{X}(\overline{\mathbf{y}} - \mathbf{y}) = 0$. This shows that $\mathbf{P}_\mathbf{X}\,\overline{\mathbf{y}} = \widehat{\mathbf{y}}$.

Finally, note that since $\overline{\mathbf{y}}$ is the projection of $\mathbf{y}$ onto $\operatorname{span}(\mathbf{X}, \mathbf{e}_n)$ and $\widehat{\mathbf{y}} \in \operatorname{span}(\mathbf{X}, \mathbf{e}_n)$, vector $\overline{\mathbf{y}} - \mathbf{y}$ is orthogonal to vector $\overline{\mathbf{y}} - \widehat{\mathbf{y}}$ and by the Pythagorean Theorem we have

$$\|\widehat{\mathbf{y}} - \mathbf{y}\|^2 = \|\overline{\mathbf{y}} - \mathbf{y}\|^2 + \|\overline{\mathbf{y}} - \widehat{\mathbf{y}}\|^2.$$

Using the definition of $\overline{\mathbf{y}}$ in (18), we have

$$\|\overline{\mathbf{y}} - \mathbf{y}\|^2 = \|\widehat{\mathbf{y}}_{-n} - \mathbf{y}_{-n}\|^2 = L(\mathbf{w}^*(-n)) - \ell_n(\mathbf{w}^*(-n)),$$

concluding the proof of the lemma. ■

Proof of Proposition 14 We construct a matrix $\overline{\mathbf{X}}$, adding vector $\overline{\mathbf{y}}$ as an extra column to matrix $\mathbf{X}$:

$$\overline{\mathbf{X}} \stackrel{def}{=} (\mathbf{X}\,,\,\overline{\mathbf{y}}) = \left(\begin{array}{c|c} \mathbf{X}_{-n} & \widehat{\mathbf{y}}_{-n} \\ \hline \mathbf{x}_n^\top & y_n \end{array}\right). \tag{20}$$

Applying "base $\times$ height" and Lemma 25, we compute the volume spanned by $\overline{\mathbf{X}}$:

$$\det(\overline{\mathbf{X}}^\top\overline{\mathbf{X}}) = \det(\mathbf{X}^\top\mathbf{X})\,\|\overline{\mathbf{y}} - \widehat{\mathbf{y}}\|^2 = \det(\mathbf{X}^\top\mathbf{X})\,(L(\mathbf{w}^*) - L(\mathbf{w}^*(-n)) + \ell_n(\mathbf{w}^*(-n))). \tag{21}$$

Next, we use the fact that volume is preserved under elementary column operations (Proposition 24a). Note, that prediction vector $\widehat{\mathbf{y}}_{-n}$ is a linear combination of the columns of $\mathbf{X}_{-n}$, with the coefficients given by $\mathbf{w}^*(-n)$. Therefore, looking at the block structure of $\overline{\mathbf{X}}$ (see (20)), we observe that performing column operations on the last column of $\overline{\mathbf{X}}$ with coefficients given by negative $\mathbf{w}^*(-n)$, we can zero out that column except for its last element:

$$\overline{\mathbf{y}} - \mathbf{X}\,\mathbf{w}^*(-n) = r\,\mathbf{e}_n,$$

where $r \stackrel{def}{=} y_n - \mathbf{x}_n^\top\mathbf{w}^*(-n)$ (see transformation (a) in (22)). Now, we consider two cases, depending on whether or not r equals zero. If $r \neq 0$, then we further transform the matrix by a second transformation (b), which zeros out the last row (the test row) using column operations. The entire sequence of operations, resulting in a matrix we call $\overline{\mathbf{X}}_0$, is shown below:

$$\overline{\mathbf{X}} = \left(\begin{array}{c|c} \mathbf{X}_{-n} & \widehat{\mathbf{y}}_{-n} \\ \hline \mathbf{x}_n^\top & y_n \end{array}\right) \stackrel{\text{(a)}}{\to} \left(\begin{array}{c|c} \mathbf{X}_{-n} & 0 \\ \hline \mathbf{x}_n^\top & r \end{array}\right) \stackrel{\text{(b)}}{\to} \left(\begin{array}{c|c} \mathbf{X}_{-n} & 0 \\ \hline 0 & r \end{array}\right) = \overline{\mathbf{X}}_0. \tag{22}$$

Note, that due to the block-diagonal structure of $\overline{\mathbf{X}}_0$, its volume can be easily described by the "base $\times$ height" formula:

$$\det(\overline{\mathbf{X}}_0^\top\overline{\mathbf{X}}_0) = \det(\mathbf{X}_{-n}^\top\mathbf{X}_{-n})\,r^2 = \det(\mathbf{X}_{-n}^\top\mathbf{X}_{-n})\,\ell_n(\mathbf{w}^*(-n)). \tag{23}$$

Since $\det(\overline{\mathbf{X}}^\top\overline{\mathbf{X}}) = \det(\overline{\mathbf{X}}_0^\top\overline{\mathbf{X}}_0)$, we can combine (21) and (23) to obtain the desired result.

Finally, if $r = 0$ we cannot perform transformation (b). However, in this case matrix $\overline{\mathbf{X}}$ has volume 0, and moreover, $\ell_n(\mathbf{w}^*(-n)) = r^2 = 0$, so once again we have

$$\det(\overline{\mathbf{X}}^\top\overline{\mathbf{X}}) = 0 = \det(\mathbf{X}_{-n}^\top\mathbf{X}_{-n})\,\ell_n(\mathbf{w}^*(-n)),$$

which concludes the proof of Proposition 14. ■

References

Ahmed El Alaoui and Michael W. Mahoney. Fast randomized kernel ridge regression with statistical guarantees. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pages 775–783, Montreal, Canada, December 2015.

Anne Auger and Benjamin Doerr. *Theory of Randomized Search Heuristics: Foundations and Recent Developments*. World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2011.

Haim Avron and Christos Boutsidis. Faster subset selection for matrices and applications. *SIAM Journal on Matrix Analysis and Applications*, 34(4):1464–1499, 2013.

Adi Ben-Israel. A volume associated with m x n matrices. *Linear Algebra and its Applications*, 167 (Supplement C):87 – 111, 1992.

Aharon Ben-Tal and Marc Teboulle. A geometric property of the least squares solution of linear equations. *Linear Algebra and its Applications*, 139:165 – 170, 1990.

Christos Boutsidis, Petros Drineas, and Malik Magdon-Ismail. Near-optimal coresets for least-squares regression. *IEEE Trans. Information Theory*, 59(10):6880–6892, 2013.

Nicolo Cesa-Bianchi and Gabor Lugosi. *Prediction, Learning, and Games*. Cambridge University Press, New York, NY, USA, 2006.

Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011.

Xue Chen and Eric Price. Active regression via linear-sample sparsification. *CoRR*, abs/1711.10051, 2017.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 81–90, New York, NY, USA, 2013. ACM.

Michał Dereziński. *Volume sampling for linear regression*. PhD thesis, University of California at Santa Cruz, CA, USA, June 2018.

Michał Dereziński and Manfred K. Warmuth. Unbiased estimates for linear regression via volume sampling. In *Advances in Neural Information Processing Systems 30*, pages 3087–3096, Long Beach, CA, USA, December 2017.

Michał Dereziński and Manfred K. Warmuth. Subsampling for ridge regression via regularized volume sampling. In Amos Storkey and Fernando Perez-Cruz, editors, *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 716–725, Playa Blanca, Lanzarote, Canary Islands, April 2018.

Amit Deshpande and Luis Rademacher. Efficient volume sampling for row/column subset selection. In *Proceedings of the 2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 329–338, Las Vegas, USA, October 2010.

Amit Deshpande, Luis Rademacher, Santosh Vempala, and Grant Wang. Matrix approximation and projective clustering via volume sampling. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithm*, pages 1117–1126, Miami, FL, USA, January 2006.

Petros Drineas, Michael W. Mahoney, and S. Muthukrishnan. Relative-error CUR matrix decompositions. *SIAM J. Matrix Anal. Appl.*, 30(2):844–881, September 2008.

Petros Drineas, Malik Magdon-Ismail, Michael W. Mahoney, and David P. Woodruff. Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, 13(1):3475–3506, December 2012.

Valerii V Fedorov. *Theory of optimal experiments*. Probability and mathematical statistics. Academic Press, New York, NY, USA, 1972.

Mike Gartrell, Ulrich Paquet, and Noam Koenigstein. Bayesian low-rank determinantal point processes. In *Proceedings of the 10th ACM Conference on Recommender Systems*, pages 349–356, Boston, MA, USA, September 2016.

Venkatesan Guruswami and Ali K. Sinop. Optimal column-based low-rank matrix reconstruction. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1207–1214, Kyoto, Japan, January 2012.

Lars Holst. Extreme value distributions for random coupon collector and birthday problems. *Extremes*, 4(2):129–145, 2001.

Daniel Hsu. Leverage scores and linear regression. Private communication, March 2017.

Svante Janson. Tail bounds for sums of geometric and exponential variables. *Statistics and Probability Letters*, 135:1 – 6, 2018.

Byungkon Kang. Fast determinantal point process sampling with application to clustering. In *Proceedings of the 26th International Conference on Neural Information Processing Systems*, NIPS'13, pages 2319–2327, USA, 2013.

Alex Kulesza and Ben Taskar. k-DPPs: Fixed-Size Determinantal Point Processes. In *Proceedings of the 28th International Conference on Machine Learning*, pages 1193–1200, Bellevue, WA, USA, June 2011.

Alex Kulesza and Ben Taskar. *Determinantal Point Processes for Machine Learning*. Now Publishers Inc., Hanover, MA, USA, 2012.

Chengtao Li, Stefanie Jegelka, and Suvrit Sra. Polynomial time algorithms for dual volume sampling. In *Advances in Neural Information Processing Systems 30*, pages 5045–5054, Long Beach, CA, USA, December 2017.

Michael W. Mahoney. Randomized algorithms for matrices and data. *Found. Trends Mach. Learn.*, 3 (2):123–224, February 2011.

Kaare B. Petersen and Michael S. Pedersen. The matrix cookbook, November 2012. URL `http://www2.imm.dtu.dk/pubdb/p.php?3274`. Version 20121115.

Tamas Sarlos. Improved approximation algorithms for large matrices via random projections. In *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '06, pages 143–152, Washington, DC, USA, 2006. IEEE Computer Society.

Masashi Sugiyama and Shinichi Nakajima. Pool-based active learning in approximate linear regression. *Mach. Learn.*, 75(3):249–274, June 2009.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, August 2012.

Journal of Machine Learning Research 19 (2018) 1-40 Submitted 1/18; Revised 7/18; Published 7/18

Universal discrete-time reservoir computers with stochastic inputs and linear readouts using non-homogeneous state-affine systems

Lyudmila Grigoryeva Lyudmila.Grigoryeva@uni-konstanz.de
Department of Mathematics and Statistics
Graduate School of Decision Sciences
Universität Konstanz
Germany

Juan-Pablo Ortega Juan-Pablo.Ortega@unisg.ch
Faculty of Mathematics and Statistics
Universität Sankt Gallen
Switzerland
Centre National de la Recherche Scientifique (CNRS)
France

Editor: Yoshua Bengio

Abstract

A new class of non-homogeneous state-affine systems is introduced for use in reservoir computing. Sufficient conditions are identified that guarantee first, that the associated reservoir computers with linear readouts are causal, time-invariant, and satisfy the fading memory property and second, that a subset of this class is universal in the category of fading memory filters with stochastic almost surely uniformly bounded inputs. This means that any discrete-time filter that satisfies the fading memory property with random inputs of that type can be uniformly approximated by elements in the non-homogeneous state-affine family.

Keywords: reservoir computing, universality, state-affine systems, SAS, echo state networks, ESN, echo state affine systems, machine learning, fading memory property, linear training, stochastic signal treatment

1. Introduction

A ***reservoir computer (RC)*** (Jaeger (2010), Jaeger and Haas (2004), Maass et al. (2002), Maass (2011), Crook (2007), Verstraeten et al. (2007), Lukoševičius and Jaeger (2009)) or a ***RC system*** is a specific type of recurrent neural network determined by two maps, namely a ***reservoir*** $F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$, $n, N \in \mathbb{N}$, and a ***readout*** map $h : \mathbb{R}^N \to \mathbb{R}$ that under certain hypotheses transform (or filter) an infinite discrete-time input $\mathbf{z} = (\ldots, \mathbf{z}_{-1}, \mathbf{z}_0, \mathbf{z}_1, \ldots) \in (\mathbb{R}^n)^{\mathbb{Z}}$ into an output signal $\mathbf{y} \in \mathbb{R}^{\mathbb{Z}}$ of the same type using the state-space transformation given by:

$$\begin{cases} \mathbf{x}_t = F(\mathbf{x}_{t-1}, \mathbf{z}_t), & (1.1) \\ y_t = h(\mathbf{x}_t), & (1.2) \end{cases}$$

where $t \in \mathbb{Z}$ and the dimension $N \in \mathbb{N}$ of the state vectors $\mathbf{x}_t \in \mathbb{R}^N$ will be referred to as the number of virtual ***neurons*** of the system. The expressions (1.1)-(1.2) determine a nonlinear state-space system and many of its dynamical properties (stability, controlability) have been studied for decades in the literature from that point of view.

This notion of reservoir computer (also known as ***liquid state machine***) is a significant generalization of the definitions found in the literature, where the readout map h is consistently taken to be linear. In many supervised machine learning applications, the reservoir map is randomly generated (see, for instance, the echo state networks in Jaeger (2010), Jaeger and Haas (2004)) and the memoryless readout is trained so that the output matches a given ***teaching signal*** that we denote by $\mathbf{d} \in \mathbb{R}^{\mathbb{Z}}$. Two important advantages of this approach lay on the fact that they reduce the training of a dynamic task to a static problem and, moreover, if the reservoir map is *rich* enough, good performances can be indeed attained with just linear readouts that are trained via a (eventually regularized) linear regression that minimizes the Euclidean distance between the output $\mathbf{y}$ and the teaching signal $\mathbf{d}$. These features circumvent well-known difficulties in the training of generic recurrent neural networks having to do with bifurcation phenomena (Doya (1992)) and that, despite recent progress in the regularization and training of deep RNN structures (see, for instance Graves et al. (2013), Pascanu et al. (2013), Zaremba et al. (2014), and references therein), render classical gradient descent methods non-convergent.

The interest for reservoir computing in both the machine learning and the signal processing communities has strongly increased in the last years. One of the main reasons for this fact is that some RC implementations are based on the computational capacities of certain non-neural dynamical systems (Crutchfield et al. (2010)), which opens the door to physical (optical or optoelectronic) realizations that have already been built using dedicated hardware (see, for instance, Jaeger et al. (2007), Atiya and Parlos (2000), Appeltant et al. (2011), Rodan and Tino (2011), Vandoorne et al. (2011), Larger et al. (2012), Paquot et al. (2012), Brunner et al. (2013), Vandoorne et al. (2014), Vinckier et al. (2015)) and that have shown unprecedented information processing speeds.

There are two central questions that need to be addressed when designing a machine learning paradigm, namely, the ***capacity*** and the ***universality*** problems. The capacity problem concerns generically the estimation of the error that is going to be committed in the execution of a specific task. In statistical learning and in the approximation theoretical treatment of static neural networks, this estimation has taken the form of generic bounds that incorporate various architecture parameters of the system like in Pisier (1981), Jones (1992), Barron (1993), Kurkova and Sanguineti (2005). In the specific context of reservoir computing, and in dynamic learning in general, one is interested in various notions of memory capacity that have been the subject of much research (Jaeger (2002), White et al. (2004), Ganguli et al. (2008), Hermans and Schrauwen (2010), Dambre et al. (2012), Grigoryeva et al. (2015), Couillet et al. (2016), Grigoryeva et al. (2016a)).

The universality problem consists in showing that the set of input/output functionals that can be generated with a specific architecture is dense in a sufficiently rich class, like the one containing, for example, all continuous or even all measurable functionals. For classical machine learning paradigms like neural networks, this question has given rise to well-known results that show that they can be considered as universal approximators in a static and deterministic setup (see, for instance, Kolmogorov (1956), Arnold (1957), Sprecher (1965, 1996, 1997), Cybenko (1989), Hornik et al. (1989), Rüschendorf and Thomsen (1998)).

There is no general recipe that allows one to conclude the universality of a given machine learning approach. The proof strategy depends much on the specific paradigm and, more importantly, on the nature of the inputs and the outputs. In the context of reservoir computing there are several situations for which universality has been established when the inputs/outputs are deterministic. There are two features that influence significantly the level of mathematical sophistication that is needed to conclude universality: first, the compactness of the time domain under consideration and second, if one works in continuous or discrete time. In the following paragraphs we briefly review the results that have already been obtained and, in passing, we present and put in context the contributions contained in this paper.

The compactness of the time domain is crucial because, as we will see later on, universality can be obtained as a consequence of various versions of the Stone-Weierstrass Theorem, which are invariably formulated for functions defined on a compact metric space. When the time domain is compact, this property is naturally inherited by the spaces relevant in the proofs. However, when it is not, it can still be secured using functionals that satisfy a condition introduced in Boyd and Chua (1985) known as the ***fading memory property***. The distinction between continuous and discrete time inputs is justified by the availability in the continuous setup of different tools coming from functional analysis that do not exist for discrete time.

Reservoir computing universality for compact time domains is a corollary of classical results in systems theory. Indeed, in the continuous time setup, it can be established for linear systems using polynomial readouts and for bilinear systems using linear readouts (see Fliess (1976), Sussmann (1976)). In the discrete-time setup, the situation is more convoluted when the readout is linear and required the introduction in Fliess and Normand-Cyrot (1980) of the so-called (homogeneous) ***state-affine systems (SAS)*** (see also Sontag (1979a,b)). The extension of these results to continuous non-compact time intervals was carried out in Boyd and Chua (1985) for fading memory filters using exponentially stable linear RCs with polynomial readouts and their bilinear counterparts with linear readouts (see also Maass and Sontag (2000) and Maass et al. (2002, 2004, 2007)). An extension to the non-compact discrete-time setup based on the Stone-Weierstrass theorem is, to our knowledge, not available in the literature and it is one of the main contributions of this paper. This problem has only been tackled from an internal approximation point of view, which consists in uniformly approximating the reservoir and readout maps in (1.1)-(1.2) in order to obtain an approximation of the resulting filter; this strategy has been introduced in Matthews (1992, 1993) for absolutely summable systems. The proofs in those works were unfortunately based on an invalid compactness assumption. Even though corrections were proposed in Perryman (1996) and in Stubberud and Perryman (1997b), this approach yields, in the best of cases, universality only within the reservoir filter category, while we aim at proving that statement in the much larger category of fading memory filters.

The paper is structured in three sections:

- All the notation and main definitions which are used later on in the paper are provided in Section 2. Important concepts like filters, reservoir filters, and the fading memory property are discussed.
- Section 3 contains two different universality results. The first one in Subsection 3.1 shows that the entire family of fading memory RCs itself is universal, as well as the much smaller one containing all the linear reservoirs with polynomial readouts, when certain spectral restrictions are imposed on the reservoir matrices (see below for details). The second universality result is contained in Subsection 3.2 and is one of the main contributions of the paper. Here we restrict ourselves to reservoir computers with linear readouts which are closer to the type of RCs used in applications. We introduce a non-homogeneous variant of the state-affine systems in Fliess and Normand-Cyrot (1980) and identify sufficient conditions that guarantee that the associated reservoir computers with linear readouts are causal, time-invariant, and satisfy the echo state and the fading memory properties. Finally, we state a universality result for a subset of this class which is shown to be universal in the category of fading memory filters with uniformly bounded inputs.
- These universality statements are generalized to the stochastic setup for almost surely uniformly bounded inputs in Section 4. In particular, it is shown that any discrete-time filter that has the fading memory property with almost surely uniformly bounded stochastic inputs can be uniformly approximated by elements in the non-homogeneous state-affine family.

Despite some preexisting work on the uniform approximation in probability of stochastic systems with finite memory (see Perryman (1996), Perryman and Stubberud (1997), Stubberud and Perryman (1997a)), the universality result in the stochastic setup is, to our knowledge, the first of its type in the literature and opens the door to new developments in the learning of stochastic processes and their obvious applications to forecasting (see Galtier et al. (2014)). In the deterministic setup, RC has been very successful (see, for instance, Jaeger and Haas (2004), Pathak et al. (2017, 2018)) in the learning of the attractors of various dynamical systems. This approach is used for forecasting by path continuation of synthetically learnt proxies, which has led to several orders of magnitude accuracy improvements with respect to most standard dynamical systems forecasting techniques based on Takens' Theorem (Takens (1981)). We expect that the results in this paper should lead to comparable improvements in the density forecasting of stochastic processes.

2. Notation, definitions, and preliminary discussions

Vector and matrix notations. Polynomials. A column vector is denoted by a bold lower case symbol like $\mathbf{r}$ and $\mathbf{r}^\top$ indicates its transpose. Given a vector $\mathbf{v}\in\mathbb{R}^n$, we denote its entries by v_i or v^i, depending on the context, with $i\in\{1,\ldots,n\}$; we also write $\mathbf{v}=(v_i)_{i\in\{1,\ldots,n\}}$. We denote by $\mathbb{M}_{n,m}$ the space of real $n\times m$ matrices with $m,n\in\mathbb{N}$. When $n=m$, we use the symbol $\mathbb{M}_n$ to refer to the space of square matrices of order n. $\mathbb{D}_n\subset\mathbb{M}_n$ is the set of diagonal matrices of order n and $\mathbb{D}$ denotes the set of diagonal matrices of any order. Given a vector $\mathbf{v}\in\mathbb{R}^n$, we denote by $\mathrm{diag}(\mathbf{v})$ the diagonal matrix in $\mathbb{M}_n$ with the elements of $\mathbf{v}$ as diagonal entries. $\mathbb{N}\mathrm{il}_n^k\subset\mathbb{M}_n$ is the set of nilpotent matrices in $\mathbb{M}_n$ of index $k\leq n$, that is, $A\in\mathbb{N}\mathrm{il}_n^k$ if and only if $A\in\mathbb{M}_n$, $A^k=0$, and $A^l\neq 0$ for any $l<k$. $\mathbb{N}\mathrm{il}$ denotes the set of nilpotent matrices of any order and any index. Given a matrix $A\in\mathbb{M}_{n,m}$, we denote its components by A_{ij} and we write $A=(A_{ij})$, with $i\in\{1,\ldots,n\}$, $j\in\{1,\ldots m\}$. Given a vector $\mathbf{v}\in\mathbb{R}^n$, the symbol $\|\mathbf{v}\|$ stands for its Euclidean norm. For any $A\in\mathbb{M}_{n,m}$, $\|A\|_2$ denotes its matrix norm induced by the Euclidean norms in $\mathbb{R}^m$ and $\mathbb{R}^n$, and satisfies that $\|A\|_2=\sigma_{\max}(A)$, with $\sigma_{\max}(A)$ the largest singular value of A (Example 5.6.6 in Horn and Johnson (2013)). $\|A\|_2$ is sometimes referred to as the spectral norm of A (Horn and Johnson (2013)).

Let V_1,V_2,W_1,W_2 be vector spaces. The symbols $V_1\oplus V_2$ and $V_1\otimes V_2$ denote the corresponding direct sum and tensor product vector spaces (Hungerford (1974)), respectively, of V_1 and V_2. Given any $\mathbf{v}_1\in V_1$ and $\mathbf{v}_2\in V_2$, the vectors $\mathbf{v}_1\oplus\mathbf{v}_2\in V_1\oplus V_2$ and $\mathbf{v}_1\otimes\mathbf{v}_2\in V_1\otimes V_2$ are the direct sum and the (pure) tensor product of $\mathbf{v}_1$ and $\mathbf{v}_2$, respectively. Given two linear maps $A_1:V_1\longrightarrow W_1$ and $A_2:V_2\longrightarrow W_2$, we denote by $A_1\oplus A_2:V_1\oplus V_2\longrightarrow W_1\oplus W_2$ and $A_1\otimes A_2:V_1\otimes V_2\longrightarrow W_1\otimes W_2$ the associated direct sum and tensor product maps, respectively, defined by $A_1\oplus A_2\left(\mathbf{v}_1\oplus\mathbf{v}_2\right):=A_1\left(\mathbf{v}_1\right)\oplus A_2\left(\mathbf{v}_2\right)$ and $A_1\otimes A_2\left(\mathbf{v}_1\otimes\mathbf{v}_2\right):=A_1\left(\mathbf{v}_1\right)\otimes A_2\left(\mathbf{v}_2\right)$. The matrix representation of $A_1\oplus A_2$ is obtained by concatenating in a block diagonal matrix the matrix representations of A_1 and A_2. As to the matrix representation of $A_1\otimes A_2$ it is obtained via the Kronecker product of the matrix representations of A_1 and A_2 (Horn and Johnson (2013)).

Given an element $\mathbf{z}\in\mathbb{R}^n$, we denote by $\mathbb{R}[\mathbf{z}]$ the real-valued multivariate polynomials on $\mathbf{z}$ with real coefficients. Analogously, $\mathrm{Pol}(\mathbb{R}^n,\mathbb{R})$ will denote the set of real-valued polynomials on $\mathbb{R}^n$. When $z\in\mathbb{R}$ and $m,n\in\mathbb{N}$, we define the set $\mathbb{M}_{m,n}[z]$ of $\mathbb{M}_{m,n}$-valued polynomials on z with coefficients in $\mathbb{M}_{m,n}$ as

$$\mathbb{M}_{m,n}[z]:=\{A_0+zA_1+z^2A_2+\cdots+z^rA_r\mid r\in\mathbb{N},A_0,A_1,A_2,\ldots,A_r\in\mathbb{M}_{m,n}\}. \tag{2.1}$$

$\mathbb{N}\mathrm{il}_n^k[z]\subset\mathbb{M}_n[z]$ is the set of nilpotent $\mathbb{M}_n$-valued polynomials on z of index k, that is, $p(z)\in\mathbb{N}\mathrm{il}_n^k[z]$ whenever k is the smallest natural number for which $p(z)^k=\mathbf{0}$, for all $z\in\mathbb{R}$. $\mathbb{N}\mathrm{il}[z]$ is the set of matrix-valued nilpotent polynomials on z of any order and any index.

Sequence spaces. $\mathbb{N}$ denotes the set of natural numbers with the zero element included. $\mathbb{Z}$ (respectively, $\mathbb{Z}_+$ and $\mathbb{Z}_-$) are the integers (respectively, the positive and the negative integers). The symbol $(\mathbb{R}^n)^{\mathbb{Z}}$ denotes the set of infinite real sequences of the form $\mathbf{z}=(\ldots,\mathbf{z}_{-1},\mathbf{z}_0,\mathbf{z}_1,\ldots)$, $\mathbf{z}_i\in\mathbb{R}^n$, $i\in\mathbb{Z}$; $(\mathbb{R}^n)^{\mathbb{Z}_-}$ and $(\mathbb{R}^n)^{\mathbb{Z}_+}$ are the subspaces consisting of, respectively, left and right infinite sequences:

$(\mathbb{R}^n)^{\mathbb{Z}_-} = \{\mathbf{z} = (\ldots, \mathbf{z}_{-2}, \mathbf{z}_{-1}, \mathbf{z}_0) \mid \mathbf{z}_i \in \mathbb{R}^n, i \in \mathbb{Z}_-\}$, $(\mathbb{R}^n)^{\mathbb{Z}_+} = \{\mathbf{z} = (\mathbf{z}_0, \mathbf{z}_1, \mathbf{z}_2, \ldots) \mid \mathbf{z}_i \in \mathbb{R}^n, i \in \mathbb{Z}_+\}$. Analogously, $(D_n)^{\mathbb{Z}}$, $(D_n)^{\mathbb{Z}_-}$, and $(D_n)^{\mathbb{Z}_+}$ stand for (semi-)infinite sequences with elements in the subset $D_n \subset \mathbb{R}^n$. In most cases we shall use in these infinite product spaces either the product topology (see Chapter 2 in Munkres (2014)) or the topology induced by the supremum norm $\|\mathbf{z}\|_\infty := \sup_{t\in\mathbb{Z}} \{\|\mathbf{z}_t\|\}$. The symbols $\ell^\infty(\mathbb{R}^n)$ and $\ell^\infty_\pm(\mathbb{R}^n)$ will be used to denote the Banach spaces formed by the elements in those infinite product spaces that have a finite supremum norm $\|\cdot\|_\infty$. The symbol $B_n(\mathbf{v}, M) \subset \mathbb{R}^n$, denotes the open ball of radius $M > 0$ and center $\mathbf{v} \in \mathbb{R}^n$ with respect to the Euclidean norm. The bars over sets stand for topological closures, in particular, $\overline{B_n(\mathbf{v}, M)}$ is the closed ball.

Filters. We will refer to the maps of the type $U : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$ as ***filters*** or ***operators*** and to those like $H : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}$ (or $H : (D_n)^{\mathbb{Z}_\pm} \longrightarrow \mathbb{R}$) as ***functionals***. A filter U is called ***causal*** when for any two elements $\mathbf{z}, \mathbf{w} \in (D_n)^{\mathbb{Z}}$ that satisfy that $\mathbf{z}_\tau = \mathbf{w}_\tau$ for all $\tau \leq t$, for any given $t \in \mathbb{Z}$, we have that $U(\mathbf{z})_t = U(\mathbf{w})_t$. Let $U_\tau : (D_n)^{\mathbb{Z}} \longrightarrow (D_n)^{\mathbb{Z}}$, $\tau \in \mathbb{Z}$, be the time delay operator defined by $U_\tau(\mathbf{z})_t := \mathbf{z}_{t-\tau}$. The filter U is called ***time-invariant*** when it commutes with the time delay operator, that is, $U_\tau \circ U = U \circ U_\tau$ (in this expression, the two time delay operators U_τ have to be understood as defined in the appropriate sequence spaces). We recall (see, for instance, Boyd and Chua (1985)) that there is a bijection between causal time-invariant filters and functionals on $(D_n)^{\mathbb{Z}_-}$. Indeed, given a time-invariant filter U, we can associate to it a functional $H_U : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ via the assignment $H_U(\mathbf{z}) := U(\mathbf{z}^e)_0$, where $\mathbf{z}^e \in (D_n)^{\mathbb{Z}}$ is an arbitrary extension of $\mathbf{z} \in (D_n)^{\mathbb{Z}_-}$ to $(D_n)^{\mathbb{Z}}$. Conversely, for any functional $H : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$, we can define a time-invariant causal filter $U_H : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$ by $U_H(\mathbf{z})_t := H((\mathbb{P}_{\mathbb{Z}_-} \circ U_{-t})(\mathbf{z}))$, where U_{-t} is the $(-t)$-time delay operator and $\mathbb{P}_{\mathbb{Z}_-} : (D_n)^{\mathbb{Z}} \longrightarrow (D_n)^{\mathbb{Z}_-}$ is the natural projection. It is easy to verify that:

$$\begin{aligned} H_{U_H} &= H, \quad \text{for any functional} \quad H : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}, \\ U_{H_U} &= U, \quad \text{for any causal time-invariant filter} \quad U : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}. \end{aligned}$$

Additionally, let $H_1, H_2 : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ and $\lambda \in \mathbb{R}$, then $U_{H_1+\lambda H_2}(\mathbf{z}) = U_{H_1}(\mathbf{z}) + \lambda U_{H_2}(\mathbf{z})$, for any $\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}}$. In the following pages and when the discussion will take place in a causal and time-invariant setup, we will use the term filter to denote exchangeably the associated functional and the filter itself.

Reservoir filters. Consider now the RC system determined by (1.1)–(1.2). It is worth mentioning that, unlike in those expressions, the reservoir and the readout maps are in general defined only on subsets $D_N, D'_N \subset \mathbb{R}^N$ and $D_n \subset \mathbb{R}^n$ and not on the entire Euclidean spaces $\mathbb{R}^N$ and $\mathbb{R}^n$, that is, $F : D_N \times D_n \longrightarrow D'_N$ and $h : D'_N \to \mathbb{R}$. Reservoir systems determine a filter when the following existence and uniqueness property holds: for each $\mathbf{z} \in (D_n)^{\mathbb{Z}}$ there exists a unique $\mathbf{x} \in (D_N)^{\mathbb{Z}}$ such that for each $t \in \mathbb{Z}$, the relation (1.1) holds. This condition is known in the literature as the ***echo state property*** (see Jaeger (2010), Yildiz et al. (2012)) and has deserved much attention in the context of echo state networks (Jaeger and Haas (2004), Buehner and Young (2006), Bai Zhang et al. (2012), Wainrib and Galtier (2016), Manjunath and Jaeger (2013)). The echo state property formulated for infinite (or semi-infinite) inputs guarantees that the output of the filter at any given time does not depend on initial conditions. We emphasize that this is a genuine condition that is not automatically satisfied by all RC systems.

We will denote by $U^F : (D_n)^{\mathbb{Z}} \longrightarrow (D_N)^{\mathbb{Z}}$ the filter determined by the reservoir map via (1.1), that is, $U^F(\mathbf{z})_t := \mathbf{x}_t \in D_N$, and by $U^F_h : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$ the one determined by the entire reservoir system, that is, $U^F_h(\mathbf{z})_t := h\left(U^F(\mathbf{z})_t\right) = y_t$. U^F_h will be called the ***reservoir filter*** associated to the RC system (1.1)–(1.2). The filters U^F and U^F_h are causal by construction and it can also be shown that they are necessarily time-invariant (Grigoryeva and Ortega (2018)). We can hence associate to U^F_h a ***reservoir functional*** $H^F_h : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ determined by $H^F_h := H_{U^F_h}$.

Weighted norms and the fading memory property (FMP). Let $w : \mathbb{N} \longrightarrow (0, 1]$ be a decreasing sequence with zero limit. We define the associated ***weighted norm*** $\|\cdot\|_w$ on $(\mathbb{R}^n)^{\mathbb{Z}_-}$ associated to the

weighting sequence w as the map:

$$\begin{array}{rccc} \|\cdot\|_w : & (\mathbb{R}^n)^{\mathbb{Z}_-} & \longrightarrow & \overline{\mathbb{R}^+} \\ & \mathbf{z} & \longmapsto & \|\mathbf{z}\|_w := \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t w_{-t}\|\}, \end{array}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$. It is worth noting that the space

$$\ell^\infty_w(\mathbb{R}^n) := \left\{\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-} \mid \|\mathbf{z}\|_w < \infty\right\}, \tag{2.2}$$

endowed with weighted norm $\|\cdot\|_w$ forms a Banach space (Grigoryeva and Ortega (2018)).

All along the paper, we will work with ***uniformly bounded*** families of sequences, both in the deterministic and the stochastic setups. The two main properties of these subspaces in relation with the weighted norms are spelled out in the following two lemmas.

Lemma 1 *Let $M > 0$ and let K_M be the set of elements in $(\mathbb{R}^n)^{\mathbb{Z}_-}$ which are uniformly bounded by M, that is,*

$$K_M := \left\{\mathbf{z} \in (\mathbb{R}^n)^{\mathbb{Z}_-} \mid \|\mathbf{z}_t\| \leq M \quad \text{for all} \quad t \in \mathbb{Z}_-\right\} = \overline{B_n(\mathbf{0}, M)}^{\mathbb{Z}_-}, \tag{2.3}$$

with $\overline{B_n(\mathbf{0}, M)} \subset \mathbb{R}^n$ the closed ball of radius M and center $\mathbf{0}$ in $\mathbb{R}^n$ with respect to the Euclidean norm. Then, for any weighting sequence w and $\mathbf{z} \in K_M$, we have that $\|\mathbf{z}\|_w < \infty$.

Additionally, let $\lambda, \rho \in (0, 1)$ and let $w, w^\rho, w^{1-\rho}$ be the weighting sequences given by $w_t := \lambda^t$, $w^\rho_t := \lambda^{\rho t}$, $w^{1-\rho}_t := \lambda^{(1-\rho)t}$, $t \in \mathbb{N}$. Then, the series $\sum_{t=0}^\infty \|\mathbf{z}_{-t}\| w_t$ is absolutely convergent and satisfies the inequalities:

$$\sum_{t=0}^\infty \|\mathbf{z}_{-t}\| w_t = \sum_{t=0}^\infty \|\mathbf{z}_{-t}\| \lambda^t \leq \|\mathbf{z}\|_{w^{1-\rho}} \frac{1}{1-\lambda^\rho}, \tag{2.4}$$

$$\sum_{t=0}^\infty \|\mathbf{z}_{-t}\| w_t = \sum_{t=0}^\infty \|\mathbf{z}_{-t}\| \lambda^t \leq \|\mathbf{z}\|_{w^\rho} \frac{1}{1-\lambda^{1-\rho}}. \tag{2.5}$$

The following result is a discrete-time version of Lemma 1 in Boyd and Chua (1985) that is easily obtained by noticing that in the discrete-time setup all functions are trivially continuous if we consider the discrete topology for their domains and, moreover, all families of functions are equicontinuous. A proof is given in the appendices for the sake of completeness.

Lemma 2 *Let $M > 0$ and let K_M be as in (2.3). Let $w : \mathbb{N} \longrightarrow (0, 1]$ be a weighting sequence. Then K_M is a compact topological space when endowed with the relative topology inherited from the norm topology in the Banach space $(\ell^\infty_w(\mathbb{R}^n), \|\cdot\|_w)$.*

Definition 3 *Let $D_n \subset \mathbb{R}^n$ and let $H_U : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ be the functional associated to the causal and time-invariant filter $U : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$. We say that U has the **fading memory property (FMP)** whenever there exists a weighting sequence $w : \mathbb{N} \longrightarrow (0, 1]$ such that the map $H_U : ((D_n)^{\mathbb{Z}_-}, \|\cdot\|_w) \longrightarrow \mathbb{R}$ is continuous. This means that for any $\mathbf{z} \in (D_n)^{\mathbb{Z}_-}$ and any $\epsilon > 0$, there exists a $\delta(\epsilon) > 0$ such that for any $\mathbf{s} \in (D_n)^{\mathbb{Z}_-}$ that satisfies that*

$$\|\mathbf{z} - \mathbf{s}\|_w = \sup_{t\in\mathbb{Z}_-}\{\|(\mathbf{z}_t - \mathbf{s}_t) w_{-t}\|\} < \delta(\epsilon), \quad \text{then} \quad |H_U(\mathbf{z}) - H_U(\mathbf{s})| < \epsilon.$$

*If the weighting sequence w is such that $w_t = \lambda^t$, for some $\lambda \in (0, 1)$ and all $t \in \mathbb{N}$, then U is said to have the λ-**exponential fading memory property**.*

Remark 4 This formulation of the fading memory property is due to Boyd and Chua (1985) and it is the key concept that allowed these authors to extend to non-compact time intervals the first filter universality results formulated in the classical works Fréchet (1910), Wiener (1958), Brilliant (1958), and George (1959), always under compactness assumptions on the input space and the time interval in which inputs are defined.

Remark 5 In the context of reservoir filters, the fading memory property is in some occasions related to the ***Lyapunov stability*** of the autonomous system associated to the reservoir map by setting the input sequence equal to zero. This connection has been made explicit, for example, for discrete-time nonlinear state-space models that are affine in their inputs, and have direct feed-through term in the output (Zang and Iglesias (2004)) or for time delay reservoirs (Grigoryeva et al. (2016b)).

Remark 6 Time-invariant fading memory filters always have the ***bounded input, bounded output (BIBO)*** property. Indeed, if for simplicity we consider functionals H_U that map the zero input to zero, that is $H_U(\mathbf{0}) = 0$, and we want bounded outputs such that $|H_U(\mathbf{z})| < k$, for a given constant $k > 0$, by Definition 3 it suffices to consider inputs $\mathbf{z} \in (\mathbb{R}^N)^{\mathbb{Z}_-}$ such that $\|\mathbf{z}\|_\infty := \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t\|\} < \delta(k)$. Indeed, if H_U has the FMP with respect to a weighting sequence w, then $\|\mathbf{z}\|_w \leq \|\mathbf{z}\|_\infty < \delta(k)$ and hence $|H_U(\mathbf{z})| < k$, as required. Another important dynamical implication of the fading memory property is the ***uniqueness of steady states*** or, equivalently, the asymptotic independence of the dynamics on the initial conditions. See Theorem 6 in Boyd and Chua (1985) for details about this fact.

The following lemma, which will be used later on in the paper, spells out how the FMP depends on the weighting sequence used to define it.

Lemma 7 *Let $D_n \subset \mathbb{R}^n$ and let $H_U : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ be the functional associated to the causal and time-invariant filter $U : (D_n)^{\mathbb{Z}} \longrightarrow (\mathbb{R})^{\mathbb{Z}}$. If H_U has the FMP with respect to a given weighting sequence w, then it also has it with respect to any other weighting sequence w' which satisfies*

$$\frac{w_t}{w'_t} < \lambda, \quad \textit{for a fixed} \quad \lambda > 0 \quad \textit{and for all} \quad t \in \mathbb{N}.$$

In particular, the thesis of the lemma holds when w' dominates w, that is when $\lambda = 1$.

It can be shown (see Grigoryeva and Ortega (2018)) that when in this lemma the set $(D_n)^{\mathbb{Z}_-}$ is made of uniformly bounded sequences, that is, $(D_n)^{\mathbb{Z}_-} = K_M$, with K_M as in (2.3) then, if a filter has the FMP with respect to a given weighting sequence, it necessarily has the same property with respect to any other weighting sequence.

3. Universality results in the deterministic setup

The goal of this section is identifying families of reservoir filters that are able to uniformly approximate any time-invariant, causal, and fading memory filter with deterministic inputs with any desired degree of accuracy. Such families of reservoir computers are said to be ***universal***.

The main mathematical tool that we use is the Stone-Weierstrass theorem for polynomial subalgebras of real-valued functions defined on compact metric spaces. This approach provides us with universal families of filters as long as we can prove that, roughly speaking, their elements form polynomial algebras using a product defined in the space of functionals. More specifically, if $D_n \subset \mathbb{R}^n$ and $H_{U_1}, H_{U_2} : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ are the functionals associated to the causal and time-invariant filters $U_1, U_2 : (D_n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$, we readily define their product $H_{U_1} \cdot H_{U_2} : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ and linear combination $H_{U_1} + \lambda H_{U_2} : (D_n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$, $\lambda \in \mathbb{R}$, as

$$\left(H_{U_1} \cdot H_{U_2}\right)(\mathbf{z}) := H_{U_1}(\mathbf{z}) \cdot H_{U_2}(\mathbf{z}), \quad \left(H_{U_1} + \lambda H_{U_2}\right)(\mathbf{z}) := H_{U_1}(\mathbf{z}) + \lambda H_{U_2}(\mathbf{z}), \quad \mathbf{z} \in (D_n)^{\mathbb{Z}_-}. \tag{3.1}$$

This section contains two different universality results. The first one shows that polynomial algebras of filters generated by reservoir systems using the operations in (3.1) that have the fading memory property and that separate points, are able to approximate any fading memory filter. Two important consequences of this result are that the entire family of fading memory RCs itself is universal, as well as the one containing all the linear reservoirs with polynomial readouts, when certain spectral restrictions are imposed on the reservoir matrices (see below for details). In the second result, we restrict ourselves to reservoir computers with linear readouts and introduce the non-homogeneous state-affine family in order to be able to obtain a similar universality statement. The linearity restriction on the readouts makes this universality statement closer to the type of RCs used in applications and to the standard notion of reservoir system that one commonly finds in the literature (see Lukoševičius and Jaeger (2009)).

The first result can be seen as a discrete-time version of the one in Boyd and Chua (1985) for continuous-time filters, while the second one is an extension to infinite time intervals of the main approximation result in Fliess and Normand-Cyrot (1980), which was originally formulated for compact time intervals using homogeneous state-affine systems.

3.1 Universality for fading memory RCs with non-linear readouts

The following statement is a direct consequence of the compactness result in Lemma 2.3 and the Stone-Weierstrass, as formulated in Theorem 7.3.1 in Dieudonné (1969). See Appendix 6.4 for a detailed proof.

All along this subsection, we work with reservoir filters with uniformly bounded inputs in a set $K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$, as defined in (2.3). These filters are generated by reservoir systems $F : D_N \times \overline{B_n(\mathbf{0}, M)} \longrightarrow D_N$ and $h : D_N \to \mathbb{R}$, for some $n, N \in \mathbb{N}$, $M > 0$, and $D_N \subset \mathbb{R}^N$.

Theorem 8 *Let $K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ be a subset of the type defined in (2.3), I an index set, and let*

$$\mathcal{R} := \{H^{F_i}_{h_i} : K_M \longrightarrow \mathbb{R} \mid h_i \in C^\infty(D_{N_i}), F_i : D_{N_i} \times \overline{B_n(\mathbf{0}, M)} \longrightarrow D_{N_i}, i \in I, N_i \in \mathbb{N}\} \tag{3.2}$$

be a set of reservoir filters defined on K_M that have the FMP with respect to a given weighted norm $\|\cdot\|_w$. Let $\mathcal{A}(\mathcal{R})$ be the polynomial algebra generated by $\mathcal{R}$, that is, the set formed by finite products and linear combinations of elements in $\mathcal{R}$ according to the operations defined in (3.1). If the algebra $\mathcal{A}(\mathcal{R})$ contains the constant functionals and separates the points in K_M, then any causal, time-invariant fading memory filter $H : K_M \longrightarrow \mathbb{R}$ can be uniformly approximated by elements in $\mathcal{A}(\mathcal{R})$, that is, $\mathcal{A}(\mathcal{R})$ is dense in the set $(C^0(K_M), \|\cdot\|_w)$ of real-valued continuous functions on $(K_M, \|\cdot\|_w)$. More explicitly, this implies that for any fading memory filter H and any $\epsilon > 0$, there exist a finite set of indices $\{i_1, \ldots, i_r\} \subset I$ and a polynomial $p : \mathbb{R}^r \longrightarrow \mathbb{R}$ such that

$$\|H - H^F_h\|_\infty := \sup_{\mathbf{z}\in K_M}\{|H(\mathbf{z}) - H^F_h(\mathbf{z})|\} < \epsilon \quad \textit{with} \quad h := p(h_{i_1}, \ldots, h_{i_r}) \quad \textit{and} \quad F := (F_{i_1}, \ldots, F_{i_r}).$$

An important fact is that *the polynomial algebra $\mathcal{A}(\mathcal{R})$ generated by a set formed by fading memory reservoir filters is made of fading memory reservoir filters.* Indeed, let $h_i \in C^\infty(D_{N_i})$, $F_i : D_{N_i} \times \overline{B_n(\mathbf{0}, M)} \longrightarrow D_{N_i}$, $i \in \{1, 2\}$, and $\lambda \in \mathbb{R}$. Then, the product $H^{F_1}_{h_1} \cdot H^{F_2}_{h_2}$ and the linear combination $H^{F_1}_{h_1} + \lambda H^{F_2}_{h_2}$ filters, as they were defined in (3.1), are such that

$$H^{F_1}_{h_1} \cdot H^{F_2}_{h_2} = H^F_h, \quad \text{with} \quad h := h_1 \cdot h_2 \in C^\infty(D_{N_1} \times D_{N_2}), \tag{3.3}$$

$$H^{F_1}_{h_1} + \lambda H^{F_2}_{h_2} = H^F_{h'}, \quad \text{with} \quad h' := h_1 + \lambda h_2 \in C^\infty(D_{N_1} \times D_{N_2}), \tag{3.4}$$

and where $F : (D_{N_1} \times D_{N_2}) \times \overline{B_n(\mathbf{0}, M)} \longrightarrow (D_{N_1} \times D_{N_2})$ is given by

$$F(((\mathbf{x}_1)_t, (\mathbf{x}_2)_t), \mathbf{z}_t) := (F_1((\mathbf{x}_1)_t, \mathbf{z}_t), F_2((\mathbf{x}_2)_t, \mathbf{z}_t)), \tag{3.5}$$

for any $((\mathbf{x}_1)_t,(\mathbf{x}_2)_t)\in D_{N_1}\times D_{N_2}$, $\mathbf{z}_t\in\overline{B_n(\mathbf{0},M)}$, and $t\in\mathbb{Z}_-$. We emphasize that the functionals H_h^F and $H_{h'}^F$ in (3.3) and (3.4) are well defined because if the reservoir maps F_1 and F_2 satisfy the echo state property then so does F. Indeed, if $\mathbf{x}_1\in(D_{N_1})^{\mathbb{Z}}$ and $\mathbf{x}_2\in(D_{N_2})^{\mathbb{Z}}$ are the solutions of the reservoir equation (1.1) for F_1 and F_2 associated to the input $\mathbf{z}\in K_M$, then so is $(\mathbf{x}_1,\mathbf{x}_2)\in(D_{N_1}\times D_{N_2})^{\mathbb{Z}}$, defined by $(\mathbf{x}_1,\mathbf{x}_2)_t:=((\mathbf{x}_1)_t,(\mathbf{x}_2)_t)$, for F in (3.5).

This observation has as a consequence that the set formed by *all* the RC systems that have the echo state property and the FMP with respect to a given weighted norm $\|\cdot\|_w$ form a polynomial algebra that contains the constant functions (they can be obtained by using as readouts constant elements in $C^\infty(D_{N_i})$) and separates points (take the trivial reservoir map $F(\mathbf{x},\mathbf{z})=\mathbf{z}$ and use the separation property of $C^\infty(D_{N_i})$ together with time-invariance). This remark and Theorem 8 yield the following corollary.

Corollary 9 *Let $K_M\subset(\mathbb{R}^n)^{\mathbb{Z}_-}$ be a subset as defined in* (2.3) *and let*

$$\mathcal{R}_w:=\{H_h^F:K_M\longrightarrow\mathbb{R}\mid h\in C^\infty(D_N),F:D_N\times\overline{B_n(\mathbf{0},M)}\longrightarrow D_N,N\in\mathbb{N}\}\tag{3.6}$$

be the set of all reservoir filters with uniformly bounded inputs in K_M and that have the FMP with respect to a given weighted norm $\|\cdot\|_w$. Then $\mathcal{R}_w$ is universal, that is, it is dense in the set $(C^0(K_M),\|\cdot\|_w)$ of real-valued continuous functions on $(K_M,\|\cdot\|_w)$.

Remark 10 The stability of reservoir filters under products and linear combinations in (3.3)-(3.4) is a feature that allows us, in Corollary 9 and in some of the results that follow later on, to identify families of reservoir filters that are able to approximate any fading memory filter. This fact is a requirement for the application of the Stone-Weierstrass theorem but does not mean that we have to carry those operations out in the construction of approximating filters, which would indeed be difficult to implement in specific applications.

According to the previous corollary, reservoir filters that have the FMP are able to approximate any time-invariant fading memory filter. We now show that actually a much smaller family of reservoirs suffices to do that, namely, certain classes of linear reservoirs with polynomial readouts. Consider the RC system determined by the expressions

$$\begin{cases}\mathbf{x}_t=A\mathbf{x}_{t-1}+\mathbf{c}\mathbf{z}_t,\quad A\in\mathbb{M}_N,\mathbf{c}\in\mathbb{M}_{N,n},&(3.7)\\ y_t=h(\mathbf{x}_t),\quad h\in\mathbb{R}[\mathbf{x}].&(3.8)\end{cases}$$

If this system has a reservoir filter associated (the next result provides a sufficient condition for this to happen) we denote by $H_h^{A,\mathbf{c}}:K_M\longrightarrow\mathbb{R}$ the associated functional and we refer to it as the ***linear reservoir functional*** determined by $A,\mathbf{c}$, and the polynomial h. These filters exhibit the following universality property that is proved in Appendix 6.5.

Corollary 11 *Let $K_M\subset(\mathbb{R}^n)^{\mathbb{Z}_-}$ be a subset of the type defined in* (2.3) *and let $0<\epsilon<1$. Consider the set $\mathcal{L}_\epsilon$ formed by all the linear reservoir systems as in* (3.7)-(3.8) *determined by matrices $A\in\mathbb{M}_N$ such that $\sigma_{\max}(A)<1-\epsilon$. Then, the elements in $\mathcal{L}_\epsilon$ generate λ_ρ-exponential fading memory reservoir functionals, with $\lambda_\rho:=(1-\epsilon)^\rho$, for any $\rho\in(0,1)$. Equivalently, $\mathcal{L}_\epsilon\subset\mathcal{R}_{w^\rho}$, with $w_t^\rho:=\lambda_\rho^t$, and $\mathcal{R}_{w^\rho}$ as in* (3.6)*. These functionals can be explicitly written as:*

$$H_h^{A,\mathbf{c}}(\mathbf{z})=h\left(\sum_{i=0}^{\infty}A^i\mathbf{c}\mathbf{z}_{-i}\right),\quad \textit{for any}\quad \mathbf{z}\in K_M.\tag{3.9}$$

This family is dense in $(C^0(K_M),\|\cdot\|_{w^\rho})$.

The same universality result can be stated for the following two smaller subfamilies of $\mathcal{L}_\epsilon$:

(i) *The family $\mathcal{DL}_\epsilon\subset\mathcal{L}_\epsilon$ formed by the linear reservoir systems in $\mathcal{L}_\epsilon$ determined by diagonal matrices $A\in\mathbb{D}$ such that $\sigma_{\max}(A)<1-\epsilon$.*

(ii) *The family $\mathcal{NL}\subset\mathcal{L}_\epsilon$ formed by the linear reservoir systems determined by nilpotent matrices $A\in\mathbb{N}\text{il}$.*

Remark 12 The elements in the family $\mathcal{NL}$ belong automatically to $\mathcal{L}_\epsilon$ because the eigenvalues of a nilpotent matrix are always zero. This implies that if a linear reservoir system is determined by a nilpotent matrix $A\in\mathbb{N}\text{il}_N^k$ of index $k\leq N$, then the reservoir functional $H_h^{A,\mathbf{c}}$ is automatically well-defined and given by a finite version of (3.9), that is,

$$H_h^{A,\mathbf{c}}(\mathbf{z})=h\left(\sum_{i=0}^{k-1}A^i\mathbf{c}\mathbf{z}_{-i}\right),\quad \text{for any}\quad \mathbf{z}\in K_M.\tag{3.10}$$

3.2 State-affine systems and universality for fading memory RCs with linear readouts

As it was explained in the introduction, the standard notion of reservoir computing that one finds in the literature concerns architectures with linear readouts. It is is particularly convenient to work with RCs that have this feature in machine learning applications since in that case the training reduces to solving a linear regression problem. That makes training feasible when there is need for a high number of neurons, as it happens in many cases. This point makes relevant the identification of families of reservoirs that are universal when the readout is restricted to be linear, which is the subject of this subsection. In order to simplify the presentation, we restrict ourselves in this case to one-dimensional input signals, that is, all along this section we set $n=1$. The extension to the general case is straightforward, even though more convoluted to write down (see Remark 22).

Definition 13 *Let $N\in\mathbb{N}$, $\mathbf{W}\in\mathbb{R}^N$, and let $p(z)\in\mathbb{M}_N[z]$ and $q(z)\in\mathbb{M}_{N,1}[z]$ be two polynomials on the variable z with matrix coefficients, as they were introduced in* (2.1)*. The* ***non-homogeneous state-affine system (SAS)*** *associated to p,q and $\mathbf{W}$ is the reservoir system determined by the state-space transformation:*

$$\begin{cases}\mathbf{x}_t=p(z_t)\mathbf{x}_{t-1}+q(z_t),&(3.11)\\ y_t=\mathbf{W}^\top\mathbf{x}_t.&(3.12)\end{cases}$$

Our next result spells out a sufficient condition that guarantees that the SAS reservoir system (3.11)-(3.12) has the echo state property. Moreover, it provides an explicit expression for the unique causal and time-invariant solution associated to a given input.

Proposition 14 *Consider a non-homogeneous state-affine system as in* (3.11)-(3.12) *determined by polynomials p,q, and a vector $\mathbf{W}$, with inputs defined on $I^{\mathbb{Z}}$, $I:=[-1,1]$. Assume that*

$$K_1:=\max_{z\in I}\|p(z)\|_2=\max_{z\in I}\sigma_{\max}(p(z))<1.\tag{3.13}$$

Then, the reservoir system (3.11)-(3.12) *has the echo state property and for each input $\mathbf{z}\in I^{\mathbb{Z}}$ it has a unique causal and time-invariant solution given by*

$$\begin{cases}\mathbf{x}_t=\displaystyle\sum_{j=0}^{\infty}\left(\prod_{k=0}^{j-1}p(z_{t-k})\right)q(z_{t-j}),&(3.14)\\ y_t=\mathbf{W}^\top\mathbf{x}_t,&(3.15)\end{cases}$$

where

$$\prod_{k=0}^{j-1} p(z_{t-k}) := p(z_t)\cdot p(z_{t-1})\cdots p(z_{t-j+1}).$$

Let now $K_2 := \max_{z\in I}\|q(z)\|_2$. *Then,*

$$\|\mathbf{x}_t\| \leq \frac{K_2}{1-K_1}, \quad \textit{for all} \quad t\in\mathbb{Z}. \tag{3.16}$$

We will denote by $U^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$ *and* $H^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ *the corresponding SAS reservoir filter and SAS functional, respectively.*

The next result presents two alternative conditions that imply the hypothesis $\max_{z\in I}\|p(z)\|_2 < 1$ in the previous proposition and that are easier to verify in practice.

Lemma 15 *Let* $p(z)\in\mathbb{M}_N[z]$ *be the polynomial given by*

$$p(z) := A_0 + zA_1 + z^2A_2 + \cdots + z^{n_1}A_{n_1}, \quad n_1\in\mathbb{N}.$$

Suppose that $z\in I$ *and consider the following three conditions:*

(i) *There exists a constant* $0<\lambda<1$, *such that* $\|A_i\|_2 = \sigma_{\max}(A_i) < \lambda$, *for any* $i\in\{0,1,\ldots,n_1\}$, *and that at the same time satisfies that* $\lambda(n_1+1)<1$.

(ii) $B_p := \|A_0\|_2 + \|A_1\|_2 + \cdots + \|A_{n_1}\|_2 < 1$.

(iii) $M_p := \max_{z\in I}\|p(z)\|_2 < 1$.

Then, condition **(i)** *implies* **(ii)** *and condition* **(ii)** *implies* **(iii)**.

We emphasize that since Proposition 14 was proved using condition **(iii)** in the previous lemma then, any of the three conditions in that statement imply the echo state property for (3.14)-(3.15) and the time-invariance of the corresponding solutions. The next result shows that the same situation holds in relation with the fading memory property.

Proposition 16 *Consider a non-homogeneous state-affine system as in* (3.11)-(3.12) *determined by polynomials* p,q, *and a vector* $\mathbf{W}$, *with inputs defined on* $I^{\mathbb{Z}}$, $I:=[-1,1]$. *If the polynomial* p *satisfies any of the three conditions in Lemma 15 then the corresponding reservoir filter has the fading memory property. More specifically, if* p *satisfies condition* **(i)** *in Lemma 15, then* $H^{p,q}_{\mathbf{W}} : (I^{\mathbb{Z}_-}, \|\cdot\|_{w^\rho}) \longrightarrow \mathbb{R}$ *is continuous with* $w^\rho_t := (n_1+1)^{\rho t}\lambda^{\rho t}$ *and* $\rho\in(0,1)$ *arbitrary. The same conclusion holds for conditions* **(ii)** *and* **(iii)** *with* $w^\rho_t := B_p^{\rho t}$ *and* $w^\rho_t := M_p^{\rho t}$, *respectively.*

The importance of SAS in relation to the universality problem has to do with the fact that they form a polynomial algebra which allows us, under certain conditions, to use the Stone-Weierstrass theorem to prove a density statement. Before we show that, we observe that for any two polynomials $p_1(z)\in\mathbb{M}_{N_1,M_1}[z]$ and $p_2(z)\in\mathbb{M}_{N_2,M_2}[z]$ given by

$$p_1(z) := A^1_0 + zA^1_1 + z^2A^1_2 + \cdots + z^{n_1}A^1_{n_1}, \tag{3.17}$$
$$p_2(z) := A^2_0 + zA^2_1 + z^2A^2_2 + \cdots + z^{n_2}A^2_{n_2}, \tag{3.18}$$

with $n_1,n_2\in\mathbb{N}$, their direct sum and their tensor product are also polynomials in z with matrix coefficients. More explicitly, $p_1\oplus p_2(z)\in\mathbb{M}_{N_1+N_2,M_1+M_2}[z]$ and is written as

$$p_1\oplus p_2(z) = A^1_0\oplus A^2_0 + zA^1_1\oplus A^2_1 + z^2A^1_2\oplus A^2_2 + \cdots + z^{n_2}A^1_{n_2}\oplus A^2_{n_2} + z^{n_2+1}A^1_{n_2+1}\oplus\mathbf{0} + \cdots + z^{n_1}A^1_{n_1}\oplus\mathbf{0}, \tag{3.19}$$

where we assumed that $n_2\leq n_1$. Analogously, their tensor product $p_1\otimes p_2(z)\in\mathbb{M}_{N_1\cdot N_2,M_1\cdot M_2}[z]$ and is written as

$$p_1\otimes p_2(z) = \sum_{i=0}^{n_1}\sum_{j=0}^{n_2} z^{i+j}A^1_i\otimes A^2_j. \tag{3.20}$$

The next result shows that the products and the linear combinations of SAS reservoir functionals are SAS reservoir functionals. Additionally, it makes explicit the polynomials that determine the corresponding SAS reservoir systems.

Proposition 17 *Let* $H^{p_1,q_1}_{\mathbf{W}_1}, H^{p_2,q_2}_{\mathbf{W}_2} : I^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ *be two SAS reservoir functionals associated to two corresponding time-invariant SAS reservoir systems. Assume that the two polynomials with matrix coefficients* $p_1(z)\in\mathbb{M}_{N_1}[z]$ *and* $p_2(z)\in\mathbb{M}_{N_2}[z]$ *satisfy that* $\|p_1(z)\|_2 < 1-\epsilon$ *and* $\|p_2(z)\|_2 < 1-\epsilon$ *for all* $z\in I := [-1,1]$ *and a given* $0<\epsilon<1$. *Then, with the notation introduced in the expressions* (3.19) *and* (3.20), *we have that:*

(i) *For any* $\lambda\in\mathbb{R}$, *the linear combination of the SAS reservoir functionals* $H^{p_1,q_1}_{\mathbf{W}_1} + \lambda H^{p_2,q_2}_{\mathbf{W}_2}$ *is a SAS reservoir functional and:*

$$H^{p_1,q_1}_{\mathbf{W}_1} + \lambda H^{p_2,q_2}_{\mathbf{W}_2} = H^{p_1\oplus p_2, q_1\oplus q_2}_{\mathbf{W}_1\oplus\lambda\mathbf{W}_2}. \tag{3.21}$$

(ii) *The product of the SAS reservoir functionals* $H^{p_1,q_1}_{\mathbf{W}_1}\cdot H^{p_2,q_2}_{\mathbf{W}_2}$ *is a SAS reservoir functional and:*

$$H^{p_1,q_1}_{\mathbf{W}_1}\cdot H^{p_2,q_2}_{\mathbf{W}_2} = H^{p,q_1\oplus q_2\oplus(q_1\otimes q_2)}_{\mathbf{0}\oplus\mathbf{0}\oplus(\mathbf{W}_1\otimes\mathbf{W}_2)}, \tag{3.22}$$

where $p(z)\in\mathbb{M}_{N_{12}}[z]$, $N_{12} := N_1+N_2+N_1\cdot N_2$, *is the polynomial with matrix coefficients in* $\mathbb{M}_{N_{12}}$ *whose block-matrix expression for the three summands in* $\mathbb{R}^{N_1}\oplus\mathbb{R}^{N_2}\oplus\left(\mathbb{R}^{N_1}\otimes\mathbb{R}^{N_2}\right)$ *is:*

$$p(z) := \begin{pmatrix} p_1(z) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & p_2(z) & \mathbf{0} \\ p_1\otimes q_2(z) & q_1\otimes p_2(z) & p_1\otimes p_2(z) \end{pmatrix}. \tag{3.23}$$

The expression $p_1\otimes p_2(z)\in\mathbb{M}_{N_1\cdot N_2}[z]$ *denotes the element defined in* (3.20). *The symbol* $p_1\otimes q_2(z)$ *(respectively,* $q_1\otimes p_2(z)$*) denotes the matrix of the linear map from* $\mathbb{R}^{N_1}$ *(respectively,* $\mathbb{R}^{N_2}$*) to* $\mathbb{R}^{N_1}\otimes\mathbb{R}^{N_2}$ *that associates to any* $\mathbf{v}_1\in\mathbb{R}^{N_1}$ *the element* $(p_1(z)\mathbf{v}_1)\otimes q_2(z)$ *(respectively,* $q_1(z)\otimes(p_2(z)\mathbf{v}_2)$, *with* $\mathbf{v}_2\in\mathbb{R}^{N_2}$*). When all the polynomials in* (3.23) *are written in terms of monomials using the conventions that we just mentioned and we factor out the different powers of the variable* z, *we obtain a polynomial with matrix coefficients in* $\mathbb{M}_{N_{12}}$ *and with degree* $\deg(p)$ *equal to*

$$\deg(p) = \max\left\{\deg(p_1)\cdot\deg(q_2), \deg(q_1)\cdot\deg(p_2), \deg(p_1)\cdot\deg(p_2)\right\}.$$

The equalities (3.21) *and* (3.22) *show that the SAS family forms a polynomial algebra.*

Remark 18 Notice that the linear reservoir equation (3.7) is a particular case of the SAS reservoir equation (3.11) that is obtained by taking for p and q polynomials of degree zero and one, respectively. Regarding that specific case, Proposition 17 shows that linear reservoirs with linear readouts do not form a polynomial algebra. Indeed, as it can be seen in (3.22), the product of two SAS filters involves the tensor product $q_1\otimes q_2$ which, when q_1 and q_2 come from a linear filter, it has degree two and it is hence not compatible with a linear reservoir filter.

Theorem 19 (Universality of SAS reservoir computers) *Let* $I^{\mathbb{Z}_-}\subset\mathbb{R}^{\mathbb{Z}_-}$ *be the subset of real uniformly bounded sequences in* $I := [-1,1]$ *as in* (2.3), *that is,*

$$I^{\mathbb{Z}_-} := \{\mathbf{z}\in\mathbb{R}^{\mathbb{Z}_-} \mid z_t\in[-1,1], \textit{ for all } \quad t\leq 0\},$$

and let $\mathcal{S}_\epsilon$ be the family of functionals $H^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ induced by the state-affine systems defined in (3.11)-(3.12) *that satisfy that $M_p := \max_{z\in I} \|p(z)\|_2 < 1-\epsilon$ and $M_q := \max_{z\in I} \|q(z)\|_2 < 1-\epsilon$. The family $\mathcal{S}_\epsilon$ forms a polynomial subalgebra of $\mathcal{R}_{w^\rho}$ (as defined in* (3.6)*) with $w^\rho_t := (1-\epsilon)^{\rho t}$ and $\rho \in (0,1)$ arbitrary, made of fading memory reservoir filters that contains the constant functions and separates points. The subfamily $\mathcal{S}_\epsilon$ is hence dense in the set $(C^0(I^{\mathbb{Z}_-}), \|\cdot\|_{w^\rho})$ of real-valued continuous functions on $(I^{\mathbb{Z}_-}, \|\cdot\|_{w^\rho})$.*

This statement implies that any causal, time-invariant fading memory filter $H : I^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ can be uniformly approximated by elements in $\mathcal{S}_\epsilon$. More specifically, for any fading memory filter H and any $\epsilon > 0$, there exist a natural number $N \in \mathbb{N}$, polynomials $p(z) \in \mathbb{M}_N[z], q(z) \in \mathbb{M}_{N,1}[z]$ with $M_p, M_q < 1-\epsilon$, and a vector $\mathbf{W} \in \mathbb{R}^N$ such that

$$\|H - H^{p,q}_{\mathbf{W}}\|_\infty := \sup_{z\in I^{\mathbb{Z}_-}} \{|H(z) - H^{p,q}_{\mathbf{W}}(z)|\} < \epsilon.$$

The same universality result can be stated for the smaller subfamily $\mathcal{NS}_\epsilon \subset \mathcal{S}_\epsilon$ formed by SAS reservoir systems determined by nilpotent polynomials $p(z) \in \mathbb{N}\mathrm{il}[z]$.

Remark 20 As it is stated in Theorem 19, it is the condition **(iii)** in Lemma 15 that yields a universal family of SAS fading memory reservoirs. As it can deduced from its proof (available in the Appendix 6.10), the families determined by conditions **(i)** or **(ii)** in that lemma contain SAS fading memory reservoirs but they do not form a polynomial algebra. In such cases, and according to Theorem 8, it is the algebras generated by them and not the families themselves that are universal.

Remark 21 A continuous-time analog of the universality result that we just proved can be obtained using the bilinear systems considered in Section 5.3 of Boyd and Chua (1985). In discrete time, but only when the number of time steps is finite, this universal approximation property is exhibited by homogeneous state-affine systems, that is, by setting $q(z) = \mathbf{0}$ in (3.11)-(3.12) (see Fliess and Normand-Cyrot (1980)).

Remark 22 Generalization to multidimensional signals. When the input signal is defined in $I_n^{\mathbb{Z}}$, with $I_n := [-1,1]^n$, a SAS family with the same universality properties can be defined by replacing the polynomials p and q in Definition 13, by polynomials of degree r and s of the form:

$$\begin{aligned} p(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,r\}\\ i_1+\cdots+i_n\leq r}} z_1^{i_1}\cdots z_n^{i_n} A_{i_1,\ldots,i_n}, \quad A_{i_1,\ldots,i_n} \in \mathbb{M}_N, \quad \mathbf{z}\in I_n \\ q(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,s\}\\ i_1+\cdots+i_n\leq s}} z_1^{i_1}\cdots z_n^{i_n} B_{i_1,\ldots,i_n}, \quad B_{i_1,\ldots,i_n} \in \mathbb{M}_{N,1}, \quad \mathbf{z}\in I_n. \end{aligned}$$

Remark 23 SAS with trigonometric polynomials. An analogous construction can be carried out using trigonometric polynomials of the type:

$$\begin{aligned} p(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,r\}\\ i_1+\cdots+i_n\leq r}} \cos\left(i_1\cdot z_1+\cdots+i_n\cdot z_n\right) A_{i_1,\ldots,i_n}, \quad A_{i_1,\ldots,i_n} \in \mathbb{M}_N, \quad \mathbf{z}\in I_n \\ q(\mathbf{z}) &= \sum_{\substack{i_1,\ldots,i_n\in\{0,\ldots,s\}\\ i_1+\cdots+i_n\leq s}} \cos\left(i_1\cdot z_1+\cdots+i_n\cdot z_n\right) B_{i_1,\ldots,i_n}, \quad B_{i_1,\ldots,i_n} \in \mathbb{M}_{N,1}, \quad \mathbf{z}\in I_n. \end{aligned}$$

In this case, it is easy to establish that the resulting SAS family forms a polynomial algebra by invoking Proposition 17 and by reformulating the expressions (3.19) and (3.20) using the trigonometric identity

$$\cos(\theta)\cos(\phi) = \frac{1}{2}\left(\cos(\theta-\phi)+\cos(\theta+\phi)\right).$$

Additionally, the corresponding SAS family includes the linear family and hence the point separation property can be established as in the proof of Theorem 19 in the Appendix 6.10.

4. Reservoir universality results in the stochastic setup

This section extends the previously stated deterministic universality results to a setup in which the reservoir inputs and outputs are stochastic, that is, the reservoir is not driven anymore by infinite sequences but by discrete-time stochastic processes. We emphasize that we restrict our discussion to reservoirs that are deterministic and hence the only source of randomness in the systems considered is the stochastic nature of the input.

The results that follow are mainly based on the observation that if we adopt a uniform approximation criterion and we assume that the random inputs satisfy almost surely the uniform boundedness that we used as hypothesis in Section 3, then important features like the fading memory property or universality are naturally inherited in the stochastic setup from the deterministic case. This fact is what we call the ***deterministic-stochastic transfer principle*** and it is contained in the statement of Theorem 27 below. In particular, this result can be easily applied to show that all the universal families with deterministic inputs introduced in Section 3 are also universal in the stochastic setup when the input processes considered produce paths that, up to a set of measure zero, are uniformly bounded.

The stochastic setup. All along this section we work on a probability space $(\Omega, \mathcal{A}, \mathbb{P})$. If a condition defined on this probability space holds everywhere except for a set with zero measure, we will say that the relation is true ***almost surely***. Let $\mathbf{X} : \Omega \longrightarrow B$ be a random variable with $(B, \|\cdot\|_B)$ a normed space endowed with a σ-algebra (for example, but not necessarily, its Borel σ-algebra). Let

$$\|\mathbf{X}\|_{L^\infty} := \operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{X}(\omega)\|_B\} = \inf\left\{\rho\in\overline{\mathbb{R}^+} \mid \|\mathbf{X}\|_B \leq \rho \quad \text{almost surely}\right\}, \tag{4.1}$$

We denote by $L^\infty(\Omega, B)$ the classes of B-valued almost surely equal random variables whose norms have a finite essential supremum or that, equivalently, have almost surely bounded norms, that is,

$$L^\infty(\Omega, B) := S_B/\sim_B, \tag{4.2}$$

where

$$S_B := \{\mathbf{X} : \Omega \longrightarrow B \ \text{ random variable} \mid \|\mathbf{X}\|_{L^\infty} < \infty\}, \tag{4.3}$$

and $\sim_B$ is the equivalence relation defined on S_B as follows: two random variables $\mathbf{Y}$ and $\mathbf{Z}$ with finite $\|\cdot\|_{L^\infty}$ norm are $\sim_B$-equivalent if and only if $\mathbb{P}(\{\omega\in\Omega : \mathbf{Y}(\omega) \neq \mathbf{Z}(\omega)\}) = 0$. As it is customary in the literature, we will not make a distinction in what follows between the elements in S_B and the classes in the quotient $L^\infty(\Omega, B)$. Using this identification we recall, for example, that $L^\infty(\Omega, B)$ is a vector space with the operations

$$(\mathbf{X} + \lambda\mathbf{Y})(\omega) := \mathbf{X}(\omega) + \lambda\mathbf{Y}(\omega) \tag{4.4}$$

for any $\mathbf{X}, \mathbf{Y} \in L^\infty(\Omega, B)$, $\lambda \in \mathbb{R}$, $\omega \in \Omega$. Moreover, $(L^\infty(\Omega, B), \|\cdot\|_{L^\infty})$ is a normed space. We emphasize that $L^\infty(\Omega, B)$ is in general not a Banach space (see pages 42 and 46 in Ledoux and Talagrand (1991)). It can be shown that whenever B is finite dimensional or, more generally, a separable Banach space, then the space $L^\infty(\Omega, B)$ is also a Banach space (Pisier (2016)).

Given an element $\mathbf{X} \in L^\infty(\Omega, B)$, we denote by $\mathrm{E}[\mathbf{X}]$ its expectation. The following lemma collects some elementary results that will be needed later on and shows, in particular, that the expectation $\mathrm{E}[\mathbf{X}]$ as well as that of all the powers $\|\mathbf{X}\|_B^k$ of its norm are finite for all the elements $\mathbf{X} \in L^\infty(\Omega, B)$.

Lemma 24 *Let $\mathbf{X} \in L^\infty(\Omega, B)$ and let $C \in \overline{\mathbb{R}^+}$. Then:*

(i) $\|\mathbf{X}\|_B \leq \|\mathbf{X}\|_{L^\infty}$ *almost surely.*

(ii) $\|\mathbf{X}\|_{L^\infty} \leq C$ *if and only if* $\|\mathbf{X}\|_B \leq C$ *almost surely.*

(iii) $\|\mathbf{X}\|_B \leq C$ *almost surely if and only if* $\mathrm{E}\left[\|\mathbf{X}\|_B^k\right] \leq C^k$ *for any* $k \in \mathbb{N}$.

(iv) *Let $B=\mathbb{R}^n$, then the components X_i of $\mathbf{X}$, $i\in\{1,\ldots,n\}$, are such that $\mathrm{E}[X_i]\leq\|\mathbf{X}\|_{L^\infty}$.*

The first point in this lemma explains why we will refer to the elements of $L^\infty(\Omega,B)$ as ***almost surely bounded*** random variables.

Stochastic inputs and outputs. The filters that we will consider in this section have ***almost surely bounded stochastic processes*** as inputs and outputs. Recall that a discrete-time stochastic process is a map of the type:

$$\begin{array}{rccc}\mathbf{z}: & \mathbb{Z}\times\Omega & \longrightarrow & \mathbb{R}^n \\ & (t,\omega) & \longmapsto & \mathbf{z}_t(\omega),\end{array} \tag{4.5}$$

such that, for each $t\in\mathbb{Z}$, the assignment $\mathbf{z}_t:\Omega\longrightarrow\mathbb{R}^n$ is a random variable. For each $\omega\in\Omega$, we will denote by $\mathbf{z}(\omega):=\{\mathbf{z}_t(\omega)\in\mathbb{R}^n\mid t\in\mathbb{Z}\}$ the ***realization*** or the ***sample path*** of the process $\mathbf{z}$. The results that follow are presented for stochastic processes indexed by $\mathbb{Z}$ but are equally valid for $\mathbb{Z}_+$ and $\mathbb{Z}_-$.

Recall that a map of the type (4.5) is a $\mathbb{R}^n$-valued stochastic process if and only if the corresponding map $\mathbf{z}:\Omega\longrightarrow(\mathbb{R}^n)^{\mathbb{Z}}$ into path space (designated with the same symbol) is a random variable when in $(\mathbb{R}^n)^{\mathbb{Z}}$ we consider the product sigma algebra generated by cylinder sets (Chapter 1 in Comets and Meyre (2006)). Then, the space of $\mathbb{R}^n$-valued stochastic processes can be made into a vector space with the same operations as in (4.4) and we can define in this space a norm $\|\cdot\|_{L^\infty}$ using the same prescription as in (4.1) by considering $(\mathbb{R}^n)^{\mathbb{Z}}$ as a normed space with the supremum norm $\|\cdot\|_\infty$, that is,

$$\|\mathbf{z}\|_{L^\infty}:=\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}(\omega)\|_\infty\}=\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{4.6}$$

The following lemma provides an alternative characterization of the norm $\|\cdot\|_{L^\infty}$ that will be very useful in the proofs of the results that follow and in which the supremum and the essential supremum have been interchanged. The last statement contained in it complements part **(ii)** of Lemma 24 for processes.

Lemma 25 *Let $\mathbf{z}:\Omega\longrightarrow(\mathbb{R}^n)^{\mathbb{Z}}$ be a stochastic process. Then,*

$$\|\mathbf{z}\|_{L^\infty}:=\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\}=\sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{4.7}$$

Equivalently, using the notation in (4.1)*,*

$$\|\mathbf{z}\|_{L^\infty}:=\left\|\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\|_{L^\infty}=\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t\|_{L^\infty}\}. \tag{4.8}$$

These equalities imply that for any non-negative real number $C\geq 0$, the process $\mathbf{z}$ satisfies that $\|\mathbf{z}\|_{L^\infty}\leq C$ if and only if $\|\mathbf{z}_t\|_{L^\infty}\leq C$ for all $t\in\mathbb{Z}$ or, equivalently, if and only if $\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t\|_{L^\infty}\}\leq C$.

Consider now the space $L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right)$ of processes with finite $\|\cdot\|_{L^\infty}$ norm. We refer to the elements of $L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right)$ as ***almost surely bounded time series***. Additionally, consider the space $L^\infty\left(\Omega,\ell^\infty(\mathbb{R}^n)\right)$ of processes whose paths are all uniformly bounded, that is, they lay in the Banach space $(\ell^\infty(\mathbb{R}^n),\|\cdot\|_\infty)$. According to the definition in (4.2), we have for both these spaces that

$$L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right):=S_{(\mathbb{R}^n)^{\mathbb{Z}}}/\sim_{(\mathbb{R}^n)^{\mathbb{Z}}},\quad L^\infty\left(\Omega,\ell^\infty(\mathbb{R}^n)\right):=S_{\ell^\infty(\mathbb{R}^n)}/\sim_{\ell^\infty(\mathbb{R}^n)}$$

with

$$S_{(\mathbb{R}^n)^{\mathbb{Z}}}:=\left\{\mathbf{z}:\mathbb{Z}\times\Omega\longrightarrow\mathbb{R}^n\ \text{ stochastic process, }\mathbf{z}(\omega)\in(\mathbb{R}^n)^{\mathbb{Z}},\ \text{for all }\omega\in\Omega\mid\|\mathbf{z}\|_{L^\infty}<\infty\right\},$$

$$S_{\ell^\infty(\mathbb{R}^n)}:=\left\{\mathbf{z}:\mathbb{Z}\times\Omega\longrightarrow\mathbb{R}^n\ \text{ stochastic process, }\mathbf{z}(\omega)\in\ell^\infty(\mathbb{R}^n),\ \text{for all }\omega\in\Omega\mid\|\mathbf{z}\|_{L^\infty}<\infty\right\},$$

and with the almost sure equality equivalence relations $\sim_{\ell^\infty(\mathbb{R}^n)}$ and $\sim_{(\mathbb{R}^n)^{\mathbb{Z}}}$ between stochastic processes with paths in $\ell^\infty(\mathbb{R}^n)$ and $(\mathbb{R}^n)^{\mathbb{Z}}$, respectively. The following result shows that the normed spaces $L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right)$ and $L^\infty\left(\Omega,\ell^\infty(\mathbb{R}^n)\right)$ are isomorphic.

Lemma 26 *In the setup that we just introduced the inclusion $\iota:S_{\ell^\infty(\mathbb{R}^n)}\hookrightarrow S_{(\mathbb{R}^n)^{\mathbb{Z}}}$ is equivariant with respect to the equivalence relations $\sim_{\ell^\infty(\mathbb{R}^n)}$ and $\sim_{(\mathbb{R}^n)^{\mathbb{Z}}}$ and drops to an isomorphism of normed spaces $\phi:\left(L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right),\|\cdot\|_{L^\infty}\right)\longrightarrow\left(L^\infty\left(\Omega,\ell^\infty(\mathbb{R}^n)\right),\|\cdot\|_{L^\infty}\right)$. Equivalently, the following diagram commutes*

$$\begin{array}{ccc} S_{\ell^\infty(\mathbb{R}^n)} & \xhookrightarrow{\ \iota\ } & S_{(\mathbb{R}^n)^{\mathbb{Z}}} \\ \Pi_{\sim_{\ell^\infty(\mathbb{R}^n)}}\downarrow & & \downarrow\Pi_{\sim_{(\mathbb{R}^n)^{\mathbb{Z}}}} \\ L^\infty\left(\Omega,\ell^\infty(\mathbb{R}^n)\right) & \xrightarrow{\ \phi\ } & L^\infty\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}}\right),\end{array}$$

where $\Pi_{\sim_{\ell^\infty(\mathbb{R}^n)}}$ and $\Pi_{\sim_{(\mathbb{R}^n)^{\mathbb{Z}}}}$ are the canonical projections.

Let now w be a weighting sequence and let $\|\cdot\|_w$ be the associated weighted norm in $(\mathbb{R}^n)^{\mathbb{Z}_-}$. If we replace in (4.6) the ℓ^∞ norm $\|\cdot\|_\infty$ by the weighted norm $\|\cdot\|_w$, we obtain a weighted norm $\|\cdot\|_{L^\infty_w}$ in the space of processes $\mathbf{z}:\mathbb{Z}_-\times\Omega\longrightarrow\mathbb{R}^n$ indexed by $\mathbb{Z}_-$ as:

$$\|\mathbf{z}\|_{L^\infty_w}:=\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}(\omega)\|_w\}=\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t(\omega)\|w_{-t}\}\right\}. \tag{4.9}$$

We will denote by $L^\infty_w\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}_-}\right)$ the space of processes with finite $\|\cdot\|_{L^\infty_w}$ norm. A result similar to Lemma 26 shows that the normed spaces $\left(L^\infty_w\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}_-}\right),\|\cdot\|_{L^\infty_w}\right)$ and $\left(L^\infty\left(\Omega,\ell^\infty_w(\mathbb{R}^n)\right),\|\cdot\|_{L^\infty_w}\right)$ are isomorphic. Additionally, as in Lemma 25, we have that for any $\mathbf{z}\in L^\infty_w\left(\Omega,(\mathbb{R}^n)^{\mathbb{Z}_-}\right)$:

$$\|\mathbf{z}\|_{L^\infty_w}:=\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}_-}\{\|\mathbf{z}_t(\omega)\|w_{-t}\}\right\}=\sup_{t\in\mathbb{Z}_-}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|w_{-t}\}\right\}. \tag{4.10}$$

Deterministic filters in a stochastic setup. As we already pointed out, we consider filters U that have almost surely bounded processes as inputs and outputs. The same conventions as in the deterministic setup are used in the identification of the different signals, namely, $\mathbf{z}$ denotes the filter input process and the symbol y is reserved for the output process. Let now $D_n\subset\mathbb{R}^n$ and let $D_n^{L^\infty_{\mathbb{Z}}}\subset L^\infty(\Omega,(\mathbb{R}^n)^{\mathbb{Z}})$ be a subset formed by processes whose paths take values in D_n almost surely. In the sequel we will restrict our attention to intrinsically ***deterministic filters*** $U:D_n^{L^\infty_{\mathbb{Z}}}\longrightarrow L^\infty(\Omega,\mathbb{R}^{\mathbb{Z}})$ that are obtained by presenting almost surely bounded stochastic inputs $\mathbf{z}\in D_n^{L^\infty_{\mathbb{Z}}}\subset L^\infty(\Omega,(\mathbb{R}^n)^{\mathbb{Z}})$ to filters $U:(D_n)^{\mathbb{Z}}\longrightarrow\mathbb{R}^{\mathbb{Z}}$ similar to those introduced in the previous section, which explains why we use the same symbol for both. This is explicitly carried out by defining the output process $U(\mathbf{z})\in L^\infty(\Omega,\mathbb{R}^{\mathbb{Z}})$ using the convention

$$(U(\mathbf{z}))(\omega):=U(\mathbf{z}(\omega)),\quad\omega\in\Omega, \tag{4.11}$$

where on the right hand side it is the filter $U:(D_n)^{\mathbb{Z}}\longrightarrow\mathbb{R}^{\mathbb{Z}}$ which is applied to the paths $\mathbf{z}(\omega):=\{\mathbf{z}_t(\omega)\in\mathbb{R}^n\mid t\in\mathbb{Z}\}\in(D_n)^{\mathbb{Z}}$ of the process $\mathbf{z}$. We call these filters deterministic because, in view of (4.11) the dependence of the image process $(U(\mathbf{z}))(\omega)\in\in L^\infty(\Omega,\mathbb{R}^{\mathbb{Z}})$ on the probability space takes place exclusively through the dependence $\mathbf{z}(\omega)$ in the input. In this section we reserve the symbol U to denote deterministic filters $U:D_n^{L^\infty_{\mathbb{Z}}}\longrightarrow L^\infty(\Omega,\mathbb{R}^{\mathbb{Z}})$. We draw attention to the fact that assuming that

the filters map into almost surely bounded processes is a genuine hypothesis that needs to be verified in each specific case considered.

The concepts of causality and time-invariance are defined as in the deterministic case by replacing equalities by almost sure equalities in the corresponding identities. More explicitly, we say that the filter $U : D_n^{L_{\mathbb{Z}}^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R}^{\mathbb{Z}})$ is time-invariant when for any $\tau \in \mathbb{Z}$ and any $\mathbf{z} \in D_n^{L_{\mathbb{Z}}^\infty}$, we have that

$$(U_\tau \circ U)(\mathbf{z}) = (U \circ U_\tau)(\mathbf{z}), \quad \text{almost surely.}$$

Analogously, we say that the filter is causal with stochastic inputs when for any two elements $\mathbf{z}, \mathbf{w} \in D_n^{L_{\mathbb{Z}}^\infty}$ that satisfy that $\mathbf{z}_\tau = \mathbf{w}_\tau$ almost surely, for any $\tau \leq t$ and for a given $t \in \mathbb{Z}$, we have that $U(\mathbf{z})_t = U(\mathbf{w})_t$, almost surely. Causal and time-invariant deterministic filters produce almost surely causal and time-invariant filters when stochastic inputs are presented to them.

In this setup, there is also a correspondence between causal and time-invariant filters $U : D_n^{L_{\mathbb{Z}}^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R}^{\mathbb{Z}})$ and functionals $H_U : D_n^{L_{\mathbb{Z}_-}^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R})$, where $D_n^{L_{\mathbb{Z}_-}^\infty} := \mathbb{P}_{\mathbb{Z}_-}\left(D_n^{L_{\mathbb{Z}}^\infty}\right)$.

Given a weighting sequence $w : \mathbb{N} \longrightarrow (0, 1]$ and a time-invariant filter $U : D_n^{L_{\mathbb{Z}_-}^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R}^{\mathbb{Z}})$ with stochastic inputs, we will say that U has the ***fading memory property*** with respect to the weighting sequence w when the corresponding functional $H_U : \left(D_n^{L_{\mathbb{Z}_-}^\infty}, \|\cdot\|_{L_w^\infty}\right) \longrightarrow L^\infty(\Omega, \mathbb{R})$ is a continuous map.

Let $M > 0$ and define, using Lemma 25,

$$K_M^{L^\infty} := \left\{\mathbf{z} \in L^\infty(\Omega, (\mathbb{R}^n)^{\mathbb{Z}_-}) \mid \|\mathbf{z}\|_{L^\infty} \leq M\right\} = \left\{\mathbf{z} \in L^\infty(\Omega, (\mathbb{R}^n)^{\mathbb{Z}_-}) \mid \|\mathbf{z}_t\|_{L^\infty} \leq M, \text{ for all } t \in \mathbb{Z}_-\right\}. \tag{4.12}$$

The sets $K_M^{L^\infty}$ are the stochastic counterparts of the sets K_M in the deterministic setup; we will say that $K_M^{L^\infty}$ is a set of ***almost surely uniformly bounded processes***. A stochastic analog of Lemma 1 can be formulated for them with K_M replaced by $K_M^{L^\infty}$, the norm $\|\cdot\|$ by $\|\cdot\|_{L^\infty}$, and the weighted norm $\|\cdot\|_w$ by $\|\cdot\|_{L_w^\infty}$. Indeed, the following result shows that the fading memory and the universality properties are naturally inherited by deterministic filters with almost surely uniformly bounded inputs. We call this fact the ***deterministic-stochastic transfer principle***.

Theorem 27 (Deterministic-stochastic transfer principle) *Let $M > 0$ and let K_M and $K_M^{L^\infty}$ be the sets of deterministic and stochastic inputs defined in (2.3) and (4.12), respectively. The following properties hold true:*

(i) *Let $H : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R}$ be a causal and time-invariant filter. Then H has the fading memory property if and only if the corresponding filter with almost surely uniformly bounded inputs has almost surely bounded outputs, that is, $H : (K_M^{L^\infty}, \|\cdot\|_{L_w^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$, and it has the fading memory property.*

(ii) *Let $\mathcal{T} := \{H_i : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R} \mid i \in I\}$ be a family of causal and time-invariant fading memory filters. Then, $\mathcal{T}$ is dense in the set $(C^0(K_M), \|\cdot\|_w)$ if and only if the corresponding family with inputs in $K_M^{L^\infty}$ is universal in the set of continuous maps of the type $H : (K_M^{L^\infty}, \|\cdot\|_{L_w^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$.*

A first universality result using RC systems. The following paragraphs contain a stochastic analog of Theorem 8 which shows that any fading memory filter with almost surely uniformly bounded inputs can be approximated using the elements of a polynomial algebra of reservoir filters with the same kind of inputs, provided that it contains the constant functionals and has the separation property. We note that, as in the deterministic case, the existence of the reservoir filter associated to a reservoir system like (1.1)-(1.2) is guaranteed only in the presence of the echo state property. The next lemma

shows that this property is inherited by deterministic fading memory reservoir filters with almost surely bounded inputs.

Lemma 28 *Consider a reservoir system determined by the relations (1.1)-(1.2) and the maps $F : D_N \times \overline{B_n(\mathbf{0}, M)} \longrightarrow D_N$ and $h : D_N \to \mathbb{R}$, for some $n, N \in \mathbb{N}$, $M > 0$, and $D_N \subset \mathbb{R}^N$. If the reservoir system has the echo state and the fading memory properties then so does the corresponding system with stochastic inputs in $K_M^{L^\infty}$ which, additionally, has an associated reservoir functional $H_h^F : (K_M^{L^\infty}, \|\cdot\|_{L_w^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ with almost surely bounded outputs that satisfies the fading memory property.*

Theorem 29 *Let $M > 0$ and let $K_M^{L^\infty}$ be the set of almost surely uniformly bounded processes introduced in (4.12). Consider the set $\mathcal{R}$*

$$\mathcal{R} := \{H_{h_i}^{F_i} : K_M^{L^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R}) \mid h_i \in \mathrm{Pol}(\mathbb{R}^{N_i}, \mathbb{R}), F_i : \mathbb{R}^{N_i} \times \mathbb{R}^n \longrightarrow \mathbb{R}^{N_i}, i \in I, N_i \in \mathbb{N}\} \tag{4.13}$$

formed by deterministic fading memory reservoir filters with respect to a given weighted norm $\|\cdot\|_w$ and driven by stochastic inputs in $K_M^{L^\infty}$. Let $\mathcal{A}(\mathcal{R})$ be the polynomial algebra generated by $\mathcal{R}$. If the algebra $\mathcal{A}(\mathcal{R})$ has the separation property and contains all the constant functionals, then any deterministic, causal, time-invariant fading memory filter $H : (K_M^{L^\infty}, \|\cdot\|_{L_w^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ can be uniformly approximated by elements in $\mathcal{A}(\mathcal{R})$, that is, for any $\epsilon > 0$, there exist a finite set of indices $\{i_1, \ldots, i_r\} \subset I$ and a polynomial $p : \mathbb{R}^r \longrightarrow \mathbb{R}$ such that

$$\|H - H_h^F\|_\infty := \sup_{\mathbf{z} \in K_M^{L^\infty}} \{\|H(\mathbf{z}) - H_h^F(\mathbf{z})\|_{L^\infty}\} < \epsilon \quad \text{with} \quad h := p(h_{i_1}, \ldots, h_{i_r}) \quad \text{and} \quad F := (F_{i_1}, \ldots, F_{i_r}).$$

In the next paragraphs we identify, as in the deterministic case, families of reservoirs that satisfy the hypotheses of this theorem and where we will eventually impose linearity constraints on the readout function. The following corollary to Theorem 29 is the stochastic analog of Corollary 9.

Corollary 30 *Let $M > 0$ and let $K_M^{L^\infty}$ be the set of almost surely uniformly bounded processes introduced in (4.12). Let*

$$\mathcal{R}_w := \{H_h^F : K_M^{L^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R}) \mid h \in \mathrm{Pol}(\mathbb{R}^N, \mathbb{R}), F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N, N \in \mathbb{N}\} \tag{4.14}$$

be the set of all the reservoir filters defined on $K_M^{L^\infty}$ that have the FMP with respect to a given weighted norm $\|\cdot\|_{L_w^\infty}$. Then $\mathcal{R}_w$ is universal, that is, for any time-invariant fading memory filter $H : (K_M^{L^\infty}, \|\cdot\|_{L_w^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ and any $\epsilon > 0$, there exists a reservoir filter $H_h^F \in \mathcal{R}_w$ such that $\|H - H_h^F\|_\infty := \sup_{\mathbf{z} \in K_M^{L^\infty}} \{\|H(\mathbf{z}) - H_h^F(\mathbf{z})\|_{L^\infty}\} < \epsilon$.

Linear reservoir computers with stochastic inputs are universal. As it was the case in the deterministic setup, we can prove in the stochastic case that the linear RC family introduced in (3.7)-(3.8) suffices to achieve universality. The proof of the following statement is a direct consequence of Corollary 11 and Theorem 27.

Corollary 31 *Let $M > 0$ and let $K_M^{L^\infty}$ be the set of almost surely uniformly bounded processes introduced in (4.12). Let $\mathcal{L}_\epsilon$ be the family introduced in Corollary 11 and formed by all the linear reservoir filters $H_p^{A,\mathbf{c}}$ determined by matrices $A \in \mathbb{M}_N$ such that $\sigma_{\max}(A) < 1 - \epsilon$. The elements in $\mathcal{L}_\epsilon$ map $K_M^{L^\infty}$ into $L^\infty(\Omega, \mathbb{R})$ and are time-invariant fading memory filters with respect to the weighted norm $\|\cdot\|_{w^\rho}^{L^\infty}$ associated to $w_t^\rho := (1 - \epsilon)^{\rho t}$, for any $\rho \in (0, 1)$. Moreover, they are universal, that is, for any time-invariant and causal fading memory filter $H : (K_M^{L^\infty}, \|\cdot\|_{L_{w^\rho}^\infty}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ and any $\varepsilon > 0$, there exists $H_p^{A,\mathbf{c}} \in \mathcal{L}_\epsilon$ such that $\|H - H_p^{A,\mathbf{c}}\|_\infty := \sup_{\mathbf{z} \in K_M^{L^\infty}} \{\|H(\mathbf{z}) - H_p^{A,\mathbf{c}}(\mathbf{z})\|_{L^\infty}\} < \varepsilon$.*

The same universality result can be stated for the subfamily $\mathcal{DL}_\epsilon \subset \mathcal{L}_\epsilon$, formed by the linear reservoir systems in $\mathcal{L}_\epsilon$ determined by diagonal matrices, and for $\mathcal{NL} \subset \mathcal{L}_\epsilon$, formed by the linear reservoir systems determined by nilpotent matrices.

Remark 32 The linear reservoir filters in $\mathcal{NL}$ determined by nilpotent matrices have been used in Gonon and Ortega (2018) to formulate a L^p version of these universality results.

Remark 33 The previous corollary has interesting consequences in the realm of time series analysis. Indeed, many well-known parametric time series models consist in autoregressive relations, possibly nonlinear, driven by independent or uncorrelated innovations. The parameter constraints that are imposed on them in order to ensure that they have (second order) stationary solutions imply, in may situations, that the resulting filter has the FMP. In those cases, Corollary 31 allows us to conclude that when those models are driven by almost surely uniformly bounded innovations, they can be arbitrarily well approximated by a polynomial function of a vector autoregressive model (VAR) of order 1. This statement applies, for example, to any stationary ARMA (see Box and Jenkins (1976), Brockwell and Davis (2006)) or GARCH model (see Engle (1982), Bollerslev (1986), Francq and Zakoian (2010)) driven by almost surely uniformly bounded innovations.

State-affine reservoir computers with almost surely uniformly bounded inputs are universal. As it was the case in the deterministic setup, non-homogeneous SAS are universal time-invariant fading memory filters in the stochastic framework with almost surely uniformly bounded inputs. Their advantage with respect to the families proposed in the previous corollary is that they use a linear readout which is of major importance in practical implementations. More specifically, the following result holds true as a direct consequence of Theorem 19 and the equivalence stated in Theorem 27.

Theorem 34 (Universality of SAS reservoir computers with almost surely uniformly bounded inputs) *Let $K_I^{L^\infty} \subset L^\infty(\Omega, \mathbb{R}^{\mathbb{Z}_-})$ be the set of almost surely and uniformly bounded processes in the interval $I = [-1, 1]$, that is,*

$$K_I^{L^\infty} := \left\{ z \in L^\infty(\Omega, \mathbb{R}^{\mathbb{Z}_-}) \mid \|z_t\|_{L^\infty} \leq 1, \quad \text{for all} \quad t \in \mathbb{Z}_- \right\}.$$

Let $\mathcal{S}_\epsilon$ be the family of functionals $H^{p,q}_{\mathbf{W}} : K_I^{L^\infty} \longrightarrow L^\infty(\Omega, \mathbb{R})$ induced by the state-affine systems defined in (3.11)-(3.12) and that satisfy $M_p := \max_{z \in I} \|p(z)\| < 1 - \epsilon$ and $M_q := \max_{z \in I} \|q(z)\| < 1 - \epsilon$. The family $\mathcal{S}_\epsilon$ forms a polynomial subalgebra of $\mathcal{R}_{w^\rho}$ (as defined in (4.14)) with $w_t^\rho := (1 - \epsilon)^{\rho t}$, made of fading memory reservoir filters that map into $L^\infty(\Omega, \mathbb{R})$.

Moreover, for any time-invariant and causal fading memory filter $H : (K_I^{L^\infty}, \|\cdot\|_{L^\infty_{w^\rho}}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ and any $\epsilon > 0$, there exist a natural number $N \in \mathbb{N}$, polynomials $p(z) \in \mathbb{M}_{N,N}[z], q(z) \in \mathbb{M}_{N,1}[z]$ with $M_p, M_q < 1 - \epsilon$, and a vector $\mathbf{W} \in \mathbb{R}^N$ such that

$$\|H - H^{p,q}_{\mathbf{W}}\|_\infty := \sup_{z \in K_I^{L^\infty}} \{\|H(z) - H^{p,q}_{\mathbf{W}}(z)\|_{L^\infty}\} < \epsilon.$$

The same universality result can be stated for the smaller subfamily $\mathcal{NS}_\epsilon \subset \mathcal{S}_\epsilon$ formed by SAS reservoir systems determined by nilpotent polynomials $p(z) \in \mathbb{N}\mathrm{il}[z]$.

5. Conclusion

This paper studies and proposes solutions for the universality problem in the approximation of fading memory filters using reservoir computer (RC) systems. RCs are a particular type of recurrent neural networks that have important applications both in machine learning and in signal processing where they exhibit superb information processing performances. Their importance is also linked to the possibility of building highly efficient hardware realizations. RC systems are in general defined as nonlinear state-space systems determined by a reservoir and a readout map. In many supervised machine learning applications the readout is chosen to be linear and the reservoir map is randomly generated, which reduces the training of a dynamic task to a static regression problem and allows to circumvent well-known difficulties in the training of generic recurrent neural networks.

The universality question that we addressed consists in finding families of RCs as simple as possible such that the set of input/output functionals that can be generated with them is dense in a sufficiently rich class. The work presented here is the dynamic counterpart of a statement of this type for neural networks in a static and deterministic setup in which they have been proved to be universal approximators.

The RC universality results stated in the paper correspond to two different situations in which the inputs are either deterministic and uniformly bounded or stochastic and almost surely uniformly bounded. In both cases we proved two different universality statements. First, we showed that the family of fading memory RCs is universal in the much larger fading memory filters category. The same applies to the much smaller RC family containing just linear reservoirs with polynomial readouts, when certain spectral restrictions are imposed on the reservoir maps. The second result concerns exclusively reservoir computers with linear readouts, which are closer to the type of RCs used in applications and hardware implementations. More specifically, we introduced the family of what we called non-homogeneous state-affine systems and identified sufficient conditions that guarantee that the associated reservoir computers with linear readouts are causal, time-invariant, and satisfy the echo state and the fading memory properties. Finally, we stated a universality result for a subset of this class which was shown to be universal in the same fading memory filters category as above. These universality statements are then generalized to the stochastic setup for almost surely uniformly bounded inputs. In particular, we showed that any discrete-time filter that has the fading memory property with almost surely uniformly bounded stochastic inputs can be uniformly approximated by elements in the non-homogeneous state-affine family. All the density statements in the paper are formulated with respect to natural uniform approximation norms that appear in each of the different cases considered.

Despite preexisting work, these universality results are, to our knowledge, the first of their type in the semi-infinite discrete-time inputs setup. In the stochastic case they open the door to new developments in the learning theory of stochastic processes.

6. Appendices

6.1 Proof of Lemma 1

Let $w : \mathbb{N} \longrightarrow (0, 1]$ be an arbitrary weighting sequence. Then, for any $\mathbf{z} \in K_M$:

$$\|\mathbf{z}\|_w := \sup_{t \in \mathbb{Z}_-} \{\|\mathbf{z}_t w_{-t}\|\} = \sup_{t \in \mathbb{Z}_-} \{\|\mathbf{z}_t\| w_{-t}\} \leq M \cdot 1 = M < \infty.$$

Regarding the inequalities (2.4) and (2.5), notice that if $w_t = \lambda^t$ then:

$$\begin{aligned}\sum_{t=0}^{\infty} \|\mathbf{z}_{-t}\| w_t = \sum_{t=0}^{\infty} \|\mathbf{z}_{-t}\| \lambda^t = \sum_{t=0}^{\infty} \|\mathbf{z}_{-t}\| (\lambda^{1-\rho} \lambda^\rho)^t = \sum_{t=0}^{\infty} \|\mathbf{z}_{-t}\| \lambda^{(1-\rho)t} \lambda^{\rho t} \\ \leq \sum_{t=0}^{\infty} \sup_{i \in \mathbb{N}} \left\{ \|\mathbf{z}_{-i}\| \lambda^{(1-\rho)i} \right\} \lambda^{\rho t} = \sup_{i \in \mathbb{N}} \left\{ \|\mathbf{z}_{-i}\| \lambda^{(1-\rho)i} \right\} \sum_{t=0}^{\infty} \lambda^{\rho t} = \|\mathbf{z}\|_{w^{1-\rho}} \frac{1}{1 - \lambda^\rho},\end{aligned}$$

which proves (2.4). The proof of (2.5) is similar and follows from noticing that:

$$\sum_{t=0}^{\infty} \|\mathbf{z}_{-t}\| \lambda^{(1-\rho)t} \lambda^{\rho t} \leq \sum_{t=0}^{\infty} \sup_{i \in \mathbb{N}} \left\{ \|\mathbf{z}_{-i}\| \lambda^{\rho i} \right\} \lambda^{(1-\rho)t)} = \sup_{i \in \mathbb{N}} \left\{ \|\mathbf{z}_{-i}\| \lambda^{\rho i} \right\} \sum_{t=0}^{\infty} \lambda^{(1-\rho)t} = \|\mathbf{z}\|_{w^\rho} \frac{1}{1 - \lambda^{1-\rho}}. \quad \blacksquare$$

6.2 Proof of Lemma 2

We recall first that by Lemma 1 we have that $\|\mathbf{z}\|_w < \infty$, for any $\mathbf{z} \in K_M$. Second, since $(\ell_w^\infty(\mathbb{R}^n), \|\cdot\|_w)$ is a Banach space (Grigoryeva and Ortega (2018)), it is hence metrizable and therefore so is $(K_M, \|\cdot\|_w)$ when endowed with the relative topology (see, for instance, Exercise 1, Chapter 2, §21, Munkres (2014)). We will then conclude the compactness of $(K_M, \|\cdot\|_w)$ by showing that this space is sequentially compact (see, for example, Theorem 28.2 in Munkres (2014)). We proceed by using the strategy in the proof of Lemma 1 in Boyd and Chua (1985).

For any $m \in \mathbb{N}$, let K_M^m be the set obtained by projecting into $(\mathbb{R}^n)^{\{-m,\ldots,-1,0\}}$ the elements of $K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$. Given an element $\mathbf{z} \in K_M$, we will denote by $\mathbf{z}^{(m)} := (\mathbf{z}_{-m}, \ldots, \mathbf{z}_0)$ its projection into K_M^m. Additionally, notice that $K_M^m = \overline{B_n(\mathbf{0}, M)}^{m+1}$ is compact (and hence sequentially compact) with the product topology, since it is a product of closed balls $\overline{B_n(\mathbf{0}, M)} \subset \mathbb{R}^n$ which are compact.

Let $\{\mathbf{z}(n)\}_{n\in\mathbb{N}} \subset K_M$ be a sequence of elements in K_M. The argument that we just stated proves that for any $k \in \mathbb{N}$, there is a subset $\mathbb{N}_k \subset \mathbb{N}$ and an element $\mathbf{z}^{(k)} \in K_M^k$ such that

$$\max_{t\in\{-k,\ldots,0\}} \left\| \mathbf{z}_t(n) - \mathbf{z}_t^{(k)} \right\| \longrightarrow 0, \quad \text{as } n \to \infty,\ n \in \mathbb{N}_k.$$

Moreover, the sets $\mathbb{N}_k$ can be constructed so that $\mathbb{N} \supset \mathbb{N}_1 \supset \mathbb{N}_2 \supset \cdots$ and so that $\mathbf{z}^{(k)}$ extends $\mathbf{z}^{(l)}$ when $k \geq l$. This implies the existence of an element $\mathbf{z} \in K_M$ such that, for each $k \in \mathbb{N}$,

$$\max_{t\in\{-k,\ldots,0\}} \|\mathbf{z}_t(n) - \mathbf{z}_t\| \longrightarrow 0, \quad \text{as } n \to \infty,\ n \in \mathbb{N}_k,$$

and hence there exists an increasing subsequence n_k such that $n_k \in \mathbb{N}_k$ and that for each k_0,

$$\max_{t\in\{-k_0,\ldots,0\}} \|\mathbf{z}_t(n_k) - \mathbf{z}_t\| \longrightarrow 0, \quad \text{as } k \longrightarrow \infty. \tag{6.1}$$

We conclude by showing that the sequence $\{\mathbf{z}(n_k)\}_{k\in\mathbb{N}}$ converges in $(K_M, \|\cdot\|_w)$ to the element $\mathbf{z} \in K_M$. First, given that $w_t \longrightarrow 0$ as $t \longrightarrow \infty$, then for any $\varepsilon > 0$ there exists k_0 such that $w_k < \varepsilon/2M$, for any $k \geq k_0$. Additionally, since $\mathbf{z}(n_k), \mathbf{z} \in K_M$ for any $k \in \mathbb{N}$, we have that

$$\sup_{t\leq -k_0} \{\|\mathbf{z}_t(n_k) - \mathbf{z}_t\| w_{-t}\} \leq 2Mw_{k_0} < \varepsilon. \tag{6.2}$$

Now, by (6.1) there exists k_1 such that for any $k \geq k_1$

$$\sup_{t\in\{-k_0,\ldots,0\}} \{\|\mathbf{z}_t(n_k) - \mathbf{z}_t\| w_{-t}\} < \sup_{t\in\{-k_0,\ldots,0\}} \{\|\mathbf{z}_t(n_k) - \mathbf{z}_t\|\} < \varepsilon. \tag{6.3}$$

Consequently, (6.2) and (6.3) imply that for any $k > \max\{k_0, k_1\}$, $\|\mathbf{z}(n_k) - \mathbf{z}\|_w < \varepsilon$, as required. ■

6.3 Proof of Lemma 7

Let $\delta^w(\epsilon)$ be the epsilon-delta relation for the FMP associated to the weighting sequence w. We now show that H_U has the FMP with respect to w' via the epsilon-delta relation given by $\delta^{w'}(\epsilon) := \delta^w(\epsilon)/\lambda$. Indeed, for any $\epsilon > 0$ and any $\mathbf{z}, \mathbf{s} \in K$ such that $\|\mathbf{z} - \mathbf{s}\|_{w'} < \delta^{w'}(\epsilon)$, we have that

$$\|\mathbf{z}-\mathbf{s}\|_w = \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{z}_t-\mathbf{s}_t\|w_{-t}\} = \sup_{t\in\mathbb{Z}_-} \left\{\|\mathbf{z}_t - \mathbf{s}_t\| \frac{w_{-t}}{w'_{-t}} w'_{-t}\right\} < \lambda \sup_{t\in\mathbb{Z}_-} \{\|\mathbf{z}_t-\mathbf{s}_t\|w'_{-t}\} < \lambda\|\mathbf{z}-\mathbf{s}\|_{w'} < \lambda\delta^{w'}(\epsilon) = \delta^w(\epsilon),$$

and consequently, since H_U has the FMP with respect to the weighting sequence w, we can conclude that $|H_U(\mathbf{z}) - H_U(\mathbf{s})| < \epsilon$. This shows that the implication

$$\|\mathbf{z} - \mathbf{s}\|_{w'} < \delta^{w'}(\epsilon) \Longrightarrow |H_U(\mathbf{z}) - H_U(\mathbf{s})| < \epsilon$$

holds, as required. ■

6.4 Proof of Theorem 8

Since the elements in $\mathcal{R}$ have the FMP with respect to a given weighted norm $\|\cdot\|_w$, then so do those in $\mathcal{A}(\mathcal{R})$ since polynomial combinations of continuous elements of the form $H_{h_i}^{F_i} : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R}$ are also continuous. Therefore, under that hypothesis, $\mathcal{A}(\mathcal{R})$ is a polynomial subalgebra of the algebra $(C^0(K_M), \|\cdot\|_w)$ of real-valued continuous functions on $(K_M, \|\cdot\|_w)$. Since by hypothesis $\mathcal{A}(\mathcal{R})$ contains the constant functionals and separates the points in K_M and, by Lemma 2, the set $(K_M, \|\cdot\|_w)$ is compact, the Stone-Weierstrass theorem (Theorem 7.3.1 in Dieudonné (1969)) implies that $\mathcal{A}(\mathcal{R})$ is dense in $(C^0(K_M), \|\cdot\|_w)$, which concludes the proof. ■

6.5 Proof of Corollary 11

In order to show that the reservoir systems in $\mathcal{L}_\epsilon$ induce reservoir filters, we first show that they have the echo state property by using the following lemma, whose proof can be found in Grigoryeva and Ortega (2018).

Lemma 35 *Let $D_N \subset \mathbb{R}^N$ and $D_n \subset \mathbb{R}^n$ and let $F : D_N \times D_n \longrightarrow D_N$ be a continuous reservoir map. Suppose that F is a contraction map with contraction constant $0 < r < 1$, that is:*

$$\|F(\mathbf{x}, \mathbf{z}) - F(\mathbf{y}, \mathbf{z})\| \leq r \|\mathbf{x} - \mathbf{y}\|, \quad \textit{for all } \mathbf{x}, \mathbf{y} \in D_N \textit{ and all } \mathbf{z} \in D_n,$$

then the corresponding reservoir system has the echo state property.

We start now by noting that the condition $\sigma_{\max}(A) < 1 - \epsilon < 1$ implies that the reservoir map $F(\mathbf{x}, \mathbf{z}) := A\mathbf{x} + \mathbf{cz}$ associated to (3.7) is a contracting map with constant $\sigma_{\max}(A)$ which, by hypothesis, is smaller than one. Indeed,

$$\|F(\mathbf{x}, \mathbf{z}) - F(\mathbf{y}, \mathbf{z})\| = \|A(\mathbf{x} - \mathbf{y})\| \leq \sigma_{\max}(A) \|\mathbf{x} - \mathbf{y}\| \quad \text{for all } \mathbf{x}, \mathbf{y} \in D_N \text{ and all } \mathbf{z} \in D_n.$$

By Lemma 35 we can conclude that this reservoir system has a reservoir filter associated that we now show is explicitly given by (3.9). We start by proving that the conditions $\sigma_{\max}(A) < 1 - \epsilon < 1$ and that the elements in K_M are uniformly bounded by a constant M imply that the infinite sum in (3.9) is convergent. Let $n, m \in \mathbb{N}$ be such that $n < m$ and let $S_n := \sum_{i=0}^n A^i \mathbf{c} z_{-i}$. Now:

$$\|S_n - S_m\| = \left\| \sum_{j=n+1}^m A^i \mathbf{c}\mathbf{z}_{-i} \right\| \leq \sum_{j=n+1}^m \|A\|_2^i \|\mathbf{c}\|_2 \|\mathbf{z}_{-i}\| \leq M \|\mathbf{c}\|_2 \sum_{j=n+1}^m \sigma_{\max}(A)^i$$
$$\leq M \|\mathbf{c}\|_2 \sum_{j=n+1}^\infty \sigma_{\max}(A)^i = M \|\mathbf{c}\|_2 \frac{\sigma_{\max}(A)^{n+1}}{1 - \sigma_{\max}(A)}.$$

The condition $\sigma_{\max}(A) < 1 - \epsilon < 1$ implies that $M \|\mathbf{c}\|_2 \frac{\sigma_{\max}(A)^{n+1}}{1-\sigma_{\max}(A)} = M \frac{\sigma_{\max}(\mathbf{c})\sigma_{\max}(A)^{n+1}}{1-\sigma_{\max}(A)} \to 0$ as $n \to \infty$ and hence $\{S_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathbb{R}^N$ that consequently converges.

The fact that the filter determined by the expression (3.9) is a solution of the recursions (3.7)-(3.8) is a straightforward verification. In order to carry it out, it suffices to use that the filter $U_h^{A,\mathbf{c}}(\mathbf{z})$ associated to the functional $H_h^{A,\mathbf{c}}(\mathbf{z})$ is given by

$$U_h^{A,\mathbf{c}}(\mathbf{z})_t = h\left(\sum_{i=0}^\infty A^i \mathbf{c}\mathbf{z}_{t-i}\right),$$

and that the time series $\widetilde{\mathbf{x}}_t$ defined by $\widetilde{\mathbf{x}}_t := \sum_{i=0}^\infty A^i \mathbf{c}\mathbf{z}_{t-i}$ satisfies the recursion relation (3.7).

We now verify by hand that the filters $U_h^{A,\mathbf{c}}$ are time-invariant. Let $\mathbf{z} \in K_M$ and $t, \tau \in \mathbb{N}$ arbitrary and let U_τ be the corresponding time delay operator, then:

$$\left(U_h^{A,\mathbf{c}} \circ U_\tau\right)(\mathbf{z})_t = \left(U_h^{A,\mathbf{c}}\left(U_\tau(\mathbf{z})\right)\right)_t = h\left(\sum_{i=0}^{\infty} A^i \mathbf{c} U_\tau(\mathbf{z})_{t-i}\right) = h\left(\sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{z}_{t-i-\tau}\right) \tag{6.4}$$

At the same time,

$$\left(U_\tau \circ U_h^{A,\mathbf{c}}\right)(\mathbf{z})_t = \left(U_\tau\left(U_h^{A,\mathbf{c}}(\mathbf{z})\right)\right)_t = U_h^{A,\mathbf{c}}(\mathbf{z})_{t-\tau} = h\left(\sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{z}_{t-\tau-i}\right),$$

which coincides with (6.4) and proves the time-invariance of $U_h^{A,\mathbf{c}}$.

The next step consists in showing that the elements in $\mathcal{L}_\epsilon$ are λ_ρ-exponential fading memory filters, with $\lambda_\rho := (1-\epsilon)^\rho$, for any $\rho \in (0,1)$, that is, $\mathcal{L}_\epsilon \subset \mathcal{R}_{w^\rho}$, with $w^\rho : \mathbb{N} \to (0,1]$ the sequence given by $w_t^\rho := (1-\epsilon)^{\rho t}$. Let $\|\cdot\|_{w^\rho}$ be the associated weighted norm in K_M and let $\mathbf{z} \in K_M$ be an arbitrary element. We start by noting that the continuity of the readout map $h : D_N \to \mathbb{R}$ implies that for any $\varepsilon > 0$ there exists an element $\delta(\varepsilon) > 0$ such that for any $\mathbf{v} \in D_N$ that satisfies

$$\left\|\mathbf{v} - \sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{z}_{t-i}\right\| < \delta(\varepsilon), \quad \text{then} \quad \left|h(\mathbf{v}) - h\left(\sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{z}_{t-i}\right)\right| < \varepsilon. \tag{6.5}$$

We now show that for any $\mathbf{s} \in K_M$ such that

$$\|\mathbf{s} - \mathbf{z}\|_{w^\rho} < \frac{\delta(\varepsilon)\left(1-(1-\epsilon)^{1-\rho}\right)}{\sigma_{\max}(\mathbf{c})}, \quad \text{then} \quad \left|H_h^{A,\mathbf{c}}(\mathbf{s}) - H_h^{A,\mathbf{c}}(\mathbf{z})\right| < \varepsilon. \tag{6.6}$$

Indeed,

$$\begin{aligned}\left\|\sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{s}_{t-i} - \sum_{i=0}^{\infty} A^i \mathbf{c}\mathbf{z}_{t-i}\right\| &= \left\|\sum_{i=0}^{\infty} A^i \mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \leq \sum_{i=0}^{\infty} \left\|A^i \mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \\ &\leq \sum_{i=0}^{\infty} \sigma_{\max}(A^i) \left\|\mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \leq \sum_{i=0}^{\infty} \sigma_{\max}(A)^i \left\|\mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \leq \sum_{i=0}^{\infty} (1-\epsilon)^i \left\|\mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\|.\end{aligned}$$

If we now use (2.5) in Lemma 1 and the hypothesis in (6.6), we can conclude that

$$\sum_{i=0}^{\infty} (1-\epsilon)^i \left\|\mathbf{c}(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \leq \sigma_{\max}(\mathbf{c}) \sum_{i=0}^{\infty} (1-\epsilon)^i \left\|(\mathbf{s}_{t-i} - \mathbf{z}_{t-i})\right\| \leq \frac{\sigma_{\max}(\mathbf{c})\|\mathbf{s} - \mathbf{z}\|_{w^\rho}}{1-(1-\epsilon)^{1-\rho}} < \delta(\varepsilon),$$

which proves the continuity of the map $H_h^{A,\mathbf{c}} : (K_M, \|\cdot\|_{w^\rho}) \longrightarrow \mathbb{R}$ and hence shows that $H_h^{A,\mathbf{c}}$ is a λ_ρ-exponential fading memory filter.

In order to establish the universality statement in the corollary we will proceed, as in the proof of Theorem 8, by showing that $\mathcal{L}_\epsilon$ is a polynomial algebra that contains the constant functionals and separates the points in K_M and then by invoking the Stone-Weierstrass theorem using the compactness of $(K_M, \|\cdot\|_{w^\rho})$.

In order to show that $(\mathcal{L}_\epsilon, \|\cdot\|_{w^\rho})$ is a polynomial algebra, notice first that if $A_1, A_2 \in \mathbb{M}_N$ are such that $\sigma_{\max}(A_1), \sigma_{\max}(A_2) < 1-\epsilon$, then

$$\sigma_{\max}(A_1 \oplus A_2) = \max\left(\sigma_{\max}(A_1), \sigma_{\max}(A_2)\right) < 1-\epsilon. \tag{6.7}$$

If we now take $\mathbf{c}_i \in \mathbb{M}_{N_i,n}$, $i \in \{1,2\}$ and h_1, h_2 two real-valued polynomials in N_1 and N_2 variables, respectively, we have by the first part of the corollary that we just proved that the filter functionals $H_{h_1}^{A_1,\mathbf{c}_1}$ and $H_{h_2}^{A_2,\mathbf{c}_2}$ are well defined. Additionally, by (3.3)-(3.4) so are the combinations $H_{h_1}^{A_1,\mathbf{c}_1} \cdot H_{h_2}^{A_2,\mathbf{c}_2}$ and $H_{h_1}^{A_1,\mathbf{c}_1} + \lambda H_{h_2}^{A_2,\mathbf{c}_2}$ that satisfy:

$$H_{h_1}^{A_1,\mathbf{c}_1} \cdot H_{h_2}^{A_2,\mathbf{c}_2} = H_{h_1 \cdot h_2}^{A_1 \oplus A_2, \mathbf{c}_1 \oplus \mathbf{c}_2}, \quad H_{h_1}^{A_1,\mathbf{c}_1} + \lambda H_{h_2}^{A_2,\mathbf{c}_2} = H_{h_1 \oplus \lambda h_2}^{A_1 \oplus A_2, \mathbf{c}_1 \oplus \mathbf{c}_2}, \quad \lambda \in \mathbb{R}. \tag{6.8}$$

Using the relations (6.8) and (6.7), we can conclude that both $H_{h_1}^{A_1,\mathbf{c}_1} \cdot H_{h_2}^{A_2,\mathbf{c}_2}$ and $H_{h_1}^{A_1,\mathbf{c}_1} + \lambda H_{h_2}^{A_2,\mathbf{c}_2}$ belong to $\mathcal{L}_\epsilon \subset \mathcal{R}_{w^\rho}$. This implies that $(\mathcal{L}_\epsilon, \|\cdot\|_{w^\rho})$ is a polynomial subalgebra of $(\mathcal{R}_{w^\rho}, \|\cdot\|_{w^\rho})$

Since $\mathcal{L}_\epsilon$ contains the constant functionals (just take constant readout maps h), in order to conclude the proof, it is enough to show that the elements in $\mathcal{L}_\epsilon$ separate points in K_M. In the proof of this statement we need the following elementary fact about analytic functions.

Lemma 36 *Let $M > 0$ and let $\mathbf{z} \in [-M, M]^{\mathbb{Z}_-}$. Define the real valued function $f_{\mathbf{z}}(x) := \sum_{j=0}^{\infty} z_{-j} x^j$. This function is real analytic in the interval $(-1,1)$. Moreover, if $\mathbf{z} \neq \mathbf{0}$, then there exists a point $x_0 \in (-1,1)$ such that $f_{\mathbf{z}}(x_0) \neq 0$.*

Proof of the lemma. We note first that for any $x \in (-1,1)$ and any $s \in \mathbb{N}$ we have that

$$\left|\sum_{j=0}^{s} z_{-j} x^j\right| \leq \sum_{j=0}^{s} |z_{-j}| \left|x^j\right| \leq M \sum_{j=0}^{s} |x|^j \leq \frac{M}{1-|x|}.$$

Taking the limit $s \to \infty$, we obtain that

$$|f_{\mathbf{z}}(x)| \leq \frac{M}{1-|x|}, \quad \text{for all } x \in (-1,1),$$

which proves the first claim in the lemma. Now, by the uniqueness theorem for the representation of analytic functions by power series (see Brown and Churchill (2009), page 217), the series $\sum_{j=0}^{\infty} z_{-j} x^j$ is the Taylor expansion around 0 of $f_{\mathbf{z}}(x)$. Since $\mathbf{z} \neq \mathbf{0}$ by hypothesis, some of the derivatives of $f_{\mathbf{z}}(x)$ are non-zero and hence this function cannot be flat, which implies that there exists a point $x_0 \in (-1,1)$ such that $f_{\mathbf{z}}(x_0) \neq 0$. ▼

We now show that the elements in $\mathcal{L}_\epsilon$ separate points in K_M. Take $\mathbf{z}_1, \mathbf{z}_2 \in K_M \subset (\mathbb{R}^n)^{\mathbb{Z}_-}$ such that $\mathbf{z}_1 \neq \mathbf{z}_2$ and let $A \in \mathbb{M}(n,n)$, with $\sigma_{\max}(A) < 1-\epsilon$, and $\mathbf{c} := \mathbb{I}_n$. Let $U^{A,\mathbf{c}} : K_M \longrightarrow (\mathbb{R}^n)^{\mathbb{Z}_-}$ be the linear filter associated to A and $\mathbf{c}$ via the recursion (3.7). Using the preceding arguments we have that

$$U^{A,\mathbf{c}}(\mathbf{z})_t = \sum_{j=0}^{\infty} A^j \mathbf{z}_{t-j}. \tag{6.9}$$

Since $\mathbf{z}_1 \neq \mathbf{z}_2$, then there exists and index $i \in \{1, \ldots, n\}$ and $t \in \mathbb{Z}_-$ such that $\left(z_1^i\right)_t \neq \left(z_2^i\right)_t$. Let now $b \in (-1+\epsilon, 1-\epsilon)$ and let $A_b := \operatorname{diag}(0, \ldots, 0, b, 0, \ldots, 0) \in \mathbb{D}_n$ be the matrix that has the element b in the i-th entry. It is easy to see using (6.9) that

$$U^{A_b,\mathbf{c}}(\mathbf{z})_t = \left(0, \ldots, 0, \sum_{j=0}^{\infty} b^j z_{t-j}^i, 0, \ldots, 0\right)^\top, \quad \text{with} \quad \sum_{j=0}^{\infty} b^j z_{t-j}^i \quad \text{in the } i\text{-th entry.} \tag{6.10}$$

Let $\mathbf{s} := \mathbf{z}_1 - \mathbf{z}_2 \neq \mathbf{0}$. Notice that by (6.10) we have that $U^{A_b,\mathbf{c}}(\mathbf{s})_0 = \left(0, \ldots, 0, \sum_{j=0}^{\infty} b^j s_{-j}^i, 0, \ldots, 0\right)^\top$. Given that the vector $\mathbf{s}^i \in \mathbb{R}^{\mathbb{Z}_-}$ is non-zero, Lemma 36, implies the existence of an element $b_0 \in$

$(-1+\epsilon, 1-\epsilon)$ such that $U^{A_{b_0},\mathbf{c}}(\mathbf{s})_0 \neq \mathbf{0}$, which is equivalent to $U^{A_{b_0},\mathbf{c}}(\mathbf{z}_1)_0 \neq U^{A_{b_0},\mathbf{c}}(\mathbf{z}_2)_0$. Using the polynomial $h(\mathbf{x}) := x_i \in \mathbb{R}$, the previous relation implies that $U_h^{A_{b_0},\mathbf{c}}(\mathbf{z}_1)_0 \neq U_h^{A_{b_0},\mathbf{c}}(\mathbf{z}_2)_0$ or, equivalently,

$$H_h^{A_{b_0},\mathbf{c}}(\mathbf{z}_1) \neq H_h^{A_{b_0},\mathbf{c}}(\mathbf{z}_2), \quad \text{as required.}$$

We conclude the proof by establishing the universality the families $\mathcal{DL}_\epsilon$ and $\mathcal{NL}$ formed by the linear reservoir filters generated by diagonal and nilpotent matrices, respectively. First, in the case of $\mathcal{DL}_\epsilon$, the statement is a consequence of (6.8) and of the fact that when the matrices A_1 and A_2 are diagonal, then the matrix associated to the linear map $A_1 \oplus A_2$ is also diagonal. Additionally, notice that the point separation property for $\mathcal{L}_\epsilon$ has been proved using diagonal matrices in (6.10) and hence it also holds for $\mathcal{DL}_\epsilon$. The claim follows from the Stone-Weierstrass theorem.

Finally, in the case of $\mathcal{NL}$, the proof also follows from (6.8) since it is straightforward to see that when the matrices A_1 and A_2 are nilpotent, then the matrix associated to the linear map $A_1 \oplus A_2$ is also nilpotent. It is only the point separation property of $\mathcal{N}$ that requires a separate argument that we provide in the following lines. Let $\mathbf{z}_1, \mathbf{z}_2 \in K_M$ such that $\mathbf{z}_1 \neq \mathbf{z}_2$ and let $t_0 \in \mathbb{N}$ be the first time index for which $(\mathbf{z}_1)_{-t_0} \neq (\mathbf{z}_2)_{-t_0}$, that is, $(\mathbf{z}_1)_{-t} = (\mathbf{z}_2)_{-t}$, for all $t \in \{0, 1, \ldots, t_0 - 1\}$. Let now $i_0 \in \{1, \ldots, n\}$ be such that $\left(z_1^{i_0}\right)_{-t_0} \neq \left(z_2^{i_0}\right)_{-t_0}$. Let now $A_{t_0+1} \in \mathbb{N}\mathrm{il}_{t_0+1}^{t_0+1}$ be the upper shift matrix in dimension t_0+1, that is, $A_{t_0+1} \in \mathbb{M}_{t_0+1}$ is by definition a superdiagonal matrix with a diagonal of ones above the main diagonal, and construct an element $\mathbf{c} \in \mathbb{M}_{t_0+1,n}$ whose last row is given by a vector of zeros with the exception of a one in the entry i_0. The nilpotency of A_{t_0+1} implies

$$U^{A_{t_0+1},\mathbf{c}}(\mathbf{z})_0 = \sum_{j=0}^{t_0} A_{t_0+1}^j \mathbf{c} \mathbf{z}_{-j}.$$

When we apply this expression to $\mathbf{z}_1$ and $\mathbf{z}_2$, since $(\mathbf{z}_1)_{-t} = (\mathbf{z}_2)_{-t}$, for all $t \in \{0, 1, \ldots, t_0 - 1\}$, we obtain that

$$U^{A_{t_0+1},\mathbf{c}}(\mathbf{z}_1 - \mathbf{z}_2)_0 = A_{t_0+1}^{t_0} \mathbf{c}(\mathbf{z_1} - \mathbf{z_2})_{-t_0} = \left(0, \ldots, 0, \left(z_1^{i_0}\right)_{-t_0} - \left(z_2^{i_0}\right)_{-t_0}\right)^\top \neq \mathbf{0}.$$

Using the polynomial $h(\mathbf{x}) := x_{t_0+1}$, this relation implies that $U_h^{A_{t_0+1},\mathbf{c}}(\mathbf{z}_1)_0 \neq U_h^{A_{t_0+1},\mathbf{c}}(\mathbf{z}_2)_0$ or, equivalently, $H_h^{A_{t_0+1},\mathbf{c}}(\mathbf{z}_1) \neq H_h^{A_{t_0+1},\mathbf{c}}(\mathbf{z}_2)$, as required. ■

6.6 Proof of Proposition 14

We start by noting, as we did in the proof of Corollary 11, that the condition (3.13) implies that the reservoir map associated to (3.11) is a contraction and hence, by Lemma 35, it satisfies the echo state property and has a well-defined associated filter.

We now prove that the condition (3.13) implies the convergence of the series in the expression (3.14). Let $K_1 := \max_{z\in I} \|p(z)\|_2 = \max_{z\in I} \sigma_{\max}(p(z)) < 1$ and $K_2 := \max_{z\in I} \|q(z)\|_2 = \max_{z\in I} \sigma_{\max}(q(z))$; notice that K_1 and K_2 are well-defined due to the compactness of I. Let now $n, m \in \mathbb{N}$ be such that $n < m$ and let $S_n := \sum_{j=0}^n \left(\prod_{k=0}^{j-1} p(z_{t-k})\right) q(z_{t-j}) \in \mathbb{R}^N$. Then,

$$\begin{aligned}\|S_n - S_m\| &= \left\| \sum_{j=n+1}^{m} \left(\prod_{k=0}^{j-1} p(z_{t-k})\right) q(z_{t-j}) \right\| \leq \sum_{j=n+1}^{m} \left\| \prod_{k=0}^{j-1} p(z_{t-k}) \right\|_2 \|q(z_{t-j})\| \\ &\leq \sum_{j=n+1}^{m} \prod_{k=0}^{j-1} \|p(z_{t-k})\|_2 \|q(z_{t-j})\| \leq K_2 \sum_{j=n+1}^{m} K_1^j \leq K_2 \sum_{j=n+1}^{\infty} K_1^j = \frac{K_2 K_1^{n+1}}{1-K_1}.\end{aligned}$$

The condition $K_1 < 1$ implies that $\frac{K_2 K_1^{n+1}}{1-K_1} \to 0$ as $n \to \infty$ and hence $\{S_n\}_{n\in\mathbb{N}}$ is a Cauchy sequence in $\mathbb{R}^N$ that consequently converges. This proves the convergence of the infinite series in (3.14) and the causal character of the filter that it defines. The time-invariance can also be easily established by mimicking the verification that we carried out in the proof of Corollary 11. We now prove that (3.14) is indeed a solution of (3.11):

$$\begin{aligned}p(z_t)\mathbf{x}_{t-1} + q(z_t) &= p(z_t)\left(\sum_{j=0}^{\infty}\left(\prod_{k=0}^{j-1} p(z_{t-1-k})\right) q(z_{t-1-j})\right) + q(z_t) = q(z_t) + p(z_t)q(z_{t-1}) \\ &+ p(z_t)p(z_{t-1})q(z_{t-2}) + p(z_t)p(z_{t-1})p(z_{t-2})q(z_{t-3}) + \cdots = \sum_{j=0}^{\infty}\left(\prod_{k=0}^{j-1} p(z_{t-k})\right) q(z_{t-j}) = \mathbf{x}_t.\end{aligned}$$

We conclude by proving the inequality in (3.16). Note first that for any $m \in \mathbb{N}$,

$$\begin{aligned}\left\|\sum_{j=0}^{m}\left(\prod_{k=0}^{j-1} p(z_{t-k})\right) q(z_{t-j})\right\| &\leq \sum_{j=0}^{m}\left\|\prod_{k=0}^{j-1} p(z_{t-k})\right\|_2 \|q(z_{t-j})\| \\ &\leq \sum_{j=0}^{m}\prod_{k=0}^{j-1} \|p(z_{t-k})\|_2 \|q(z_{t-j})\| \leq \frac{K_2\left(1-K_1^{m+1}\right)}{1-K_1},\end{aligned}$$

and hence, by the continuity of the norm and for any $t \in \mathbb{Z}$:

$$\|\mathbf{x}_t\| = \lim_{m\to\infty}\left\|\sum_{j=0}^{m}\left(\prod_{k=0}^{j-1} p(z_{t-k})\right) q(z_{t-j})\right\| \leq \lim_{m\to\infty}\frac{K_2\left(1-K_1^{m+1}\right)}{1-K_1} = \frac{K_2}{1-K_1}. \quad ■$$

6.7 Proof of Lemma 15

(i) $\Longrightarrow$ (ii): $\|A_0\|_2 + \|A_1\|_2 + \cdots + \|A_n\|_2 < \sum_{i=0}^{n_1} \lambda = \lambda(n_1+1) < 1$.
(ii) $\Longrightarrow$ (iii): $\|p(z)\|_2 = \|A_0 + zA_1 + z^2A_2 + \cdots + z^{n_1}A_{n_1}\|_2 \leq \|A_0\|_2 + |z|\|A_1\|_2 + |z^2|\|A_2\|_2 + \cdots + |z^{n_1}|\|A_{n_1}\|_2 < \|A_0\|_2 + \|A_1\|_2 + \cdots + \|A_{n_1}\|_2 < 1$. ■

6.8 Proof of Proposition 16

We start by formulating and proving an elementary result that will be needed later on.

Lemma 37 *Let $\mathbf{f} : U \subset \mathbb{R}^n \longrightarrow \mathbb{M}_m$ be a differentiable function defined on the convex set U. For any $\mathbf{z} \in U$ denote by $\partial_i \mathbf{f}(\mathbf{z}) \in \mathbb{M}_m$ the matrix containing the partial derivatives of the components of $\mathbf{f}$ with respect to their ith-entry, $i \in \{1, \ldots, n\}$. Then, for any $\mathbf{x}, \mathbf{y} \in U$ we have:*

$$\|\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x})\|_2 \leq \sqrt{nm} \max_{i\in\{1,\ldots,n\}}\left(\sup_{\mathbf{z}\in U}\{\|\partial_i \mathbf{f}(\mathbf{z})\|_2\}\right)\|\mathbf{x} - \mathbf{y}\|. \tag{6.11}$$

Proof. Given $A = (A_{i,j}) \in \mathbb{M}_{n,m}$, let $\|A\|_F := \mathrm{tr}\left(A^\top A\right) = \sum_{i=1}^n \sum_{j=1}^m A_{i,j}^2$ be its Frobenius norm. Recall (see Theorem 5.6.34 and Exercise 5.6.P24 in Horn and Johnson (2013)) that

$$\|A\|_2 \leq \|A\|_F \leq \sqrt{r}\,\|A\|_2, \tag{6.12}$$

where r is the rank of A. Consider now $\mathbf{x}, \mathbf{y} \in U$ arbitrary and let $D\mathbf{f}(\mathbf{z}) : \mathbb{R}^n \longrightarrow \mathbb{M}_m$ be the differential of $\mathbf{f}$ evaluated at $\mathbf{z} \in U$. The convexity of U implies that the Mean Value Inequality holds (see Theorem 2.4.8 in Abraham et al. (1988)) and hence:

$$\|\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x})\|_F \leq \sup_{t\in[0,1]} \{\|D\mathbf{f}((1-t)\mathbf{x} + t\mathbf{y})\|_2 \|\mathbf{x} - \mathbf{y}\|\}. \tag{6.13}$$

The first inequality in (6.12) and (6.13) imply that

$$\|\mathbf{f}(\mathbf{y}) - \mathbf{f}(\mathbf{x})\|_2 \leq \sup_{\mathbf{z}\in U} \{\|D\mathbf{f}(\mathbf{z})\|_2 \|\mathbf{x} - \mathbf{y}\|\}. \tag{6.14}$$

At the same time, notice that by (6.12)

$$\|D\mathbf{f}(\mathbf{z})\|_2^2 \leq \|D\mathbf{f}(\mathbf{z})\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{m} \partial_i f_{jk}^2(\mathbf{z}) = \sum_{i=1}^{n} \|\partial_i \mathbf{f}(\mathbf{z})\|_F^2$$
$$\leq m\sum_{i=1}^{n} \|\partial_i \mathbf{f}(\mathbf{z})\|_2^2 \leq mn \max_{i\in\{1,..,n\}} \left(\|\partial_i \mathbf{f}(\mathbf{z})\|_2^2\right).$$

This inequality, together with (6.14), imply the statement (6.11) since the maximum and the supremum can be trivially exchanged. ▼

We now carry out the proof of the proposition under the hypothesis **(iii)** in Lemma 15 which is implied by the other two. The modifications necessary to establish the result under the other two hypotheses are straightforward. Consider two arbitrary elements $\mathbf{z}, \mathbf{s} \in I^{\mathbb{Z}_-}$. Then, by the Cauchy-Schwarz and Minkowski inequalities:

$$|H^{p,q}_{\mathbf{W}}(\mathbf{z}) - H^{p,q}_{\mathbf{W}}(\mathbf{s})| = \left|\mathbf{W}^\top \left[\sum_{j=0}^{\infty}\left(\left(\prod_{k=0}^{j-1} p(z_{-k})\right) q(z_{-j}) - \left(\prod_{k=0}^{j-1} p(s_{-k})\right) q(s_{-j})\right)\right]\right|$$
$$\leq \|\mathbf{W}\| \sum_{j=0}^{\infty} \left\|a_j(\underline{z_{-j+1}})q(z_{-j}) - a_j(\underline{s_{-j+1}})q(s_{-j})\right\|, \quad \text{where} \quad a_j(\underline{z_{-j+1}}) := \prod_{k=0}^{j-1} p(z_{-k}). \tag{6.15}$$

We now bound the right hand side of (6.15) as follows:

$$\sum_{j=0}^{\infty} \left\|a_j(\underline{z_{-j+1}})q(z_{-j}) - a_j(\underline{s_{-j+1}})q(s_{-j})\right\|$$
$$= \sum_{j=0}^{\infty} \left\|a_j(\underline{z_{-j+1}})q(z_{-j}) + a_j(\underline{z_{-j+1}})q(s_{-j}) - a_j(\underline{z_{-j+1}})q(s_{-j}) - a_j(\underline{s_{-j+1}})q(s_{-j})\right\|$$
$$\leq \sum_{j=0}^{\infty} \left\|a_j(\underline{z_{-j+1}})\right\|_2 \|q(z_{-j}) - q(s_{-j})\| + \left\|a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}})\right\|_2 \|q(s_{-j})\| \tag{6.16}$$

If L_q is a Lipschitz constant of $q : I \longrightarrow \mathbb{R}^N$ then

$$\left\|a_j(\underline{z_{-j+1}})\right\|_2 \|q(z_{-j}) - q(s_{-j})\| \leq M_p^j L_q |z_{-j} - s_{-j}|, \tag{6.17}$$

which inserted in (6.16) and in (6.15) implies that

$$|H^{p,q}_{\mathbf{W}}(\mathbf{z}) - H^{p,q}_{\mathbf{W}}(\mathbf{s})| \leq \|\mathbf{W}\| L_q \left[\sum_{j=0}^{\infty} M_p^j |z_{-j} - s_{-j}| + \sum_{j=0}^{\infty} \left\|a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}})\right\|_2\right] \tag{6.18}$$

We now bound above the second summand in (6.18) using the inequality (6.11) in the statement of Lemma 37 as well as the following identity:

$$a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}}) = \sum_{l=0}^{j-1}(p(s_0)\cdots p(s_{-(l-1)})\cdot p(z_{-l})\cdot p(z_{-(l+1)})\cdots p(z_{-(j-1)})$$
$$- p(s_0)\cdots p(s_{-(l-1)})\cdot p(s_{-l})\cdot p(z_{-(l+1)})\cdots p(z_{-(j-1)})). \tag{6.19}$$

This equality simply follows from writing:

$$\begin{aligned}
a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}}) &= \prod_{l=0}^{j-1} p(z_{-l}) - \prod_{l=0}^{j-1} p(s_{-l}) = p(z_0)p(z_{-1})\cdots p(z_{-(j-1)}) - p(s_0)p(s_{-1})\cdots p(s_{-(j-1)}) \\
&= p(z_0)p(z_{-1})\cdots p(z_{-(j-1)}) - p(s_0)p(s_{-1})\cdots p(s_{-(j-1)}) \\
&+ \Big\{ p(s_0)p(z_{-1})\cdots p(z_{-(j-1)}) - p(s_0)p(z_{-1})\cdots p(z_{-(j-1)}) \\
&+ p(s_0)p(s_{-1})p(z_{-2})\cdots p(z_{-(j-1)}) - p(s_0)p(s_{-1})p(z_{-2})\cdots p(z_{-(j-1)}) \\
&+\cdots + p(s_0)\cdots p(s_{-(l-1)})p(z_{-l})p(z_{-(l+1)})\cdots p(z_{-(j-1)}) - p(s_0)\cdots p(s_{-(l-1)})p(z_{-l})p(z_{-(l+1)})\cdots p(z_{-(j-1)}) \\
&+ \cdots + p(s_0)\cdots p(s_{-(j-2)})p(z_{-(j-1)}) - p(s_0)\cdots p(s_{-(j-2)})p(z_{-(j-1)}) \Big\} \\
&= \sum_{l=0}^{j-1}(p(s_0)\cdots p(s_{-(l-1)})\cdot p(z_{-l})\cdot p(z_{-(l+1)})\cdots p(z_{-(j-1)}) \\
&\qquad - p(s_0)\cdots p(s_{-(l-1)})\cdot p(s_{-l})\cdot p(z_{-(l+1)})\cdots p(z_{-(j-1)})),
\end{aligned}$$

where the $2(j-1)$ summands inside the braces are obtained by adding and subtracting polynomials recursively constructed out of $a_j(\underline{z_{-j+1}})$ by changing the variables of the first k factors, $k \in \{1, \cdots, j-1\}$. We then combine all the $(2k-1)$-th with the $(2k+2)-th$ summands of the resulting expression in order to obtain the first $j-1$ terms in the sum in (6.19). Then the last j-th term results from combining the second with the one before last summands, that is, $p(s_0)p(s_{-1})\cdots p(s_{-(j-1)})$ and $p(s_0)\cdots p(s_{-(j-2)})p(z_{-(j-1)})$, respectively.

Using the relation (6.19) we can write:

$$\left\|a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}})\right\|_2 \leq \sum_{l=0}^{j-1} \left\|p(s_0)\cdots p(s_{-(l-1)})\cdot(p(z_{-l}) - p(s_{-l}))\cdot p(z_{-(l+1)})\cdots p(z_{-(j-1)})\right\|_2$$
$$\leq \sum_{l=0}^{j-1} \|p(s_0)\|_2 \cdots \left\|p(s_{-(l-1)})\right\|_2 \cdot \|p(z_{-l}) - p(s_{-l})\|_2 \cdot \left\|p(z_{-(l+1)})\right\|_2 \cdots \left\|p(z_{-(j-1)})\right\|_2$$
$$\leq M_p^{j-1}\sqrt{N}\sup_{z\in I}\{\|p'(z)\|_2\} \sum_{l=1}^{j} |z_{-j+l} - s_{-j+l}|,$$

where the last inequality is a consequence of (6.11). Let $M_{p'} := \sqrt{N}\sup_{z\in I}\{\|p'(z)\|_2\}$, then

$$\left\|a_j(\underline{z_{-j+1}}) - a_j(\underline{s_{-j+1}})\right\|_2 \leq \frac{M_{p'}}{M_p} M_p^j \sum_{l=1}^{j} |z_{-j+l} - s_{-j+l}| = \frac{M_{p'}}{M_p} \sum_{l=1}^{j} M_p^l M_p^{j-l} \left|z_{-(j-l)} - s_{-(j-l)}\right|$$

Since the last term in this inequality is one summand of the Cauchy product of the series with general terms M_p^j and $M_p^j\left|z_{-j}-s_{-j}\right|$ and these two series are absolutely convergent (recall the statement (2.4)), we can conclude (see, for instance, §8.24 in Apostol (1974)) that

$$\sum_{j=0}^{\infty}\left\|a_j(\underline{z_{-j+1}})-a_j(\underline{s_{-j+1}})\right\|_2 \leq \frac{M_{p'}}{M_p}\sum_{j=0}^{\infty}\sum_{l=1}^{j}M_p^l M_p^{j-l}\left|z_{-(j-l)}-s_{-(j-l)}\right| \\ = \frac{M_{p'}}{M_p}\frac{1}{1-M_p}\sum_{j=0}^{\infty}M_p^j\left|z_{-j}-s_{-j}\right|.$$

If we now substitute this relation in (6.18) and we use Lemma 1 with weighting sequences $w_t^\rho := M_p^{\rho t}$, for any $\rho\in(0,1)$, we obtain that:

$$\begin{aligned}\left|H_{\mathbf{W}}^{p,q}(\mathbf{z})-H_{\mathbf{W}}^{p,q}(\mathbf{s})\right| &\leq \|\mathbf{W}\|\,L_q\left(1+\frac{M_{p'}}{M_p}\frac{1}{1-M_p}\right)\sum_{j=0}^{\infty}M_p^j\left|z_{-j}-s_{-j}\right| \\ &\leq \|\mathbf{W}\|\,L_q\left(1+\frac{M_{p'}}{M_p}\frac{1}{1-M_p}\right)\left(\frac{1}{1-M_p^{1-\rho}}\right)\|\mathbf{z}-\mathbf{s}\|_{w^\rho},\end{aligned}$$

which proves the continuity of the map $H_{\mathbf{W}}^{p,q}:(I^{\mathbb{Z}_-},\|\cdot\|_{w^\rho})\longrightarrow\mathbb{R}$, as required. ■

6.9 Proof of Proposition 17

We first recall that since by hypothesis the reservoir functionals $H_{\mathbf{W}_1}^{p_1,q_1}, H_{\mathbf{W}_2}^{p_2,q_2}$ are well-defined then, by the comments that follow (3.5), so are $H_{\mathbf{W}_1}^{p_1,q_1}+\lambda H_{\mathbf{W}_2}^{p_2,q_2}$ and $H_{\mathbf{W}_1}^{p_1,q_1}\cdot H_{\mathbf{W}_2}^{p_2,q_2}$.

The proof of (**i**) is a straightforward verification. As to (**ii**), denote first by y_t^1, y_t^2 and $\mathbf{x}_t^1,\mathbf{x}_t^2$ the outputs and the state variables, respectively, of the SAS corresponding to the two functionals that we are considering. We note first that by (3.12):

$$y_t^1\cdot y_t^2=\mathbf{W}_1^\top\mathbf{x}_t^1\cdot\mathbf{W}_2^\top\mathbf{x}_t^2=(\mathbf{W}_1\otimes\mathbf{W}_2)^\top(\mathbf{x}_t^1\otimes\mathbf{x}_t^2).$$

Using (3.11) it can be readily verified that the time evolution of the tensor product $\mathbf{x}_t^1\otimes\mathbf{x}_t^2$ is given by

$$\begin{aligned}\mathbf{x}_t^1\otimes\mathbf{x}_t^2 &= (p_1(z_t)\otimes p_2(z_t))(\mathbf{x}_{t-1}^1\otimes\mathbf{x}_{t-1}^2)+p_1(z_t)\mathbf{x}_{t-1}^1\otimes q_2(z_t)+q_1(z_t)\otimes p_2(z_t)\mathbf{x}_{t-1}^2+q_1(z_t)\otimes q_2(z_t), \\ &= (p_1\otimes p_2)(z_t)(\mathbf{x}_{t-1}^1\otimes\mathbf{x}_{t-1}^2)+p_1(z_t)\mathbf{x}_{t-1}^1\otimes q_2(z_t)+q_1(z_t)\otimes p_2(z_t)\mathbf{x}_{t-1}^2+(q_1\otimes q_2)(z_t),\end{aligned}$$

which proves (3.23) and hence (3.22).

In order to show that the reservoir functionals on the right hand side of (3.21) and (3.22) are well-defined we prove the following lemma.

Lemma 38 *Let $p_1(z)\in\mathbb{M}_{N_1,M_1}[z]$ and $p_2(z)\in\mathbb{M}_{N_2,M_2}[z]$ be two polynomials with matrix coefficients and assume that they satisfy that $\|p_1(z)\|_2<1-\epsilon$ and $\|p_2(z)\|_2<1-\epsilon$ for all $z\in I:=[-1,1]$ and a given $0<\epsilon>1$. Then:*

(i) $\|p_1\oplus p_2(z)\|_2<1-\epsilon$,

(ii) $\|p_1\otimes p_2(z)\|_2<1-\epsilon$,

for all $z\in I:=[-1,1]$.

Proof of the lemma. Let $\mathbf{x}=\mathbf{x}_1\oplus\mathbf{x}_2\in\mathbb{R}^{M_1}\oplus\mathbb{R}^{M_2}$. Then, in order to prove part **(i)** note that

$$\begin{aligned}\|(p_1\oplus p_2)(z)\cdot\mathbf{x}\|^2 &= \|(p_1(z)\cdot\mathbf{x}_1,p_2(z)\cdot\mathbf{x}_2)\|^2=\|p_1(z)\cdot\mathbf{x}_1\|^2+\|p_2(z)\cdot\mathbf{x}_2\|^2 \\ &\leq \|p_1(z)\|_2^2\|\mathbf{x}_1\|^2+\|p_2(z)\|_2^2\|\mathbf{x}_2\|^2\leq(1-\epsilon)^2\left(\|\mathbf{x}_1\|^2+\|\mathbf{x}_2\|^2\right)=(1-\epsilon)^2\|\mathbf{x}\|^2.\end{aligned}$$

This inequality implies that

$$\|p_1\oplus p_2(z)\|_2=\sup_{\mathbf{x}\neq\mathbf{0}}\left\{\frac{\|(p_1\oplus p_2)(z)\cdot\mathbf{x}\|}{\|\mathbf{x}\|}\right\}\leq\sup_{\mathbf{x}\neq\mathbf{0}}\left\{\frac{(1-\epsilon)\|\mathbf{x}\|}{\|\mathbf{x}\|}\right\}=1-\epsilon,\quad\text{as required.}$$

As to the statement in part **(ii)**:

$$\|p_1\otimes p_2(z)\|_2=\sigma_{\max}(p_1\otimes p_2(z))=\sigma_{\max}(p_1(z))\sigma_{\max}(p_2(z))=\|p_1(z)\|_2\|p_2(z)\|_2<(1-\epsilon)^2<(1-\epsilon).\ \blacktriangledown$$

Now, the first part of this lemma and Proposition 14 guarantee that $H_{\mathbf{W}_1\oplus\lambda\mathbf{W}_2}^{p_1\oplus p_2,q_1\oplus q_2}$ is well-defined. The same conclusion holds for $H_{\mathbf{0}\oplus\mathbf{0}\oplus(\mathbf{W}_1\otimes\mathbf{W}_2)}^{p,q_1\oplus q_2\oplus(q_1\otimes q_2)}$ because due to the block diagonal character of (3.23) then $\sigma_{\max}(p(z))=\sigma_{\max}((p_1(z)\oplus p_2(z)\oplus(p_1\otimes p_2)(z))=\|p_1(z)\oplus p_2(z)\oplus(p_1\otimes p_2)(z)\|_2$. By parts **(i)** and **(ii)** in Lemma 38 we can conclude that $\|p(z)\|_2<1-\epsilon$ for all $z\in[-1,1]$ and, again by Proposition 14, the reservoir functional $H_{\mathbf{0}\oplus\mathbf{0}\oplus(\mathbf{W}_1\otimes\mathbf{W}_2)}^{p,q_1\oplus q_2\oplus(q_1\otimes q_2)}$ is well-defined. ■

6.10 Proof of Theorem 19

Note first that the hypothesis $M_p<1-\epsilon<1$ on the polynomials p associated to the elements in $\mathcal{S}_\epsilon$ implies, by Propositions 14 and 16, that this family is made of time-invariant reservoir filters that have the FMP with respect to weighting sequences of the form $w_t^p:=M_p^{pt}$, $\rho\in(0,1)$. Additionally, using Lemma 7 and the hypothesis $M_p<1-\epsilon$, for a fixed given $\epsilon\in(0,1)$, we can conclude that all the reservoir filters in $\mathcal{S}_\epsilon$ have the FMP with the common weighting sequence $w_t^\rho:=(1-\epsilon)^{\rho t}$, $\rho\in(0,1)$.

The elements in $\mathcal{S}_\epsilon$ form a polynomial algebra as a consequence of Lemma 38 and Proposition 17. Moreover, the family $\mathcal{S}_\epsilon$ has the point separation property and contains all the constant functionals. Indeed, since $\mathcal{S}_\epsilon$ includes the linear family $\mathcal{L}_\epsilon$, we recall that in Appendix 6.5 we proved that given $\mathbf{z}_1,\mathbf{z}_2\in K_M\subset(\mathbb{R}^n)^{\mathbb{Z}_-}$ such that $\mathbf{z}_1\neq\mathbf{z}_2$, there exists $A\in\mathbb{M}(n,n)$, with $\sigma_{\max}(A)<1-\epsilon$ and $\mathbf{c}:=\mathbb{I}_n$ such that $U^{A,\mathbf{c}}(\mathbf{z}_1)_0\neq U^{A,\mathbf{c}}(\mathbf{z}_2)_0$. The point separation property follows from choosing any vector $\mathbf{W}\in\mathbb{R}^N$ such that $\mathbf{W}^\top(U^{A,\mathbf{c}}(\mathbf{z}_1))_0\neq\mathbf{W}^\top(U^{A,\mathbf{c}}(\mathbf{z}_2))_0$, which implies that $U_{\mathbf{W}}^{A,\mathbf{c}}(\mathbf{z}_1)_0\neq U_{\mathbf{W}}^{A,\mathbf{c}}(\mathbf{z}_2)_0$ and hence $H_{U_{\mathbf{W}}^{A,\mathbf{c}}}(\mathbf{z}_1)\neq H_{U_{\mathbf{W}}^{A,\mathbf{c}}}(\mathbf{z}_2)$, as required.

All the constant functionals can be obtained by taking for p the zero polynomial and for q the constant polynomials (q has degree zero). In that case, the state variables are a constant sequence $\mathbf{x}_t=q$ and the associated functional is the constant map $H_{\mathbf{W}}^{0,q}(\mathbf{z})=\mathbf{W}^\top q$, for all $\mathbf{z}\in K_M$.

The universality result follows hence from the Stone-Weierstrass Theorem and the compactness of $(I^{\mathbb{Z}_-},\|\cdot\|_{w^\rho})$ established in Lemma 2.

Finally, we prove the statement regarding the family $\mathcal{NS}_\epsilon$ determined by nilpotent polynomials p. First, by expressions (3.21), (3.22), and (3.23), it is easy to show that this family is a polynomial algebra. The only point that requires some detail is the fact that the k-th power of the polynomial p in (3.23) that is obtained in the product of the two SAS reservoir functionals $H_{\mathbf{W}_1}^{p_1,q_1}$ and $H_{\mathbf{W}_2}^{p_2,q_2}$ is given by

$$p^k(z):=\begin{pmatrix}p_1^k(z) & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & p_2^k(z) & \mathbf{0} \\ p_1^k\otimes q_2^{k-1}(z) & q_1^{k-1}\otimes p_2^k(z) & p_1^k\otimes p_2^k(z)\end{pmatrix},$$

which shows that if p_1 and p_2 are nilpotent then so is the associated polynomial p. The point separation property is, again, inherited from the proof of linear case provided in the Appendix 6.5. ■

6.11 Proof of Lemma 24

(i) Let $A := \left\{\rho \in \overline{\mathbb{R}_+} \mid \|\mathbf{X}\|_B \leq \rho \quad \text{almost surely}\right\}$. It suffices to show that $\|\mathbf{X}\|_{L^\infty} := \inf A \in A$, which implies that $\|\mathbf{X}\|_B \leq \|\mathbf{X}\|_{L^\infty}$ almost surely. Indeed, consider the sequence $\|\mathbf{X}\|_{L^\infty} + 1/j$, $j \in \mathbb{N}$. By the approximation property of the infimum, there exists a decreasing sequence of numbers $\{\rho_j\}_{j\in\mathbb{N}} \subset A$ in A satisfying $\|\mathbf{X}\|_{L^\infty} \leq \rho_j < \|\mathbf{X}\|_{L^\infty} + 1/j$ for all $j \in \mathbb{N}$. Define $F := \{\omega \in \Omega \mid \|\mathbf{X}(\omega)\|_B > \|\mathbf{X}\|_{L^\infty}\}$ and $F_j := \{\omega \in \Omega \mid \|\mathbf{X}(\omega)\|_B > \rho_j\}$. It is easy to see that $F_j \subset F_{j+1}$, $j \in \mathbb{N}$ and that $\lim_{j\to\infty} F_j = F$ and, consequently, (see Lemma 5, page 7 in Grimmett and Stirzaker (2001)) $\lim_{j\to\infty} \mathbb{P}(F_j) = \mathbb{P}(F)$. Since by construction $\mathbb{P}(F_j) = 0$ for all $j \in \mathbb{N}$ then $\mathbb{P}(F) = 0$ necessarily, which shows that $\|\mathbf{X}\|_{L^\infty} \in A$, as required.
(ii) If $\|\mathbf{X}\|_{L^\infty} \leq C$ then by part **(i)**, $\|\mathbf{X}\|_B \leq \|\mathbf{X}\|_{L^\infty} \leq C$ almost surely. Conversely, if $\|\mathbf{X}\|_B \leq C$ almost surely, then $C \in A = \left\{\rho \in \overline{\mathbb{R}^+} \mid \|\mathbf{X}\|_B \leq \rho \quad \text{almost surely}\right\}$. Consequently, $\|\mathbf{X}\|_{L^\infty} = \inf A \leq C \in A$, as required.
(iii) Suppose first that $\|\mathbf{X}\|_B \leq C$ almost surely and define $F := \{\omega \in \Omega \mid \|\mathbf{X}(\omega)\|_B > C\}$. By hypothesis, we have that $\mathbb{P}(F) = 0$ and $\mathbb{P}(\Omega \setminus F) = 1$. Then,

$$\begin{aligned}\mathrm{E}\left[\|\mathbf{X}\|_B^k\right] &= \int_\Omega \|\mathbf{X}\|_B^k \, d\mathbb{P} = \int_{\Omega\setminus F} \|\mathbf{X}\|_B^k \, d\mathbb{P} + \int_F \|\mathbf{X}\|_B^k \, d\mathbb{P} \\ &= \int_{\Omega\setminus F} \|\mathbf{X}\|_B^k \, d\mathbb{P} \leq \int_{\Omega\setminus F} C^k d\mathbb{P} = C^k \mathbb{P}(\Omega \setminus F) = C^k,\end{aligned}$$

as required. Conversely, assume that $\mathrm{E}\left[\|\mathbf{X}\|_B^k\right] \leq C^k$, for any $k \in \mathbb{N}$, and define

$$F_n := \left\{\omega \in \Omega \mid \|\mathbf{X}(\omega)\|_B > C + \frac{1}{n}\right\},$$

for all $n \geq 1$. It is easy to see that $F_n \subset F_{n+1}$ and that $\lim_{n\to\infty} F_n = F$ and, consequently, (see Lemma 5, page 7 in Grimmett and Stirzaker (2001)) $\lim_{n\to\infty} \mathbb{P}(F_n) = \mathbb{P}(F)$. Now,

$$\begin{aligned}C^k &\geq \mathrm{E}\left[\|\mathbf{X}\|_B^k\right] = \int_\Omega \|\mathbf{X}\|_B^k \, d\mathbb{P} = \int_{\Omega\setminus F_n} \|\mathbf{X}\|_B^k \, d\mathbb{P} + \int_{F_n} \|\mathbf{X}\|_B^k \, d\mathbb{P} \\ &\geq \int_{F_n} \|\mathbf{X}\|_B^k \, d\mathbb{P} \geq \int_{F_n} \left(C + \frac{1}{n}\right)^k d\mathbb{P} = \left(C + \frac{1}{n}\right)^k \mathbb{P}(F_n),\end{aligned}$$

which implies that $\mathbb{P}(F_n) \leq C^k / \left(C + \frac{1}{n}\right)^k$ for any $k \in \mathbb{N}$ and hence, by taking the limit $k \to \infty$, we can conclude that $\mathbb{P}(F_n) = 0$. Consequently, $\mathbb{P}(F) = \lim_{n\to\infty} \mathbb{P}(F_n) = 0$, which shows that $\|\mathbf{X}\|_B \leq C$ almost surely.
(iv) Let $||\cdot||$ denote the Euclidean norm on $\mathbb{R}^n$. Since $|X_i| \leq \|\mathbf{X}\|$ always and by part **(i)** $\|\mathbf{X}\| \leq \|\mathbf{X}\|_{L^\infty}$ almost surely, we can conclude that $|X_i| \leq \|\mathbf{X}\|_{L^\infty}$ almost surely. This implies that $X_i \in L^\infty(\Omega, \mathbb{R})$ and hence the statement follows from part **(iii)**. ■

6.12 Proof of Lemma 25

We start by proving by contradiction that

$$\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\} \geq \sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{6.20}$$

Indeed, suppose that

$$\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\} < \sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{6.21}$$

By the approximation property of the supremum (see Theorem 1.14 in Apostol (1974)), there exists $t_0 \in \mathbb{Z}$ such that

$$\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\} < \operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_{t_0}(\omega)\|\} \leq \sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{6.22}$$

However, $\|\mathbf{z}_{t_0}(\omega)\| \leq \sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}$ for all $\omega \in \Omega$ and hence by part **(i)** in Lemma 24

$$\|\mathbf{z}_{t_0}(\omega)\| \leq \sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\} \leq \operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\}, \quad \text{almost surely.}$$

Now, by part **(ii)** in Lemma 24, this implies that

$$\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_{t_0}(\omega)\|\} \leq \operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\}.$$

However, this expression is in contradiction with the first inequality in (6.22) and hence the assumption (6.21) cannot be correct. This argument implies that the inequality (6.20) holds.

We now prove the reverse inequality, that is,

$$\operatorname*{ess\,sup}_{\omega\in\Omega}\left\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\right\} \leq \sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}. \tag{6.23}$$

By part **(ii)** of Lemma 24, this inequality holds if and only if

$$\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\} \leq \sup_{t\in\mathbb{Z}}\left\{\operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}\right\}, \quad \text{almost surely.} \tag{6.24}$$

Now, by part **(i)** in Lemma 24, we have that $\|\mathbf{z}_t(\omega)\| \leq \operatorname{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}$, almost surely and for each fixed $t \in \mathbb{Z}$. Let $A_t \subset \Omega$ be the zero-measure set such that $\|\mathbf{z}_t(\omega)\| > \operatorname{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}$ for all $\omega \in A_t$. Let $A := \bigcup_{t\in\mathbb{Z}} A_t$. Notice that $\mathbb{P}(A) = \mathbb{P}\left(\bigcup_{t\in\mathbb{Z}} A_t\right) \leq \sum_{t\in\mathbb{Z}} \mathbb{P}(A_t) = 0$ and hence $B := A^{\mathsf{c}}$ has measure one and

$$\|\mathbf{z}_t(\omega)\| \leq \operatorname*{ess\,sup}_{\omega\in\Omega}\{\|\mathbf{z}_t(\omega)\|\}, \quad \text{for all } \omega \in B \text{ and all } t \in \mathbb{Z}.$$

Since B has measure one, this inequality is equivalent to (6.24), which guarantees that (6.23) holds. The inequalities (6.20) and (6.23) that we just proved imply that the equality (4.7) holds true. ■

6.13 Proof of Lemma 26

It is obvious that $S_{\ell^\infty(\mathbb{R}^n)} \subset S_{(\mathbb{R}^n)^\mathbb{Z}}$ and hence the inclusion map

$$\iota : S_{\ell^\infty(\mathbb{R}^n)} \hookrightarrow S_{(\mathbb{R}^n)^\mathbb{Z}}, \tag{6.25}$$

is well-defined. The equivariance with respect to the equivalence relations $\sim_{\ell^\infty(\mathbb{R}^n)}$ and $\sim_{(\mathbb{R}^n)^\mathbb{Z}}$ follows trivially from noticing that if $\mathbf{z}_1, \mathbf{z}_2 \in S_{\ell^\infty(\mathbb{R}^n)}$ are such that $\mathbf{z}_1 \sim_{\ell^\infty(\mathbb{R}^n)} \mathbf{z}_2$ one obviously have that $\iota(\mathbf{z})_1 \sim_{(\mathbb{R}^n)^\mathbb{Z}} \iota(\mathbf{z}_2)$. This shows the existence of the projected map ϕ that makes the diagram

$$\begin{array}{ccc} S_{\ell^\infty(\mathbb{R}^n)} & \xhookrightarrow{\ \iota\ } & S_{(\mathbb{R}^n)^\mathbb{Z}} \\ {\scriptstyle \Pi_{\sim_{\ell^\infty(\mathbb{R}^n)}}}\downarrow & & \downarrow{\scriptstyle \Pi_{\sim_{(\mathbb{R}^n)^\mathbb{Z}}}} \\ L^\infty\left(\Omega, \ell^\infty(\mathbb{R}^n)\right) & \xrightarrow{\ \phi\ } & L^\infty\left(\Omega, (\mathbb{R}^n)^\mathbb{Z}\right), \end{array}$$

commutative where $\Pi_{\sim_{\ell^\infty(\mathbb{R}^n)}}$ and $\Pi_{\sim_{(\mathbb{R}^n)^\mathbb{Z}}}$ map the elements in $S_{\ell^\infty(\mathbb{R}^n)}$ and $S_{(\mathbb{R}^n)^\mathbb{Z}}$ onto their corresponding equivalence classes with respect to the associated equivalence relations. One can easily prove that the norm preservation following the diagram. It is a straightforward exercise to verify that ϕ is injective and preserves the norm $\|\cdot\|_{L^\infty}$. In order to show that ϕ is surjective, let $\mathbf{z} \in L^\infty\left(\Omega, (\mathbb{R}^n)^\mathbb{Z}\right)$. Given that $\|\mathbf{z}\|_{L^\infty} < \infty$ or, equivalently, $\operatorname{ess\,sup}_{\omega\in\Omega}\{\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\}\} < \infty$, by part **(i)** in Lemma 24, this implies that

$$\sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\} < \infty, \quad \text{almost surely.} \tag{6.26}$$

Since the elements in the spaces in $L^\infty\left(\Omega, \ell^\infty(\mathbb{R}^n)\right)$ and $L^\infty\left(\Omega, (\mathbb{R}^n)^\mathbb{Z}\right)$ are equivalence classes containing almost surely equal random variables, we can take another representative $\mathbf{z}^* : \Omega \longrightarrow (\mathbb{R}^n)^\mathbb{Z}$ for the class containing $\mathbf{z} \in L^\infty\left(\Omega, (\mathbb{R}^n)^\mathbb{Z}\right)$ defined as

$$\mathbf{z}^*(\omega) := \begin{cases} \mathbf{z}(\omega), & \text{when} \quad \sup_{t\in\mathbb{Z}}\{\|\mathbf{z}_t(\omega)\|\} < \infty, \\ 0, & \text{otherwise.} \end{cases}$$

Since the processes $\mathbf{z}$ and $\mathbf{z}^*$ differ by (6.26) only in a set of zero measure, they are equal in $L^\infty\left(\Omega, (\mathbb{R}^n)^\mathbb{Z}\right)$ but, this time, $\mathbf{z}^* \in L^\infty\left(\Omega, \ell^\infty(\mathbb{R}^n)\right)$ and $\phi(\mathbf{z}^*) = \mathbf{z}$, as required. ■

6.14 Proof of Theorem 27

Proof of part (i). All along this proof we will denote the elements in K_M with a lower bold case ($\mathbf{z} \in K_M$) and those in $K_M^{L^\infty}$ with an upper bold case ($\mathbf{Z} \in K_M^{L^\infty}$).

We first assume that the functional $H : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R}$ has the fading memory property. This means that H is a continuous map and since by Lemma 2 the space $(K_M, \|\cdot\|_w)$ is compact, then so is the image $H(K_M)$ as a subset of the real line. This implies that there exists a finite real number $L > 0$ such that $H(K_M) \subset [-L, L]$. Let now $\mathbf{Z} \in K_M^{L^\infty}$; the condition $\|\mathbf{Z}\|_{L^\infty} \leq M$ is equivalent to $\|\mathbf{Z}_t\| \leq M$, for all $t \in \mathbb{Z}_-$, almost surely, and hence implies that $H(\mathbf{Z}) \in [-L, L]$, almost surely or, equivalently, that $\|H(\mathbf{Z})\|_{L^\infty} \leq L$. This, in turn, implies that $H(\mathbf{Z}) \in L^\infty(\Omega, \mathbb{R})$ for any $\mathbf{Z} \in K_M^{L^\infty}$, as required.

We now show that $H : (K_M^{L^\infty}, \|\cdot\|_{L^\infty_w}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ has the FMP. The FMP hypothesis on $H : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R}$ implies that for any $\mathbf{z} \in K_M$ and any $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that for any $\mathbf{s} \in K_M$ that satisfies that

$$\|\mathbf{z} - \mathbf{s}\|_w = \sup_{t\in\mathbb{Z}_-}\{\|(\mathbf{z}_t - \mathbf{s}_t)w_{-t}\|\} < \delta(\epsilon), \quad \text{then} \quad |H(\mathbf{z}) - H(\mathbf{s})| < \epsilon. \tag{6.27}$$

Moreover, since by Lemma 2 the space $(K_M, \|\cdot\|_w)$ is compact, the Uniform Continuity Theorem (Theorem 7.3 in Munkres (2014)) guarantees that the relation $\delta(\epsilon)$ does not depend on the point $\mathbf{z} \in K_M$.

We now prove the statement by showing that for any $\epsilon > 0$ and $\mathbf{Z} \in K_M^{L^\infty}$ then $\|H(\mathbf{Z}) - H(\mathbf{S})\|_{L^\infty} < \epsilon$, for all $\mathbf{S} \in K_M^{L^\infty}$ such that $\|\mathbf{Z} - \mathbf{S}\|_{L^\infty_w} < \delta(\epsilon)$. Indeed, the inequality $\|\mathbf{Z} - \mathbf{S}\|_{L^\infty_w} < \delta(\epsilon)$ holds if and only if $\sup_{t\in\mathbb{Z}_-}\{\|\mathbf{Z}_t - \mathbf{S}_t\|_{L^\infty} w_{-t}\} < \delta(\epsilon)$. Given that for any $l \in \mathbb{Z}_-$ we have that $\|\mathbf{Z}_l - \mathbf{S}_l\|_{L^\infty} w_{-l} \leq \sup_{t\in\mathbb{Z}_-}\{\|\mathbf{Z}_t - \mathbf{S}_t\|_{L^\infty} w_{-t}\} < \delta(\epsilon)$, part **(ii)** in Lemma 24 implies that $\|\mathbf{Z}_l - \mathbf{S}_l\| w_{-l} < \delta(\epsilon)$ almost surely for any $l \in \mathbb{Z}_-$ and hence $\sup_{t\in\mathbb{Z}_-}\{\|\mathbf{Z}_t - \mathbf{S}_t\| w_{-t}\} = \|\mathbf{Z} - \mathbf{S}\|_w < \delta(\epsilon)$, almost surely. This implies, using (6.27), that $|H(\mathbf{Z}) - H(\mathbf{S})| < \epsilon$, almost surely, which by part **(ii)** in Lemma 24 implies that $\|H(\mathbf{Z}) - H(\mathbf{S})\|_{L^\infty} < \epsilon$, as required.

Conversely, if $H : (K_M^{L^\infty}, \|\cdot\|_{L^\infty_w}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ has the fading memory property then so does $H : (K_M, \|\cdot\|_w) \longrightarrow \mathbb{R}$ because $K_M \subset K_M^{L^\infty}$ and $\|\mathbf{z}\| = \|\mathbf{z}\|_{L^\infty}$ for the elements $\mathbf{z} \in K_M$.

Proof of part (ii). We suppose first that $\mathcal{T}$ is dense in the set $(C^0(K_M), \|\cdot\|_w)$ and show that the corresponding family with intputs in $K_M^{L^\infty}$ is universal. Let $H : (K_M^{L^\infty}, \|\cdot\|_{L^\infty_w}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ be an arbitrary causal and time-invariant FMP filter and let $H_S \in \mathcal{T}$ be such that $\sup_{\mathbf{z}\in K_M}\{\|H(\mathbf{z}) - H_S(\mathbf{z})\|_{L^\infty}\} < \epsilon$. The existence of H_S is ensured by the density hypothesis on $\mathcal{T}$. We show that

this ensures that $\sup_{\mathbf{Z}\in K_M^{L^\infty}}\{\|H(\mathbf{Z}) - H_S(\mathbf{Z})\|_{L^\infty}\} < \epsilon$. Indeed, this conclusion is true if $\|H(\mathbf{Z}) - H_S(\mathbf{Z})\|_{L^\infty} < \epsilon$ for any $\mathbf{Z} \in K_M^{L^\infty}$ which, by part **(ii)** in Lemma 24 is equivalent to $|H(\mathbf{Z}) - H_S(\mathbf{Z})| < \epsilon$ almost surely, for any $\mathbf{Z} \in K_M^{L^\infty}$. This condition is in turn true because as $\mathbf{Z} \in K_M^{L^\infty}$, then $\|\mathbf{Z}_t\| \leq M$ almost surely for all $t \in \mathbb{Z}_-$ and hence $\mathbf{Z} \in K_M$ almost surely. Since H_S approximates H for deterministic inputs, we have that $|H(\mathbf{Z}) - H_S(\mathbf{Z})| < \epsilon$ almost surely, as required.

Conversely, if the family $\mathcal{T}$ with intputs in $K_M^{L^\infty}$ is universal in the set of continuous maps of the type $H : (K_M^{L^\infty}, \|\cdot\|_{L^\infty_w}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ we can easily show that $\mathcal{T}$ is dense in $(C^0(K_M), \|\cdot\|_w)$. Let $H \in (C^0(K_M), \|\cdot\|_w)$ and let $H_S : (K_M^{L^\infty}, \|\cdot\|_{L^\infty_w}) \longrightarrow L^\infty(\Omega, \mathbb{R})$ be the element that, for a given $\epsilon > 0$, satisfies $\|H - H_S\|_{L^\infty} = \sup_{\mathbf{Z}\in K_M^{L^\infty}}\{\|H(\mathbf{Z}) - H_S(\mathbf{Z})\|_{L^\infty}\} < \epsilon$. Given that, as we pointed out, $K_M \subset K_M^{L^\infty}$ and $\|\mathbf{z}\| = \|\mathbf{z}\|_{L^\infty}$, for the elements $\mathbf{z} \in K_M$, we have

$$\|H - H_S\| = \sup_{\mathbf{z}\in K_M}\{\|H(\mathbf{z}) - H_S(\mathbf{z})\|\} = \sup_{\mathbf{z}\in K_M}\{\|H(\mathbf{z}) - H_S(\mathbf{z})\|_{L^\infty}\} \leq \sup_{\mathbf{Z}\in K_M^{L^\infty}}\{\|H(\mathbf{Z}) - H_S(\mathbf{Z})\|_{L^\infty}\} < \epsilon. \quad ■$$

6.15 Proof of Lemma 28

As we pointed out in Section 2, if the reservoir system determined by $F : D_N \times \overline{B_n(\mathbf{0}, M)} \longrightarrow D_N$ and $h : D_N \to \mathbb{R}$ has the echo state property, a result in Grigoryeva and Ortega (2018) guarantees that the associated filter is automatically causal and time-invariant. This implies the existence of a functional $H_h^F : (\mathbb{R}^n)^{\mathbb{Z}_-} \longrightarrow \mathbb{R}$ that, by hypothesis, has the fading memory property. The rest of the statement is a consequence of part **(i)** in Theorem 27. ■

6.16 Proof of Theorem 29

We first notice that the polynomial algebra $\mathcal{A}(\mathcal{R})$ is, by Theorem 8 and the first part of Theorem 27, made of fading memory reservoir filters that map into $L^\infty(\Omega, \mathbb{R})$. Using the other hypotheses in the statement we can easily conclude that the family $\mathcal{A}(\mathcal{R})$ satisfies the thesis of Theorem 8 and it is hence universal in the deterministic setup. The result follows from the second part of Theorem 27. ■

Acknowledgments: We thank Philipp Harms and Herbert Jaeger for carefully looking at early versions of this work and for making suggestions that have significantly improved some of our results. We thank Josef Teichmann for fruitful discussions. We also thank the editor and two remarkable anonymous referees whose input has significantly improved the presentation and the contents of the paper. The authors acknowledge partial financial support of the French ANR "BIPHOPROC" project (ANR-14-OHRI-0002-02) as well as the hospitality of the Centre Interfacultaire Bernoulli of the Ecole Polytechnique Fédérale de Lausanne during the program "Stochastic Dynamical Models in Mathematical Finance, Econometrics, and Actuarial Sciences" that made possible the collaboration that led to some of the results included in this paper. LG acknowledges partial financial support of the Graduate School of Decision Sciences and the Young Scholar Fund AFF of the Universität Konstanz. JPO acknowledges partial financial support coming from the Research Commission of the Universität Sankt Gallen and the Swiss National Science Foundation (grant number 200021_175801/1).

Glossary of Symbols

$\ell^\infty_w(\mathbb{R}^n)$	Banach space of semi-infinite sequences with finite weighted norm
$\mathbb{D}$	Space of diagonal matrices of any order
$\mathbb{D}_n$	Space of diagonal matrices of order $n \in \mathbb{N}$
$\mathbb{M}_n$	Space of square matrices of order $n \in \mathbb{N}$
$\mathbb{M}_{m,n}[z]$	$\mathbb{M}_{m,n}$-valued polynomials on z with coefficients in $\mathbb{M}_{m,n}$

$\mathbb{M}_{n,m}$	Space of real $n \times m$ matrices with $m, n \in \mathbb{N}$
$\mathbb{N}\mathrm{il}$	Space of nilpotent matrices of any order and any index
$\mathbb{N}\mathrm{il}[z]$	Space of matrix-valued nilpotent polynomials on z of any order and any index
$\mathbb{N}\mathrm{il}_n^k$	Space of nilpotent matrices of index $k \in \mathbb{N}$ in $\mathbb{M}_n$
$\mathbb{N}\mathrm{il}_n^k[z]$	Space of nilpotent $\mathbb{M}_n$-valued polynomials on z with coefficients in $\mathbb{M}_n$ of index k
$\mathcal{A}(\mathcal{R})$	Polynomial algebra generated by the set $\mathcal{R}$ of reservoir filters defined on K_M
$\mathcal{DL}_\epsilon$	Set of linear reservoir systems determined by diagonal matrices $A \in \mathbb{D}$ such that $\sigma_{\max}(A) < 1 - \epsilon$
$\mathcal{L}_\epsilon$	Set of linear reservoir systems determined by matrices $A \in \mathbb{M}_N$ such that $\sigma_{\max}(A) < 1 - \epsilon$
$\mathcal{NL}$	Set of linear reservoir systems determined by nilpotent matrices $A \in \mathbb{N}\mathrm{il}$
$\mathcal{NS}_\epsilon$	Subfamily of $\mathcal{S}_\epsilon$ formed by SAS reservoir systems determined by nilpotent polynomials p
$\mathcal{R}$	Set of reservoir filters defined on K_M
$\mathcal{S}_\epsilon$	State affine reservoir systems (SAS) $H^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}-} \longrightarrow \mathbb{R}$ with $M_p < 1-\epsilon$ and $M_q < 1-\epsilon$
$B_n(\mathbf{0}, M)$	Ball of radius M and center $\mathbf{0}$ in $\mathbb{R}^n$ with respect to the Euclidean norm
$F : \mathbb{R}^N \times \mathbb{R}^n \longrightarrow \mathbb{R}^N$	Reservoir map
$H_U : (\mathbb{R}^n)^{\mathbb{Z}-} \longrightarrow \mathbb{R}$	Functional associated to the causal and time-invariant filter $U : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$
$h : \mathbb{R}^N \to \mathbb{R}$	Generic readout map
$H_h^{A,\mathbf{c}} : K_M \longrightarrow \mathbb{R}$	Linear reservoir functional determined by $A, \mathbf{c}$, and the polynomial h
$H^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}-} \longrightarrow \mathbb{R}$	SAS reservoir functional
K_M	Space of semi-infinite sequences that are uniformly bounded by M
$K_M^{L^\infty}$	Space of semi-infinite processes that are almost surely uniformly bounded by M
$L^\infty\left(\Omega, (\mathbb{R}^n)^{\mathbb{Z}}\right)$	Space of almost surely bounded time series or discrete-time stochastic processes with values in $\mathbb{R}^n$
$L^\infty_w\left(\Omega, (\mathbb{R}^n)^{\mathbb{Z}-}\right)$	Space of time series or discrete-time stochastic processes with values in $\mathbb{R}^n$ with finite L^∞_w-norm
N	Number of virtual neurons. Dimension of the reservoir state vectors
n	Dimension of the elements of the input signal
$U_h^F : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$	Reservoir filter
$U^F : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow (\mathbb{R}^N)^{\mathbb{Z}}$	Filter determined by the reservoir map F
$U : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$	Filter with inputs in $\mathbb{R}^n$ and outputs in $\mathbb{R}$
$U_h^{A,\mathbf{c}} : K_M \longrightarrow \mathbb{R}^{\mathbb{Z}}$	Linear reservoir filter determined by $A, \mathbf{c}$, and the polynomial h
$U_H : (\mathbb{R}^n)^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$	Causal and time-invariant filter associated to the functional $H : (\mathbb{R}^n)^{\mathbb{Z}-} \longrightarrow \mathbb{R}$
$U^{p,q}_{\mathbf{W}} : I^{\mathbb{Z}} \longrightarrow \mathbb{R}^{\mathbb{Z}}$	SAS reservoir filter
$w : \mathbb{N} \longrightarrow (0, 1]$	Weighting sequence
$\mathbf{x}$	(Semi)-infinite sequence containing the reservoir states. The elements of this sequence are denoted by $\mathbf{x}_t \in \mathbb{R}^N$
$\mathbf{y}$	(Semi)-infinite output signal. The elements of this sequence are denoted by $y_t \in \mathbb{R}$
$\mathbf{z}$	(Semi)-infinite input signal. The elements of this sequence are denoted by $\mathbf{z}_t \in \mathbb{R}^n$

References

R. Abraham, J. E. Marsden, and T. S. Ratiu. *Manifolds, Tensor Analysis, and Applications*, volume 75. Applied Mathematical Sciences. Springer-Verlag, 1988.

T. Apostol. *Mathematical Analysis*. Addison Wesley, second edition, 1974.

L. Appeltant, M. C. Soriano, G. Van der Sande, J. Danckaert, S. Massar, J. Dambre, B. Schrauwen, C. R. Mirasso, and I. Fischer. Information processing using a single dynamical node as complex system. *Nature Communications*, 2:468, jan 2011.

V. I. Arnold. On functions of three variables. *Proceedings of the USSR Academy of Sciences*, 114: 679–681, 1957.

A. F. Atiya and A. G. Parlos. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Transactions on Neural Networks*, 11(3):697–709, jan 2000.

Bai Zhang, D. J. Miller, and Yue Wang. Nonlinear system modeling with random matrices: echo state networks revisited. *IEEE Transactions on Neural Networks and Learning Systems*, 23(1):175–182, jan 2012.

A. Barron. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, 39(3):930–945, may 1993.

T. Bollerslev. Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31 (3):307–327, 1986.

G. E. P. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, 1976.

S. Boyd and L. Chua. Fading memory and the problem of approximating nonlinear operators with Volterra series. *IEEE Transactions on Circuits and Systems*, 32(11):1150–1161, nov 1985.

M. B. Brilliant. Theory of the analysis of nonlinear systems. Technical report, Massachusetts Institute of Technology, Research Laboratory of Electronics, 1958.

P. J. Brockwell and R. A. Davis. *Time Series: Theory and Methods*. Springer-Verlag, 2006.

J. W. Brown and R. V. Churchill. *Complex Variables and Applications Eighth Edition*. McGraw-Hill, eighth edition, 2009.

D. Brunner, M. C. Soriano, C. R. Mirasso, and I. Fischer. Parallel photonic information processing at gigabyte per second data rates using transient states. *Nature Communications*, 4(1364), 2013.

M. Buehner and P. Young. A tighter bound for the echo state property. *IEEE Transactions on Neural Networks*, 17(3):820–824, 2006.

F. Comets and T. Meyre. *Calcul Stochastique et Modèles de Diffusions*. Dunod, Paris, 2006.

R. Couillet, G. Wainrib, H. Sevi, and H. T. Ali. The asymptotic performance of linear echo state neural networks. *Journal of Machine Learning Research*, 17(178):1–35, 2016.

N. Crook. Nonlinear transient computation. *Neurocomputing*, 70:1167–1176, 2007.

J. P. Crutchfield, W. L. Ditto, and S. Sinha. Introduction to focus issue: intrinsic and designed computation: information processing in dynamical systems-beyond the digital hegemony. *Chaos*, 20(3): 037101, sep 2010.

G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, dec 1989.

J. Dambre, D. Verstraeten, B. Schrauwen, and S. Massar. Information processing capacity of dynamical systems. *Scientific reports*, 2(514), 2012.

J. Dieudonné. *Foundations of Modern Analysis*. Academic Press, 1969.

K. Doya. Bifurcations in the learning of recurrent neural networks. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 6, pages 2777–2780. IEEE, 1992.

R. F. Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4):987–1007, 1982.

M. Fliess. Un outil algebrique : les series formelles non commutatives. In G. Marchesini and S. K. Mitter, editors, *Mathematical Systems Theory*, pages 122–148. Springer Verlag, 1976.

M. Fliess and D. Normand-Cyrot. Vers une approche algébrique des systèmes non linéaires en temps discret. In A. Bensoussan and J. Lions, editors, *Analysis and Optimization of Systems. Lecture Notes in Control and Information Sciences, vol. 28*. Springer Berlin Heidelberg, 1980.

C. Francq and J.-M. Zakoian. *GARCH Models: Structure, Statistical Inference and Financial Applications*. Wiley, 2010.

M. Fréchet. Sur les fonctionnelles continues. *Annales scientifiques de l'Ecole Normale Supérieure. 3ème série.*, 27:193–216, 1910.

M. N. Galtier, C. Marini, G. Wainrib, and H. Jaeger. Relative entropy minimizing noisy non-linear neural network to approximate stochastic processes. *Neural Networks*, 56:10–21, 2014.

S. Ganguli, D. Huh, and H. Sompolinsky. Memory traces in dynamical systems. *Proceedings of the National Academy of Sciences of the United States of America*, 105(48):18970–5, dec 2008.

D. A. George. Continuous nonlinear systems. Technical report, Massachusetts Institute of Technology, Research Laboratory of Electronics, 1959.

L. Gonon and J.-P. Ortega. Reservoir computing universality with stochastic inputs. *Preprint*, 2018.

A. Graves, A.-R. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649. IEEE, may 2013.

L. Grigoryeva and J.-P. Ortega. Echo state networks are universal. *Preprint*, 2018.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Optimal nonlinear information processing capacity in delay-based reservoir computers. *Scientific Reports*, 5(12858):1–11, 2015.

L. Grigoryeva, J. Henriques, L. Larger, and J.-P. Ortega. Nonlinear memory capacity of parallel time-delay reservoir computers in the processing of multidimensional signals. *Neural Computation*, 28: 1411–1451, 2016a.

L. Grigoryeva, J. Henriques, and J.-P. Ortega. Reservoir computing: information processing of stationary signals. In *Proceedings of the 19th IEEE International Conference on Computational Science and Engineering*, pages 496–503, 2016b.

G. Grimmett and D. Stirzaker. *Probability and Random Processes*. Oxford University Press, 2001.

M. Hermans and B. Schrauwen. Memory in linear recurrent neural networks in continuous time. *Neural networks : the official journal of the International Neural Network Society*, 23(3):341–55, apr 2010.

R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, second edition, 2013.

K. Hornik, M. Stinchcombe, and H. White. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989.

T. W. Hungerford. *Algebra*. Springer New York, 1974.

H. Jaeger. Short term memory in echo state networks. *Fraunhofer Institute for Autonomous Intelligent Systems. Technical Report.*, 152, 2002.

H. Jaeger. The 'echo state' approach to analysing and training recurrent neural networks with an erratum note. Technical report, German National Research Center for Information Technology, 2010.

H. Jaeger and H. Haas. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science*, 304(5667):78–80, 2004.

H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert. Optimization and applications of echo state networks with leaky-integrator neurons. *Neural Networks*, 20(3):335–352, 2007.

L. K. Jones. A simple lemma on greedy approximation in hilbert space and convergence rates for projection pursuit regression and neural network training. *The Annals of Statistics*, 20(1):608–613, 1992.

A. N. Kolmogorov. On the representation of continuous functions of several variables as superpositions of functions of smaller number of variables. *Soviet Math. Dokl*, 108:179–182, 1956.

V. Kurkova and M. Sanguineti. Learning with generalization capability by kernel methods of bounded complexity. *Journal of Complexity*, 21(3):350–367, 2005.

L. Larger, M. C. Soriano, D. Brunner, L. Appeltant, J. M. Gutierrez, L. Pesquera, C. R. Mirasso, and I. Fischer. Photonic information processing beyond Turing: an optoelectronic implementation of reservoir computing. *Optics Express*, 20(3):3241, jan 2012.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces*. Springer-Verlag, 1991.

M. Lukoševičius and H. Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009.

W. Maass. Liquid state machines: motivation, theory, and applications. In S. S. Barry Cooper and A. Sorbi, editors, *Computability In Context: Computation and Logic in the Real World*, chapter 8, pages 275–296. 2011.

W. Maass and E. D. Sontag. Neural Systems as Nonlinear Filters. *Neural Computation*, 12(8):1743–1772, aug 2000.

W. Maass, T. Natschläger, and H. Markram. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14:2531–2560, 2002.

W. Maass, T. Natschläger, and H. Markram. Fading memory and kernel properties of generic cortical microcircuit models. *Journal of Physiology Paris*, 98(4-6 SPEC. ISS.):315–330, 2004.

W. Maass, P. Joshi, and E. D. Sontag. Computational aspects of feedback in neural circuits. *PLoS Computational Biology*, 3(1):e165, 2007.

G. Manjunath and H. Jaeger. Echo state property linked to an input: exploring a fundamental characteristic of recurrent neural networks. *Neural Computation*, 25(3):671–696, 2013.

M. B. Matthews. *On the Uniform Approximation of Nonlinear Discrete-Time Fading-Memory Systems Using Neural Network Models*. PhD thesis, ETH Zürich, 1992.

M. B. Matthews. Approximating nonlinear fading-memory operators using neural network models. *Circuits, Systems, and Signal Processing*, 12(2):279–307, jun 1993.

J. Munkres. *Topology*. Pearson, second edition, 2014.

Y. Paquot, F. Duport, A. Smerieri, J. Dambre, B. Schrauwen, M. Haelterman, and S. Massar. Optoelectronic reservoir computing. *Scientific reports*, 2:287, jan 2012.

R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio. How to construct deep recurrent neural networks. *arXiv*, dec 2013.

J. Pathak, Z. Lu, B. R. Hunt, M. Girvan, and E. Ott. Using machine learning to replicate chaotic attractors and calculate Lyapunov exponents from data. *Chaos*, 27(12), 2017.

J. Pathak, B. Hunt, M. Girvan, Z. Lu, and E. Ott. Model-Free Prediction of Large Spatiotemporally Chaotic Systems from Data: A Reservoir Computing Approach. *Physical Review Letters*, 120(2): 24102, 2018.

P. Perryman and A. Stubberud. Uniform, in-probability approximation of stochastic systems. In *Conference Record of The Thirtieth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 146–150. IEEE Comput. Soc. Press, 1997.

P. C. Perryman. *Approximation Theory for Deterministic and Stochastic Nonlinear Systems*. PhD thesis, University of California, Irvine, 1996.

G. Pisier. Remarques sur un résultat non publié de B. Maurey. *Séminaire d'analyse fonctionnelle École Polytechnique*, pages 1–12, 1981.

G. Pisier. *Martingales in Banach Spaces*. Cambridge University Press, 2016.

A. Rodan and P. Tino. Minimum complexity echo state network. *IEEE Transactions on Neural Networks*, 22(1):131–44, jan 2011.

L. Rüschendorf and W. Thomsen. Closedness of sum spaces and the generalized schrödinger Problem. *Theory of Probability & Its Applications*, 42(3):483–494, jan 1998.

E. Sontag. Realization theory of discrete-time nonlinear systems: Part I-The bounded case. *IEEE Transactions on Circuits and Systems*, 26(5):342–356, may 1979a.

E. D. Sontag. Polynomial Response Maps. In *Lecture Notes Control in Control and Information Sciences. Vol. 13*. Springer Verlag, 1979b.

D. A. Sprecher. A representation theorem for continuous functions of several variables. *Proceedings of the American Mathematical Society*, 16(2):200, apr 1965.

D. A. Sprecher. A numerical implementation of Kolmogorov's superpositions. *Neural Networks*, 9(5): 765–772, 1996.

D. A. Sprecher. A numerical implementation of Kolmogorov's superpositions II. *Neural Networks*, 10 (3):447–457, 1997.

A. Stubberud and P. Perryman. State of system approximation for stochastic systems. In *Proceedings of 13th International Conference on Digital Signal Processing*, volume 2, pages 711–714. IEEE, 1997a.

A. Stubberud and P. Perryman. Current state of system approximation for deterministic and stochastic systems. In *Conference Record of The Thirtieth Asilomar Conference on Signals, Systems and Computers*, volume 1, pages 141–145. IEEE Comput. Soc. Press, 1997b.

H. J. Sussmann. Semigroup representations, bilinear approximations of input-output maps, and generalized inputs. In G. Marchesini and S. K. MItter, editors, *Mathematical Systems Theory*, pages 172–191. Springer Verlag, 1976.

F. Takens. Detecting strange attractors in turbulence. pages 366–381. Springer Berlin Heidelberg, 1981.

K. Vandoorne, J. Dambre, D. Verstraeten, B. Schrauwen, and P. Bienstman. Parallel reservoir computing using optical amplifiers. *IEEE Transactions on Neural Networks*, 22(9):1469–1481, sep 2011.

K. Vandoorne, P. Mechet, T. Van Vaerenbergh, M. Fiers, G. Morthier, D. Verstraeten, B. Schrauwen, J. Dambre, and P. Bienstman. Experimental demonstration of reservoir computing on a silicon photonics chip. *Nature Communications*, 5:78–80, mar 2014.

D. Verstraeten, B. Schrauwen, M. D'Haene, and D. Stroobandt. An experimental unification of reservoir computing methods. *Neural Networks*, 20:391–403, 2007.

Q. Vinckier, F. Duport, A. Smerieri, K. Vandoorne, P. Bienstman, M. Haelterman, and S. Massar. High-performance photonic reservoir computer based on a coherently driven passive cavity. *Optica*, 2(5):438–446, 2015.

G. Wainrib and M. N. Galtier. A local echo state property through the largest Lyapunov exponent. *Neural Networks*, 76:39–45, apr 2016.

O. White, D. Lee, and H. Sompolinsky. Short-Term Memory in Orthogonal Neural Networks. *Physical Review Letters*, 92(14):148102, apr 2004.

N. Wiener. *Nonlinear Problems in Random Theory*. The Technology Press of MIT, 1958.

I. B. Yildiz, H. Jaeger, and S. J. Kiebel. Re-visiting the echo state property. *Neural Networks*, 35:1–9, nov 2012.

G. Zang and P. A. Iglesias. Fading memory and stability. *Journal of the Franklin Institute*, 340(6-7): 489–502, 2004.

W. Zaremba, I. Sutskever, and O. Vinyals. Recurrent neural network regularization. *arXiv*, sep 2014.

Journal of Machine Learning Research 19 (2018) 1-24 Submitted 1/18; Revised 7/18; Published 7/18

Deep Hidden Physics Models: Deep Learning of Nonlinear Partial Differential Equations

Maziar Raissi maziar_raissi@brown.edu
Division of Applied Mathematics
Brown University
Providence, RI, 02912, USA

Editor: Manfred Opper

Abstract

We put forth a deep learning approach for discovering nonlinear partial differential equations from scattered and potentially noisy observations in space and time. Specifically, we approximate the unknown solution as well as the nonlinear dynamics by two deep neural networks. The first network acts as a prior on the unknown solution and essentially enables us to avoid numerical differentiations which are inherently ill-conditioned and unstable. The second network represents the nonlinear dynamics and helps us distill the mechanisms that govern the evolution of a given spatiotemporal data-set. We test the effectiveness of our approach for several benchmark problems spanning a number of scientific domains and demonstrate how the proposed framework can help us accurately learn the underlying dynamics and forecast future states of the system. In particular, we study the Burgers', Korteweg-de Vries (KdV), Kuramoto-Sivashinsky, nonlinear Schrödinger, and Navier-Stokes equations.

Keywords: Systems Identification, Data-driven Scientific Discovery, Physics Informed Machine Learning, Predictive Modeling, Nonlinear Dynamics, Big Data

1. Introduction

Recent advances in machine learning in addition to new data recordings and sensor technologies have the potential to revolutionize our understanding of the physical world in modern application areas such as neuroscience, epidemiology, finance, and dynamic network analysis where first-principles derivations may be intractable (Rudy et al., 2017). In particular, many concepts from statistical learning can be integrated with classical methods in applied mathematics to help us discover sufficiently sophisticated and accurate mathematical models of complex dynamical systems directly from data. This integration of nonlinear dynamics and machine learning opens the door for principled methods for model construction, predictive modeling, nonlinear control, and reinforcement learning strategies. The literature on data-driven discovery of dynamical systems (Crutchfield and McNamara, 1987) is vast and encompasses equation-free modeling (Kevrekidis et al., 2003), artificial neural networks (Raissi et al., 2018a; Gonzalez-Garcia et al., 1998; Anderson et al., 1996; Rico-Martinez et al., 1992), nonlinear regression (Voss et al., 1999), empirical dynamic modeling (Sugihara et al., 2012; Ye et al., 2015), modeling emergent behavior (Roberts, 2014), automated inference of dynamics (Schmidt et al., 2011; Daniels and Nemenman, 2015a,b), normal form identification in climate (Majda et al., 2009), nonlinear Laplacian spectral analysis (Giannakis and Majda, 2012), modeling emergent behavior (Roberts, 2014), Koopman analysis (Mezić, 2005; Budišić et al., 2012; Mezić, 2013; Brunton et al., 2017), automated inference of dynamics (Schmidt et al., 2011; Daniels and Nemenman, 2015a,b), and symbolic regression (Bongard and Lipson, 2007; Schmidt and Lipson, 2009). More recently, sparsity (Tibshirani, 1996) has been used to determine the governing dynamical system (Brunton et al., 2016; Mangan et al., 2016; Wang et al., 2011; Schaeffer et al., 2013; Ozoliņš et al., 2013; Mackey et al., 2014; Brunton et al., 2014; Proctor et al., 2014; Bai et al., 2014; Tran and Ward, 2016).

Less well studied is how to discover closed form mathematical models of the physical world expressed by partial differential equations from scattered data collected in space and time. Inspired by recent developments in *physics-informed deep learning* (Raissi et al., 2017c,d), we construct structured nonlinear regression models that can uncover the dynamic dependencies in a given set of spatio-temporal data, and return a closed form model that can be subsequently used to forecast future states. In contrast to recent approaches to systems identification (Brunton et al., 2016; Rudy et al., 2017), here we do not need to have direct access or approximations to temporal or spatial derivatives. Moreover, we are using a richer class of function approximators to represent the nonlinear dynamics and consequently we do not have to commit to a particular family of basis functions. Specifically, we consider nonlinear partial differential equations of the general form

$$u_t = \mathcal{N}(t, x, u, u_x, u_{xx}, \ldots), \tag{1}$$

where $\mathcal{N}$ is a nonlinear function of time t, space x, solution u and its derivatives.[1] Here, the subscripts denote partial differentiation in either time t or space x.[2] Given a set of scattered and potentially noisy observations of the solution u, we are interested in learning the nonlinear function $\mathcal{N}$ and consequently identifying an infinite dimensional dynamical system (i.e., partial differential equation) that governs the evolution of the observed spatio-temporal data.

For instance, let us assume that we would like to discover the Burger's equation (Basdevant et al., 1986) in one space dimension $u_t = -uu_x + 0.1u_{xx}$. Although not pursued in the current work, a viable approach (Rudy et al., 2017) to tackle this problem is to create a dictionary of possible terms and write the following expansion

$$\begin{aligned}\mathcal{N}(t, x, u, u_x, u_{xx}, \ldots) \;=\; & \alpha_{0,0} + \alpha_{1,0}u + \alpha_{2,0}u^2 + \alpha_{3,0}u^3 + \\ & \alpha_{0,1}u_x + \alpha_{1,1}uu_x + \alpha_{2,1}u^2u_x + \alpha_{3,1}u^3u_x + \\ & \alpha_{0,2}u_{xx} + \alpha_{1,2}uu_{xx} + \alpha_{2,2}u^2u_{xx} + \alpha_{3,2}u^3u_{xx} + \\ & \alpha_{0,3}u_{xxx} + \alpha_{1,3}uu_{xxx} + \alpha_{2,3}u^2u_{xxx} + \alpha_{3,3}u^3u_{xxx}.\end{aligned}$$

1. The solution $u = (u_1, \ldots, u_n)$ could be n dimensional in which case u_x denotes the collection of all element-wise first order derivatives $\frac{\partial u_1}{\partial x}, \ldots, \frac{\partial u_n}{\partial x}$. Similarly, u_{xx} includes all element-wise second order derivatives $\frac{\partial^2 u_1}{\partial x^2}, \ldots, \frac{\partial^2 u_n}{\partial x^2}$.
2. The space $x = (x_1, x_2, \ldots, x_m)$ could be a vector of dimension m. In this case, u_x denotes the collection of all first order derivative $u_{x_1}, u_{x_2}, \ldots, u_{x_m}$ and u_{xx} represents the set of all second order derivatives $u_{x_1x_1}, u_{x_1x_2}, \ldots, u_{x_1x_m}, \ldots, u_{x_mx_m}$.

Given the aforementioned large collection of candidate terms for constructing the partial differential equation, one could then use sparse regression techniques (Rudy et al., 2017) to determine the coefficients $\alpha_{i,j}$ and consequently the right-hand-side terms that are contributing to the dynamics. A huge advantage of this approach is the interpretability of the learned equations. However, there are two major drawbacks associated with this method.

First, it relies on numerical differentiation to compute the derivatives involved in equation (1). Derivatives are taken either using finite differences for clean data or with polynomial interpolation in the presence of noise. Numerical approximations of derivatives are inherently ill-conditioned and unstable (Baydin et al., 2015) even in the absence of noise in the data. This is due to the introduction of truncation and round-off errors inflicted by the limited precision of computations and the chosen value of the step size for finite differencing. Thus, this approach requires far more data points than library functions. This need for using a large number of points lies more in the numerical evaluation of derivatives than in supplying sufficient data for the regression.

Second, in applying the algorithm (Rudy et al., 2017) outlined above we assume that the chosen library is sufficiently rich to have a sparse representation of the time dynamics of the dataset. However, when applying this approach to a dataset where the dynamics are in fact unknown it is not unlikely that the basis chosen above is insufficient. Specially, in higher dimensions (i.e., for input x or output u) the required number of terms to include in the library increases exponentially. Moreover, an additional issue with this approach is that it can only estimate parameters appearing as coefficients. For example, this method cannot estimate parameters of a partial differential equation (e.g., the sine-Gordon equation) involving a term like $\sin(\alpha u(x))$ with α being the unknown parameter, even if we include sines and cosines in the dictionary of possible terms.

One could avoid the first drawback concerning numerical differentiation by assigning prior distributions in the forms of Gaussian processes (Raissi et al., 2017b; Raissi and Karniadakis, 2017; Raissi et al., 2017a, 2018b) or neural networks (Raissi et al., 2017c,d) to the unknown solution u. Derivatives of the prior on u can now be evaluated at machine precision using symbolic or automatic differentiation (Baydin et al., 2015). This removes the requirement for having or generating data on derivatives of the solution u. This is enabling as it allows us to work with noisy observations of the solution u, scattered in space and time. Moreover, this approach requires far fewer data points than the method proposed in (Rudy et al., 2017) simply because, as explained above, the need for using a large number of data points was due to the numerical evaluation of derivatives. The choice to place a prior on the unknown solution u is motivated by modern techniques for solving forward and inverse problems involving partial differential equations, where the unknown solution is approximated either by a neural network (Raissi et al., 2017c,d; Raissi, 2018; Raissi et al., 2018a) or a Gaussian process (Raissi et al., 2018b; Raissi and Karniadakis, 2017; Raissi et al., 2017a,b; Raissi, 2017; Perdikaris et al., 2017; Raissi and Karniadakis, 2016).

The second drawback can be addressed in a similar fashion by approximating the nonlinear function $\mathcal{N}$ (see equation 1) with a neural network. Representing the nonlinear function

$\mathcal{N}$ by a deep neural network is the novelty of the current work. Deep neural networks are a richer family of function approximators and consequently we do not have to commit to a particular class of basis functions such as polynomials or sines and cosines. This expressiveness comes at the cost of losing interpretability of the learned dynamics. However, there is nothing hindering the use of a particular class of basis functions in order obtain more interpretable equations (Raissi et al., 2017d).

2. Solution methodology

We proceed by approximating both the solution u and the nonlinear function $\mathcal{N}$ with two deep neural networks[3] and define a *deep hidden physics model* f to be given by

$$f := u_t - \mathcal{N}(t, x, u, u_x, u_{xx}, \ldots). \tag{2}$$

We obtain the derivatives of the neural network u with respect to time t and space x by applying the chain rule for differentiating compositions of functions using automatic differentiation (Baydin et al., 2015). It is worth emphasizing that automatic differentiation is different from, and in several aspects superior to, numerical or symbolic differentiation; two commonly encountered techniques of computing derivatives. Numerical differentiation entails the finite difference approximation of derivatives using values of the original function evaluated at some sample points. Due to the introduction of truncation and round-off errors inflicted by the limited precision of computations and the chosen value of the step size for finite differencing, numerical approximations of derivatives are inherently ill-conditioned and unstable (Baydin et al., 2015). Symbolic differentiation uses expression manipulation in computer algebra systems such as Mathematica, Maxima, and Maple. Symbolic methods require models to be defined as closed-form expressions, ruling out or severely limiting algorithmic control flow and expressivity (Baydin et al., 2015).

Without proper introduction, one might assume that automatic differentiation is either a type of numerical or symbolic differentiation (Baydin et al., 2015). Confusion can arise because automatic differentiation does in fact provide numerical values of derivatives (as opposed to derivative expressions) and it does so by using symbolic rules of differentiation (but keeping track of derivative values as opposed to the resulting expressions), giving it a two-sided nature that is partly symbolic and partly numerical (Baydin et al., 2015). In its most basic description (Baydin et al., 2015), automatic differentiation relies on the fact that all numerical computations are ultimately compositions of a finite set of elementary operations for which derivatives are known. Combining the derivatives of the constituent operations through the chain rule gives the derivative of the overall composition. This allows accurate evaluation of derivatives at machine precision with ideal asymptotic efficiency and only a small constant factor of overhead. In particular, to compute the derivatives involved in equation (2) we rely on Tensorflow (Abadi et al., 2016) which is a popular and relatively well documented open source software library for automatic differentiation and deep learning computations. In TensorFlow, before a model is run, its computational graph is defined

3. Representing the solution u by a deep neural network is inspired by recent developments in *physics-informed deep learning* (Raissi et al., 2017c,d), while approximating the nonlinear function $\mathcal{N}$ by another network is the novelty of this work.

statically rather than dynamically as for instance in PyTorch (Paszke et al., 2017). This is an important feature as it allows us to create and compile the entire computational graph for a *deep hidden physics model* (2) only once and keep it fixed throughout the training procedure. This leads to significant reduction in the computational cost of the proposed framework.

Parameters of the neural networks u and $\mathcal{N}$ can be learned by minimizing the sum of squared errors loss function

$$SSE := \sum_{i=1}^{N} \left(|u(t^i, x^i) - u^i|^2 + |f(t^i, x^i)|^2 \right), \tag{3}$$

where $\{t^i, x^i, u^i\}_{i=1}^N$ denote the training data on u. The term $|u(t^i, x^i) - u^i|^2$ tries to fit the data by adjusting the parameters of the neural network u while the term $|f(t^i, x^i)|^2$ learns the parameters of the network $\mathcal{N}$ by trying to satisfy the partial differential equation (1) at the collocation points (t^i, x^i). Training the parameters of the neural networks u and $\mathcal{N}$ can be performed simultaneously by minimizing the sum of squared error (3) or in a sequential fashion by training u first and $\mathcal{N}$ second.

How can we make sure that the algorithm presented above results in an acceptable function $\mathcal{N}$? One answer would be to solve the learned equations and compare the resulting solution to the solution of the exact partial differential equation. However, it should be pointed out that the learned function $\mathcal{N}$ is a *black-box* function; i.e., we do not know its functional form. Consequently, none of the classical partial differential equation solvers such as finite differences, finite elements or spectral methods are applicable here. Therefore, to solve the learned equations we have no other choice than to resort to modern black-box solvers such as *physic informed neural networks* (PINNs) introduced in (Raissi et al., 2017c). The steps involved in PINNs as solvers[4] are similar to equations (1), (2), and (3) with the nonlinear function $\mathcal{N}$ being known and the data residing on the boundary of the domain.

In the following, to keep the paper self-contained, we briefly explain the PINNs algorithm (Raissi et al., 2017c) for solving non-linear partial differential equations in as few sentences as possible. The PINNs algorithm is very general and for a more detailed exposure we refer the interested reader to (Raissi et al., 2017c). However, for pedagogical purposes, we explain the algorithm by applying it to the problem of solving the Burgers' equation (see section 3.1) as an example accompanied by periodic boundary conditions; i.e.,

$$\begin{aligned} &u_t + uu_x - 0.1u_{xx} = 0, \quad x \in [-8, 8], \\ &u(0, x) = -\sin(\pi x/8), \\ &u(t, -8) = u(t, 8), \\ &u_x(t, -8) = u_x(t, 8). \end{aligned}$$

We approximate the unknown solution $u(t, x)$ to the Burgers' equation by a deep neural network. Consequently, the corresponding *physic informed neural network* (PINN) takes

4. PINNs have also been used in (Raissi et al., 2017d) to solve inverse problems involving nonlinear partial differential equations in cases where the physics of the problem are well understood and the nonlinear function $\mathcal{N}$ is known up to a set of parameters.

the form

$$f := u_t + uu_x - 0.1u_{xx}.$$

To be precise, for the cases studied in the current work, the physic informed neural network $f(t, x)$ has a form similar to $f := \mathcal{N}(u, u_x, u_{xx})$ for some pre-trained (see equation 3) neural network $\mathcal{N}$ with its parameters being kept fixed. We acquire the required derivatives to compute the residual network f by applying the chain rule for differentiating compositions of functions using automatic differentiation (Baydin et al., 2015). The shared parameters between the neural networks $u(t, x)$ and $f(t, x)$ can be learned by minimizing the mean squared errors loss function

$$MSE = MSE_0 + MSE_b + MSE_f,$$

where

$$\begin{aligned} MSE_0 &= \tfrac{1}{N_0} \textstyle\sum_{i=1}^{N_0} |u(0, x_0^i) - u_0^i|^2, \\ MSE_b &= \tfrac{1}{N_b} \textstyle\sum_{i=1}^{N_b} \left(|u(t_b^i, -8) - u(t_b^i, 8)|^2 + |u_x(t_b^i, -8) - u_x(t_b^i, 8)|^2 \right), \\ MSE_f &= \tfrac{1}{N_f} \textstyle\sum_{i=1}^{N_f} |f(t_f^i, x_f^i)|^2. \end{aligned}$$

Here, $\{x_0^i, h_0^i\}_{i=1}^{N_0}$ denote the initial data generated in this example by the initial function $-\sin(\pi x/8)$, $\{t_b^i\}_{i=1}^{N_b}$ correspond to the collocation points on the boundary, and $\{t_f^i, x_f^i\}_{i=1}^{N_f}$ represents the collocation points on the residual network $f(t, x)$. Consequently, MSE_0 corresponds to the loss on the initial data, MSE_b enforces the periodic boundary conditions, and MSE_f penalizes the Burgers' equation for not being satisfied on the collocation points. This example encapsulates all of the important ingredients of the PINNs algorithm (Raissi et al., 2017c) and can be straightforwardly generalized to arbitrary partial differential equations where the boundary conditions should be treated on a case by case basis (see Raissi et al., 2017c).

3. Results

The proposed framework provides a universal treatment of nonlinear partial differential equations of fundamentally different nature. This generality will be demonstrated by applying the algorithm to a wide range of canonical problems spanning a number of scientific domains including the Burgers', Korteweg-de Vries (KdV), Kuramoto-Sivashinsky, nonlinear Schrödinger, and Navier-Stokes equations. These examples are motivated by the pioneering work of Rudy et al. (2017). All data and codes used in this manuscript are publicly available on GitHub at `https://github.com/maziarraissi/DeepHPMs`.

3.1 Burgers' equation

Let us start with the Burgers' equation arising in various areas of engineering and applied mathematics, including fluid mechanics, nonlinear acoustics, gas dynamics, and traffic flow (Basdevant et al., 1986). In one space dimension, the Burgers' equation reads as

$$u_t = -uu_x + 0.1u_{xx}. \tag{4}$$

To obtain a set of training and test data, we simulate the Burger's equation (4) using conventional spectral methods. Specifically, starting from an initial condition $u(0,x) = -\sin(\pi x/8)$, $x \in [-8,8]$ and assuming periodic boundary conditions, we integrate equation (4) up to the final time $t = 10$. We use the Chebfun package (Driscoll et al., 2014) with a spectral Fourier discretization with 256 modes and a fourth-order explicit Runge-Kutta temporal integrator with time-step size 10^{-4}. The solution is saved every $\Delta t = 0.05$ to give us a total of 201 snapshots. Out of this data-set, we generate a smaller training subset, scattered in space and time, by randomly sub-sampling 10000 data points from time $t = 0$ to $t = 6.7$. We call the portion of the domain from time $t = 0$ to $t = 6.7$ the training portion. The rest of the domain from time $t = 6.7$ to the final time $t = 10$ will be referred to as the test portion. Using this terminology, we are in fact sub-sampling from the original dataset only in the training portion of the domain. Given the training data, we are interested in learning $\mathcal{N}$ as a function of the solution u and its derivatives up to the 2nd order[5]; i.e.,

$$u_t = \mathcal{N}(u, u_x, u_{xx}). \tag{5}$$

We represent the solution u by a 5-layer deep neural network with 50 neurons per hidden layer. Furthermore, we let $\mathcal{N}$ to be a neural network with 2 hidden layers and 100 neurons per hidden layer. As for the activation functions, we use $\sin(x)$. In general, the choice of a neural network's architecture (e.g., number of layers/neurons and form of activation functions) is crucial and in many cases still remains an art that relies on one's ability to balance the trade off between *expressivity* and *trainability* of the neural network (Raghu et al., 2016). Our empirical findings so far indicate that deeper and wider networks are usually more expressive (i.e., they can capture a larger class of functions) but are often more costly to train (i.e., a feed-forward evaluation of the neural network takes more time and the optimizer requires more iterations to converge). Moreover, the sinusoid (i.e., $\sin(x)$) activation function seems to be numerically more stable than $\tanh(x)$, at least while computing the residual neural network f (see equation 2). However, these observations should be interpreted as conjectures rather than as firm results[6]. In this work, we have tried to choose the neural networks' architectures in a consistent fashion throughout the manuscript. Consequently, there might exist other architectures that improve some of the results reported in the current work.

The neural networks u and $\mathcal{N}$ are trained by minimizing the sum of squared errors loss of equation (3). To illustrate the effectiveness of our approach, we solve the learned partial differential equation (5), along with periodic boundary conditions and the same initial condition as the one used to generate the original dataset, using the PINNs algorithm (Raissi et al., 2017c). The original dataset alongside the resulting solution of the learned partial differential equation are depicted in figure 1. This figure indicates that our algorithm is able to accurately identify the underlying partial differential equation with a relative L^2-error of 4.78e-03. It should be highlighted that the training data are collected in roughly two-thirds of the domain between times $t = 0$ and $t = 6.7$. The algorithm is thus extrapolating from

5. A detailed study of the choice of the order will be provided later in this section.
6. We encourage the interested reader to check out the codes corresponding to this paper on GitHub at `https://github.com/maziarraissi/DeepHPMs` and experiment with different choices for the neural networks' architectures.

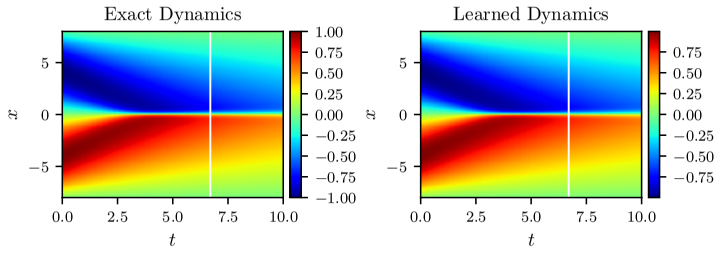


Figure 1: *Burgers equation:* A solution to the Burger's equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the solution with a relative L^2-error of 4.78e-03. It should be emphasized that the training data are collected in roughly two-thirds of the domain between times $t = 0$ and $t = 6.7$ represented by the white vertical lines. The algorithm is thus extrapolating from time $t = 6.7$ onwards. The relative L^2-error on the training portion of the domain is 3.89e-03.

	Clean data	1% noise	2% noise	5% noise
Relative L^2-error	4.78e-03	2.64e-02	1.09e-01	4.56e-01

Table 1: *Burgers' equation:* Relative L^2-error between solutions of the Burgers' equation and the learned partial differential equation as a function of noise corruption level in the data. Here, the total number of training data as well as the neural network architectures are kept fixed.

time $t = 6.7$ onwards. The relative L^2-error on the training portion of the domain is 3.89e-03.

Furthermore, we performed a systematic study of the reported results in figure 1 with respect to noise levels in the data by keeping the total number of training observations as well as the neural network architectures fixed to the settings described above. In particular, we added white noise with magnitude equal to one, two, and five percent of the standard deviation of the solution function. The results of this study are summarized in table 1. The key observation here is that less noise in the data enhances the performance of the algorithm. Our experience so far indicates that the negative consequences of more noise in the data can be remedied to some extent by obtaining more data. Another fundamental point to make is that the choice of the neural network architectures, i.e., activation functions and number of layers/neurons, is of great importance. However, in many cases of practical interest, this choice still remains an art, and systematic studies with respect to the neural network architectures often fail to reveal consistent patterns (Raissi et al., 2017c,d, 2018a). We usually observe some variability and non monotonic trends in systematic studies pertaining to the network architectures. In this regard, there exist a series of open questions

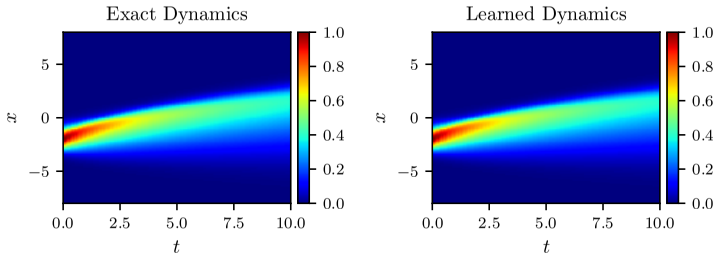


Figure 2: *Burgers equation:* A solution to the Burger's equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the solution with a relative L^2-error of 3.44e-02. It should be highlighted that the algorithm is trained on a dataset totally different from the one used in this figure.

mandating further investigations. For instance, how common techniques such as batch normalization, drop out, and L^1/L^2 regularization (Goodfellow et al., 2016) could enhance the robustness of the proposed algorithm with respect to the neural network architectures as well as noise in the data.

To further scrutinize the performance of the algorithm, let us change the initial condition to $-\exp(-(x+2)^2)$ and solve the Burgers' equation (4) using a classical partial differential equation solver. In particular, the new data-set (Rudy et al., 2017) contains 101 time snapshots of a solution to the Burgers' equation (4) with a Gaussian initial condition, propagating into a traveling wave. The snapshots are $\Delta t = 0.1$ apart and stretch from time $t = 0$ to $t = 10$. The spatial discretization of each snapshot involves a uniform grid with 256 cells. We compare the resulting solution to the one obtained by solving the learned partial differential equation (5) using the PINNs algorithm (Raissi et al., 2017c). It is worth emphasizing that the algorithm is trained on the dataset depicted in figure 1 and is being tested on a totally different dataset as shown in figure 2. The surprising result reported in figure 2 is a strong indication that the algorithm is capable of accurately identifying the underlying partial differential equation. The algorithm has not seen even a single observation of the dataset shown in figure 2 during model training and is yet capable of achieving a relatively accurate approximation of the true solution. The identified system reproduces the solution to the Burgers' equation with a relative L^2-error of 3.44e-02.

However, the aforementioned result seems to be sensitive to making the nonlinear function $\mathcal{N}$ of equation (5) depend on either time t or space x, or both. For instance, if we look for equations of the form $u_t = \mathcal{N}(x, u, u_x, u_{xx})$, the relative L^2-error between the exact and the learned solutions corresponding to figure 2 increases to 4.25e-01. Similarly, if we look for equations of the form $u_t = \mathcal{N}(t, u, u_x, u_{xx})$, the relative L^2-error increases to 2.58e-01.

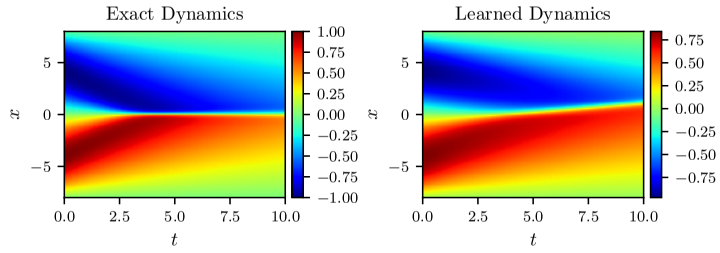


Figure 3: *Burgers equation:* A solution to the Burger's equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The algorithm fails to generalize to the new test data. The relative L^2-error between the exact and the learned solutions is equal to 2.99e-01. Here, we are training on the dataset presented in figure 2 and testing on the one depicted in figure 1. In other words, we are swapping the roles of these two datasets. This observation suggests that the dynamics (see figure 1) generated by $-\sin(\pi x/8)$ as the initial condition is a reacher one compared to the dynamics (see figure 2) generated by $\exp(-(x+2)^2)$. This is also evident from a visual comparison of the two datasets given in figures 1 and 2.

Moreover, if we make the nonlinear function $\mathcal{N}$ both time and space dependent the relative L^2-error increases even further to 1.46e+00. A possible explanation for this behavior could be that some of the dynamics in the training data is now being explained by time t or space x. This makes the algorithm over-fit to the training data and consequently harder for it to generalize to datasets it has never seen before. Based on our experience more data seems to help resolve this issue. Another interesting observation is that if we train on the dataset presented in figure 2 and test on the one depicted in figure 1, i.e., swap the roles of these two datasets, the algorithm fails to generalize to the new test data. To be precise, as depicted in figure 3, the relative L^2-error between the exact and the learned solutions is now equal to 2.99e-01. This observation suggests that the dynamics (see figure 1) generated by $-\sin(\pi x/8)$ as the initial condition is a richer one compared to the dynamics (see figure 2) generated by $\exp(-(x+2)^2)$. This is also evident from a visual comparison of the datasets given in figures 1 and 2.

Let us now take a closer look at equation (5) and ask: what would happen if we included derivatives of higher order than two in our formulation? As will be demonstrated in the following, the algorithm proposed in the current work is capable of handling such cases, however, this is a fundamental question worthy of a moment of reflection. The choice of the order of the partial differential equation (5) determines the form and the number of boundary conditions needed to end up with a well-posed problem. For instance, in the one dimensional setting of equation (5) including a third order derivative requires three boundary conditions, namely $u(t,-8) = u(t,8)$, $u_x(t,-8) = u_x(t,8)$, and $u_{xx}(t,-8) = u_{xx}(t,8)$,

	1st order	2nd order	3rd order	4th order
Relative L^2-error	1.14e+00	1.29e-02	3.42e-02	5.54e-02

Table 2: *Burgers' equation:* Relative L^2-error between solutions of the Burgers' equation and the learned partial differential equation as a function of the highest order of spatial derivatives included in our formulation. For instance, the case corresponding to the 3rd order means that we are looking for a nonlinear function $\mathcal{N}$ such that $u_t = \mathcal{N}(u, u_x, u_{xx}, u_{xxx})$. Here, the total number of training data as well as the neural network architectures are kept fixed and the data are assumed to be noiseless.

assuming period boundary conditions. Consequently, in cases of practical interest, the available information on the boundary of the domain could help us determine the order of the partial differential equation we are try to identify. With this in mind, let us study the robustness of our framework with respect to the highest order of the derivatives included in equation (5). As for the boundary conditions, we use $u(t,-8) = u(t,8)$ and $u_x(t,-8) = u_x(t,8)$ when solving the identified partial differential equation regardless of the initial choice of its order. The results are summarized in table 2. The first column of table 2 demonstrates that a single first order derivative is clearly not enough to capture the second order dynamics of the Burgers' equation. Moreover, the method seems to be generally robust with respect to the number and order of derivatives included in the nonlinear function $\mathcal{N}$. Therefore, in addition to any information residing on the domain boundary, studies such as table 2, albeit for training or validation datasets, could help us choose the best order for the underlying partial differential equation. In this case, table 2 suggests the order of the equation to be two.

In addition, it must be stated that including higher order derivatives comes at the cost of reducing the speed of the algorithm due to the growth in the complexity of the resulting computational graph for the corresponding *deep hidden physics model* (see equation 2). Also, another drawback is that higher order derivatives are usually less accurate specially if one uses single precision floating-point system (float32) rather than double precision (float64). It is true that automatic differentiation enables us to evaluate derivatives at machine precision, however, for float32 the machine epsilon is approximately 1.19e-07. For improved performance in terms of speed of the algorithm and constrained by usual GPU (graphics processing unit) platforms we often end up using float32 which boils down to committing an error of approximately 1.19e-07 while computing the required derivatives. The aforementioned issues do not seem to be too serious since computer infrastructure (both hardware and software) for deep learning is constantly improving.

3.2 The KdV equation

As a mathematical model of waves on shallow water surfaces one could consider the Korteweg-de Vries (KdV) equation. The KdV equation reads as

$$u_t = -uu_x - u_{xxx}. \tag{6}$$

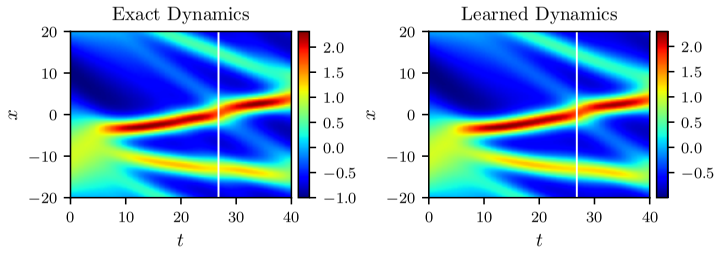


Figure 4: *The KdV equation:* A solution to the KdV equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the solution with a relative L^2-error of 6.28e-02. It should be emphasized that the training data are collected in roughly two-thirds of the domain between times $t = 0$ and $t = 26.8$ represented by the white vertical lines. The algorithm is thus extrapolating from time $t = 26.8$ onwards. The relative L^2-error on the training portion of the domain is 3.78e-02.

To obtain a set of training data we simulate the KdV equation (6) using conventional spectral methods. In particular, we start from an initial condition $u(0,x) = -\sin(\pi x/20)$, $x \in [-20, 20]$ and assume periodic boundary conditions. We integrate equation (6) up to the final time $t = 40$. We use the Chebfun package (Driscoll et al., 2014) with a spectral Fourier discretization with 512 modes and a fourth-order explicit Runge-Kutta temporal integrator with time-step size 10^{-4}. The solution is saved every $\Delta t = 0.2$ to give us a total of 201 snapshots. Out of this data-set, we generate a smaller training subset, scattered in space and time, by randomly sub-sampling 10000 data points from time $t = 0$ to $t = 26.8$. In other words, we are sub-sampling from the original dataset only in the training portion of the domain from time $t = 0$ to $t = 26.8$. Given the training data, we are interested in learning $\mathcal{N}$ as a function of the solution u and its derivatives up to the 3rd order[7]; i.e.,

$$u_t = \mathcal{N}(u, u_x, u_{xx}, u_{xxx}). \tag{7}$$

We represent the solution u by a 5-layer deep neural network with 50 neurons per hidden layer. Furthermore, we let $\mathcal{N}$ to be a neural network with 2 hidden layers and 100 neurons per hidden layer. As for the activation functions, we use $\sin(x)$. These two networks are trained by minimizing the sum of squared errors loss of equation (3). To illustrate the effectiveness of our approach, we solve the learned partial differential equation (7) using the PINNs algorithm (Raissi et al., 2017c). We assume periodic boundary conditions and the same initial condition as the one used to generate the original dataset. The resulting solution of the learned partial differential equation as well as the exact solution of the KdV equation are depicted in figure 4. This figure indicates that our algorithm is capable of accurately identifying the underlying partial differential equation with a relative L^2-error of

7. A detailed study of the choice of the order is provided in section 3.1 for the Burgers' equation.

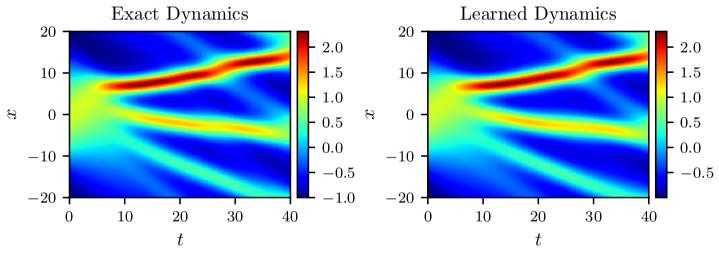


Figure 5: *The KdV equation:* A solution to the KdV equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the solution with a relative L^2-error of 3.44e-02. It should be highlighted that the algorithm is trained on a dataset different from the one shown in this figure.

6.28e-02. It should be highlighted that the training data are collected in roughly two-thirds of the domain between times $t = 0$ and $t = 26.8$. The algorithm is thus extrapolating from time $t = 26.8$ onwards. The corresponding relative L^2-error on the training portion of the domain is 3.78e-02.

To test the algorithm even further, let us change the initial condition to $\cos(-\pi x/20)$ and solve the KdV equation (6) using the conventional spectral method outlined above. We compare the resulting solution to the one obtained by solving the learned partial differential equation (5) using the PINNs algorithm (Raissi et al., 2017c). It is worth emphasizing that the algorithm is trained on the dataset depicted in figure 4 and is being tested on a different dataset as shown in figure 5. The surprising result reported in figure 5 strongly indicates that the algorithm is accurately learning the underlying partial differential equation; i.e., the nonlinear function $\mathcal{N}$. The algorithm hasn't seen the dataset shown in figure 5 during model training and is yet capable of achieving a relatively accurate approximation of the true solution. To be precise, the identified system reproduces the solution to the KdV equation with a relative L^2-error of 3.44e-02.

3.3 Kuramoto-Sivashinsky equation

As a canonical model of a pattern forming system with spatio-temporal chaotic behavior we consider the Kuramoto-Sivashinsky equation. In one space dimension the Kuramoto-Sivashinsky equation reads as

$$u_t = -uu_x - u_{xx} - u_{xxxx}. \tag{8}$$

We generate a dataset containing a direct numerical solution of the Kuramoto-Sivashinsky (8) equation with 512 spatial points and 251 snapshots. To be precise, assuming periodic boundary conditions, we start from the initial condition $u(0, x) = -\sin(\pi x/10)$, $x \in$

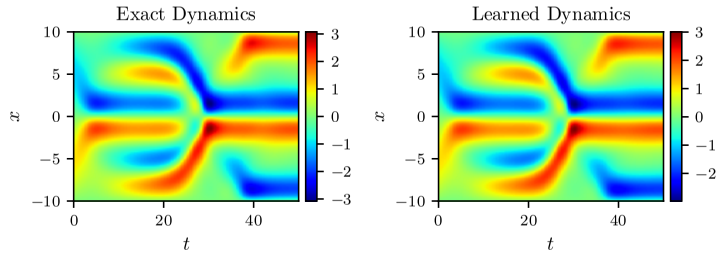


Figure 6: *Kuramoto-Sivashinsky equation:* A solution to the Kuramoto-Sivashinsky equation (left panel) is compared to the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and reproduces the solution with a relative L^2-error of 7.63e-02.

$[-10, 10]$ and integrate equation (8) up to the final time $t = 50$. We employ the Chebfun package (Driscoll et al., 2014) with a spectral Fourier discretization with 512 modes and a fourth-order explicit Runge-Kutta temporal integrator with time-step size 10^{-5}. The snapshots are saved every $\Delta t = 0.2$. From this dataset, we create a smaller training subset, scattered in space and time, by randomly sub-sampling 10000 data points from time $t = 0$ to the final time $t = 50.0$. Given the resulting training data, we are interested in learning $\mathcal{N}$ as a function of the solution u and its derivatives up to the 4rd order[8]; i.e.,

$$u_t = \mathcal{N}(u, u_x, u_{xx}, u_{xxx}, u_{xxxx}). \tag{9}$$

We let the solution u to be represented by a 5-layer deep neural network with 50 neurons per hidden layer. Furthermore, we approximate the nonlinear function $\mathcal{N}$ by a neural network with 2 hidden layers and 100 neurons per hidden layer. As for the activation functions, we use $\sin(x)$. These two networks are trained by minimizing the sum of squared errors loss of equation (3). To demonstrate the effectiveness of our approach, we solve the learned partial differential equation (9) using the PINNs algorithm (Raissi et al., 2017c). We assume the same initial and boundary conditions as the ones used to generate the original dataset. The resulting solution of the learned partial differential equation alongside the exact solution of the Kuramoto-Sivashinsky equation are depicted in figure 6. This figure indicates that our algorithm is capable of identifying the underlying partial differential equation with a relative L^2-error of 7.63e-02. Here and in the rest of the current manuscript, the test and training datasets appear to be the same. However, it should be emphasized that during test time the only pieces of information given to the PINNs algorithm are the initial and boundary data in addition to the learned function $\mathcal{N}$. This means that results such as figure 6 are sufficient to clearly communicate the message that the algorithm has approximately learned the correct equations. Moreover, whether a particular dataset is rich enough so that

8. A detailed study of the choice of the order is provided in section 3.1 for the Burgers' equation.

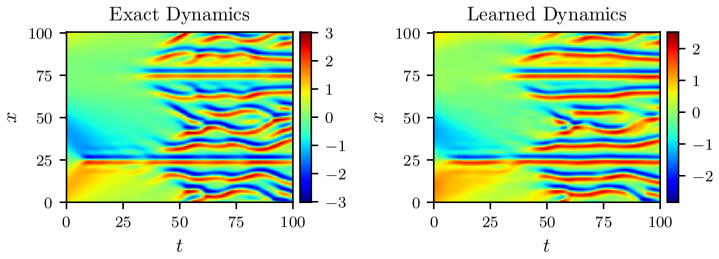


Figure 7: *Kuramoto-Sivashinsky equation:* A solution to the Kuramoto-Sivashinsky equation is depicted in the left panel. The corresponding solution of the learned partial differential equation is shown in the right panel. This problem remains stubbornly unsolved in the face of the algorithm proposed in the current work. In fact, the identified system reproduces the solution with a relative L^2-error of 4.17e-01, which is unsatisfactorily large.

the algorithm generalizes to other datasets generated by different initial conditions is a fundamentally different question which has been extensively studied so far in the manuscript using the Burgers' and KdV examples.

However, it must be mentioned that we are avoiding the regimes where the Kuramoto-Sivashinsky equation becomes chaotic. For instance, by changing the domain to $[0, 32\pi]$ and the initial condition to $\cos(x/16)(1 + \sin(x/16))$ and by integrating the Kuramoto-Sivashinsky equation (8) up to the final time $t = 100$, one could end up with a relatively complicated solution as depicted in the left panel of figure 7. This problem remains stubbornly unsolved in the face of the algorithm proposed in the current work as well as the PINNs framework introduced in (Raissi et al., 2017c,d). In fact, the identified system (see figure 7) reproduces the solution with a relative L^2-error of 4.17e-01, which is unsatisfactorily large. According to our empirical findings, making the neural network for $u(t, x)$ more expressive does not seem to help resolve this issue. As a matter of fact, it is not difficult for a plain vanilla neural network to approximate the function depicted in the left panel of figure 7. However, as we compute the derivatives required in equation (2), minimizing the loss function (3) becomes a challenge. More precisely, the optimizer does not seem to converge to the right values for the parameters of the neural networks (see figure 7). Understanding what makes this problem hard to solve should be at the core of future extensions of this line of research. One possible explanation could be that the resulting optimization problem inherits the complicated nature of the underlying partial differential equation. Moreover, it could indeed be the case that the algorithm proposed in the current work is learning the underlying dynamics correctly while the PINNs algorithm (Raissi et al., 2017c) is failing to solve the learned partial differential equation. The PINNs algorithm (Raissi et al., 2017c) as a solver is admittedly not yet mature but is currently our only option. Devising black-box

solvers for partial differential equations is still in its infancy and more collaborative work is needed to set the foundations in this field.

3.4 Nonlinear Schrödinger equation

The one-dimensional nonlinear Schrödinger equation is a classical field equation that is used to study nonlinear wave propagation in optical fibers and/or waveguides, Bose-Einstein condensates, and plasma waves. This example aims to highlight the ability of our framework to handle complex-valued solutions as well as different types of nonlinearities in the governing partial differential equations. The nonlinear Schrödinger equation is given by

$$\psi_t = 0.5i\psi_{xx} + i|\psi|^2\psi. \tag{10}$$

Let u denote the real part of ψ and v the imaginary part. Then, the nonlinear Schrödinger equation can be equivalently written as a system of partial differential equations

$$\begin{aligned} u_t &= -0.5v_{xx} - (u^2 + v^2)v, \\ v_t &= 0.5u_{xx} + (u^2 + v^2)u. \end{aligned} \tag{11}$$

In order to assess the performance of our method, we simulate equation (10) using conventional spectral methods to create a high-resolution data set. Specifically, starting from an initial state $\psi(0, x) = 2\ \mathrm{sech}(x)$ and assuming periodic boundary conditions $\psi(t, -5) = \psi(t, 5)$ and $\psi_x(t, -5) = \psi_x(t, 5)$, we integrate equation (10) up to a final time $t = \pi/2$ using the Chebfun package (Driscoll et al., 2014). We are in fact using a spectral Fourier discretization with 512 modes and a fourth-order explicit Runge-Kutta temporal integrator with time-step $\Delta t = \pi/2 \cdot 10^{-6}$. The solution is saved approximately every $\Delta t = 0.0031$ to give us a total of 501 snapshots. Out of this data-set, we generate a smaller training subset, scattered in space and time, by randomly sub-sampling 10000 data points from time $t = 0$ up to the final time $t = \pi/2$. Given the resulting training data, we are interested in learning two nonlinear functions $\mathcal{N}_1$ and $\mathcal{N}_2$ as functions of the solutions u, v and their derivatives up to the 2nd order[9]; i.e.,

$$\begin{aligned} u_t &= \mathcal{N}_1(u, v, u_x, v_x, u_{xx}, v_{xx}), \\ v_t &= \mathcal{N}_2(u, v, u_x, v_x, u_{xx}, v_{xx}). \end{aligned} \tag{12}$$

We represent the solutions u and v by two 5-layer deep neural networks with 50 neurons per hidden layer. Furthermore, we let $\mathcal{N}_1$ and $\mathcal{N}_2$ to be two neural networks with 2 hidden layers and 100 neurons per hidden layer. As for the activation functions, we use $\sin(x)$. These four networks are trained by minimizing a sum of squared errors loss function similar to equation (3). To illustrate the effectiveness of our approach, we solve the learned partial differential equation (12), along with periodic boundary conditions and the same initial condition as the one used to generate the original dataset, using the PINNs algorithm (Raissi et al., 2017c). The original dataset (in absolute values, i.e., $|\psi| = \sqrt{u^2 + v^2}$) alongside the resulting solution (also in absolute values) of the learned partial differential equation are depicted in figure 8. This figure indicates that our algorithm is able to accurately identify the underlying partial differential equation with a relative L^2-error of 6.28e-03.

9. A detailed study of the choice of the order is provided in section 3.1 for the Burgers' equation.

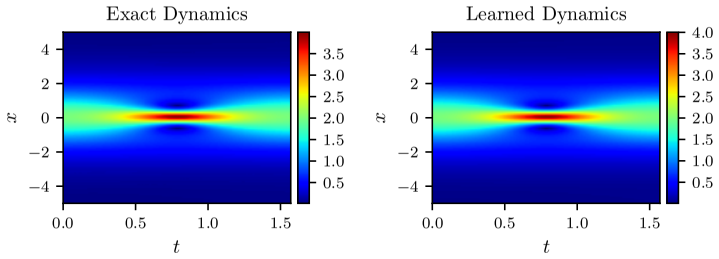


Figure 8: *Nonlinear Schrödinger equation:* Absolute value of a solution to the nonlinear Schrödinger equation (left panel) is compared to the absolute value of the corresponding solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the absolute value of the solution with a relative L^2-error of 6.28e-03.

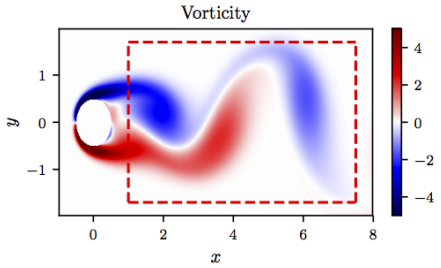


Figure 9: *Navier-Stokes equation:* A snapshot of the vorticity field of a solution to the Navier-Stokes equations for the fluid flow past a cylinder. The dashed red box in this panel specifies the sampling region.

3.5 Navier-Stokes equation

Let us consider the Navier-Stokes equation in two dimensions[10] (2D) given explicitly by

$$w_t = -uw_x - vw_y + 0.01(w_{xx} + w_{yy}), \tag{13}$$

where w denotes the vorticity, u the x-component of the velocity field, and v the y-component. To generate a training dataset for this problem we follow the exact same instructions as the ones provided in (Kutz et al., 2016; Rudy et al., 2017). Specifically,

10. It is straightforward to generalize the proposed framework to the Navier-Stokes equation in three dimensions (3D).

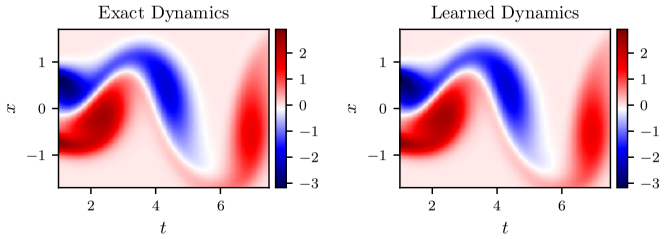


Figure 10: *Navier-Stokes equation:* A randomly picked snapshot of a solution to the Navier-Stokes equation (left panel) is compared to the corresponding snapshot of the solution of the learned partial differential equation (right panel). The identified system correctly captures the form of the dynamics and accurately reproduces the solution with a relative L^2-error of 5.79e-03 in space and time.

we simulate the Navier-Stokes equations describing the two-dimensional fluid flow past a circular cylinder at Reynolds number 100 using the Immersed Boundary Projection Method (Taira and Colonius, 2007; Colonius and Taira, 2008). This approach utilizes a multi-domain scheme with four nested domains, each successive grid being twice as large as the previous one. Length and time are nondimensionalized so that the cylinder has unit diameter and the flow has unit velocity. Data is collected on the finest domain with dimensions 9×4 at a grid resolution of 449×199. The flow solver uses a 3rd-order Runge Kutta integration scheme with a time step of $\Delta t = 0.02$, which has been verified to yield well-resolved and converged flow fields. After simulation converges to steady periodic vortex shedding, 151 flow snapshots are saved every $\Delta t = 0.02$. We use a small portion of the resulting data-set for model training. In particular, we subsample 50000 data points, scattered in space and time, in the rectangular region (dashed red box) downstream of the cylinder as shown in figure 9.

Given the training data, we are interested in learning $\mathcal{N}$ as a function of the stream-wise u and transverse v velocity components in addition to the vorticity w and its derivatives up to the 2nd order[11]; i.e.,

$$w_t = \mathcal{N}(u, v, w, w_x, w_{xx}). \tag{14}$$

We represent the solution w by a 5-layer deep neural network with 200 neurons per hidden layer. Furthermore, we let $\mathcal{N}$ to be a neural network with 2 hidden layers and 100 neurons per hidden layer. As for the activation functions, we use $\sin(x)$. These two networks are trained by minimizing the sum of squared errors loss

$$SSE := \sum_{i=1}^{N} \left(|w(t^i, x^i, y^i) - w^i|^2 + |f(t^i, x^i, y^i)|^2 \right),$$

11. A detailed study of the choice of the order is provided in this section 4 for the Burgers' equation.

where $\{t^i, x^i, u^i, v^i, w^i\}_{i=1}^N$ denote the training data on u and $f(t^i, x^i, y^i)$ is given by

$$f(t^i, x^i, y^i) := w_t(t^i, x^i, y^i) - \mathcal{N}(u^i, v^i, w(t^i, x^i, y^i), w_x(t^i, x^i, y^i), w_{xx}(t^i, x^i, y^i)).$$

To illustrate the effectiveness of our approach, we solve the learned partial differential equation (14), in the region specified in figure 9 by the dashed red box, using the PINNs algorithm (Raissi et al., 2017c). We use the exact solution to provide us with the required Dirichlet boundary conditions as well as the initial condition needed to solve the learned partial differential equation (14). A randomly picked snapshot of the vorticity field in the original dataset alongside the corresponding snapshot of the solution of the learned partial differential equation are depicted in figure 10. This figure indicates that our algorithm is able to accurately identify the underlying partial differential equation with a relative L^2-error of 5.79e-03 in space and time.

4. Summary and Discussion

We have presented a deep learning approach for extracting nonlinear partial differential equations from spatio-temporal datasets. The proposed algorithm leverages recent developments in automatic differentiation to construct efficient algorithms for learning infinite dimensional dynamical systems using deep neural networks. In order to validate the performance of our approach we had no other choice than to rely on black-box solvers (see Raissi et al., 2017c). This signifies the importance of developing general purpose partial differential equation solvers. Developing these types of solvers is still in its infancy and more collaborative work is needed to bring them to the maturity level of conventional methods such as finite elements, finite differences, and spectral methods which have been around for more than half a century or so.

There exist a series of open questions mandating further investigations. For instance, many real-world partial differential equations depend on parameters and, when the parameters are varied, they may go through bifurcations (e.g., the Reynold number for the Navier-Stokes equations). Here, the goal would be to collect data from the underlying dynamics corresponding to various parameter values, and infer the parameterized partial differential equation. Another exciting avenue of future research would be to apply convolutional architectures (Goodfellow et al., 2016) for mitigating the complexity associated with partial differential equations with very high-dimensional inputs. These types of equations appear routinely in dynamic programming, optimal control, or reinforcement learning. We envision that the proposed framework, as outlined in the current work, can be straightforwardly extended to such high-dimensional cases. Early evidence of this claim can be found in (Raissi, 2018), where the author devises an algorithm that is scalable to high-dimensions and is capable of circumventing the tyranny of numerical discretization. In addition, the approach advocated in the current work is also highly scalable to the big data regimes encountered while studying such high-dimensional cases simply because the data can be processed in mini-batches.

Moreover, a quick glance at the list of nonlinear partial differential equations on Wikipedia reveals that many of these equations take the form specified in equation (1). However, a

handful of them do not take this form, including the Boussinesq type equation $u_{tt} - u_{xx} - 2\alpha(uu_x)_x - \beta u_{xxtt} = 0$. It would be interesting to extend the framework outlined in the current work to incorporate all such cases. In addition, it would be an exciting continuation of the current work to extend the proposed methodology to stochastic partial differential equations. In fact, independent realizations of the underlying noise process (e.g., Brownian motion) could act as training data (Raissi, 2018). In the end, it is not always clear what measurements of a dynamical system to take. Even if we did know, collecting these measurements might be prohibitively expensive. It is well-known that time-delay coordinates of a single variable can act as additional variables. It might be interesting to investigate this idea for the infinite dimensional setting of partial differential equations. In terms of applications, it would be intriguing to see how the proposed framework would perform on Geostationary Operational Environmental Satellites (GOES) data (e.g., sea surface temperature data) which could be retrieved from `https://podaac.jpl.nasa.gov/`.

Acknowledgments

This work received support by the DARPA EQUiPS grant N66001-15-2-4055 and the AFOSR grant FA9550-17-1-0013. All data and codes used in this manuscript are publicly available on GitHub at `https://github.com/maziarraissi/DeepHPMs`.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*, 2016.

JS Anderson, IG Kevrekidis, and R Rico-Martinez. A comparison of recurrent training algorithms for time series analysis and system identification. *Computers & chemical engineering*, 20:S751–S756, 1996.

Zhe Bai, Thakshila Wimalajeewa, Zachary Berger, Guannan Wang, Mark Glauser, and Pramod K Varshney. Low-dimensional approach for reconstruction of airfoil data via compressive sensing. *AIAA Journal*, 2014.

Cea Basdevant, M Deville, P Haldenwang, JM Lacroix, J Ouazzani, R Peyret, Paolo Orlandi, and AT Patera. Spectral and finite difference solutions of the Burgers equation. *Computers & fluids*, 14(1):23–41, 1986.

Atilim Gunes Baydin, Barak A Pearlmutter, Alexey Andreyevich Radul, and Jeffrey Mark Siskind. Automatic differentiation in machine learning: a survey. *arXiv preprint arXiv:1502.05767*, 2015.

Josh Bongard and Hod Lipson. Automated reverse engineering of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 104(24):9943–9948, 2007.

Steven L Brunton, Jonathan H Tu, Ido Bright, and J Nathan Kutz. Compressive sensing and low-rank libraries for classification of bifurcation regimes in nonlinear dynamical systems. *SIAM Journal on Applied Dynamical Systems*, 13(4):1716–1732, 2014.

Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Discovering governing equations from data by sparse identification of nonlinear dynamical systems. *Proceedings of the National Academy of Sciences*, 113(15):3932–3937, 2016.

Steven L Brunton, Bingni W Brunton, Joshua L Proctor, Eurika Kaiser, and J Nathan Kutz. Chaos as an intermittently forced linear system. *Nature Communications*, 8, 2017.

Marko Budišić, Ryan Mohr, and Igor Mezić. Applied koopmanism a. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 22(4):047510, 2012.

Tim Colonius and Kunihiko Taira. A fast immersed boundary method using a nullspace approach and multi-domain far-field boundary conditions. *Computer Methods in Applied Mechanics and Engineering*, 197(25):2131–2146, 2008.

James P Crutchfield and Bruce S McNamara. Equations of motion from a data series. *Complex systems*, 1(417-452):121, 1987.

Bryan C Daniels and Ilya Nemenman. Automated adaptive inference of phenomenological dynamical models. *Nature communications*, 6, 2015a.

Bryan C Daniels and Ilya Nemenman. Efficient inference of parsimonious phenomenological models of cellular dynamics using s-systems and alternating regression. *PloS one*, 10(3): e0119821, 2015b.

Tobin A Driscoll, Nicholas Hale, and Lloyd N Trefethen. Chebfun guide, 2014.

Dimitrios Giannakis and Andrew J Majda. Nonlinear laplacian spectral analysis for time series with intermittency and low-frequency variability. *Proceedings of the National Academy of Sciences*, 109(7):2222–2227, 2012.

R Gonzalez-Garcia, R Rico-Martinez, and IG Kevrekidis. Identification of distributed parameter systems: A neural net based approach. *Computers & chemical engineering*, 22: S965–S968, 1998.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Ioannis G Kevrekidis, C William Gear, James M Hyman, Panagiotis G Kevrekidid, Olof Runborg, Constantinos Theodoropoulos, et al. Equation-free, coarse-grained multiscale computation: Enabling mocroscopic simulators to perform system-level analysis. *Communications in Mathematical Sciences*, 1(4):715–762, 2003.

J Nathan Kutz, Steven L Brunton, Bingni W Brunton, and Joshua L Proctor. *Dynamic Mode Decomposition: Data-Driven Modeling of Complex Systems*, volume 149. SIAM, 2016.

Alan Mackey, Hayden Schaeffer, and Stanley Osher. On the compressive spectral method. *Multiscale Modeling & Simulation*, 12(4):1800–1827, 2014.

Andrew J Majda, Christian Franzke, and Daan Crommelin. Normal forms for reduced stochastic climate models. *Proceedings of the National Academy of Sciences*, 106(10): 3649–3653, 2009.

Niall M Mangan, Steven L Brunton, Joshua L Proctor, and J Nathan Kutz. Inferring biological networks by sparse identification of nonlinear dynamics. *IEEE Transactions on Molecular, Biological and Multi-Scale Communications*, 2(1):52–63, 2016.

Igor Mezić. Spectral properties of dynamical systems, model reduction and decompositions. *Nonlinear Dynamics*, 41(1):309–325, 2005.

Igor Mezić. Analysis of fluid flows via spectral properties of the koopman operator. *Annual Review of Fluid Mechanics*, 45:357–378, 2013.

Vidvuds Ozoliņš, Rongjie Lai, Russel Caflisch, and Stanley Osher. Compressed modes for variational problems in mathematics and physics. *Proceedings of the National Academy of Sciences*, 110(46):18368–18373, 2013.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Paris Perdikaris, Maziar Raissi, Andreas Damianou, ND Lawrence, and George Em Karniadakis. Nonlinear information fusion algorithms for data-efficient multi-fidelity modelling. *Proc. R. Soc. A*, 473(2198):20160751, 2017.

Joshua L Proctor, Steven L Brunton, Bingni W Brunton, and JN Kutz. Exploiting sparsity and equation-free architectures in complex systems. *The European Physical Journal Special Topics*, 223(13):2665–2684, 2014.

M. Raghu, B. Poole, J. Kleinberg, S. Ganguli, and J. Sohl-Dickstein. On the expressive power of deep neural networks. *arXiv preprint arXiv:1606.05336*, 2016.

Maziar Raissi. Parametric gaussian process regression for big data. *arXiv preprint arXiv:1704.03144*, 2017.

Maziar Raissi. Forward-backward stochastic neural networks: Deep learning of high-dimensional partial differential equations. *arXiv preprint arXiv:1804.07010*, 2018.

Maziar Raissi and George Karniadakis. Deep multi-fidelity Gaussian processes. *arXiv preprint arXiv:1604.07484*, 2016.

Maziar Raissi and George Em Karniadakis. Hidden physics models: Machine learning of nonlinear partial differential equations. *Journal of Computational Physics*, 2017.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Inferring solutions of differential equations using noisy multi-fidelity data. *Journal of Computational Physics*, 335: 736–746, 2017a.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Machine learning of linear differential equations using Gaussian processes. *Journal of Computational Physics*, 348: 683 – 693, 2017b. ISSN 0021-9991. doi: http://dx.doi.org/10.1016/j.jcp.2017.07.050. URL `http://www.sciencedirect.com/science/article/pii/S0021999117305582`.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10561*, 2017c.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv preprint arXiv:1711.10566*, 2017d.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Multistep neural networks for data-driven discovery of nonlinear dynamical systems. *arXiv preprint arXiv:1801.01236*, 2018a.

Maziar Raissi, Paris Perdikaris, and George Em Karniadakis. Numerical gaussian processes for time-dependent and nonlinear partial differential equations. *SIAM Journal on Scientific Computing*, 40(1):A172–A198, 2018b.

R Rico-Martinez, K Krischer, IG Kevrekidis, MC Kube, and JL Hudson. Discrete-vs. continuous-time nonlinear signal processing of cu electrodissolution data. *Chemical Engineering Communications*, 118(1):25–48, 1992.

Anthony John Roberts. *Model emergent dynamics in complex systems*. SIAM, 2014.

Samuel H. Rudy, Steven L. Brunton, Joshua L. Proctor, and J. Nathan Kutz. Data-driven discovery of partial differential equations. *Science Advances*, 3(4), 2017. doi: 10.1126/sciadv.1602614. URL `http://advances.sciencemag.org/content/3/4/e1602614`.

Hayden Schaeffer, Russel Caflisch, Cory D Hauck, and Stanley Osher. Sparse dynamics for partial differential equations. *Proceedings of the National Academy of Sciences*, 110(17): 6634–6639, 2013.

Michael Schmidt and Hod Lipson. Distilling free-form natural laws from experimental data. *science*, 324(5923):81–85, 2009.

Michael D Schmidt, Ravishankar R Vallabhajosyula, Jerry W Jenkins, Jonathan E Hood, Abhishek S Soni, John P Wikswo, and Hod Lipson. Automated refinement and inference of analytical models for metabolic networks. *Physical biology*, 8(5):055011, 2011.

George Sugihara, Robert May, Hao Ye, Chih-hao Hsieh, Ethan Deyle, Michael Fogarty, and Stephan Munch. Detecting causality in complex ecosystems. *science*, 338(6106):496–500, 2012.

Kunihiko Taira and Tim Colonius. The immersed boundary method: a projection approach. *Journal of Computational Physics*, 225(2):2118–2137, 2007.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.

Giang Tran and Rachel Ward. Exact recovery of chaotic systems from highly corrupted data. *arXiv preprint arXiv:1607.01067*, 2016.

Henning U Voss, Paul Kolodner, Markus Abel, and Jürgen Kurths. Amplitude equations from spatiotemporal binary-fluid convection data. *Physical review letters*, 83(17):3422, 1999.

Wen-Xu Wang, Rui Yang, Ying-Cheng Lai, Vassilios Kovanis, and Celso Grebogi. Predicting catastrophes in nonlinear dynamical systems by compressive sensing. *Physical Review Letters*, 106(15):154101, 2011.

Hao Ye, Richard J Beamish, Sarah M Glaser, Sue CH Grant, Chih-hao Hsieh, Laura J Richards, Jon T Schnute, and George Sugihara. Equation-free mechanistic ecosystem forecasting using empirical dynamic modeling. *Proceedings of the National Academy of Sciences*, 112(13):E1569–E1576, 2015.

Journal of Machine Learning Research 18 (2018) 1-6 Submitted 2/18; Revised 6/18; Published 8/18

OpenEnsembles: A Python Resource for Ensemble Clustering

Tom Ronan tom_ronan@hotmail.com
Shawn Anastasio shawn.anastasio@wustl.edu
Zhijie Qi qizhijie@wustl.edu
Roman Sloutsky sloutsky@wustl.edu
Kristen M. Naegle knaegle@wustl.edu
Department of Biomedical Engineering and the Center for Biological Systems Engineering
Washington University in St. Louis
St. Louis, MO 63122, USA

Pedro Henrique S. Vieira Tavares pedrohsvt@gmail.com
Department of Computer Science
University of Arizona
Tucson, AZ 85721, USA

Editor: Antti Honkela

Abstract

In this paper we introduce OpenEnsembles, a Python toolkit for performing and analyzing ensemble clustering. Ensemble clustering is the process of creating many clustering solutions for a given dataset and utilizing the relationships observed across the ensemble to identify final solutions, which are more robust, stable or better than the individual solutions within the ensemble. The OpenEnsembles library provides a unified interface for applying transformations to data, clustering data, visualizing individual clustering solutions, visualizing and finishing the ensemble, and calculating validation metrics for a clustering solution for any given partitioning of the data. We have documented examples of using OpenEnsembles to create, analyze, and visualize a number of different types of ensemble approaches on toy and example datasets. OpenEnsembles is released under the GNU General Public License version 3, can be installed via Conda or the Python Package Index (pip), and is available at `https://github.com/NaegleLab/OpenEnsembles`.

Keywords: Unsupervised Learning, Ensembles, Clustering, Ensemble Clustering, Finishing Techniques

1. Introduction

Clustering, a type of unsupervised learning, has been instrumental in a large number of fields for reducing data dimensionality, identifying important features, and uncovering the underlying structure of relationships that give rise to the sampled data under consideration. However, a single clustering result may represent a spurious solution (such as when an algorithm is stuck in a local minima) or instead represent just one of many possible structures within of complex data (such as when partitioning of an image identifies underlying shapes but fails to find differences in lighting). Ensemble clustering can be used to overcome the limitations of a single clustering solution by clustering repeatedly, each time making some perturbation to either the data or the approach to clustering. The ensemble of clustering results provides more information about the structure of the underlying data, and how likely any particular grouping of objects is across the ensemble. There are several aspects to performing ensemble clustering, which include: (i) determining and implementing the appropriate perturbations, (ii) deriving a single clustering result that is representative of all the clustering solutions in the ensemble (finishing), and (iii) assessing the results of the finished ensemble, in comparison to the results of an individual solution, to understand the structure and quality of solutions (validation metrics). Our previous review on clustering in biological data covers the types of ensemble clustering and its applications (Ronan et al., 2016) in detail, where Figure 1 contains a table of example ensemble approaches and their motivations. Here, we describe the open source Python toolkit for easily implementing, visualizing, and analyzing ensemble clustering.

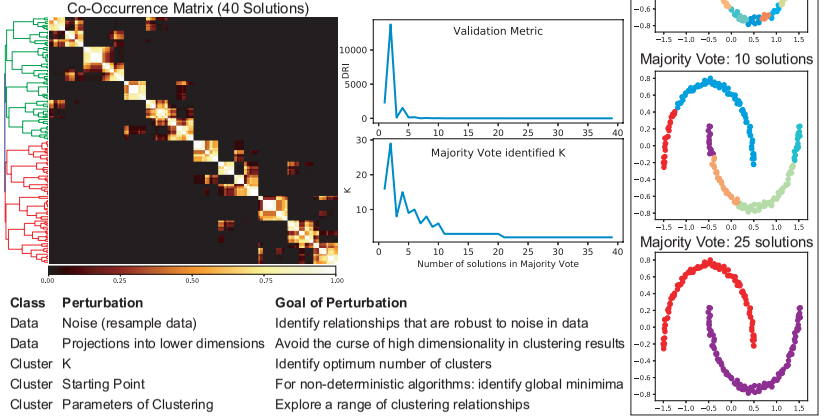
2. Architecture

There are three main OpenEnsembles classes that pair with the main aspects of ensemble clustering: data (storing and manipulating data), cluster (clustering data), and validation (assessing the degree of success of a clustering solution according to an objective such as compactness or connectedness). A common feature across these classes is that they are container objects for housing data, either data matrices (data class), clustering solutions (cluster class), or validation metric results of a clustering solution (validation class). Additionally, each class has a primary function (to transform, cluster, or calculate), where the classes create a common interface for interacting with a variety of transformations, clustering algorithms, and validation metrics. Interacting with any specific transformation of data, retrieving a specific clustering solution or validation metric is done by using the user-defined dictionary key that describes the specific instance. Here we will demonstrate many of the OpenEnsembles features by recreating the first publication of the use of ensemble clustering – Ana Fred's use of ensembles to form a final solution through the majority vote across many non-deterministic k-means solutions (Fred, 2001). OpenEnsembles is built using many open source Python projects, including: scikit-learn (Pedregosa et al., 2012), Pandas (McKinney, 2010), Matplotlib (Hunter, 2007), NetworkX (Hagberg et al., 2008), and NumPy (Walt et al., 2011).

Data

Data is instantiated using a pandas DataFrame object, therefore all the flexibility of loading data from different file formats can be used to coerce the data of interest into an OpenEnsembles data object and metadata can be retained for future analysis. Once instantiated, data can be transformed or used as the basis for clustering or plotting. The *data* container class keeps track of new transformations of the 'parent' data with user-defined keys. Figure 1 shows the instantiation of an OpenEnsembles data object with 200 samples making two half-moon structures. Available transformations currently include: log, principal component analysis (PCA), range scaling, z-score, and internal normalization (such as normalizing to a specific data feature).

```
1  import pandas as pd
2  from sklearn import datasets
3  import openensembles as oe
4  #Set up a dataset and put in pandas DataFrame.
5  x, y = datasets.make_moons(n_samples=200, shuffle=True, noise=0.02, random_state=None)
6  df = pd.DataFrame(x)
7  #instantiate the oe data object
8  dataObj = oe.data(df, [1,2])
9  #instantiate an oe clustering object
10 c = oe.cluster(dataObj)
11 c_MV_arr = []
12 val_arr = []
13 for i in range(0,39):
14     # add a new clustering solution, with a unique name
15     name = 'kmeans_' + str(i)
16     c.cluster('parent', 'kmeans', name, K=16, init = 'random', n_init = 1)
17     # calculate a new majority vote solution, where c has one more solution on each iteration
18     c_MV_arr.append(c.finish_majority_vote(threshold=0.5))
19     #calculate the determinant ratio metric for each majority vote solution
20     v = oe.validation(dataObj, c_MV_arr[i])
21     val_name = v.calculate('det_ratio', 'majority_vote', 'parent')
22     val_arr.append(v.validation[val_name])
23 #calculate the co-occurrence matrix
24 coMat = c.co_occurrence_matrix()
```



Class	Perturbation	Goal of Perturbation
Data	Noise (resample data)	Identify relationships that are robust to noise in data
Data	Projections into lower dimensions	Avoid the curse of high dimensionality in clustering results
Cluster	K	Identify optimum number of clusters
Cluster	Starting Point	For non-deterministic algorithms: identify global minimima
Cluster	Parameters of Clustering	Explore a range of clustering relationships

Figure 1: **Code and outputs of OpenEnsembles.** An ensemble approach (Fred, 2001) to find stable and optimal solutions from the combination of many solutions of the non-deterministic k-means algorithm with majority vote. Code for plotting figure panels is not included (for brevity). K-means can only find clusters that are spheroidally shaped, but as an ensemble, it can identify alternate structures. DRI stands for determinant ratio index – a measure of connectedness in clusters

Cluster

The cluster class is instantiated based on an OpenEnsembles data object, on which all clustering will be performed. OpenEnsembles provides a common interface to clustering, currently providing an interface to all available scikit-learn clustering algorithms. For example, code line 16 (Figure 1) shows how OpenEnsembles is used to call k-means clustering, with K=16 clusters, on the 'parent' data source. Any algorithmic parameters desired are then passed as a dictionary of variable parameters. For example, here, in order to reproduce Ana Fred's original study, we have to overwrite the default arguments of scikit-learn's k-means algorithm, which has ensemble clustering baked into it to lend increased determinism. By looping around OpenEnsembles clustering with random initializations, we are producing

a new clustering solution each time, which is added to the clustering object container and is accessed using the unique key (here it is k-means_⟨number⟩).

Often, one wants to produce a final partition derived from consensus across the ensemble of solutions – i.e. 'finish' the ensemble by building a final, single partitioning of the data. In the current version of OpenEnsembles, we have implemented majority vote as proposed by Ana Fred (Fred, 2001) and mixture models as proposed by Topchy et al. (Topchy et al., 2005). Additionally, there are two finishing techniques that operate on the co-occurrence matrix, as we have proposed previously (Schaberg et al., 2017) (Naegle et al., 2012). An entry in the co-occurrence matrix is the number of times a pair of objects cluster across the ensemble. One method treats the co-occurrence matrix as a similarity matrix and uses linkage clustering to identify clusters (Figure 1). The second method treats the co-occurrence matrix as an adjacency matrix and then finds complete subgraphs within the thresholded co-occurrence matrix via k-cliques and percolation Palla et al. (2005). Final partitions are returned as clustering objects and can be plotted and evaluated like any other clustering solution (Figure 1).

Validation

Quantitative evaluation of how well data is clustered, according to a particular objective function, is called a validation metric. Different validation metrics prioritize different aspects of clustering outcomes such as connectivity, compactness, or separation. We have written a Python package of 28 validation metrics, covering the breadth of the clValid R package of validation metrics (Brock et al., 2008). These validation metrics are available for direct use, or through the OpenEnsembles validation class, which wraps calls to validation metrics for a unified interface with other OpenEnsembles classes. As one can see from the determinant ratio index plot in Figure 1, as the number of clustering solutions in the ensemble increases, the solution both (a) stabilizes and (b) correctly identifies the number of inherent, connected clusters within the data.

Conclusion

Implementation and analysis of ensemble clustering may now be done within in succinct and readable Python code (here, roughly 20 lines of code to reproduce an entire paper of results, Figure 1). Beyond the main functionality of OpenEnsembles, to perform, finish, and validate clustering solutions on data, OpenEnsembles also contains features for calculating co-occurrence, mutual information (overlap/similarity between clustering solutions), and plotting of data and matrices generated. Future work will ideally incorporate expanded selections of algorithms and fuzzy partitioning of data, which is especially amenable to ensemble clustering.

Acknowledgments

We wish to thank the developers of the open source Python scientific and machine learning community. Pedro Henrique S. Vieira Tavares was supported by the Coordination for the

Improvement of Higher Education Personnel (CAPES) and the Brazil Scientific Mobility Program.

References

Guy Brock, Vasyl Pihur, Susmita Datta, and Somnath Datta. clValid : An R Package for Cluster Validation. *J. Stat. Softw.*, 25(4):1–22, 2008.

Ana Fred. Finding consistent clusters in data partitions. In Josef Kittler and Fabio Roli, editors, *Mult. Classif. Syst.*, pages 309–318. Springer, lncs 2096 edition, 2001.

Aric A Hagberg, Los Alamos National, and Los Alamos. Exploring Network Structure, Dynamics, and Function using NetworkX. In *Proc. 7th Python Sci. Conf. (SciPy 2008)*, number SciPy 2008, pages 11–15, 2008.

John D. Hunter. Matplotlib: A 2D Graphics Environment. *Comput. Sci. Eng.*, (9):90–95, 2007.

Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proc. 9th Python Sci. Conf.*, pages 51–56, 2010.

Kristen M Naegle, Forest M White, Douglas A Lauffenburger, and Michael B Yaffe. Robust co-regulation of tyrosine phosphorylation sites on proteins reveals novel protein interactions. *Mol. Biosyst.*, 8(10):2771–2782, Aug 2012.

Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043): 814–8, 2005.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.*, 12:2825–2830, 2012.

Tom Ronan, Zhijie Qi, and Kristen M. Naegle. Avoiding common pitfalls when clustering biological data. *Sci. Signal*, 9(432):re6, 2016.

Katherine E. Schaberg, Venktesh S Shirure, Elizabeth A Worley, Steven C George, and Kristen M Naegle. Ensemble clustering of phosphoproteomic data identifies differences in protein interactions and cell-cell junction integrity of HER2-overexpressing cells. *Integr. Biol.*, 9:539–547, 2017.

Alexander Topchy, Anil K Jain, and William Punch. Clustering ensembles: models of consensus and weak partitions. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(12):1866–81, Dec 2005.

S. V. D. Walt, S. C. Colbert, and G. Varoquaux. The NumPy Array: A structure for efficient numerical computation. *Comput. Sci. Eng.*, 13(2):22–30, 2011.

Journal of Machine Learning Research 19 (2018) 1-21 Submitted 5/16; Revised 1/18; Published 8/18

Importance Sampling for Minibatches

Dominik Csiba cdominik@gmail.com
School of Mathematics
University of Edinburgh
Edinburgh, United Kingdom

Peter Richtárik* peter.richtarik@ed.ac.uk
School of Mathematics
University of Edinburgh
Edinburgh, United Kingdom

Editor: Benjamin Recht

Abstract

Minibatching is a very well studied and highly popular technique in supervised learning, used by practitioners due to its ability to accelerate training through better utilization of parallel processing power and reduction of stochastic variance. Another popular technique is importance sampling—a strategy for preferential sampling of more important examples also capable of accelerating the training process. However, despite considerable effort by the community in these areas, and due to the inherent technical difficulty of the problem, there is virtually no existing work combining the power of importance sampling with the strength of minibatching. In this paper we propose the first practical *importance sampling for minibatches* and give simple and rigorous complexity analysis of its performance. We illustrate on synthetic problems that for training data of certain properties, our sampling can lead to several orders of magnitude improvement in training time. We then test the new sampling on several popular data sets, and show that the improvement can reach an order of magnitude.

keywords: empirical risk minimization; importance sampling; minibatching; variance-reduced methods; convex optimization

1. Introduction

Supervised learning is a widely adopted learning paradigm with important applications such as regression, classification and prediction. The most popular approach to training supervised learning models is via empirical risk minimization (ERM). In ERM, the practitioner collects data composed of example-label pairs, and seeks to identify the best predictor by minimizing the empirical risk, i.e., the average risk associated with the predictor over the training data.

With ever increasing demand for accuracy of the predictors, largely due to successful industrial applications, and with ever more sophisticated models that need to be trained, such as deep neural networks Hinton (2007); Krizhevsky et al. (2012), or multiclass classification Huang et al. (2012), increasing volumes of data are used in the training phase. This leads to huge and hence extremely computationally intensive ERM problems.

Batch algorithms—methods that need to look at all the data before taking a single step to update the predictor—have long been known to be prohibitively impractical to use. Typical examples of batch methods are gradient descent and classical quasi-Newton methods. One of the most popular algorithms for overcoming the deluge-of-data issue is stochastic gradient descent (SGD), which can be traced back to a seminal work of Robbins and Monro (1951). In SGD, a single random example is selected in each iteration, and the predictor is updated using the information obtained by computing the gradient of the loss function associated with this example. This leads to a much more fine-grained iterative process, but at the same time introduces considerable stochastic noise, which eventually—typically after one or a few passes over the data—effectively halts the progress of the method, rendering it unable to push the training error (empirical risk) to the realm of small values.

1.1 Strategies for dealing with stochastic noise

Several approaches have been proposed to deal with the issue of stochastic noise in the finite-data regime. The most important of these are i) decreasing stepsizes, ii) minibatching, iii) importance sampling and iv) variance reduction via "shift", listed here from historically first to the most modern.

The first strategy, *decreasing stepsizes*, takes care of the noise issue by a gradual and direct scale-down process, which ensures that SGD converges to the ERM optimum Zhang (2004). However, an unwelcome side effect of this is a considerable slowdown of the iterative process Bottou (2010). For instance, the convergence rate is sublinear even if the function to be minimized is strongly convex.

The second strategy, *minibatching*, deals with the noise by utilizing a random set of examples in the estimate of the gradient, which effectively decreases the variance of the estimate Shalev-Shwartz et al. (2011). However, this has the unwelcome side-effect of requiring more computation. On the other hand, if a parallel processing machine is available, the computation can be done concurrently, which ultimately leads to speedup. This strategy does not result in an improvement of the convergence rate (unless progressively larger minibatch sizes are used, at the cost of further computational burden Friedlander and Schmidt (2012)), but can lead to massive improvement of the leading constant, which ultimately means acceleration (almost linear speedup for sparse data) Takáč et al. (2013).

The third strategy, *importance sampling*, operates by a careful data-driven design of the probabilities of selecting examples in the iterative process, leading to a reduction of the variance of the stochastic gradient thus selected. Typically, the overhead associated with computing the sampling probabilities and with sampling from the resulting distribution is negligible, and hence the net effect is speedup. In terms of theory, for standard SGD this improves a non-dominant term in the complexity. On the other hand, when SGD is combined with variance reduction, then this strategy leads to the improvement of the leading constant in the complexity estimate, typically via replacing the maximum of certain data-dependent quantities by their average Richtárik and Takáč (2016b); Konečný et al. (2017); Zhao and Zhang (2015); Qu et al. (2015); Needell et al. (2014); Csiba and Richtárik (2015); Csiba et al. (2015).

∗. This author would like to acknowledge support from the EPSRC Grant EP/K02325X/1, *Accelerated Coordinate Descent Methods for Big Data Optimization* and the EPSRC Fellowship EP/N005538/1, *Randomized Algorithms for Extreme Convex Optimization*.

Finally, and most recently, there has been a considerable amount of research activity due to the ground-breaking realization that one can gain the benefits of SGD (cheap iterations) without having to pay through the side effects mentioned above (e.g., halt in convergence due to decreasing stepsizes or increase of workload due to the use of minibatches) in the finite data regime. The result, in theory, is that for strongly convex losses (for example), one does not have to suffer sublinear convergence any more, but instead a fast linear rate "kicks in". In practice, these methods dramatically surpass all previous existing approaches.

The main algorithmic idea is to change the search direction itself, via a properly designed and cheaply maintainable *"variance-reducing shift"* (control variate). Methods in this category are of two types: those operating in the primal space (i.e., directly on ERM) and those operating in a dual space (i.e., with the dual of the ERM problem). Methods of the primal variety include SAG Schmidt et al. (2013), SVRG Johnson and Zhang (2013), S2GD Konečný and Richtárik (2017), proxSVRG Xiao and Zhang (2014), SAGA Defazio et al. (2014), mS2GD Konečný et al. (2016) and MISO Mairal (2015). Methods of the dual variety work by updating randomly selected dual variables, which correspond to examples. These methods include SCD Shalev-Shwartz and Tewari (2011), RCDM Nesterov (2012); Richtárik and Takáč (2014), SDCA Shalev-Shwartz and Zhang (2013b), Hydra Richtárik and Takáč (2016a); Fercoq et al. (2014), mSDCA Takáč et al. (2013), APCG Lin et al. (2015), AsySPDC Liu and Wright (2015), RCD Necoara and Patrascu (2014), APPROX Fercoq and Richtárik (2015), SPDC Zhang and Xiao (2015), ProxSDCA Shalev-Shwartz and Zhang (2012), ASDCA Shalev-Shwartz and Zhang (2013a), IProx-SDCA Zhao and Zhang (2015), and QUARTZ Qu et al. (2015).

1.2 Combining strategies

We wish to stress that the key strategies, mini-batching, importance sampling and variance-reducing shift, should be seen as orthogonal tricks, and as such they can be combined, achieving an amplification effect. For instance, the first primal variance-reduced method allowing for mini-batching was Konečný et al. (2016); while dual-based methods in this category include Shalev-Shwartz and Zhang (2013a); Qu et al. (2015); Csiba and Richtárik (2015). Variance-reduced methods with importance sampling include Nesterov (2012); Richtárik and Takáč (2014); Richtárik and Takáč (2016b); Qu and Richtárik (2016) for general convex minimization problems, and Zhao and Zhang (2015); Qu et al. (2015); Needell et al. (2014); Csiba and Richtárik (2015) for ERM.

2. Contributions

Despite considerable effort of the machine learning and optimization research communities, virtually no importance sampling for minibatches was previously proposed, nor analyzed.[1]. The reason for this lies in the underlying theoretical and computational difficulties associated with the design and successful implementation of such a sampling. One needs to come up with a way to focus on a reasonable set of subsets (minibatches) of the examples to be used in each iteration (issue: there are many subsets; which ones to choose?), assign meaningful data-dependent non-uniform probabilities to them (issue: how?), and then be able to sample these subsets according to the chosen distribution (issue: this could be computationally expensive).

1. A brief note in Richtárik and Takáč (2016b) is an exception, but the sampling is different from ours, was not implemented nor tested, leads to the necessity to solve a linear program and hence is impractical. Another exception is Harikandeh et al. (2015).

The tools that would enable one to consider these questions did not exist until recently. However, due to a recent line of work on analyzing variance-reduced methods utilizing what is known as *arbitrary sampling* Richtárik and Takáč (2016b); Qu et al. (2015); Qu and Richtárik (2016); Qu and Richtárik (2016); Csiba and Richtárik (2015), we are able to ask these questions and provide answers. In this work we design a novel family of samplings—*bucket samplings*—and a particular member of this family—*importance sampling for minibatches*. We illustrate the power of this sampling in combination with the reduced-variance dfSDCA method for ERM. This method is a primal variant of SDCA, first analyzed by Shalev-Shwartz (2015), and extended by Csiba and Richtárik (2015) to the arbitrary sampling setting. However, our sampling can be combined with any stochastic method for ERM, such as SGD or S2GD, and extends beyond the realm of ERM, to convex optimization problems in general. However, for simplicity, we do not discuss these extensions in this work.

We analyze the performance of the new sampling theoretically, and by inspecting the results we are able to comment on when can one expect to be able to benefit from it. We illustrate on synthetic data sets with varying distributions of example sizes that our approach can lead to *dramatic speedups* when compared against standard (uniform) minibatching, of *one or more degrees of magnitude*. We then test our method on real data sets and confirm that the use of importance minibatching leads to up to an order of magnitude speedup. Based on our experiments and theory, we predict that for real data with particular shapes and distributions of example sizes, importance sampling for minibatches will operate in a favourable regime, and can lead to speedup higher than one order of magnitude.

2.1 Related work

The idea of using non-uniform sampling in the parallel regime is by no means new. In the following we highlight several recent approaches in a chronological order and we describe their main differences to our method.

The first attempt for a potential speed-up using a non-uniform parallel sampling was proposed in Richtárik and Takáč (2016b). However, to compute the optimal probability vector one has to solve a linear programming problem, which can easily be more complex than the original problem. The authors do not propose a practical version, which would overcome this issue.

The approach described in Zhao and Zhang (2014) uses the idea of a stratified sampling, which is a well-known strategy in statistics. The authors use clustering to group the examples into several partitions and sample an example from each of the partitions uniformly. This approach is similar to ours, with two main differences: i) we do not need clustering for our approach (it can be computationally very expensive) ii) we allow non-uniform sampling inside each of the partitions, which leads to the main speed-up in our work.

Instead of directly improving the convergence rate of the methods, the authors in Csiba and Richtárik (2015) propose a strategy to improve the synchronized parallel implementation of a method by a load-balancing scheme. The method divides the examples into

groups, which have similar sum of the amount of nonzero entries. When each core processes a single group, it should take the same time to finish as all the other groups, which leads to shorter waiting time in synchronization. Although this is a non-uniform parallel sampling, this approach takes a completely different direction than our method. The only speedup of the method proposed in the above work is achieved due to a shorter waiting time during the synchronization between parallel processing units, while the method proposed in this work directly decreases the iteration complexity.

Lastly, in Harikandeh et al. (2015) the authors actually propose a scheme for importance sampling with minibatches. In the paper they assume, that they can sample a minibatch with a fixed size (without repetition), such that the probabilities of sampling individual examples will be proportional to some given values. However, this is easier said than done—until our work there was no sampling scheme, which would allow for such minibatches. Therefore, the authors theoretically described an idea, which can be used in practice using our scheme.

3. The Problem

Let $\mathbf{X} \in \mathbb{R}^{d\times n}$ be a data matrix in which features are represented in rows and examples in columns, and let $y \in \mathbb{R}^n$ be a vector of labels corresponding to the examples. Our goal is to find a linear predictor $w \in \mathbb{R}^d$ such that $x_i^\top w \sim y_i$, where the pair $x_i, y_i \in \mathbb{R}^d \times \mathbb{R}$ is sampled from the underlying distribution over data-label pairs. In the L2-regularized Empirical Risk Minimization problem, we find w by solving the optimization problem

$$\min_{w\in\mathbb{R}^d} \left[P(w) := \frac{1}{n}\sum_{i=1}^n \phi_i(\mathbf{X}_{:i}^\top w) + \frac{\lambda}{2}\|w\|_2^2 \right], \tag{1}$$

where $\phi_i : \mathbb{R} \to \mathbb{R}$ is a loss function associated with example-label pair $(\mathbf{X}_{:i}, y_i)$, and $\lambda > 0$. For instance, the square loss function is given by $\phi_i(t) = 0.5(t - y_i)^2$. Our results are not limited to L2-regularized problems though: an arbitrary strongly convex regularizer can be used instead Qu et al. (2015). We shall assume throughout that the loss functions are convex and $1/\gamma$-smooth, where $\gamma > 0$. The latter means that for all $x, y \in \mathbb{R}$ and all $i \in [n] := \{1, 2, \ldots, n\}$, we have

$$|\phi_i'(x) - \phi_i'(y)| \le \frac{1}{\gamma}|x - y|.$$

This setup includes ridge and logistic regression, smoothed hinge loss, and many other problems as special cases Shalev-Shwartz and Zhang (2013b). Again, our sampling can be adapted to settings with non-smooth losses, such as the hinge loss.

4. The Algorithm

In this paper we illustrate the power of our new sampling in tandem with Algorithm 1 (dfSDCA) for solving (1).

The method has two parameters. A "sampling" $\hat{S}$, which is a random set-valued mapping Richtárik and Takáč (2016) with values being subsets of $[n]$, the set of examples. No

Algorithm 1 dfSDCA Csiba and Richtárik (2015)

Parameters: Sampling $\hat{S}$, stepsize $\theta > 0$
Initialization: Choose $\alpha^{(0)} \in \mathbb{R}^n$,
set $w^{(0)} = \frac{1}{\lambda n}\sum_{i=1}^n \mathbf{X}_{:i}\alpha_i^{(0)}$, $p_i = \mathbf{Prob}(i \in \hat{S})$
for $t \ge 1$ **do**
  Sample a fresh random set S_t according to $\hat{S}$
  for $i \in S_t$ **do**
    $\Delta_i = \phi_i'(\mathbf{X}_{:i}^\top w^{(t-1)}) + \alpha_i^{(t-1)}$
    $\alpha_i^{(t)} = \alpha_i^{(t-1)} - \theta p_i^{-1}\Delta_i$
  end for
  $w^{(t)} = w^{(t-1)} - \sum_{i\in S_t} \theta(n\lambda p_i)^{-1}\Delta_i \mathbf{X}_{:i}$
end for

assumptions are made on the distribution of $\hat{S}$ apart from requiring that p_i is positive for each i, which simply means that each example has to have a chance of being picked. The second parameter is a stepsize θ, which should be as large as possible, but not larger than a certain theoretically allowable maximum depending on P and $\hat{S}$, beyond which the method could diverge.

Algorithm 1 maintains n "dual" variables, $\alpha_1^{(t)}, \ldots, \alpha_n^{(t)} \in \mathbb{R}$, which act as variance-reduction shifts. This is most easily seen in the case when we assume that $S_t = \{i\}$ (no minibatching). Indeed, in that case we have

$$w^{(t)} = w^{(t-1)} - \frac{\theta}{n\lambda p_i}(g_i^{(t-1)} + \mathbf{X}_{:i}\alpha_i^{(t-1)}),$$

where $g_i^{(t-1)} := \mathbf{X}_{:i}\Delta_i$ is the stochastic gradient. If θ is set to a proper value, as we shall see next, then it turns out that for all $i \in [n]$, α_i is converging $\alpha_i^* := -\phi_i'(\mathbf{X}_{:i}^\top w^*)$, where w^* is the solution to (1), which means that the shifted stochastic gradient converges to zero. This means that its variance is progressively vanishing, and hence no additional strategies, such as decreasing stepsizes or minibatching are necessary to reduce the variance and stabilize the process. In general, dfSDCA in each step picks a random subset of the examples, denoted as S_t, updates variables $\alpha_i^{(t)}$ for $i \in S_t$, and then uses these to update the predictor w.

4.1 Complexity of dfSDCA

In order to state the theoretical properties of the method, we define

$$E^{(t)} := \frac{\lambda}{2}\|w^{(t)} - w^*\|_2^2 + \frac{\gamma}{2n}\|\alpha^{(t)} - \alpha^*\|_2^2.$$

Most crucially to this paper, we assume the knowledge of parameters $v_1, \ldots, v_n > 0$ for which the following ESO[2] inequality holds

$$\mathbf{E}\left[\left\|\sum_{i\in S_t} h_i \mathbf{X}_{:i}\right\|^2\right] \le \sum_{i=1}^n p_i v_i h_i^2 \tag{2}$$

2. ESO = Expected Separable Overapproximation Richtárik and Takáč (2016); Qu and Richtárik (2016).

holds for all $h \in \mathbb{R}^n$. Tight and easily computable formulas for such parameters can be found in Qu and Richtárik (2016). For instance, whenever $\mathbf{Prob}(|S_t| \leq \tau) = 1$, inequality (2) holds with $v_i = \tau\|\mathbf{X}_{:i}\|^2$. However, this is a conservative choice of the parameters. Convergence of dfSDCA is described in the next theorem.

Theorem 1 (Csiba and Richtárik (2015)) *Assume that all loss functions $\{\phi_i\}$ are convex and $1/\gamma$ smooth. If we run Algorithm 1 with parameter θ satisfying the inequality*

$$\theta \leq \min_i \frac{p_i n\lambda\gamma}{v_i + n\lambda\gamma}, \tag{3}$$

where $\{v_i\}$ satisfy (2), then the potential $E^{(t)}$ decays exponentially to zero as

$$\mathbf{E}\left[E^{(t)}\right] \leq e^{-\theta t}E^{(0)}.$$

Moreover, if we set θ equal to the upper bound in (3) so that

$$\frac{1}{\theta} = \max_i \left(\frac{1}{p_i} + \frac{v_i}{p_i n\lambda\gamma}\right) \tag{4}$$

then

$$t \geq \frac{1}{\theta}\log\left(\frac{(1+\lambda\gamma)E^{(0)}}{\lambda\gamma\epsilon}\right) \qquad \Rightarrow \qquad \mathbf{E}[P(w^{(t)}) - P(w^*)] \leq \epsilon.$$

5. Bucket Sampling

We shall first explain the concept of "standard" importance sampling.

5.1 Standard importance sampling

Assume that $\hat{S}$ always picks a single example only. In this case, (2) holds for $v_i = \|\mathbf{X}_{:i}\|^2$, independently of $p := (p_1, \ldots, p_n)$ Qu and Richtárik (2016). This allows us to choose the sampling probabilities as $p_i \sim v_i + n\lambda\gamma$, which ensures that (4) is minimized. This is *importance sampling*. The number of iterations of dfSDCA is in this case proportional to

$$\frac{1}{\theta^{(\text{imp})}} := n + \frac{\sum_{i=1}^n v_i}{n\lambda\gamma}.$$

If uniform probabilities are used, the average in the above formula gets replaced by the maximum:

$$\frac{1}{\theta^{(\text{unif})}} := n + \frac{\max_i v_i}{\lambda\gamma}.$$

Hence, one should expect the following *speedup* when comparing the importance and uniform samplings:

$$\sigma := \frac{\max_i \|\mathbf{X}_{:i}\|^2}{\frac{1}{n}\sum_{i=1}^n \|\mathbf{X}_{:i}\|^2}. \tag{5}$$

If $\sigma = 10$ for instance, then dfSDCA with importance sampling is $10\times$ faster than dfSDCA with uniform sampling.

5.2 Uniform minibatch sampling

In machine learning, the term "minibatch" is virtually synonymous with a special sampling, which we shall here refer to by the name τ-nice sampling Richtárik and Takáč (2016). Sampling $\hat{S}$ is τ-nice if it picks uniformly at random from the collection of all subsets of $[n]$ of cardinality τ. Clearly, $p_i = \tau/n$ and, moreover, it was show by Qu and Richtárik (2016) that (2) holds with $\{v_i\}$ defined by

$$v_i^{(\tau\text{-nice})} = \sum_{j=1}^d \left(1 + \frac{(|J_j|-1)(\tau-1)}{n-1}\right)\mathbf{X}_{ji}^2, \tag{6}$$

where $J_j := \{i \in [n] : \mathbf{X}_{ji} \neq 0\}$. In the case of τ-nice sampling we have the stepsize and complexity given by

$$\theta^{(\tau\text{-nice})} = \min_i \frac{\tau\lambda\gamma}{v_i^{(\tau\text{-nice})} + n\lambda\gamma}, \tag{7}$$

$$\frac{1}{\theta^{(\tau\text{-nice})}} = \frac{n}{\tau} + \frac{\max_i v_i^{(\tau\text{-nice})}}{\tau\lambda\gamma}. \tag{8}$$

Learning from the difference between the uniform and importance sampling of single example (Section 5.1), one would ideally wish the importance minibatch sampling, which we are yet to define, to lead to complexity of the type (8), where the maximum is replaced by an average.

5.3 Bucket sampling: definition

We now propose a family of samplings, which we call *bucket samplings*. Let $B_1, \ldots, B_\tau$ be a partition of $[n] = \{1, 2, \ldots, n\}$ into τ nonempty sets ("buckets").

Definition 2 (Bucket sampling) *We say that $\hat{S}$ is a bucket sampling if for all $i \in [\tau]$, $|\hat{S} \cap B_i| = 1$ with probability 1.*

Informally, a bucket sampling picks one example from each of the τ buckets, forming a minibatch. Hence, $|\hat{S}| = \tau$ and $\sum_{i\in B_l} p_i = 1$ for each $l = 1, 2 \ldots, \tau$, where, as before, $p_i := \mathbf{Prob}(i \in \hat{S})$. Notice that given the partition, the vector $p = (p_1, \ldots, p_n)$ *uniquely determines* a bucket sampling. Hence, we have a family of samplings indexed by a single n-dimensional vector. Let $\mathcal{P}_B$ be the set of all vectors $p \in \mathbb{R}^n$ describing bucket samplings associated with partition $B = \{B_1, \ldots, B_\tau\}$. Clearly,

$$\mathcal{P}_B = \left\{p \in \mathbb{R}^n : \sum_{i\in B_l} p_i = 1 \text{ for all } l \;\&\; p_i \geq 0 \text{ for all } i\right\}.$$

Note, that the sampling inside each bucket B_i can be performed in $\mathcal{O}(\log|B_i|)$ time using a binary tree, with an initial overhead and memory of $\mathcal{O}(|B_i|\log|B_i|)$, as explained in Nesterov (2012).

5.4 Optimal bucket sampling

The optimal bucket sampling is that for which (4) is minimized, which leads to a complicated optimization problem:

$$\min_{p\in\mathcal{P}_B} \max_i \frac{1}{p_i} + \frac{v_i}{p_i n\lambda\gamma} \quad \text{subject to } \{v_i\} \text{ satisfy (2).}$$

A particular difficulty here is the fact that the parameters $\{v_i\}$ depend on the vector p in a complicated way. In order to resolve this issue, we prove the following result.

Theorem 3 *Let $\hat{S}$ be a bucket sampling described by partition $B = \{B_1, \dots, B_\tau\}$ and vector p. Then the ESO inequality (2) holds for parameters $\{v_i\}$ set to*

$$v_i = \sum_{j=1}^{d} \left(1 + \left(1 - \tfrac{1}{\omega'_j}\right)\delta_j\right)\mathbf{X}_{ji}^2, \tag{9}$$

where $J_j := \{i \in [n] \,:\, \mathbf{X}_{ji} \neq 0\}$, $\delta_j := \sum_{i\in J_j} p_i$ and $\omega'_j := |\{l \,:\, J_j \cap B_l \neq \emptyset\}|$.

Observe that J_j is the set of examples which express feature j, and ω'_j is the number of buckets intersecting with J_j. Clearly, that $1 \leq \omega'_j \leq \tau$ (if $\omega'_j = 0$, we simply discard this feature from our data as it is not needed). Note that the effect of the quantities $\{\omega'_j\}$ on the value of v_i is small. Indeed, unless we are in the extreme situation when $\omega'_j = 1$, which has the effect of neutralizing δ_j, the quantity $1 - 1/\omega'_j$ is between $1 - 1/2$ and $1 - 1/\tau$. Hence, for simplicity, we could instead use the slightly more conservative parameters:

$$v_i = \sum_{j=1}^{d} \left(1 + \left(1 - \frac{1}{\tau}\right)\delta_j\right)\mathbf{X}_{ji}^2.$$

5.5 Uniform bucket sampling

Assume all buckets are of the same size: $|B_l| = n/\tau$ for all l. Further, assume that $p_i = 1/|B_l| = \tau/n$ for all i. Then $\delta_j = \tau|J_j|/n$, and hence Theorem 3 says that

$$v_i^{(\text{unif})} = \sum_{j=1}^{d} \left(1 + \left(1 - \frac{1}{\omega'_j}\right)\frac{\tau|J_j|}{n}\right)\mathbf{X}_{ji}^2, \tag{10}$$

and in view of (4), the complexity of dfSDCA with this sampling becomes

$$\frac{1}{\theta^{(\text{unif})}} = \frac{n}{\tau} + \frac{\max_i v_i^{(\text{unif})}}{\tau\lambda\gamma}. \tag{11}$$

Formula (6) is very similar to the one for τ-nice sampling (10), despite the fact that the sets/minibatches generated by the uniform bucket sampling have a special structure with respect to the buckets. Indeed, it is easily seen that the difference between between $1 + \frac{\tau|J_j|}{n}$ and $1 + \frac{(\tau-1)(|J_j|-1)}{(n-1)}$ is negligible. Moreover, if either $\tau = 1$ or $|J_j| = 1$ for all j, then $\omega'_j = 1$ for all j and hence $v_i = \|\mathbf{X}_{:i}\|^2$. This is also what we get for the τ-nice sampling.

quantity \ iteration	1	2	3	4	5	6
$\max_i(\lvert p_i^{\text{new}} - p_i^{\text{old}}\rvert)$	$7\cdot10^{-5}$	$7\cdot10^{-6}$	$7\cdot10^{-7}$	$8\cdot10^{-8}$	$8\cdot10^{-9}$	$9\cdot10^{-10}$
$\|p^{\text{new}} - p^{\text{old}}\|_2$	$1\cdot10^{-3}$	$2\cdot10^{-4}$	$2\cdot10^{-5}$	$2\cdot10^{-6}$	$2\cdot10^{-7}$	$2\cdot10^{-8}$

Table 1: Example of the convergence speed of the alternating optimization scheme for $w8a$ data set (see Table 5) with $\tau = 8$. The table demonstrates the difference in probabilities for two successive iterations (p^{old} and p^{new}). We observed a similar behaviour for all data sets and all choices of τ.

6. Importance Minibatch Sampling

In the light of Theorem 3, we can formulate the problem of searching for the optimal bucket sampling as

$$\min_{p\in\mathcal{P}_B} \max_i \frac{1}{p_i} + \frac{v_i}{p_i n\lambda\gamma} \quad \text{subject to } \{v_i\} \text{ satisfy (9).} \tag{12}$$

Still, this is not an easy problem. *Importance minibatch sampling* arises as an approximate solution of (12). Note that the uniform minibatch sampling is a feasible solution of the above problem, and hence we should be able to improve upon its performance.

6.1 Approach 1: alternating optimization

Given a probability distribution $p \in \mathcal{P}_B$, we can easily find v using Theorem 3. On the other hand, for any fixed v, we can minimize (12) over $p \in \mathcal{P}_B$ by choosing the probabilities in each group B_l and for each $i \in B_l$ via

$$p_i = \frac{n\lambda\gamma + v_i}{\sum_{j\in B_l} n\lambda\gamma + v_j}. \tag{13}$$

This leads to a natural alternating optimization strategy. An example of the standard convergence behaviour of this scheme is showed in Table 6.1. Empirically, this strategy converges to a pair (p^*, v^*) for which (13) holds. Therefore, the resulting complexity will be

$$\frac{1}{\theta^{(\tau\text{-imp})}} = \frac{n}{\tau} + \max_{l\in[\tau]} \frac{\frac{\tau}{n}\sum_{i\in B_l} v_i^*}{\tau\lambda\gamma}. \tag{14}$$

We can compare this result against the complexity of τ-nice in (8). We can observe that the terms are very similar, up to two differences. First, the importance minibatch sampling has a maximum over group averages instead of a maximum over everything, which leads to speedup, other things equal. On the other hand, $v^{(\tau\text{-nice})}$ and v^* are different quantities. The alternating optimization procedure for computation of (v^*, p^*) is costly, as one iteration takes a pass over all data. Therefore, in the next subsection we propose a closed form formula which, as we found empirically, offers nearly optimal convergence rate.

6.2 Approach 2: practical formula

For each group B_l, let us choose for all $i \in B_l$ the probabilities as follows:

$$\boxed{p_i^* = \frac{n\lambda\gamma + v_i^{(\text{unif})}}{\sum_{k\in B_l} n\lambda\gamma + v_k^{(\text{unif})}}} \tag{15}$$

where $v_i^{(\text{unif})}$ is given by (10). Note that computing all $v_i^{(\text{unif})}$ can be done by visiting every non-zero entry of $\mathbf{X}$ once and and computing all p_i^* is a simple re-weighting. This is the same computational cost as for standard serial importance sampling. Also, this process can be straightforwardly parallelized, fully utilizing all the cores, which leads to τ times faster computations. The overhead of using this sampling approach is therefore at most one pass over the data, which is negligible in most scenarios considered.

After doing some simplifications, the associated complexity result is

$$\frac{1}{\theta^{(\tau\text{-imp})}} = \max_l \left\{ \left(\frac{n}{\tau} + \frac{\frac{\tau}{n}\sum_{i\in B_l} v_i^{(\text{unif})}}{\tau\lambda\gamma} \right) \beta_l \right\}, \tag{16}$$

where

$$\beta_l := \max_{i\in B_l} \frac{n\lambda\gamma + s_i}{n\lambda\gamma + v_i^{(\text{unif})}}, \qquad s_i := \sum_{j=1}^{d} \left(1 + \left(1 - \frac{1}{\omega_j'} \right) \sum_{k\in J_j} p_k^* \right) \mathbf{X}_{ji}^2.$$

We would ideally want to have $\beta_l = 1$ for all l (this is what we get for importance sampling without minibatches). If $\beta_l \approx 1$ for all l, then the complexity $1/\theta^{(\tau\text{-imp})}$ is an improvement on the complexity of the uniform minibatch sampling since the maximum of group averages is always better than the maximum of all elements $v_i^{(\text{uni})}$:

$$\frac{n}{\tau} + \frac{\max_l \left(\frac{\tau}{n}\sum_{i\in B_l} v_i^{(\text{unif})} \right)}{\tau\lambda\gamma} \le \frac{n}{\tau} + \frac{\max_i v_i^{(\text{unif})}}{\tau\lambda\gamma}.$$

Indeed, the difference can be very large.

Finally, we would like to comment on the choice of the partitions $B_1, \ldots, B_\tau$, as they clearly affect the convergence rate. The optimal choice of the partitions is given by minimizing in $B_1, \ldots, B_\tau$ the maximum over group sums in (16), which is a complicated optimization problem. Instead, we used random partitions of the same size in our experiments, which we believe is a good solution for the partitioning problem. The logic is simple: the minimum of the maximum over the group sums will be achieved, when all the group sums have similar values. If we set the partitions to the same size and we distribute the examples randomly, there is a good chance that the group sums will have similar values (especially for large amounts of data).

7. Experiments

We now comment on the results of our numerical experiments, with both synthetic and real data sets. We plot the optimality gap $P(w^{(t)}) - P(w^*)$ and in the case of real data also

the test error (vertical axis) against the computational effort (horizontal axis). We measure computational effort by the number of effective passes through the data divided by τ. We divide by τ as a normalization factor; since we shall compare methods with a range of values of τ. This is reasonable as it simply indicates that the τ updates are performed in parallel. Hence, what we plot is an implementation-independent model for time.

We compared two algorithms:

1) **τ-nice**: dfSDCA using the τ-nice sampling with stepsizes given by (7) and (6),

2) **τ-imp**: dfSDCA using τ-importance sampling (i.e., importance minibatch sampling) defined in Subsection 6.2.

As the methods are randomized, we always plot the average over 5 runs. For each data set we provide two plots. In the left figure we plot the convergence of τ-nice for different values of τ, and in the right figure we do the same for τ-importance. The horizontal axis has the same range in both plots, so they are easily comparable. The values of τ we used to plot are $\tau \in \{1, 2, 4, 8, 16, 32\}$. In all experiments we used the logistic loss: $\phi_i(z) = \log(1 + e^{-y_i z})$ and set the regularizer to $\lambda = \max_i \|\mathbf{X}_{:i}\|/n$. We will observe the theoretical and empirical ratio $\theta^{(\tau\text{-imp})}/\theta^{(\tau\text{-nice})}$. The theoretical ratio is computed from the corresponding theory. The empirical ratio is the ratio between the horizontal axis values at the moments when the algorithms reached the precision 10^{-10}.

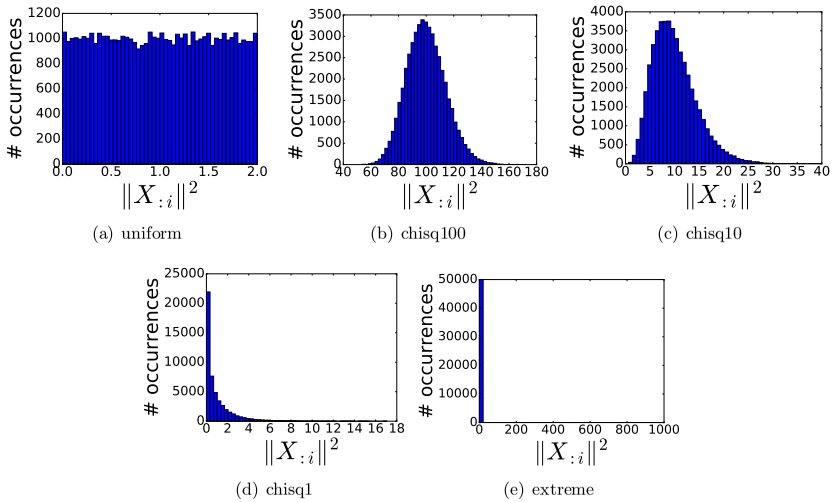
7.1 Artificial data

We start with experiments using artificial data, where we can control the sparsity pattern of $\mathbf{X}$ and the distribution of $\{\|\mathbf{X}_{:i}\|^2\}$. We fix $n = 50,000$ and choose $d = 10,000$ and $d = 1,000$. For each feature we sampled a random sparsity coefficient $\omega_i' \in [0, 1]$ to have the average sparsity $\omega' := \frac{1}{d}\sum_i^d \omega_i'$ under control. We used two different regimes of sparsity: $\omega' = 0.1$ (10% nonzeros) and $\omega' = 0.8$ (80% nonzeros). After deciding on the sparsity pattern, we rescaled the examples to match a specific distribution of norms $L_i = \|\mathbf{X}_{:i}\|^2$; see Table 2. The code column shows the corresponding code in Julia to create the vector of norms L. The distributions can be also observed as histograms in Figure 1.

label	code	σ
extreme	L = ones(n);L[1] = 1000	980.4
chisq1	L = rand(chisq(1),n)	17.1
chisq10	L = rand(chisq(10),n)	3.9
chisq100	L = rand(chisq(100),n)	1.7
uniform	L = 2*rand(n)	2.0

Table 2: Distributions of $\|\mathbf{X}_{:i}\|^2$ used in artificial experiments.

The corresponding experiments can be found in Figure 4 and Figure 5. The theoretical and empirical speedup are also summarized in Tables 3 and 4.



(a) uniform (b) chisq100 (c) chisq10

(d) chisq1 (e) extreme

Figure 1: The distribution of $\|\mathbf{X}_{:i}\|^2$ for synthetic data

Data	$\tau=1$	$\tau=2$	$\tau=4$	$\tau=8$	$\tau=16$	$\tau=32$
uniform	1.2 : 1.0	1.2 : 1.1	1.2 : 1.1	1.2 : 1.1	1.3 : 1.1	1.4 : 1.1
chisq100	1.5 : 1.3	1.5 : 1.3	1.5 : 1.4	1.6 : 1.4	1.6 : 1.4	1.6 : 1.4
chisq10	1.9 : 1.4	1.9 : 1.5	2.0 : 1.4	2.2 : 1.5	2.5 : 1.6	2.8 : 1.7
chisq1	1.9 : 1.4	2.0 : 1.4	2.2 : 1.5	2.5 : 1.6	3.1 : 1.6	4.2 : 1.7
extreme	8.8 : 4.8	9.6 : 6.6	11 : 6.4	14 : 6.4	20 : 6.9	32 : 6.1

Table 3: The **theoretical** : **empirical** ratios $\theta^{(\tau\text{-imp})}/\theta^{(\tau\text{-nice})}$ for sparse artificial data ($\omega'=0.1$)

Data	$\tau=1$	$\tau=2$	$\tau=4$	$\tau=8$	$\tau=16$	$\tau=32$
uniform	1.2 : 1.1	1.2 : 1.1	1.4 : 1.2	1.5 : 1.2	1.7 : 1.3	1.8 : 1.3
chisq100	1.5 : 1.3	1.6 : 1.4	1.6 : 1.5	1.7 : 1.5	1.7 : 1.6	1.7 : 1.6
chisq10	1.9 : 1.3	2.2 : 1.6	2.7 : 2.1	3.1 : 2.3	3.5 : 2.5	3.6 : 2.7
chisq1	1.9 : 1.3	2.6 : 1.8	3.7 : 2.3	5.6 : 2.9	7.9 : 3.2	10 : 3.9
extreme	8.8 : 5.0	15 : 7.8	27 : 12	50 : 16	91 : 21	154 : 28

Table 4: The **theoretical** : **empirical** ratios $\theta^{(\tau\text{-imp})}/\theta^{(\tau\text{-nice})}$. Artificial data with $\omega'=0.8$ (dense)

7.2 Real data

We used several publicly available data sets[3], summarized in Table 5, which we randomly split into a train (80%) and a test (20%) part. The test error is measured by the empirical risk (1) on the test data without a regularizer. The resulting test error was compared against the best achievable test error, which we computed by minimizing the corresponding risk. Experimental results are in Figure 7.3 and Figure 7.3. The theoretical and empirical speedup table for these data sets can be found in Table 6.

Data set	#samples	#features	sparsity	σ
ijcnn1	35,000	23	60.1%	2.04
protein	17,766	358	29.1%	1.82
w8a	49,749	301	4.2%	9.09
url	2,396,130	3,231,962	0.04 %	4.83
aloi	108,000	129	24.6%	26.01

Table 5: Summary of real data sets (σ = predicted speedup).

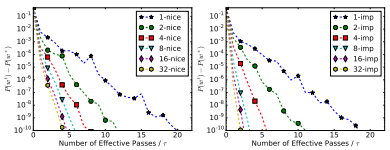
Data	$\tau=1$	$\tau=2$	$\tau=4$	$\tau=8$	$\tau=16$	$\tau=32$
ijcnn1	1.2 : 1.1	1.4 : 1.1	1.6 : 1.3	1.9 : 1.6	2.2 : 1.6	2.3 : 1.8
protein	1.3 : 1.2	1.4 : 1.2	1.5 : 1.4	1.7 : 1.4	1.8 : 1.5	1.9 : 1.5
w8a	2.8 : 2.0	2.9 : 1.9	2.9 : 1.9	3.0 : 1.9	3.0 : 1.8	3.0 : 1.8
url	3.0 : 2.3	2.6 : 2.1	2.0 : 1.8	1.7 : 1.6	1.8 : 1.6	1.8 : 1.7
aloi	13 : 7.8	12 : 8.0	11 : 7.7	9.9 : 7.4	9.3 : 7.0	8.8 : 6.7

Table 6: The **theoretical** : **empirical** ratios $\theta^{(\tau\text{-imp})}/\theta^{(\tau\text{-nice})}$.

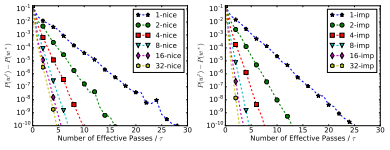
7.3 Conclusion

In all experiments, τ-importance sampling performs significantly better than τ-nice sampling. The theoretical speedup factor computed by $\theta^{(\tau\text{-imp})}/\theta^{(\tau\text{-nice})}$ provides an excellent estimate of the actual speedup. We can observe that on denser data the speedup is higher than on sparse data. This matches the theoretical intuition for v_i for both samplings. Similar behaviour can be also observed for the test error, which is pleasing. As we observed for artificial data, for extreme data sets the speedup can be arbitrary large, even several orders of magnitude. *A rule of thumb: if one has data with large σ, practical speedup from using importance minibatch sampling will likely be dramatic.*

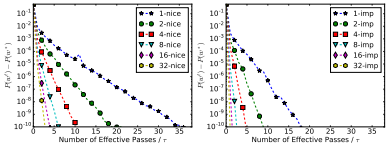
3. https://www.csie.ntu.edu.tw/ cjlin/libsvmtools/datasets/



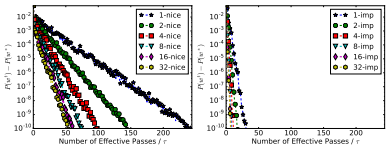
(a) ijcnn1, τ-nice (left), τ-importance (right)



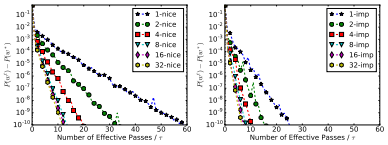
(b) protein, τ-nice (left), τ-importance (right)



(c) w8a, τ-nice (left), τ-importance (right)

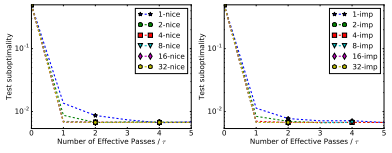


(d) aloi, τ-nice (left), τ-importance (right)

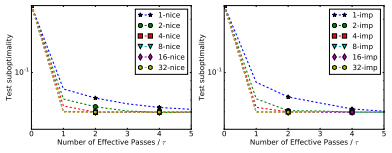


(e) url, τ-nice (left), τ-importance (right)

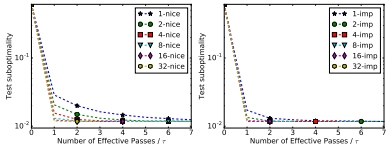
Figure 2: Train error over iterations for data sets from Table 5



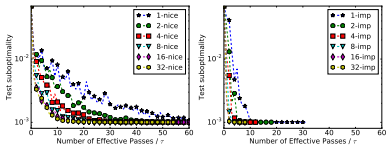
(a) ijcnn1, τ-nice (left), τ-importance (right)



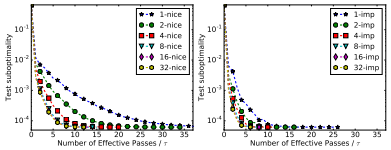
(b) protein, τ-nice (left), τ-importance (right)



(c) w8a, τ-nice (left), τ-importance (right)

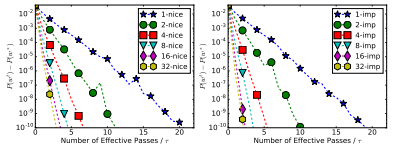


(d) aloi, τ-nice (left), τ-importance (right)

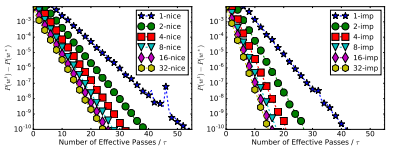


(e) url, τ-nice (left), τ-importance (right)

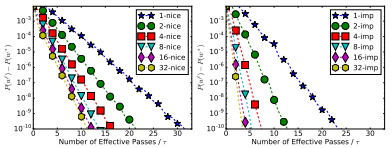
Figure 3: Test error over iterations for data sets from Table 5



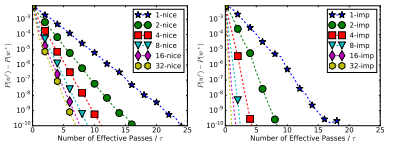
(a) uniform, τ-nice (left), τ-importance (right)



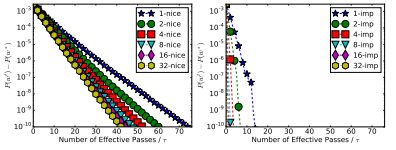
(b) chisq100, τ-nice (left), τ-importance (right)



(c) chisq10, τ-nice (left), τ-importance (right)

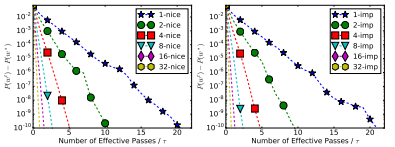


(d) chisq1, τ-nice (left), τ-importance (right)

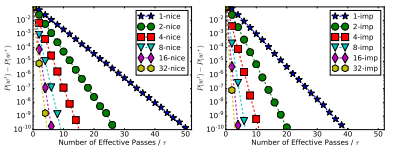


(e) extreme, τ-nice (left), τ-importance (right)

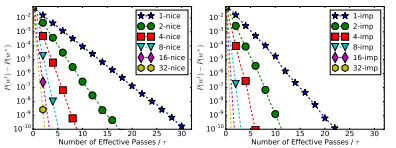
Figure 4: Artificial data sets from Table 2 with $\omega = 0.8$



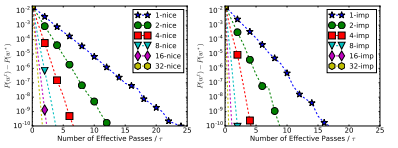
(a) uniform, τ-nice (left), τ-importance (right)



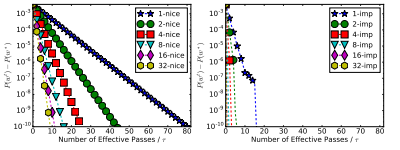
(b) chisq100, τ-nice (left), τ-importance (right)



(c) chisq10, τ-nice (left), τ-importance (right)



(d) chisq1, τ-nice (left), τ-importance (right)



(e) extreme, τ-nice (left), τ-importance (right)

Figure 5: Artificial data sets from Table 2 with $\omega = 0.1$

8. Proof of Theorem 3

8.1 Three lemmas

We first establish three lemmas, and then proceed with the proof of the main theorem. With each sampling $\hat{S}$ we associate an $n \times n$ "probability matrix" defined as follows: $\mathbf{P}_{ij}(\hat{S}) = \mathbf{Prob}(i \in \hat{S}, j \in \hat{S})$. Our first lemma characterizes the probability matrix of the bucket sampling.

Lemma 4 *If $\hat{S}$ is a bucket sampling, then*

$$\mathbf{P}(\hat{S}) = pp^\top \circ (\mathbf{E} - \mathbf{B}) + \mathrm{Diag}(p), \tag{17}$$

where $\mathbf{E} \in \mathbb{R}^{n\times n}$ is the matrix of all ones,

$$\mathbf{B} := \sum_{l=1}^{\tau} \mathbf{P}(B_l), \tag{18}$$

and $\circ$ denotes the Hadamard (elementwise) product of matrices. Note that $\mathbf{B}$ is the 0-1 matrix given by $\mathbf{B}_{ij} = 1$ if and only if i, j belong to the same bucket B_l for some l.

Proof Let $\mathbf{P} = \mathbf{P}(\hat{S})$. By definition

$$\mathbf{P}_{ij} = \begin{cases} p_i & i = j \\ p_i p_j & i \in B_l,\ j \in B_k,\ l \neq k \\ 0 & \text{otherwise.} \end{cases}$$

It only remains to compare this to (17). ■

Lemma 5 *Let J be a nonempty subset of $[n]$, let $\mathbf{B}$ be as in Lemma 4 and put $\omega'_J := |\{l : J \cap B_l \neq \emptyset\}|$. Then*

$$\mathbf{P}(J) \circ \mathbf{B} \succeq \frac{1}{\omega'_J}\mathbf{P}(J). \tag{19}$$

Proof For any $h \in \mathbb{R}^n$, we have

$$h^\top \mathbf{P}(J) h = \left(\sum_{i\in J} h_i\right)^2 = \left(\sum_{l=1}^{\tau} \sum_{i\in J\cap B_l} h_i\right)^2 \leq \omega'_J \sum_{l=1}^{\tau} \left(\sum_{i\in J\cap B_l} h_i\right)^2 = \omega'_J \sum_{l=1}^{\tau} h^\top \mathbf{P}(J\cap B_l) h,$$

where we used the Cauchy-Schwarz inequality. Using this, we obtain

$$\mathbf{P}(J) \circ \mathbf{B} \overset{(18)}{=} \mathbf{P}(J) \circ \sum_{l=1}^{\tau} \mathbf{P}(B_l) = \sum_{l=1}^{\tau} \mathbf{P}(J) \circ \mathbf{P}(B_l) = \sum_{l=1}^{\tau} \mathbf{P}(J \cap B_l) \overset{(8.1)}{\succeq} \frac{1}{\omega'} \mathbf{P}(J).$$

■

Lemma 6 *Let J be any nonempty subset of $[n]$ and $\hat{S}$ be a bucket sampling. Then*

$$\mathbf{P}(J) \circ pp^\top \preceq \left(\sum_{i\in J} p_i\right) \mathrm{Diag}(\mathbf{P}(J \cap \hat{S})). \tag{20}$$

Proof Choose any $h \in \mathbb{R}^n$ and note that

$$h^\top (\mathbf{P}(J) \circ pp^\top) h = \left(\sum_{i\in J} p_i h_i\right)^2 = \left(\sum_{i\in J} x_i y_i\right)^2,$$

where $x_i = \sqrt{p_i} h_i$ and $y_i = \sqrt{p_i}$. It remains to apply the Cauchy-Schwarz inequality:

$$\sum_{i\in J} x_i y_i \leq \sum_{i\in J} x_i^2 \sum_{i\in J} y_i^2$$

and notice that the i-th element on the diagonal of $\mathbf{P}(J \cap \hat{S})$ is p_i for $i \in J$ and 0 for $i \notin J$ ■

8.2 Proof of Theorem 3

By Theorem 5.2 in Qu and Richtárik (2016), we know that inequality (2) holds for parameters $\{v_i\}$ set to

$$v_i = \sum_{j=1}^{d} \lambda'(\mathbf{P}(J_j \cap \hat{S}))\mathbf{X}_{ji}^2,$$

where $\lambda'(\mathbf{M})$ is the largest normalized eigenvalue of symmetric matrix $\mathbf{M}$ defined as

$$\lambda'(\mathbf{M}) := \max_h \left\{ h^\top \mathbf{M} h \ :\ h^\top \mathrm{Diag}(\mathbf{M}) h \leq 1 \right\}.$$

Furthermore,

$$\begin{aligned} \mathbf{P}(J_j \cap \hat{S}) &= \mathbf{P}(J_j) \circ \mathbf{P}(\hat{S}) \\ &\overset{(17)}{=} \mathbf{P}(J_j) \circ pp^\top - \mathbf{P}(J_j) \circ pp^\top \circ \mathbf{B} + \mathbf{P}(J_j) \circ \mathrm{Diag}(p) \\ &\overset{(19)}{\preceq} \left(1 - \frac{1}{\omega'_J}\right) \mathbf{P}(J_j) \circ pp^\top + \mathbf{P}(J_j) \circ \mathrm{Diag}(p) \\ &\overset{(20)}{\preceq} \left(1 - \frac{1}{\omega'_J}\right) \delta_j \mathrm{Diag}(\mathbf{P}(J_j \cap \hat{S})) + \mathrm{Diag}(\mathbf{P}(J_j \cap \hat{S})), \end{aligned}$$

whence $\lambda'(\mathbf{P}(J_j \cap \hat{S})) \leq 1 + (1 - 1/\omega'_j)\,\delta_j$, which concludes the proof.

References

Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT*, pages 177–186. Springer, 2010.

Dominik Csiba and Peter Richtárik. Primal method for ERM with flexible mini-batching schemes and non-convex losses. *arXiv:1506.02227*, 2015.

Dominik Csiba, Zheng Qu, and Peter Richtárik. Stochastic dual coordinate ascent with adaptive probabilities. ICML, 2015.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. In *NIPS 27*, pages 1646–1654, 2014.

Olivier Fercoq and Peter Richtárik. Accelerated, parallel, and proximal coordinate descent. *SIAM Journal on Optimization*, 25(4):1997–2023, 2015.

Olivier Fercoq, Zheng Qu, Peter Richtárik, and Martin Takáč. Fast distributed coordinate descent for minimizing non-strongly convex losses. *IEEE International Workshop on Machine Learning for Signal Processing*, 2014.

Michael P Friedlander and Mark Schmidt. Hybrid deterministic-stochastic methods for data fitting. *SIAM Journal on Scientific Computing*, 34(3):A1380–A1405, 2012.

Reza Harikandeh, Mohamed Osama Ahmed, Alim Virani, Mark Schmidt, Jakub Konečný, and Scott Sallinen. Stop wasting my gradients: Practical SVRG. In *NIPS 28*, pages 2251–2259, 2015.

Geoffrey E Hinton. Learning multiple layers of representation. *Trends in cognitive sciences*, 11(10):428–434, 2007.

Guang-Bin Huang, Hongming Zhou, Xiaojian Ding, and Rui Zhang. Extreme learning machine for regression and multiclass classification. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 42(2):513–529, 2012.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *NIPS 26*, 2013.

Jakub Konečný and Peter Richtárik. S2GD: Semi-stochastic gradient descent methods. *Frontiers in Applied Mathematics and Statistics*, 3(9):1–14, 2017.

Jakub Konečný, Jie Lu, Peter Richtárik, and Martin Takáč. mS2GD: Mini-batch semi-stochastic gradient descent in the proximal setting. *IEEE Journal of Selected Topics in Signal Processing*, 10(2):242–255, 2016.

Jakub Konečný, Zheng Qu, and Peter Richtárik. Semi-stochastic coordinate descent. *Optimization Methods and Software*, 32(5):993–1005, 2017.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS 25*, pages 1097–1105, 2012.

Qihang Lin, Zhaosong Lu, and Lin Xiao. An accelerated proximal coordinate gradient method and its application to regularized empirical risk minimization. *SIAM Journal on Optimization*, 25(4):2244–2273, 2015.

Ji Liu and Stephen J Wright. Asynchronous stochastic coordinate descent: Parallelism and convergence properties. *SIAM Journal on Optimization*, 25(1):351–376, 2015.

Julien Mairal. Incremental majorization-minimization optimization with application to large-scale machine learning. *SIAM Journal on Optimization*, 25(2):829–855, 2015.

Ion Necoara and Andrei Patrascu. A random coordinate descent algorithm for optimization problems with composite objective function and linear coupled constraints. *Computational Optimization and Applications*, 57(2):307–337, 2014.

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In *NIPS 27*, pages 1017–1025, 2014.

Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

Zheng Qu and Peter Richtárik. Coordinate descent methods with arbitrary sampling I: algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

Zheng Qu and Peter Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *NIPS 28*, pages 865–873, 2015.

Peter Richtárik and Martin Takáč. Distributed coordinate descent method for learning with big data. *JMLR*, 17(75):1–25, 2016a.

Peter Richtárik and Martin Takáč. On optimal probabilities in stochastic coordinate descent methods. *Optimization Letters*, 10(6):1233–1243, 2016b.

Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144 (2):1–38, 2014.

Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016.

Herbert Robbins and Sutton Monro. A stochastic approximation method. *Ann. Math. Statist.*, 22(3):400–407, 1951.

Mark Schmidt, Nicolas Le Roux, and Francis Bach. Minimizing finite sums with the stochastic average gradient. *arXiv:1309.2388*, 2013.

Shai Shalev-Shwartz. SDCA without duality. *arXiv:1502.06177*, 2015.

Shai Shalev-Shwartz and Ambuj Tewari. Stochastic methods for ℓ_1-regularized loss minimization. *JMLR*, 12:1865–1892, 2011.

Shai Shalev-Shwartz and Tong Zhang. Proximal stochastic dual coordinate ascent. *arXiv:1211.2717*, 2012.

Shai Shalev-Shwartz and Tong Zhang. Accelerated mini-batch stochastic dual coordinate ascent. In *NIPS 26*, pages 378–385. 2013a.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *JMLR*, 14(1):567–599, 2013b.

Shai Shalev-Shwartz, Yoram Singer, Nathan Srebro, and Andrew Cotter. Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, 127(1):3–30, 2011.

Martin Takáč, Avleen Singh Bijral, Peter Richtárik, and Nathan Srebro. Mini-batch primal and dual methods for SVMs. In *ICML*, 2013.

Lin Xiao and Tong Zhang. A proximal stochastic gradient method with progressive variance reduction. *SIAM Journal on Optimization*, 24(4):2057–2075, 2014.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.

Yuchen Zhang and Lin Xiao. Stochastic primal-dual coordinate method for regularized empirical risk minimization. In *ICML*, 2015.

Peilin Zhao and Tong Zhang. Accelerating minibatch stochastic gradient descent using stratified sampling. *arXiv:1405.3080*, 2014.

Peilin Zhao and Tong Zhang. Stochastic optimization with importance sampling. In *ICML*, 2015.

Journal of Machine Learning Research 19 (2018) 1-42 Submitted 8/16; Published 9/18

Generalized Rank-Breaking: Computational and Statistical Tradeoffs

Ashish Khetan khetan2@illinois.edu
Sewoong Oh swoh@illinois.edu
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

Editor: Guy Lebanon

Abstract

For massive and heterogeneous modern datasets, it is of fundamental interest to provide guarantees on the accuracy of estimation when computational resources are limited. In the application of rank aggregation, for the Plackett-Luce model, we provide a hierarchy of rank-breaking mechanisms ordered by the complexity in thus generated sketch of the data. This allows the number of data points collected to be gracefully traded off against computational resources available, while guaranteeing the desired level of accuracy. Theoretical guarantees on the proposed generalized rank-breaking implicitly provide such trade-offs, which can be explicitly characterized under certain canonical scenarios on the structure of the data. Further, the proposed generalized rank-breaking algorithm involves set-wise comparisons as opposed to traditional pairwise comparisons. The maximum likelihood estimate of pairwise comparisons is computed efficiently using the celebrated minorization maximization algorithm (Hunter, 2004). To compute the pseudo-maximum likelihood estimate of the set-wise comparisons, we provide a generalization of the minorization maximization algorithm and give guarantees on its convergence.

Keywords: Rank aggregation, Plackett-Luce model, Sample and Computational tradeoff.

1. Introduction

In classical statistical inference, we are typically interested in characterizing how more data points improve the accuracy, with little restrictions or considerations on computational aspects of solving the inference problem. However, with massive growths of the amount of data available and also the complexity and heterogeneity of the collected data, computational resources, such as time and memory, are major bottlenecks in many modern applications. As a solution, recent advances in learning theory introduce hierarchies of algorithmic solutions, ordered by the respective computational complexity, for several fundamental machine learning applications in Bousquet and Bottou (2008); Shalev-Shwartz and Srebro (2008); Chandrasekaran and Jordan (2013); Agarwal et al. (2012); Lucic et al. (2015). Guided by sharp analyses on the sample complexity, these approaches provide theoretically sound guidelines that allow the analyst the flexibility to fall back to simpler algorithms to enjoy the full merit of the improved run-time.

Inspired by these advances, we study the time-data tradeoff in rank aggregation. In many applications such as election, policy making, polling, and recommendation systems, we want to aggregate individual preferences to produce a global ranking that best represents the collective social preference. We assume that the data comes from a parametric family of choice models, and learns the parameters that determine the global ranking. Traditionally, each revealed preference is assumed to have one of the following three structures. *Pairwise comparison*, where one item is preferred over another, is common in sports and chess matches. *Best-out-of-κ comparison*, where one is chosen among a set of κ alternatives, is common in historical purchase data. *κ-way comparison*, where we observe a linear ordering of a set of κ candidates, is used in some elections and surveys. We will refer to such structures as *traditional* in comparisons to modern datasets with non-traditional structures. For such traditional preferences, efficient schemes for rank aggregation have been proposed, such as Ford Jr. (1957); Hunter (2004); Hajek et al. (2014); Chen and Suh (2015), which we explain in detail in Section 2.1. However, modern datasets are unstructured and heterogeneous. As Khetan and Oh (2016) show, this can lead to significant increase in the computational complexity, requiring exponential run-time in the size of the problem in the worst case.

To alleviate this computational challenge, we propose a hierarchy of estimators which we call *generalized rank-breaking*, ordered in increasing computational complexity and achieving increasing accuracy. The key idea is to break down the heterogeneous revealed preferences into simpler pieces of ordinal relations, and apply an estimator tailored for those simple structures treating each piece as independent. Several aspects of rank-breaking makes this problem interesting and challenging. A priori, it is not clear which choices of the simple ordinal relations are rich enough to be statistically efficient and yet lead to tractable estimators. Even if we identify which ordinal relations to extract, the ignored correlations among those pieces can lead to an inconsistent estimate, unless we choose carefully which pieces to include and which to omit in the estimation. We further want sharp analysis on the sample complexity, which reveals how computational and statistical efficiencies trade off. We would like to address all these challenges in providing generalized rank-breaking methods.

2. Problem formulation.

We study the problem of aggregating ordinal data based on users' preferences that are expressed in the form of *partially ordered sets (poset)*. A poset is a collection of ordinal relations among items. For example, consider a poset $\{(i_6 \prec \{i_5, i_4\}), (i_5 \prec i_3), (\{i_3, i_4\} \prec \{i_1, i_2\})\}$ over items $\{i_1, \ldots, i_6\}$, where $(i_6 \prec \{i_5, i_4\})$ indicates that item i_5 and i_4 are both preferred over item i_6. Such a relation is extracted from, for example, the user giving a 2-star rating to i_5 and i_4 and a 1-star to i_6.

We assume there are n users and d items. We denote the set of n users by $[n] = \{1, \ldots, n\}$ and the set of d items by $[d]$. We assume that each user $j \in [n]$ is presented with a subset of items $S_j \subseteq [d]$, and independently provides her ordinal preference in the form of a poset, where the ordering is drawn from the Plackett-Luce (PL) model. Since, an ordering drawn from the PL model is consistent, a poset can be represented as a directed acyclic graph (DAG). Let $\mathcal{G}_j$ denote the DAG representation of the poset provided by the user j over $S_j \subseteq [d]$ according to the PL model with weights θ^*. The task is to learn $\widehat{\theta}$, an estimate of the true weights θ^*. Below is an example of a DAG $\mathcal{G}_j$. We use index i to denote items and j to denote users.

Plackett-Luce model. The PL model is a popular choice model from operations research and psychology, used to model how people make choices under uncertainty. It is a special case of *random utility models*, where each item i is parametrized by a latent true utility $\theta_i \in \mathbb{R}$. When offered with

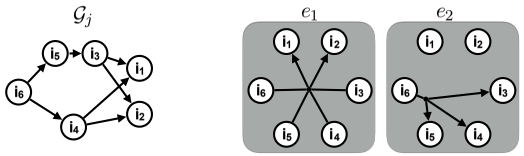


Figure 1: An example of $\mathcal{G}_j$ for user j's consistent poset, and two rank-breaking hyper edges extracted from it: $e_1 = (\{i_6, i_5, i_4, i_3\} \prec \{i_2, i_1\})$ and $e_2 = (\{i_6\} \prec \{i_5, i_4, i_3\})$.

S_j, the user samples the perceived utility U_i for each item independently according to $U_i = \theta_i + Z_i$, where Z_i's are i.i.d. noise. In particular, the PL model assumes Z_i's follow the standard Gumbel distribution. The observed poset is a partial observation of the ordering according to this perceived utilities. We discuss possible extensions to general class of random utility models in Section 2.1.

The particular choice of the Gumbel distribution has several merits, largely stemming from the fact that the Gumbel distribution has a log-concave pdf and is inherently memoryless. In our analyses, we use the log-concavity to show that our proposed algorithm is a concave maximization (Remark 1) and the memoryless property forms the basis of our rank-breaking idea. Precisely, the PL model is statistically equivalent to the following procedure. Consider a ranking as a mapping from a position in the rank to an item, i.e. $\sigma_j : [|S_j|] \to S_j$. It can be shown that the PL model is generated by first independently assigning each item $i \in S_j$ an unobserved value Y_i, exponentially distributed with mean $e^{-\theta_i}$, and the resulting ranking σ_j is inversely ordered in Y_i's so that $Y_{\sigma_j(1)} \le Y_{\sigma_j(2)} \le \cdots \le Y_{\sigma_j(|S_j|)}$.

This inherits the memoryless property of exponential variables, such that $\mathbb{P}(Y_1 < Y_2 < Y_3) = \mathbb{P}(Y_1 < \{Y_2, Y_3\})\mathbb{P}(Y_2 < Y_3)$, leading to a simple interpretation of the PL model as sequential choices:

$$\mathbb{P}_\theta(i_3 \prec i_2 \prec i_1) \;=\; \mathbb{P}_\theta(\{i_3, i_2\} \prec i_1)\mathbb{P}_\theta(i_3 \prec i_2) \;=\; \frac{e^{\theta_{i_1}}}{e^{\theta_{i_1}} + e^{\theta_{i_2}} + e^{\theta_{i_3}}} \times \frac{e^{\theta_{i_2}}}{e^{\theta_{i_2}} + e^{\theta_{i_3}}} \,.$$

In general, for true utility θ^*, we have

$$\mathbb{P}_{\theta^*}[\sigma_j] \;=\; \prod_{i=1}^{|S_j|-1} \frac{e^{\theta^*_{\sigma_j(i)}}}{\sum_{i'=i}^{|S_j|} e^{\theta^*_{\sigma_j(i')}}} \,.$$

We assume that the true utility $\theta^* \in \Omega_b$ where

$$\Omega_b \;=\; \left\{ \theta \in \mathbb{R}^d \;\middle|\; \sum_{i \in [d]} \theta_i = 0, |\theta_i| \le b \text{ for all } i \in [d] \right\}. \tag{1}$$

Notice that centering of θ ensures its uniqueness as PL model is invariant under shifting of θ. The bound b on θ_i is written explicitly to capture the dependence in our main results. We interchangeably refer θ as utilities and weights.

Maximum Likelihood Estimate of DAG. Probability of observing a DAG $\mathcal{G}_j$ is the sum of probabilities of all possible rankings that are consistent with it. Precisely, under the PL model, for

a DAG $\mathcal{G}_j$, we have,

$$\mathbb{P}_\theta[\mathcal{G}_j] \;=\; \sum_{\sigma \in \mathcal{G}_j} \mathbb{P}_\theta[\sigma] \,,$$

where we slightly abuse the notation $\mathcal{G}_j$ to denote the set of all rankings σ that are consistent with the observation. For example, if $\mathcal{G}_j$ consists of only one hyper edge $e_1 = (\{i_3\} \prec \{i_2, i_1\})$ then $\mathbb{P}[\mathcal{G}_j] = \mathbb{P}(i_3 \prec i_2 \prec i_1) + \mathbb{P}(i_3 \prec i_1 \prec i_2)$. The maximum likelihood estimate (MLE) maximizes log-likelihood of observing $\mathcal{G}_j$ for each j:

$$\widehat{\theta} \;\in\; \arg\max_{\theta \in \Omega_b} \left\{ \sum_{j=1}^{n} \log \mathbb{P}_\theta[\mathcal{G}_j] \right\}. \tag{2}$$

When $\mathcal{G}_j$ has a *traditional* structure as explained earlier in this section, then the optimization is a simple multinomial logit regression, that can be solved efficiently with off-the-shelf convex optimization tools. Hajek et al. (2014) provides full analysis of the statistical complexity of this MLE under traditional structures. For general posets, it can be shown that the above optimization is a concave maximization, using similar techniques as Remark 1. However, the summation over rankings in $\mathcal{G}_j$ can involve number of terms super exponential in the size $|S_j|$, in the worst case. This renders MLE intractable and impractical.

Pairwise rank-breaking. A common remedy to this computational blow-up is to use rank-breaking. Rank-breaking traditionally refers to *pairwise rank-breaking*, where a bag of all the pairwise comparisons is extracted from observations $\{\mathcal{G}_j\}_{j \in [n]}$ and is applied to estimators that are tailored for pairwise comparisons, treating each paired outcome as independent. This is one of the motivations behind the algorithmic advances in the popular topic of aggregation from pairwise comparisons in (Ford Jr., 1957; Hunter, 2004; Negahban et al., 2014; Shah et al., 2015a; Maystre and Grossglauser, 2015).

It is computationally efficient to apply maximum likelihood estimator assuming independent pairwise comparisons, which takes $O(d^2)$ operations to evaluate. However, this computational gain comes at the cost of statistical efficiency. Azari Soufiani et al. (2014) showed that if we include all paired comparisons, then the resulting estimate can be statistically inconsistent due to the ignored correlations among the paired orderings, even with infinite samples. In the example from Figure 1, there are 12 paired relations implied by the DAG: $(i_6 \prec i_5), (i_6 \prec i_4), (i_6 \prec i_3), \ldots, (i_3 \prec i_1), (i_4 \prec i_1)$. In order to get a consistent estimate, Azari Soufiani et al. (2014) provide a rule for choosing which pairs to include, and Khetan and Oh (2016) provide an estimator that optimizes how to weigh each of those chosen pairs to get the best finite sample complexity bound. However, such a consistent pairwise rank-breaking results in throwing away many of the ordered relations, resulting in significant loss in accuracy. For example, including any paired relation from $\mathcal{G}_j$ in the example results in a biased estimator. None of the pairwise orderings can be used from $\mathcal{G}_j$, without making the estimator inconsistent as shown in Azari Soufiani et al. (2013). Whether we include all paired comparisons or only a subset of consistent ones, there is a significant loss in accuracy as illustrated in Figure 4. For the precise condition for consistent rank-breaking we refer to (Azari Soufiani et al., 2013, 2014; Khetan and Oh, 2016).

The state-of-the-art approaches operate on either one of the two extreme points on the computational and statistical trade-off. The MLE in (2) requires $O(\sum_{j \in [n]} |S_j|!)$ summations to just

evaluate the objective function, in the worst case. On the other hand, the pairwise rank-breaking requires only $O(d^2)$ summations, but suffers from significant loss in the sample complexity. Ideally, we would like to give the analyst the flexibility to choose a target computational complexity she is willing to tolerate, and provide an algorithm that achieves the optimal trade-off at the chosen operating point.

Contribution. We introduce a novel *generalized rank-breaking* that bridges the gap between MLE and pairwise rank-breaking. Our approach allows the user the freedom to choose the level of computational resources to be used, and provides an estimator tailored for the desired complexity. We prove that the proposed estimator is tractable and consistent, and provide an upper bound and a lower bound on the error rate in the finite sample regime. The analysis explicitly characterizes the dependence on the topology of the data. This in turn provides a guideline for designing surveys and experiments in practice, in order to maximize the sample efficiency. The proposed generalized rank-breaking mechanism involves set-wise comparisons as opposed to traditional pairwise comparisons. In order to compute the rank-breaking estimate, we generalize the celebrated minorization maximization algorithm for computing maximum likelihood estimate of pairwise comparisons (Hunter, 2004) to more general set-wise comparisons and give guarantees on its convergence.

2.1 Related work

In classical statistics, one is interested in the tradeoff between the sample size and the accuracy, with little considerations to the computational complexity or time. As more computations are typically required with increasing availability of data, the computational resources are often the bottleneck. Recently, a novel idea known as "algorithmic weakening" has been investigated to overcome such a bottleneck, in which a hierarchy of algorithms is proposed to allow for faster algorithms at the expense of decreased accuracy. When guided by sound theoretical analyses, this idea allows the statistician to achieve the same level of accuracy and *save* time when more data is available. This is radically different from classical setting where processing more data typically requires more computational time.

Depending on the application, several algorithmic weakenings have been studied. In the application of supervised learning, Bousquet and Bottou (2008) proposed the idea that weaker approximate optimization algorithms are sufficient for learning when more data is available. Various gradient based algorithms are analyzed that show the time-accuracy-sample tradeoff. In a similar context, Shalev-Shwartz and Srebro (2008) analyze a particular implementation of support vector machine and show that the target accuracy can be achieved faster when more data is available, by running the iterative algorithm for shorter amount of time. In the application of de-noising, Chandrasekaran and Jordan (2013) provide a hierarchy of convex relaxations where constraints are defined by convex geometry with increasing complexity. For unsupervised learning, Lucic et al. (2015) introduce a hierarchy of data representations that provide more representative elements when more data is available at no additional computation. Standard clustering algorithms can be applied to thus generated summary of the data, requiring less computational complexity.

In the application of rank aggregation, we follow the principle of algorithmic weakening and propose a novel rank-breaking to allow the practitioner to navigate gracefully the time-sample trade off as shown in the Figure 2. We propose a hierarchy of estimators indexed by $M \in \mathbb{Z}^+$ indicating how complex the estimator is (defined formally in Section 3). Figure 2 shows the result of a experiment on synthetic datasets on how much time (in seconds) and how many samples are

required to achieve a target accuracy. If we are given more samples, then it is possible to achieve the target accuracy, which in this example is MSE$\leq 0.3d^2 \times 10^{-6}$, with fewer operations by using a simpler estimator with smaller M. The details of the experiment is explained in Figure 4.

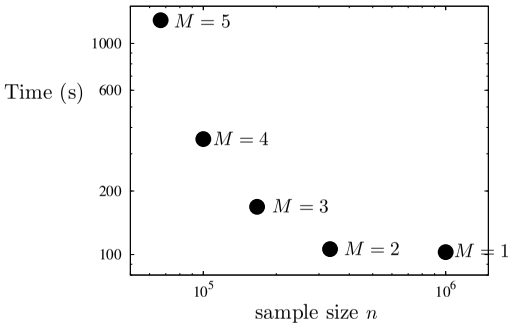


Figure 2: Depending on how much computational resources are available, the various choices of M achieve different operating points on the time-data trade-off to achieve some fixed target accuracy $\varepsilon > 0$. If more samples are available, one can resort to faster methods with smaller M while achieving the same level of accuracy.

Rank aggregation under the PL model has been studied extensively under the *traditional* scenario dating back to Zermelo (1929) who first introduced the PL model for pairwise comparisons. Various approaches for estimating the PL weights from traditional samples have been proposed. The problem can be formulated as a convex optimization that can be solved efficiently using the off-the-shelf solvers. However, tailored algorithms for finding the optimal solution have been proposed in Ford Jr. (1957) and Hunter (2004), which iteratively finds the fixed point of the KKT condition. Negahban et al. (2014) introduce Rank Centrality, a novel spectral ranking algorithm which formulates a random walk from the given data, and show that the stationary distribution provides accurate estimates of the PL weights. Maystre and Grossglauser (2015) provide a connection between those previous approaches, and give a unified random walk approach that finds the fixed point of the KKT conditions.

On the theoretical side, when samples consist of pairwise comparisons, Simons and Yao (1999) first established consistency and asymptotic normality of the maximum likelihood estimate when all teams play against each other. For a broader class of scenarios where we allow for sparse observations, where the number of total comparisons grow linearly in the number of teams, Negahban et al. (2014) show that Rank Centrality achieves optimal sample complexity by comparing it to a lower bound on the minimax rate. For a more general class of traditional observations, including pairwise comparisons, Hajek et al. (2014) provide similar optimal guarantee for the maximum likelihood estimator. Chen and Suh (2015) introduced Spectral MLE that applies Rank Centrality followed by MLE, and showed that the resulting estimate is optimal in L_∞ error as well as the previously analyzed L_2 error. Shah et al. (2015a) study a new measure of the error induced by the

Laplacian of the comparisons graph and prove a sharper upper and lower bounds that match up to a constant factor.

However, in modern applications, the computational complexity of the existing approaches blow-up due to the heterogeneity of modern datasets. Although, statistical and computational tradeoffs have been investigated under other popular choice models such as the Mallows models by Betzler et al. (2014) or stochastically transitive models by Shah et al. (2015b), the algorithmic solutions do not apply to random utility models and the analysis techniques do not extend. We provide a novel rank-breaking algorithms and provide finite sample complexity analysis under the PL model. This approach readily generalizes to some RUMs such as the flipped Gumbel distribution. However, it is also known from Azari Soufiani et al. (2014), that for general RUMs there is no consistent rank-breaking, and the proposed approach does not generalize.

3. Generalized rank-breaking

Given $\mathcal{G}_j$'s representing the users' preferences, *generalized rank-breaking* extracts a set of ordered relations and applies an estimator treating each ordered relation as independent. Concretely, for each $\mathcal{G}_j$, we first extract a maximal ordered partition $\mathcal{P}_j$ of S_j that is consistent with $\mathcal{G}_j$. An ordered partition is a partition with a linear ordering among the subsets, e.g. $\mathcal{P}_j = (\{i_6\} \prec \{i_5, i_4, i_3\} \prec \{i_2, i_1\})$ for $\mathcal{G}_j$ from Figure 1. This is maximal, since we cannot further partition any of the subsets without creating artificial ordered relations that are not present in the original $\mathcal{G}_j$.

To precisely define maximal ordered partition $\mathcal{P}_j$, first, let's define an ordered partition $\widetilde{\mathcal{P}}_j$ of S_j that is consistent with $\mathcal{G}_j$. Consider disjoint subsets $\mathcal{C}_1, \cdots, \mathcal{C}_{\ell_j} \subseteq S_j$ such that their union is S_j that is $\cup_{a=1}^{\ell_j} C_a = S_j$. The subsets $\mathcal{C}_1, \cdots, \mathcal{C}_{\ell_j}$ define an ordered partition

$$\widetilde{\mathcal{P}}_j = \mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_{\ell_j} ,$$

if each ordered relation that can be read from this linear ordering of subsets is present in the DAG $\mathcal{G}_j$. Let $|\mathcal{P}_j|$ denote the size of the partition, that is $|\mathcal{P}_j| = \ell_j$. A maximal ordered partition $\mathcal{P}_j$ is the one which has the largest size.

$$\mathcal{P}_j = \arg\max_{\widetilde{\mathcal{P}}_j} \left\{ |\widetilde{\mathcal{P}}_j| \right\} .$$

In the following we provide an algorithm to find a maximal ordered partition $\mathcal{P}_j$ of S_j that is consistent with a given $\mathcal{G}_j$.

Finding Maximal Ordered Partition. Given a DAG $\mathcal{G}_j$, a corresponding maximal ordered partition $\mathcal{P}_j$ can be extracted by recursively finding common ancestors of the sink-nodes of the vertex induced sub-graph starting with the DAG $\mathcal{G}_j$. Algorithm 1 gives a pseudocode to find $\mathcal{P}_j$'s. Common ancestors of all the sink nodes of a DAG can be found in time $O(d^{2.6})$ using fast algorithms given in (Czumaj et al., 2007; Bender et al., 2005). Therefore, computational complexity of the Algorithm 1 is $O(d^{3.6})$. In line 2, Algorithm 1, $V(\mathcal{G})$ denotes the set of vertices of DAG $\mathcal{G}$. In line 5, $\mathcal{G}(S)$ denote the vertex induced subgraph of graph $\mathcal{G}$ corresponding to vertex set S. Note that Algorithm 1 returns a unique maximal ordered partition $\mathcal{P}_j$ for a given DAG $\mathcal{G}_j$.

In general there is no one-to-one mapping from a DAG $\mathcal{G}_j$ to its maximal ordered partition $\mathcal{P}_j$. There may be many ordered relations present in $\mathcal{G}_j$ that are not represented in the ordered

Algorithm 1 Finding Maximal Ordered Partition

Require: DAG $\mathcal{G}_j$
Ensure: maximal ordered partition $\mathcal{P}_j$
1: $\mathcal{G} \leftarrow \mathcal{G}_j$, $\mathcal{P}_j = \{\}$
2: **while** $|V(\mathcal{G})| > 0$ **do**
3: $S \leftarrow$ Common ancestors of all sink-nodes of DAG $\mathcal{G}$ (Czumaj et al., 2007)
4: $\mathcal{P}_j \leftarrow \mathcal{P}_j \succ \{V(\mathcal{G}) \setminus S\}$
5: $\mathcal{G} \leftarrow \mathcal{G}(S)$
6: **end while**

partition $\mathcal{P}_j$. This gives a many-to-one mapping from $\mathcal{G}_j$ to $\mathcal{P}_j$. In our generalized rank-breaking framework, we can only use those ordered relations that can be represented in an ordered partition. This is required for the estimator to be consistent. This is the cost we pay to reduce computational complexity from $O(|S_j|!)$, complexity of MLE of DAG (2), to $O(M!)$ for a suitably desired $M \in \mathbb{Z}^+$ as explained below. However, if the DAG $\mathcal{G}_j$ represents a full ranking or a traditional structure then its maximal ordered partition $\mathcal{P}_j$ will represent all the ordered relations present in $\mathcal{G}_j$ and our rank-breaking will reduce to MLE of the DAG $\mathcal{G}_j$. In such a case, all the subsets of $\mathcal{P}_j$ will have cardinality one except the least preferred set which can have more than one item in case of best-out-of κ comparison.

Rank-Breaking Graph. The extracted maximal ordered partition $\mathcal{P}_j$ is represented by a directed hypergraph $G_j(S_j, E_j)$, which we call a *rank-breaking graph*. Each edge $e = (B(e), T(e)) \in E_j$ is a directed hyper edge from a subset of nodes $B(e) \subseteq S_j$ to another subset $T(e) \subseteq S_j$. The number of edges in E_j is $|\mathcal{P}_j| - 1$. For each subset in $\mathcal{P}_j$ except for the least preferred subset, there is a corresponding edge whose *top-set* $T(e)$ is the subset, and the *bottom-set* $B(e)$ is the set of all items less preferred than $T(e)$. For the example in Figure 1, we have $E_j = \{e_1, e_2\}$ where $e_1 = (B(e_1), T(e_1)) = (\{i_6, i_5, i_4, i_3\}, \{i_2, i_1\})$ and $e_2 = (B(e_2), T(e_2) = (\{i_6\}, \{i_5, i_4, i_3\})$ extracted from $\mathcal{G}_j$. Algorithm 2 gives the precise method to construct a rank-breaking graph.

Algorithm 2 Constructing Rank-Breaking Graph

Require: maximal ordered partition $\mathcal{P}_j = \mathcal{C}_1 \prec \mathcal{C}_2 \prec \cdots \prec \mathcal{C}_{\ell_j}$ of set S_j
Ensure: directed hypergraph $G_j(S_j, E_j)$
1: construct directed hypergraph $G_j(S_j, E_j = \{\})$
2: **for** $a = 2$ to ℓ_j **do**
3: construct hyper edge e between top-set $T(e) = \mathcal{C}_a$ and bottom-set $B(e) = \cup_{a'=1}^{a-1} \mathcal{C}_{a'}$
4: $E_j \leftarrow E_j \cup e$
5: **end for**
6: Return $G_j(S_j, E_j)$

Denote the probability that $T(e)$ is preferred over $B(e)$ when $T(e) \cup B(e)$ is offered as

$$\mathbb{P}_\theta(e) \;=\; \mathbb{P}_\theta\big(B(e) \prec T(e)\big) \;=\; \sum_{\sigma \in \Lambda_{T(e)}} \frac{\exp\left(\sum_{c=1}^{|T(e)|} \theta_{\sigma(c)}\right)}{\prod_{u=1}^{|T(e)|} \left(\sum_{c'=u}^{|T(e)|} \exp\left(\theta_{\sigma(c')}\right) + \sum_{i \in B(e)} \exp\left(\theta_i\right)\right)} , \qquad (3)$$

which follows from the definition of the PL model, where $\Lambda_{T(e)}$ is the set of all rankings over $T(e)$. The computational complexity of evaluating this probability is determined by the size of the *top-set* $|T(e)|$, as it involves $(|T(e)|!)$ summations.

In the subsequent results, we show that by maximizing likelihood of the hyper edges assuming they are independent we get a consistent estimator. Therefore, our approach provides flexibility to choose which hyper edge to include in the likelihood maximization function. We let the analyst choose the order $M \in \mathbb{Z}^+$ depending on how much computational resource is available, and include only those edges with $|T(e)| \leq M$ in the likelihood objective function.

If in a given G_j there are no hyper edges with top sets of size less than M, then the analyst does not get any ordered relations from that rank-breaking graph under her computational constraint reflected in the particular choice of M. Artificially reducing the size of the top-sets so as to get the hyper edges with top sets of size less than M implies we need to add new ordered relations that are not present in the DAG provided by the user. Such an estimator could result in a non-zero bias. A concrete example of such cases has been studied in Azari Soufiani et al. 2013, where the authors showed that for $M = 1$, applying rank-breaking to those comparisons with top-set larger than one results in non-zero bias.

We emphasize that M is chosen by the analyst and our estimator works for any choice of $M \in \mathbb{Z}^+$. Given unlimited computational resources, an analyst would chose $M = d$, and all the hyper edges would be included in the likelihood objective function.

Sampling. We assume that for each $j \in [n]$, the topology of DAG $\mathcal{G}_j$ which represent the partial preference order provided by the j-th user is fixed apriori. Also, the set of hyper edges $e \in E_j$ of each rank breaking graph $G_j(S_j, E_j)$ that are included in the likelihood objective function are fixed apriori. The randomness that we observe is in the position of the S_j items in the DAG $\mathcal{G}_j$. For an hyper edge $e \in E_j$, the randomness is in which items of the set S_j appear in the bottom $|B(e)|$ positions and the bottom $|T(e)| + |B(e)|$ positions in the preference order of the user j. Note that this precisely captures the randomness due to the PL model in the observed DAG $\mathcal{G}_j$. We do not impose any restrictions on the topology of the DAG $\mathcal{G}_j$'s and each of them can be different. Further, our analysis captures effect of their topologies on the statistical efficiency of the estimation.

Pseudo-MLE of Rank-Breaking Graph. We apply the MLE for comparisons over paired subsets, assuming all hyper edges in the rank-breaking graph G_j are independently drawn. Precisely, for any choice of $M \in \mathbb{Z}^+$, we propose *order-M rank-breaking estimate*, which is the solution that maximizes the log-likelihood under the independence assumption:

$$\begin{aligned} \widehat{\theta} &\in \arg\max_{\theta \in \Omega_b} \mathcal{L}_{\mathrm{RB}}(\theta)\,, \text{ where} \\ \mathcal{L}_{\mathrm{RB}}(\theta) &= \sum_{j \in [n]} \sum_{e \in E_j : |T(e)| \leq M} \ln \mathbb{P}_\theta(e)\,. \end{aligned} \tag{4}$$

Due to independence assumption, we refer to it as pseudo-MLE. In a special case when $M = 1$, this can be transformed into the traditional pairwise rank-breaking, where (i) this is a concave maximization; (ii) the estimate is (asymptotically) unbiased and consistent as shown in Azari Soufiani et al. (2013, 2014); and (iii) the finite sample complexity have been analyzed in Khetan and Oh (2016). Although, this order-1 rank-breaking provides a significant gain in computational efficiency, the information contained in higher-order edges are unused, resulting in a significant loss in accuracy.

We provide the analyst the freedom to choose the computational complexity he/she is willing to tolerate. However, for general M, it has not been known if the optimization in (4) is tractable and/or if the solution is consistent. Since $\mathbb{P}_\theta(B(e) \prec T(e))$ as explicitly written in (3) is a sum of log-concave functions, it is not clear if the sum is also log-concave. Due to the ignored dependency in the formulation (4), it does not follow immediately that the resulting estimate is consistent.

We first establish that (4) is a concave maximization, Section 3.1. Though one can use any off-the-shelf convex maximization tool to compute $\widehat{\theta}$, we provide an efficient minorization-maximization (MM) algorithm for estimating $\widehat{\theta}$, Section 3.2. In Section 3.3, we show that the MM algorithm converges to the unique global optimal solution $\widehat{\theta}$ under the standard assumption given by Ford Jr. (1957) for pairwise comparisons. Under the same assumption, we show that the estimate $\widehat{\theta}$ is consistent, Section 3.4. In Section 3.5, we give the complete algorithm to compute $\widehat{\theta}$ using the proposed MM algorithm, given $\mathcal{G}_j$'s representing users' preferences. In Section 4 and Section 6, we provide a sharp analysis of the performance in the finite sample regime, characterizing the trade-off between computation and sample size, and verify the results from the numerical experiments.

3.1 Concavity of likelihood of rank-breaking graph

In the following, we show that likelihood of a hyper edge is log-concave for a family of Random Utility Models including the PL model.

Remark 1 *$\mathcal{L}_{\mathrm{RB}}(\theta)$ is concave in $\theta \in \mathbb{R}^d$.*

Proof Recall that $\mathbb{P}_\theta(B(e) \prec T(e))$ is the probability that an agent ranks the collection of items $T(e)$ above $B(e)$ when offered $S = B(e) \cup T(e)$. We want to show that $\mathbb{P}_\theta(B(e) \prec T(e))$ is log-concave under the PL model. We prove a slightly general result which works for a family of RUMs in the location family. RUM are defined as a probabilistic model where there is a real-valued utility parameter θ_i associated with each items $i \in S$, and an agent independently samples random utilities $\{U_i\}_{i \in S}$ for each item i with conditional distribution $\mu_i(\cdot|\theta_i)$. Then the ranking is obtained by sorting the items in decreasing order as per the observed random utilities U_i's. *Location family* is a subset of RUMs where the shapes of μ_i's are fixed and the only parameters are the means of the distributions. For location family, the noisy utilities can be written as $U_i = \theta_i + Z_i$ for i.i.d. random variable Z_i's. In particular, it is PL model when Z_i's follow the independent standard Gumbel distribution. We will show that for the location family if the probability density function for each Z_i's is log-concave then $\log \mathbb{P}_\theta(B(e) \prec T(e))$ is concave. The desired claim follows as the pdf of standard Gumbel distribution is log-concave. We use the following Theorem from Prékopa (1980). A similar technique was used to prove concavity when $|T(e)| = 1$ in Azari Soufiani et al. (2012).

Lemma 2 (Extension of Theorem 9 in Prékopa (1980)) *Suppose $g_1(\theta, Y), \cdots, g_r(\theta, Y)$ are concave functions in $\mathbb{R}^{2q}$, where $\theta, Y \in \mathbb{R}^q$, and Z is a $q-$component random vector whose probability distribution is logarithmic concave in $\mathbb{R}^q$, then the function*

$$h(\theta) = \mathbb{P}[g_1(\theta, Z) \geq 0, \cdots, g_r(\theta, Z) \geq 0], \qquad \text{for } \theta \in \mathbb{R}^q$$

is logarithmic concave on $\mathbb{R}^q$. Moreover, concavity is strict if the probability density function of Z is strictly logarithmic concave and $\theta \neq \tilde{\theta}$ implies $H(\theta) \neq H(\tilde{\theta})$. Where $H(\theta)$ is

$$H(\theta) \equiv \left\{ Y \;\middle|\; g_i(\theta, Y) \geq 0, \quad i = 1, \cdots, r \right\}.$$

Proof Theorem 9 in Prékopa (1980) proves concavity. The strict concavity follows from the fact that for a strictly logarithmic concave measure the following inequality is strict if $H(\theta) \neq H(\tilde{\theta})$.

$$\mathbb{P}[Z \in \lambda H(\theta) + (1-\lambda)H(\tilde{\theta})] \;\; \geq \;\; \mathbb{P}[Z \in \lambda H(\theta)]^{\lambda} \mathbb{P}[Z \in (1-\lambda)H(\tilde{\theta})]^{1-\lambda}\,,$$

where $\lambda \in (0,1)$. For a detailed proof, we refer the reader to the proof of Theorem 9 in Prékopa (1980). ■

To apply the above lemma to get concavity, let $q = |S|$, $r = 1$, $g_1(\theta, Y) = \min_{i \in T(e)}\{\theta_i + Y_i\} - \max_{i' \in B(e)}\{\theta_{i'} + Y_{i'}\}$. Observe that $g_1(\theta, Y)$ is concave in $\mathbb{R}^{2q}$, and $\mathbb{P}_\theta(B(e) \prec T(e)) = \mathbb{P}(g_1(\theta, Z) \geq 0)$. We use strict concavity part of the lemma in the subsequent section. ■

3.2 Minorization-maximization algorithm for pseudo-MLE of rank-breaking graph

We give a minorization-maximization algorithm for computing $\widehat{\theta}$ defined in (4). It is inspired from the MM algorithm given by Hunter (2004) for the case of pairwise comparisons and full-ranking. For any fixed parameter $\theta^{(t)} \in \mathbb{R}^d$, and a hyper edge e in a rank breaking graph G, define $Q(e, \theta; \theta^{(t)})$ as

$$Q(e,\theta;\theta^{(t)}) \;\; \equiv \;\; \sum_{\sigma \in \Lambda_{T(e)}} \left(\frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \sum_{u=1}^{|T(e)|} \left(\theta_{\sigma(u)} - \frac{\sum_{c'=u}^{|T(e)|} \exp\left(\theta_{\sigma(c')}\right) + \sum_{i \in B(e)} \exp\left(\theta_i\right)}{\sum_{c'=u}^{|T(e)|} \exp\left(\theta^{(t)}_{\sigma(c')}\right) + \sum_{i \in B(e)} \exp\left(\theta^{(t)}_i\right)} \right) \right)\,,$$

where $\mathbb{P}_\theta(e, \sigma)$ is defined such that $\mathbb{P}_\theta(e) = \sum_{\sigma \in \Lambda_{T(e)}} \mathbb{P}_\theta(e, \sigma)$. Recall from Equation (3) that $\Lambda_{T(e)}$ is the set of all rankings over $T(e)$.

$$\mathbb{P}_\theta(e,\sigma) \;\; \equiv \;\; \frac{\exp\left(\sum_{c=1}^{|T(e)|} \theta_{\sigma(c)}\right)}{\prod_{u=1}^{|T(e)|} \left(\sum_{c'=u}^{|T(e)|} \exp\left(\theta_{\sigma(c')}\right) + \sum_{i \in B(e)} \exp\left(\theta_i\right) \right)}.$$

We show that $Q(e, \theta; \theta^{(t)})$ minorizes $\ln(\mathbb{P}_\theta(e))$ at $\theta^{(t)}$. It is equal to $\ln(\mathbb{P}_\theta(e))$, up to a constant, if and only if $\theta^{(t)} = \theta$.

Lemma 3

$$Q(e,\theta;\theta^{(t)}) + f(e,\theta^{(t)}) \;\; \leq \;\; \ln(\mathbb{P}_\theta(e)) \quad \textit{with equality if } \theta = \theta^{(t)}\,,$$

where $f(e, \theta^{(t)})$ is a function of the hyper edge e and the parameter $\theta^{(t)}$, it does not depend upon θ.

We give a proof of the Lemma in Section 7.1. It follows that for any $Q(e, \theta; \theta^{(t)})$ satisfying minorizing condition in the above lemma,

$$Q(e,\theta;\theta^{(t)}) \;\geq\; Q(e,\theta^{(t)};\theta^{(t)}) \quad \text{implies} \quad \ln(\mathbb{P}_\theta(e)) \;\geq\; \ln(\mathbb{P}_{\theta^{(t)}}(e))\,. \tag{5}$$

Property (5) suggests an iterative algorithm in which we let $\theta^{(t)}$ be the parameter vector before the t-th iteration and define $\theta^{(t+1)}$ to be the maximizer of the $Q(e, \theta; \theta^{(t)})$. Since this algorithm consists of alternately creating a minorizing function $Q(e, \theta; \theta^{(t)})$ and then maximizing it, it is called an MM

algorithm (Hunter and Lange, 2000). To compute $\widehat{\theta}$ in (4), starting from an arbitrary initialization $\theta^{(1)}$, we estimate $\theta^{(t+1)}$ by maximizing

$$\theta^{(t+1)} \;\; = \;\; \arg\max_{\theta \in \mathbb{R}^d} \left\{ \sum_{j=1}^{n} \sum_{e \in E_j : |T(e)| \leq M} Q(e,\theta;\theta^{(t)}) \right\}.$$

Since the parameters $\{\theta_i\}_{i \in [d]}$ are separated in $Q(e, \theta; \theta^{(t)})$, its maximization can be explicitly accomplished as, for $i \in [d]$

$$e^{\theta_i^{(t+1)}} = \frac{N_i}{\sum_{j=1}^{n} \sum_{e \in E_j : |T(e)| \leq M} \sum_{\sigma \in \Lambda_{T(e)}} \frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \sum_{u=1}^{|T(e)|} \delta_{i,e,\sigma,u} \left(\sum_{c'=u}^{|T(e)|} \exp\left(\theta^{(t)}_{\sigma(c')}\right) + \sum_{i \in B(e)} \exp\left(\theta^{(t)}_i\right) \right)^{-1}}\,, \tag{6}$$

where N_i is the total number of hyper edges in which the i-th item is in the top set.

$$N_i \;\; = \;\; \sum_{j=1}^{n} \sum_{e \in E_j : |T(e)| \leq M} \mathbb{I}[i \in T(e)]\,.$$

$\delta_{i,e,\sigma,u}$ is the indicator variable defined as

$$\delta_{i,e,\sigma,u} = \begin{cases} 1\,, & \text{if } i \in \{T(e) \cup B(e)\} \text{ and } \sigma^{-1}(i) \geq u\,, \\ 0\,, & \text{otherwise.} \end{cases}$$

3.3 Convergence properties of the MM algorithm

In the following we show that $\lim_{t \to \infty} \theta^{(t)}$, (6), converges to the global optimal solution of the pseudo-likelihood objective given in (4) under standard assumption on the observed comparisons.

For pairwise comparisons, Ford Jr. (1957) noted that if it is possible to partition the set of items into two subsets A and B such that there are never any inter-set comparisons, then there is no basis for rating any item in set A with respect to any item in set B. On the other hand, if in all the inter-set comparisons, items from set A are preferred over the items in set B, then if all the parameters θ_i belonging to set A are doubled and the resulting vector θ renormalized, the likelihood must increase; thus the likelihood has no maximizer. The following assumption (Ford Jr., 1957) eliminates these possibilities.

Assumption 4 *In every possible partition of the items into two nonempty subsets, some item in the second set is preferred over some item in the first set at least once.*

Assumption 4 has a graph-theoretic interpretation. Suppose, items are denoted by nodes of a graph G and a directed edge (i, j) represents that the there is at least one user who prefers item i over item j. Then the Assumption 4 is equivalent to the statement that there is a path from i to j for all nodes i and j of the graph G. This assumption implies that there exists a unique maximizer of the log-likelihood function of pairwise comparisons.

In our setting, Assumption 4 makes sense if we interpret item i being preferred over item j to mean that there exists a hyper edge e such that the item i is in its top set $T(e)$ and the item j is in its bottom set $B(e)$. In the following theorem, we prove the convergence properties of the MM algorithm.

Theorem 5 *Under Assumption 4 the iterative minorization-maximization algorithm given in Equation* (6) *produces a sequence* $\theta^{(1)}, \theta^{(2)}, \cdots$ *guaranteed to converge to the unique estimate of the optimization problem given in Equation* (4).

Proof In general, it is always not possible to prove that the sequence of parameters $\theta^{(t)}$ defined by an MM algorithm converges at all. Nonetheless, it is often possible to obtain convergence results in specific cases. For pairwise comparisons, using property of stationary point, Ford Jr. (1957) showed that the MM algorithm converges to the unique maximum likelihood estimate under Assumption 4. Hunter (2004) established strict concavity of the likelihood function under Assumption 4 and proved the same result using the Liapounov's theorem. We follow the approach used by Hunter (2004). The following Liapounov's theorem guarantees that the MM algorithm converges to the stationary point of the pseudo log-likelihood objective (4). In Lemma 7, we show that the likelihood function (4) has a unique stationary point, namely the global maximizer. Which concludes that the MM algorithm converges to the unique global optimal solution irrespective of its starting point.

Lemma 6 (Liapounov's theorem) *Suppose* $M : \Omega \to \Omega$ *is continuous and* $\mathcal{L}_{\mathrm{RB}} : \Omega \to \mathbb{R}$ *is differentiable and for all* $\theta \in \Omega$ *we have* $\mathcal{L}_{\mathrm{RB}}(M(\theta)) \geq \mathcal{L}_{\mathrm{RB}}(\theta)$*, with equality only if* θ *is a stationary point of* $\mathcal{L}_{\mathrm{RB}}(\cdot)$*. Then, for arbitrary* $\theta^{(1)} \in \Omega$*, any limit point of the sequence* $\{\theta^{(t+1)} = M(\theta^{(t)})\}_{t\geq 1}$ *is a stationary point of* $\mathcal{L}_{\mathrm{RB}}(\theta)$.

Let $\Omega = \{\theta \in \mathbb{R}^d | \sum_{i\in[d]} \theta_i = 0\}$ be the parameter space Ω_b defined in (1) with $b = \infty$. Taking M to be the map implicitly defined by one iteration of the MM algorithm, we have $\mathcal{L}_{\mathrm{RB}}(M(\theta)) \geq \mathcal{L}_{\mathrm{RB}}(\theta)$ from (5). $\mathcal{L}_{\mathrm{RB}}(M(\theta)) = \mathcal{L}_{\mathrm{RB}}(\theta)$ implies that θ is a stationary point which follows from the fact that the differentiable minorizing function Q is a tangent to the log-likelihood $\mathcal{L}_{\mathrm{RB}}(\theta)$ at the current iterate $\theta^{(t)}$. Therefore, $\lim_{t\to\infty} \theta^{(t)}$, defined by the MM algorithm in (6) converges to the stationary point of the pseudo-likelihood objective (4). It remains to prove that $\mathcal{L}_{\mathrm{RB}}(\theta)$ has a unique stationary point, the global maximizer.

Lemma 7 *Under Assumption 4* $\mathcal{L}_{\mathrm{RB}}(\theta)$ *has a unique stationary point.*

Proof First, in Lemma 8, we show that $\mathcal{L}_{\mathrm{RB}}(\theta)$ is an upper compact function under the Assumption 4. $\mathcal{L}_{\mathrm{RB}}(\theta)$ is defined to be upper compact if, for any constant c, the set $\{\theta \in \Omega : \mathcal{L}_{\mathrm{RB}}(\theta) \geq c\}$ is a compact set of the parameter space Ω. Second, in Lemma 9, we show that $\mathcal{L}_{\mathrm{RB}}(\theta)$ is strictly concave. Since upper compactness implies the existence of at least one limit point and strict concavity implies the existence of at most one stationary point, we conclude that $\mathcal{L}_{\mathrm{RB}}(\theta)$ has a unique stationary point. ■

■

Lemma 8 $\mathcal{L}_{\mathrm{RB}}(\theta)$ *defined in Equation* (4) *is an upper compact function of* θ *under the Assumption 4.*

Proof We prove upper compactness following the idea of Hunter (2004). Consider what happens to $\mathcal{L}_{\mathrm{RB}}(\theta)$ when θ approaches the boundary of Ω. If $\tilde{\theta}$ lies on the boundary of Ω, then $\tilde{\theta}_i \to -\infty$ and $\tilde{\theta}_j \to \infty$ for some items i and j. If items are nodes of a directed graph in which edge (i, i') represent that there is at least one user who prefers i over i', then Assumption 4 implies that a directed path exists from i to j. Therefore, there must be some item a with $\tilde{\theta}_a \to -\infty$ which is preferred over item b with $\tilde{\theta}_b > C$, for some constant C. That is there exists an hyper edge e with $a \in T(e)$ and $b \in B(e)$. Which means that for $\theta \in \Omega$, taking limits in

$$\begin{aligned}\mathcal{L}_{\mathrm{RB}}(\theta) &\leq \ln \mathbb{P}_\theta(e) \\ &= \ln\left(\sum_{\sigma\in\Lambda_{T(e)}} \frac{\exp\left(\sum_{c=1}^{|T(e)|}\theta_{\sigma(c)}\right)}{\prod_{u=1}^{|T(e)|}\left(\sum_{c'=u}^{|T(e)|}\exp\left(\theta_{\sigma(c')}\right)+\sum_{i\in B(e)}\exp\left(\theta_i\right)\right)}\right)\end{aligned}$$

gives $\lim_{\theta\to\tilde{\theta}} \mathcal{L}_{\mathrm{RB}}(\theta) = -\infty$. Thus, for any given constant c, the set $\{\theta \in \Omega : \mathcal{L}_{\mathrm{RB}}(\theta) \geq c\}$ is a closed and bounded set, and hence a compact set. ■

Lemma 9 $\mathcal{L}_{\mathrm{RB}}(\theta)$ *defined in Equation* (4) *is strictly concave in* θ.

Proof To prove strict concavity, we use Lemma 2. Define $\widetilde{\Omega} = \{\theta \in \mathbb{R}^d | \theta_1 = 0\}$, a reparameterization of the set $\Omega = \{\theta \in \mathbb{R}^d | \sum_{i\in[d]} \theta_i = 0\}$. To apply Lemma 2 to prove strict concavity of log-likelihood of an hyper edge e, take $g_{ij}(\theta, Y) = (\theta_i + Y_i) - (\theta_j + Y_j)$, for all $i \in T(e)$ and $j \in B(e)$. Consider $\theta, \tilde{\theta} \in \widetilde{\Omega}$. $H(\theta) = H(\tilde{\theta})$ implies that $\theta_i - \theta_j = \tilde{\theta}_i - \tilde{\theta}_j$, for all $i \in T(e)$ and $j \in B(e)$. This follows from the fact that for a fixed parameter θ, the hyper planes $\{g_{ij}(\theta, Y) \geq 0\}_{ij}$ are linearly independent. Thus, Assumption 4 combined with the fact that $\theta_1 = \tilde{\theta}_1$ means that $\theta = \tilde{\theta}$. Since the Gumbel distribution has strictly logarithmic concave density function, we conclude that $\mathcal{L}_{\mathrm{RB}}(\theta)$ is strictly concave. ■

3.4 Consistency of pseudo-MLE of rank-breaking graph

In order to discuss consistency of the proposed approach, we need to specify how we sample the set of items to be offered S_j and also which partial ordering over S_j is to be observed. Here, we consider a simple but canonical scenario for sampling ordered relations, and show the proposed method is consistent for all non-degenerate cases. Later, in Section 4, we study a more general sampling scenario, when we analyze the order-M estimator in the finite sample regime.

We define a canonical sampling scenario in the following. There is a set of ℓ integers $(m_1, \ldots, m_\ell)$ whose sum is strictly less than d. A new arriving user is presented with all d items and is asked to provide her top m_1 items as an unordered set, and then the next m_2 items, and so on. This is sampling from the PL model and observing an ordered partition with $(\ell + 1)$ subsets of sizes m_a's, and the last subset includes all remaining items. We apply the generalized rank-breaking to get rank-breaking graphs $\{G_j\}$ with ℓ edges each, and order-M estimate is computed. We show that this is consistent, i.e. asymptotically unbiased in the limit of the number of users n.

Remark 10 *Under the* PL *model and the above sampling scenario, the order-*M *rank-breaking estimate* $\widehat{\theta}$ *in* (4) *is consistent for all choices of* $M \geq \min_{a\in\ell} m_a$.

Proof It is sufficient to show that (a) the estimate $\widehat{\theta}$, (4) is unique under the above sampling scenario, and (b) expectation of the gradient of $\mathcal{L}_{\mathrm{RB}}(\theta^*)$ is zero, i.e, $\mathbb{E}_{\theta^*}[\nabla \mathcal{L}_{\mathrm{RB}}(\theta^*)] = 0$, (Azari Soufiani et al., 2013). For the above sampling scenario in the limit of the number of users n, Assumption 4 is satisfied. Therefore, from Lemma 7, the estimate $\widehat{\theta}$, (4) is unique. In Lemma 13, we show that $\mathbb{E}_{\theta^*}[\nabla \mathcal{L}_{\mathrm{RB}}(\theta^*)] = 0$. We would like to mention that the Lemma 13 crucially relies on the memoryless property of the PL model.

■

3.5 Algorithm to estimate $\widehat{\theta}$ given DAG $\mathcal{G}_j$'s

Summarizing the rank-breaking approach explained in the previous sections, we give Algorithm 3, an algorithm to compute $\widehat{\theta}$, (4), an estimate of θ^*. Algorithm 3 takes as input DAG $\mathcal{G}_j$'s generated under PL model with parameter θ^*, rank-breaking order $M \in \mathbb{Z}^+$, a desired error threshold ϵ, and returns $\widehat{\theta}$.

Algorithm 3 Estimate θ^* given DAG $\mathcal{G}_j$'s.

Require: DAG $\{\mathcal{G}_j\}_{1\le j\le n}$ generated under PL model with parameter θ^*, rank-breaking order M, error threshold ϵ
Ensure: $\widehat{\theta}$ - an estimate of θ^*
find maximal ordered partitions $\{\mathcal{P}_j\}_{1\le j\le n}$ consistent with $\{\mathcal{G}_j\}_{1\le j\le n}$ [Algorithm 1]
construct rank breaking graph $\{G_j(S_j, E_j)\}_{1\le j\le n}$ from $\{\mathcal{P}_j\}_{1\le j\le n}$ [Algorithm 2]
$\widehat{\theta} \leftarrow \mathbf{0}_{d\times 1}$
repeat
  $\widetilde{\theta} \leftarrow \widehat{\theta}$
  for $i = 1$ to d **do**
    $\widehat{\theta}_i \leftarrow$ from minorizing maximizing Equation (6) using $\widetilde{\theta}$, $\{G_j(S_j, E_j)\}_{1\le j\le n}$, M
  end for
until $\|\widehat{\theta} - \widetilde{\theta}\|_\infty \le \epsilon$
return $\widehat{\theta}$

4. Analysis of the Algorithm

We first summarize the notations defined so far and introduce some new notations that are used in our theoretical results. We define a *comparison graph* that captures the topology of the offer sets S_j. Our upper and lower bounds both depend on the spectral properties of the comparison graph. Then, we present main theoretical analyses and numerical simulations confirming the theoretical results.

Notations. Following is a summary of all the notations defined above. Also, we introduce some new notations that are used in our theoretical results. We use n to denote the number of users providing partial rankings, indexed by $j \in [n]$ where $[n] = \{1, 2, \ldots, n\}$. We use d to denote the number of items, indexed by $i \in [d]$. Given rank-breaking graphs $\{G_j(S_j, E_j)\}_{j\in[n]}$ extracted from the DAGs $\{\mathcal{G}_j\}$, we first define the order M rank-breaking graphs $\{G_j^{(M)}(S_j, E_j^{(M)})\}$, where $E_j^{(M)}$ is a subset of E_j that includes only those edges $e_j \in E_j$ with $|T(e_j)| \le M$. This represents those edges that are included in the estimation for a choice of M. For finite sample analysis, the following quantities capture how the error depends on the topology of the data collected. Let $\kappa_j \equiv |S_j|$ and $\ell_j \equiv |E_j^{(M)}|$. We index each edge e_j in $E_j^{(M)}$ by $a \in [\ell_j]$ and define $m_{j,a} \equiv |T(e_{j,a})|$, size of top-set, for the a-th hyper edge of the j-th rank-breaking graph, and $r_{j,a} \equiv |T(e_{j,a})| + |B(e_{j,a})|$, sum of size of the top-set and the bottom-set. We let $p_j \equiv \sum_{a\in[\ell_j]} m_{j,a}$ denote the effective sample size for the observation $G_j^{(M)}$.

$$m_{j,a} \equiv |T(e_{j,a})|\,, \quad \text{size of top-set for the } e_{j,a} \text{ hyper edge of rank-breaking graph } G_j^{(M)}. \tag{7}$$

$$r_{j,a} \equiv |T(e_{j,a})| + |B(e_{j,a})|\,, \quad \text{sum of size of the top-set and the bottom-set for the } E_{j,a}. \tag{8}$$

$$p_j \equiv \sum_{a\in[\ell_j]} m_{j,a}\,, \quad \text{sum of size of all top-sets of } G_j^{(M)} \text{ (which are smaller than } M). \tag{9}$$

Notice that although we do not explicitly write the dependence on M, all of the above quantities implicitly depend on the choice of M. For ease of notations, we remove the superscript M from $G_j^{(M)}$ in the following.

For a ranking σ over S, i.e., σ is a mapping from $[|S|]$ to S, let σ^{-1} denote the inverse mapping. For a vector x, let $\|x\|_2$ denote the standard l_2 norm. Let $\mathbf{1}$ denote the all-ones vector and $\mathbf{0}$ denote the all-zeros vector with the appropriate dimension. Let $\mathcal{S}^d$ denote the set of $d \times d$ symmetric matrices with real-valued entries. For $X \in \mathcal{S}^d$, let $\lambda_1(X) \le \lambda_2(X) \le \cdots \le \lambda_d(X)$ denote eigenvalues of X sorted in increasing order. Let $\mathrm{Tr}(X) = \sum_{i=1}^d \lambda_i(X)$ denote trace of X and $\|X\| = \max\{|\lambda_1(X)|, |\lambda_d(X)|\}$ denote spectral norm of X. For two matrices $X, Y \in \mathcal{S}^d$, we write $X \succeq Y$ if $X - Y$ is positive semi-definite, i.e., $\lambda_1(X - Y) \ge 0$. Let e_i denote a unit vector in $\mathbb{R}^d$ along the i-th direction.

4.1 Comparison graph

We define a comparison graph $\mathcal{H}([d], E)$ as a weighted undirected graph with weights

$$A_{ii'} = \sum_{j\in[n]: i,i'\in S_j} \frac{p_j}{\kappa_j(\kappa_j - 1)}.$$

That is we put an edge (i, i') if there exists a user j whose offerings is a set S_j such that $i, i' \in S_j$. Define a diagonal matrix $D = \mathrm{diag}(A\mathbf{1})$, and the corresponding graph Laplacian $L = D - A$ such that

$$L \equiv \sum_{j=1}^n \frac{p_j}{\kappa_j(\kappa_j - 1)} \sum_{i<i'\in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top. \tag{10}$$

It is immediate that $\lambda_1(L) = 0$ with $\mathbf{1}$ as the eigenvector. There are remaining $d-1$ eigenvalues that sum to $\mathrm{Tr}(L) = \sum_j p_j$. The rescaled $\lambda_2(L)$ and $\lambda_d(L)$ capture the dependency on the topology:

$$\alpha \equiv \frac{\lambda_2(L)(d-1)}{\mathrm{Tr}(L)} \quad , \quad \beta \equiv \frac{\mathrm{Tr}(L)}{\lambda_d(L)(d-1)}. \tag{11}$$

In an ideal case where the graph is well connected, then the spectral gap of the Laplacian is large. The chosen rescaling ensures that if all the non-zero eigenvalues are of the same order then there

exists constants $0 \le c_1, c_2 \le 1$ such that $c_1 \le \alpha \le 1$ and $c_2 < \beta \le 1$. If the graph is connected then c_1 is strictly greater than zero. If $\lambda_2(L) = \cdots = \lambda_d(L)$ then $\alpha = \beta = 1$. We will show that the performance of our estimator depends upon topology of the comparison graph through these two parameters. The larger the rescaled spectral gap α the smaller the error we get with the same effective sample size. The rescaled largest eigenvalue β along with α determine how many samples are required for the analysis to hold. In general, α and β depend upon both the topology of the offer sets S_j and the topology of the rank-breaking graphs G_j, through the edge weights $A_{ii'}$. However, if topology of all the rank-breaking graphs G_j's is same then the comparison graph $\mathcal{H}$ and α, β depend only upon the topology of the offer sets S_j. For such a comparison graph, Khetan and Oh (2016) provides a detailed discussion on the spectral gap for various canonical graphs following the setup given in Shah et al. (2015a).

The concavity of $\mathcal{L}_{\mathrm{RB}}(\theta)$ also depends on the following quantities.

$$\gamma_1 \equiv \min_{j,a} \left\{ \left(\frac{r_{j,a} - m_{j,a}}{\kappa_j} \right)^{2e^{2b}-2} \right\}, \quad \gamma_2 \equiv \min_{j,a} \left\{ \left(\frac{r_{j,a} - m_{j,a}}{r_{j,a}} \right)^{2} \right\}. \tag{12}$$

The parameter γ_1 incorporates asymmetry in probabilities of items being ranked at different positions depending upon their weight θ^*_i. Recall that b is the upper bound on $\|\theta^*\|_\infty$, Equation 1. The parameter γ_1 is 1 for $b = 0$ that is when all the items have the same weight θ^*_i, and it decreases exponentially with increase in b. The exponential decrease is tight and reflects the fact that under PL model probability of the highest weight item being ranked last is exponentially smaller than its probability of being ranked first.

Suppose the number of items d grows to infinity and the rank-breaking graphs G_j's are determined such that size of the offered subsets κ_j's are increasing with d that is $\kappa_j = \Theta(d)$. If all the top-set sizes are much smaller than the size of the rank-breaking edge such that, $m_{j,a} = o(r_{j,a})$ and $r_{j,a} = \Theta(\kappa_j)$, then for $b = O(1)$, γ_1 can be made arbitrarily close to one, for large enough problem size d. On the other hand, when either $r_{j,a}$ is much smaller than κ_j or if $r_{j,a} = \Theta(\kappa_j)$ but $r_{j,a} - m_{j,a} = O(1)$ then γ_1 can be arbitrarily close to zero and accuracy of the parameter estimation can degrade significantly as stronger alternatives will have small chance of showing up in the bottom set. The value of γ_1 is quite sensitive to b. The parameter γ_2 controls the range of the size of the top-set with respect to the size of the bottom-set for which the error decays with the rate of 1/(size of the top-set). It does not depend upon the size of of the rank-breaking edge, $r_{j,a}$, in comparison to the offer set size κ_j. It only depends upon the size of the top sets $m_{j,a}$ in comparison to the size of the rank breaking edge $r_{j,a}$. If size of top sets $m_{j,a} = o(r_j, a)$ then γ_2 would be close to one. The dependence of accuracy on γ_1, γ_2 is demonstrated in simulations in Figure 5.

We define the following additional quantities that control our upper bound. The dependence in γ_3 and ν are due to weakness in the analysis, and ensures that the Hessian matrix is strictly negative definite.

$$\gamma_3 \equiv 1 - \max_{j,a} \left\{ \frac{4e^{16b}}{\gamma_1} \frac{m^2_{j,a} r^2_{j,a} \kappa^2_j}{(r_{j,a} - m_{j,a})^5} \right\}, \quad \nu \equiv \max_{j,a} \left\{ \frac{m_{j,a} \kappa^2_j}{(r_{j,a} - m_{j,a})^2} \right\}. \tag{13}$$

For our analysis to hold we need $\gamma_3 > 0$ which in addition to the conditions needed for γ_1 being close to one require that $m_{j,a} \le c\sqrt{r_{j,a}}$, for a sufficiently small positive constant c. We believe this is a limitation on our analysis and the results should hold for any values of $m_{j,a} = o(r_{j,a})$. For

the special case when $m_{j,a} \le 3$ for all j, a, we provide a tighter result that does not depend upon γ_3. However, in general getting rid of γ_3 is challenging. ν shows up in the number of samples required for our analysis to hold. Note that the quantities defined in this section implicitly depend on the choice of M, which controls the necessary computational power, via the definition of the rank-breaking graphs $\{G_j^{(M)}\}_{j \in [n]}$.

4.2 Upper bound on the achievable error

We provide an upper bound on the error for the order-M rank-breaking Algorithm 3, showing the explicit dependence on the topology of the offered sets $\{S_j\}_{j \in [n]}$. Recall from the sampling assumptions in Section 3 that we assume the topology of the observed DAG $\mathcal{G}_j$'s, and the rank-breaking order M is fixed apriori. The randomness that we observe is in the position of S_j items in the DAG $\mathcal{G}_j$. For an hyper edge $e \in E_j$, the randomness is in which items of the set S_j appear in the bottom $|B(e)|$ positions and the bottom $|T(e)| + |B(e)|$ positions in the preference order of the user j. This precisely captures the randomness due to the PL model in the observed DAG $\mathcal{G}_j$. The following theorem provides an upper bound on the achieved error, and a proof is provided in Section 7.

Theorem 11 *Suppose there are n users, d items parametrized by $\theta^* \in \Omega_b$, and each user $j \in [n]$ is presented with a set of offerings $S_j \subseteq [d]$ and the user provides a partial ordering under the PL model consistent with the topology of the apriori fixed DAG $\mathcal{G}_j$. For a choice of $M \in \mathbb{Z}^+$, if $\gamma_3 > 0$ and the effective sample size $\sum_{j=1}^n p_j$ is large enough such that*

$$\sum_{j=1}^{n} p_j \ge \frac{2^{14} e^{20b} \nu^2}{(\alpha \gamma_1 \gamma_2 \gamma_3)^2 \beta} \frac{p_{\max}}{\kappa_{\min}} d \log d \,, \tag{14}$$

where $b \equiv \max_i |\theta^*_i|$ is the dynamic range, $p_{\max} = \max_{j \in [n]} p_j$, $\kappa_{\min} = \min_{j \in [n]} \kappa_j$, α is the (rescaled) spectral gap, β is the (rescaled) spectral radius in (11)*, and γ_1, γ_2, γ_3, and ν are defined in* (12) *and* (13)*, then the generalized rank-breaking estimator in* (4) *achieves*

$$\frac{1}{\sqrt{d}} \|\widehat{\theta} - \theta^*\|_2 \le \frac{40 e^{7b}}{\alpha \gamma_1 \gamma_2^{3/2} \gamma_3} \sqrt{\frac{d \log d}{\sum_{j=1}^n p_j}} \,, \tag{15}$$

with probability at least $1 - 3e^3 d^{-3}$. Moreover, for $M \le 3$ the above bound holds with γ_3 replaced by one, giving a tighter result.

Note that the dependence on the choice of M is not explicit in the bound, but rather is implicit in the construction of the comparison graph and the number of effective samples.

Suppose the number of items d is large enough and the size of the offered subsets κ_j's and the size of the rank breaking edges $r_{j,a}$'s are increasing with d, that is there exists positive constants c_1, c_2 such that $\kappa_j \ge c_1 d$, and $r_{j,a} \ge c_2 \kappa_j$. Then for $b = O(1)$ there exists a universal constant c_3 such that if top-set sizes $m_{j,a} \le c_3 (r_{j,a})^{1/3}$ then there exists constants $0 < c_4, c_5, c_6 \le 1$ such that $c_4 \le \gamma_1 < 1$, $c_5 \le \gamma_2 < 1$ and $c_6 \le \gamma_3 < 1$. Further, if the comparison graph $\mathcal{H}$ is well connected then there exists a constant $0 < c_7, c_8 \le 1$ such that the rescaled spectral gap $c_7 \le \alpha \le 1$ and rescaled largest eigenvalue $c_8 \le \beta \le 1$. In this ideal case, the condition on the effective sample size is met with $\sum_j p_j = O(d \log d)$, (14). Therefore the effective sample size $\sum_{j=1}^n p_j = \Omega(d \log d)$ is sufficient to

ensure $\|\widehat{\theta}-\theta^*\|_2 = o(\sqrt{d})$ which is only a logarithmic factor larger than the number of parameters. We need $m_{j,a} \le c_3(r_{j,a})^{1/3}$ to satisfy $(\nu^2 p_{\max})/\kappa_{\min} = O(1)$, otherwise $m_{j,a} \le c_3(r_{j,a})^{1/2}$ is sufficient to ensure $\gamma_3 > 0$. We believe that dependence in γ_3 is weakness of our analysis and there is no dependence as long as $m_{j,a} < r_{j,a}$. For, rank-breaking order $M \le 3$, we are able to give tighter results where there is no dependence on γ_3.

As explained above, in the ideal case, for large enough problem size d, there exists a positive constant C such that $\|\widehat{\theta}-\theta^*\|_2^2 \le Cd^2 \log d/(\sum_{j=1}^n p_j)$. Recall from the construction of the likelihood objective function, $\mathcal{L}_{\mathrm{RB}}(\theta) = \sum_{j\in[n]} \sum_{e\in E_j:|T(e)|\le M} \ln \mathbb{P}_\theta(e)$. If we fix all the problem parameters including topology of the DAG $\mathcal{G}_j$'s and increase M then $p_j = \sum_{a\in[\ell_j]} m_{j,a}$ increases. Therefore, by increasing M we can get the same number of effective samples $\sum_{j=1}^n p_j$ with smaller number of rankings n. However, increasing M increases computational complexity as $M!$. Therefore, to achieve a fixed target accuracy $\|\widehat{\theta}-\theta^*\|_2$, an analyst can trade-off the required number of rankings with the budgeted computational complexity.

If the DAG $\mathcal{G}_j$'s are complete graph that is each user provides a full ranking over the offered subset S_j, we get $m_{j,a} = 1$, $\ell_j = \kappa_j - 1$, and the total effective sample size $\sum_j p_j = \sum_{j\in[n]}(\kappa_j - 1)$. Therefore, from the above theorem, $\sum_{j\in[n]}(\kappa_j - 1) = \Omega(d \log d)$ is sufficient to ensure $\|\widehat{\theta}-\theta^*\|_2 = o(\sqrt{d})$. It matches with the results for full rankings given in Hajek et al. (2014); Khetan and Oh (2016).

Unordered vs. ordered top-m ranking. In the ideal case, a perhaps surprising observation is that, for a ranking j, sizes of the top-sets $\{m_{j,a}\}_{a\in[\ell_j]}$ impacts estimation accuracy only via $p_j = \sum_{a\in[\ell_j]} m_{j,a}$, when $m_{j,a}$'s are sufficiently small in comparison to $r_{j,a}$'s, sum of the top-set size and the bottom-set size. In particular, for estimation accuracy it does not matter whether users reveal their top-m choices in the ordered way $\{i_1\} \succ \{i_2\} \succ \cdots \succ \{i_m\} \succ \{i_{m+1}, \cdots, i_k\}$ or the unorderd way $\{i_1, i_2, \cdots, i_m\} \succ \{i_{m+1}, \cdots i_k\}$, when m is sufficiently small in comparison to k. Numerical results in Figure 5 confirm this.

Proof idea. The analysis of the optimization in (4) shows that, with high probability, $\mathcal{L}_{\mathrm{RB}}(\theta)$ is strictly concave with $\lambda_2(H(\theta)) \le -C_b\gamma_1\gamma_2\gamma_3\lambda_2(L) < 0$ for all $\theta \in \Omega_b$ (Lemma 15), and the gradient is also bounded with $\|\nabla\mathcal{L}_{\mathrm{RB}}(\theta^*)\| \le C_b'\gamma_2^{-1/2}(\sum_j p_j \log d)^{1/2}$ (Lemma 14). This leads to Theorem 11:

$$\|\widehat{\theta}-\theta^*\|_2 \;\le\; \frac{2\|\nabla\mathcal{L}_{\mathrm{RB}}(\theta^*)\|}{-\lambda_2(H(\theta))} \;\le\; C_b'' \frac{\sqrt{\sum_j p_j \log d}}{\gamma_1\gamma_2^{3/2}\gamma_3\lambda_2(L)}\,,$$

where C_b, C_b', and C_b'' are constants that only depend on b, and $\lambda_2(H(\theta))$ is the second largest eigenvalue of a negative semidefinite Hessian matrix $H(\theta)$ of $\mathcal{L}_{\mathrm{RB}}(\theta)$. Recall that $\theta^\top \mathbf{1} = 0$ since we restrict our search in Ω_b. Hence, the error depends on $\lambda_2(H(\theta))$ instead of $\lambda_1(H(\theta))$ whose corresponding eigenvector is the all-ones vector.

4.3 Lower bound on computationally unbounded estimators

Suppose $M = d$. We prove a fundamental lower bound on the achievable error rate that holds for any *unbiased* estimator with no restrictions on the computational complexity. For each (j,a),

define $\eta_{j,a}$ as

$$\eta_{j,a} \;\equiv\; \sum_{u=0}^{m_{j,a}-1} \Big(\frac{1}{r_{j,a}-u} + \frac{u(m_{j,a}-u)}{m_{j,a}(r_{j,a}-u)^2}\Big) + \sum_{u<u'\in[m_{j,a}-1]} \frac{2u}{m_{j,a}(r_{j,a}-u)} \frac{m_{j,a}-u'}{r_{j,a}-u'} \tag{16}$$

$$< \sum_{u=0}^{m_{j,a}-1} \Big(\frac{1}{m_{j,a}-u} + \frac{u}{m_{j,a}(m_{j,a}-u)}\Big) + \sum_{u<u'\in[m_{j,a}-1]} \frac{2u}{m_{j,a}(m_{j,a}-u)} \tag{17}$$

$$= \sum_{u=0}^{m_{j,a}-1} \Big(\frac{1}{m_{j,a}-u} + \frac{u}{m_{j,a}(m_{j,a}-u)} + \frac{2u(m_{j,a}-1-u)}{m_{j,a}(m_{j,a}-u)}\Big) = m_{j,a}\,,$$

where (17) follows from the fact that (16) is monotonically strictly decreasing in $r_{j,a}$ for $r_{j,a} \ge m_{j,a}$. Since by definition $r_{j,a} > m_{j,a}$, we substitute $r_{j,a} = m_{j,a}$ to get a strict upper bound.

Theorem 12 *Let $\mathcal{U}$ denote the set of all unbiased estimators of θ^* that are centered such that $\widehat{\theta}\mathbf{1} = 0$, and let $\mu = \max_{j\in[n],a\in[\ell_j]}\{m_{j,a} - \eta_{j,a}\}$. For all $b > 0$,*

$$\inf_{\widehat{\theta}\in\mathcal{U}} \sup_{\theta^*\in\Omega_b} \mathbb{E}[\|\widehat{\theta}-\theta^*\|^2] \;\ge\; \max\left\{ \frac{(d-1)^2}{\sum_{j=1}^n \sum_{a=1}^{\ell_j}(m_{j,a}-\eta_{j,a})}\,,\; \frac{1}{\mu}\sum_{i=2}^{d}\frac{1}{\lambda_i(L)} \right\}. \tag{18}$$

The proof relies on the Cramer-Rao bound and is provided in Section 7.6. Since $0 < \eta_{j,a} < m_{j,a}$, the mean squared error is lower bounded by $(d-1)^2/(\sum_{j=1}^n \sum_{a=1}^{\ell_j} m_{j,a}) = (d-1)^2/(\sum_{j=1}^n p_j)$, where $\sum_{j=1}^n p_j$ is the effective sample size. Comparing it to the upper bound in (15), this is tight up to a logarithmic factor when (a) the topology of the data is well-behaved such that all the quantities $\gamma_1, \gamma_2, \gamma_3, \alpha, \beta$ are greater than a positive constant $c \le 1$; and (b) there is no limit on the computational power and M can be made as large as we need. For full-rankings, this bound reduces to the one given in Hajek et al. (2014); Khetan and Oh (2016). For full rankings, $\sum_{a=1}^{\ell_j}(m_{j,a} - \eta_{j,a}) = (\kappa_j - 1)^2/\kappa_j$.

The bound in Eq. (18) further gives a tighter lower bound, capturing the dependency in $\eta_{j,a}$'s and $\lambda_i(L)$'s. The second term in (18) implies we get a tighter bound when $\lambda_2(L)$ is smaller. If the comparison graph $\mathcal{H}$ is disconnected that is $\lambda_2(L) = 0$, the bound shows that θ^* can not be estimated.

To understand the impact of $\eta_{j,a}$ on MSE, we plot $(m_{j,a} - \eta_{j,a})/r_{j,a}$ as a function of $m_{j,a}/r_{j,a}$ for different values of $r_{j,a}$ in Figure 3. Recall that $m_{j,a}$ is the size of the top-set, (7) and $r_{j,a}$ is the sum of size of the top-set and the bottom-set, (8). We vary $m_{j,a}$ from 1 to $r_{j,a} - 1$, for $r_{j,a}$ in $\{2, 4, 8, 16, 32, 256, 1024\}$. From the Theorem 12, contribution of an hyper edge $e_{j,a}$ to the effective samples is $(m_{j,a} - \eta_{j,a})$. Since $\eta_{j,a}$ increases with $m_{j,a}$, a natural question is what is the optimal value of $m_{j,a}$ that gives the smallest MSE, for a fixed $r_{j,a}$. Figure 3 shows that $(m_{j,a} - \eta_{j,a})/r_{j,a}$ achieves its maximum value at $m/r \approx 0.8$ when r is sufficiently large. It also shows that $(m_{j,a} - \eta_{j,a}) \ge c\, m_{j,a}$, for $m_{j,a}/r_{j,a} \le c_1 (\approx 0.8)$, for positive constants $c, c_1 < 1$, when $r_{j,a}$ is large. That is the contribution of an hyper edge $e_{j,a}$ to the effective sample size is at least $c\, m_{j,a}$ for $m_{j,a}/r_{j,a} \le c_1$. Comparing this with the lower bound for top-$m_{j,a}$ ranking given in Khetan and Oh (2016), it can be concluded that the (unobserved) relative ordering among the items in the top-set of the hyper edge $e_{j,a}$ has limited impact on the MSE. Khetan and Oh (2016) show in the their lower bound that the contribution of top-m ranking on the effective sample size is m.

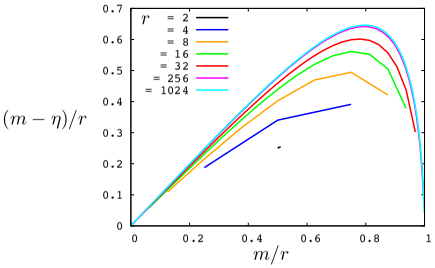


Figure 3: It shows how η varies as a function of m, size of the top-set, for a fixed value of r, sum of top-set and the bottom-set sizes, Equation (16).

Note that the lower bound is derived for the easiest case, $b = 0$, when all the items $i \in [d]$ have the same weight $\theta^*_i = \theta^*_0$. Therefore, the above conclusion that the relative ordering among the items in the top-set of the hyper edge $e_{j,a}$ has limited impact on the accuracy can be made only for this case when all the items have the same PL weight. However, the upper bound shows that this conclusion holds true in general. The 'unordered vs. ordered top-m ranking' paragraph in the previous section explains that for the ideal case when $m_{j,a}$ is sufficiently small in comparison to $r_{j,a}$, the relative ordering has limited impact.

Recall that in the upper bound, γ_1 and γ_2 capture the impact of $m_{j,a}/r_{j,a}$ on the effective number of samples. However, for $b = 0$, $\gamma_1 = 1$, and for $b > 0$ it captures asymmetry in the probability of the highest weight item appearing in bottom set. $\gamma_2 = \min_{j,a} \left\{ \left(\frac{r_{j,a} - m_{j,a}}{r_{j,a}} \right)^2 \right\}$ captures the role played by $\eta_{j,a}$ in the lower bound.

5. Numerical results

We provide extensive numerical results on simulated and real-world datasets confirming our theoretical results and performance gains of the generalized rank-breaking algorithm over the pairwise rank-breaking algorithm.

5.1 Simulated datasets

In the following, we give numerical results confirming our theoretical results. Our numerical experiments show that the dependence of MSE on $n, d, \kappa_j, r_{j,a}, m_{j,a}, \ell_j$ as given in Theorem 11, Equation (15) holds true, even when the conditions for the theorem to hold are not met. For the theorem to hold, it is required that the number of items d, the set sizes κ_j and the hyper edge sizes $r_{j,a}$ are sufficiently large such that $\gamma_3 > 0$ and the number of effective samples satisfies (14). However, in all our experiments the number of items $d <= 512$ and $b = 2$, therefore from (13) $\gamma_3 < 0$, and also the condition in (14) is not met.

In particular, in our numerical setting $\gamma_1 =$

Impact of the number of independent rankings n and the number of rank-breaking hyper edges ℓ_j on accuracy. Figure 4 (first panel) shows the accuracy-sample tradeoff for

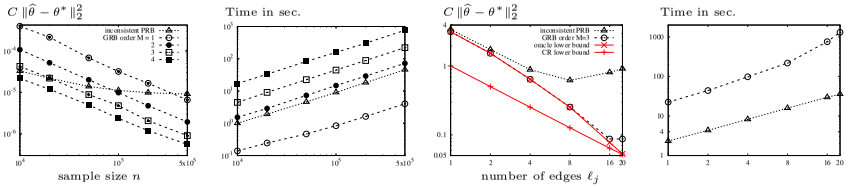


Figure 4: On the first panel, we fix $d = 256$, $\kappa = 32$, $\ell = 4$, $m_a = a$ for $a \in \{1, 2, 3, 4\}$, and sample posets from the canonical scenario explained in Section 3.4. On the third panel, we let $n = 10^5$, $d = 512$, $\kappa = 64$, $m_a = 3$ for all $a \in [\ell]$ and vary $\ell \in \{1, 2, 4, 8, 16\}$. The second and the fourth panel show the computation time for the first and the third panel respectively. The PL weights are chosen uniformly spaced over $[-2, 2]$. Smaller error is achieved when using more computational resources with larger M and using all paired comparisons results in an inconsistent Pairwise Rank-Breaking (PRB) whose error does not vanish with sample size (first panel). Generalized Rank-Breaking (GRB) utilizes all the observations achieving the oracle lower bound (third panel).

increasing computation M on the same data. As predicted by the anlaysis, generalized rank-breaking (GRB) is consistent (Remark 10) and the mean square error (MSE) decays at the rate $(1/n)$, and decreases with increase in M, order of rank-breaking (Theorem 11). For comparison, we also plot the MSE achieved by pairwise rank-breaking (PRB) approach where we include all paired relations derived from data, which we call *inconsistent PRB*. As predicted by Azari Soufiani et al. (2014), this results in an inconsistent estimate, whose MSE does not vanish as we increase the sample size. Notice that including all paired comparisons increases bias, but also decreases variance of the estimate. Hence, when sample size is limited and variance is dominating the bias, it is actually beneficial to include those biased paired relations to gain in variance at the cost of increased bias. Theoretical analysis of such a bias-variance tradeoff is outside the scope of this paper, but proposes an interesting research direction.

In the third panel, the GRB with $M = 3$ achieves decreasing MSE, whereas for PRB the increased bias dominates the MSE. For comparisons, we provide the error achieved by an oracle estimator who knows the exact ordering among those items belonging to the top-sets and runs MLE. For example, if $\ell = 2$, the GRB observes an ordering $(\{i_1, i_2, i_4, i_5, \ldots\} \prec \{i_{17}, i_3, i_6\} \prec \{i_9, i_2, i_{11}\})$ whereas the oracle estimator has extra information on the ordering among those top sets, i.e. $(\{i_1, i_2, i_4, i_5, \ldots\} \prec i_{17} \prec i_3 \prec i_6 \prec i_9 \prec i_2 \prec i_{11}\})$. Perhaps surprisingly, GRB is able to achieve a similar performance without this significant extra information, unless ℓ is large. The performance degradation in large ℓ regime stems from the fact that the ratio of m_a and r_a approaches 1 for a close to ℓ when ℓ is large. Therefore the parameters γ_1 and γ_2 become small, and the upper bound MSE increases consequentially. The normalization constant C is $1/d^2$ for the first panel and nm/d^2 for the third panel. All the numerical results in this paper are averaged over 10 instances. Standard error is very small in all the results, therefore we do not give error bars, except in the first panel in Figure 4.

Impact of the top-set size m and the set-size κ on accuracy. In Figure 5, the first and the third panel, we compare performance of our algorithm with pairwise breaking, Cramer Rao

lower bound and oracle MLE lower bound. Oracle MLE knows relative ordering of items in the top-sets $T(e)$ and hence is strictly better than the GRB. For the settings chosen, Oracle MLE gets the ordered ranking of top-m items whereas GRB gets unordered top-m items. As predicted by our analysis, GRB matches with the oracle MLE which means relative ordering of top-m items among themselves is statistically insignificant when m is sufficiently small in comparison to $r = \kappa$. For $r = \kappa = 32$ in the first panel, MSE decays as m increases from 1 to 5. However, when $r = \kappa = 16$ in the third panel, for the same increase of m from 1 to 5 MSE starts increasing when m grows beyond 4. The reason is that the quantities γ_1 and γ_2 get smaller as m increases, and the upper bound increases consequently. The normalization constant C is n/d^2 for these two panels.

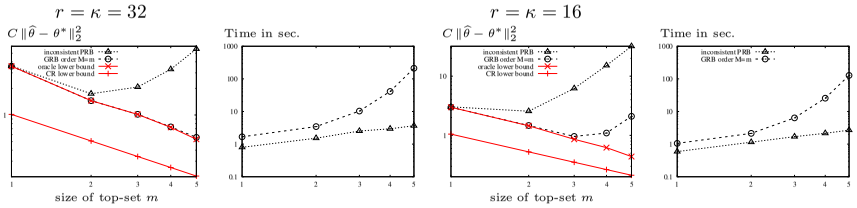


Figure 5: PRB: pairwise rank-breaking, GRB: generalized rank-breaking. θ^* is chosen uniformly spaced over $[-2, 2]$ and $d = 512$, $n = 10^5$ and number of hyperedges $\ell = 1$. The second and the fourth panel show the computation time for the first and the third panel respectively. MSE decreases as m increases when r, sum of the size of the top-set and the bottom-set is sufficiently large (first panel). When r is small, with increase in m MSE initially decreases but as m grows large MSE starts increasing (third panel).

Impact of the dynamic range b on accuracy. In Figure 6, we show the impact of b and $r = \kappa$ on the accuracy for fixed $m = 4$. When κ is small, γ_2 is small, and hence error is large; when b is large γ_1 is exponentially small, and hence error is significantly large. This is different from learning Mallows models in Ali and Meilă (2012) where peaked distributions are easier to learn, and is related to the fact that we are not only interested in recovering the (ordinal) ranking but also the (cardinal) weight. The normalization constant C is nm/d^2.

5.2 Real-world datasets

On sushi preferences (Kamishima, 2003) and jester dataset (Goldberg et al., 2001), we improve over pairwise breaking and achieve same performance as the oracle MLE.

Sushi dataset. There are $d = 100$ types of sushi. Full rankings over subsets S_j of size $\kappa = 10$ are provided by $n = 5000$ individuals. The offering subsets S_j are chosen uniformly at random from the entire set d. We set the ground truth θ^* to be the MLE of the PL weights over the entire data. In the left panel of Figure 7, for each $m \in \{3, 4, 5, 6\}$, we remove the known ordering among the top-m and bottom-$(10 - m)$ sushi in each set, and run our estimator with one rank-breaking hyper edge between top-m and bottom-$(10 - m)$ items. We compare our algorithm with inconsistent pairwise breaking (using optimal choice of parameters from Khetan and Oh (2016)) and the oracle MLE. For $m \leq 6$, the proposed rank-breaking performs as good as the oracle who knows the relative

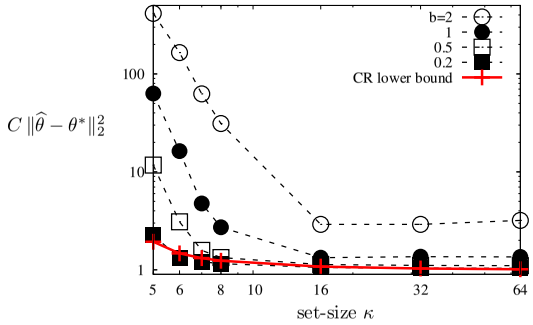


Figure 6: $d = 512$, $n = 10^5$ and θ^* is chosen uniformly spaced over $[-2, 2]$. Number of hyper edges $\ell = 1$ with $r = \kappa$ and $m = 4$. MSE increases as the dynamic range b gets large.

ordering among the top m items. In other words, an individual providing a set of ordered top-6 sushi or a set of unordered top-6 sushi statistically reveals the same information, for the purpose of estimating the ground truth parameters. As predicted by our theory, error decreases with increase in top-set size m.

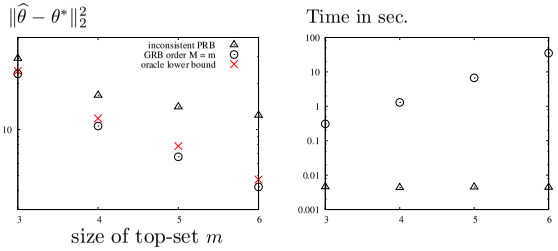


Figure 7: The sushi dataset has $d = 100$, $n = 5000$, and $\kappa = 10$. The right panel shows computation time. Generalized rank-breaking improves over pairwise RB and performs as good as oracle MLE on sushi dataset.

Jester dataset. It consists of continuous ratings between -10 to $+10$ of 100 jokes on sets of size κ, $36 \leq \kappa \leq 100$, by 24,983 users. We convert cardinal ratings into ordinal full rankings. The ground truth θ^* is set to be the MLE of the PL weights over the entire data. For $m \in \{2, 3, 4, 5\}$, we convert each full ranking into a poset that has $\ell = \lfloor \kappa/m \rfloor$ partitions of size m, by removing known relative ordering from each partition. The leads to total number of effective samples $\sum_j p_j = \sum_j \sum_{a \in [\ell_j]} m_{j,a} = \sum_j (\kappa_j - m)$, which is approximately equal for each $m \in \{2, 3, 4, 5\}$. However, with increasing m, the quantities $\gamma_1, \gamma_2, \gamma_3$ become smaller and hence the error increases (third panel in Figure 8). Figure 8 compares the three algorithms for two different settings. In the first

panel, we fix $m = 4$ and vary the number of samples n. Mean square error decreases with increase in the number of samples. In the third panel, we use $n = 5000$ samples, and vary $m \in \{2, 3, 4, 5\}$.

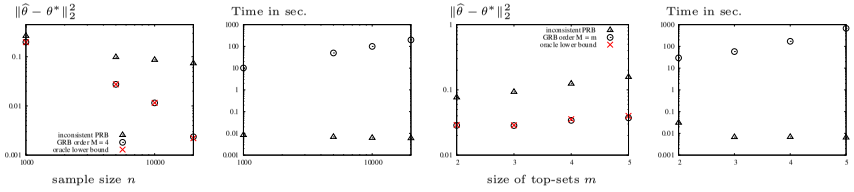


Figure 8: The jester dataset which has $d = 100$, $n = 24,983$, and $36 \leq \kappa_j \leq 100$. The second and the fourth panel show the computation time for the first and the third panel respectively. Generalized rank-breaking improves over pairwise RB and performs as good as oracle MLE on jester dataset.

6. Computational and statistical tradeoff

For estimators with limited computational power, however, the lower bound Theorem 12 fails to capture the dependency on the allowed computational power. Understanding such fundamental trade-offs is a challenging problem, which has been studied only in a few special cases, e.g. planted clique problem (Deshpande and Montanari, 2015; Meka et al., 2015). This is outside the scope of this paper, and we instead investigate the trade-off achieved by the proposed rank-breaking approach. When we are limited on computational power, Theorem 11 implicitly captures this dependence when order-M rank-breaking is used. The dependence is captured indirectly via the resulting rank-breaking $\{G_{j,a}\}_{j\in[n],a\in[\ell_j]}$ and the topology of it. We make this trade-off explicit by considering a simple but canonical example. Suppose $\theta^* \in \Omega_b$ with $b = O(1)$. Each user gives an i.i.d. partial ranking, where all items are offered and the partial ranking is based on an ordered partition with $\ell_j = \lfloor \sqrt{2c}d^{1/3} \rfloor$ subsets for a constant c. The top subset has size $m_{j,1} = 1$, and the a-th subset has size $m_{j,a} = a$, up to $a < \ell_j$. The choice of ℓ_j with a sufficiently small constant c ensures that all the conditions of the ideal case explained in the previous section for holding the Theorem 11 are satisfied.

Computation. For a choice of M such that $M \leq \ell_j - 1$, we consider the computational complexity in computing $\theta^{(t)}$, (6) in one iteration of the minorization-maximization algorithm, which scales as $T(M, n) = O(M! \times dn)$. A detailed analysis of the convergence rate of the MM algorithm is outside the scope of this paper.

Accuracy. Under the canonical setting, for $M \leq \ell_j - 1$, Laplacian matrix L of the comparison graph $\mathcal{H}$ is $L = nM(M+1)/(2d(d-1))\big(d\mathbb{I} - \mathbf{1}\mathbf{1}^\top\big)$. All the non-zero eigenvalues of this complete graph are equal, $\lambda_2(L) = \cdots = \lambda_d(L) = \mathrm{Tr}(L)/(d-1)$. Therefore, the resclaed spectral gap $\alpha = 1$, and the rescaled largest eigenvalue $\beta = 1$. Since the effective sample size is $\sum_{j,a} m_{j,a}\mathbb{I}\{m_{j,a} \leq M\} = nM(M+1)/2$, it follows from Theorem 11 that the (rescaled) root mean squared error is $O(\sqrt{(d\log d)/(nM^2)})$. In order to achieve a smaller target error rate of ε for a fixed problem size d, an analyst can increase the rank-breaking order M and/or increase n that is collect more i.i.d. rankings. Fixing the rank-breaking order M, we need to collect $n = \Omega((d\log d)/(\varepsilon^2 M^2))$ i.i.d. rankings. The resulting trade-off between run-time and root mean squared error ε is $T(\varepsilon) \propto (M!(d^2\log d)/(\varepsilon^2 M^2)$. The computational complexity is quadratic in the target error ϵ, when we can collect more rankings. On the other hand, fixing the number of rankings n, we need to choose $M = \Omega((1/\varepsilon)\sqrt{(d\log d)/n})$. The resulting trade-off between run-time and root mean squared error ε is $T(\epsilon) \propto (\lceil(1/\varepsilon)\sqrt{(d\log d)/n}\rceil)!dn$. The computational complexity is super exponential in the target error ϵ, for a fixed problem size d and the number of rankings n. Super exponential complexity is unavoidable as computing likelihood is super exponential in M. However, our approach provides flexibility to the analyst to choose between collecting more rankings n or increasing the rank-breaking order M to achieve the desired target error. We show numerical experiment under this canonical setting in Figure 4 (left) with $d = 256$ and $M \in \{1, 2, 3, 4, 5\}$, illustrating the trade-off in practice.

7. Proofs

We provide the proofs of the main results.

7.1 Proof of Lemma 3

In the following, we show that $Q(e, \theta; \theta^{(t)})$ minorizes $\ln(\mathbb{P}_\theta(e))$ at $\theta^{(t)}$. Using Jensen's inequality $\ln(\mathbb{E}[X]) \geq \mathbb{E}[\ln(X)]$, for any given parameter $\theta^{(t)} \in \mathbb{R}^d$, we have,

$$
\begin{aligned}
& \ln(\mathbb{P}_\theta(e)) \\
= \; & \ln\big(\mathbb{P}_\theta\big(B(e) \prec T(e)\big)\big) \\
= \; & \ln\left(\sum_{\sigma\in\Lambda_{T(e)}} \frac{\exp\left(\sum_{c=1}^{|T(e)|}\theta_{\sigma(c)}\right)}{\prod_{u=1}^{|T(e)|}\left(\sum_{c'=u}^{|T(e)|}\exp\left(\theta_{\sigma(c')}\right) + \sum_{i\in B(e)}\exp\left(\theta_i\right)\right)}\right) \\
\geq \; & \sum_{\sigma\in\Lambda_{T(e)}} \frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \ln\left(\frac{\exp\left(\sum_{c=1}^{|T(e)|}\theta_{\sigma(c)}\right)}{\prod_{u=1}^{|T(e)|}\left(\sum_{c'=u}^{|T(e)|}\exp\left(\theta_{\sigma(c')}\right) + \sum_{i\in B(e)}\exp\left(\theta_i\right)\right)} \frac{\mathbb{P}_{\theta^{(t)}}(e)}{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}\right) \qquad (19) \\
= \; & \sum_{\sigma\in\Lambda_{T(e)}} \frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \sum_{u=1}^{|T(e)|}\left(\theta_{\sigma(u)} - \ln\left(\sum_{c'=u}^{|T(e)|}\exp\left(\theta_{\sigma(c')}\right) + \sum_{i\in B(e)}\exp\left(\theta_i\right)\right)\right) \\
& + \sum_{\sigma\in\Lambda_{T(e)}} \frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \ln\left(\frac{\mathbb{P}_{\theta^{(t)}}(e)}{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}\right) \\
\geq \; & \sum_{\sigma\in\Lambda_{T(e)}} \frac{\mathbb{P}_{\theta^{(t)}}(e,\sigma)}{\mathbb{P}_{\theta^{(t)}}(e)} \sum_{u=1}^{|T(e)|}\left(\theta_{\sigma(u)} - \frac{\sum_{c'=u}^{|T(e)|}\exp\left(\theta_{\sigma(c')}\right) + \sum_{i\in B(e)}\exp\left(\theta_i\right)}{\sum_{c'=u}^{|T(e)|}\exp\left(\theta^{(t)}_{\sigma(c')}\right) + \sum_{i\in B(e)}\exp\left(\theta^{(t)}_i\right)}\right) + f(e,\theta^{(t)}) \\
\equiv \; & Q(e,\theta;\theta^{(t)})\,.
\end{aligned}
$$

Note that inequality in (19) is tight if $\theta_t = \theta$. The last inequality follows from the fact that for any positive x and y, we have

$$-\ln x \geq 1 - \ln y - (x/y) \quad \text{with equality if and only if } x = y.$$

Therefore, $Q(e, \theta; \theta^{(t)})$ minorizes $\ln(\mathbb{P}_\theta(e))$ and is equal to $\ln(\mathbb{P}_\theta(e))$ if and only if $\theta^{(t)} = \theta$.

7.2 Proof of Theorem 11

We define few additional notations. $p \equiv (1/n)\sum_{j=1}^{n} p_j$. $V(e_{j,a}) \equiv T(e_{j,a}) \cup B(e_{j,a})$ for all $j \in [n]$ and $a \in [\ell_j]$. Note that by definition of rank-breaking edge $e_{j,a}$, $V(e_{j,a})$ is a random set of items that are ranked in bottom $r_{j,a}$ positions in a set of S_j items by the user j.

The proof sketch is inspired from Khetan and Oh (2016). The main difference and technical challenge is in showing the strict concavity of $\mathcal{L}_{\mathrm{RB}}(\theta)$ when restricted to Ω_b. We want to prove an upper bound on $\Delta = \widehat{\theta} - \theta^*$, where $\widehat{\theta}$ is the sample dependent solution of the optimization (4) and θ^* is the true utility parameter from which the samples are drawn. Since $\widehat{\theta}, \theta^* \in \Omega_b$, it follows that $\Delta\mathbf{1} = 0$. Since $\widehat{\theta}$ is the maximizer of $\mathcal{L}_{\mathrm{RB}}(\theta)$, we have the following inequality,

$$\mathcal{L}_{\mathrm{RB}}(\widehat{\theta}) - \mathcal{L}_{\mathrm{RB}}(\theta^*) - \langle \nabla \mathcal{L}_{\mathrm{RB}}(\theta^*), \Delta \rangle \;\geq\; -\langle \nabla \mathcal{L}_{\mathrm{RB}}(\theta^*), \Delta \rangle \;\geq\; -\|\nabla \mathcal{L}_{\mathrm{RB}}(\theta^*)\|_2 \|\Delta\|_2, \tag{20}$$

where the last inequality uses the Cauchy-Schwartz inequality. By the mean value theorem, there exists a $\theta = c\widehat{\theta} + (1-c)\theta^*$ for some $c \in [0,1]$ such that $\theta \in \Omega_b$ and

$$\mathcal{L}_{\mathrm{RB}}(\widehat{\theta}) - \mathcal{L}_{\mathrm{RB}}(\theta^*) - \langle \nabla \mathcal{L}_{\mathrm{RB}}(\theta^*), \Delta \rangle \;=\; \frac{1}{2}\Delta^\top H(\theta)\Delta \leq -\frac{1}{2}\lambda_2(-H(\theta))\|\Delta\|_2^2, \tag{21}$$

where $\lambda_2(-H(\theta))$ is the second smallest eigen value of $-H(\theta)$. We will show in Lemma 15 that $-H(\theta)$ is positive semi definite with one eigenvalue at zero with a corresponding eigen vector $\mathbf{1} = [1, \ldots, 1]^\top$. The last inequality follows since $\Delta^\top \mathbf{1} = 0$. Combining Equations (20) and (21),

$$\|\Delta\|_2 \;\leq\; \frac{2\|\nabla \mathcal{L}_{\mathrm{RB}}(\theta^*)\|_2}{\lambda_2(-H(\theta))},$$

where we used the fact that $\lambda_2(-H(\theta)) > 0$ from Lemma 15. The following technical lemmas prove that the norm of the gradient is upper bounded by $\gamma_2^{-1/2} e^b \sqrt{6np \log d}$ with high probability and the second smallest eigen value of negative of the Hessian is lower bounded by $(1/8)\, e^{-6b}\alpha\gamma_1\gamma_2\gamma_3(np/(d-1))$. This finishes the proof of Theorem 11.

The (random) gradient of the log likelihood in (4) can be written as the following, where the randomness is in which items ended up in the top set $T(e_{j,a})$ and the bottom set $B(e_{j,a})$:

$$\nabla_i \mathcal{L}_{\mathrm{RB}}(\theta) \;=\; \sum_{j=1}^{n}\sum_{a=1}^{\ell_j} \sum_{\substack{\mathcal{C}\subseteq S_j,\\ |\mathcal{C}|=r_{j,a}-1}} \mathbb{I}\{\, V(e_{j,a}) = \{\mathcal{C}, i\}\,\} \frac{\partial \log \mathbb{P}_\theta(e_{j,a})}{\partial \theta_i}\,.$$

Note that we are intentionally decomposing each summand as a summation over all $\mathcal{C}$ of size $r_{j,a}-1$, such that we can separate the analysis of the expectation in the following lemma. The random variable $\mathbb{I}\{\{\mathcal{C}, i\} = V(e_{j,a})\}$ indicates that we only include one term for any given instance of the sample. Note that the event $\mathbb{I}\{\{\mathcal{C}, i\} = V(e_{j,a})\}$ is equivalent to the event that the $\{\mathcal{C}, i\}$ items are ranked in bottom $r_{j,a}$ positions in the set S_j, that is $V(e_{j,a})$ items are ranked in bottom $r_{j,a}$ positions in the set S_j.

Lemma 13 *If the j-th poset is drawn from the PL model with weights θ^* then for any given $\mathcal{C}' \subseteq S_j$ with $|\mathcal{C}'| = r_{j,a}$,*

$$\mathbb{E}\left[\mathbb{I}\{\mathcal{C}' = V(e_{j,a})\} \frac{\partial \log \mathbb{P}_{\theta^*}(e_{j,a})}{\partial \theta_i^*} \middle| \{e_{j,a'}\}_{a'<a}\right] \;=\; 0\,.$$

First, this lemma implies that $\mathbb{E}\big[\mathbb{I}\{\,\mathcal{C}' = V(e_{j,a})\,\}\frac{\partial \log \mathbb{P}_{\theta^*}(e_{j,a})}{\partial \theta_i^*}\big] = 0$. Secondly, the above lemma allows us to construct a vector-valued martingale and apply a generalization of Azuma-Hoeffding's tail bound on the norm to prove the following concentration of measure. This proves the desired bound on the gradient.

Lemma 14 *If n posets are independently drawn over d items from the PL model with weights θ^* then with probability at least $1 - 2e^3 d^{-3}$,*

$$\|\nabla \mathcal{L}_{\mathrm{RB}}(\theta^*)\| \leq \gamma_2^{-1/2} e^b \sqrt{6np \log d}\,,$$

where γ_2 depend on the choice of the rank-breaking and are defined in Section 4.1.

We will prove in (26) that the Hessian matrix $H(\theta) \in \mathcal{S}^d$ with $H_{ii'}(\theta) = \frac{\partial^2 \mathcal{L}_{\mathrm{RB}}(\theta)}{\partial \theta_i \partial \theta_{i'}}$ can be expressed as

$$-H(\theta) = \sum_{j=1}^{n}\sum_{a=1}^{\ell_j}\sum_{i<i'\in S_j} \mathbb{I}\{(i,i') \subseteq V(e_{j,a})\} \left(\frac{\partial^2 \log \mathbb{P}_\theta(e_{j,a})}{\partial \theta_i \partial \theta_{i'}} (e_i - e_{i'})(e_i - e_{i'})^\top \right). \tag{22}$$

It is easy to see that $H(\theta)\mathbf{1} = 0$. The following lemma proves a lower bound on the second smallest eigenvalue $\lambda_2(-H(\theta))$ in terms of re-scaled spectral gap α of the comparison graph $\mathcal{H}$ defined in Section 4.1.

Lemma 15 *Under the hypothesis of Theorem 11, if the assumptions in Equation (14) are satisfied then with probability at least $1 - d^{-3}$, the following holds for any $\theta \in \Omega_b$:*

$$\lambda_2(-H(\theta)) \;\geq\; \frac{e^{-6b}\alpha\gamma_1\gamma_2\gamma_3}{8} \frac{np}{(d-1)}\,,$$

and $\lambda_1(-H(\theta)) = 0$ with corresponding eigenvector $\mathbf{1}$.

This finishes the proof of the desired claim.

7.3 Proof of Lemma 13

Recall that $e_{j,a}$ is a random event where randomness is in which items ended up in the top-set $T(e_{j,a})$ and the bottom-set $B(e_{j,a})$, and $\mathbb{P}_{\theta^*}(e_{j,a}) = \mathbb{P}_{\theta^*}[B(e_{j,a}) \prec T(e_{j,a})]$ that is the probability of observing $B(e_{j,a}) \prec T(e_{j,a})$ when the offer set is $B(e_{j,a}) \cup T(e_{j,a})$ as defined in (3). Define, $\mathbb{P}_{\theta^*,S_j}[e_{j,a}|V(e_{j,a}) = \mathcal{C}']$ to be the conditional probability of observing $B(e_{j,a}) \prec T(e_{j,a})$, when the offer set is S_j, conditioned on the event that $V(e_{j,a}) = \mathcal{C}'$. Note that we have put subscript S_j in $\mathbb{P}_{\theta^*}$ to specify that the offer set is S_j. Observe that for any set $\mathcal{C}' \subseteq S_j$, the event $\{\mathcal{C}' = V(e_{j,a})\}$ is equivalent to $\mathcal{C}'$ items being ranked in bottom $r_{j,a}$ positions when the offer set is S_j. In other words, it is conditioned on the event that the subset $V(e_{j,a})$ items are ranked in bottom $r_{j,a}$ positions when the offer set is S_j. In Equation (23), we show that under PL model

$$\mathbb{P}_{\theta^*,S_j}[e_{j,a}|V(e_{j,a}) = \mathcal{C}'] = \mathbb{P}_{\theta^*}[e_{j,a}]\,.$$

Also, by conditioning on any outcome of $\{e_{j,a'}\}_{a'<a}$ it can be checked that

$$\mathbb{P}_{\theta^*,S_j}[e_{j,a}|V(e_{j,a}) = \mathcal{C}', \{e_{j,a'}\}_{a'<a}] = \mathbb{P}_{\theta^*,S_j}[e_{j,a}|V(e_{j,a}) = \mathcal{C}'].$$

Therefore, we have

$$\begin{aligned}
&\mathbb{E}\Big[\frac{\partial \log \mathbb{P}_{\theta^*}\big[e_{j,a}\big]}{\partial \theta_i^*}\Big| V(e_{j,a})=\mathcal{C}', \{e_{j,a'}\}_{a'<a}\Big] \\
&= \mathbb{E}\Big[\frac{\partial \log \mathbb{P}_{\theta^*,S_j}\big[e_{j,a}|V(e_{j,a})=\mathcal{C}', \{e_{j,a'}\}_{a'<a}\big]}{\partial \theta_i^*}\Big| V(e_{j,a})=\mathcal{C}', \{e_{j,a'}\}_{a'<a}\Big] \\
&= \sum_{\substack{e_{j,a}:V(e_{j,a})=\mathcal{C}' \\ \{e_{j,a'}\}_{a'<a}}} \mathbb{P}_{\theta^*,S_j}\Big[e_{j,a}\Big|V(e_{j,a})=\mathcal{C}', \{e_{j,a'}\}_{a'<a}\Big]\, \frac{\partial}{\partial \theta_i^*} \log \mathbb{P}_{\theta^*,S_j}\Big[e_{j,a}\Big|V(e_{j,a})=\mathcal{C}', \{e_{j,a'}\}_{a'<a}\Big] \\
&= \frac{\partial}{\partial \theta_i^*} \sum_{e_{j,a}:V(e_{j,a})=\mathcal{C}'} \mathbb{P}_{\theta^*,S_j}\Big[e_{j,a}\Big|V(e_{j,a})=\mathcal{C}'\Big] = \frac{\partial}{\partial \theta_i^*} 1 = 0\,,
\end{aligned}$$

where we used $\{e_{j,a} : V(e_{j,a}) = \mathcal{C}'\} = \{e_{j,a} : V(e_{j,a}) = \mathcal{C}', \{e_{j,a'}\}_{a'<a}\}$ which follows from the definition of rank-breaking edges $e_{j,a}$. This proves the desired claim. It remains to show that

$$\mathbb{P}_{\theta^*,S_j}[e_{j,a}|V(e_{j,a})=\mathcal{C}'] = \mathbb{P}_{\theta^*}[e_{j,a}]\,.$$

This follows from the fact that under PL model for any disjoint set of items $\{\mathcal{C}_i\}_{i\in[\ell]}$ such that $\cup_{i=1}^{\ell}\mathcal{C}_i = S$,

$$\mathbb{P}\big(\mathcal{C}_\ell \prec \mathcal{C}_{\ell-1} \prec \cdots \prec \mathcal{C}_1\big) = \mathbb{P}\big(\mathcal{C}_\ell \prec \mathcal{C}_{\ell-1}\big)\mathbb{P}\big(\{\mathcal{C}_\ell, \mathcal{C}_{\ell-1}\} \prec \mathcal{C}_{\ell-2}\big)\cdots\mathbb{P}\big(\{\mathcal{C}_\ell, \mathcal{C}_{\ell-1}, \cdots, \mathcal{C}_2\} \prec \mathcal{C}_1\big)\,, \quad (23)$$

where $\mathbb{P}(\mathcal{C}_{i_1} \prec \mathcal{C}_{i_2})$ is the probability that $\mathcal{C}_{i_2}$ items are ranked higher than $\mathcal{C}_{i_1}$ items when the offer set is $\{\mathcal{C}_{i_1} \cup \mathcal{C}_{i_2}\}$.

7.4 Proof of Lemma 14

We view $\nabla\mathcal{L}_{\mathrm{RB}}(\theta^*)$ as the final value of a discrete time vector-valued martingale with values in $\mathbb{R}^d$. Define $\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})} \in \mathbb{R}^d$ as the gradient vector arising out of each rank-breaking edge $\{e_{j,a}\}_{j\in[n],a\in[\ell_j]}$ as

$$\nabla_i \mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}(\theta^*) \;\equiv\; \sum_{\mathcal{C}\subseteq S_j} \mathbb{I}\big\{V(e_{j,a}) = \{\mathcal{C}, i\}\big\} \nabla_i \log \mathbb{P}_{\theta^*}(e_{j,a})\,,$$

such that $\nabla\mathcal{L}_{\mathrm{RB}}(\theta^*) = \sum_{j\in[n]}\sum_{a\in[\ell_j]} \nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}$. We take $\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}$ as the incremental random vector in a martingale of $\sum_{j=1}^{n}\ell_j$ time steps. Let $H_{j,a}$ denote (the sigma algebra of) the history up to $e_{j,a}$ and define a sequence of random vectors in $\mathbb{R}^d$:

$$Z_{j,a} \;\equiv\; \mathbb{E}[\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}(\theta^*)|H_{j,a}]\,,$$

with the convention that $Z_{1,1} = \mathbb{E}[\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}(\theta^*)] = 0$ as proved in Lemma 13. It also follows from Lemma 13 that $\mathbb{E}[Z_{j,a+1}|Z_{j,a}] = Z_{j,a}$ for $a < \ell_j$. Also, from the independence of samples, it follows that $\mathbb{E}[Z_{j+1,1}|Z_{j,\ell_j}] = Z_{j,\ell_j}$. Applying a generalized version of the vector Azuma-Hoeffding inequality which readily follows from [Theorem 1.8, Hayes (2005)], we have

$$\mathbb{P}\big[\,\|\nabla\mathcal{L}_{\mathrm{RB}}(\theta^*)\| \geq \delta\,\big] \;\leq\; 2e^3 \exp\left(-\frac{\delta^2}{\sum_{j=1}^{n}\sum_{a=1}^{\ell_j} m_{j,a} 2\gamma_2^{-1} e^{2b}}\right),$$

where we used $\|\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}\|^2 \leq m_{j,a}2\gamma_2^{-1}e^{2b}$. Choosing $\delta = \gamma_2^{-1}e^b\sqrt{6np\log d}$ gives the desired bound.

Now we are left to show that $\|\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}\|^2 \leq 2m_{j,a}\gamma_2^{-1}e^{2b}$ for any $\theta \in \Omega_b$. Recall that $\sigma \in \Lambda_{T(e_{j,a})}$ is the set of all full rankings over $T(e_{j,a})$ items. In rest of the proof, with a slight abuse of notations, we extend each of these ranking σ over $T(e_{j,a}) \cup B(e_{j,a})$ items in the following way. Consider any full ranking $\tilde{\sigma}$ over $B(e_{j,a})$ items. Then for each $\sigma \in \Lambda_{T(e_{j,a})}$, the extension is such that $\sigma(|T(e_{j,a})| + c) = \tilde{\sigma}(c)$ for $1 \leq c \leq |B(e_{j,a})|$. The choice of ranking $\tilde{\sigma}$ will have no impact on any of the following mathematical expressions. From the definition of $\mathbb{P}_\theta(e_{j,a})$ (3), we have, for any $i \in V(e_{j,a})$,

$$\begin{aligned}
\frac{\partial \mathbb{P}_\theta(e_{j,a})}{\partial \theta_i} &= \mathbb{I}\{i \in T(e_{j,a})\}\mathbb{P}_\theta(e_{j,a}) \\
&\quad - \underbrace{\sum_{\sigma\in\Lambda_{T(e_{j,a})}} \underbrace{\frac{\exp\left(\sum_{c=1}^{m_{j,a}}\theta_{\sigma(c)}\right)}{\prod_{u=1}^{m_{j,a}}\left(\sum_{c'=u}^{r_{j,a}}\exp\left(\theta_{\sigma(c')}\right)\right)}}_{\equiv A_\sigma} \underbrace{\left(\sum_{u'=1}^{m_{j,a}} \frac{\mathbb{I}\{\sigma^{-1}(i) \geq u'\}\exp(\theta_i)}{\sum_{c'=u'}^{r_{j,a}}\exp\left(\theta_{\sigma(c')}\right)}\right)}_{\equiv B_{\sigma,i}}}_{\equiv E_i}\,. \qquad (24)
\end{aligned}$$

Note that $A_\sigma, B_{\sigma,i}$ and E_i depend on $e_{j,a}$. Observe that for any $1 \leq u' \leq m_{j,a}$ and any $\sigma \in \Lambda_{T(e_{j,a})}$,

$$\sum_{i\in V(e_{j,a})} \mathbb{I}\{\sigma^{-1}(i) \geq u'\}\exp(\theta_i) = \sum_{c'=u'}^{r_{j,a}} \exp\left(\theta_{\sigma(c')}\right).$$

Therefore, $\sum_{i\in V(e_{j,a})} B_{\sigma,i} = m_{j,a}$. It follows that

$$\sum_{i\in V(e_{j,a})} E_i \;=\; \sum_{\sigma\in\Lambda_{T(e_{j,a})}} A_\sigma\Big(\sum_{i\in V(e_{j,a})} B_{\sigma,i}\Big) \;=\; m_{j,a}\sum_{\sigma\in\Lambda_{T(e_{j,a})}} A_\sigma \;=\; m_{j,a}\mathbb{P}_\theta(e_{j,a})\,, \qquad (25)$$

where the last equality follows from the definition of $\mathbb{P}_\theta(e_{j,a})$ (4). Also, since for any i, i', $e^{(\theta_i-\theta_{i'})} \leq e^{2b}$; for any i, $B_{\sigma,i} \leq e^{2b}\sum_{k=r_{j,a}-m_{j,a}+1}^{r_{j,a}}(1/k) \leq e^{2b}(1+\log(r_{j,a}/(r_{j,a}-m_{j,a}+1))) \leq \gamma_2^{-1}e^{2b}$, where the last inequality follows from the definition of γ_2 (12) and the fact that $x \leq \sqrt{1+\log x}$ for all $x \geq 1$. Therefore, $E_i \leq \gamma_2^{-1}e^{2b}\sum_{\sigma\in\Lambda_{T(e_{j,a})}} A_\sigma = \gamma_2^{-1}e^{2b}\mathbb{P}_\theta(e_{j,a})$. We have $\partial\log\mathbb{P}_\theta(e_{j,a})/\partial\theta_i = (1/\mathbb{P}_\theta(e_{j,a}))\partial\mathbb{P}_\theta(e_{j,a})/\partial\theta_i = \mathbb{I}\{i \in T(e_{j,a})\} - E_i/\mathbb{P}_\theta(e_{j,a})$. Since $|T(e_{j,a})| = m_{j,a}$, $\|\nabla\mathcal{L}_{\mathrm{RB}}^{(e_{j,a})}\|^2 \leq m_{j,a} + \sum_{i\in V(e_{j,a})}(E_i/\mathbb{P}_\theta(e_{j,a}))^2 \leq 2m_{j,a}\gamma_2^{-1}e^{2b}$, where we used (25) and the fact that $\gamma_2^{-1} \geq 1$.

7.4.1 Proof of Lemma 15

First, we prove (22). For brevity, remove $\{j,a\}$ from $\mathbb{P}_\theta(e_{j,a})$. From Equations (24) and (25), and $|T(e_{j,a})| = m_{j,a}$, we have $\sum_{i\in V(e_{j,a})}\frac{\partial}{\partial\theta_i}\mathbb{P}_\theta(e) = m_{j,a}\mathbb{P}_\theta(e) - m_{j,a}\mathbb{P}_\theta(e) = 0$. It follows that

$$\begin{aligned}
&\sum_{i\in V(e_{j,a})}\left(\frac{\partial^2\log\mathbb{P}_\theta(e)}{\partial\theta_{i'}\partial\theta_i}\right) = \\
&\frac{1}{\mathbb{P}_\theta(e)}\frac{\partial}{\partial\theta_{i'}}\left(\sum_{i\in V(e_{j,a})}\left(\frac{\partial\mathbb{P}_\theta(e)}{\partial\theta_i}\right)\right) - \frac{1}{(\mathbb{P}_\theta(e))^2}\frac{\partial\mathbb{P}_\theta(e)}{\partial\theta_{i'}}\left(\sum_{i\in V(e_{j,a})}\left(\frac{\partial\mathbb{P}_\theta(e)}{\partial\theta_i}\right)\right) = 0\,. \qquad (26)
\end{aligned}$$

Since by definition $\mathcal{L}_{\mathrm{RB}}(\theta) = \sum_{j=1}^{n}\sum_{a=1}^{\ell_j}\log \mathbb{P}_\theta(e_{j,a})$, and $H_{ii'}(\theta) = \frac{\partial^2 \mathcal{L}_{\mathrm{RB}}(\theta)}{\partial\theta_i \partial\theta_{i'}}$ which is a symmetric matrix, Equation (26) implies that it can be expressed as given in Equation (22). It follows that all-ones is an eigenvector of $H(-\theta)$ with the corresponding eigenvalue being zero.

To get a lower bound on $\lambda_2(-H(\theta))$, we apply Weyl's inequality

$$\lambda_2(-H(\theta)) \geq \lambda_2(\mathbb{E}[-H(\theta)]) - \|H(\theta) - \mathbb{E}[H(\theta)]\|\,.$$

We will show in (27) that $\lambda_2(\mathbb{E}[-H(\theta)]) \geq e^{-6b}\alpha\gamma_1\gamma_2\gamma_3(np/(4(d-1)))$ and in (39) that $\|H(\theta) - \mathbb{E}[H(\theta)]\| \leq 16e^{4b}\nu\sqrt{\frac{p_{\max}}{\kappa_{\min}}\frac{np}{\beta(d-1)}\log d}$. Putting these together,

$$\begin{aligned}\lambda_2(-H(\theta)) &\geq e^{-6b}\alpha\gamma_1\gamma_2\gamma_3\frac{np}{4(d-1)} - 16e^{4b}\nu\sqrt{\frac{p_{\max}}{\kappa_{\min}}\frac{np}{\beta(d-1)}\log d}\\ &\geq \frac{e^{-6b}\alpha\gamma_1\gamma_2\gamma_3}{8}\frac{np}{(d-1)}\,,\end{aligned}$$

where the last inequality follows from the assumption on $n\kappa_{\min}$ given in (14).

To prove a lower bound on $\lambda_2(\mathbb{E}[-H(\theta)])$, we claim that for $\theta \in \Omega_b$,

$$\begin{aligned}\mathbb{E}\big[-H(\theta)\big] &\succeq e^{-6b}\gamma_1\gamma_2\gamma_3\sum_{j=1}^{n}\frac{p_j}{4\kappa_j(\kappa_j-1)}\sum_{i<i'\in S_j}(e_i-e_{i'})(e_i-e_{i'})^\top \qquad (27)\\ &= \frac{e^{-6b}\gamma_1\gamma_2\gamma_3}{4}L\,,\end{aligned}$$

where $L \in \mathcal{S}^d$ is defined in (10). Using $\lambda_2(L) = np\alpha/(d-1)$ from (11), we have $\lambda_2(-H(\theta)) \geq e^{-6b}\alpha\gamma_1\gamma_2\gamma_3(np/(4(d-1)))$. To prove (27), notice that

$$\mathbb{E}[-H(\theta)_{ii'}] = \mathbb{E}\Big[\sum_{j\in[n]}\sum_{a\in[\ell_j]}\mathbb{I}\big\{(i,i')\subseteq V(e_{j,a})\big\}\frac{\partial^2\log\mathbb{P}_\theta(e_{j,a})}{\partial\theta_i\partial\theta_{i'}}\Big]\,, \qquad (28)$$

when $i \neq i'$. We will show that for any $i \neq i' \in V(e_{j,a})$,

$$\frac{\partial^2\log\mathbb{P}_\theta(e_{j,a})}{\partial\theta_i\partial\theta_{i'}} \geq \begin{cases}\frac{e^{-2b}m_{j,a}}{r_{j,a}^2} & \text{if } i,i'\in B(e_{j,a})\\ -\frac{e^{4b}m_{j,a}^2}{(r_{j,a}-m_{j,a}+1)^2} & \text{otherwise}\,.\end{cases} \qquad (29)$$

We need to bound the probability of two items appearing in the bottom-set $B(e_{j,a})$ and in the top-set $T(e_{j,a})$.

Lemma 16 *Consider a ranking σ over a set $S \subseteq [d]$ such that $|S| = \kappa$. For any two items $i, i' \in S$, $\theta \in \Omega_b$, and $1 \leq \ell, \ell_1, \ell_2 \leq \kappa - 1$,*

$$\begin{aligned}\mathbb{P}_\theta\big[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\big] &\geq \frac{e^{-4b}(\kappa-\ell)(\kappa-\ell-1)}{\kappa(\kappa-1)}\left(1-\frac{\ell}{\kappa}\right)^{2e^{2b}-2}, \qquad (30)\\ \mathbb{P}_\theta\big[\sigma^{-1}(i) = \ell\big] &\leq \frac{e^{6b}}{\kappa-\ell}\,, \qquad (31)\\ \mathbb{P}_\theta\big[\sigma^{-1}(i) = \ell_1, \sigma^{-1}(i') = \ell_2\big] &\leq \frac{e^{10b}}{(\kappa-\ell_1-1)(\kappa-\ell_2)}\,. \qquad (32)\end{aligned}$$

where the probability $\mathbb{P}_\theta$ is with respect to the sampled ranking resulting from PL weights $\theta \in \Omega_b$.

Substituting $\ell = \kappa_j - r_{j,a} + m_{j,a}$ in (30), and $\ell, \ell_1, \ell_2 \leq \kappa_j - r_{j,a} + m_{j,a}$ in (31) and (32), we have,

$$\begin{aligned}\mathbb{P}_\theta\big[(i,i')\subseteq B(e_{j,a})\big] &\geq \frac{e^{-4b}(r_{j,a}-m_{j,a})^2}{4\kappa_j(\kappa_j-1)}\Big(\frac{r_{j,a}-m_{j,a}}{\kappa_j}\Big)^{2e^{2b}-2}, \qquad (33)\\ \mathbb{P}_\theta\big[i\in T(e_{j,a}), i'\in B(e_{j,a})\big] &\leq m_{j,a}\max_{\ell\in[\kappa_j-r_{j,a}+m_{j,a}]}\mathbb{P}(\sigma^{-1}(i)=\ell)\\ &\leq \frac{e^{6b}m_{j,a}}{r_{j,a}-m_{j,a}}\,, \qquad (34)\\ \mathbb{P}_\theta\big[(i,i')\subseteq T(e_{j,a})\big] &\leq m_{j,a}^2\max_{\ell_1,\ell_2\in[\kappa_j-r_{j,a}+m_{j,a}]}\mathbb{P}(\sigma^{-1}(i)=\ell_1,\sigma^{-1}(i')=\ell_2)\\ &\leq \frac{e^{10b}m_{j,a}^2}{2\,(r_{j,a}-m_{j,a}-1)(r_{j,a}-m_{j,a})}\,, \qquad (35)\end{aligned}$$

where (33) uses $r_{j,a} - m_{j,a} - 1 \geq (r_{j,a} - m_{j,a})/4$, (34) uses $\mathbb{P}_\theta[i \in T(e_{j,a}), i' \in B(e_{j,a})] \leq \mathbb{P}_\theta[i \in T(e_{j,a})]$, and (34)-(35) uses counting on the possible choices. The bound in (35) is smaller than the one in (34) as per our assumption that $\gamma_3 > 0$.

Using Equations (28)-(29) and (33)-(35), and the definitions of $\gamma_1, \gamma_2, \gamma_3$ from Section 4.1, we get

$$\begin{aligned}&\mathbb{E}[-H(\theta)_{ii'}] \geq\\ &\sum_{j\in[n]}\sum_{a\in[\ell_j]}\Big\{\underbrace{\Big(\frac{r_{j,a}-m_{j,a}}{\kappa_j}\Big)^{2e^{2b}-2}}_{\geq\gamma_1}\underbrace{\Big(\frac{r_{j,a}-m_{j,a}}{r_{j,a}}\Big)^{2}}_{\geq\gamma_2}\frac{e^{-6b}m_{j,a}}{4\kappa_j(\kappa_j-1)} - \frac{e^{6b}m_{j,a}}{r_{j,a}-m_{j,a}}\frac{e^{4b}m_{j,a}^2}{(r_{j,a}-m_{j,a}+1)^2}\Big\}\\ &\geq \sum_{j,a}\frac{\gamma_1\gamma_2 e^{-6b}m_{j,a}}{4\kappa_j(\kappa_j-1)}\underbrace{\Big(1-\frac{4e^{16b}}{\gamma_1}\frac{m_{j,a}^2r_{j,a}^2\kappa_j^2}{(r_{j,a}-m_{j,a})^5}\Big)}_{\geq\gamma_3}\,.\end{aligned}$$

This combined with (22) proves the desired claim (27). Further, in Appendix 7.7, we show that if $m_{j,a} \leq 3$ for all $\{j,a\}$ then $\frac{\partial^2\log\mathbb{P}_\theta(e_{j,a})}{\partial\theta_i\partial\theta_{i'}}$ is non-negative even for $i \neq i' \in T(e_{j,a})$, and $i \in T(e_{j,a}), i' \in B(e_{j,a})$ as opposed to a negative lower-bound given in (29). Therefore, bound on $\mathbb{E}[-H(\theta)]$ in (27) can be tightened by a factor of γ_3.

To prove claim (29), define the following for $\sigma \in \Lambda_{T(e_{j,a})}$,

$$\begin{aligned}A_\sigma &\equiv \frac{\exp\big(\sum_{c=1}^{m_{j,a}}\theta_{\sigma(c)}\big)}{\prod_{u=1}^{m_{j,a}}\big(\sum_{c'=u}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)\big)}\,, B_\sigma \equiv \sum_{u'=1}^{m_{j,a}}\frac{1}{\sum_{c'=u'}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)}\,,\\ B_{\sigma,i} &\equiv \sum_{u'=1}^{m_{j,a}}\frac{\mathbb{I}\{\sigma^{-1}(i)\geq u'\}}{\sum_{c'=u'}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)}\,, C_\sigma \equiv \sum_{u'=1}^{m_{j,a}}\frac{1}{\big(\sum_{c'=u'}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)\big)^2}\,,\\ C_{\sigma,i} &\equiv \sum_{u'=1}^{m_{j,a}}\frac{\mathbb{I}\{\sigma^{-1}(i)\geq u'\}}{\big(\sum_{c'=u'}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)\big)^2}\,, C_{\sigma,i,i'} \equiv \sum_{u'=1}^{m_{j,a}}\frac{\mathbb{I}\{\sigma^{-1}(i),\sigma^{-1}(i')\geq u'\}}{\big(\sum_{c'=u'}^{r_{j,a}}\exp\big(\theta_{\sigma(c')}\big)\big)^2}\,. \qquad (36)\end{aligned}$$

First, a few observations about the expression of A_σ. For any $\sigma \in \Lambda_{T(e_{j,a})}$ and any $i \in V(e_{j,a})$, θ_i is in the numerator if and only if $i \in T(e_{j,a})$, since in all the rankings that are consistent with

the observation $e_{j,a}$, $T(e_{j,a})$ items are ranked in top $m_{j,a}$ positions. For any $\sigma \in \Lambda_{T(e_{j,a})}$ and any $i \in B(e_{j,a})$, θ_i is in all the product terms $\prod_{u=1}^{m_{j,a}}(\cdot)$ of the denominator, since in all the consistent rankings these items are ranked below $m_{j,a}$ position. For any $i \in T(e_{j,a})$, θ_i appears in product term corresponding to index u if and only if item i is ranked at position u or lower than u in the ranking $\sigma \in \Lambda_{T(e_{j,a})}$. Now, observe that B_σ is defined such that the partial derivative of A_σ with respect to any $i \in B(e_{j,a})$ is $-A_\sigma B_\sigma e^{\theta_i}$, and $B_{\sigma,i}$ is defined such that the partial derivative of A_σ with respect to any $i \in T(e_{j,a})$ is $A_\sigma - A_\sigma B_\sigma e^{\theta_i}$. Further, observe that $-C_\sigma e^{\theta_i}$ is the partial derivative of B_σ with respect to $i \in B(e_{j,a})$, $-C_{\sigma,i}e^{\theta_i}$ is the partial derivative of $B_{\sigma,i}$ with respect to $i \in T(e_{j,a})$, and $-C_{\sigma,i}e^{\theta_{i'}}$ is the partial derivative of $B_{\sigma,i}$ with respect to $i' \in B(e_{j,a})$. $-C_{\sigma,i,i'}e^{\theta_{i'}}$ is the partial derivative of $B_{\sigma,i}$ with respect to $i' \neq i \in T(e_{j,a})$.

For ease of notation, we omit subscript (j,a) whenever it is clear from the context. Also, we use $\sum_\sigma$ to denote $\sum_{\sigma\in\Lambda_{T(e_{j,a})}}$. With the above defined notations, from (4), we have, $\mathbb{P}_\theta(e) = \sum_\sigma A_\sigma$. With the above given observations for the notations in (36), first partial derivative of $\mathbb{P}_\theta(e)$ can be expressed as following:

$$\frac{\partial \mathbb{P}_\theta(e)}{\partial \theta_i} = \begin{cases} \sum_\sigma \left(A_\sigma - A_\sigma B_{\sigma,i} e^{\theta_i}\right) & \text{if } i \in T(e_{j,a}) \\ \sum_\sigma \left(-A_\sigma B_\sigma e^{\theta_i}\right) & \text{if } i \in B(e_{j,a})\,. \end{cases} \tag{37}$$

It follows that for $i \neq i' \in V(e_{j,a})$,

$$\begin{aligned} &\frac{\partial^2 \mathbb{P}_\theta(e)}{\partial\theta_i \partial\theta_{i'}} \\ &= \begin{cases} \sum_\sigma \left((A_\sigma (B_\sigma)^2 + A_\sigma C_\sigma) e^{(\theta_i+\theta_{i'})}\right) & \text{if } i,i' \in B(e_{j,a}) \\ \sum_\sigma \left(A_\sigma - A_\sigma B_{\sigma,i'} e^{\theta_{i'}} + (A_\sigma B_{\sigma,i} B_{\sigma,i'} + A_\sigma C_{\sigma,i,i'}) e^{(\theta_i+\theta_{i'})} - A_\sigma B_{\sigma,i} e^{\theta_i}\right) & \text{if } i,i' \in T(e_{j,a}) \\ \sum_\sigma \left((A_\sigma B_\sigma B_{\sigma,i} + A_\sigma C_{\sigma,i}) e^{(\theta_i+\theta_{i'})} - A_\sigma B_\sigma e^{\theta_{i'}}\right) & \text{otherwise}\,. \end{cases} \end{aligned}$$

Using $\frac{\partial^2 \log \mathbb{P}_\theta(e)}{\partial\theta_i\partial\theta_{i'}} = \frac{1}{\mathbb{P}_\theta(e)}\frac{\partial^2 \mathbb{P}_\theta(e)}{\partial\theta_i\partial\theta_{i'}} - \frac{1}{(\mathbb{P}_\theta(e))^2}\frac{\partial \mathbb{P}_\theta(e)}{\partial\theta_i}\frac{\partial \mathbb{P}_\theta(e)}{\partial\theta_{i'}}$, with above derived first and second derivatives, and after following some algebra, we have

$$\begin{aligned} &\frac{(\mathbb{P}_\theta(e))^2}{e^{(\theta_i+\theta_{i'})}} \frac{\partial^2 \log \mathbb{P}_\theta(e)}{\partial\theta_i\partial\theta_{i'}} \\ &= \begin{cases} (\sum_\sigma A_\sigma)(\sum_\sigma A_\sigma (B_\sigma)^2) - (\sum_\sigma A_\sigma B_\sigma)^2 + (\sum_\sigma A_\sigma)(\sum_\sigma A_\sigma C_\sigma) & \text{if } i,i' \in B(e_{j,a}) \\ (\sum_\sigma A_\sigma)(\sum_\sigma A_\sigma B_{\sigma,i} B_{\sigma,i'} + A_\sigma C_{\sigma,i,i'}) - (\sum_\sigma A_\sigma B_{\sigma,i})(\sum_\sigma A_\sigma B_{\sigma,i'}) & \text{if } i,i' \in T(e_{j,a}) \\ (\sum_\sigma A_\sigma)(\sum_\sigma A_\sigma B_\sigma B_{\sigma,i} + A_\sigma C_{\sigma,i}) - (\sum_\sigma A_\sigma B_\sigma)(\sum_\sigma A_\sigma B_{\sigma,i}) & \text{otherwise}\,. \end{cases} \end{aligned} \tag{38}$$

Observe that from Cauchy-Schwartz inequality $(\sum_\sigma A_\sigma)(\sum_\sigma A_\sigma (B_\sigma)^2) - (\sum_\sigma A_\sigma B_\sigma)^2 \geq 0$. Also, we have $e^{(\theta_i+\theta_{i'})} C_\sigma \geq e^{-2b}(m/r^2)$ and $e^{\theta_i} B_{\sigma,i} \leq e^{\theta_i} B_\sigma \leq e^{2b}(m/(r-m+1))$ for any $i \in V(e_{j,a})$. This proves the desired claim (29).

Next we need to upper bound deviation of $-H(\theta)$ from its expectation. From (38), we have, $\left|\frac{\partial^2 \log \mathbb{P}_\theta(e_{j,a})}{\partial\theta_i\partial\theta_{i'}}\right| \leq 3e^{4b} m_{j,a}^2/(r_{j,a}-m_{j,a}+1)^2 \leq 3e^{4b}\nu m_{j,a}/(\kappa_j(\kappa_j-1))$, where the last inequality

follows from the definition of ν (13). Therefore,

$$\begin{aligned} -H(\theta) &\preceq 3e^{4b}\nu \sum_{j=1}^n \sum_{a=1}^{\ell_j} \sum_{i<i'\in S_j} \mathbb{I}\{(i,i') \subseteq V(e_{j,a})\} \frac{m_{j,a}}{\kappa_j(\kappa_j-1)} (e_i - e_{i'})(e_i-e_{i'})^\top \\ &\preceq 3e^{4b}\nu \sum_{j=1}^n \sum_{i<i'\in S_j} \frac{\sum_{a=1}^{\ell_j} m_{j,a}}{\kappa_j(\kappa_j-1)} (e_i-e_{i'})(e_i-e_{i'})^\top \equiv \sum_{j=1}^n y_j L_j\,, \end{aligned}$$

where $y_j = (3e^{4b}\nu p_j)/(\kappa_j(\kappa_j-1))$ and $L_j = \sum_{i<i'\in S_j}(e_i-e_{i'})(e_i-e_{i'})^\top = \kappa_j \mathrm{diag}(e_{S_j}) - e_{S_j}e_{S_j}^\top$ for $e_{S_j} = \sum_{i\in S_j} e_i$. Observe that $\|y_j L_j\| \leq (3e^{4b}\nu p_{\max})/\kappa_{\min}$. Moreover, $L_j^2 \preceq \kappa_j L_j$, and it follows that

$$\sum_{j=1}^n y_j^2 L_j^2 \preceq 9e^{8b}\nu^2 \sum_{j=1}^n \frac{p_j^2}{\kappa_j^2(\kappa_j-1)^2}\kappa_j L_j \preceq \frac{9e^{8b}\nu^2 p_{\max}}{\kappa_{\min}} L\,,$$

where we used the fact that $L = (p_j/(\kappa_j(\kappa_j-1)))\sum_{j=1}^n L_j$, for L defined in (10). Using $\lambda_d(L) = np/(\beta(d-1))$ from (11), it follows that $\|\sum_{j=1}^n \mathbb{E}_\theta[y_j^2 Y_j^2]\| \leq \frac{9e^{8b}\nu^2 p_{\max}}{\kappa_{\min}}\frac{np}{\beta(d-1)}$. By the matrix Bernstien inequality, with probability at least $1-d^{-3}$,

$$\begin{aligned} \|H(\theta) - \mathbb{E}[H(\theta)]\| &\leq 12e^{4b}\nu\sqrt{\frac{p_{\max}}{\kappa_{\min}}\frac{np}{\beta(d-1)}\log d} + \frac{8e^{4b}\nu p_{\max}\log d}{\kappa_{\min}} \\ &\leq 16e^{4b}\nu\sqrt{\frac{p_{\max}}{\kappa_{\min}}\frac{np}{\beta(d-1)}\log d}\,, \end{aligned} \tag{39}$$

where the last inequality follows from the assumption on $n\kappa_{\min}$ given in (14).

7.5 Proof of Lemma 16

Claim (30)**:** Since providing a lower bound on $\mathbb{P}_\theta\left[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\right]$ for arbitrary θ is challenging, we construct a new set of parameters $\{\widetilde{\theta}_j\}_{j\in[d]}$ from the original θ. These new parameters are constructed such that it is both easy to compute the probability and also provides a lower bound on the original distribution. Define $\widetilde{\alpha}_{i,i',\ell,\theta}$ as

$$\widetilde{\alpha}_{i,i',\ell,\theta} \equiv \max_{\ell'\in[\ell]} \max_{\substack{\Omega\subseteq S\setminus\{i,i'\} \\ :|\Omega|=\kappa-\ell'}} \left\{ \frac{\exp(\theta_i)+\exp(\theta_{i'})}{\left(\sum_{j\in\Omega}\exp(\theta_j)\right)/|\Omega|} \right\}, \tag{40}$$

and $\alpha_{i,i',\ell,\theta} = \left\lceil \widetilde{\alpha}_{i,i',\ell,\theta} \right\rceil$. For ease of notation we remove the subscript from α and $\widetilde{\alpha}$. We denote the sum of the weights by $W \equiv \sum_{j\in S}\exp(\theta_j)$. We define a new set of parameters $\{\widetilde{\theta}_j\}_{j\in S}$:

$$\widetilde{\theta}_j = \begin{cases} \log(\widetilde{\alpha}/2) & \text{for } j = i \text{ or } i'\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Similarly define $\widetilde{W} \equiv \sum_{j\in S}\exp(\widetilde{\theta}_j) = \kappa - 2 + \widetilde{\alpha}$. We have,

$$\begin{aligned}
&\mathbb{P}_\theta\Big[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\Big] \\
&= \sum_{\substack{j_1\in S\\ j_1\neq i,i'}} \left(\frac{\exp(\theta_{j_1})}{W} \sum_{\substack{j_2\in S\\ j_2\neq i,i', j_1}} \left(\frac{\exp(\theta_{j_2})}{W-\exp(\theta_{j_1})} \cdots \left(\sum_{\substack{j_\ell\in S\\ j_\ell\neq i,i',\\ j_1,\cdots,j_{\ell-1}}} \frac{\exp(\theta_{j_\ell})}{W-\sum_{k=j_1}^{j_{\ell-1}}\exp(\theta_k)}\right)\cdots\right)\right) \\
&= \sum_{\substack{j_1\in S\\ j_1\neq i,i'}} \left(\frac{\exp(\theta_{j_1})}{W-\exp(\theta_{j_1})} \cdots \sum_{\substack{j_{\ell-1}\in S\\ j_{\ell-1}\neq i,i',\\ j_1,\cdots,j_{\ell-2}}} \left(\frac{\exp(\theta_{j_{\ell-1}})}{W-\sum_{k=j_1}^{j_{\ell-1}}\exp(\theta_k)} \sum_{\substack{j_\ell\in S\\ j_\ell\neq i,i',\\ j_1,\cdots,j_{\ell-1}}} \left(\frac{\exp(\theta_{j_\ell})}{W}\right)\cdots\right)\right)
\end{aligned} \tag{41}$$

Consider the second-last summation term in the above equation and let $\Omega_\ell = S\setminus\{i,i',j_1,\ldots,j_{\ell-2}\}$. Observe that, $|\Omega_\ell| = \kappa-\ell$ and from equation (40), $\frac{\exp(\theta_i)+\exp(\theta_{i'})}{\sum_{j\in\Omega_\ell}\exp(\theta_j)} \le \frac{\widetilde{\alpha}}{\kappa-\ell}$. We have,

$$\begin{aligned}
&\sum_{j_{\ell-1}\in\Omega_\ell} \frac{\exp(\theta_{j_{\ell-1}})}{W-\sum_{k=j_1}^{j_{\ell-1}}\exp(\theta_k)} \\
&= \sum_{j_{\ell-1}\in\Omega_\ell} \frac{\exp(\theta_{j_{\ell-1}})}{W-\sum_{k=j_1}^{j_{\ell-2}}\exp(\theta_k)-\exp(\theta_{j_{\ell-1}})} \\
&\ge \frac{\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})}{W-\sum_{k=j_1}^{j_{\ell-2}}\exp(\theta_k)-\big(\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})\big)/|\Omega_\ell|} \qquad (42)\\
&= \frac{\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})}{\exp(\theta_i)+\exp(\theta_{i'})+\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})-\big(\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})\big)/|\Omega_\ell|} \\
&= \left(\frac{\exp(\theta_i)+\exp(\theta_{i'})}{\sum_{j_{\ell-1}\in\Omega_\ell}\exp(\theta_{j_{\ell-1}})}+1-\frac{1}{\kappa-\ell}\right)^{-1} \\
&\ge \left(\frac{\widetilde{\alpha}}{\kappa-\ell}+1-\frac{1}{\kappa-\ell}\right)^{-1} \qquad (43)\\
&= \frac{\kappa-\ell}{\widetilde{\alpha}+\kappa-\ell-1} = \sum_{j_{\ell-1}\in\Omega_\ell}\frac{\exp(\widetilde{\theta}_{j_{\ell-1}})}{\widetilde{W}-\sum_{k=j_1}^{j_{\ell-1}}\exp(\widetilde{\theta}_k)}\,, \qquad (44)
\end{aligned}$$

where (42) follows from the Jensen's inequality and the fact that for any $c>0$, $0<x<c$, $\frac{x}{c-x}$ is convex in x. Equation (43) follows from the definition of $\widetilde{\alpha}_{i,i',\ell,\theta}$, (40), and the fact that $|\Omega_\ell| = \kappa-\ell$. Equation (44) uses the definition of $\{\widetilde{\theta}_j\}_{j\in S}$.

Consider $\{\Omega_{\widetilde{\ell}}\}_{2\le\widetilde{\ell}\le\ell-1}$, $|\Omega_{\widetilde{\ell}}| = \kappa-\widetilde{\ell}$, corresponding to the subsequent summation terms in (41). Observe that $\frac{\exp(\theta_i)+\exp(\theta_{i'})}{\sum_{j\in\Omega_{\widetilde{\ell}}}\exp(\theta_j)} \le \alpha/|\Omega_{\widetilde{\ell}}|$. Therefore, each summation term in equation (41) can be

lower bounded by the corresponding term where $\{\theta_j\}_{j\in S}$ is replaced by $\{\widetilde{\theta}_j\}_{j\in S}$. Hence, we have

$$\begin{aligned}
&\mathbb{P}_\theta\Big[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\Big] \\
&\ge \sum_{\substack{j_1\in S\\ j_1\neq i,i'}} \left(\frac{\exp(\widetilde{\theta}_{j_1})}{\widetilde{W}-\exp(\widetilde{\theta}_{j_1})} \cdots \sum_{\substack{j_{\ell-1}\in S\\ j_{\ell-1}\neq i,i',\\ j_1,\cdots,j_{\ell-2}}} \left(\frac{\exp(\widetilde{\theta}_{j_{\ell-1}})}{\widetilde{W}-\sum_{k=j_1}^{j_{\ell-1}}\exp(\widetilde{\theta}_k)} \sum_{\substack{j_\ell\in S\\ j_\ell\neq i,i',\\ j_1,\cdots,j_{\ell-1}}} \left(\frac{\exp(\theta_{j_\ell})}{W}\right)\cdots\right)\right) \\
&\ge e^{-4b} \sum_{\substack{j_1\in S\\ j_1\neq i,i'}} \left(\frac{\exp(\widetilde{\theta}_{j_1})}{\widetilde{W}-\exp(\widetilde{\theta}_{j_1})} \cdots \sum_{\substack{j_{\ell-1}\in S\\ j_{\ell-1}\neq i,i',\\ j_1,\cdots,j_{\ell-2}}} \left(\frac{\exp(\widetilde{\theta}_{j_{\ell-1}})}{\widetilde{W}-\sum_{k=j_1}^{j_{\ell-1}}\exp(\widetilde{\theta}_k)} \sum_{\substack{j_\ell\in S\\ j_\ell\neq i,i',\\ j_1,\cdots,j_{\ell-1}}} \left(\frac{\exp(\widetilde{\theta}_{j_\ell})}{\widetilde{W}}\right)\cdots\right)\right) \\
&= \big(e^{-4b}\big)\mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\Big]\,.
\end{aligned} \tag{45}$$

The second inequality uses $\frac{\exp(\theta_i)}{W} \ge e^{-2b}/\kappa$ and $\frac{\exp(\widetilde{\theta}_i)}{\widetilde{W}} \le e^{2b}/\kappa$. Observe that $\exp(\widetilde{\theta}_j) = 1$ for all $j\neq i,i'$ and $\exp(\widetilde{\theta}_i)+\exp(\widetilde{\theta}_{i'}) = \widetilde{\alpha} \le \lceil\widetilde{\alpha}\rceil = \alpha \ge 1$. Therefore, we have

$$\begin{aligned}
\mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i), \sigma^{-1}(i') > \ell\Big] &= \binom{\kappa-2}{\ell}\frac{\ell!}{(\kappa-2+\widetilde{\alpha})(\kappa-2+\widetilde{\alpha}-1)\cdots(\kappa-2+\widetilde{\alpha}-(\ell-1))} \\
&\ge \frac{(\kappa-2)!}{(\kappa-\ell-2)!}\frac{1}{(\kappa+\alpha-2)(\kappa+\alpha-3)\cdots(\kappa+\alpha-(\ell+1))} \\
&\ge \frac{(\kappa-\ell+\alpha-2)(\kappa-\ell+\alpha-3)\cdots(\kappa-\ell-1)}{(\kappa+\alpha-2)(\kappa+\alpha-3)\cdots(\kappa-1)} \\
&\ge \frac{(\kappa-\ell)(\kappa-\ell-1)}{\kappa(\kappa-1)}\left(1-\frac{\ell}{\kappa+1}\right)^{\alpha-2}.
\end{aligned} \tag{46}$$

Claim (30) follows by combining Equations (45) and (46) and using the fact that $\alpha \le 2e^{2b}$.

Claim (31)**:** Define,

$$\widetilde{\alpha}_{\ell,\theta} \equiv \min_{i\in S}\min_{\ell'\in[\ell]}\min_{\substack{\Omega\in S\setminus\{i\}\\ :|\Omega|=\kappa-\ell'+1}} \left\{\frac{\exp(\theta_i)}{\big(\sum_{j\in\Omega}\exp(\theta_j)\big)/|\Omega|}\right\}. \tag{47}$$

Also, define $\alpha_{\ell,\theta} \equiv \lfloor\widetilde{\alpha}_{\ell,\theta}\rfloor$. Note that $\alpha_{\ell,\theta}\ge 0$ and $\widetilde{\alpha}_{\ell,\theta} \le e^{2b}$. We denote the sum of the weights by $W \equiv \sum_{j\in S}\exp(\theta_j)$. Analogous to the proof of claim (30), we define the new set of parameters $\{\widetilde{\theta}_j\}_{j\in S}$:

$$\widetilde{\theta}_j = \begin{cases} \log(\widetilde{\alpha}_{\ell,\theta}) & \text{for } j = i\,,\\ 0 & \text{otherwise}\,. \end{cases}$$

Similarly define $\widetilde{W} \equiv \sum_{j\in S}\exp(\widetilde{\theta}_j) = \kappa - 1 + \widetilde{\alpha}_{\ell,\theta}$. Using the techniques similar to the ones used in proof of claim (30), we have,

$$\mathbb{P}_\theta\Big[\sigma^{-1}(i) = \ell\Big] \le e^{4b}\mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i) = \ell\Big]\,. \tag{48}$$

Observe that $\exp(\widetilde{\theta}_j) = 1$ for all $j \neq i$ and $\exp(\widetilde{\theta}_i) = \widetilde{\alpha}_{\ell,\theta} \geq \lfloor \widetilde{\alpha}_{\ell,\theta} \rfloor = \alpha_{\ell,\theta} \geq 0$. Therefore, we have

$$\begin{aligned}
\mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i) = \ell\Big] &= \binom{\kappa-1}{\ell-1} \frac{\widetilde{\alpha}_{\ell,\theta}(\ell-1)!}{(\kappa-1+\widetilde{\alpha}_{\ell,\theta})(\kappa-2+\widetilde{\alpha}_{\ell,\theta})\cdots(\kappa-\ell+\widetilde{\alpha}_{\ell,\theta})} \\
&\leq \frac{(\kappa-1)!}{(\kappa-\ell)!} \frac{e^{2b}}{(\kappa-1+\alpha_{\ell,\theta})(\kappa-2+\alpha_{\ell,\theta})\cdots(\kappa-\ell+\alpha_{\ell,\theta})} \\
&\leq \frac{e^{2b}}{\kappa}\left(1 - \frac{\ell}{\kappa+\alpha_{\ell,\theta}}\right)^{\alpha_{\ell,\theta}-1} \leq \frac{e^{2b}}{\kappa-\ell}\,. \qquad (49)
\end{aligned}$$

Claim 31 follows by combining Equations (48) and (49).

Claim (32)**:** Again, we construct a new set of parameters $\{\widetilde{\theta}_j\}_{j\in[d]}$ from the original θ using $\widetilde{\alpha}_{\ell,\theta}$ defined in (47):

$$\widetilde{\theta}_j = \begin{cases} \log(\widetilde{\alpha}_{\ell,\theta}) & \text{for } j \in \{i, i'\}\,, \\ 0 & \text{otherwise}\,. \end{cases}$$

Similarly define $\widetilde{W} \equiv \sum_{j\in S}\exp(\widetilde{\theta}_j) = \kappa - 2 + 2\widetilde{\alpha}_{\ell,\theta}$. Using the techniques similar to the ones used in proof of claim (30), we have,

$$\mathbb{P}_{\theta}\Big[\sigma^{-1}(i) = \ell_1, \sigma^{-1}(i') = \ell_2\Big] \leq e^{8b}\mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i) = \ell_1, \sigma^{-1}(i') = \ell_2\Big] \qquad (50)$$

Observe that $\exp(\widetilde{\theta}_j) = 1$ for all $j \neq i, i'$ and $\exp(\widetilde{\theta}_i) = \exp(\widetilde{\theta}_{i'}) = \widetilde{\alpha}_{\ell,\theta} \geq \lfloor \widetilde{\alpha} \rfloor_{\ell,\theta} = \alpha_{\ell,\theta} \geq 0$. Therefore, we have

$$\begin{aligned}
&= \mathbb{P}_{\widetilde{\theta}}\Big[\sigma^{-1}(i) = \ell_1, \sigma^{-1}(i') = \ell_2\Big] \\
&= \Bigg(\frac{\binom{\kappa-2}{\ell_2-2}\widetilde{\alpha}^2_{\ell,\theta}(\ell_2-2)!}{(\kappa-2+2\widetilde{\alpha}_{\ell,\theta})(\kappa-1+2\widetilde{\alpha}_{\ell,\theta})\cdots(\kappa-2+2\widetilde{\alpha}_{\ell,\theta}-(\ell_1-1))} \\
&\quad \frac{1}{(\kappa-2+\widetilde{\alpha}_{\ell,\theta}-(\ell_1-1))\cdots(\kappa-2+\widetilde{\alpha}_{\ell,\theta}-(\ell_2-2))}\Bigg) \\
&\leq \frac{(\kappa-2)!}{(\kappa-\ell_2)!}\frac{e^{4b}}{(\kappa-2)(\kappa-1)\cdots(\kappa-\ell_1-1)(\kappa-\ell_1-1)\cdots(\kappa-\ell_2)} \\
&\leq \frac{e^{4b}}{(\kappa-\ell_1-1)(\kappa-\ell_2)}\,. \qquad (51)
\end{aligned}$$

Claim 32 follows by combining Equations (50) and (51).

7.6 Proof of Theorem 12

Let $H(\theta) \in \mathcal{S}^d$ be Hessian matrix such that $H_{ii'}(\theta) = \frac{\partial^2 \mathcal{L}_{\mathrm{RB}}(\theta)}{\partial\theta_i \partial\theta_{i'}}$. The Fisher information matrix is defined as $I(\theta) = -\mathbb{E}_\theta[H(\theta)]$. From lemma 1, $\mathcal{L}_{\mathrm{RB}}(\theta)$ is concave. This implies that $I(\theta)$ is positive-semidefinite and from (22) its smallest eigenvalue is zero with all-ones being the corresponding eigenvector. Fix any unbiased estimator $\widehat{\theta}$ of $\theta \in \Omega_b$. Since, $\widehat{\theta} \in \mathcal{U}$, $\widehat{\theta} - \theta$ is orthogonal to $\mathbf{1}$. The

Cramer-Rao lower bound then implies that $\mathbb{E}[\|\widehat{\theta} - \theta^*\|^2] \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\theta))}$. Taking supremum over both sides gives

$$\sup_\theta \mathbb{E}[\|\widehat{\theta} - \theta^*\|^2] \geq \sup_\theta \sum_{i=2}^d \frac{1}{\lambda_i(I(\theta))} \geq \sum_{i=2}^d \frac{1}{\lambda_i(I(\mathbf{0}))}\,.$$

In the following, we will show that

$$\begin{aligned}
I(\mathbf{0}) &= -\mathbb{E}_\theta[H(\mathbf{0})] \preceq \sum_{j=1}^n \sum_{a=1}^{\ell_j} \frac{m_{j,a} - \eta_{j,a}}{\kappa_j(\kappa_j - 1)} \sum_{i<i'\in S_j} (e_i - e_{i'})(e_i - e_{i'})^\top \\
&\preceq \max_{j,a}\big\{m_{j,a} - \eta_{j,a}\big\}\, L\,. \qquad (52)
\end{aligned}$$

Using Jensen's inequality, we have $\sum_{i=2}^d \frac{1}{\lambda_i(I(\mathbf{0}))} \geq \frac{(d-1)^2}{\sum_{i=2}^d \lambda_i(I(\mathbf{0}))} = \frac{(d-1)^2}{\mathrm{Tr}(I(\mathbf{0}))}$. From (52), we have $\mathrm{Tr}(I(\mathbf{0})) \leq \sum_{j,a}(m_{j,a} - \eta_{j,a})$. From (52), we have $\sum_{i=2}^d 1/\lambda_i(I(\mathbf{0})) \geq (1/\max\{m_{j,a} - \eta_{j,a}\})\sum_{i=1}^d 1/\lambda_i(L)$. This proves the desired claim.

Now we are left to show claim (52). Consider a rank-breaking edge $e_{j,a}$. Using notations defined in lemma 15, in particular Equation (36), and omitting subscript $\{j, a\}$ whenever it is clear from the context, we have, for any $i \in V(e_{j,a})$,

$$\frac{\partial^2 \mathbb{P}_\theta(e_{j,a})}{\partial^2\theta_i} = \begin{cases} \sum_\sigma\big(-A_\sigma B_\sigma e^{\theta_i} + A_\sigma(B_\sigma)^2 e^{2\theta_i} + A_\sigma C_\sigma e^{\theta_i}\big) & \text{if } i \in B(e_{j,a}) \\ \sum_\sigma\big(A_\sigma - 3A_\sigma B_{\sigma,i}e^{\theta_i} + A_\sigma C_{\sigma,i})e^{2\theta_i} + A_\sigma(B_{\sigma,i})^2 e^{2\theta_i}\big) & \text{if } i \in T(e_{j,a})\,, \end{cases}$$

and using (37), we have

$$\frac{\partial^2 \log \mathbb{P}_\theta(e_{j,a})}{\partial^2\theta_i}\Big|_{\theta=\mathbf{0}} = \begin{cases} \big((C_\sigma - B_\sigma)\big)_{\theta=\mathbf{0}} & \text{if } i \in B(e_{j,a}) \\ \big(\frac{1}{m_{j,a}!}\sum_\sigma\big(C_{\sigma,i} - B_{\sigma,i} + (B_{\sigma,i})^2\big) - \big(\sum_\sigma \frac{B_{\sigma,i}}{m_{j,a}!}\big)^2\big)_{\theta=\mathbf{0}} & \text{if } i \in T(e_{j,a})\,, \end{cases}$$

where $\sigma \in \Lambda_{T(e_{j,a})}$ and the subscript $\theta = 0$ indicates the the respective quantities are evaluated at $\theta = 0$. From the definitions given in (36), for $\theta = \mathbf{0}$, we have $B_\sigma - C_\sigma = \sum_{u=0}^{m-1}\frac{(r-u-1)}{(r-u)^2}$ and, $\sum_\sigma(B_{\sigma,i} - C_{\sigma,i})/(m!) = \frac{1}{m}\sum_{u=0}^{m-1}\frac{(m-u)(r-u-1)}{(r-u)^2}$. Also, $\sum_\sigma B_{\sigma,i}/(m!) = \frac{1}{m}\sum_{u=0}^{m-1}\frac{m-u}{r-u}$ and $\sum_\sigma(B_{\sigma,i})^2/(m!) = \frac{1}{m}\sum_{u=0}^{m-1}\big(\sum_{u'=0}^{u}\frac{1}{r-u'}\big)^2$. Combining all these and, using $\mathbb{P}_{\theta=\mathbf{0}}[i \in T(e_{j,a})] = m/\kappa$ and $\mathbb{P}_{\theta=\mathbf{0}}[i \in B(e_{j,a})] = (r-m)/\kappa$, and after following some algebra, we have for any $i \in S_j$,

$$\begin{aligned}
&-\mathbb{E}\left[\frac{\partial^2 \log \mathbb{P}_\theta(e_{j,a})}{\partial^2\theta_i}\Big|_{\theta=\mathbf{0}}\right] \\
&= \frac{1}{\kappa}\Bigg(m - \sum_{u=0}^{m-1}\frac{1}{r-u} - \frac{1}{m}\sum_{u=0}^{m-1}\frac{u(m-u)}{(r-u)^2} - \frac{1}{m}\sum_{u=0}^{m-2}\frac{2u}{r-u}\Bigg(\sum_{u'>u}^{m-1}\frac{m-u'}{r-u'}\Bigg)\Bigg) \\
&= \frac{m_{j,a} - \eta_{j,a}}{\kappa_j}\,, \qquad (53)
\end{aligned}$$

where $\eta_{j,a}$ is defined in (16). Since row-sums of $H(\theta)$ are zeroes, (22), and for $\theta = \mathbf{0}$, all the items are exchangeable, we have for any $i \neq i' \in S_j$,

$$\mathbb{E}\left[\frac{\partial^2 \log \mathbb{P}_\theta(e_{j,a})}{\partial\theta_i\partial\theta_{i'}}\Big|_{\theta=\mathbf{0}}\right] = \frac{m_{j,a} - \eta_{j,a}}{\kappa_j(\kappa_j - 1)}\,,$$

The claim (52) follows from the expression of $H(\theta)$, Equation (22).

To verify (53), observe that $(r-m)(B_\sigma - C_\sigma) + m(\sum_\sigma B_{\sigma,i}/(m!)) = m - \sum_{u=0}^{m-1} \frac{1}{r-u}$. And,

$$\begin{aligned}
&\frac{1}{m}\Big(\sum_{u=0}^{m-1}\frac{m-u}{r-u}\Big)^2 - \sum_{u=0}^{m-1}\Big(\sum_{u'=0}^{u}\frac{1}{r-u'}\Big)^2 \\
&= \sum_{u=0}^{m-1}\left(\frac{(m-u)^2}{m(r-u)^2} - \frac{m-u}{(r-u)^2}\right) + \sum_{0\le u<u'\le m-1}\left(\frac{2(m-u)(m-u')}{m(r-u)(r-u')} - \frac{2(m-u')}{(r-u)(r-u')}\right) \\
&= \sum_{u=0}^{m-1}\frac{-u(m-u)}{m(r-u)^2} + \sum_{0\le u<u'\le m-1}\frac{-2u(m-u')}{m(r-u)(r-u')}\,.
\end{aligned}$$

7.7 Tightening of Lemma 15

Recall that $\mathbb{P}_\theta(e_{j,a})$ is same as probability of $\mathbb{P}_\theta[T(e_{j,a}) \succ B(e_{j,a})]$ that is the probability that an agent ranks $T(e_{j,a})$ items above $B(e_{j,a})$ items when provided with a set comprising $V(e_{j,a})$ items. As earlier, for brevity of notations, we omit subscript $\{j,a\}$ whenever it is clear from the context. For $m = 1$ or 2, it is easy to check that all off-diagonal elements in hessian matrix of $\log \mathbb{P}_\theta(e)$ are non-negative. However, since number of terms in summation in $\mathbb{P}_\theta(e)$ grows as $m!$, for $m \ge 3$ the straight-forward approach becomes too complex. Below, we derive expressions for cross-derivatives in hessian, for general m, using alternate definition (sorting of independent exponential r.v.'s in increasing order) of PL model, where the number of terms grow only as 2^m. However, we are unable to analytically prove that the cross-derivatives are non-negative for $m > 2$. Feeding these expressions in MATLAB and using symbolic computation, for $m = 3$, we can simplify these expressions and it turns out that they are sum of only positive numbers. For $m = 4$, with limited computational power it becomes intractable. We believe that it should hold for any value of $m < r$. Using (29), we need to check only for cross-derivatives for the case when $i \ne i' \in T(e_{j,a})$ or $i \in T(e_{j,a}), i' \in B(e_{j,a})$. Since, minimum of exponential random variables is an exponential random variable, we can assume that $|B(e_{j,a})| = 1$ that is $r = m+1$. Define $\lambda_i \equiv e^{\theta_i}$. Without loss of generality, assume $T(e_{j,a}) = \{2, \cdots, m+1\}$ and $B(e_{j,a}) = \{1\}$. Define $C_x = \prod_{i=3}^{m+1}(1-e^{-\lambda_i x})$. Then, using the alternate definition of the PL model, we have, $\mathbb{P}_\theta(e) = \int_0^\infty C_x(1-e^{-\lambda_2 x})\lambda_1 e^{-\lambda_1 x}dx$. Following some algebra, $\frac{\partial^2 \log \mathbb{P}_\theta(e)}{\partial\theta_1\partial\theta_2} \ge 0$ is equivalent to $A_1 \ge 0$, where $A_1 \equiv$

$$\left(\int C_x\big(xe^{-\lambda_1 x} - xe^{-\lambda x}\big)dx\right)\left(\int C_x x e^{-\lambda x}dx\right) - \left(\int C_x(e^{\lambda_1 x} - e^{-\lambda x})dx\right)\left(\int C_x x^2 e^{-\lambda x}dx\right),$$

where all integrals are from 0 to ∞ and, $\lambda \equiv \lambda_1 + \lambda_2$. Consider A_1 as a function of λ_1. Since $A_1(\lambda_1) = 0$ for $\lambda_1 = \lambda$, showing $\partial A_1/\partial\lambda_1 \le 0$ for $0 \le \lambda_1 \le \lambda$ would suffice. Following some algebra, and using $\lambda_1 \le \lambda$, $\partial A_1/\partial\lambda_1 \le 0$ is equivalent to $A_2(\lambda_1) \equiv \big(\int_0^\infty C_x x e^{-\lambda_1 x}\big)/\big(\int_0^\infty C_x x^2 e^{-\lambda_1 x}\big)$ being monotonically non-decreasing in λ_1. To further simplify the condition, define $f^{(0)}(y) = 1/y^2$, $g^{(0)}(y) = 1/y^3$ and, $f^{(1)}(y) = f^{(0)}(y) - f^{(0)}(y+\lambda_3)$, and recursively $f^{(m-1)}(y) = f^{(m-2)}(y) - f^{(m-2)}(y+\lambda_{m+1})$. Similarly define $g^{(0)}, \cdots, g^{(m-1)}$. Using these recursively defined functions,

$$\begin{aligned}
2A_2(\lambda_1) &= \frac{f^{(m-1)}(\lambda_1)}{g^{(m-1)}(\lambda_1)}\,, \\
\text{for } m=3,\ 2A_2(\lambda_1) &= \frac{\lambda_1^{-2} - (\lambda_1+\lambda_3)^{-2} - (\lambda_1+\lambda_4)^{-2} + (\lambda_1+\lambda_3+\lambda_4)^{-2}}{\lambda_1^{-3} - (\lambda_1+\lambda_3)^{-3} - (\lambda_1+\lambda_4)^{-3} + (\lambda_1+\lambda_3+\lambda_4)^{-3}}\,.
\end{aligned}$$

Therefore, we need to show that $A_2(\lambda_1)$ is monotonically non-decreasing in $\lambda_1 \ge 0$ for any non-negative $\lambda_3, \cdots, \lambda_m$, and that would suffice to prove that the cross-derivatives arising from $i \in T(e_{j,a}), i' \in B(e_{j,a})$ are non-negative.

For cross-derivatives arising from $i \ne i' \in T(e_{j,a})$, define $B_x = \prod_{i=4}^{m+1}(1-e^{\lambda_i x})e^{-\lambda_1 x}$. $\frac{\partial^2 \log \mathbb{P}_\theta(e)}{\partial\theta_2\partial\theta_3} \ge 0$ is equivalent to $A_3 \ge 0$, where $A_3 \equiv$

$$\begin{aligned}
&\left(\int B_x(1-e^{-\lambda_2 x})(1-e^{-\lambda_3 x})dx\right)\left(\int B_x x^2 e^{-(\lambda_2+\lambda_3)x}dx\right) \\
&-\left(\int B_x(1-e^{-\lambda_2 x})xe^{-\lambda_3 x}dx\right)\left(\int B_x(1-e^{-\lambda_3 x})xe^{-\lambda_2 x}dx\right),
\end{aligned}$$

where all integrals are from 0 to ∞. For $m = 3$, using MATLAB one can check that the above stated conditions hold true. Therefore both types of cross-derivatives are non-negative.

Acknowledgements

This work is supported by NSF SaTC award CNS-1527754, and NSF CISE award CCF-1553452.

References

A. Agarwal, P. L. Bartlett, and J. C. Duchi. Oracle inequalities for computationally adaptive model selection. *arXiv preprint arXiv:1208.0129*, 2012.

A. Ali and M. Meilă. Experiments with kemeny ranking: What works when? *Mathematical Social Sciences*, 64(1):28–40, 2012.

H. Azari Soufiani, D. C. Parkes, and L. Xia. Random utility theory for social choice. In *NIPS*, pages 126–134, 2012.

H. Azari Soufiani, W. Chen, D. C Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems 26*, pages 2706–2714, 2013.

H. Azari Soufiani, D. Parkes, and L. Xia. Computing parametric ranking models via rank-breaking. In *Proceedings of The 31st International Conference on Machine Learning*, pages 360–368, 2014.

Michael A Bender, Martin Farach-Colton, Giridhar Pemmasani, Steven Skiena, and Pavel Sumazin. Lowest common ancestors in trees and directed acyclic graphs. *Journal of Algorithms*, 57(2):75–94, 2005.

N. Betzler, R. Bredereck, and R. Niedermeier. Theoretical and empirical evaluation of data reduction for exact kemeny rank aggregation. *Autonomous Agents and Multi-Agent Systems*, 28(5): 721–748, 2014.

O. Bousquet and L. Bottou. The tradeoffs of large scale learning. In *Advances in neural information processing systems*, pages 161–168, 2008.

V. Chandrasekaran and M. I. Jordan. Computational and statistical tradeoffs via convex relaxation. *Proceedings of the National Academy of Sciences*, 110(13):E1181–E1190, 2013.

Y. Chen and C. Suh. Spectral mle: Top-k rank aggregation from pairwise comparisons. *arXiv:1504.07218*, 2015.

Artur Czumaj, Mirosław Kowaluk, and Andrzej Lingas. Faster algorithms for finding lowest common ancestors in directed acyclic graphs. *Theoretical Computer Science*, 380(1-2):37–46, 2007.

Y. Deshpande and A. Montanari. Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. *arXiv preprint arXiv:1502.06590*, 2015.

L. R. Ford Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957.

K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

B. Hajek, S. Oh, and J. Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems 27*, pages 1475–1483, 2014.

T. P. Hayes. A large-deviation inequality for vector-valued martingales. *Combinatorics, Probability and Computing*, 2005.

D. R. Hunter. Mm algorithms for generalized bradley-terry models. *Ann. of Stat.*, pages 384–406, 2004.

David R Hunter and Kenneth Lange. Rejoinder. *Journal of Computational and Graphical Statistics*, 9(1):52–59, 2000.

T. Kamishima. Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588. ACM, 2003.

A. Khetan and S. Oh. Data-driven rank breaking for efficient rank aggregation. In *International Conference on Machine Learning*, 2016.

M. Lucic, M. I. Ohannessian, A. Karbasi, and A. Krause. Tradeoffs for space, time, data and risk in unsupervised learning. In *AISTATS*, 2015.

L. Maystre and M. Grossglauser. Fast and accurate inference of plackett-luce models. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

R. Meka, A. Potechin, and A. Wigderson. Sum-of-squares lower bounds for planted clique. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 87–96. ACM, 2015.

S. Negahban, S. Oh, and D. Shah. Rank centrality: Ranking from pair-wise comparisons. preprint arXiv:1209.1688, 2014.

A. Prékopa. Logarithmic concave measures and related topics. In *Stochastic programming*, 1980.

N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *arXiv:1505.01462*, 2015a.

N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *arXiv preprint arXiv:1510.05610*, 2015b.

S. Shalev-Shwartz and N. Srebro. Svm optimization: inverse dependence on training set size. In *Proceedings of the 25th international conference on Machine learning*, pages 928–935. ACM, 2008.

G. Simons and Y. Yao. Asymptotics when the number of parameters tends to infinity in the bradley-terry model for paired comparisons. *The Annals of Statistics*, 27(3):1041–1060, 1999.

E. Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

Journal of Machine Learning Research 19 (2018) 1-44 Submitted 9/16; Revised 7/18; Published 8/18

Gradient Descent Learns Linear Dynamical Systems

Moritz Hardt M@MRTZ.ORG
Department of Electrical Engineering and Computer Science
University of California, Berkeley

Tengyu Ma TENGYUMA@STANFORD.EDU
Facebook AI Research

Benjamin Recht BRECHT@BERKELEY.EDU
Department of Electrical Engineering and Computer Science
University of California, Berkeley

Editor: Sujay Sanghavi

Abstract

We prove that stochastic gradient descent efficiently converges to the global optimizer of the maximum likelihood objective of an unknown linear time-invariant dynamical system from a sequence of noisy observations generated by the system. Even though the objective function is non-convex, we provide polynomial running time and sample complexity bounds under strong but natural assumptions. Linear systems identification has been studied for many decades, yet, to the best of our knowledge, these are the first polynomial guarantees for the problem we consider.

Key-words: non-convex optimization, linear dynamical system, stochastic gradient descent, generalization bounds, time series, over-parameterization

1. Introduction

Many learning problems are by their nature sequence problems where the goal is to fit a model that maps a sequence of input words $x_1, \ldots, x_T$ to a corresponding sequence of observations $y_1, \ldots, y_T$. Text translation, speech recognition, time series prediction, video captioning and question answering systems, to name a few, are all sequence to sequence learning problems. For a sequence model to be both expressive and parsimonious in its parameterization, it is crucial to equip the model with memory thus allowing its prediction at time t to depend on previously seen inputs.

Recurrent neural networks form an expressive class of non-linear sequence models. Through their many variants, such as long-short-term-memory Hochreiter and Schmidhuber (1997), recurrent neural networks have seen remarkable empirical success in a broad range of domains. At the core, neural networks are typically trained using some form of (stochastic) gradient descent. Even though the training objective is non-convex, it is widely observed in practice that gradient descent quickly approaches a good set of model parameters. Understanding the effectiveness of gradient descent for non-convex objectives on theoretical grounds is a major open problem in this area.

If we remove all non-linear state transitions from a recurrent neural network, we are left with the state transition representation of a linear dynamical system. Notwithstanding, the natural training objective for linear systems remains non-convex due to the composition of multiple linear operators in the system. If there is any hope of eventually understanding recurrent neural networks, it will be inevitable to develop a solid understanding of this special case first.

To be sure, linear dynamical systems are important in their own right and have been studied for many decades independently of machine learning within the control theory community. Control theory provides a rich set techniques for identifying and manipulating linear systems. The learning problem in this context corresponds to "linear dynamical system identification". Maximum likelihood estimation with gradient descent is a popular heuristic for dynamical system identification Ljung (1998). In the context of machine learning, linear systems play an important role in numerous tasks. For example, their estimation arises as subroutines of reinforcement learning in robotics Levine and Koltun (2013), location and mapping estimation in robotic systems Durrant-Whyte and Bailey (2006), and estimation of pose from video Rahimi et al. (2005).

In this work, we show that gradient descent efficiently minimizes the maximum likelihood objective of an unknown linear system given noisy observations generated by the system. More formally, we receive noisy observations generated by the following *time-invariant linear system*:

$$\begin{aligned} h_{t+1} &= Ah_t + Bx_t \\ y_t &= Ch_t + Dx_t + \xi_t \end{aligned} \tag{1.1}$$

Here, A, B, C, D are linear transformations with compatible dimensions and we denote by $\Theta = (A, B, C, D)$ the parameters of the system. The vector h_t represents the hidden state of the model at time t. Its dimension n is called the *order* of the system. The stochastic noise variables $\{\xi_t\}$ perturb the output of the system which is why the model is called an *output error model* in control theory. We assume the variables are drawn i.i.d. from a distribution with mean 0 and variance σ^2.

Throughout the paper we focus on *controllable* and *externally stable* systems. A linear system is externally stable (or *equivalently bounded-input bounded-output stable*) if and only if the spectral radius of A, denoted $\rho(A)$, is strictly bounded by 1. Controllability is a mild non-degeneracy assumption that we formally define later. Without loss of generality, we further assume that the transformations B, C and D have bounded Frobenius norm. This can be achieved by a rescaling of the output variables. We assume we have N pairs of sequences (x, y) as training examples,

$$S = \left\{ (x^{(1)}, y^{(1)}), \ldots, (x^{(N)}, y^{(N)}) \right\}.$$

Each input sequence $x \in \mathbb{R}^T$ of length T is drawn from a distribution and y is the corresponding output of the system above generated from an unknown initial state h. We allow the unknown initial state to vary from one input sequence to the next. This only makes the learning problem more challenging.

Our goal is to fit a linear system to the observations. We parameterize our model in exactly the same way as (1.1). That is, for linear mappings $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, the trained model

is defined as:

$$\hat{h}_{t+1} = \hat{A}\hat{h}_t + \hat{B}x_t\,, \qquad \hat{y}_t = \hat{C}\hat{h}_t + \hat{D}x_t \tag{1.2}$$

The *(population) risk* of the model is obtained by feeding the learned system with the correct initial states and comparing its predictions with the ground truth in expectation over inputs and errors. Denoting by $\hat{y}_t$ the t-th prediction of the trained model starting from the correct initial state that generated y_t, and using $\widehat{\Theta}$ as a short hand for $(\hat{A}, \hat{B}, \hat{C}, \hat{D})$, we formally define population risk as:

$$f(\widehat{\Theta}) = \mathop{\mathbb{E}}_{\{x_t\},\{\xi_t\}}\left[\frac{1}{T}\sum_{t=1}^{T}\|\hat{y}_t - y_t\|^2\right] \tag{1.3}$$

Note that even though the prediction $\hat{y}_t$ is generated from the correct initial state, the learning algorithm does not have access to the correct initial state for its training sequences.

While the squared loss objective turns out to be non-convex, it has many appealing properties. Assuming the inputs x_t and errors ξ_t are drawn independently from a Gaussian distribution, the corresponding population objective corresponds to maximum likelihood estimation. In this work, we make the weaker assumption that the inputs and errors are drawn independently from possibly different distributions. The independence assumption is certainly idealized for some learning applications. However, in control applications the inputs can often be chosen by the controller rather than by nature. Moreover, the outputs of the system at various time steps are correlated through the unknown hidden state and therefore not independent even if the inputs are.

1.1 Our results

We show that we can efficiently minimize the population risk using projected stochastic gradient descent. The bulk of our work applies to single-input single-output (SISO) systems meaning that inputs and outputs are scalars $x_t, y_t \in \mathbb{R}$. However, the hidden state can have arbitrary dimension n. Every controllable SISO admits a convenient canonical form called *controllable canonical form* that we formally introduce later in Section 1.7. In this canonical form, the transition matrix A is governed by n parameters $a_1, \ldots, a_n$ which coincide with the coefficients of the characteristic polynomial of A. The minimal assumption under which we might hope to learn the system is that the spectral radius of A is smaller than 1. However, the set of such matrices is non-convex and does not have enough structure for our analysis.[1] We will therefore make additional assumptions. The assumptions we need differ between the case where we are trying to learn A with n parameter system, and the case where we allow ourselves to over-specify the trained model with $n' > n$ parameters. The former is sometimes called proper learning, while the latter is called improper learning. In the improper case, we are essentially able to learn any system with spectral radius less than 1 under a mild separation condition on the roots of the characteristic polynomial. Our assumption in the proper case is stronger and we introduce it next.

1. In both the controllable canonical form and the standard parameterization of the matrix A, the set of matrices with spectral radius less than 1 is not convex.

1.2 Proper learning

Suppose the state transition matrix A has characteristic polynomial $\det(zI - A) = z^n + a_1 z^{n-1} + \cdots + a_n$, which turns out to by and large decide the difficulty of the learning according to our analysis. (In fact, we will parameterize A in a way so that the coefficients of the characteristic polynomials are the parameters of learning problem. See Section 1.7 for the detailed setup.) Consider the corresponding polynomial $q(z) = 1 + a_1 z + a_2 z^2 + \cdots + a_n z^n$ over the complex numbers $\mathbb{C}$.

We will require the state transition matrix satisfy that the image of the unit circle on the complex plane under the polynomial q is contained in the cone of complex numbers whose real part is larger than their absolute imaginary part. Formally, for all $z \in \mathbb{C}$ such that $|z| = 1$, we require that $\Re(q(z)) > |\Im(q(z))|$. Here, $\Re(z)$ and $\Im(z)$ denote the real and imaginary part of z, respectively. We illustrate this condition in Figure 1 on the right for a degree 4 system.

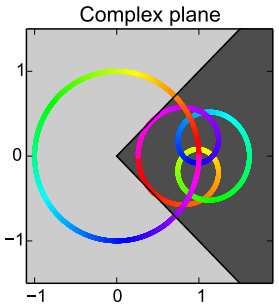


Figure 1: An example of polynomial q that satisfies our assumption. The unit circle is the collection of the inputs of q and the other curve shows the corresponding outputs (with the corresponding colors.) We see the image of the polynomial stays in the wedge which contains all the complex number z satisfying $\Re(q(z)) > |\Im(q(z))|$.

Our assumption has three important implications. First, it implies (via Rouché's theorem) that the spectral radius of A is smaller than 1 and therefore ensures the stability of the system. Second, the vectors satisfying our assumption form a convex set in $\mathbb{R}^n$. Finally, it ensures that the objective function is *weakly quasi-convex*, a condition we introduce later when we show that it enables stochastic gradient descent to make sufficient progress.

We note in passing that our assumption can be satisfied via the ℓ_1-norm constraint $\|a\|_1 \le \sqrt{2}/2$. Moreover, if we pick random Gaussian coefficients with expected norm bounded by $o(1/\sqrt{\log n})$, then the resulting vector will satisfy our assumption with probability $1 - o(1)$. Roughly speaking, the assumption requires the roots of the characteristic polynomial $p(z) = z^n + a_1 z^{n-1} + \cdots + a_n$ are relatively dispersed inside the unit circle. (For comparison, on the other end of the spectrum, the polynomial $p(z) = (z - 0.99)^n$ have all its roots colliding at the same point and doesn't satisfy the assumption.)

Theorem 1.1 (Informal) *Under our assumption, projected stochastic gradient descent, when given N sample sequence of length T, returns parameters $\widehat{\Theta}$ with population risk*

$$\mathbb{E}\, f(\widehat{\Theta}) \le f(\Theta) + O\left(\sqrt{\frac{n^5 + \sigma^2 n^3}{TN}}\right).$$

Here the expectation on LHS is with respect to the randomness of the algorithm. Recall that $f(\Theta)$ is the population risk of the optimal system, and σ^2 refers to the variance of the noise variables. We also assume that the inputs x_t are drawn from a pairwise independent distribution with mean 0 and variance 1. Note, however, that this does not imply

independence of the outputs as these are correlated by a common hidden state. The stated version of our result glosses over the fact that we need our assumption to hold with a small amount of slack; a precise version follows in Section 4. Our theorem establishes a polynomial convergence rate for stochastic gradient descent. Since each iteration of the algorithm only requires a sequence of matrix operations and an efficient projection step, the running time is polynomial, as well. Likewise, the sample requirements are polynomial since each iteration requires only a single fresh example. An important feature of this result is that the error decreases with both the length T and the number of samples N. The dependency on the dimension n, on the other hand, is likely to be quite loose, and tighter bounds are left for future work.

The algorithm requires a (polynomial-time) projection step to a convex set at every iteration (formally defined in Section 4 and Algorithm 1). Computationally, it can be a bottleneck, although it is unlikely to be required in practice and may be an artifact of our analysis.

1.3 The power of over-parameterization

Endowing the model with additional parameters compared to the ground truth turns out to be surprisingly powerful. We show that we can essentially remove the assumption we previously made in proper learning. The idea is simple. If p is the characteristic polynomial of A of degree n. We can find a system of order $n' > n$ such that the characteristic polynomial of its transition matrix becomes $p \cdot p'$ for some polynomial p' of order $n' - n$. This means that to apply our result we only need the polynomial $p \cdot p'$ to satisfy our assumption. At this point, we can choose p' to be an approximation of the inverse p^{-1}. For sufficiently good approximation, the resulting polynomial $p \cdot p'$ is close to 1 and therefore satisfies our assumption. Such an approximation exists generically for $n' = O(n)$ under mild non-degeneracy assumptions on the roots of p. In particular, any small random perturbation of the roots would suffice.

Theorem 1.2 (Informal) *Under a mild non-degeneracy assumption, stochastic gradient descent returns parameters $\widehat{\Theta}$ corresponding to a system of order $n' = O(n)$ with population risk*

$$f(\widehat{\Theta}) \leq f(\Theta) + O\left(\sqrt{\frac{n^5 + \sigma^2 n^3}{TN}}\right),$$

when given N sample sequences of length T.

We remark that the idea we sketched also shows that, in the extreme, improper learning of linear dynamical systems becomes easy in the sense that the problem can be solved using linear regression against the outputs of the system. However, our general result interpolates between the proper case and the regime where linear regression works. We discuss in more details in Section 6.3.

1.4 Multi-input multi-output systems

Both results we saw immediately extend to single-input multi-output (SIMO) systems as the dimensionality of C and D are irrelevant for us. The case of multi-input multi-output

(MIMO) systems is more delicate. Specifically, our results carry over to a broad family of multi-input multi-output systems. However, in general MIMO systems no longer enjoy canonical forms like SISO systems. In Section 7, we introduce a natural generalization of controllable canonical form for MIMO systems and extend our results to this case.

1.5 Related work

System identification is a core problem in dynamical systems and has been studied in depth for many years. The most popular reference on this topic is the text by Ljung Ljung (1998). Nonetheless, the list of non-asymptotic results on identifying linear systems from noisy data is surprisingly short. Several authors have recently tried to estimate the sample complexity of dynamical system identification using machine learning tools Vidyasagar and Karandikar (2008); Campi and Weyer (2002); Weyer and Campi (1999). All of these result are rather pessimistic with sample complexity bounds that are exponential in the degree of the linear system and other relevant quantities. Contrastingly, we prove that gradient descent has an associated polynomial sample complexity in all of these parameters. Moreover, all of these papers only focus on how well empirical risk approximates the true population risk and do not provide guarantees about any algorithmic schemes for minimizing the empirical risk.

The only result to our knowledge which provides polynomial sample complexity for identifying linear dynamical systems is in Shah *et al* Shah et al. (2012). Here, the authors show that if certain *frequency domain* information about the linear dynamical system is observed, then the linear system can be identified by solving a second-order cone programming problem. This result is about improper learning only, and the size of the resulting system may be quite large, scaling as the $(1 - \rho(A))^{-2}$. As we describe in this work, very simple algorithms work in the improper setting when the system degree is allowed to be polynomial in $(1 - \rho(A))^{-1}$. Moreover, it is not immediately clear how to translate the frequency-domain results to the time-domain identification problem discussed above.

Our main assumption about the image of the polynomial $q(z)$ is an appeal to the theory of passive systems. A system is passive if the dot product between the input sequence u_t and output sequence y_t are strictly positive. Physically, this notion corresponds to systems that cannot create energy. For example, a circuit made solely of resistors, capacitors, and inductors would be a passive electrical system. If one added an amplifier to the internals of the system, then it would no longer be passive. The set of passive systems is a subset of the set of stable systems, and the subclass is somewhat easier to work with mathematically. Indeed, Megretski used tools from passive systems to provide a relaxation technique for a family of identification problems in dynamical systems Megretski (2008). His approach is to lower bound a nonlinear least-squares cost with a convex functional. However, he does not prove that his technique can identify any of the systems, even asymptotically. Stoica and Söderström Söderström and Stoica (1982); Stoica and Söderström (1982, 1984) and later Bazanella *et al.* Bazanella et al. (2008); Eckhard and Bazanella (2011) prove the quasi-convexity of a cost function under a passivity condition in the context of system identification, but no sample complexity or global convergence proofs are provided.

1.6 Proof overview

The first important step in our proof is to develop population risk in Fourier domain where it is closely related to what we call *idealized risk*. Idealized risk essentially captures the ℓ_2-difference between the *transfer function* of the learned system and the ground truth. The transfer function is a fundamental object in control theory. Any linear system is completely characterized by its transfer function $G(z) = C(zI - A)^{-1}B$. In the case of a SISO, the transfer function is a rational function of degree n over the complex numbers and can be written as $G(z) = s(z)/p(z)$. In the canonical form introduced in Section 1.7, the coefficients of $p(z)$ are precisely the parameters that specify A. Moreover, $z^n p(1/z) = 1 + a_1 z + a_2 z^2 + \cdots + a_n z^n$ is the polynomial we encountered in the introduction. Under the assumption illustrated earlier, we show in Section 3 that the idealized risk is *weakly quasi-convex* (Lemma 3.3). Quasi-convexity implies that gradients cannot vanish except at the optimum of the objective function; we review this (mostly known) material in Section 2. In particular, this lemma implies that in principle we can hope to show that gradient descent converges to a global optimum. However, there are several important issues that we need to address. First, the result only applies to idealized risk, not our actual population risk objective. Therefore it is not clear how to obtain unbiased gradients of the idealized risk objective. Second, there is a subtlety in even defining a suitable empirical risk objective. The reason is that risk is defined with respect to the correct initial state of the system which we do not have access to during training. We overcome both of these problems. In particular, we design an almost unbiased estimator of the gradient of the idealized risk in Lemma 5.4 and give variance bounds of the gradient estimator (Lemma 5.5).

Our results on improper learning in Section 6 rely on a surprisingly simple but powerful insight. We can extend the degree of the transfer function $G(z)$ by extending both numerator and denominator with a polynomial $u(z)$ such that $G(z) = s(z)u(z)/p(z)u(z)$. While this results in an equivalent system in terms of input-output behavior, it can dramatically change the geometry of the optimization landscape. In particular, we can see that only $p(z)u(z)$ has to satisfy the assumption of our proper learning algorithm. This allows us, for example, to put $u(z) \approx p(z)^{-1}$ so that $p(z)u(z) \approx 1$, hence trivially satisfying our assumption. A suitable inverse approximation exists under light assumptions and requires degree no more than $d = O(n)$. Algorithmically, there is almost no change. We simply run stochastic gradient descent with $n + d$ model parameters rather than n model parameters.

1.7 Preliminaries

For complex matrix (or vector, number) C, we use $\Re(C)$ to denote the real part and $\Im(C)$ the imaginary part, and $\bar{C}$ the conjugate and $C^* = \bar{C}^\top$ its conjugate transpose . We use $|\cdot|$ to denote the absolute value of a complex number c. For complex vector u and v, we use $\langle u, v\rangle = u^* v$ to denote the inner product and $\|u\| = \sqrt{u^* u}$ is the norm of u. For complex matrix A and B with same dimension, $\langle A, B\rangle = \mathrm{tr}(A^* B)$ defines an inner product, and $\|A\|_F = \sqrt{\mathrm{tr}(A^* A)}$ is the Frobenius norm. For a square matrix A, we use $\rho(A)$ to denote the spectral radius of A, that is, the largest absolute value of the elements in the spectrum of A. We use I_n to denote the identity matrix with dimension $n \times n$, and we drop the subscript when it's clear from the context.We let e_i denote the i-th standard basis vector.

A SISO of order n is in *controllable canonical form* if A and B have the following form

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_n & -a_{n-1} & -a_{n-2} & \cdots & -a_1 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{1.4}$$

We will parameterize $\hat{A}, \hat{B}, \hat{C}, \hat{D}$ accordingly. We will write $A = \mathrm{CC}(a)$ for brevity, where a is used to denote the unknown last row $[-a_n, \ldots, -a_1]$ of matrix A. We will use $\hat{a}$ to denote the corresponding training variables for a. Since here B is known, so $\hat{B}$ is no longer a trainable parameter, and is forced to be equal to B. Moreover, C is a row vector and we use $[c_1, \cdots, c_n]$ to denote its coordinates (and similarly for $\hat{C}$).

A SISO is *controllable* if and only if the matrix $[B \mid AB \mid A^2 B \mid \cdots \mid A^{n-1}B]$ has rank n. This statement corresponds to the condition that all hidden states should be reachable from some initial condition and input trajectory. Any controllable system admits a controllable canonical form Heij et al. (2007). For vector $a = [a_n, \ldots, a_1]$, let $p_a(z)$ denote the polynomial

$$p_a(z) = z^n + a_1 z^{n-1} + \cdots + a_n \,. \tag{1.5}$$

When a defines the matrix A that appears in controllable canonical form, then p_a is precisely the characteristic polynomial of A. That is, $p_a(z) = \det(zI - A)$.

2. Gradient descent and quasi-convexity

It is known that under certain mild conditions (stochastic) gradient descent converges even on non-convex functions to local minimum Ge et al. (2015); Lee et al. (2016). Though usually for concrete problems the challenge is to prove that there is no spurious local minimum other than the target solution. Here we introduce a condition similar to the quasi-convexity notion in Hazan et al. (2015), which ensures that any point with vanishing gradient is the optimal solution . Roughly speaking, the condition says that at any point θ the negative of the gradient $-\nabla f(\theta)$ should be positively correlated with direction $\theta^* - \theta$ pointing towards the optimum. Our condition is slightly weaker than that in Hazan et al. (2015) since we only require quasi-convexity and smoothness with respect to the optimum, and this (simple) extension will be necessary for our analysis.

Definition 2.1 (Weak quasi-convexity) *We say an objective function f is τ-*weakly-quasi-convex *(*τ*-WQC) over a domain $\mathcal{B}$ with respect to global minimum θ^* if there is a positive constant $\tau > 0$ such that for all $\theta \in \mathcal{B}$,*

$$\nabla f(\theta)^\top (\theta - \theta^*) \geq \tau(f(\theta) - f(\theta^*)) \,. \tag{2.1}$$

We further say f is Γ-weakly-smooth if for for any point θ, $\|\nabla f(\theta)\|^2 \leq \Gamma(f(\theta) - f(\theta^*))$.

Note that indeed any Γ-smooth function in the usual sense (that is, $\|\nabla^2 f\| \leq \Gamma$) is $O(\Gamma)$-weakly-smooth. For a random vector $X \in \mathbb{R}^n$, we define it's variance to be $\mathrm{Var}[X] = \mathbb{E}[\|X - \mathbb{E}X\|^2]$.

Definition 2.2 *We call $\mathfrak{r}(\theta)$ an unbiased estimator of $\nabla f(\theta)$ with variance V if it satisfies $\mathbb{E}[\mathfrak{r}(\theta)] = \nabla f(\theta)$ and $\mathrm{Var}[\mathfrak{r}(\theta)] \le V$.*

Projected stochastic gradient descent over some closed convex set $\mathcal{B}$ with *learning rate* $\eta > 0$ refers to the following algorithm in which $\Pi_{\mathcal{B}}$ denotes the Euclidean projection onto $\mathcal{B}$:

$$\begin{aligned}
&\textbf{for } k = 0 \text{ to } K-1: \\
&\quad w_{k+1} = \theta_k - \eta \mathfrak{r}(\theta_k) \\
&\quad \theta_{k+1} = \Pi_{\mathcal{B}}(w_{k+1}) \\
&\textbf{return } \theta_j \text{ with } j \text{ uniformly picked from } 1, \dots, K
\end{aligned} \tag{2.2}$$

The following Proposition is well known for convex objective functions (corresponding to 1-weakly-quasi-convex functions). We extend it (straightforwardly) to the case when τ-WQC holds with any positive constant τ.

Proposition 2.3 *Suppose the objective function f is τ-weakly-quasi-convex and Γ-weakly-smooth, and $\mathfrak{r}(\cdot)$ is an unbiased estimator for $\nabla f(\theta)$ with variance V. Moreover, suppose the global minimum θ^* belongs to $\mathcal{B}$, and the initial point θ_0 satisfies $\|\theta_0 - \theta^*\| \le R$. Then projected gradient descent (2.2) with a proper learning rate returns θ_K in K iterations with expected error*

$$\mathbb{E} f(\theta_K) - f(\theta^*) \le O\left(\max\left\{\frac{\Gamma R^2}{\tau^2 K}, \frac{R\sqrt{V}}{\tau\sqrt{K}}\right\}\right).$$

Remark 2.4 *It's straightforward to see (from the proof) that the algorithm tolerates inverse exponential bias, namely bias on the order of $\exp(-\Omega(n))$, in the gradient estimator. Technically, suppose $\mathbb{E}[\mathfrak{r}(\theta)] = \nabla f(\theta) \pm \zeta$ then $f(\theta_K) \le O\left(\max\left\{\frac{\Gamma R^2}{\tau^2 K}, \frac{R\sqrt{V}}{\tau\sqrt{K}}\right\}\right) + \mathrm{poly}(K)\cdot\zeta$. Throughout the paper, we assume that the error that we are shooting for is inverse polynomial, namely $1/n^C$ for some absolute constant C, and therefore the effect of inverse exponential bias is negligible.*

We defer the proof of Proposition 2.3 to Appendix A which is a simple variation of the standard convergence analysis of stochastic gradient descent (see, for example, Bottou (1998)). Finally, we note that the sum of two quasi-convex functions may no longer be quasi-convex. However, if a sequence functions is τ-WQC with respect to a common point θ^*, then their sum is also τ-WQC. This follows from the linearity of gradient operation.

Proposition 2.5 *Suppose functions $f_1, \dots, f_n$ are individually τ-weakly-quasi-convex in $\mathcal{B}$ with respect to a common global minimum θ^* , then for non-negative $w_1, \dots, w_n$ the linear combination $f = \sum_{i=1}^n w_i f_i$ is also τ-weakly-quasi-convex with respect to θ^* in $\mathcal{B}$.*

3. Population risk in frequency domain

We next establish conditions under which risk is weakly-quasi-convex. Our strategy is to first approximate the risk functional $f(\widehat{\Theta})$ by what we call *idealized risk*. This approximation

of the objective function is fairly straightforward; we justify it toward the end of the section. We can show that

$$f(\widehat{\Theta}) \approx \|D - \hat{D}\|^2 + \textstyle\sum_{k=0}^{\infty} \left(\hat{C}\hat{A}^k B - CA^k B\right)^2. \tag{3.1}$$

The leading term $\|D - \hat{D}\|^2$ is convex in $\hat{D}$ which appears nowhere else in the objective. It therefore doesn't affect the convergence of the algorithm (up to lower order terms) by virtue of Proposition 2.5, and we restrict our attention to the remaining terms.

Definition 3.1 (Idealized risk) *We define the* idealized risk *as*

$$g(\hat{A}, \hat{C}) = \sum_{k=0}^{\infty} \left(\hat{C}\hat{A}^k B - CA^k B\right)^2. \tag{3.2}$$

We now use basic concepts from control theory (see Heij et al. (2007); Hespanha (2009) for more background) to express the idealized risk (3.2) in Fourier domain. The *transfer function* of the linear system is

$$G(z) = C(zI - A)^{-1}B. \tag{3.3}$$

Note that $G(z)$ is a rational function over the complex numbers of degree n and hence we can find polynomials $s(z)$ and $p(z)$ such that $G(z) = \frac{s(z)}{p(z)}$, with the convention that the leading coefficient of $p(z)$ is 1. In controllable canonical form (1.4), the coefficients of p will correspond to the last row of the A, while the coefficients of s correspond to the entries of C. Also note that

$$G(z) = \sum_{t=1}^{\infty} z^{-t} C A^{t-1} B = \sum_{t=1}^{\infty} z^{-t} r_{t-1}$$

The sequence $r = (r_0, r_1, \dots, r_t, \dots) = (CB, CAB, \dots, CA^tB, \dots)$ is called the *impulse response* of the linear system. The behavior of a linear system is uniquely determined by the impulse response and therefore by $G(z)$. Analogously, we denote the transfer function of the learned system by $\widehat{G}(z) = \hat{C}(zI - \hat{A})^{-1}B = \hat{s}(z)/\hat{p}(z)$. The idealized risk (3.2) is only a function of the impulse response $\hat{r}$ of the learned system, and therefore it is only a function of $\widehat{G}(z)$.

Recall that $C = [c_1, \dots, c_n]$ is defined in Section 1.7. For future reference, we note that by some elementary calculation (see Lemma B.1), we have

$$G(z) = C(zI - A)^{-1}B = \frac{c_1 + \cdots + c_n z^{n-1}}{z^n + a_1 z^{n-1} + \cdots + a_n}, \tag{3.4}$$

which implies that $s(z) = c_1 + \cdots + c_n z^{n-1}$ and $p(z) = z^n + a_1 z^{n-1} + \cdots + a_n$.

With these definitions in mind, we are ready to express idealized risk in Fourier domain.

Proposition 3.2 *Suppose $p_{\hat{a}}(z)$ has all its roots inside unit circle, then the idealized risk $g(\hat{a}, \hat{C})$ can be written in the Fourier domain as*

$$g(\hat{A}, \hat{C}) = \int_0^{2\pi} \left|\widehat{G}(e^{i\theta}) - G(e^{i\theta})\right|^2 \mathrm{d}\theta.$$

Proof Note that $G(e^{i\theta})$ is the Fourier transform of the sequence $\{r_k\}$ and so is $\widehat{G}(e^{i\theta})$ the Fourier transform[2] of $\hat{r}_k$. Therefore by Parseval' Theorem, we have that

$$g(\hat{A},\hat{C}) = \sum_{k=0}^{\infty} \|\hat{r}_k - r_k\|^2 = \int_0^{2\pi} |\widehat{G}(e^{i\theta}) - G(e^{i\theta})|^2 d\theta\,. \tag{3.5}$$

■

3.1 Quasi-convexity of the idealized risk

Now that we have a convenient expression for risk in Fourier domain, we can prove that the idealized risk $g(\hat{A},\hat{C})$ is weakly-quasi-convex when $\hat{a}$ is not so far from the true a in the sense that $p_a(z)$ and $\hat{p}_a(z)$ have an angle less than $\pi/2$ for every z on the unit circle. We will use the convention that a and $\hat{a}$ refer to the parameters that specify A and $\hat{A}$, respectively.

Lemma 3.3 *For $\tau > 0$ and every $\hat{C}$, the idealized risk $g(\hat{A},\hat{C})$ is τ-weakly-quasi-convex over the domain*

$$\mathcal{N}_\tau(a) = \left\{ \hat{a} \in \mathbb{R}^n \colon \Re\left(\frac{p_a(z)}{p_{\hat{a}}(z)}\right) \geq \tau/2, \forall\ z \in \mathbb{C},\ \text{s.t. } |z| = 1 \right\}. \tag{3.6}$$

Proof We first analyze a single term $h = |\widehat{G}(z) - G(z)|^2$. Recall that $\widehat{G}(z) = \hat{s}(z)/\hat{p}(z)$ where $\hat{p}(z) = p_{\hat{a}}(z) = z^n + \hat{a}_1 z^{n-1} + \cdots + \hat{a}_n$. Note that z is fixed and h is a function of $\hat{a}$ and $\hat{C}$. Then it is straightforward to see that

$$\frac{\partial h}{\partial \hat{s}(z)} = 2\Re\left\{ \frac{1}{\hat{p}(z)} \left(\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right)^* \right\}. \tag{3.7}$$

and

$$\frac{\partial h}{\partial \hat{p}(z)} = -2\Re\left\{ \frac{\hat{s}(z)}{\hat{p}(z)^2} \left(\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right)^* \right\}. \tag{3.8}$$

Since $\hat{s}(z)$ and $\hat{p}(z)$ are linear in $\hat{C}$ and $\hat{a}$ respectively, by chain rule we have that

$$\begin{aligned}
\langle \frac{\partial h}{\partial \hat{a}}, \hat{a} - a\rangle + \langle \frac{\partial h}{\partial \hat{C}}, \hat{C} - C\rangle &= \frac{\partial h}{\partial \hat{p}(z)} \langle \frac{\partial \hat{p}(z)}{\partial \hat{a}}, \hat{a} - a\rangle + \frac{\partial h}{\partial \hat{s}(z)} \langle \frac{\partial \hat{s}(z)}{\partial \hat{C}}, \hat{C} - C\rangle \\
&= \frac{\partial h}{\partial \hat{p}(z)} (\hat{p}(z) - p(z)) + \frac{\partial h}{\partial \hat{s}(z)} (\hat{s}(z) - s(z))\,.
\end{aligned}$$

2. The Fourier transform exists since $\|r_k\|^2 = \|\hat{C}\hat{A}^k\hat{B}\|^2 \leq \|\hat{C}\|\|\hat{A}^k\|\|\hat{B}\| \leq c\rho(\hat{A})^k$ where c doesn't depend on k and $\rho(\hat{A}) < 1$.

Plugging the formulas (3.7) and (3.8) for $\frac{\partial h}{\partial \hat{s}(z)}$ and $\frac{\partial h}{\partial \hat{p}(z)}$ into the equation above, we obtain that

$$\begin{aligned}
\langle \frac{\partial h}{\partial \hat{a}}, \hat{a} - a\rangle + \langle \frac{\partial h}{\partial \hat{C}}, \hat{C} - C\rangle &= 2\Re\left\{ \frac{-\hat{s}(z)(\hat{p}(z) - p(z)) + \hat{p}(z)(\hat{s}(z) - s(z))}{\hat{p}(z)^2} \left(\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right)^* \right\} \\
&= 2\Re\left\{ \frac{\hat{s}(z)p(z) - s(z)\hat{p}(z)}{\hat{p}(z)^2} \left(\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right)^* \right\} \\
&= 2\Re\left\{ \frac{p(z)}{\hat{p}(z)} \left|\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right|^2 \right\} \\
&= 2\Re\left\{ \frac{p(z)}{\hat{p}(z)} \right\} \left|\widehat{G}(z) - G(z)\right|^2.
\end{aligned}$$

Hence $h = |\widehat{G}(z) - G(z)|^2$ is τ-weakly-quasi-convex with $\tau = 2\min_{|z|=1} \Re\left\{\frac{p(z)}{\hat{p}(z)}\right\}$. This implies our claim, since by Proposition 3.2, the idealized risk g is convex combination of functions of the form $|\widehat{G}(z) - G(z)|^2$ for $|z| = 1$. Moreover, Proposition 2.5 shows that convex combination preserves weak quasi-convexity. ■

For future reference, we also prove that the idealized risk is $O(n^2/\tau_1^4)$-weakly smooth.

Lemma 3.4 *The idealized risk $g(\hat{A},\hat{C})$ is Γ-weakly smooth with $\Gamma = O(n^2/\tau_1^4)$.*

Proof By equation (3.8) and the chain rule we get that

$$\frac{\partial g}{\partial \hat{C}} = \int_{\mathbb{T}} \frac{\partial |\widehat{G}(z) - G(z)|^2}{\partial p(z)} \cdot \frac{\partial p(z)}{\partial \hat{C}} dz = \int_{\mathbb{T}} 2\Re\left\{ \frac{1}{\hat{p}(z)} \left(\frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)}\right)^* \right\} \cdot [1, \ldots, z^{n-1}] dz\,.$$

therefore we can bound the norm of the gradient by

$$\left\| \frac{\partial g}{\partial \hat{C}} \right\|^2 \leq \left(\int_{\mathbb{T}} \left| \frac{\hat{s}(z)}{\hat{p}(z)} - \frac{s(z)}{p(z)} \right|^2 dz \right) \cdot \left(\int_{\mathbb{T}} 4\|[1, \ldots, z^{n-1}]\|^2 \cdot |\frac{1}{p(z)}|^2 dz \right) \leq O(n/\tau_1^2) \cdot g(\hat{A},\hat{C})\,.$$

Similarly, we could show that $\left\|\frac{\partial g}{\partial \hat{a}}\right\|^2 \leq O(n^2/\tau_1^2) \cdot g(\hat{A},\hat{C})$. ■

3.2 Justifying idealized risk

We need to justify the approximation we made in Equation (3.1).

Lemma 3.5 *Assume that ξ_t and x_t are drawn i.i.d. from an arbitrary distribution with mean 0 and variance 1. Then the population risk $f(\widehat{\Theta})$ can be written as,*

$$f(\widehat{\Theta}) = (\hat{D} - D)^2 + \sum_{k=1}^{T-1} \left(1 - \frac{k}{T}\right) \left(\hat{C}\hat{A}^{k-1}B - CA^{k-1}B\right)^2 + \sigma^2\,. \tag{3.9}$$

The idealized risk is upper bound of the population risk $f(\widehat{\Theta})$ according to equation (3.1) and (3.9). We don't have to quantify the gap between them because later in Algorithm 1, we will directly optimize the idealized risk by constructing an estimator of its gradient, and thus the optimization will guarantee a bounded idealized risk which translates to a bounded population risk. See Section 5 for details.

Proof [Proof of Lemma 3.5] Under the distributional assumptions on ξ_t and x_t, we can calculate the objective functions above analytically. We write out $y_t, \hat{y}_t$ in terms of the inputs,

$$y_t = Dx_t + \sum_{k=1}^{t-1} CA^{t-k-1}Bx_k + CA^{t-1}h_0 + \xi_t\,, \qquad \hat{y}_t = \hat{D}x_t + \sum_{k=1}^{t-1} \hat{C}\hat{A}^{t-k-1}\hat{B}x_k + CA^{t-1}h_0\,.$$

Therefore, using the fact that x_t's are independent and with mean 0 and covariance I, the expectation of the error can be calculated (formally by Claim B.2),

$$\mathbb{E}\left[\|\hat{y}_t - y_t\|^2\right] = \|\hat{D} - D\|_F^2 + \textstyle\sum_{k=1}^{t-1} \left\|\hat{C}\hat{A}^{t-k-1}\hat{B} - CA^{t-k-1}B\right\|_F^2 + \mathbb{E}[\|\xi_t\|^2]\,. \tag{3.10}$$

Using $\mathbb{E}[\|\xi_t\|^2] = \sigma^2$, it follows that

$$f(\widehat{\Theta}) = \|\hat{D} - D\|_F^2 + \textstyle\sum_{k=1}^{T-1} \left(1 - \frac{k}{T}\right)\left\|\hat{C}\hat{A}^{k-1}\hat{B} - CA^{k-1}B\right\|_F^2 + \sigma^2\,. \tag{3.11}$$

Recall that under the controllable canonical form (1.4), $B = e_n$ is known and therefore $\hat{B} = B$ is no longer a variable. Then the expected objective function (3.11) simplifies to

$$f(\widehat{\Theta}) = (\hat{D} - D)^2 + \textstyle\sum_{k=1}^{T-1} \left(1 - \frac{k}{T}\right)\left(\hat{C}\hat{A}^{k-1}B - CA^{k-1}B\right)^2 + \sigma^2\,.$$

■

The previous lemma does not yet control higher order contributions present in the idealized risk. This requires additional structure that we introduce in the next section.

4. Effective relaxations of spectral radius

The previous section showed quasi-convexity of the idealized risk. However, several steps are missing towards showing finite sample guarantees for stochastic gradient descent. In particular, we will need to control the variance of the stochastic gradient at any system that we encounter in the training. For this purpose we formally introduce our main assumption now and show that it serves as an effective relaxation of spectral radius. This results below will be used for proving convergence of stochastic gradient descent in Section 5.

Consider the following convex region $\mathcal{C}$ in the complex plane,

$$\mathcal{C} = \{z\colon \Re z \geq (1+\tau_0)|\Im z|\} \cap \{z\colon \tau_1 < \Re z < \tau_2\}\,. \tag{4.1}$$

where $\tau_0, \tau_1, \tau_2 > 0$ are constants that are considered as fixed constant throughout the paper. Our bounds will have polynomial dependency on these parameters. Pictorially, this convex set is pretty much the dark area in Figure 1 (with the corner chopped). This set in $\mathbb{C}$ induces a convex set in the parameter space which is a subset of the transition matrix with spectral radius less than α.

Definition 4.1 *We say a polynomial $p(z)$ is* α-acquiescent *if $\{p(z)/z^n\colon |z| = \alpha\} \subseteq \mathcal{C}$. A linear system with transfer function $G(z) = s(z)/p(z)$ is α-acquiescent if the denominator $p(z)$ is.*

The set of coefficients $a \in \mathbb{R}^n$ defining acquiescent systems form a convex set. Formally, for a positive $\alpha > 0$, define the convex set $\mathcal{B}_\alpha \subseteq \mathbb{R}^n$ as

$$\mathcal{B}_\alpha = \left\{a \in \mathbb{R}^n : \{p_a(z)/z^n\colon |z| = \alpha\} \subseteq \mathcal{C}\right\}. \tag{4.2}$$

We note that definition (4.2) is equivalent to the definition $\mathcal{B}_\alpha = \left\{a \in \mathbb{R}^n : \{z^n p(1/z)\colon |z| = 1/\alpha\} \subseteq \mathcal{C}\right\}$, which is the version that we used in introduction for simplicity. Indeed, we can verify the convexity of $\mathcal{B}_\alpha$ by definition and the convexity of $\mathcal{C}$: $a, b \in B_\alpha$ implies that $p_a(z)/z^n, p_b(z)/z^n \in \mathcal{C}$ and therefore, $p_{(a+b)/2}(z)/z^n = \frac{1}{2}(p_a(z)/z^n + p_b(z)/z^n) \in \mathcal{C}$. We also note that the parameter α in the definition of acquiescence corresponds to the spectral radius of the companion matrix. In particular, an acquiescent system is stable for $\alpha < 1$.

Lemma 4.2 *Suppose $a \in \mathcal{B}_\alpha$, then the roots of polynomial $p_a(z)$ have magnitudes bounded by α. Therefore the controllable canonical form $A = \mathrm{CC}(a)$ defined by a has spectral radius $\rho(A) < \alpha$.*

Proof Define holomorphic function $f(z) = z^n$ and $g(z) = p_a(z) = z^n + a_1 z^{n-1} + \cdots + a_n$. We apply the symmetric form of Rouche's theorem Estermann (1962) on the circle $\mathcal{K} = \{z : |z| = \alpha\}$. For any point z on $\mathcal{K}$, we have that $|f(z)| = \alpha^n$, and that $|f(z) - g(z)| = \alpha^n \cdot |1 - p_a(z)/z^n|$. Since $a \in \mathcal{B}_\alpha$, we have that $p_a(z)/z^n \in \mathcal{C}$ for any z with $|z| = \alpha$. Observe that for any $c \in \mathcal{C}$ we have that $|1 - c| < 1 + |c|$, therefore we have that

$$|f(z) - g(z)| = \alpha^n |1 - p_a(z)/z^n| < \alpha^n(1 + |p_a(z)|/|z^n|) = |f(z)| + |p_a(z)| = |f(z)| + |g(z)|\,.$$

Hence, using Rouche's Theorem, we conclude that f and g have same number of roots inside circle $\mathcal{K}$. Note that function $f = z^n$ has exactly n roots in $\mathcal{K}$ and therefore g have all its n roots inside circle $\mathcal{K}$. ■

The following lemma establishes the fact that $\mathcal{B}_\alpha$ is a monotone family of sets in α. The proof follows from the maximum modulo principle of the harmonic functions $\Re(z^n p(1/z))$ and $\Im(z^n p(1/z))$. We defer the short proof to Section C.1. We remark that there are larger convex sets than $\mathcal{B}_\alpha$ that ensure bounded spectral radius. However, in order to also guarantee monotonicity and the no blow-up property below, we have to restrict our attention to $\mathcal{B}_\alpha$.

Lemma 4.3 (Monotonicity of $\mathcal{B}_\alpha$) *For any $0 < \alpha < \beta$, we have that $\mathcal{B}_\alpha \subset \mathcal{B}_\beta$.*

Our next lemma entails that acquiescent systems have well behaved impulse responses.

Lemma 4.4 (No blow-up property) *Suppose $a \in \mathcal{B}_\alpha$ for some $\alpha \leq 1$. Then the companion matrix $A = \mathrm{CC}(a)$ satisfies*

$$\sum_{k=0}^{\infty} \|\alpha^{-k} A^k B\|^2 \leq 2\pi n \alpha^{-2n}/\tau_1^2. \tag{4.3}$$

Moreover, it holds that for any $k \geq 0$,

$$\|A^k B\|^2 \leq \min\{2\pi n/\tau_1^2, 2\pi n \alpha^{2k-2n}/\tau_1^2\}\,.$$

Proof [Proof of Lemma 4.4]

Let $f_\lambda = \sum_{k=0}^{\infty} e^{i\lambda k}\alpha^{-k}A^k B$ be the Fourier transform of the series $\alpha^{-k}A^k B$. Then using Parseval's Theorem, we have

$$\begin{aligned}\sum_{k=0}^{\infty}\|\alpha^{-k}A^k B\|^2 &= \int_0^{2\pi}|f_\lambda|^2 d\lambda = \int_0^{2\pi}|(I-\alpha^{-1}e^{i\lambda}A)^{-1}B|^2 d\lambda \\ &= \int_0^{2\pi}\frac{\sum_{j=1}^{n}\alpha^{2j}}{|p_a(\alpha e^{-i\lambda})|^2}d\lambda \leq \int_0^{2\pi}\frac{n}{|p_a(\alpha e^{-i\lambda})|^2}d\lambda. \qquad (4.4)\end{aligned}$$

where at the last step we used the fact that $(I-wA)^{-1}B = \frac{1}{p_a(w^{-1})}[w^{-1}, w^{-2}\ldots, z^{-n}]^\top$ (see Lemma B.1), and that $\alpha \leq 1$. Since $a \in \mathcal{B}_\alpha$, we have that $|q_a(\alpha^{-1}e^{i\lambda})| \geq \tau_1$, and therefore $p_a(\alpha e^{-i\lambda}) = \alpha^n e^{-in\lambda}q(e^{i\lambda}/\alpha)$ has magnitude at least $\tau_1\alpha^n$. Plugging in this into equation (4.4), we conclude that

$$\sum_{k=0}^{\infty}\|\alpha^{-k}A^k B\|^2 \leq \int_0^{2\pi}\frac{n}{|p_a(\alpha e^{-i\lambda})|^2}d\lambda \leq 2\pi n\alpha^{-2n}/\tau_1^2.$$

Finally we establish the bound for $\|A^k B\|^2$. By Lemma 4.3, we have $\mathcal{B}_\alpha \subset \mathcal{B}_1$ for $\alpha \leq 1$. Therefore we can pick $\alpha = 1$ in equation (4.3) and it still holds. That is, we have that

$$\sum_{k=0}^{\infty}\|A^k B\|^2 \leq 2\pi n/\tau_1^2.$$

This also implies that $\|A^k B\|^2 \leq 2\pi n/\tau_1^2$. ■

4.1 Efficiently computing the projection

In our algorithm, we require a projection onto $\mathcal{B}_\alpha$. However, the only requirement of the projection step is that it projects onto a set contained inside $\mathcal{B}_\alpha$ that also contains the true linear system. So a variety of subroutines can be used to compute this projection or an approximation. First, the explicit projection onto $\mathcal{B}_\alpha$ is representable by a semidefinite program. This is because each of the three constrains can be checked by testing if a trigonometric polynomial is non-negative. A simple inner approximation can be constructed by requiring the constraints to hold on an a finite grid of size $O(n)$. One can check that this provides a tight, polyhedral approximation to the set $\mathcal{B}_\alpha$, following an argument similar to Appendix C of Bhaskar *et al* Bhaskar et al. (2013). Projection to this polyhedral takes at most $O(n^{3.5})$ time by linear programming and potentially can be made faster by using fast Fourier transform. See Section F for more detailed discussion on why projection on a polytope suffices. Furthermore, sometimes we can replace the constraint by an ℓ_1 or ℓ_2-constraint if we know that the system satisfies the corresponding assumption. Removing the projection step entirely is an interesting open problem.

5. Learning acquiescent systems

Next we show that we can learn acquiescent systems.

Theorem 5.1 *Suppose the true system Θ is α-acquiescent and satisfies $\|C\| \leq 1$. Then with N samples of length $T \geq \Omega(n + 1/(1-\alpha))$, stochastic gradient descent (Algorithm 1) with projection set $\mathcal{B}_\alpha$ returns parameters $\widehat{\Theta} = (\hat{A}, \hat{B}, \hat{C}, \hat{D})$ with population risk*

$$\mathbb{E}\, f(\widehat{\Theta}) \leq f(\Theta) + O\left(\frac{n^2}{N} + \sqrt{\frac{n^5 + \sigma^2 n^3}{TN}}\right), \qquad (5.1)$$

where $O(\cdot)$-notation hides polynomial dependencies on $1/(1-\alpha)$, $1/\tau_0, 1/\tau_1, \tau_2$, and $R = \|a\|$. The expectation is taken over the randomness of the algorithms and the examples.

Algorithm 1 Projected stochastic gradient descent with partial loss

For $i = 0$ to N:

1. Take a fresh sample $((x_1, \ldots, x_T), (y_1, \ldots, y_T))$. Let $\tilde{y}_t$ be the simulated outputs[3] of system $\widehat{\Theta}$ on inputs x and initial states $h_0 = 0$.
2. Let $T_1 = T/4$. Run stochastic gradient descent[4] on loss function $\ell((x,y),\widehat{\Theta}) = \frac{1}{T-T_1}\sum_{t>T_1}\|\tilde{y}_t - y_t\|^2$. Concretely, let $G_A = \frac{\partial \ell}{\partial \hat{a}}$, $G_C = \frac{\partial \ell}{\partial \hat{C}}$, and , $G_D = \frac{\partial \ell}{\partial \hat{D}}$, we update
$$[\hat{a}, \hat{C}, \hat{D}] \to [\hat{a}, \hat{C}, \hat{D}] - \eta[G_A, G_C, G_D]\,.$$
3. Project $\widehat{\Theta} = (\hat{a}, \hat{C}, \hat{D})$ to the set $\mathcal{B}_\alpha \otimes \mathbb{R}^n \otimes \mathbb{R}$.

Recall that T is the length of the sequence and N is the number of samples. The first term in the bound (5.1) comes from the smoothness of the population risk and the second comes from the variance of the gradient estimator of population risk (which will be described in detail below). An important (but not surprising) feature here is the variance scale in $1/T$ and therefore for long sequence actually we got $1/N$ convergence instead of $1/\sqrt{N}$ (for relatively small N).

Computational complexity: Step 2 in each iteration of the algorithm takes $O(Tn)$ arithmetic operations, and the projection step takes $O(n^{3.5})$ time to solve an linear programming problem. The project step is unlikely to be required in practice and may be an artifact of our analysis.

We can further balance the variance of the estimator with the number of samples by breaking each long sequence of length T into $\Theta(T/n)$ short sequences of length $\Theta(n)$, and then run back-propagation (1) on these TN/n shorter sequences. This leads us to the following bound which gives the right dependency in T and N as we expected: TN should be counted as the true number of samples for the sequence-to-sequence model.

4. Note that $\tilde{y}_t$ is different from $\hat{y}_t$ defined in equation (1.2) which is used to define the population risk: here $\hat{y}_t$ is obtained from the (wrong) initial state $h_0 = 0$ while $\hat{y}_t$ is obtained from the correct initial state.
4. See Algorithm Box 3 for a detailed back-propagation algorithm that computes the gradient.

Corollary 5.2 *Under the assumption of Theorem 5.1, Algorithm 2 returns parameters* $\widehat{\Theta}$ *with population risk*

$$\mathbb{E}\, f(\widehat{\Theta}) \le f(\Theta) + O\left(\sqrt{\frac{n^5+\sigma^2 n^3}{TN}}\right),$$

where $O(\cdot)$*-notation hides polynomial dependencies on* $1/(1-\alpha)$, $1/\tau_0, 1/\tau_1, \tau_2$, *and* $R = \|a\|$.

Algorithm 2 Projected stochastic gradient descent for long sequences
Input: N samples sequences of length T
Output: Learned system $\widehat{\Theta}$

1. Divide each sample of length T into $T/(\beta n)$ samples of length βn where β is a large enough constant. Then run algorithm 1 with the new samples and obtain $\widehat{\Theta}$.

We remark the the gradient computation procedure takes time linear in Tn since one can use chain-rule (also called back-propagation) to compute the gradient efficiently . For completeness, Algorithm 3 gives a detailed implementation. Finally and importantly, we remark that although we defined the population risk as the expected error with respected to sequence of length T, actually our error bound generalizes to any longer (or shorter) sequences of length $T' \gg \max\{n, 1/(1-\alpha)\}$. By the explicit formula for $f(\widehat{\Theta})$ (Lemma 3.5) and the fact that $\|CA^kB\|$ decays exponentially for $k \gg n$ (Lemma 4.4), we can bound the population risk on sequences of different lengths. Concretely, let $f_{T'}(\widehat{\Theta})$ denote the population risk on sequence of length T', we have for all $T' \gg \max\{n, 1/(1-\alpha)\}$,

$$f_{T'}(\widehat{\Theta}) \le 1.1 f(\widehat{\Theta}) + \exp(-(1-\alpha)\min\{T, T'\}) \le O\left(\sqrt{\frac{n^5+\sigma^2 n^3}{TN}}\right).$$

We note that generalization to longer sequence does deserve attention. Indeed in practice, it's usually difficult to train non-linear recurrent networks that generalize to longer sequences than the training data.

We could hope to achieve linear convergence by showing that the empirical risk also satisfies the weakly-quasi-convexity. Then, we can re-use the samples and hope to use strong optimization tools (such as SVRG) to achieve the linear convergence. This is beyond the scope of this paper and left to future work.

Our proof of Theorem 5.1 simply consists of three parts: a) showing the idealized risk is weakly quasi-convex in the convex set $\mathcal{B}_\alpha$ (Lemma 5.3); b) designing an (almost) unbiased estimator of the gradient of the idealized risk (Lemma 5.4); c) variance bounds of the gradient estimator (Lemma 5.5).

First of all, using the theory developed in Section 3 (Lemma 3.3 and Lemma 3.4), it is straightforward to verify that in the convex set $\mathcal{B}_\alpha \otimes \mathbb{R}^n$, the idealized risk is both weakly-quasi-convex and weakly-smooth.

Lemma 5.3 *Under the condition of Theorem 5.1, the idealized risk* (3.2) *is* τ*-weakly-quasi-convex in the convex set* $\mathcal{B}_\alpha \otimes \mathbb{R}^n$ *and* Γ*-weakly smooth, where* $\tau = \Omega(\tau_0\tau_1/\tau_2)$ *and* $\Gamma = O(n^2/\tau_1^4)$.

Proof [Proof of Lemma 5.3] It suffices to show that for all $\hat{a}, a \in \mathcal{B}_\alpha$, it satisfies $\hat{a} \in \mathcal{N}_\tau(a)$ for $\tau = \Omega(\tau_0\tau_1/\tau_2)$. Indeed, by the monotonicity of the family of sets $\mathcal{B}_\alpha$ (Lemma 4.3), we have that $\hat{a}, a \in \mathcal{B}_1$, which by definition means for every z on unit circle, $p_a(z)/z^n, p_{\hat{a}}(z)/z^n \in \mathcal{C}$. By definition of $\mathcal{C}$, for any point $w, \hat{w} \in \mathcal{C}$, the angle ϕ between w and $\hat{w}$ is at most $\pi - \Omega(\tau_0)$ and ratio of the magnitude is at least τ_1/τ_2, which implies that $\Re(w/\hat{w}) = |w|/|\hat{w}| \cdot \cos(\phi) \ge \Omega(\tau_0\tau_1/\tau_2)$. Therefore $\Re(p_a(z)/p_{\hat{a}}(z)) \ge \Omega(\tau_0\tau_1/\tau_2)$, and we conclude that $\hat{a} \in \mathcal{N}_\tau(a)$. The smoothness bound was established in Lemma 3.4. ■

Towards designing an unbiased estimator of the gradient, we note that there is a small caveat here that prevents us to just use the gradient of the empirical risk, as commonly done for other (static) problems. Recall that the population risk is defined as the expected risk with *known* initial state h_0, while in the training we don't have access to the initial states and therefore using the naive approach we couldn't even estimate population risk from samples without knowing the initial states.

We argue that being able to handle the missing initial states is indeed desired: in most of the interesting applications h_0 is unknown (or even to be learned). Moreover, the ability of handling unknown h_0 allows us to break a long sequence into shorter sequences, which helps us to obtain Corollary 5.2. Here the difficulty is essentially that we have a supervised learning problem with missing data h_0. We get around it by simply ignoring first $T_1 = \Omega(T)$ outputs of the system and setting the corresponding errors to 0. Since the influence of h_0 to any outputs later than time $k \ge T_1 \gg \max\{n, 1/(1-\alpha)\}$ is inverse exponentially small, we could safely assume $h_0 = 0$ when the error earlier than time T_1 is not taken into account.

This small trick also makes our algorithm suitable to the cases when these early outputs are actually not observed. This is indeed an interesting setting, since in many sequence-to-sequence model Sutskever et al. (2014), there is no output in the first half fraction of iterations (of course these models have non-linear operation that we cannot handle).

The proof of the correctness of the estimator is almost trivial and deferred to Section C.

Lemma 5.4 *Under the assumption of Theorem 5.1, suppose* $\hat{a}, a \in \mathcal{B}_\alpha$. *Then in Algorithm 1, at each iteration,* G_A, G_C *are unbiased estimators of the gradient of the idealized risk* (3.2) *in the sense that:*

$$\mathbb{E}\,[G_A, G_C] = \left[\frac{\partial g}{\partial \hat{a}}, \frac{\partial g}{\partial \hat{C}}\right] \pm \exp(-\Omega((1-\alpha)T)). \tag{5.2}$$

Finally, we control the variance of the gradient estimator.

Lemma 5.5 *The (almost) unbiased estimator* (G_A, G_C) *of the gradient of* $g(\hat{A}, \hat{C})$ *has variance bounded by*

$$\mathrm{Var}\,[G_A] + \mathrm{Var}\,[G_C] \le \frac{O\left(n^3\Lambda^2/\tau_1^6 + \sigma^2 n^2 \Lambda/\tau_1^4\right)}{T}.$$

where $\Lambda = O(\max\{n, 1/(1-\alpha)\log 1/(1-\alpha)\})$.

Note that Lemma 5.5 does not directly follow from the Γ-weakly-smoothness of the population risk, since it's not clear whether the loss function $\ell((x,y),\widehat{\Theta})$ is also Γ-smooth for every sample. Moreover, even if it could work out, from smoothness the variance bound can be only as small as Γ^2, while the true variance scales linearly in $1/T$. Here the discrepancy comes from that smoothness implies an upper bound of the expected squared norm of the gradient, which is equal to the variance plus the expected squared mean. Though typically for many other problems variance is on the same order as the squared mean, here for our sequence-to-sequence model, actually the variance decreases in length of the data, and therefore the bound of variance from smoothness is pessimistic.

We bound directly the variance instead. It's tedious but simple in spirit. We mainly need Lemma 4.4 to control various difference sums that shows up from calculating the expectation. The only tricky part is to obtain the $1/T$ dependency which corresponds to the cancellation of the contribution from the cross terms. In the proof we will basically write out the variance as a (complicated) function of $\hat{A},\hat{C}$ which consists of sums of terms involving $(\hat{C}\hat{A}^kB - CA^kB)$ and $\hat{A}^kB$. We control these sums using Lemma 4.4. The proof is deferred to Section C.

Finally we are ready to prove Theorem 5.1. We essentially just combine Lemma 5.3, Lemma 5.4 and Lemma 5.5 with the generic convergence Proposition 2.3. This will give us low error in idealized risk and then we relate the idealized risk to the population risk.

Proof [Proof of Theorem 5.1] We consider $g'(\hat{A},\hat{C},\hat{D}) = (\hat{D}-D)^2 + g(\hat{A},\hat{C})$, an extended version of the idealized risk which takes the contribution of $\hat{D}$ into account. By Lemma 5.4 we have that Algorithm 1 computes G_A, G_C which are almost unbiased estimators of the gradients of g' up to negligible error $\exp(-\Omega((1-\alpha)T))$, and by Lemma C.2 we have G_D is an unbiased estimator of g' with respect to $\hat{D}$. Moreover by Lemma 5.5, these unbiased estimator has total variance $V = \frac{O(n^5+\sigma^2n^3)}{T}$ where $O(\cdot)$ hides dependency on τ_1 and $(1-\alpha)$. Applying Proposition 2.3 (which only requires an unbiased estimator of the gradient of g'), we obtain that after T iterations, we converge to a point with $g'(\hat{a},\hat{C},\hat{D}) \le O\left(\frac{n^2}{N} + \sqrt{\frac{n^5+\sigma^2n^3}{TN}}\right)$. Then, by Lemma 3.5 we have $f(\widehat{\Theta}) \le g'(\hat{a},\hat{C},\hat{D}) + \sigma^2 = g'(\hat{a},\hat{C},\hat{D}) + f(\Theta) \le O\left(\frac{n^2}{N} + \sqrt{\frac{n^5+\sigma^2n^3}{TN}}\right) + f(\Theta)$ which completes the proof. ■

6. The power of improper learning

We observe an interesting and important fact about the theory in Section 5: it solely requires a condition on the characteristic function $p(z)$. This suggests that the geometry of the training objective function depends mostly on the denominator of the transfer function, even though the system is uniquely determined by the transfer function $G(z) = s(z)/p(z)$. This might seem to be an undesirable discrepancy between the behavior of the system and our analysis of the optimization problem.

However, we can actually exploit this discrepancy to design improper learning algorithms that succeed under much weaker assumptions. We rely on the following simple observation about the invariance of a system $G(z) = \frac{s(z)}{p(z)}$. For an arbitrary polynomial $u(z)$ of leading

coefficient 1, we can write $G(z)$ as

$$G(z) = \frac{s(z)u(z)}{p(z)u(z)} = \frac{\tilde{s}(z)}{\tilde{p}(z)}\,,$$

where $\tilde{s} = su$ and $\tilde{p} = pu$. Therefore the system $\tilde{s}(z)/\tilde{p}(z)$ has identical behavior as G. Although this is a redundant representation of $G(z)$, it should counted as an acceptable solution. After all, learning the minimum representation[5] of linear system is impossible in general. In fact, we will encounter an example in Section 6.1.

While not changing the behavior of the system, the extension from $p(z)$ to $\tilde{p}(z)$, does affect the geometry of the optimization problem. In particular, if $\tilde{p}(z)$ is now an α-acquiescent characteristic polynomial as defined in Definition 4.1, then we could find it simply using stochastic gradient descent as shown in Section 5. Observe that we don't require knowledge of $u(z)$ but only its existence. Denoting by d the degree of u, the algorithm itself is simply stochastic gradient descent with $n+d$ model parameters instead of n.

Our discussion motivates the following definition.

Definition 6.1 *A polynomial $p(z)$ of degree n is* α-acquiescent by extension of degree d *if there exists a polynomial $u(z)$ of degree d and leading coefficient 1 such that $p(z)u(z)$ is α-acquiescent.*

For a transfer function $G(z)$, we define it's $\mathcal{H}_2$ norm as

$$\|G\|_{\mathcal{H}_2}^2 = \frac{1}{2\pi}\int_0^{2\pi} |G(e^{i\theta})|^2 d\theta\,.$$

We assume (with loss of generality) that the true transfer function $G(z)$ has bounded $\mathcal{H}_2$ norm, that is, $\|G\|_{\mathcal{H}_2} \le 1$. This can be achieve by a rescaling[6] of the matrix C.

Theorem 6.2 *Suppose the true system has transfer function $G(z) = s(z)/p(z)$ with a characteristic function $p(z)$ that is α-acquiescent by extension of degree d, and $\|G\|_{\mathcal{H}_2} \le 1$, then projected stochastic gradient descent with $m = n+d$ states (that is, Algorithm 2 with m states) returns a system $\widehat{\Theta}$ with population risk*

$$f(\widehat{\Theta}) \le O\left(\sqrt{\frac{m^5+\sigma^2m^3}{TK}}\right)\,.$$

where the $O(\cdot)$ notation hides polynomial dependencies on $\tau_0, \tau_1, \tau_2, 1/(1-\alpha)$.

The theorem follows directly from Corollary 5.2 (with some additional care about the scaling.

Proof [Proof of Theorem 6.2] Let $\tilde{p}(z) = p(z)u(z)$ be the acquiescent extension of $p(z)$. Since $\tau_2 \ge |u(z)p(z)| = |\tilde{p}(z)| \ge \tau_0$ on the unit circle, we have that $|\tilde{s}(z)| = |s(z)||u(z)| =$

5. The minimum representation of a transfer function $G(z)$ is defined as the representation $G(z) = s(z)/p(z)$ with $p(z)$ having minimum degree.
6. In fact, this is a natural scaling that makes comparing error easier. Recall that the population risk is essentially $\|\hat{G} - G\|_{\mathcal{H}_2}$, therefore rescaling C so that $\|G\|_{\mathcal{H}_2} = 1$ implies that when error $\ll 1$ we achieve non-trivial performance.

$s(z) \cdot O_\tau(1/p(z))$. Therefore we have that $\tilde{s}(z)$ satisfies that $\|\tilde{s}\|_{\mathcal{H}_2} = O_\tau(\|s(z)/p(z)\|_{\mathcal{H}_2}) = O_\tau(\|G(z)\|_{\mathcal{H}_2}) \le O_\tau(1)$. That means that the vector C that determines the coefficients of $\tilde{s}$ satisfies that $\|C\| \le O_\tau(1)$, since for a polynomial $h(z) = b_0 + \cdots + b_{n-1}z^{n-1}$, we have $\|h\|_{\mathcal{H}_2} = \|b\|$. Therefore we can apply Corollary 5.2 to complete the proof. ■

In the rest of this section, we discuss in subsection 6.1 the instability of the minimum representation in subsection, and in subsection 6.2 we show several examples where the characteristic function $p(z)$ is not α-acquiescent but is α-acquiescent by extension with small degree d.

As a final remark, the examples illustrated in the following sub-sections may be far from optimally analyzed. It is beyond the scope of this paper to understand the optimal condition under which $p(z)$ is acquiescent by extension.

6.1 Instability of the minimum representation

We begin by constructing a contrived example where the minimum representation of $G(z)$ is not stable at all and as a consequence one can't hope to recover the minimum representation of $G(z)$.

Consider $G(z) = \frac{s(z)}{p(z)} := \frac{z^n - 0.8^{-n}}{(z-0.1)(z^n - 0.9^{-n})}$ and $G'(z) = \frac{s'(z)}{p'(z)} := \frac{1}{z-0.1}$. Clearly these are the minimum representations of the $G(z)$ and $G'(z)$, which also both satisfy acquiescence. On the one hand, the characteristic polynomial $p(z)$ and $p'(z)$ are very different. On the other hand, the transfer functions $G(z)$ and $G'(z)$ have almost the same values on unit circle up to exponentially small error,

$$|G(z) - G'(z)| \le \frac{0.8^{-n} - 0.9^{-n}}{(z-0.1)(z-0.9^{-n})} \le \exp(-\Omega(n)) .$$

Moreover, the transfer functions $G(z)$ and $\hat{G}(z)$ are on the order of $\Theta(1)$ on unit circle. These suggest that from an (inverse polynomially accurate) approximation of the transfer function $G(z)$, we cannot hope to recover the minimum representation in any sense, even if the minimum representation satisfies acquiescence.

6.2 Power of improper learning in various cases

We illustrate the use of improper learning through various examples below.

6.2.1 Example: artificial construction

We consider a simple contrived example where improper learning can help us learn the transfer function dramatically. We will show an example of characteristic function which is not 1-acquiescent but $(\alpha+1)/2$-$(\alpha+1)/2$-acquiescent by extension of degree 3.

Let n be a large enough integer and α be a constant. Let $J = \{1, n-1, n\}$ and $\omega = e^{2\pi i/n}$, and then define $p(z) = z^3 \prod_{j\in[n], j\notin J}(z - \alpha\omega^j)$. Therefore we have that

$$p(z)/z^n = z^3 \prod_{j\in[n], j\in J}(1 - \alpha\omega^j/z) = \frac{1 - \alpha^n/z^n}{(1-\omega/z)(1-\omega^{-1}/z)(1-1/z)} \tag{6.1}$$

Taking $z = e^{-i\pi/2}$ we have that $p(z)/z^n$ has argument (phase) roughly $-3\pi/4$, and therefore it's not in $\mathcal{C}$, which implies that $p(z)$ is *not* 1-acquiescent. On the other hand, picking $u(z) = (z-\omega)(z-1)(z-\omega^{-1})$ as the helper function, from equation (6.1) we have $p(z)u(z)/z^{n+3} = 1 - \alpha^n/z^n$ takes values inverse exponentially close to 1 on the circle with radius $(\alpha+1)/2$. Therefore $p(z)u(z)$ is $(\alpha+1)/2$-acquiescent.

6.2.2 Example: characteristic function with separated roots

A characteristic polynomial with well separated roots will be acquiescent by extension. Our bound will depend on the following quantity of p that characterizes the separateness of the roots.

Definition 6.3 *For a polynomial $h(z)$ of degree n with roots $\lambda_1, \ldots, \lambda_n$ inside unit circle, define the quantity $\Gamma(\cdot)$ of the polynomial h as:*

$$\Gamma(h) := \sum_{j\in[n]} \left| \frac{\lambda_j^n}{\prod_{i\neq j}(\lambda_i - \lambda_j)} \right| .$$

Lemma 6.4 *Suppose $p(z)$ is a polynomial of degree n with distinct roots inside circle with radius α. Let $\Gamma = \Gamma(p)$, then $p(z)$ is α-acquiescent by extension of degree $d \quad = O(\max\{(1-\alpha)^{-1}\log(\sqrt{n}\Gamma \cdot \|p\|_{\mathcal{H}_2}), 0\})$.*

Our main idea to extend $p(z)$ by multiplying some polynomial u that approximates p^{-1} (in a relatively weak sense) and therefore pu will always take values in the set $\mathcal{C}$. We believe the following lemma should be known though for completeness we provide the proof in Section D.

Lemma 6.5 (Approximation of inverse of a polynomial) *Suppose $p(z)$ is a polynomial of degree n and leading coefficient 1 with distinct roots inside circle with radius α, and $\Gamma = \Gamma(p)$. Then for $d = O(\max\{(\frac{1}{1-\alpha}\log\frac{\Gamma}{(1-\alpha)\zeta}, 0\})$, there exists a polynomial $h(z)$ of degree d and leading coefficient 1 such that for all z on unit circle,*

$$\left| \frac{z^{n+d}}{p(z)} - h(z) \right| \le \zeta .$$

Proof [Proof of Lemma 6.4] Let $\gamma = 1 - \alpha$. Using Lemma 6.5 with $\zeta = 0.5\|p\|_{\mathcal{H}_\infty}^{-1}$, we have that there exists polynomial u of degree $d = O(\max\{\frac{1}{1-\alpha}\log(\Gamma\|p\|_{\mathcal{H}_\infty}), 0\})$ such that

$$\left| \frac{z^{n+d}}{p(z)} - u(z) \right| \le \zeta .$$

Then we have that

$$\left| p(z)u(z)/z^{n+d} - 1 \right| \le \zeta|p(z)| < 0.5 .$$

Therefore $p(z)u(z)/z^{n+d} \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}$ for constant τ_0, τ_1, τ_2. Finally noting that for degree n polynomial we have $\|h\|_{\mathcal{H}_\infty} \le \sqrt{n} \cdot \|h\|_{\mathcal{H}_2}$, which completes the proof. ■

6.2.3 Example: Characteristic polynomial with random roots

We consider the following generative model for characteristic polynomial of degree $2n$. We generate n complex numbers $\lambda_1, \ldots, \lambda_n$ uniformly randomly on circle with radius $\alpha < 1$, and take $\lambda_i, \bar{\lambda}_i$ for $i = 1, \ldots, n$ as the roots of $p(z)$. That is, $p(z) = (z-\lambda_1)(z-\bar{\lambda}_1)\ldots(z-\lambda_n)(z-\bar{\lambda}_n)$. We show that with good probability (over the randomness of λ_i's), polynomial $p(z)$ will satisfy the condition in subsection 6.2.2 so that it can be learned efficiently by our improper learning algorithm.

Theorem 6.6 *Suppose $p(z)$ with random roots inside circle of radius α is generated from the process described above. Then with high probability over the choice of p, we have that $\Gamma(p) \le \exp(\widetilde{O}(\sqrt{n}))$ and $\|p\|_{\mathcal{H}_2} \le \exp(\tilde{O}(\sqrt{n}))$. As a corollary, $p(z)$ is α-acquiescent by extension of degree $\widetilde{O}((1-\alpha)^{-1}n)$.*

Towards proving Theorem 6.6, we need the following lemma about the expected distance of two random points with radius ρ and r in log-space.

Lemma 6.7 *Let $x \in \mathbb{C}$ be a fixed point with $|x| = \rho$, and λ uniformly drawn on the circle with radius r. Then $\mathbb{E}\left[\ln|x-\lambda|\right] = \ln\max\{\rho, r\}$.*

Proof When $r \neq \rho$, let N be an integer and $\omega = e^{2i\pi/N}$. Then we have that

$$\mathbb{E}[\ln|x-\lambda| \mid r] = \lim_{N\to\infty} \frac{1}{N}\sum_{k=1}^{N} \ln|x - r\omega^k| \tag{6.2}$$

The right hand of equation (6.2) can be computed easily by observing that $\frac{1}{N}\sum_{k=1}^{N}\ln|x - r\omega^k| = \frac{1}{N}\ln\left|\prod_{k=1}^{N}(x-r\omega^k)\right| = \frac{1}{N}\ln|x^N - r^N|$. Therefore, when $\rho > r$, we have $\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\ln|x - r\omega^k| = \lim_{N\to\infty}\rho + \frac{1}{N}\ln|(x/\rho)^N - (r/\rho)^N| = \ln\rho$. On the other hand, when $\rho < r$, we have that $\lim_{N\to\infty}\frac{1}{N}\sum_{k=1}^{N}\ln|x - r\omega^k| = \ln r$. Therefore we have that $\mathbb{E}[\ln|x-\lambda| \mid r] = \ln(\max\rho, r)$. For $\rho = r$, similarly proof (with more careful concern of regularity condition) we can show that $\mathbb{E}[\ln|x-\lambda| \mid r] = \ln r$. ■

Now we are ready to prove Theorem 6.6.

Proof [Proof of Theorem 6.6] Fixing index i, and the choice of λ_i, we consider the random variable $Y_i = \ln(\frac{|\lambda_i|^{2n}}{\prod_{j\neq i}|\lambda_i-\lambda_j|\prod_{j\neq i}|\lambda_i-\bar{\lambda}_j|})n\ln|\lambda_i| - \sum_{j\neq i}\ln|\lambda_i - \lambda_j|$. By Lemma 6.7, we have that $\mathbb{E}[Y_i] = n\ln|\lambda_i| - \sum_{j\neq i}\mathbb{E}[\ln|\lambda_i-\lambda_j|] = \ln(1-\delta)$. Let $Z_j = \ln|\lambda_i - \lambda_j|$. Then we have that Z_j are random variable with mean 0 and ψ_1-Orlicz norm bounded by 1 since $\mathbb{E}[e^{\ln|\lambda_i-\lambda_j|} - 1] \le 1$. Therefore by Bernstein inequality for sub-exponential tail random variable (for example, (Ledoux and Talagrand, 2013, Theorem 6.21)), we have that with high probability $(1-n^{-10})$, it holds that $\left|\sum_{j\neq i} Z_j\right| \le \widetilde{O}(\sqrt{n})$ where $\widetilde{O}$ hides logarithmic factors. Therefore, with high probability, we have $|Y_i| \le \widetilde{O}(\sqrt{n})$.

Finally we take union bound over all $i \in [n]$, and obtain that with high probability, for $\forall i \in [n], |Y_i| \le \widetilde{O}(\sqrt{n})$, which implies that $\sum_{i=1}^{n}\exp(Y_i) \le \exp(\widetilde{O}(\sqrt{n}))$. With similar technique, we can prove that $\|p\|_{\mathcal{H}_2} \le \exp(\tilde{O}(\sqrt{n}))$. ■

6.2.4 Example: Passive systems

We will show that with improper learning we can learn almost all passive systems, an important class of stable linear dynamical system as we discussed earlier. We start off with the definition of a strict-input passive system.

Definition 6.8 (Passive System, c.f Kottenstette and Antsaklis (2010)) *A SISO linear system is strict-input passive if and only if for some $\tau_0 > 0$ and any z on unit circle, $\Re(G(z)) \ge \tau_0$.*

In order to learn the passive system, we need to add assumptions in the definition of strict passivity. To make it precise, we define the following subsets of complex plane: For positive constant τ_0, τ_1, τ_2, define

$$\mathcal{C}^{+}_{\tau_0,\tau_1,\tau_2} = \{z \in \mathbb{C} : |z| \le \tau_2, \Re(z) \ge \tau_1, \Re(z) \ge \tau_0|\Im(z)|\}. \tag{6.3}$$

We say a transfer function $G(z) = s(z)/p(z)$ is (τ_0, τ_1, τ_2)-strict input passive if for any z on unit circle we have $G(z) \in \mathcal{C}^{+}_{\tau_0,\tau_1,\tau_2}$. Note that for small constant τ_0, τ_1 and large constant τ_2, this basically means the system is strict-input passive.

Now we are ready to state our main theorem in this subsection. We will prove that passive systems could be learned improperly with a constant factor more states (dimensions), assuming $s(z)$ has all its roots strictly inside unit circles and $\Gamma(s) \le \exp(O(n))$.

Theorem 6.9 *Suppose $G(z) = s(z)/p(z)$ is (τ_0, τ_1, τ_2)-strict-input passive. Moreover, suppose the roots of $s(z)$ have magnitudes inside circle with radius α and $\Gamma = \Gamma(s) \le \exp(O(n))$ and $\|p\|_{\mathcal{H}_2} \le \exp(O(n))$. Then $p(z)$ is α-acquiescent by extension of degree $d = O_{\tau,\alpha}(n)$, and as a consequence we can learn $G(z)$ with $n+d$ states in polynomial time.*

Moreover, suppose in addition we assume that $G(z) \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}$ for every z on unit circle. Then $p(z)$ is α-acquiescent by extension of degree $d = O_{\tau,\alpha}(n)$.

The proof of Theorem 6.9 is similar in spirit to that of Lemma 6.4, and is deferred to Section D.

6.3 Improper learning using linear regression

In this subsection, we show that under stronger assumption than α-acquiescent by extension, we can improperly learn a linear dynamical system with linear regression, up to some fixed bias.

The basic idea is to fit a linear function that maps $[x_{k-\ell}, \ldots, x_k]$ to y_k. This is equivalent to a dynamical system with ℓ hidden states and with the companion matrix A in (1.4) being chosen as $a_\ell = 1$ and $a_{\ell-1} = \cdots = a_1 = 0$. In this case, the hidden states exactly memorize all the previous ℓ inputs, and the output is a linear combination of the hidden states.

Equivalently, in the frequency space, this corresponds to fitting the transfer function $G(z) = s(z)/p(z)$ with a rational function of the form $\frac{c_1 z^{\ell-1} + \cdots + c_1}{z^{\ell-1}} = c_1 z^{-(\ell-1)} + \cdots + c_n$. The following is a sufficient condition on the characteristic polynomial $p(x)$ that guarantees the existence of such fitting,

Definition 6.10 *A polynomial $p(z)$ of degree n is extremely-acquiescent by extension of degree d with bias ε if there exists a polynomial $u(z)$ of degree d and leading coefficient 1 such that for all z on unit circle,*

$$\left|p(z)u(z)/z^{n+d} - 1\right| \le \varepsilon \tag{6.4}$$

We remark that if $p(z)$ is 1-acquiescent by extension of degree d, then there exists $u(z)$ such that $p(z)u(z)/z^{n+d} \in \mathcal{C}$. Therefore, equation (6.4) above is a much stronger requirement than acquiescence by extension.[7]

When $p(z)$ is extremely-acquiescent, we see that the transfer function $G(z) = s(z)/p(z)$ can be approximated by $s(z)u(z)/z^{n+d}$ up to bias ε. Let $\ell = n+d+1$ and $s(z)u(z) = c_1 z^{\ell-1} + \cdots + c_\ell$. Then we have that $G(z)$ can be approximated by the following dynamical system of ℓ hidden states with ε bias: we choose $A = \mathrm{CC}(a)$ with $a_\ell = 1$ and $a_{\ell-1} = \cdots = a_1 = 0$, and $C = [c_1, \ldots, c_\ell]$. As we have argued previously, such a dynamical system simply memorizes all the previous ℓ inputs, and therefore it is equivalent to linear regression from the feature $[x_{k-\ell}, \ldots, x_k]$ to output y_k.

Proposition 6.11 (Informal) *If the true system $G(z) = s(z)/p(z)$ satisfies that $p(z)$ is extremely-acquiescent by extension of degree d. Then using linear regression we can learn mapping from $[x_{k-\ell}, \ldots, x_k]$ to y_k with bias ε and polynomial sampling complexity.*

We remark that with linear regression the bias ε will only go to zero as we increase the length ℓ of the feature, but not as we increase the number of samples. Moreover, linear regression requires a stronger assumption than the improper learning results in previous subsections do. The latter can be viewed as an interpolation between the proper case and the regime where linear regression works.

7. Learning multi-input multi-output (MIMO) systems

We consider multi-input multi-output systems with the transfer functions that have a common denominator $p(z)$,

$$G(z) = \frac{1}{p(z)} \cdot S(z) \tag{7.1}$$

where $S(z)$ is an $\ell_{\text{in}} \times \ell_{\text{out}}$ matrix with each entry being a polynomial with real coefficients of degree at most n and $p(z) = z^n + a_1 z^{n-1} + \cdots + a_n$. Note that here we use ℓ_{in} to denote the dimension of the inputs of the system and ℓ_{out} the dimension of the outputs.

Although a special case of a general MIMO system, this class of systems still contains many interesting cases, such as the transfer functions studied in Fazel et al. (2001, 2004), where $G(z)$ is assumed to take the form $G(z) = R_0 + \sum_{i=1}^{n} \frac{R_i}{z-\lambda_i}$, for $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$ with conjugate symmetry and $R_i \in \mathbb{C}^{\ell_{\text{out}} \times \ell_{\text{in}}}$ satisfies that $R_i = \bar{R}_j$ whenever $\lambda_i = \bar{\lambda}_j$.

In order to learn the system $G(z)$, we parametrize $p(z)$ by its coefficients $a_1, \ldots, a_n$ and $S(z)$ by the coefficients of its entries. Note that each entry of $S(z)$ depends on $n+1$ real

7. We need $(1-\delta)$-acquiescence by extension in previous subsections for small $\delta > 0$, though this is merely additional technicality needed for the sample complexity. We ignore this difference between $1-\delta$-acquiescence and 1-acquiescence and for the purpose of this subsection

coefficients and therefore the collection of coefficients forms a third order tensor of dimension $\ell_{\text{out}} \times \ell_{\text{in}} \times (n+1)$. It will be convenient to collect the leading coefficients of the entries of $S(z)$ into a matrix of dimension $\ell_{\text{out}} \times \ell_{\text{in}}$, named D, and the rest of the coefficients into a matrix of dimension $\ell_{\text{out}} \times \ell_{\text{in}} n$, denoted by C. This will be particularly intuitive when a state-space representation is used to learn the system with samples as discussed later. We parameterize the training transfer function $\hat{G}(z)$ by $\hat{a}$, $\hat{C}$ and $\hat{D}$ using the same way.

Let's define the risk function in the frequency domain as,

$$g(\hat{A}, \hat{C}, \hat{D}) = \int_0^{2\pi} \left\| G(e^{i\theta}) - \hat{G}(e^{i\theta}) \right\|_F^2 d\theta \,. \tag{7.2}$$

The following lemma is an analog of Lemma 3.3 for the MIMO case. Itss proof actually follows from a straightforward extension of the proof of Lemma 3.3 by observing that matrix $S(z)$ (or $\hat{S}(z)$) commute with scalar $p(z)$ and $\hat{p}(z)$, and that $\hat{S}(z), \hat{p}(z)$ are linear in $\hat{a}, \hat{C}$.

Lemma 7.1 *The risk function $g(\hat{a}, \hat{C})$ defined in (7.2) is τ-weakly-quasi-convex in the domain*

$$\mathcal{N}_\tau(a) = \left\{ \hat{a} \in \mathbb{R}^n \colon \Re\left(\frac{p_a(z)}{p_{\hat{a}}(z)}\right) \ge \tau/2, \forall\, z \in \mathbb{C},\ \text{s.t. } |z| = 1 \right\} \otimes \mathbb{R}^{\ell_{\text{in}} \times \ell_{\text{out}} \times n'}$$

Finally, as alluded before, we use a particular state space representation for learning the system in time domain with example sequences. It is known that any transfer function of the form (7.1) can be realized uniquely by the state space system of the following special case of Brunovsky normal form Brunovsky (1970),

$$A = \begin{bmatrix} 0 & I_{\ell_{\text{in}}} & 0 & \cdots & 0 \\ 0 & 0 & I_{\ell_{\text{in}}} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & I_{\ell_{\text{in}}} \\ -a_n I_{\ell_{\text{in}}} & -a_{n-1} I_{\ell_{\text{in}}} & -a_{n-2} I_{\ell_{\text{in}}} & \cdots & -a_1 I_{\ell_{\text{in}}} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ I_{\ell_{\text{in}}} \end{bmatrix}, \tag{7.3}$$

and,

$$C \in \mathbb{R}^{\ell_{\text{out}} \times n\ell_{\text{in}}}, \quad D \in \mathbb{R}^{\ell_{\text{out}} \times \ell_{\text{in}}} \,.$$

The following Theorem is a straightforward extension of Corollary 5.2 and Theorem 6.2 to the MIMO case.

Theorem 7.2 *Suppose transfer function $G(z)$ of a MIMO system takes form (7.1), and has norm $\|G\|_{\mathcal{H}_2} \le 1$. If the common denominator $p(z)$ is α-acquiescent by extension of degree d then projected stochastic gradient descent over the state space representation (7.3) will return $\widehat{\Theta}$ with risk*

$$f(\widehat{\Theta}) \le \frac{\mathrm{poly}(n+d, \sigma, \tau, (1-\alpha)^{-1})}{TN} \,.$$

We note that since A and B are simply the tensor product of $I_{\ell_{\text{in}}}$ with $\mathrm{CC}(a)$ and e_n, the no blow-up property (Lemma 4.4) for $A^k B$ still remains true. Therefore to prove Theorem 7.2, we essentially only need to run the proof of Lemma 5.5 with matrix notation and matrix norm. We defer the proof to the full version.

8. Simulations

In this section, we provide proof-of-concepts experiments on synthetic data. We will demonstrate that

1) plain SGD tends to blow up even with relatively small learning rate, especially on hard instances

2) SGD with our projection step converges with reasonably large learning rate, and with over-parameterization the final error is competitive

3) SGD with gradient clipping has the strongest performance in terms both of the convergence speed and the final error

Here gradient clipping refers to the technique of using a normalized gradient instead of the true gradient. Specifically, for some positive hyper parameter B, we follow the approximate gradient

$$g_{\text{clip}} = \begin{cases} g & \text{if } \|g\| \leq B \\ Bg/\|g\| & otherwise \end{cases}$$

This method is commonly applied in training recurrent neural networks Pascanu et al. (2013).

Bullet 1) suggests that stability is indeed a real concern. Bullet 2) corroborates our theoretical study. Finding 3) suggests the instability of SGD partly arises from the noise in the batches, and such noise is reduced by the gradient clipping. Our experiments suggest that the landscape of the objective function may be even nicer than what is predicted by our theoretical development. It remains possible that the objective has no non-global local minima, possibly even outside the convex set to which our algorithm projects.

We generate the true system with state dimension $d = 20$ by randomly picking the conjugate pairs of roots of the characteristic polynomial inside the circle with radius $\rho = 0.95$ and randomly generating the vector C from standard normal distribution. The distribution of the norm of the impulse response r (defined in Section 3) of such systems has a heavy-tail. When the norm of r is several magnitudes larger than the median it's difficult to learn the system. Thus we select systems with reasonable $\|r\|$ for experiments, and we observe that the difficulty of learning increases as $\|r\|$ increases. The inputs of the dynamical model are generated from standard normal distribution with length $T = 500$. We note that we generate new fresh inputs and outputs at every iterations and therefore the training loss is equal to the test loss (in expectation.) We use initial learning rate 0.01 in the projected gradient descent and SGD with gradient clipping. We use batch size 100 for all experiments, and decay the learning rate at 200K and 250K iteration by a factor of 10 in all experiments.

Acknowledgments

We thank Amir Globerson, Alexandre Megretski, Pablo A. Parrilo, Yoram Singer, Peter Stoica, and Ruixiang Zhang for helpful discussions. We are indebted to Mark Tobenkin for pointers to relevant prior work. We also thank Alexandre Megretski for helpful feedback, insights into passive systems and suggestions on how to organize Section 3.

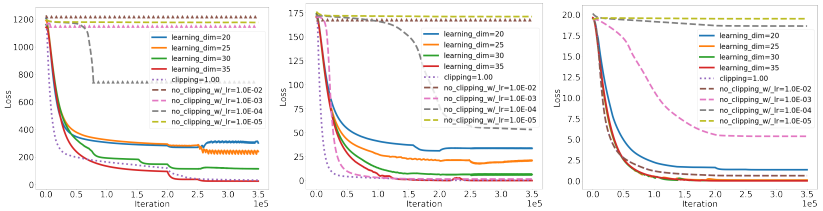


Figure 2: The performance of projected stochastic gradient descent with over-parameterization, vanilla SGD, and SGD with gradient clipping, on three different instance of dynamical systems with true state dimension = 20. The solid lines are from our proposed projected SGD with (over-parameterized) state dimension = 20, 25, 30, 35. The dot line corresponds to SGD with gradient clipped to Frobenius norm 1. The dashed lines correspond vanilla SGD and the triangle marker means the error blows up to infinity. The plot demonstrates the effect of the over-parameterization to our our algorithm. We note that the loss are different scales because the true systems in these three instances have different norms of impulse responses (which is equal to the loss of zero fitting).

References

Alexandre S. Bazanella, Michel Gevers, Ljubisa Miskovic, and Brian D.O. Anderson. Iterative minimization of h_2 control performance criteria. *Automatica*, 44:2549–2559, 2008.

Badri Narayan Bhaskar, Gongguo Tang, and Benjamin Recht. Atomic norm denoising with applications to line spectral estimation. *IEEE Transactions on Signal Processing*, 61(23): 5987–5999, 2013.

Léon Bottou. On-line learning in neural networks. chapter On-line Learning and Stochastic Approximations, pages 9–42. Cambridge University Press, New York, NY, USA, 1998. ISBN 0-521-65263-4. URL `http://dl.acm.org/citation.cfm?id=304710.304720`.

Pavol Brunovsky. A classification of linear controllable systems. *Kybernetika*, 06(3):(173)–188, 1970. URL `http://eudml.org/doc/28376`.

M. C. Campi and Erik Weyer. Finite sample properties of system identification methods. *IEEE Transactions on Automatic Control*, 47(8):1329–1334, 2002.

Hugh Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part i. *Robotics & Automation Magazine, IEEE*, 13(2):99–110, 2006.

Diego Eckhard and Alexandre Sanfelice Bazanella. On the global convergence of identification of output error models. In *Proc. 18th IFAC World congress*, 2011.

T. Estermann. *Complex numbers and functions*. Athlone Press, 1962. URL `https://books.google.com/books?id=ITbvAAAAMAAJ`.

Maryam Fazel, Haitham Hindi, and Stephen P Boyd. A rank minimization heuristic with application to minimum order system approximation. In *Proc. American Control Conference*, volume 6, pages 4734–4739. IEEE, 2001.

Maryam Fazel, Haitham Hindi, and S Boyd. Rank minimization and applications in system theory. In *Proc. American Control Conference*, volume 4, pages 3273–3278. IEEE, 2004.

Rong Ge, Furong Huang, Chi Jin, and Yang Yuan. Escaping from saddle points - online stochastic gradient for tensor decomposition. In *Proc. 28th COLT*, pages 797–842, 2015.

E. Hazan, K. Y. Levy, and S. Shalev-Shwartz. Beyond Convexity: Stochastic Quasi-Convex Optimization. *ArXiv e-prints*, July 2015.

Christiaan Heij, André Ran, and Freek van Schagen. *Introduction to mathematical systems theory : linear systems, identification and control*. Birkhäuser, Basel, Boston, Berlin, 2007. ISBN 3-7643-7548-5. URL `http://opac.inria.fr/record=b1130636`.

Joao P Hespanha. *Linear systems theory*. Princeton university press, 2009.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735–1780, 1997. doi: 10.1162/neco.1997.9.8.1735. URL `http://dx.doi.org/10.1162/neco.1997.9.8.1735`.

Nicholas Kottenstette and Panos J Antsaklis. Relationships between positive real, passive dissipative, & positive systems. In *American Control Conference (ACC), 2010*, pages 409–416. IEEE, 2010.

Michel Ledoux and Michel Talagrand. *Probability in Banach Spaces: isoperimetry and processes*, volume 23. Springer Science & Business Media, 2013.

J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient Descent Converges to Minimizers. *ArXiv e-prints*, February 2016.

Sergey Levine and Vladlen Koltun. Guided policy search. In *Proceedings of The 30th International Conference on Machine Learning*, pages 1–9, 2013.

Lennart Ljung. *System Identification. Theory for the user*. Prentice Hall, Upper Saddle River, NJ, 2nd edition, 1998.

Alexandre Megretski. Convex optimization in robust identification of nonlinear feedback. In *Proceedings of the 47th Conference on Decision and Control*, 2008.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318, 2013.

Ali Rahimi, Ben Recht, and Trevor Darrell. Learning appearance manifolds from video. In *Proc. IEEE CVPR*, 2005.

A. C. Schaeffer. Inequalities of a. markoff and s. bernstein for polynomials and related functions. *Bull. Amer. Math. Soc.*, 47(8):565–579, 08 1941. URL `http://projecteuclid.org/euclid.bams/1183503783`.

Parikshit Shah, Badri Narayan Bhaskar, Gongguo Tang, and Benjamin Recht. Linear system identification via atomic norm regularization. In *Proceedings of the 51st Conference on Decision and Control*, 2012.

Torsten Söderström and Petre Stoica. Some properties of the output error method. *Automatica*, 18(1):93–99, 1982.

Petre Stoica and Torsten Söderström. Uniqueness of prediction error estimates of multivariable moving average models. *IFAC Proceedings Volumes*, 15(4):199–204, 1982.

Petre Stoica and Torsten Söderström. Uniqueness of estimated k-step prediction models of arma processes. *Systems & control letters*, 4(6):325–331, 1984.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks. In *Proc. 27th NIPS*, pages 3104–3112, 2014. URL `http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks`.

M. Vidyasagar and Rajeeva L. Karandikar. A learning theory approach to system identification and stochastic adaptive control. *Journal of Process Control*, 18(3):421–430, 2008.

Erik Weyer and M. C. Campi. Finite sample properties of system identification methods. In *Proceedings of the 38th Conference on Decision and Control*, 1999.

Appendix A. Background on optimization

The proof below uses the standard analysis of gradient descent for non-smooth objectives and demonstrates that the argument still works for weakly-quasi-convex functions.

Proof [Proof of Proposition 2.3] We start by using the weakly-quasi-convex condition and then the rest follows a variant of the standard analysis of non-smooth projected sub-gradient descent[8]. We conditioned on θ_k, and have that

$$\begin{aligned}
\tau(f(\theta_k) - f(\theta^*)) &\le \nabla f(\theta_k)^\top(\theta_k - \theta^*) = \mathbb{E}[\mathfrak{r}(\theta_k)^\top(\theta_k - \theta^*) \mid \theta_k] \\
&= \mathbb{E}\left[\frac{1}{\eta}(\theta_k - w_{k+1})(\theta_k - \theta^*) \mid \theta_k\right] \\
&= \frac{1}{\eta}\left(\mathbb{E}\left[\|\theta_k - w_{k+1}\|^2 \mid \theta_k\right] + \|\theta_k - \theta^*\|^2 - \mathbb{E}\left[\|w_{k+1} - \theta^*\|^2 \mid \theta_k\right]\right) \\
&= \eta\,\mathbb{E}\left[\|\mathfrak{r}(\theta_k)\|^2\right] + \frac{1}{\eta}\left(\|\theta_k - \theta^*\|^2 - \mathbb{E}\left[\|w_{k+1} - \theta^*\|^2 \mid \theta_k\right]\right) \qquad \text{(A.1)}
\end{aligned}$$

where the first inequality uses weakly-quasi-convex and the rest of lines are simply algebraic manipulations. Since θ_{k+1} is the projection of w_{k+1} to $\mathcal{B}$ and θ^* belongs to $\mathcal{B}$, we have $\|w_{k+1} - \theta^*\| \ge \|\theta_{k+1} - \theta^*\|$. Together with (A.1), and

$$\mathbb{E}\left[\|\mathfrak{r}(\theta_k)\|^2\right] = \|\nabla f(\theta_k)\|^2 + \mathrm{Var}[\mathfrak{r}(\theta_k)] \le \Gamma(f(\theta_k) - f(\theta^*)) + V,$$

8. Although we used weak smoothness to get a slightly better bound

we obtain that

$$\tau(f(\theta_k)-f(\theta^*)) \le \eta\Gamma(f(\theta_k)-f(\theta^*)) + \eta V + \frac{1}{\eta}\left(\|\theta_k-\theta^*\|^2 - \mathbb{E}\left[\|\theta_{k+1}-\theta^*\|^2 \mid \theta_k\right]\right).$$

Taking expectation over all the randomness and summing over k we obtain that

$$\sum_{k=0}^{K-1}\mathbb{E}\left[f(\theta_k)-f(\theta^*)\right] \le \frac{1}{\tau-\eta\Gamma}\left(\eta KV + \frac{1}{\eta}\|\theta_0-\theta^*\|^2\right) \le \frac{1}{\tau-\eta\Gamma}\left(\eta KV + \frac{1}{\eta}R^2\right).$$

where we use the assumption that $\|\theta_0-\theta^*\| \le R$. Suppose $K \ge \frac{4R^2\Gamma^2}{V\tau^2}$, then we take $\eta = \frac{R}{\sqrt{VK}}$. Therefore we have that $\tau-\eta\Gamma \ge \tau/2$ and therefore

$$\sum_{k=0}^{K-1}\mathbb{E}\left[f(\theta_k)-f(\theta^*)\right] \le \frac{4R\sqrt{V}\sqrt{K}}{\tau}. \tag{A.2}$$

On the other hand, if $K \le \frac{4R^2\Gamma^2}{V\tau^2}$, we pick $\eta = \frac{\tau}{2\Gamma}$ and obtain that

$$\sum_{k=0}^{K-1}\mathbb{E}\left[f(\theta_k)-f(\theta^*)\right] \le \frac{2}{\tau}\left(\frac{\tau KV}{2\Gamma} + \frac{2\Gamma R^2}{\tau}\right) \le \frac{8\Gamma R^2}{\tau^2}. \tag{A.3}$$

Therefore using equation (A.3) and (A.2) we obtain that when choosing η properly according to K as above,

$$\mathop{\mathbb{E}}_{k\in[K]}\left[f(\theta_k)-f(\theta^*)\right] \le \max\left\{\frac{8\Gamma R^2}{\tau^2 K}, \frac{4R\sqrt{V}}{\tau\sqrt{K}}\right\}.$$

■

Appendix B. Toolbox

Lemma B.1 *Let $B = e_n \in \mathbb{R}^{n\times 1}$ and $\lambda \in [0, 2\pi]$, $w \in \mathbb{C}$. Suppose A with $\rho(A)\cdot|w| < 1$ has the controllable canonical form $A = \mathrm{CC}(a)$. Then*

$$(I-wA)^{-1}B = \frac{1}{p_a(w^{-1})}\begin{bmatrix} w^{-1} \\ w^{-2} \\ \vdots \\ w^{-n}\end{bmatrix}$$

where $p_a(x) = x^n + a_1x^{n-1} + \cdots + a_n$ is the characteristic polynomial of A.

Proof let $v = (I-wA)^{-1}B$ then we have $(I-wA)v = B$. Note that $B = e_n$, and $I-wA$ is of the form

$$I - wA = \begin{bmatrix} 1 & -w & 0 & \cdots & 0 \\ 0 & 1 & -w & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & -w \\ a_nw & a_{n-1}w & a_{n-2}w & \cdots & 1+a_1w \end{bmatrix} \tag{B.1}$$

Therefore we obtain that $v_k = wv_{k+1}$ for $1 \le k \le n-1$. That is, $v_k = v_0w^{-k}$ for $v_0 = v_1w^1$. Using the fact that $((I-wA)v)_n = 1$, we obtain that $v_0 = p_a(w^{-1})^{-1}$ where $p_a(\cdot)$ is the polynomial $p_a(x) = x^n + a_1x^{n-1} + \cdots + a_n$. Then we have that $u(I-wA)^{-1}B = \frac{u_1w^{-1}+\cdots+u_nw^{-n}}{p_a(w^{-1})}$ ■

Lemma B.2 *Suppose $x_1, \ldots, x_n$ are independent variables with mean 0 and covariance matrices and I, $U_1, \ldots, U_d$ are fixed matrices, then*

$$\mathbb{E}\left[\|\sum_{k=1}^n U_kx_k\|^2\right] = \sum_{k=1}^n \|U_k\|_F^2.$$

Proof We have that

$$\mathbb{E}\left[\|\sum_{k=1}^n U_kx_k\|_F^2\right] = \mathbb{E}\sum_{k,\ell}^n \mathrm{tr}(U_kx_kx_\ell^\top U_\ell^\top) = \sum_k^n \mathrm{tr}(U_kx_kx_k^\top U_k^\top) = \sum_{k=1}^n \|U_k\|_F^2$$

■

Appendix C. Missing proofs in Sections 4 and 5

C.1 Monotonicity of acquiescence: Proof of Lemma 4.3

Lemma C.1 (Lemma 4.3 restated) *For any $0 < \alpha < \beta$, we have that $\mathcal{B}_\alpha \subset \mathcal{B}_\beta$.*

Proof Let $q_a(z) = 1 + a_1z + \cdots + a_nz^n$. Note that $q(z^{-1}) = p_a(z)/z^n$. Therefore we note that $B_\alpha = \{a : q_a(z) \in \mathcal{C}, \forall |z| = 1/\alpha\}$. Suppose $a \in \mathcal{B}_\alpha$, then $\Re(q_a(z)) \ge \tau_1$ for any z with $|z| = 1/\alpha$. Since $\Re(q_a(z))$ is the real part of the holomorphic function $q_a(z)$, its a harmonic function. By maximum (minimum) principle of the harmonic functions, we have that for any $|z| \le 1/\alpha$, $\Re(q_a(z)) \ge \inf_{|z|=1/\alpha}\Re(q_a(z)) \ge \tau_1$. In particular, it holds that for $|z| = 1/\beta < 1/\alpha$, $\Re(q_a(z)) \ge \tau_1$. Similarly we can prove that for z with $|z| = 1/\beta$, $\Re(q_a(z)) \ge (1+\tau_0)\Im(q_a(z))$, and other conditions for a being in $\mathcal{B}_\beta$. ■

C.2 Proof of Lemma 5.4

Lemma 5.4 follows directly from the following general Lemma which also handles the multi-input multi-output case. It can be seen simply from calculation similar to the proof of Lemma 3.5. We mainly need to control the tail of the series using the no-blow up property (Lemma 4.4) and argue that the wrong value of the initial states h_0 won't cause any trouble to the partial loss function $\ell((x,y),\widehat{\Theta})$ (defined in Algorithm 1). This is simply because after time $T_1 = T/4$, the influence of the initial state is already washed out.

Lemma C.2 *In algorithm 3 the values of G_A, G_C, G_D are equal to the gradients of $g(\hat{A},\hat{C})+(\hat{D}-D)^2$ with respect to $\hat{A}$, $\hat{C}$ and $\hat{D}$ up to inverse exponentially small error.*

Proof [Proof of Lemma C.2] We first show that the partial empirical loss function $\ell((x,y),\widehat{\Theta})$ has expectation almost equal to the idealized risk (up to the term for $\hat{D}$ and exponential small error),

$$\mathbb{E}[\ell((x,y),\widehat{\Theta})] = g(\hat{A},\hat{C}) + (\hat{D}-D)^2 \pm \exp(-\Omega((1-\alpha)T)).$$

This can be seen simply from similar calculation to the proof of Lemma 3.5. Note that

$$y_t = Dx_t + \sum_{k=1}^{t-1} CA^{t-k-1}Bx_k + CA^{t-1}h_0 + \xi_t \quad \text{and} \quad \tilde{y}_t = \hat{D}x_t + \sum_{k=1}^{t-1} \hat{C}\hat{A}^{t-k-1}\hat{B}x_k\,. \tag{C.1}$$

Therefore noting that when $t \geq T_1 \geq \Omega(T)$, we have that $\|CA^{t-1}h_0\| \leq \exp(-\Omega((1-\alpha)T)$ and therefore the effect of h_0 is negligible. Then we have that

$$\begin{aligned}
\mathbb{E}[\ell((x,y),\widehat{\Theta})] &= \frac{1}{T-T_1}\,\mathbb{E}\left[\sum_{t>T_1}^{T} \|y_t - y_t\|^2\right] \pm \exp(-\Omega((1-\alpha)T)) \\
&= \|\hat{D}-D\|^2 + \frac{1}{T-T_1}\sum_{T\geq t>T_1}\sum_{0\leq j\leq t-1} \|\hat{C}\hat{A}^jB - CA^jB\|^2 \pm \exp(-\Omega((1-\alpha)T)) \\
&= \|\hat{D}-D\|^2 + \sum_{j=0}^{T_1} \|\hat{C}\hat{A}^jB - CA^jB\|^2 + \sum_{T\geq j\geq T_1} \frac{T-j}{T-T_1}\|\hat{C}\hat{A}^jB - CA^jB\|^2 \pm \exp(-\Omega((1-\alpha)T)) \\
&= \|\hat{D}-D\|^2 + \sum_{j=0}^{\infty} \|\hat{C}\hat{A}^jB - CA^jB\|^2 \pm \exp(-\Omega((1-\alpha)T))\,.
\end{aligned}$$

where the first line use the fact that $\|CA^{t-1}h_0\| \leq \exp(-\Omega((1-\alpha)T)$, the second uses equation (3.10) and the last line uses the no-blowing up property of A^kB (Lemma 4.4).

Similarly, we can prove that the gradient of $\mathbb{E}[\ell((x,y),\widehat{\Theta})]$ is also close to the gradient of $g(\hat{A},\hat{C}) + (\hat{D}-D)^2$ up to inverse exponential error. ■

C.3 Proof of Lemma 5.5

Proof [Proof of Lemma 5.5] Both G_A and G_C can be written in the form of a quadratic form (with vector coefficients) of $x_1,\ldots,x_T$ and $\xi_1,\ldots,\xi_T$. That is, we will write

$$G_A = \sum_{s,t} x_s x_t u_{st} + \sum_{s,t} x_s \xi_t u'_{st} \quad \text{and} \quad G_C = \sum_{s,t} x_s x_t v_{st} + \sum_{s,t} x_s \xi_t v'_{st}\,.$$

where u_{st} and v_{st} are vectors that will be calculated later. By Lemma C.3, we have that

$$\mathrm{Var}\left[\sum_{s,t} x_s x_s u_{st} + \sum_{s,t} x_s \xi_t u'_{st}\right] \leq O(1)\sum_{s,t}\|u_{st}\|^2 + O(\sigma^2)\sum_{s,t}\|u'_{st}\|^2\,. \tag{C.2}$$

Therefore in order to bound from above $\mathrm{Var}[G_A]$, it suffices to bound $\sum\|u_{st}\|^2$ and $\sum\|u'_{st}\|^2$, and similarly for G_C.

We begin by writing out u_{st} for fixed $s,t\in[T]$ and bounding its norm. We use the same set of notations as int the proof of Lemma 5.4. Recall that we set $r_k = CA^kB$ and $\hat{r}_k = \hat{C}\hat{A}^kB$, and $\Delta r_k = \hat{r}_k - r_k$. Moreover, let $z_k = \hat{A}^kB$. We note that the sums of $\|z_k\|^2$ and r_k^2 can be controlled. By the assumption of the Lemma, we have that

$$\sum_{k=t}^{\infty}\|z_k\|^2 \leq 2\pi n\tau_1^{-2}, \quad \|z_k\|^2 \leq 2\pi n\alpha^{2k-2n}\tau_1^{-2}\,. \tag{C.3}$$

$$\sum_{k=t}^{\infty}\Delta r_k^2 \leq 4\pi n\tau_1^{-2}, \quad \|\Delta r_k\|^2 \leq 4\pi n\alpha^{2k-2n}\tau_1^{-2}\,. \tag{C.4}$$

which will be used many times in the proof that follows.

We calculate the explicit form of G_A using the explicit back-propagation Algorithm 3. We have that in Algorithm 3,

$$\hat{h}_k = \textstyle\sum_{j=1}^{k}\hat{A}^{k-j}Bx_j = \sum_{j=1}^{k} z_{k-j}x_j \tag{C.5}$$

and

$$\Delta h_k = \sum_{j=k}^{T}(\hat{A}^\top)^{j-k}\hat{C}^\top\Delta y_j = \sum_{j=k}^{T}\alpha^{j-k}(\hat{A}^\top)^{j-k}\hat{C}^\top\mathbf{1}_{j>T_1}\left(\xi_j + \sum_{\ell=1}^{j}\Delta r_{j-\ell}x_\ell\right) \tag{C.6}$$

Then using $G_A = \sum_{k\geq 2} B^\top\Delta h_k h_{k-1}'^\top$ and equation (C.5) and equation (C.6) above, we have that

$$\begin{aligned}
u_{st} &= \textstyle\sum_{k=2}^{T}\left(\sum_{j\geq\max\{k,s,T_1+1\}}\Delta r_{j-s}\hat{C}\hat{A}^{j-k}B\right)\mathbf{1}_{k\geq t+1}\cdot\hat{A}^{k-t-1}B \\
&= \textstyle\sum_{k=2}^{T}\left(\sum_{j\geq\max\{k,s,T_1+1\}}\Delta r_{j-s}\hat{r}_{j-k}\right)\mathbf{1}_{k\geq t+1}\cdot z_{k-t-1}\,.
\end{aligned} \tag{C.7}$$

and that,

$$u'_{st} = \sum_{k=2}^{T} z_{k-1-s}\cdot\mathbf{1}_{k\geq s+1}\cdot\hat{r}'_{t-k}\cdot\mathbf{1}_{t>\max\{T_1,k\}} = \sum_{s+1\leq k\leq t} z_{k-1-s}\cdot\hat{r}'_{t-k}\cdot\mathbf{1}_{t>\max\{T_1\}} \tag{C.8}$$

Towards bounding $\|u_{st}\|$, we consider four different cases. Let $\Lambda = \Omega\left(\{\max\{n,(1-\alpha)^{-1}\log(\frac{1}{1-\alpha})\}\right)$ be a threshold.

Case 1: When $0\leq s-t\leq\Lambda$, we rewrite u_{st} by rearranging equation (C.7),

$$\begin{aligned}
u_{st} &= \sum_{T\geq k\geq s} z_{k-t-1}\sum_{j\geq\max\{k,T_1+1\}}\Delta r_{j-s}\hat{r}_{j-k} + \sum_{t<k<s} z_{k-t-1}\sum_{j\geq\max\{s,T_1+1\}}\Delta r_{j-s}\hat{r}_{j-k} \\
&= \sum_{\ell\geq 0,\ell\geq T_1+1-s}\Delta r_\ell\sum_{s\leq k\leq l+s,k\leq T}\hat{r}_{\ell+s-k}z_{k-t-1} + \sum_{\ell\geq 0,\ell\geq T_1+1-s}\Delta r_\ell\sum_{s>k>t}\hat{r}_{\ell+s-k}z_{k-t-1}
\end{aligned}$$

where at the second line, we did the change of variables $\ell = j - s$. Then by Cauchy-Schartz inequality, we have,

$$\|u_{st}\|^2 \le 2\left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\Delta r_\ell^2\right)\underbrace{\left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{s\le k\le l+s,k\le T}\hat{r}_{\ell+s-k}z_{k-t-1}\right\|^2\right)}_{T_1}$$
$$+2\left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\Delta r_\ell^2\right)\underbrace{\left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{s>k>t}\hat{r}_{\ell+s-k}z_{k-t-1}\right\|^2\right)}_{T_2}. \tag{C.9}$$

We could bound the contribution from Δr_k^2 ssing equation (C.4), and it remains to bound terms T_1 and T_2. Using the tail bounds for $\|z_k\|$ (equation (C.3)) and the fact that $|\hat{r}_k| = |\hat{C}\hat{A}^k B| \le \|\hat{A}^k B\| = \|z_k\|$, we have that

$$T_1 = \sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{s\le k\le l+s,k\le T}\hat{r}_{\ell+s-k}z_{k-t-1}\right\|^2 \le \sum_{\ell\ge 0}\left(\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\|z_{k-t-1}\|\right)^2. \tag{C.10}$$

We bound the inner sum of RHS of (C.10) using the fact that $\|z_k\|^2 \le O(n\alpha^{2k-2n}/\tau_1^2)$ and obtain that,

$$\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\|z_{k-t-1}\| \le \sum_{s\le k\le \ell+s}O(n\alpha^{(\ell+s-t-1)-2n}/\tau_1^2)$$
$$\le O(\ell n\alpha^{(\ell+s-t-1)-2n}/\tau_1^2). \tag{C.11}$$

Note that equation (C.11) is particular effective when $\ell > \Lambda$. When $\ell \le \Lambda$, we can refine the bound using equation (C.3) and obtain that

$$\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\|z_{k-t-1}\| \le \left(\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|^2\right)^{1/2}\left(\sum_{s\le k\le \ell+s}\|z_{k-t-1}\|^2\right)^{1/2}$$
$$\le O(\sqrt{n}/\tau_1)\cdot O(\sqrt{n}/\tau_1) = O(n/\tau_1^2). \tag{C.12}$$

Plugging equation (C.12) and (C.11) into equation (C.10), we have that

$$\sum_{\ell\ge 0}\left(\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\|z_{k-t-1}\|\right)^2 \le \sum_{\Lambda\ge\ell\ge 0}O(n^2/\tau_1^4)+\sum_{\ell>\Lambda}O(\ell^2n^2\alpha^{2(\ell+s-t-1)-4n}/\tau_1^4)$$
$$\le O(n^2\Lambda/\tau_1^4)+O(n^2/\tau_1^4) = O(n^2\Lambda/\tau_1^4). \tag{C.13}$$

For the second term in equation (C.9), we bound similarly,

$$T_2 \le \sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{s>k>t}\hat{r}_{\ell+s-k}z_{k-t-1}\right\|^2 \le O(n^2\Lambda/\tau_1^4). \tag{C.14}$$

Therefore using the bounds for T_1 and T_2 we obtain that,

$$\|u_{st}\|^2 \le O(n^3\Lambda/\tau_1^6) \tag{C.15}$$

Case 2: When $s - t > \Lambda$, we tighten equation (C.13) by observing that,

$$T_1 \le \sum_{\ell\ge 0}\left(\sum_{s\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\|z_{k-t-1}\|\right)^2 \le \alpha^{2(s-t-1)-4n}\sum_{\ell\ge 0}O(\ell^2n^2\alpha^{2\ell}/\tau_1^4)$$
$$\le \alpha^{s-t-1}\cdot O(n^2/(\tau_1^4(1-\alpha)^3)). \tag{C.16}$$

where we used equation (C.11). Similarly we can prove that

$$T_2 \le \alpha^{s-t-1}\cdot O(n^2/(\tau_1^4(1-\alpha)^3)).$$

Therefore, we have when $s - t \ge \Lambda$,

$$\|u_{st}\|^2 \le O(n^3/((1-\alpha)^3\tau_1^6))\cdot\alpha^{s-t-1}. \tag{C.17}$$

Case 3: When $-\Lambda \le s-t \le 0$, we can rewrite u_st and use the Cauchy-Schwartz inequality and obtain that

$$u_{st} = \sum_{T\ge k\ge t+1}z_{k-t-1}\sum_{j\ge\max\{k,T_1+1\}}\Delta r_{j-s}\hat{r}_{j-k} = \sum_{\ell\ge 0,\ell\ge T_1+1-s}\Delta r_\ell\sum_{t+1\le k\le l+s,k\le T}\hat{r}_{\ell+s-k}z_{k-t-1}.$$

and,

$$\|u_{st}\|^2 \le \left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\Delta r_\ell^2\right)\left(\sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{t+1\le k\le l+s,k\le T}\hat{r}_{\ell+s-k}z_{k-t-1}\right\|^2\right).$$

Using almost the same arguments as in equation (C.11) and (C.12), we that

$$\sum_{t+1\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\cdot\|z_{k-t-1}\| \le O(\ell n\alpha^{(\ell+s-t-1)-2n}/\tau_1^2)$$
$$\text{and}\qquad \sum_{t+1\le k\le \ell+s}|\hat{r}_{\ell+s-k}|\cdot\|z_{k-t-1}\| \le O(\sqrt{n}/\tau_1)\cdot O(\sqrt{n}/\tau_1) = O(n/\tau_1^2).$$

Then using a same type of argument as equation (C.13), we can have that

$$\sum_{\ell\ge 0,\ell\ge T_1+1-s}\left\|\sum_{t+1\le k\le l+s,k\le T}\hat{r}'_{\ell+s-k}z'_{k-t-1}\right\|^2 \le O(n^2\Lambda/\tau_1^4)+O(n^2/\tau_1^4)$$
$$= O(n^2\Lambda/\tau_1^4).$$

It follows that in this case $\|u_{st}\|$ can be bounded with the same bound in (C.15).

Case 4: When $s - t \le -\Lambda$, we use a different simplification of u_{st} from above. First of all, it follows (C.7) that

$$\begin{aligned}\|u_{st}\| &\le \sum_{k=2}^{T}\left(\sum_{j\ge\max\{k,s,T_1+1\}} \|\Delta r_{j-s}\hat{r}'_{j-k}z_{k-t-1}\|\mathbf{1}_{k\ge t+1}\right) \\ &\le \sum_{k\ge t+1}\|z'_{k-t-1}\| \sum_{j\ge\max\{k,T_1+1\}} |\Delta r_{j-s}\hat{r}'_{j-k}|\,.\end{aligned} \tag{C.18}$$

Since $j - s \ge k - s > 4n$ and it follows that

$$\begin{aligned}\sum_{j\ge\max\{k,T_1+1\}} |\Delta r_{j-s}\hat{r}'_{j-k}| &\le \sum_{j\ge\max\{k,T_1+1\}} O(\sqrt{n}/\tau_1\cdot\alpha^{j-s-n})\cdot O(\sqrt{n}/\tau_1\cdot\alpha^{j-k-n}) \\ &\le O(n/(\tau_1^2(1-\alpha))\cdot\alpha^{k-s-n})\end{aligned}$$

Then we have that

$$\begin{aligned}\|u_{st}\|^2 &\le \sum_{k\ge t+1}\|z'_{k-t-1}\| \sum_{j\ge\max\{k,T_1+1\}} |\Delta r_{j-s}\hat{r}'_{j-k}| \\ &\le \left(\sum_{k\ge t+1}\|z'_{k-t-1}\|^2\right)\left(\sum_{k\ge t+1}\left(\sum_{j\ge\max\{k,T_1+1\}} |\Delta r_{j-s}\hat{r}'_{j-k}|\right)^2\right) \\ &\le O(n/\tau_1^2)\cdot O(n^2/(\tau_1^4(1-\alpha)^3)\alpha^{t-s}) = O(n^3/(\tau_1^6\delta^3)\alpha^{t-s})\end{aligned}$$

Therefore, using the bound for $\|u_{st}\|^2$ obtained in the four cases above, taking sum over s, t, we obtain that

$$\begin{aligned}\sum_{1\le s,t\le T}\|u_{st}\|^2 &\le \sum_{s,t\in[T]:|s-t|\le\Lambda} O(n^3\Lambda/\tau_1^6) + \sum_{s,t:|s-t|\ge\Lambda} O(n^3/(\tau_1^6(1-\alpha)^3)\alpha^{|t-s|-1}) \\ &\le O(Tn^3\Lambda^2/\tau_1^6) + O(n^3/\tau_1^6) = O(Tn^3\Lambda^2/\tau_1^6)\,.\end{aligned} \tag{C.19}$$

We finished the bounds for $\|u_{st}\|$ and now we turn to bound $\|u'_{st}\|^2$. Using the formula for u'_{st} (equation C.8), we have that for $t \le s+1$, $u'_{st} = 0$. For $s + \Lambda \ge t \ge s + 2$, we have that by Cauchy-Schwartz inequality,

$$\|u'_{st}\| \le \left(\sum_{s+1\le k\le t}\|z_{k-1-s}\|^2\right)^{1/2}\left(\sum_{s+1\le k\le t}|\hat{r}'_{t-k}|^2\right)^{1/2} \le O(n/\tau_1^2) \le O(n/\tau_1^2)\,.$$

On the other hand, for $t > s + \Lambda$, by the bound that $|\hat{r}'_k|^2 \le \|z'_k\|^2 \le O(n\alpha^{2k-2n}/\tau_1^2)$, we have,

$$\begin{aligned}\|u'_{st}\| &\le \sum_{s+1\le k\le t-1}\|z_{k-1-s}\|\cdot|\hat{r}'_{t-k}| \le \sum_{s+1\le k\le t-1} n\alpha^{t-s-1}/\tau_1^2 \\ &\le O(n(t-s)\alpha^{t-s-1}/\tau_1^2)\,.\end{aligned}$$

Therefore taking sum over s, t, similarly to equation (C.19),

$$\sum_{s,t\in[T]}\|u'_{st}\|^2 \le O(Tn^2\Lambda/\tau_1^4)\,. \tag{C.20}$$

Then using equation (C.2) and equation (C.19) and (C.20), we obtain that

$$\mathrm{Var}[\|G_A\|^2] \le O\left(Tn^3\Lambda^2/\tau_1^6 + \sigma^2Tn^2\Lambda/\tau_1^4\right).$$

Hence, it follows that

$$\mathrm{Var}[G_A] \le \frac{1}{(T-T_1)^2}\mathrm{Var}[G_A] \le \frac{O\left(n^3\Lambda^2/\tau_1^6 + \sigma^2n^2\Lambda/\tau_1^4\right)}{T}\,.$$

We can prove the bound for G_C similarly. ∎

Lemma C.3 *Let $x_1, \ldots, x_T$ be independent random variables with mean 0 and variance 1 and 4-th moment bounded by $O(1)$, and u_{ij} be vectors for $i, j \in [T]$. Moreover, let $\xi_1, \ldots, \xi_T$ be independent random variables with mean 0 and variance σ^2 and u'_{ij} be vectors for $i, j \in [T]$. Then,*

$$\mathrm{Var}\left[\textstyle\sum_{i,j} x_ix_ju_{ij} + \sum_{i,j} x_i\xi_ju'_{ij}\right] \le O(1)\textstyle\sum_{i,j}\|u_{ij}\|^2 + O(\sigma^2)\sum_{i,j}\|u'_{ij}\|^2\,.$$

Proof

Note that the two sums in the target are independent with mean 0, therefore we only need to bound the variance of both sums individually. The proof follows the linearity of expectation and the independence of x_i's:

$$\begin{aligned}\mathbb{E}\left[\left\|\textstyle\sum_{i,j} x_ix_ju_{ij}\right\|^2\right] &= \sum_{i,j}\sum_{k,\ell}\mathbb{E}\left[x_ix_jx_kx_\ell u_{ij}^\top u_{k\ell}\right] \\ &= \sum_i\mathbb{E}[u_{ii}^\top u_{ii}x_i^4] + \sum_{i\ne j}\mathbb{E}[u_{ii}^\top u_{jj}x_i^2x_j^2] + \sum_{i,j}\mathbb{E}\left[x_i^2x_j^2(u_{ij}^\top u_{ij} + u_{ij}^\top u_{ji})\right] \\ &\le \sum_{i,j}u_{ii}^\top u_{jj} + O(1)\sum_{i,j}\|u_{ij}+u_{ji}\|^2 \\ &= \|\textstyle\sum_i u_{ii}\|^2 + O(1)\sum_{i,j}\|u_{ij}\|^2\end{aligned}$$

where at second line we used the fact that for any monomial x^α with an odd degree on one of the x_i's, $\mathbb{E}[x^\alpha] = 0$. Note that $\mathbb{E}[\sum_{i,j} x_ix_ju_{ij}] = \sum_i u_{ii}$. Therefore,

$$\mathrm{Var}\left[\textstyle\sum_{i,j} x_ix_ju_{ij}\right] = \mathbb{E}\left[\|\textstyle\sum_{i,j} x_ix_ju_{ij}\|^2\right] - \|\mathbb{E}[\textstyle\sum_{i,j} x_ix_ju_{ij}]\|^2 \le O(1)\textstyle\sum_{i,j}\|u_{ij}\|^2 \tag{C.21}$$

Similarly, we can control $\mathrm{Var}\left[\sum_{i,j} x_i\xi_ju'_{ij}\right]$ by $O(\sigma^2)\sum_{i,j}\|u'_{ij}\|^2$. ∎

Appendix D. Missing proofs in Section 6

D.1 Proof of Lemma 6.5

Towards proving Lemma 6.5, we use the following lemma to express the inverse of a polynomial as a sum of inverses of degree-1 polynomials.

Lemma D.1 *Let $p(z) = (z-\lambda_1)\dots(z-\lambda_n)$ where λ_j's are distinct. Then we have that*

$$\frac{1}{p(z)} = \sum_{j=1}^{n} \frac{t_j}{z-\lambda_j}, \qquad \text{where} \quad t_j = \left(\prod_{i\neq j}(\lambda_j-\lambda_i)\right)^{-1}. \tag{D.1}$$

Proof [Proof of Lemma D.1] By interpolating constant function at points $\lambda_1,\dots,\lambda_n$ using Lagrange interpolating formula, we have that

$$1 = \sum_{j=1}^{n} \frac{\prod_{i\neq j}(x-\lambda_i)}{\prod_{i\neq j}(\lambda_j-\lambda_i)} \cdot 1 \tag{D.2}$$

Dividing $p(z)$ on both sides we obtain equation (D.1). ■

The following lemma computes the Fourier transform of function $1/(z-\lambda)$.

Lemma D.2 *Let $m \in \mathbb{Z}$, and $\mathcal{K}$ be the unit circle in complex plane, and $\lambda \in \mathbb{C}$ inside the $\mathcal{K}$. Then we have that*

$$\int_{\mathcal{K}} \frac{z^m}{z-\lambda}\,\mathrm{d}z = \begin{cases} 2\pi i\lambda^m & \text{for } m \geq 0 \\ 0 & \text{o.w.} \end{cases}$$

Proof [Proof of Lemma D.2] For $m \geq 0$, since z^m is a holomorphic function, by Cauchy's integral formula, we have that

$$\int_{\mathcal{K}} \frac{z^m}{z-\lambda}\,\mathrm{d}z = 2\pi i\lambda^m.$$

For $m < 0$, by changing of variable $y = z^{-1}$ we have that

$$\int_{\mathcal{K}} \frac{z^m}{z-\lambda}\,\mathrm{d}z = \int_{\mathcal{K}} \frac{y^{-m-1}}{1-\lambda y}\,\mathrm{d}y.$$

since $|\lambda y| = |\lambda| < 1$, then we by Taylor expansion we have,

$$\int_{\mathcal{K}} \frac{y^{-m-1}}{1-\lambda y}\,\mathrm{d}y = \int_{\mathcal{K}} y^{-m-1}\left(\sum_{k=0}^{\infty}(\lambda y)^k\right)\,\mathrm{d}y.$$

Since the series λy is dominated by $|\lambda|^k$ which converges, we can switch the integral with the sum. Note that y^{-m-1} is holomorphic for $m < 0$, and therefore we conclude that

$$\int_{\mathcal{K}} \frac{y^{-m-1}}{1-\lambda y}\,\mathrm{d}y = 0.$$

■

Now we are ready to prove Lemma 6.5.

Proof [Proof of Lemma 6.5] Let $m = n+d$. We compute the Fourier transform of $z^m/p(z)$. That is, we write

$$\frac{e^{im\theta}}{p(e^{i\theta})} = \sum_{k=-\infty}^{\infty} \beta_k e^{ik\theta}.$$

where

$$\beta_k = \frac{1}{2\pi}\int_0^{2\pi} \frac{e^{i(m-k)\theta}}{p(e^{i\theta})}d\theta = \frac{1}{2\pi i}\int_{\mathcal{K}} \frac{z^{m-k-1}}{p(z)}\,\mathrm{d}z$$

By Lemma D.1, we write

$$\frac{1}{p(z)} = \sum_{j=1}^{n} \frac{t_j}{z-\lambda_j}.$$

Then it follows that

$$\beta_k = \frac{1}{2\pi i}\sum_{j=1}^{n} t_j \int_{\mathcal{K}} \frac{z^{m-k-1}}{z-\lambda_j}\,\mathrm{d}z$$

Using Lemma D.2, we obtain that

$$\beta_k = \begin{cases} \sum_{j=1}^{n} t_j\lambda_j^{m-k-1} & \text{if } -\infty \leq k \leq m-1 \\ 0 & \text{o.w.} \end{cases} \tag{D.3}$$

We claim that

$$\sum_{j=1}^{n} t_j\lambda_j^{n-1} = 1, \qquad \text{and} \qquad \sum_{j=1}^{n} t_j\lambda_j^{s} = 0, \quad 0 \leq s < n-1.$$

Indeed these can be obtained by writing out the lagrange interpolation for polynomial $f(x) = x^s$ with $s \leq n-1$ and compare the leading coefficient. Therefore, we further simplify β_k to

$$\beta_k = \begin{cases} \sum_{j=1}^{n} t_j\lambda_j^{m-k-1} & \text{if } -\infty < k < m-n \\ 1 & \text{if } k = m-n \\ 0 & \text{o.w.} \end{cases} \tag{D.4}$$

Let $h(z) = \sum_{k\geq 0}\beta_k z^k$. Then we have that $h(z)$ is a polynomial with degree $d = m-n$ and leading term 1. Moreover, for our choice of d,

$$\left|\frac{z^m}{p(z)} - h(z)\right| = \left|\sum_{k<0}\beta_k z^k\right| \leq \sum_{k<0}|\beta_k| \leq \max_j |t_j|(1-\lambda_j)^n \sum_{k<0}(1-\gamma)^{d-k-1}$$
$$\leq \Gamma(1-\gamma)^d/\gamma < \zeta.$$

■

D.2 Proof of Theorem 6.9

Theorem 6.9 follows directly from a combination of Lemma D.3 and Lemma D.4 below. Lemma D.3 shows that the denominator of a function (under the stated assumptions) can be extended to a polynomial that takes values in $\mathcal{C}^+$ on unit circle. Lemma D.4 shows that it can be further extended to another polynomial that takes values in $\mathcal{C}$.

Lemma D.3 *Suppose the roots of s are inside circle with radius $\alpha < 1$, and $\Gamma = \Gamma(s)$. If transfer function $G(z) = s(z)/p(z)$ satisfies that $G(z) \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}$ (or $G(z) \in \mathcal{C}^+_{\tau_0,\tau_1,\tau_2}$) for any z on unit circle, then there exists $u(z)$ of degree $d = O_\tau(\max\{(\frac{1}{1-\alpha}\log\frac{\sqrt{n}\Gamma\cdot\|p\|_{\mathcal{H}_2}}{1-\alpha}, 0\})$ such that $p(z)u(z)/z^{n+d} \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$ (or $p(z)u(z)/z^{n+d} \in \mathcal{C}^+_{\tau_0',\tau_1',\tau_2'}$ respectively) for $\tau' = \Theta_\tau(1)$, where $O_\tau(\cdot), \Theta_\tau(\cdot)$ hide the polynomial dependencies on τ_0, τ_1, τ_2.*

Proof [Proof of Lemma D.3] By the fact that $G(z) = s(z)/p(z) \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}$, we have that $p(z)/s(z) \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$ for some τ' that polynomially depend on τ. Using Lemma 6.5, there exists $u(z)$ of degree d such that

$$\left|\frac{z^{n+d}}{s(z)} - u(z)\right| \le \zeta\,.$$

where we set $\zeta \ll \min\{\tau_0', \tau_1'\}/\tau_2' \cdot \|p\|_{\mathcal{H}_\infty}^{-1}$. Then we have that

$$\left|p(z)u(z)/z^{n+d} - \frac{p(z)}{s(z)}\right| \le |p(z)|\zeta \ll \min\{\tau_0', \tau_1'\}. \tag{D.5}$$

It follows from equation (D.5) implies that that $p(z)u(z)/z^{n+d} \in \mathcal{C}_{\tau_0'',\tau_1'',\tau_2''}$, where τ'' polynomially depends on τ. The same proof still works when we replace $\mathcal{C}$ by $\mathcal{C}^+$. ■

Lemma D.4 *Suppose $p(z)$ of degree n and leading coefficient 1 satisfies that $p(z) \in \mathcal{C}^+_{\tau_0,\tau_1,\tau_2}$ for any z on unit circle. Then there exists $u(z)$ of degree d such that $p(z)u(z)/z^{n+d} \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$ for any z on unit circle with $d = O_\tau(n)$ and $\tau_0', \tau_1', \tau_2' = \Theta_\tau(1)$, where $O_\tau(\cdot), \Theta_\tau(\cdot)$ hide the dependencies on τ_0, τ_1, τ_2.*

Proof [Proof of Lemma D.4] We fix z on unit circle first. Let's defined $\sqrt{p(z)/z^n}$ be the square root of $p(z)/z^n$ with principle value. Let's write $p(z)/z^n = \tau_2(1 + (\frac{p(z)}{\tau_2 z^n} - 1))$ and we take Taylor expansion for $\frac{1}{\sqrt{p(z)/z^n}} = \tau_2^{-1/2}(1 + (\frac{p(z)}{\tau_2 z^n} - 1))^{-1/2} = \tau_2^{-1/2}\left(\sum_{k=0}^{\infty}(\frac{p(z)}{\tau_2 z^n} - 1)^k\right)$. Note that since $\tau_1 < |p(z)| < \tau_2$, we have that $|\frac{p(z)}{\tau_2 z^n} - 1| < 1 - \tau_1/\tau_2$. Therefore truncating the Taylor series at $k = O_\tau(1)$ we obtain a polynomial a rational function $h(z)$ of the form

$$h(z) = \textstyle\sum_{j\ge 0}^{k}(\frac{p(z)}{\tau_2 z^n} - 1)^j\,,$$

which approximates $\frac{1}{\sqrt{p(z)/z^n}}$ with precision $\zeta \ll \min\{\tau_0, \tau_1\}/\tau_2$, that is, $\left|\frac{1}{\sqrt{p(z)/z^n}} - h(z)\right| \le \zeta$. Therefore, we obtain that $\left|\frac{p(z)h(z)}{z^n} - \sqrt{p(z)/z^n}\right| \le \zeta|p(z)/z^n| \le \zeta\tau_2$. Note that since $p(z)/z^n \in \mathcal{C}^+_{\tau_0,\tau_1,\tau_2}$, we have that $\sqrt{p(z)/z^n} \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$ for some constants $\tau_0', \tau_1', \tau_2'$. Therefore $\frac{p(z)h(z)}{z^n} \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$. Note that $h(z)$ is not a polynomial yet. Let $u(z) = z^{nk}h(z)$ and then $u(z)$ is polynomial of degree at most nk and $p(z)u(z)/z^{(n+1)k} \in \mathcal{C}_{\tau_0',\tau_1',\tau_2'}$ for any z on unit circle. ■

Appendix E. Back-propagation implementation

In this section we give a detailed implementation of using back-propagation to compute the gradient of the loss function. The algorithm is for general MIMO case with the parameterization (7.3). To obtain the SISO sub-case, simply take $\ell_{\text{in}} = \ell_{\text{out}} = 1$.

Algorithm 3 Back-propagation

Parameters: $\hat{a} \in \mathbb{R}^n$, $\hat{C} \in \mathbb{R}^{\ell_{\text{in}} \times n\ell_{\text{out}}}$, and $\hat{D} \in \mathbb{R}^{\ell_{\text{in}} \times \ell_{\text{out}}}$. Let $\hat{A} = \text{MCC}(\hat{a}) = \text{CC}(\hat{a}) \otimes I_{\ell_{\text{in}}}$ and $\hat{B} = e_n \otimes I_{\ell_{\text{in}}}$.
Input: samples $((x^{(1)}, y^1), \ldots, x^{(N)}, y^{(N)})$ and projection set $\mathcal{B}_\alpha$.

for each sample $(x^{(j)}, y^j) = ((x_1, \ldots, x_T), (y_1, \ldots, y_T))$ **do**
  Feed-forward pass:
    $h_0 = 0 \in \mathbb{R}^{n\ell_{\text{in}}}$.
    for $k = 1$ to T
      $\hat{h}_k \leftarrow \hat{A}\hat{h}_{k-1} + \hat{B}x_k$, $\hat{y}_t \leftarrow \hat{C}h_k + \hat{D}x_k$ and $\hat{h}_k \leftarrow \hat{A}h_{k-1} + \hat{B}x_k$.
    end for
  Back-propagation:
    $\Delta h_{T+1} \leftarrow 0$, $G_A \leftarrow 0$, $G_C \leftarrow 0$. $G_D \leftarrow 0$
    $T_1 \leftarrow T/4$
    for $k = T$ to 1
      if $k > T_1$, $\Delta y_k \leftarrow \hat{y}_k - y_k$, o.w. $\Delta y_k \leftarrow 0$. Let $\Delta h_k \leftarrow \hat{C}^\top \Delta y_k + \hat{A}^\top \Delta h_{k+1}$.
      update $G_C \leftarrow G_C + \frac{1}{T-T_1}\Delta y_k \hat{h}_k$, $G_A \leftarrow G_A - \frac{1}{T-T_1}B^\top \Delta h_k \hat{h}_{k-1}^\top$, and $G_D \leftarrow G_D + \frac{1}{T-T_1}\Delta y_k x_k$.
    end for
  Gradient update: $\hat{A} \leftarrow \hat{A} - \eta \cdot G_A$, $\hat{C} \leftarrow \hat{C} - \eta \cdot G_C$, $\hat{D} \leftarrow \hat{D} - \eta \cdot G_D$.
  Projection step: Obtain $\hat{a}$ from $\hat{A}$ and set $\hat{a} \leftarrow \Pi_{\mathcal{B}}(\hat{a})$, and $\hat{A} = \text{MCC}(\hat{a})$
end for

Appendix F. Projection to the set $\mathcal{B}_\alpha$

In order to have a fast projection algorithm to the convex set $\mathcal{B}_\alpha$, we consider a grid $\mathcal{G}_M$ of size M over the circle with radius α. We will show that $M = O_\tau(n)$ will be enough to approximate the set $\mathcal{B}_\alpha$ in the sense that projecting to the approximating set suffices for the convergence.

Let $\mathcal{B}'_{\alpha,\tau_0,\tau_1,\tau_2} = \{a : p_a(z)/z^n \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}, \forall z \in \mathcal{G}_M\}$ and $\mathcal{B}_{\alpha,\tau_0,\tau_1,\tau_2} = \{a : p_a(z)/z^n \in \mathcal{C}_{\tau_0,\tau_1,\tau_2}, \forall |z| = \alpha\}$. Here $\mathcal{C}_{\tau_0,\tau_1,\tau_2}$ is defined the same as before though we used the subscript

to emphasize the dependency on τ_i's,

$$\mathcal{C}_{\tau_0,\tau_1,\tau_2} = \{z : \Re z \geq (1+\tau_0)|\Im z|\} \cap \{z : \tau_1 < \Re z < \tau_2\}\,. \tag{F.1}$$

We will first show that with $M = O_\tau(n)$, we can make $\mathcal{B}'_{\alpha,\tau_1,\tau_2,\tau_3}$ to be sandwiched within to two sets $\mathcal{B}_{\alpha,\tau_0,\tau_1,\tau_2}$ and $\mathcal{B}_{\alpha,\tau'_0,\tau'_1,\tau'_2}$.

Lemma F.1 *For any $\tau_0 > \tau'_0, \tau_1 > \tau'_1, \tau_2 < \tau'_2$, we have that for $M = O_\tau(n)$, there exists $\kappa_0, \kappa_1, \kappa_2$ that polynomially depend on τ_i, τ'_i's such that $\mathcal{B}_{\alpha,\tau_0,\tau_1,\tau_2} \subset \mathcal{B}'_{\alpha,\kappa_0,\kappa_1,\kappa_2} \subset \mathcal{B}_{\alpha,\tau'_0,\tau'_1,\tau'_2}$*

Before proving the lemma, we demonstrate how to use the lemma in our algorithm: We will pick $\tau'_0 = \tau_0/2$, $\tau'_1 = \tau_1/2$ and $\tau'_2 = 2\tau_2$, and find κ_i's guaranteed in the lemma above. Then we use $\mathcal{B}'_{\alpha,\kappa_0,\kappa_1,\kappa_2}$ as the projection set in the algorithm (instead of $\mathcal{B}_{\alpha,\tau_0,\tau_1,\tau_2}$)). First of all, the ground-truth solution Θ is in the set $\mathcal{B}'_{\alpha,\kappa_0,\kappa_1,\kappa_2}$. Moreover, since $\mathcal{B}'_{\alpha,\kappa_0,\kappa_1,\kappa_2} \subset \mathcal{B}_{\alpha,\tau'_0,\tau'_1,\tau'_2}$, we will guarantee that the iterates $\widehat{\Theta}$ will remain in the set $\mathcal{B}_{\alpha,\tau'_0,\tau'_1,\tau'_2}$ and therefore the quasi-convexity of the objective function still holds[9].

Note that the set $\mathcal{B}'_{\alpha,\kappa_0,\kappa_1,\kappa_2}$ contains $O(n)$ linear constraints and therefore we can use linear programming to solve the projection problem. Moreover, since the points on the grid forms a Fourier basis and therefore Fast Fourier transform can be potentially used to speed up the projection. Finally, we will prove Lemma F.1. We need S. Bernstein's inequality for polynomials.

Theorem F.2 (Bernstein's inequality, see, for example, Schaeffer (1941)) *Let $p(z)$ be any polynomial of degree n with complex coefficients. Then,*

$$\sup_{|z|\leq 1} |p'(z)| \leq n \sup_{|z|\leq 1} |p(z)|.$$

We will use the following corollary of Bernstein's inequality.

Corollary F.3 *Let $p(z)$ be any polynomial of degree n with complex coefficients. Then, for $m = 20n$,*

$$\sup_{|z|\leq 1} |p'(z)| \leq 2n \sup_{k\in[m]} |p(e^{2ik\pi/m})|.$$

Proof For simplicity let $\tau = \sup_{k\in[m]} |p(e^{2ik\pi/m})|$, and let $\tau' = \sup_{k\in[m]} |p(e^{2ik\pi/m})|$. If $\tau' \leq 2\tau$ then we are done by Bernstein's inequality. Now let's assume that $\tau' > 2\tau$. Suppose $p(z) = \tau'$. Then there exists k such that $|z - e^{2\pi ik/m}| \leq 4/m$ and $|p(e^{2\pi ik/m})| \leq \tau$. Therefore by Cauchy mean-value theorem we have that there exists ξ that lies between z and $e^{2\pi ik/m}$ such that $p'(\xi) \geq m(\tau' - \tau)/4 \geq 1.1n\tau'$, which contradicts Bernstein's inequality. ■

Lemma F.4 *Suppose a polynomial of degree n satisfies that $|p(w)| \leq \tau$ for every $w = \alpha e^{2i\pi k/m}$ for some $m \geq 20n$. Then for every z with $|z| = \alpha$ there exists $w = \alpha e^{2i\pi k/m}$ such that $|p(z) - p(w)| \leq O(n\alpha\tau/m)$.*

9. with a slightly worse parameter up to constant factor since τ_i's are different from τ_i's up to constant factors

Proof Let $g(z) = p(\alpha z)$ by a polynomial of degree at most n. Therefore we have $g'(z) = \alpha p(z)$. Let $w = \alpha e^{2i\pi k/m}$ such that $|z - w| \leq O(\alpha/m)$. Then we have

$$\begin{aligned}
|p(z) - p(w)| = |g(z/\alpha) - p(w/\alpha)| &\leq \sup_{|x|\leq 1} |g'(x)| \cdot \frac{1}{\alpha}|z - w| \\
&\qquad \text{(By Cauchy's mean-value Theorem)} \\
&\leq \sup_{|x|\leq 1} |p'(x)| \cdot |z - w| \leq n\tau|z - w|\,. \quad \text{(Corallary F.3)} \\
&\leq O(\alpha n\tau/m)\,.
\end{aligned}$$

■

Now we are ready to prove Lemma F.1.

Proof [Proof of Lemma F.1] We choose $\kappa_i = \frac{1}{2}(\tau_i + \tau'_i)$.The first inequality is trivial. We prove the second one. Consider a such that $a \in \mathcal{B}_{\alpha,\kappa_0,\kappa_1,\kappa_2}$. We wil show that $a \in \mathcal{B}'_{\alpha,\tau'_0,\tau'_1,\tau'_2}$. Let $q_a(z) = p(z^{-1})z^n$. By Lemma F.4, for every z with $|z| = 1/\alpha$, we have that there exists $w = \alpha^{-1}e^{2\pi ik/M}$ for some integer k such that $|q_a(z) - q_a(w)| \leq O(\tau_2 n/(\alpha M))$. Therefore let $M = cn$ for sufficiently large c (which depends on τ_i's), we have that for every z with $|z| = 1/\alpha$, $q_a(z) \in \mathcal{C}_{\tau'_0,\tau'_1,\tau'_2}$. This completes the proof. ■

Journal of Machine Learning Research 19 (2018) 1-29 Submitted 11/16; Revised 8/18; Published 8/18

Parallelizing Spectrally Regularized Kernel Algorithms

Nicole Mücke NICOLE.MUECKE@MATHEMATIK.UNI-STUTTGART.DE
Institute of Stochastics and Applications, University of Stuttgart
Pfaffenwaldring 57
70569 Stuttgart, Germany

Gilles Blanchard* GILLES.BLANCHARD@MATH.UNI-POTSDAM.DE
Institute of Mathematics, University of Potsdam
Karl-Liebknecht-Strae 24-25
14476 Potsdam, Germany

Editor: Ingo Steinwart

Abstract

We consider a distributed learning approach in supervised learning for a large class of spectral regularization methods in an reproducing kernel Hilbert space (RKHS) framework. The data set of size n is partitioned into $m = O(n^\alpha)$, $\alpha < \frac{1}{2}$, disjoint subsamples. On each subsample, some spectral regularization method (belonging to a large class, including in particular Kernel Ridge Regression, L^2-boosting and spectral cut-off) is applied. The regression function f is then estimated via simple averaging, leading to a substantial reduction in computation time. We show that minimax optimal rates of convergence are preserved if m grows sufficiently slowly (corresponding to an upper bound for α) as $n \to \infty$, depending on the smoothness assumptions on f and the intrinsic dimensionality. In spirit, the analysis relies on a classical bias/stochastic error analysis.

Keywords: Distributed Learning, Spectral Regularization, Minimax Optimality

1. Introduction

Distributed learning (DL) algorithms are a standard tool for reducing computational burden in machine learning problems where massive datasets are involved. Assuming a complexity cost (for time and/or memory) of $O(n^\beta)$ ($\beta > 1$, $\beta \in [2,3]$ being common) of the base learning algorithm without parallelization, dividing randomly data of cardinality n into m disjoint, equally-sized subsamples and processing them in parallel using the same base learning algorithm has therefore complexity cost of $O(m.(n/m)^\beta) = O(n^\beta/m^{\beta-1})$, roughly gaining a factor $m^{\beta-1}$ (for time and memory) compared to the single machine approach. The final output is obtained from averaging the individual outputs.

Recently, DL was studied in several machine learning contexts. In point estimation (Li et al., 2013), matrix factorization (Mackey et al., 2011), smoothing spline models and testing (Cheng and Shang, 2016), local average regression (Chang et al., 2017), in classification (Hsieh et al., 2014; Guo et al., 2015), and also in kernel ridge regression (Zhang et al., 2013; Lin et al., 2017; Xu et al., 2016).

In this paper, we study the DL approach for the statistical learning problem

$$Y_i := f(X_j) + \epsilon_i \,, j = 1 \,, \ldots, n \,, \tag{1}$$

at random i.i.d. data points $X_1, \ldots, X_n$ drawn according to a probability distribution ν on $\mathcal{X}$, where ϵ_j are independent centered noise variables. The unknown regression function f is real-valued and belongs to some reproducing kernel Hilbert space with bounded kernel K. We partition the given data set $D = \{(X_1, Y_1), \ldots, (X_n, Y_n)\} \subset \mathcal{X} \times \mathbb{R}$ into m disjoint equal-size subsamples $D_1, \ldots, D_m$. On each subsample D_j, we compute a local estimator $\hat{f}^\lambda_{D_j}$, using a spectral regularization method. The final estimator for the target function f is obtained by simple averaging: $\bar{f}^\lambda_D := \frac{1}{m}\sum_{j=1}^m \hat{f}^\lambda_{D_j}$.
The non-distributed setting ($m = 1$) has been studied in the recent paper of Blanchard and Mücke (2017), building the root position of our results in the distributed setting, where weak and strong minimax optimal rates of convergence are established. Our aim is to extend these results to distributed learning and to derive minimax optimal rates. We again apply a fairly large class of spectral regularization methods, including the popular kernel ridge regression (KRR), L^2-boosting and spectral cut-off. Using the same notation as Blanchard and Mücke (2017), we let

$$T : f \in \mathcal{H}_K \mapsto \int f(x)K(x,\cdot)d\nu(x) \in \mathcal{H}_K$$

denote the kernel integral operator associated to K and the sampling measure ν. We denote $\bar{T} = \kappa^{-2}T$, with κ^2 the upper bound of K. Our rates of convergence are governed by a *source condition* assumption on f of the form

$$\Omega(r,R) := \{f \in \mathcal{H}_K : f = \bar{T}^r h, \ \|h\|_{\mathcal{H}_K} \leq R\}$$

for some constants $r, R > 0$ as well as by the *ill-posedness* of the problem, as measured by an assumed power decay of the eigenvalues of T with exponent $b > 1$. We show that for $s \in [0, \frac{1}{2}]$ in the sense of p-th moment ($p \geq 1$) expectation

$$\left\| \bar{T}^s (f - \bar{f}^{\lambda_n}_D) \right\|_{\mathcal{H}_K} \lesssim R \left(\frac{\sigma^2}{R^2 n} \right)^{\frac{(r+s)}{2r+1+1/b}} , \tag{2}$$

for an appropriate choice of the regularization parameter λ_n , depending on the global sample size n as well as on R and the noise variance σ^2 (but not on the number m of subsample sets). Note that $s = 0$ corresponds to the reconstruction error (i.e. $\mathcal{H}_K$-norm), and $s = \frac{1}{2}$ to the

∗. Financial support by the DFG via Research Unit 1735 "Structural Inference in Statistics" as well as SFB 1294 "Data Assimilation" is gratefully acknowledged.

prediction error (i.e., $L^2(\nu)$-norm). The symbol $\lesssim$ means that the inequality holds up to a multiplicative constant that can depend on various parameters entering in the assumptions of the result, but not on n, m, σ, nor R. An important assumption is that the inequality $q \geq r + s$ should hold, where q is the *qualification* of the regularization method, a quantity defined in the classical theory of inverse problems (see Section 2.3 for a precise definition). Basic problems are the choice of the regularization parameter on the subsamples and, most importantly, the proper choice of m, since it is well known that choosing m too large gives a suboptimal convergence rate in the limit $n \to \infty$ (see, e.g., Xu et al., 2016).

Our approach to this problem is based on a relatively classical bias-variance decomposition principle. Choosing the global regularization parameter as the optimal choice for a *single* sample of size n results in a bias estimate which is identical for all subsamples, is unchanged by averaging, and is straightforward from the single-sample analysis. On the other hand, the reduced sample size of each of the m individual subsamples causes an inflation of variance. However, since the m subsamples are independent, so are the outputs of the learning algorithm applied to each one of them; as a consequence averaging reduces the inflated variance sufficiently to get minimax optimality. We can write the variance as a sum of independent random variables, allowing to successfully apply a Rosenthal's inequality in the Hilbert space setting due to Pinelis (1994). The technical "limiting factors" in this argument give rise to the limitation on the number of subsamples m; for m larger than the allowed range, some remainder terms are no longer negligible using our proof technique, and rate optimality is not guaranteed any longer.

The outline of the paper is as follows. Section 2 contains notation and the setting. Section 3 states our main result on distributed learning. Section 4 presents numerical studies. A concluding discussion in Section 5 contains a more detailed comparison of our results with related results available in the literature. Section 6 contains the proofs of the theorems.

2. Notation, statistical model and distributed learning algorithm

In this section, we specify the mathematical background and the statistical model for (distributed) regularized learning. We have included this section for self sufficiency and reader convenience. It essentially repeats the setting in Blanchard and Mücke (2017) in summarized form.

2.1 Kernel-induced operators

We assume that the input space $\mathcal{X}$ is a standard Borel space endowed with a probability measure ν, the output space is equal to $\mathbb{R}$. We let K be a real-valued positive semidefinite kernel on $\mathcal{X} \times \mathcal{X}$ which is bounded by κ^2. The associated reproducing kernel Hilbert space will be denoted by $\mathcal{H}_K$. It is assumed that all functions $f \in \mathcal{H}_K$ are measurable and bounded in supremum norm, i.e. $\|f\|_\infty \leq \kappa \|f\|_{\mathcal{H}_K}$ for all $f \in \mathcal{H}_K$. Therefore, $\mathcal{H}_K$ is a subset of $L^2(\mathcal{X}, \nu)$, with $S : \mathcal{H}_K \longrightarrow L^2(\mathcal{X}, \nu)$ being the inclusion operator, satisfying

$\|S\| \leq \kappa$. The adjoint operator $S^* : L^2(\mathcal{X}, \nu) \longrightarrow \mathcal{H}_K$ is identified as

$$S^* g = \int_{\mathcal{X}} g(x) K_x \, \nu(dx) \, ,$$

where K_x denotes the element of $\mathcal{H}_K$ equal to the function $t \mapsto K(x,t)$. The covariance operator $T : \mathcal{H}_K \longrightarrow \mathcal{H}_K$ is given by

$$T = \int_{\mathcal{X}} \langle \cdot, K_x \rangle_{\mathcal{H}_K} K_x \, \nu(dx) \, ,$$

which can be shown to be positive self-adjoint trace class (and hence is compact). The empirical versions of these operators, corresponding formally to taking the empirical distribution $\widehat{\nu}_n = \frac{1}{n} \sum_{i=1}^n \delta_{x_i}$ in place of ν in the above formulas, are given by

$$S_{\mathbf{x}} : \mathcal{H}_K \longrightarrow \mathbb{R}^n \, , \qquad (S_{\mathbf{x}} f)_j = \langle f, K_{x_j} \rangle_{\mathcal{H}_K} \, ,$$

$$S_{\mathbf{x}}^* : \mathbb{R}^n \longrightarrow \mathcal{H}_K \, , \qquad S_{\mathbf{x}}^* \mathbf{y} = \frac{1}{n} \sum_{j=1}^n y_j K_{x_j} \, ,$$

$$T_{\mathbf{x}} := S_{\mathbf{x}}^* S_{\mathbf{x}} : \mathcal{H}_K \longrightarrow \mathcal{H}_K \, , \qquad T_{\mathbf{x}} = \frac{1}{n} \sum_{j=1}^n \langle \cdot, K_{x_j} \rangle_{\mathcal{H}_K} K_{x_j} \, .$$

We introduce the shortcut notation $\bar{T} = \kappa^{-2} T$ and $\bar{T}_{\mathbf{x}} := \kappa^{-2} T_{\mathbf{x}}$, ensuring $\|\bar{T}\| \leq 1$ and $\|\bar{T}_x\| \leq 1$, for any $x \in \mathcal{X}$. Similarly, $\bar{S} = \kappa^{-1} S$ and $\bar{S}_{\mathbf{x}_j} := \kappa^{-1} S_{\mathbf{x}_j}$, ensuring $\|\bar{S}\| \leq 1$ and $\|\bar{S}_x\| \leq 1$, for any $x \in \mathcal{X}$. The numbers μ_j are the positive eigenvalues of $\bar{T}$ satisfying $0 < \mu_{j+1} \leq \mu_j$ for all $j > 0$ and $\mu_j \searrow 0$.

2.2 Noise assumption and prior classes

In our setting of kernel learning, the sampling is assumed to be random i.i.d., where each observation point (X_i, Y_i) follows the model $Y = f_\rho(X) + \varepsilon$. For (X, Y) having distribution ρ, we assume that the conditional expectation wrt. ρ of Y given X exists and belongs to $\mathcal{H}_K$, that is, it holds for ν-almost all $x \in X$:

$$\mathbb{E}_\rho[Y | X = x] = \bar{S}_x f_\rho \, , \text{ for some } f_\rho \in \mathcal{H}_K \, . \tag{3}$$

Furthermore, we will make the following assumption on the observation noise distribution: There exists $\sigma > 0$ and $M > 0$ such that for any $l \geq 2$

$$\mathbb{E}[\, |Y - \bar{S}_X f_\rho|^l \, | \, X \,] \leq \frac{1}{2} l! \sigma^2 M^{l-2} \, , \quad \nu - \text{a.s.} \tag{4}$$

To derive nontrivial rates of convergence, we concentrate our attention on specific subsets (also called *models*) of the class of probability measures. If $\mathcal{P}$ denotes the set of all probability distributions on $\mathcal{X}$, we define classes of sampling distributions by introducing a decay

condition on the *effective dimension* $\mathcal{N}(\lambda)$, being a measure for the complexity of $\mathcal{H}_K$ with respect to the marginal distribution ν: For $\lambda \in (0,1]$ we set

$$\mathcal{N}(\lambda) = \mathrm{Trace}[\ (\bar{T}+\lambda)^{-1}\bar{T}\ ]\ . \tag{5}$$

Note that $\mathcal{N}(\lambda) \leq 1$. For any $b > 1$ we introduce

$$\mathcal{P}^{<}(b) := \{\nu \in \mathcal{P}:\ \mathcal{N}(\lambda) \leq C_b(\kappa^2\lambda)^{-\frac{1}{b}}\ \}\ . \tag{6}$$

In De Vito and Caponnetto, 2006, Proposition 3, it is shown that such a condition is implied by polynomially decreasing eigenvalues of $\bar{T}$. More precisely, if the eigenvalues μ_i satisfy $\mu_j \leq \beta/j^b\ \ \forall j \geq 1$ or $b > 1$ and $\beta > 0$, then

$$\mathcal{N}(\lambda) \leq \frac{\beta^{\frac{1}{b}}b}{b-1}(\kappa^2\lambda)^{-\frac{1}{b}}\ .$$

For a subset $\Omega \subseteq \mathcal{H}_K$, we let $\mathcal{K}(\Omega)$ be the set of regular conditional probability distributions $\rho(\cdot|\cdot)$ on $\mathcal{B}(\mathbb{R})\times\mathcal{X}$ such that (3) and (4) hold for some $f_\rho \in \Omega$. We will focus on a *Hölder-type source condition*, i.e. given $r > 0, R > 0$ and $\nu \in \mathcal{P}$, we define

$$\Omega(r,R) := \{f \in \mathcal{H}_K : f = \bar{T}^r h,\ \|h\|_{\mathcal{H}_K} \leq R\}. \tag{7}$$

Then the class of models which we will consider will be defined as

$$\mathcal{M}(r,R,\mathcal{P}')\ :=\ \{\ \rho(dx,dy) = \rho(dy|x)\nu(dx)\ :\ \rho(\cdot|\cdot) \in \mathcal{K}(\Omega(r,R)),\ \nu \in \mathcal{P}'\ \}\ , \tag{8}$$

with $\mathcal{P}' = \mathcal{P}^{<}(b)$. As a consequence, the class of models depends not only on the smoothness properties of the solution (reflected in the parameters $R > 0,\ r > 0$), but also essentially on spectral properties of $\bar{T}$, reflected in $\mathcal{N}(\lambda)$.

2.3 Spectral regularization

In this subsection, we introduce the class of linear regularization methods based on spectral theory for self-adjoint linear operators. These are standard methods for finding stable solutions for ill-posed inverse problems. Originally, these methods were developed in the deterministic context (see Engl et al., 2000). Later on, they have been applied to probabilistic problems in machine learning (see, e.g., Bauer et al., 2007; De Vito and Caponnetto, 2006; Dicker et al., 2017 or Blanchard and Mücke, 2017).

Definition 1 (Regularization function) *Let $g : (0,1]\times[0,1] \longrightarrow \mathbb{R}$ be a function and write $g_\lambda = g(\lambda,\cdot)$. The family $\{g_\lambda\}_\lambda$ is called regularization function, if the following conditions hold:*

(i) There exists a constant $D' < \infty$ such that for any $0 < \lambda \leq 1$

$$\sup_{0\leq t\leq 1}|tg_\lambda(t)| \leq D'\ . \tag{9}$$

(ii) There exists a constant $E < \infty$ such that for any $0 < \lambda \leq 1$

$$\sup_{0\leq t\leq 1}|g_\lambda(t)| \leq \frac{E}{\lambda}\ . \tag{10}$$

(iii) Defining the residual *$r_\lambda(t) := 1 - g_\lambda(t)t$, there exists a constant $\gamma_0 < \infty$ such that for any $0 < \lambda \leq 1$*

$$\sup_{0\leq t\leq 1}|r_\lambda(t)| \leq \gamma_0\ .$$

It has been shown in e.g. Gerfo et al. (2008), Dicker et al. (2017), Blanchard and Mücke (2017) that attainable learning rates are essentially linked with the qualification of the regularization $\{g_\lambda\}_\lambda$, being the maximal q such that for any $0 < \lambda \leq 1$

$$\sup_{0\leq t\leq 1}|r_\lambda(t)|t^q \leq \gamma_q\lambda^q. \tag{11}$$

for some constant $\gamma_q > 0$. Note that by (iii) , using interpolation, we have validity of (11) also for any $q' \in [0,q]$ with constant $\gamma_{q'} = \gamma_0^{1-\frac{q'}{q}}\gamma_q^{\frac{q'}{q}}$.

The most popular examples include:

Example 1 *(Tikhonov Regularization, Kernel Ridge Regression) The choice $g_\lambda(t) = \frac{1}{\lambda+t}$ corresponds to Tikhonov regularization. In this case we have $D' = E = \gamma_0 = 1$. The qualification of this method is $q = 1$ with $\gamma_q = 1$.*

Example 2 *(Landweber Iteration, gradient descent) The Landweber Iteration (gradient descent algorithm with constant stepsize) is defined by*

$$g_k(t) = \sum_{j=0}^{k-1}(1-t)^j\ \ \text{with } k = 1/\lambda \in \mathbb{N}\ .$$

We have $D' = E = \gamma_0 = 1$. The qualification q of this algorithm can be arbitrary with $\gamma_q = 1$ if $0 < q \leq 1$ and $\gamma_q = q^q$ if $q > 1$.

Example 3 (ν- method) *The $\nu-$ method belongs to the class of so called semi-iterative regularization methods. This method has finite qualification $q = \nu$ with γ_q a positive constant. Moreover, $D = 1$ and $E = 2$. The filter is given by $g_k(t) = p_k(t)$, a polynomial of degree $k-1$, with regularization parameter $\lambda \sim k^{-2}$, which makes this method much faster as e.g. gradient descent.*

2.4 Distributed learning algorithm

We let $D = \{(x_j, y_j)\}_{j=1}^n \subset \mathcal{X} \times \mathcal{Y}$ be the dataset, which we partition into m disjoint subsamples[1] $D_1, \ldots, D_m$, each having size $\frac{n}{m}$. Denote the jth data subsample by $(\mathbf{x}_j, \mathbf{y}_j) \in (\mathcal{X} \times \mathbb{R})^{\frac{n}{m}}$. On each subsample we compute a local estimator for a suitable a-priori parameter choice $\lambda = \lambda_n$ according to

$$f_{D_j}^{\lambda_n} := g_{\lambda_n}(\bar{T}_{\mathbf{x}_j}) \bar{S}_{\mathbf{x}_j}^* \mathbf{y}_j \ . \tag{12}$$

By f_D^λ we will denote the estimator using the whole sample $m = 1$. The final estimator is given by simple averaging of the local ones:

$$\bar{f}_D^\lambda := \frac{1}{m} \sum_{j=1}^m f_{D_j}^\lambda \ . \tag{13}$$

3. Main results

This section presents our main results. Theorem 3 and Theorem 4 contain separate estimates on the approximation error and the sample error and lead to Corollary 5 which gives an upper bound for the error $\left\| \bar{T}^s (f_\rho - \bar{f}_D^\lambda) \right\|_{\mathcal{H}_K}$ and presents an upper rate of convergence for the sequence of distributed learning algorithms.

For the sake of the reader we recall Theorem 6, which was already shown in Blanchard and Mücke (2017), presenting the minimax optimal rate for the single machine problem. This yields an estimate on the difference between the single machine and the distributed learning algorithm in Corollary 7.

We want to track the precise behavior of these rates not only for what concerns the exponent in the number of examples n, but also in terms of their scaling (multiplicative constant) as a function of some important parameters (namely the noise variance σ^2 and the complexity radius R in the source condition, see Remark 9 below). For this reason, we introduce a notion of a family of rates over a family of models. More precisely, we consider an indexed family $(\mathcal{M}_\theta)_{\theta \in \Theta}$, where for all $\theta \in \Theta$, $\mathcal{M}_\theta$ is a class of Borel probability distributions on $\mathcal{X} \times \mathbb{R}$ satisfying the basic general assumptions (3) and (4). We consider rates of convergence in the sense of the p-th moments of the estimation error, where $1 \leq p < \infty$ is a fixed real number.

1. For the sake of simplicity, throughout this paper we assume that n is divisible by m. This could always be achieved by disregarding some data; alternatively, it is straightforward to show that admitting one smaller block in the partition does not affect the asymptotic results of this paper. We shall not try to discuss this point in greater detail. In particular, we shall not analyze in which general framework our simple averages could be replaced by weighted averages.

As already mentioned in the introduction, our proofs are based on a classical bias-variance decomposition as follows: Introducing

$$\tilde{f}_D^\lambda = \frac{1}{m} \sum_{j=1}^m g_\lambda(\bar{T}_{\mathbf{x}_j}) \bar{T}_{\mathbf{x}_j} f_\rho \ , \tag{14}$$

we write

$$\begin{aligned} \bar{T}^s (f_\rho - \bar{f}_D^\lambda) &= \bar{T}^s (f_\rho - \tilde{f}_D^\lambda) + \bar{T}^s (\tilde{f}_D^\lambda - \bar{f}_D^\lambda) \\ &= \frac{1}{m} \sum_{j=1}^m \bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j}) f_\rho + \frac{1}{m} \sum_{j=1}^m \bar{T}^s g_\lambda(\bar{T}_{\mathbf{x}_j}) (\bar{T}_{\mathbf{x}_j} f_\rho - \bar{S}_{\mathbf{x}_j}^* \mathbf{y}_j) \ . \end{aligned} \tag{15}$$

In all the forthcoming results in this section, we assume:

Assumption 2 *Let $s \in [0, \frac{1}{2}]$, $p \geq 1$ and consider the model $\mathcal{M}_{\sigma,M,R} := \mathcal{M}(r, R, \mathcal{P}^<(b))$ where $r > 0$ and $b > 1$ are fixed, and $\theta = (R, M, \sigma)$ varies in $\Theta = \mathbb{R}_+^3$. Given a sample $D \subset (\mathcal{X} \times \mathbb{R})$ of size n, define $\bar{f}_D^{\lambda_n}$, $f_D^{\lambda_n}$ as in Section 2.4 and $\tilde{f}_D^{\lambda_n}$ as in* (14)*, using a regularization function of qualification $q \geq r + s$, with parameter sequence*

$$\lambda_n := \lambda_{n,(\sigma,R)} := \min \left(\left(\frac{\sigma^2}{R^2 n} \right)^{\frac{b}{2br+b+1}}, 1 \right) \ , \tag{16}$$

independent on M. Define the sequence

$$a_n := a_{n,(\sigma,R)} := R \left(\frac{\sigma^2}{R^2 n} \right)^{\frac{b(r+s)}{2br+b+1}} \ . \tag{17}$$

We recall that we shall always assume that n is a multiple of m. With these preparations, our main results are:

Theorem 3 (Approximation error) *Under Assumption 2, we have: If the number m_n of subsample sets satisfies*

$$m_n \leq n^\alpha \ , \quad \alpha < \frac{2b \min\{r, 1\}}{2br + b + 1} \ , \tag{18}$$

then

$$\sup_{(\sigma,M,R) \in \mathbb{R}_+^3} \limsup_{n \to \infty} \sup_{\rho \in \mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}} \left\| \bar{T}^s (f_\rho - \tilde{f}_D^{\lambda_n}) \right\|_{\mathcal{H}_K}^p \right]^{\frac{1}{p}}}{a_n} < \infty \ .$$

Theorem 4 (Sample Error) *Under Assumption 2, we have: If the number m_n of subsample sets satisfies*

$$m_n \leq n^\alpha \,, \quad \alpha < \frac{2br}{2br+b+1} \,, \tag{19}$$

Then

$$\sup_{(\sigma,M,R)\in\mathbb{R}^3_+} \limsup_{n\to\infty} \sup_{\rho\in\mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}} \left\|\bar{T}^s(\tilde{f}_D^{\lambda_n} - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}}{a_n} < \infty \,.$$

And, as consequence (by (15) and applying the triangle inequality for the L^p-norm):

Corollary 5 *Under Assumption 2, we have: If the number m_n of subsample sets satisfies*

$$m_n \leq n^\alpha \,, \quad \alpha < \frac{2b\min\{r,1\}}{2br+b+1} \,, \tag{20}$$

then the sequence (17) *is an upper rate of convergence in L^p for all $p>0$, for the interpolation norm of parameter s, for the sequence of estimated solutions $(\bar{f}_D^{\lambda_{n,(\sigma,R)}})$ over the family of models $(\mathcal{M}_{\sigma,M,R})_{(\sigma,M,R)\in\mathbb{R}^3_+}$, i.e.*

$$\sup_{(\sigma,M,R)\in\mathbb{R}^3_+} \limsup_{n\to\infty} \sup_{\rho\in\mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}} \left\|\bar{T}^s(f_\rho - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}}{a_n} < \infty \,.$$

Theorem 6 (Blanchard and Mücke, 2017) *The sequence* (17) *is an upper rate of convergence in L^p for all $p>0$, for the interpolation norm of parameter s, for the sequence of estimated solutions $(f_D^{\lambda_{n,(\sigma,R)}})$ over the family of models $(\mathcal{M}_{\sigma,M,R})_{(\sigma,M,R)\in\mathbb{R}^3_+}$, i.e.*

$$\sup_{(\sigma,M,R)\in\mathbb{R}^3_+} \limsup_{n\to\infty} \sup_{\rho\in\mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}} \left\|\bar{T}^s(f_\rho - f_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}}{a_n} < \infty \,.$$

Combining Corollary 5 with Theorem 6 by applying the triangle inequality immediately yields:

Corollary 7 *If the number m_n of subsample sets satisfies*

$$m_n \leq n^\alpha \,, \quad \alpha < \frac{2b\min\{r,1\}}{2br+b+1} \,, \tag{21}$$

then

$$\sup_{(\sigma,M,R)\in\mathbb{R}^3_+} \limsup_{n\to\infty} \sup_{\rho\in\mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}} \left\|\bar{T}^s(f_D^{\lambda_n} - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}}{a_n} < \infty \,.$$

Remark 8 *Our results in the distributed setting slightly differ from those obtained in Theorem 6 from Blanchard and Mücke (2017) in two several respects:*

- *While in the single machine approach, rates of convergence are obtained for any $p>0$, the proofs in Section 6 only hold for $p \geq 1$ due to loss of subadditivity of p-th moments for $0<p<1$.*
- *While the upper upper rates of convergence in Blanchard and Mücke (2017) are derived over classes of marginals ν induced by assuming a decay condition for the eigenvalues of $\bar{T}$, we somewhat enlarge this class by assuming a decay condition for $\mathcal{N}(\lambda)$ in* (6)*. Theorem 6 also holds under this weaker condition. Note that it is an open problem if lower rates of convergence can also be obtained by weakening the condition for eigenvalue decay.*

Remark 9 (Signal-to-noise-ratio) *Our results show that the choice of the regularization parameter λ_n in* (16) *and thus the rate of convergence a_n in* (17) *highly depend on the signal-to-noise-ratio $\frac{\sigma^2}{R^2}$, a quantity which naturally appears in the theory of regularization of ill-posed inverse problems. As a general rule, the degree of regularization should increase with the level of noise in the data, i.e., the importance of the priors should increase as the model fit decreases. Our theoretical results precisely show this behavior.*

4. Numerical studies

In this section we numerically study the error in $\mathcal{H}_K$-norm, corresponding to $s=0$ in Corollary 5 (in expectation with $p=2$) both in the single machine and distributed learning setting. Our main interest is to study the upper bound for our theoretical exponent α, parametrizing the size of subsamples in terms of the total sample size, $m=n^\alpha$, in different smoothness regimes. In addition we shall demonstrate in which way parallelization serves as a form of regularization.

More specifically, we let $\mathcal{H}_K = H_0^1[0,1]$ be the Sobolev space consisting of absolutely continuous functions f on $[0,1]$ with weak derivative of order 1 in $L^2[0,1]$, with boundary condition $f(0)=f(1)=0$. The reproducing kernel is given by $K(x,t)=x\wedge t - xt$. For all experiments in this section, we simulate data from the regression model

$$Y_i = f_\rho(X_i) + \epsilon_i \,, \qquad i=1,\dots,n \,,$$

where the input variables $X_i \sim Unif[0,1]$ are uniformly distributed and the noise variables $\varepsilon_i \sim N(0,\sigma^2)$ are normally distributed with standard deviation $\sigma=0.005$. We choose the target function f_ρ according to two different cases, namely $r<1$ (*low smoothness regime*) and $r=\infty$ (*high smoothness regime*). To accurately determine the degree of smoothness $r>0$, we apply Proposition 10 below by explicitly calculating the Fourier coefficients $(\langle f_\rho, e_j\rangle_{\mathcal{H}_K})_{j\in\mathbb{N}}$, where $e_j(x)=\frac{\sqrt{2}}{\pi j}\cos(\pi j x)$, for $j\in\mathbb{N}^*$, forms an ONB of $\mathcal{H}_K$. Recall that the rate of eigenvalue decay is explicitly given by $b=2$, meaning that we have full control over all parameters in (21). We need the following characterization:

Proposition 10 (Engl et al., 2000, Prop. 3.13) *Let $\mathcal{H}_K, \mathcal{H}_2$ be separable Hilbert spaces and $S : \mathcal{H}_K \longrightarrow \mathcal{H}_2$ be a compact linear operator with singular system* [2] $\{\sigma_j, \varphi_j, \psi_j\}$. *Denoting by $S^\dagger$ the generalized inverse*[3] *of S, one has for any $r > 0$ and $g \in \mathcal{H}_2$:*

g is in the domain of $S^\dagger$ and $S^\dagger g \in \mathrm{Im}((S^*S)^r)$ if and only if

$$\sum_{j=0}^{\infty} \frac{|\langle g, \psi_j \rangle_{\mathcal{H}_2}|^2}{\sigma_j^{2+4r}} < \infty .$$

In our case, $\mathcal{H}_K$ is as above, $\mathcal{H}_2$ is $L^2([0,1])$ with Lebesgue measure and $S : H_0^1[0,1] \to L^2([0,1])$ is the inclusion. Since $H_0^1[0,1]$ is dense in $L^2([0,1])$, we know that $(\mathrm{Im}(S))^\perp$ is trivial, giving $SS^\dagger = id$ on $\mathrm{Im}(S)$. Furthermore, $\varphi_j = e_j$ is a normalized eigenbasis of $T = S^*S$ with eigenvalues $\sigma_j^2 = (\pi j)^{-2}$. With $\psi_j = \frac{S\varphi_j}{\|S\varphi_j\|_{L^2}}$ we obtain for $f \in H_0^1[0,1]$

$$\langle Sf, \psi_j \rangle_{L^2} = \left\langle Sf, \frac{Se_j}{\|Se_j\|} \right\rangle_{L^2} = \left\langle f, \frac{S^*Se_j}{\|Se_j\|} \right\rangle_{H_0^1} = \sigma_j \langle f, e_j \rangle_{H_0^1}.$$

Thus, applying Proposition 10 gives

Corollary 11 *For S and $T = S^*S$ defined in Section 2, we have for any $r > 0$: $f \in \mathrm{Im}(T^r)$ if and only if*

$$\sum_{j=1}^{\infty} j^{4r} |\langle f, e_j \rangle_{L^2}|^2 < \infty .$$

Thus, as expected, abstract smoothness measured by the parameter r in the source condition corresponds in this special case to decay of the classical Fourier coefficients which by the classical theory of Fourier series measures smoothness of the periodic continuation of $f \in L^2([0,1])$ to the real line.

4.1 Low smoothness regime

We choose $f_\rho(x) = \frac{1}{2}x(1-x)$ which clearly belongs to $\mathcal{H}_K$. A straightforward calculation gives the Fourier coefficient $\langle f_\rho, e_j \rangle = -2(\pi j)^{-2}$ for j odd (vanishing for j even). Thus, by the above criterion, f_ρ satisfies the source condition $f_\rho \in Ran(\bar{T}^r)$ precisely for $0 < r < 0.75$. (Observe that although f_ρ is smooth on $[0,1]$, its periodic continuation on the real line is not, hence the low smoothness regime.) According to Theorem 6, the worst case rate in the single machine problem is given by $n^{-\gamma}$, with $\gamma = 0.25$. Regularization is done using the $\nu-$ method (see Example 3), with qualification $q = \nu = 1$. Recall that the stopping index

2. i.e., the φ_j are the normalized eigenfunctions of S^*S with eigenvalues σ_j^2 and $\psi_j = S\varphi_j/\|S\varphi_j\|$; thus $S = \sum \sigma_j \langle \varphi_j, \cdot \rangle \psi_j$.
3. the unique unbounded linear operator with domain $\mathrm{Im}(S) \oplus (\mathrm{Im}(S))^\perp$ in $\mathcal{H}_2$ vanishing on $(\mathrm{Im}(S))^\perp$ and satisfying $SS^\dagger = 1$ on $\mathrm{Im}(S)$, with range orthogonal to the null space $N(S)$.

k_{stop} serves as the regularization parameter λ, where $k_{stop} \sim \lambda^{-2}$. We consider sample sizes from $500, \ldots, 9000$. In the model selection step, we estimate the performance of different models and choose the *oracle stopping time* $\hat{k}_{oracle}$ by minimizing the reconstruction error:

$$\hat{k}_{oracle} = \arg\min_k \left(\frac{1}{M} \sum_{j=1}^{M} \left\| f_\rho - \hat{f}_j^k \right\|_{\mathcal{H}_K}^2 \right)^{\frac{1}{2}}$$

over $M = 30$ runs.

In the model assessment step, we partition the dataset into $m \sim n^\alpha$ subsamples, for any $\alpha \in \{0, 0.05, 0.1, \ldots, 0.85\}$. On each subsample we regularize using the oracle stopping time $\hat{k}_{oracle}$ (determined by using the whole sample). Corresponding to Corollary 5, the accuracy should be comparable to the one using the whole sample as long as $\alpha < 0.5$. In Figure 1 (left panel) we plot the reconstruction error $\|\bar{f}^{\hat{k}} - f_\rho\|_{\mathcal{H}_K}$ versus the ratio $\alpha = \log(m)/\log(n)$ for different sample sizes. We execute each simulation $M = 30$ times. The plot supports our theoretical finding. The right panel shows the reconstruction error versus the total number of samples using different partitions of the data. The black curve ($\alpha = 0$) corresponds to the baseline error ($m = 0$, no partition of data). Error curves below a threshold $\alpha < 0.6$ are roughly comparable, whereas curves above this threshold show a gap in performances.

In another experiment we study the performances in case of (very) different regularization: Only partitioning the data (no regularization), underregularization (higher stopping index) and overregularization (lower stopping index). The outcome of this experiment amplifies the regularization effect of parallelizing. Figure 2 shows the main point: Overregularization is always hopeless, underregularization is better. In the extreme case of (almost) no regularization, there is a sharp minimum in the reconstruction error which is only slightly larger than the minimax optimal value for the oracle regularization parameter and which is achieved at an attractively large degree of parallelization. Qualitatively, this agrees very well with the intuitive notion that parallelizing serves as regularization.

We emphasize that numerical results seem to indicate that parallelization is possible to a slightly larger degree than indicated by our theoretical estimate. A similar result was reported in the paper Zhang et al. (2013), which also treats the low smoothness case.

4.2 High smoothness regime

We choose $f_\rho(x) = \frac{1}{2\pi}\sin(2\pi x)$, which corresponds to just one non-vanishing Fourier coefficient and by our criterion Corollary 11 has $r = \infty$. In view of our main Corollary 5 this requires a regularization method with higher qualification; we take the *Gradient Descent* method (see Example 2).

The appearance of the term $2b\min\{1, r\}$ in our theoretical result 5 gives a predicted value $\alpha = 0$ (and would imply that parallelization is strictly forbidden for infinite smoothness). More specifically, the left panel in Figure 3 shows the absence of any plateau for the reconstruction error as a function of α. This corresponds to the right panel showing that

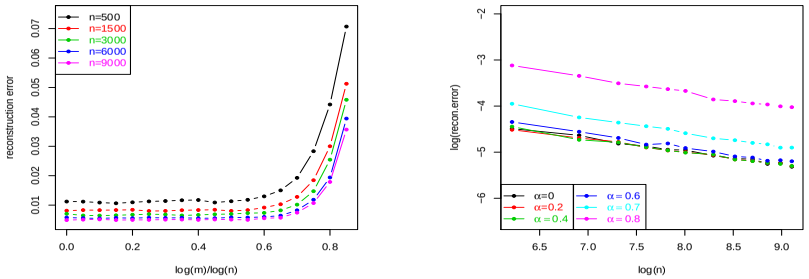


Figure 1: The reconstruction error $\|\bar{f}_D^{k_{oracle}} - f_\rho\|_{\mathcal{H}_K}$ in the low smoothness case. Left plot: Reconstruction error curves for various (but fixed) total sample sizes, as a function of the number m of subsamples. Right plot: Reconstruction error curves for various subsample number scalings $m = n^\alpha$, as a function of the sample size (on log-scale).

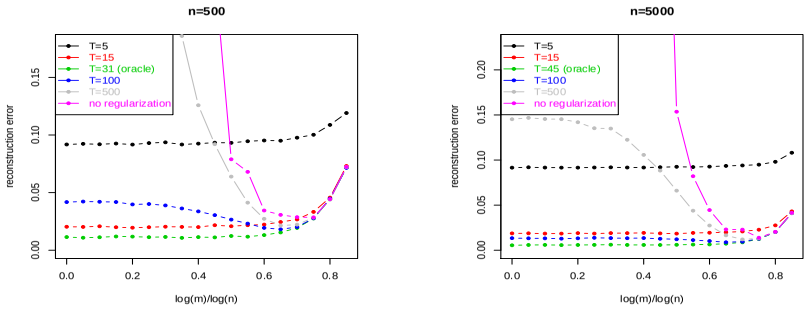


Figure 2: The reconstruction error $\|\bar{f}_D^{\lambda} - f_\rho\|_{\mathcal{H}_K}$ in the low smoothness case. Left plot: Error curves for different stopping times for $n = 500$, as a function of the number m of subsamples. Right plot: Error curves for different stopping times for $n = 5000$, as a function of the number m of subsamples.

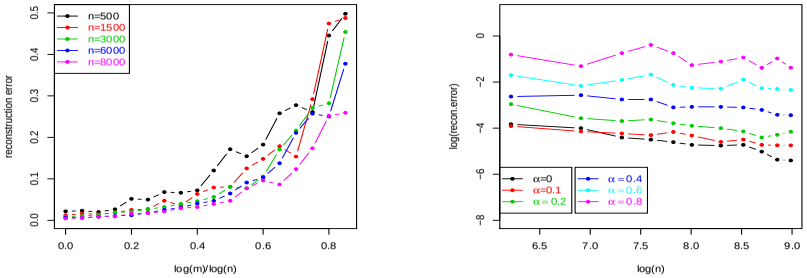


Figure 3: The reconstruction error $\|\bar{f}_D^{\lambda_{oracle}} - f_\rho\|_{\mathcal{H}_K}$ in the high smoothness case. Left plot: Reconstruction error curves for various (but fixed) sample sizes as a function of the number m of subsamples. Right plot: Reconstruction error curves for various subsample number scalings $m = n^\alpha$, as a function of the sample size (on log-scale).

no group of values of α performs roughly equivalently, meaning that we do not have any optimality guarantees.

Plotting different values of regularization in Figure 4 we again identify overregularization as hopeless, while severe underregularization exhibits a sharp minimum in the reconstruction error. But its value at roughly 0.25 is much less attractive compared to the case of low smoothness where the error is an order of magnitude less.

5. Discussion

Minimax Optimality: We have shown that for a large class of spectral regularization methods the error of the distributed algorithm $\|\bar{T}^s(\bar{f}_D^{\lambda_n} - f_\rho)\|_{\mathcal{H}_K}$ satisfies the same upper bound as the error $\left\|\bar{T}^s(f_D^{\lambda_n} - f_\rho)\right\|_{\mathcal{H}_K}$ for the single machine problem, if the regularization parameter λ_n is chosen according to (16), provided the number of subsamples grows sufficiently slowly with the sample size n. Since, the rates for the latter are minimax optimal (Blanchard and Mücke, 2017), our rates in Corollary 5 are minimax optimal also.

Comparison with other results: Zhang et al. (2013) derive minimax-optimal rates in three settings: finite rank kernels, sub-Gaussian decay of eigenvalues of the kernel and polynomial decay, provided m satisfies a certain upper bound, depending on the rate of decay of the eigenvalues under two crucial assumptions on the eigenfunctions of the integral operator associated to the kernel: For any $j \in \mathbb{N}$

$$\mathbb{E}[\phi_j(X)^{2k}] \leq \rho^{2k} \,, \tag{22}$$

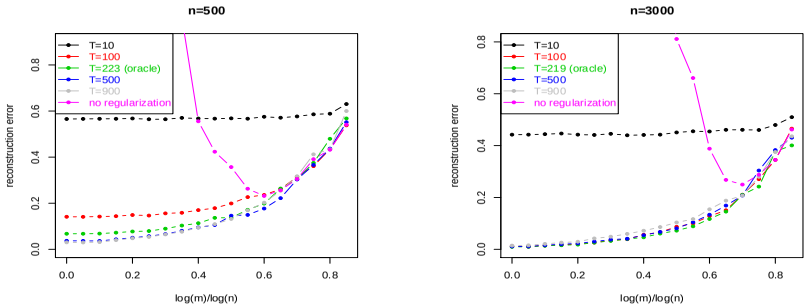


Figure 4: The reconstruction error $\|\bar{f}_D^{\lambda} - f_\rho\|$ in the high smoothness case. Left plot: Error curves for different stopping times for $n = 500$ samples, as a function of the number of subsamples. Right plot: Error curves for different stopping times for $n = 5000$ samples, as a function of the number of subsamples.

for some $k \geq 2$ and $\rho < \infty$ or even stronger, it is assumed that the eigenfunctions are uniformly bounded, i.e.

$$\sup_{x \in \mathcal{X}} |\phi_j(x)| \leq \rho \,, \tag{23}$$

or any $j \in \mathbb{N}$ and some $\rho < \infty$. We shall describe in more detail the case of polynomially decaying eigenvalues, which corresponds to our setting. Assuming eigenvalue decay $\mu_j \lesssim j^{-b}$ with $b > 1$, the authors choose a regularization parameter $\lambda_n = n^{-\frac{b}{b+1}}$ and

$$m \lesssim \left(\frac{n^{\frac{b(k-4)-k}{b+1}}}{\rho^{4k} \log^k(n)} \right)^{\frac{1}{k-2}} \,.$$

leading to an error in L^2-norm

$$\mathbb{E}[\|\bar{f}_D^{\lambda_n} - f_\rho\|_{L^2}^2] \lesssim n^{-\frac{b}{b+1}} \,,$$

being minimax optimal. Note that this choice of λ_n and the resulting rate correspond to our case $r = 0$, i.e., no smoothness of f_ρ is assumed (just that f_ρ belongs to the RKHS).

For $k < 4$ the bound becomes less meaningful (compared to the case where $k \geq 4$) since $m \to 0$ as $n \to \infty$ in this case (for any sort of eigenvalue decay). On the other hand, if k and b might be taken arbitrarily large, corresponding to almost bounded eigenfunctions and arbitrarily large polynomial decay of eigenvalues, m might be chosen proportional to $n^{1-\epsilon}$, for any $\epsilon > 0$. As might be expected, replacing the L^{2k} bound on the eigenfunctions by a bound in L^∞, gives an upper bound on m which simply is the limit for $k \to \infty$ in the

bound given above, namely

$$m \lesssim \frac{n^{\frac{b-1}{b+1}}}{\rho^4 \log n} \,,$$

which for large b behaves as above. Granted bounds on the eigenfunctions in L^{2k} for (very) large k, this is a strong result. While the decay rate of the eigenvalues can be determined by the smoothness of K (see, e.g., Ferreira and Menegatto, 2009 and references therein), it is a widely open question which general properties of the kernel imply estimates as in (22) and (23) on the eigenfunctions.

Zhou (2002) even gives a counterexample and presents a C^∞ Mercer kernel on $[0, 1]$ where the eigenfunctions of the corresponding integral operator are *not* uniformly bounded. Thus, smoothness of the kernel is not a sufficient condition for (23) to hold.

Moreover, we point out that the upper bound (22) on the eigenfunctions (and thus the upper bound for m in Zhang et al., 2013) depends on the unknown marginal distribution ν. Only the strongest assumption, a bound in sup-norm (23), does not depend on ν. Concerning this point, our approach is "agnostic".

As already mentioned in the Introduction, these bounds on the eigenfunctions have been eliminated by Lin et al. (2017), for KRR, imposing polynomial decay of eigenvalues as above. This is very similar to our approach. As a general rule, our bounds on m and the bounds obtained by Lin et al. (2017) are worse than the bounds of Zhang et al. (2013) for eigenfunctions in (or close to) L^∞, but in the complementary case where nothing is known on the eigenfunctions m still can be chosen as an increasing function of n, namely $m = n^\alpha$. More precisely, choosing λ_n as in (16), Lin et al. (2017) derive as an upper bound

$$m \lesssim n^\alpha \,, \qquad \alpha = \frac{2br}{2br + b + 1} \,,$$

with r being the smoothness parameter arising in the source condition. We recall here that due to our assumption $q \geq r + s$, the smoothness parameter r is restricted to the interval $(0, \frac{1}{2}]$ for KRR ($q = 1$) and L^2-risk ($s = \frac{1}{2}$).

Our results (which hold for a general class of spectral regularization methods) are in some ways comparable to those of Lin et al. (2017). Specialized to KRR, our estimates for the exponent α in $m = O(n^\alpha)$ coincide with the result of Lin et al. (2017). Furthermore, we emphasize that Zhang et al. (2013) and Lin et al. (2017) estimate the DL-error only for $s = 1/2$ in our notation (corresponding to $L^2(\nu)-$norm), while our result holds for all values of $s \in [0, 1/2]$ which smoothly interpolates between $L^2(\nu)$-norm and RKHS-norm and, in addition, for all values of $p \in [1, \infty)$. Thus, our results also apply to the case of non-parametric inverse regression, where one is particularly interested in the reconstruction error, i.e. $\mathcal{H}_K$-norm (see, e.g., Blanchard and Mücke, 2017). Additionally, we precisely analyze the dependence of the noise variance σ^2 and the complexity radius R in the source condition.

Concerning general strategy, while Lin et al. (2017) use a novel second order decomposition in an essential way, our approach is more classical. We clearly distinguish between estimat-

ing the approximation error and the sample error. The bias using a subsample should be of the same order as when using the whole sample, whereas the estimation error is higher on each subsample, but gets reduced by averaging by writing the variance as a sum of i.i.d. random variables (which allows to use Rosenthal's inequality).

Finally, we want to mention the recent works of Lin and Zhou (2018) and Guo et al. (2017), which were worked out indepently from our work. Guo et al. (2017) also treat general spectral regularization methods (going beyond kernel ridge) and obtain essentially the same results, but with error bounds only in L^2-norm, excluding inverse learning problems. Lin and Zhou (2018) investigate distributed learning on the example of gradient descent algorithms, which have infinite qualification and allow larger smoothness of the regression function. They are able to improve the upper bound for the number of local machines to

$$m \lesssim \frac{n^\alpha}{\log^5(n)+1}\,, \qquad \alpha < \frac{br}{2br+b+1},$$

which is larger in the case $r > 2$. In the intermediate case $1 < r < 2$, our bound in (20) is still better. An interesting feature is the fact that it is possible to allow more local machines by using additional unlabeled data. This indicates that finding the upper bound for the number of machines in the high smoothness regime is still an open problem.

Adaptivity: It is clear from the theoretical results that both the regularization parameter λ and the allowed cardinality of subsamples m depend on the parameters r and b, which in general are unknown. Thus, an adaptive approach to both parameters b and r for choosing λ and m is of interest. To the best of our knowledge, there are yet no rigorous results on adaptivity in this more general sense. Progress in this field may well be crucial in finally assessing the relative merits of the distributed learning approach as compared with alternative strategies to effectively deal with large data sets.

We sketch an alternative naive approach to adaptivity, based on hold-out in the direct case, where we consider each $f \in \mathcal{H}_K$ also as a function in $L^2(\mathcal{X},\nu)$. We split the data $\mathbf{z} \in (\mathcal{X}\times\mathcal{Y})^n$ into a training and validation part $\mathbf{z} = (\mathbf{z}^t, \mathbf{z}^v)$ of cardinality m_t, m_v. We further subdivide $\mathbf{z}^t$ into m_k subsamples, roughly of size m_t/m_k, where $m_k \le m_t$, $k = 1, 2, \ldots$ is some strictly decreasing sequence. For each k and each subsample $\mathbf{z}_j$, $1 \le j \le m_k$, we define the estimators $\hat{f}^\lambda_{\mathbf{z}_j}$ as in (12) and their average

$$\bar{f}^\lambda_{k,\mathbf{z}^t} := \frac{1}{m_k}\sum_{j=1}^{m_k} \hat{f}^\lambda_{\mathbf{z}_j}\,. \tag{24}$$

Here, λ varies in some sufficiently fine lattice Λ. Then evaluation on $\mathbf{z}^v$ gives the associated empirical L^2- error

$$\mathrm{Err}^\lambda_k(\mathbf{z}^v) := \frac{1}{m_v}\sum_{i=1}^{m_v}(y^v_i - \bar{f}^\lambda_{k,\mathbf{z}^t}(x^v_i))^2\,, \quad \mathbf{z}^v = (\mathbf{y}^v, \mathbf{x}^v)\,, \quad \mathbf{y}^v = (y^v_1,\ldots,y^v_{m_v})\,, \tag{25}$$

leading us to define

$$\hat{\lambda}_k := \mathrm{argmin}_{\lambda\in\Lambda}\mathrm{Err}^\lambda_k(\mathbf{z}^v)\,, \qquad \mathrm{Err}(k) := \mathrm{Err}^{\hat{\lambda}_k}_k(\mathbf{z}^v). \tag{26}$$

Then, an appropriate stopping criterion for k might be to stop at

$$k^* := \min\{k \ge 3\ :\ \Delta(k) \le \delta \inf_{2\le j<k}\Delta(j)\}\,, \qquad \Delta(j) := |\mathrm{Err}(j) - \mathrm{Err}(j-1)|\,, \tag{27}$$

for some $\delta < 1$ (which might require tuning). The corresponding regularization parameter is $\hat{\lambda} = \hat{\lambda}_{k^*}$, given by (26). At least intuitively, it is then reasonable to define a purely data driven estimator as

$$\widehat{f}_n := \bar{f}^{\hat{\lambda}}_{k^*,\mathbf{z}^t}\,. \tag{28}$$

Note that the training data $\mathbf{z}^t$ enter the definition of $\widehat{f}_n$ via the explicit formula (24) encoding our kernel based approach, while $\mathbf{z}^v$ serves to determine $(k^*, \hat{\lambda}^*)$ via minimization of the empirical L^2-error and a criterion, which tells one to stop where $\mathrm{Err}(j)$ does not appreciably improve anymore. It is open if such a procedure achieves optimal rates, and we leave this for future research.

6. Proofs

For ease of reading we make use of the following conventions:

- for a (bounded) linear operator A, $\|A\|$ denotes the operator norm;
- we are interested in a precise dependence of multiplicative constants on the parameters σ, M, R, m, n and η. (To be clear about the role of the latter quantity: the proofs rely on high-probability statements on deviations, typically holding with high probability $1-\eta$.)
- the dependence of multiplicative constants on various other parameters, including the kernel parameter κ, the interpolating parameter $s \in [0, \frac{1}{2}]$, the parameters arising from the regularization method, $b > 1$, $\beta > 0$, $r > 0$, etc. will (generally) be omitted and simply indicated by the symbol $\blacktriangle$.
- the dependence of the norm parameter p will also be indicated, but will not be given explicitly.
- the values of $C_\blacktriangle$ and $C_{\blacktriangle,p}$ might change from line to line.
- the expression "for n sufficiently large" means that the statement holds for $n \ge n_0$, with n_0 potentially depending on all model parameters (including σ, M and R), but not on η.

6.1 Preliminaries

Proposition 12 (Guo et al., 2017, Proposition 1) *Define*

$$\mathcal{B}_n(\lambda) := \left[1 + \left(\frac{2}{n\lambda} + \sqrt{\frac{\mathcal{N}(\lambda)}{n\lambda}}\right)^2\right]. \tag{29}$$

For any $\lambda > 0$, $\eta \in (0,1]$, with probability at least $1-\eta$ one has

$$\left\|(\bar{T}_{\mathbf{x}}+\lambda)^{-1}(\bar{T}+\lambda)\right\| \leq 8\log^2(2\eta^{-1})\mathcal{B}_n(\lambda)\ . \tag{30}$$

Corollary 13 *Let $\eta \in (0,1)$. For $n \in \mathbb{N}$ let $\tilde{\lambda}_n$ be implicitly defined as the unique solution of $\mathcal{N}(\tilde{\lambda}_n) = n\tilde{\lambda}_n$. Then for any $\lambda \in [\max(\tilde{\lambda}_n, n^{-1}), 1]$, one has*

$$\mathcal{B}_n(\lambda) \leq 10\ .$$

In particular,

$$\left\|(\bar{T}_{\mathbf{x}}+\lambda)^{-1}(\bar{T}+\lambda)\right\| \leq 80\log^2(2\eta^{-1})$$

holds with probability at least $1-\eta$.

We remark that the trace of $\bar{T}$ is bounded by 1. This ensures that the interval $[\tilde{\lambda}_n, 1]$ is non-empty.

Proof [of Corollary 13] Let $\tilde{\lambda}_n$ be defined via $\mathcal{N}(\tilde{\lambda}_n) = n\tilde{\lambda}_n$. Since $\mathcal{N}(\lambda)/\lambda$ is decreasing, we have for any $\lambda \geq \tilde{\lambda}_n$

$$\sqrt{\frac{\mathcal{N}(\lambda)}{n\lambda}} \leq \sqrt{\frac{\mathcal{N}(\tilde{\lambda}_n)}{n\tilde{\lambda}_n}} = 1\ .$$

Inserting this bound as well as $n\lambda \geq 1$ into (29) and (30) leads to the conclusion. ■

Corollary 14 *Assume the marginal distribution ν of $\mathcal{X}$ belongs to $\mathcal{P}^<(b,\beta)$ with $b > 1$ and $\beta > 0$. If λ_n is defined by (16) and if*

$$m_n \leq n^\alpha\ , \qquad \alpha < \frac{2br}{2br+b+1}\ ,$$

one has

$$\mathcal{B}_{\frac{n}{m_n}}(\lambda_n) \leq 2\ ,$$

provided n is sufficiently large.

Proof [of Corollary 14] We can for starters assume that n is sufficiently large so that $\lambda_n < 1$, i.e. $\lambda_n = \left(\frac{\sigma^2}{R^2 n}\right)^{\frac{b}{2br+b+1}}$ from (16). Recall that $\nu \in \mathcal{P}^<(b,\beta)$ implies $\mathcal{N}(\lambda_n) \leq C_\blacktriangle \lambda_n^{-\frac{1}{b}}$. Looking at the terms entering in $\mathcal{B}_{\frac{n}{m}}(\lambda_n)$, see (29), we have first, using the definition of λ_n in (16):

$$\frac{\mathcal{N}(\lambda_n)}{\frac{n}{m}\lambda_n} \leq C_\blacktriangle m\frac{\lambda_n^{-\frac{b+1}{b}}}{n} = C_\blacktriangle \frac{m}{n}\left(\frac{nR^2}{\sigma^2}\right)^{\frac{b+1}{2br+b+1}}\ ,$$

which (for fixed R, σ and other parameters entering in $C_\blacktriangle$) is $O(m_n n^{-\frac{2br}{2br+r+1}})$, and hence $o(1)$ provided

$$m_n \leq n^\alpha\ , \qquad \alpha < \frac{2br}{2br+b+1}\ .$$

For the second term entering in $\mathcal{B}_{\frac{n}{m}}(\lambda_n)$, we have

$$\frac{1}{\frac{n}{m}\lambda_n} = \frac{m}{n}\left(\frac{nR^2}{\sigma^2}\right)^{\frac{b}{2br+b+1}}\ ,$$

which is $O(m_n n^{-\frac{2br+1}{2br+b+1}}) = o(1)$, provided

$$m_n \leq n^\alpha\ , \qquad \alpha < \frac{2br+1}{2br+b+1}\ ,$$

which is implied by the previous stronger condition. ■

We shortly illustrate how Corollary 13 and Proposition 12 will be used. Let $u \in [0,1]$, $\tilde{\lambda}_n \leq \lambda$ as above and $f \in \mathcal{H}_K$. We have for any bounded operator A

$$\begin{aligned}\left\|\bar{T}^u A\right\| &= \left\|\bar{T}^u(\bar{T}+\lambda)^{-u}(\bar{T}+\lambda)^u(\bar{T}_{\mathbf{x}}+\lambda)^{-u}(\bar{T}_{\mathbf{x}}+\lambda)^u A\right\| \\ &\leq \left\|\bar{T}^u(\bar{T}+\lambda)^{-u}\right\| \left\|(\bar{T}+\lambda)^u(\bar{T}_{\mathbf{x}}+\lambda)^{-u}\right\| \left\|(\bar{T}_{\mathbf{x}}+\lambda)^u A\right\| \\ &\leq 8\log^{2u}(2\eta^{-1})\mathcal{B}_n(\lambda)^u \left\|(\bar{T}_{\mathbf{x}}+\lambda)^u A\right\|\ ,\end{aligned} \tag{31}$$

with probability at least $1-\eta$, for any $\eta \in (0,1)$; for the last inequality we have used that the first factor is less than 1, and for the second factor Proposition 12 in combination with the Cordes inequality (see Proposition 22 in the Appendix). In particular, for any $\max(\tilde{\lambda}_n, n^{-1}) \leq \lambda$ (with $\tilde{\lambda}_n$ as in Corollary 13)

$$\left\|\bar{T}^u A\right\| \leq 80^u \log^{2u}(2\eta^{-1}) \left\|(\bar{T}_{\mathbf{x}}+\lambda)^u A\right\|\ , \tag{32}$$

with probability at least $1-\eta$.

In the following, we constantly use (31). Furthermore, to bound terms involving residuals we will frequently use the following estimate: for $v \geq 0, u \in [0,2]$, and provided $u+v \leq q$ (q being the qualification):

$$\begin{aligned}\sup_{t\in[0,1]} |r_\lambda(t)t^v(t+\lambda)^u| &\leq 2\left(\sup_{t\in[0,1]} \left|r_\lambda(t)t^{v+u}\right| + \lambda^u \sup_{t\in[0,1]} |r_\lambda(t)t^v|\right) \\ &\leq C_\blacktriangle \lambda^{v+u},\end{aligned} \tag{33}$$

using twice (11) since $q \geq u+v$.

6.2 Approximation error bound

Recall that ν denotes the X-marginal of the sampling distribution ρ and $\mathcal{P}$ the set of all probability distributions on the input space $\mathcal{X}$.

Lemma 15 *Let $\nu \in \mathcal{P}$, $v \in \mathbb{R}$ and let $\mathbf{x} \in \mathcal{X}^{\frac{n}{m}}$ be an i.i.d. sample of size n/m, drawn according to ν. Assume the regularization $(g_\lambda)_\lambda$ has qualification $q \geq v+1+s$. Then with probability at least $1-\eta$:*

$$\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v(\bar{T}-\bar{T}_{\mathbf{x}})\right\| \leq C_{\blacktriangle}\log^4(4\eta^{-1})\lambda^{s+v+1}\mathcal{B}_{\frac{n}{m}}^{s+1}(\lambda)\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right).$$

Proof [of Lemma 15] From (30),(31) and from Proposition 20 recalled in the Appendix, one has

$$\begin{aligned}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v(\bar{T}-\bar{T}_{\mathbf{x}})\right\| &\leq C_{\blacktriangle}\log^{2(s+1)}(4\eta^{-1})\mathcal{B}_{\frac{n}{m}}^{s+1}(\lambda)\\ &\qquad \left\|(\bar{T}_{\mathbf{x}}+\lambda)^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v(\bar{T}_{\mathbf{x}}+\lambda)\right\|\left\|(\bar{T}+\lambda)^{-1}(\bar{T}-\bar{T}_{\mathbf{x}})\right\|\\ &\leq C_{\blacktriangle}\log^4(4\eta^{-1})\lambda^{s+v+1}\mathcal{B}_{\frac{n}{m}}^{s+1}(\lambda)\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right),\end{aligned}$$

for any $\lambda \in (0,1]$, $\eta \in (0,1]$, with probability at least $1-\eta$. We also used that $s \leq \frac{1}{2}$, and the estimate (33). ∎

Lemma 16 *Let $\nu \in \mathcal{P}$, $v \in \mathbb{R}$ and let $\mathbf{x} \in \mathcal{X}^{\frac{n}{m}}$ be an i.i.d. sample of size n/m drawn according to ν. Assume the regularization $(g_\lambda)_\lambda$ has qualification $q \geq v+s$. Then for any $\lambda \in (0,1]$, $\eta \in (0,1]$, with probability at least $1-\eta$*

$$\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v\right\| \leq C_{\blacktriangle}\log^{2s}(2\eta^{-1})\mathcal{B}_{\frac{n}{m}}^{s}(\lambda)\lambda^{s+v},$$

for some $C_{\blacktriangle} < \infty$.

Proof [of Lemma 16] Using (31), (33), since $q \geq v+s$, it holds

$$\begin{aligned}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v\right\| &\leq C_{\blacktriangle}\log^{2s}(2\eta^{-1})\mathcal{B}_{\frac{n}{m}}^{s}(\lambda)\left\|(\bar{T}_{\mathbf{x}}+\lambda)^s r_\lambda(\bar{T}_{\mathbf{x}})\bar{T}_{\mathbf{x}}^v\right\|\\ &\leq C_{\blacktriangle}\log^{2s}(2\eta^{-1})\mathcal{B}_{\frac{n}{m}}^{s}(\lambda)\lambda^{s+v},\end{aligned}$$

with probability at least $1-\eta$. ∎

Proposition 17 (Expectation of approximation error) *Let $f_\rho \in \Omega(r,R)$, $\lambda \in (0,1]$ and let $\mathcal{B}_{\frac{n}{m}}(\lambda)$ be defined in (29). Assume the regularization has qualification $q \geq r+s$. For any $p \geq 1$ one has:*

1. *If $r \leq 1$, then*

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}_D^\lambda)\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}R\,\lambda^{s+r}\,\mathcal{B}_{\frac{n}{m}}^{s+r}(\lambda).$$

2. *If $r > 1$, then*

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}_D^\lambda)\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}R\lambda^s\mathcal{B}_{\frac{n}{m}}^{s+1}(\lambda)\left(\lambda^r + \lambda\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right)\right).$$

In 1. and 2. the constant $C_{\blacktriangle,p}$ does not depend on $(\sigma, M, R) \in \mathbb{R}_+^3$.

Proof [of Proposition 17] Since $f_\rho \in \Omega(r,R)$,

$$\begin{aligned}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}_D^\lambda)\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} &= \left[\mathbb{E}_{\rho^{\otimes n}}\left\|\frac{1}{m}\sum_{j=1}^m \bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})f_\rho\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}\\ &\leq \frac{1}{m}\sum_{j=1}^m\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})f_\rho\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}\\ &\leq \frac{R}{m}\sum_{j=1}^m\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^r\right\|^p\right]^{\frac{1}{p}}. \qquad (34)\end{aligned}$$

The first inequality is just the triangle inequality for the p-norm $\|f\|_p = \mathbb{E}[\|f\|_{\mathcal{H}_K}^p]^{\frac{1}{p}}$. We bound the expectation for each separate subsample of size $\frac{n}{m}$ by first deriving a probabilistic estimate and then we integrate.

Consider first the case where $r \leq 1$. Using (31), the Cordes inequality (Proposition 22 in the Appendix), and (33) one has for any $j = 1, \ldots, m$,

$$\begin{aligned}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^r\right\| &\leq C_{\blacktriangle}\log^{2(s+r)}(4\eta^{-1})\mathcal{B}_{\frac{n}{m}}^{s+r}(\lambda)\left\|(\bar{T}_{\mathbf{x}_j}+\lambda)^s r_\lambda(\bar{T}_{\mathbf{x}_j})(\bar{T}_{\mathbf{x}_j}+\lambda)^r\right\|\\ &\leq C_{\blacktriangle}\log^3(4\eta^{-1})\lambda^{s+r}\mathcal{B}_{\frac{n}{m}}^{s+r}(\lambda),\end{aligned}$$

with probability at least $1-\eta$ and where $\mathcal{B}_{\frac{n}{m}}(\lambda)$ is defined in (29). Recall that the regularization has qualification $q \geq r+s$. By integration one has

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^r\right\|^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}\,\lambda^{s+r}\,\mathcal{B}_{\frac{n}{m}}^{s+r}(\lambda),$$

for some $C_{\blacktriangle,p} < \infty$, not depending on σ, M, R. Finally, from (34)

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}_D^\lambda)\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}R\,\lambda^{s+r}\,\mathcal{B}_{\frac{n}{m}}^{s+r}(\lambda).$$

In the case where $r \geq 1$, we write $r = k+u$, with $k = \lfloor r \rfloor$ and $u = r-k < 1$. We shall use the decomposition

$$\bar{T}^k = \sum_{l=0}^{k-1}\bar{T}_{\mathbf{x}}^l(\bar{T}-\bar{T}_{\mathbf{x}})\bar{T}^{k-(l+1)} + \bar{T}_{\mathbf{x}}^k. \qquad (35)$$

We proceed by bounding (34) according to decomposition (35) . For any $j = 1, \dots, m$, one has

$$\begin{aligned}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^{k+u}\right\|^p\right]^{\frac{1}{p}} &\leq \sum_{l=0}^{k-1}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^l_{\mathbf{x}_j}(\bar{T}-\bar{T}_{\mathbf{x}_j})\bar{T}^{k-(l+1)+u}\right\|^p\right]^{\frac{1}{p}} \\ &\quad + \left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^k_{\mathbf{x}_j}\bar{T}^u\right\|^p\right]^{\frac{1}{p}} \\ &\leq \sum_{l=0}^{k-1}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^l_{\mathbf{x}_j}(\bar{T}-\bar{T}_{\mathbf{x}_j})\right\|^p\right]^{\frac{1}{p}} \\ &\quad + \left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^k_{\mathbf{x}_j}\bar{T}^u\right\|^p\right]^{\frac{1}{p}} . \end{aligned} \tag{36}$$

Here we use that $\left\|\bar{T}^{k-(l+1)+u}\right\|$ is bounded by 1. By Lemma 16 and by (31), (33), with probability at least $1-\eta$

$$\begin{aligned}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^k_{\mathbf{x}_j}\bar{T}^u\right\| &\leq C_{\blacktriangle}\log^{2(s+u)}(2\eta^{-1})\mathcal{B}^{s+u}_{\frac{n}{m}}(\lambda)\left\|(\bar{T}_{\mathbf{x}_j}+\lambda)^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^k_{\mathbf{x}_j}(\bar{T}_{\mathbf{x}_j}+\lambda)^u\right\| \\ &\leq C_{\blacktriangle}\log^{2(s+u)}(2\eta^{-1})\mathcal{B}^{s+u}_{\frac{n}{m}}(\lambda)\lambda^{s+r},\end{aligned}$$

and thus integration yields

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^r_{\mathbf{x}_j}\bar{T}^u\right\|^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}\mathcal{B}^{s+u}_{\frac{n}{m}}(\lambda)\lambda^{s+r} . \tag{37}$$

For estimating the first term in (36) we may use Lemma 15. For any $l = 0, \dots, k-1$, we have $l+s+1 \leq k+s \leq r+s \leq q$, hence for any $j = 1, \dots, m$ with probability at least $1-\eta$

$$\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^l_{\mathbf{x}_j}(\bar{T}-\bar{T}_{\mathbf{x}_j})\right\| \leq C_{\blacktriangle}\log^4(8\eta^{-1})\lambda^{s+l+1}\mathcal{B}^{s+1}_{\frac{n}{m}}(\lambda)\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right) .$$

Again by integration, since $\lambda^l \leq 1$ for any $l = 0, \dots, k-1$, one has

$$\sum_{l=0}^{k-1}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s r_\lambda(\bar{T}_{\mathbf{x}_j})\bar{T}^l_{\mathbf{x}_j}(\bar{T}-\bar{T}_{\mathbf{x}_j})\right\|^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}\lfloor r\rfloor\lambda^{s+1}\mathcal{B}^{s+1}_{\frac{n}{m}}(\lambda)\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right) . \tag{38}$$

Finally, combining (37) and (38) into (36), then (34), gives in the case where $r > 1$

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}^\lambda_D)\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}\lambda^s\mathcal{B}^{s+1}_{\frac{n}{m}}(\lambda)\left(\lambda^r + \lambda\left(\frac{m}{n\lambda}+\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right)\right) .$$

The rest of the proof follows from (36).

■

Proof [of Theorem 3] Let λ_n be as defined by (16). According to Corollary 14, we have $\mathcal{B}_{\frac{n}{m_n}}(\lambda_n) \leq 2$ provided $\alpha < \frac{2br}{2br+b+1}$, for n sufficiently large. We can also assume n sufficiently large so that $\lambda_n < 1$, i.e., $R\lambda_n^{r+s} = a_n$ (from (16), (17)). Under these conditions, we immediately obtain from the first part of Proposition 17 in the case where $r \leq 1$

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}^{\lambda_n}_D)\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}R\,\lambda_n^{s+r} = C_{\blacktriangle,p}\,a_n .$$

We turn to the case where $r > 1$. We apply the second part of Proposition 17. By Corollary 14 we have

$$\begin{aligned}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}^{\lambda_n}_D)\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}} &\leq C_{\blacktriangle,p}R\lambda_n^s\mathcal{B}^{s+1}_{\frac{n}{m_n}}(\lambda_n)\left(\lambda_n^r + \lambda_n\left(\frac{m_n}{n\lambda_n}+\sqrt{\frac{m_n\mathcal{N}(\lambda_n)}{n\lambda_n}}\right)\right) \\ &\leq C_{\blacktriangle,p}\,R\lambda_n^s\,\left(\lambda_n^r + \lambda_n\left(\frac{m_n}{n\lambda_n}+\sqrt{m_n}\frac{R}{\sigma}\lambda_n^r\right)\right),\end{aligned}$$

where we used that $\mathcal{N}(\lambda_n) \leq C_{\blacktriangle}\lambda_n^{-1/b}$ and $\sigma\sqrt{\frac{\lambda_n^{-\frac{1}{b}}}{n\lambda_n}} = R\lambda_n^r$ coming from the definition of λ_n, and $\lambda_n < 1$. Furthermore,

$$\frac{m_n}{n\lambda_n} = o\left(\sqrt{m}_n\lambda_n^r\right) ,$$

provided

$$m_n \leq n^\alpha , \qquad \alpha < \frac{2(br+1)}{2br+b+1} .$$

Finally, for n sufficiently large, $\frac{R}{\sigma}\sqrt{m_n}\lambda_n \leq 1$, provided that

$$\alpha < \frac{2b}{2br+b+1} .$$

As a result, for any $p \geq 1$:

$$\limsup_{n\to\infty}\sup_{\rho\in\mathcal{M}_{\sigma,M,R}}\frac{\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(f_\rho - \tilde{f}^{\lambda_n}_D)\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}}}{a_n} \leq C_{\blacktriangle,p} ,$$

for some $C_{\blacktriangle,p} < \infty$, not depending on σ, M, R. ■

6.3 Sample error bound

The main idea for deriving an upper bound for the sample error is to identify it as a sum of unbiased Hilbert space-valued i.i.d. variables and then to apply a suitable version of Rosenthal's inequality.

Given $\lambda \in (0,1]$, we define the random variable $\xi_\lambda : (\mathcal{X}\times\mathbb{R})^{\frac{n}{m}} \longrightarrow \mathcal{H}_K$ by

$$\xi_\lambda(\mathbf{x},\mathbf{y}) := \bar{T}^s g_\lambda(\bar{T}_\mathbf{x})(\bar{T}_\mathbf{x} f_\rho - \bar{S}^*_\mathbf{x}\mathbf{y}) \,.$$

Recall that according to Assumption (3), the conditional expectation w.r.t. ρ of Y given X satisfies

$$\mathbb{E}_\rho[Y|X=x] = \bar{S}_x f_\rho \,,$$

implying that ξ_λ has zero expectation (since $\bar{T}_\mathbf{x} = \bar{S}^*_\mathbf{x}\bar{S}_\mathbf{x}$). Thus,

$$\bar{T}^s(\tilde{f}^\lambda_D - \bar{f}^\lambda_D) \;=\; \frac{1}{m}\sum_{j=1}^m \xi_\lambda(\mathbf{x}_j,\mathbf{y}_j) \tag{39}$$

is a sum of centered i.i.d. random variables.

Furthermore, we need the following result (Pinelis, 1994, Theorem 5.2) , which generalizes Rosenthal's (1970) inequalities (originally only formulated for real valued random variables) to random variables with values in a Banach space. For Hilbert spaces this looks particularly nice.

Proposition 18 *Let $\mathcal{H}$ be a Hilbert space and $\xi_1,\dots,\xi_m$ be a finite sequence of independent, mean zero $\mathcal{H}$- valued random variables. If $2 \le p < \infty$, then there exists a constant $C_p > 0$, only depending on p, such that*

$$\left(\mathbb{E}\Big\| \frac{1}{m}\sum_{j=1}^m \xi_j \Big\|^p_\mathcal{H}\right)^{\frac{1}{p}} \;\le\; \frac{C_p}{m}\;\max\left\{\left(\sum_{j=1}^m \mathbb{E}\|\xi_j\|^p_\mathcal{H}\right)^{\frac{1}{p}}, \left(\sum_{j=1}^m \mathbb{E}\|\xi_j\|^2_\mathcal{H}\right)^{\frac{1}{2}}\right\}. \tag{40}$$

We remark in passing that Dirksen (2011), Corollary 1.22, establishes the interesting result that in addition to the upper bound in (40) there is also a corresponding lower bound where the constant C_p is replaced by another constant $C'_p > 0$, only depending on p.

Proposition 19 (Expectation of sample error) *Let ρ be a source distribution belonging to $\mathcal{M}_{\sigma,M,R}$, $s \in [0,\frac{1}{2}]$ and let $\lambda \in (0,1]$. Define $\mathcal{B}_{\frac{n}{m}}(\lambda)$ as in* (29)*. Assume the regularization has qualification $q \ge r+s$. For any $p \ge 1$ one has:*

$$\left[\mathbb{E}_{\rho^{\otimes n}}\big\|\bar{T}^s(\tilde{f}^\lambda_D - \bar{f}^\lambda_D)\big\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}} \le C_{\blacktriangle,p}\, m^{-\frac{1}{2}}\mathcal{B}_{\frac{n}{m}}(\lambda)^{\frac{1}{2}+s}\lambda^s\left(\frac{mM}{n\lambda} + \sigma\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right),$$

where C_p does not depend on $(\sigma,M,R) \in \mathbb{R}^3_+$.

Proof [of Proposition 19] Let $\lambda \in (0,1]$ and $p \ge 2$. From Proposition 18

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s\tilde{f}^\lambda_D - \bar{f}^\lambda_D\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}} = \left[\mathbb{E}_{\rho^{\otimes n}}\left\|\frac{1}{m}\sum_{j=1}^m \xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\right\|^p_{\mathcal{H}_K}\right]^{\frac{1}{p}}$$
$$\le \frac{C_p}{m}\;\max\left\{\left(\sum_{j=1}^m\left[\mathbb{E}_{\rho^{\otimes n}}\|\xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\|^p_{\mathcal{H}_K}\right]\right)^{\frac{1}{p}}, \left(\sum_{j=1}^m\left[\mathbb{E}_{\rho^{\otimes n}}\|\xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\|^2_{\mathcal{H}_K}\right]\right)^{\frac{1}{2}}\right\}. \tag{41}$$

Again, the estimates in expectation will follow from integrating a bound holding with high probability. By (31), one has for any $j = 1,\dots,m$,

$$\begin{aligned}\|\xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\|_{\mathcal{H}_K} &= \|\bar{T}^s g_\lambda(\bar{T}_{\mathbf{x}_j})(\bar{T}_{\mathbf{x}_j} f_\rho - \bar{S}^*_{\mathbf{x}_j}\mathbf{y}_j)\|_{\mathcal{H}_K} \\ &\le 8\log^{2s}(4\eta^{-1})\mathcal{B}_{\frac{n}{m}}(\lambda)^s\|(\bar{T}_{\mathbf{x}_j}+\lambda)^s g_\lambda(\bar{T}_{\mathbf{x}_j})(\bar{T}_{\mathbf{x}_j} f_\rho - \bar{S}^*_{\mathbf{x}_j}\mathbf{y}_j)\|_{\mathcal{H}_K}\,, \end{aligned}\tag{42}$$

holding with probability at least $1-\frac{\eta}{2}$, where $\mathcal{B}_{\frac{n}{m}}(\lambda)$ is defined in (29). We proceed by splitting:

$$(\bar{T}_{\mathbf{x}_j}+\lambda)^s g_\lambda(\bar{T}_{\mathbf{x}_j})(\bar{T}_{\mathbf{x}_j} f_\rho - \bar{S}^\star_{\mathbf{x}_j}\mathbf{y}_j) = H^{(1)}_{\mathbf{x}_j}\cdot H^{(2)}_{\mathbf{x}_j}\cdot h^\lambda_{\mathbf{z}_j}\,,$$

with

$$\begin{aligned} H^{(1)}_{\mathbf{x}_j} &:= ||(\bar{T}_{\mathbf{x}_j}+\lambda)^s g_\lambda(\bar{T}_{\mathbf{x}_j})(\bar{T}_{\mathbf{x}_j}+\lambda)^{\frac{1}{2}}||, \\ H^{(2)}_{\mathbf{x}_j} &:= ||(\bar{T}_{\mathbf{x}_j}+\lambda)^{-\frac{1}{2}}(\bar{T}+\lambda)^{\frac{1}{2}}||, \\ h^\lambda_{\mathbf{z}_j} &:= ||(\bar{T}+\lambda)^{-\frac{1}{2}}(\bar{T}_{\mathbf{x}_j} f_\rho - \bar{S}^\star_{\mathbf{x}_j}\mathbf{y}_j)||_{\mathcal{H}_K}\,. \end{aligned}$$

The first term is estimated using (9),(10) and gives for $s \in [0,\frac{1}{2}]$

$$\begin{aligned} H^{(1)}_{\mathbf{x}_j} &\le \sup_{t\in[0,1]}\left(g_\lambda(t)(t+\lambda)^{s+\frac{1}{2}}\right) \\ &\le 2\Big(\sup_{t\in[0,1]} g_\lambda(t)t^{s+\frac{1}{2}} + \lambda^{s+\frac{1}{2}}\sup_{t\in[0,1]} g_\lambda(t)\Big) \\ &\le 2\Big(\Big(\sup_{t\in[0,1]} g_\lambda(t)\Big)^{\frac{1}{2}-s}\Big(\sup_{t\in[0,1]} g_\lambda(t)t\Big)^{s+\frac{1}{2}} + \lambda^{s+\frac{1}{2}}\sup_{t\in[0,1]} g_\lambda(t)\Big) \\ &\le C_\blacktriangle\lambda^{s-\frac{1}{2}}\,. \end{aligned}\tag{43}$$

The second term is now bounded using (31) once more. One has with probability at least $1-\frac{\eta}{4}$

$$H^{(2)}_{\mathbf{x}_j} \le 8\log(8\eta^{-1})\mathcal{B}_{\frac{n}{m}}(\lambda)^{\frac{1}{2}}\,. \tag{44}$$

Finally, $h^\lambda_{\mathbf{z}_j}$ is estimated using Proposition 21:

$$h^\lambda_{\mathbf{z}_j} \le 2\log(8\eta^{-1})\left(\frac{mM}{n\sqrt{\lambda}} + \sigma\sqrt{\frac{m\mathcal{N}(\lambda)}{n}}\right), \tag{45}$$

holding with probability at least $1-\frac{\eta}{4}$. Thus, combining (43), (44) and (45) with (42) gives for any $j = 1,\dots,m$,

$$\|\xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\|_{\mathcal{H}_K} \le C_\blacktriangle\log^{2(s+1)}(8\eta^{-1})\mathcal{B}_{\frac{n}{m}}(\lambda)^{\frac{1}{2}+s}\lambda^s\left(\frac{mM}{n\lambda} + \sigma\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right),$$

with probability at least $1-\eta$. Integration gives for any $p \ge 2$:

$$\sum_{j=1}^m\left[\mathbb{E}_{\rho^{\otimes n}}\big\|\xi_\lambda(\mathbf{x}_j,\mathbf{y}_j)\big\|^p_{\mathcal{H}_K}\right] \le C_{\blacktriangle,p}\, m\mathcal{A}^p\,,$$

with

$$\mathcal{A} := \mathcal{A}_{\frac{n}{m}}(\lambda) := \mathcal{B}_{\frac{n}{m}}(\lambda)^{\frac{1}{2}+s}\lambda^s \left(\frac{mM}{n\lambda} + \sigma\sqrt{\frac{m\mathcal{N}(\lambda)}{n\lambda}}\right) .$$

Combining this with (41) implies, since $p \geq 2$:

$$\begin{aligned}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(\tilde{f}_D^{\lambda} - \bar{f}_D^{\lambda})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} &\leq \frac{C_{\blacktriangle,p}}{m}\max\left((m\mathcal{A}^p)^{\frac{1}{p}}, (m\mathcal{A}^2)^{\frac{1}{2}}\right) \\ &= \frac{C_{\blacktriangle,p}}{m}\,\mathcal{A}\max\left(m^{\frac{1}{p}}, m^{\frac{1}{2}}\right) \\ &= \frac{C_{\blacktriangle,p}}{\sqrt{m}}\,\mathcal{A}\,,\end{aligned}$$

where $C_{\blacktriangle,p}$ does not depend on $(\sigma, M, R) \in \mathbb{R}_+^3$. The result for the case $1 \leq p \leq 2$ immediately follows from Hölder's inequality. ■

Proof [of Theorem 4] Let λ_n be as defined by (16); as earlier we assume n is big enough so that $\lambda_n < 1$. According to Corollary 14, we have $\mathcal{B}_{\frac{n}{m}}(\lambda_n) \leq 2$ provided $\alpha < \frac{2br}{2br+b+1}$ and n is sufficiently large. Under this condition we immediately obtain from Proposition 19:

$$\begin{aligned}\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(\tilde{f}_D^{\lambda_n} - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} &\leq \frac{C_{\blacktriangle,p}}{\sqrt{m}}\,\lambda_n^s \left(\frac{mM}{n\lambda_n} + \sigma\sqrt{\frac{m\mathcal{N}(\lambda_n)}{n\lambda_n}}\right) \\ &\leq C_{\blacktriangle,p}\lambda_n^s \left(\frac{\sqrt{m}M}{n\lambda_n} + \sigma\sqrt{\frac{\lambda_n^{-\frac{1}{b}}}{n\lambda_n}}\right) ,\end{aligned}$$

where we used again that $\mathcal{N}(\lambda_n) \leq C_{\blacktriangle}\lambda_n^{-1/b}$; now

$$\frac{\sqrt{m}_n M}{n\lambda_n} = o\left(\sigma\sqrt{\frac{\lambda_n^{-1/b}}{n\lambda_n}}\right) ,$$

provided

$$m_n \leq n^{\alpha} , \qquad \alpha < \frac{2(br+1)}{2br+b+1} .$$

Recalling that $\sigma\sqrt{\frac{\lambda_n^{-1/b}}{n\lambda_n}} = R\lambda_n^r = \lambda_n^{-s}a_n$, we arrive at

$$\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(\tilde{f}_D^{\lambda_n} - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}} \leq C_{\blacktriangle,p}\, a_n .$$

As a result, for any $p \geq 1$:

$$\limsup_{n\to\infty}\sup_{\rho\in\mathcal{M}_{\sigma,M,R}} \frac{\left[\mathbb{E}_{\rho^{\otimes n}}\left\|\bar{T}^s(\tilde{f}_D^{\lambda_n} - \bar{f}_D^{\lambda_n})\right\|_{\mathcal{H}_K}^p\right]^{\frac{1}{p}}}{a_n} \leq C_{\blacktriangle,p} ,$$

for some $C_{\blacktriangle,p}$, not depending on the model parameters $(\sigma, M, R) \in \mathbb{R}_+^3$, thus leading to the conclusion. ■

Appendix A

Proposition 20 (see e.g. Blanchard and Mücke, 2017, Proposition 5.3) *For any $n \in \mathbb{N}$, $\lambda \in (0,1]$ and $\eta \in (0,1)$, one has with probability at least $1-\eta$:*

$$\left\|(\bar{T}+\lambda)^{-1}(\bar{T}-\bar{T}_{\mathbf{x}})\right\|_{\mathrm{HS}} \leq 2\log(2\eta^{-1})\left(\frac{2}{n\lambda} + \sqrt{\frac{\mathcal{N}(\lambda)}{n\lambda}}\right) ,$$

where $\|.\|_{HS}$ denotes the Hilbert-Schmidt norm. (Since the operator norm is bounded by the Hilbert-Schmidt norm, the above statement also holds for the operator norm.)

Proposition 21 (see e.g. Blanchard and Mücke, 2017, Proposition 5.2) *For $n \in \mathbb{N}$, $\lambda \in (0,1]$ and $\eta \in (0,1]$, it holds with probability at least $1-\eta$:*

$$\left\|(\bar{T}+\lambda)^{-\frac{1}{2}}\ \left(\bar{T}_{\mathbf{x}}f_\rho - \bar{S}_{\mathbf{x}}^{\star}\mathbf{y}\right)\right\|_{\mathcal{H}_K} \leq 2\log(2\eta^{-1})\left(\frac{M}{n\sqrt{\lambda}} + \sqrt{\frac{\sigma^2\mathcal{N}(\lambda)}{n}}\right) .$$

Proposition 22 (Cordes Inequality, see e.g. Bhatia, 1997, Theorem IX.2.1-2) *Let A, B be to self-adjoint, positive operators on a Hilbert space. Then for any $s \in [0,1]$:*

$$\|A^sB^s\| \leq \|AB\|^s . \tag{46}$$

References

F. Bauer, S. Pereverzev, and L. Rosasco. On regularization algorithms in learning theory. *J. Complexity*, 23(1):52–72, 2007.

R. Bhatia. *Matrix Analysis*. Springer, 1997.

G. Blanchard and N. Mücke. Optimal rates for regularization of statistical inverse learning problems. *Foundations of Computational Mathematics*, 2017.

X. Chang, S.-B. Lin, and Y. Wang. Divide and conquer local average regression. *Electron. J. Statist.*, 11(1):1326–1350, 2017. doi: 10.1214/17-EJS1265. URL `https://doi.org/10.1214/17-EJS1265`.

G. Cheng and Z. Shang. Computational limits of divide-and-conquer method. Technical report, arXiv:1512.09226, 2016.

E. De Vito and A. Caponnetto. Optimal rates for regularized least-squares algorithm. *Foundations of Computational Mathematics*, 7(3):331–368, 2006.

L. H. Dicker, D. P. Foster, and D. Hsu. Kernel ridge vs. principal component regression: Minimax bounds and the qualification of regularization operators. *Electron. J. Statist.*, 11(1):1022–1047, 2017. doi: 10.1214/17-EJS1258.

S. Dirksen. *Noncommutative and vector-valued Rosenthal inequalities*. PhD thesis, Delft Univ. Technology, 2011.

H. Engl, M. Hanke, and A. Neubauer. *Regularization of Inverse Problems*. Kluwer Academic Publishers, 2000.

J. C. Ferreira and V. A. Menegatto. Eigenvalues of integral operators defined by smooth positive definite kernels. *Integral Equations and Operator Theory*, 64(1):61–81, May 2009. ISSN 1420-8989.

L. L. Gerfo, L. Rosasco, F. Odone, E. De Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

Q. Guo, B.-W. Chen, S. Rho, W. Ji, F. Jiang, X. Ji, and S.-Y. Kung. Efficient divide-and-conquer classification based on parallel feature-space decomposition for distributed systems. *IEEE Systems Journal*, 2015.

Z.-C. Guo, S.-B. Lin, and D.-X. Zhou. Learning theory of distributed spectral algorithms. *Inverse Problems*, 33(7):074009, 2017.

C. J. Hsieh, S. Si, and I. Dhillon. A divide-and-conquer solver for kernel support vector machine. *Proceedings of the 31. International Conference on Machine Learning*, 32(1): 575–583, 2014.

R. Li, D. K. J. Lin, and B. Li. Statistical inference in massive data sets. *Applied Stochastic Models in Business and Industry*, 29 (5):399–409, 2013.

S. Lin, X. Guo, and D.-X. Zhou. Distributed learning with regularized least squares. *Journal of Machine Learning Research*, 18:1–31, 2017.

S.-B. Lin and D.-X. Zhou. Distributed kernel-based gradient descent algorithms. *Constructive Approximation*, 47(2):249–276, 2018.

L. Mackey, A. Talwalkar, and M. I. Jordan. Divide-and-conquer matrix factorization. *Advances in Neural Information Processing Systems 24 (NIPS 2011)*, 2011.

I. Pinelis. Optimum bounds for the distributions of martingales in Banach spaces. *The Annals of Probability*, 22(4):1679–1706, 1994.

H. P. Rosenthal. On the subspaces of L^p $(p > 2)$ spanned by sequences of independent random variables. *Israel J. Math.*, 8:273–303, 1970.

C. Xu, Y. Zhang, R. Li, and X. Wu. On the feasibility of distributed kernel regression for big data. *IEEE Transactions on Knowledge and Data Engineering*, 28:3041–3052, 2016.

Y. Zhang, J. Duchi, and M. Wainwright. Divide and conquer kernel ridge regression. *JMLR: Workshop and Conference Proceedings*, 30:1–26, 2013.

D.-X. Zhou. The covering number in learning theory. *Journal of Complexity*, 18 (3):739–767, 2002.

Journal of Machine Learning Research 19 (2018) 1-37 Submitted 5/17; Revised 4/18; Published 9/18

A Direct Approach for Sparse Quadratic Discriminant Analysis

Binyan Jiang BY.JIANG@POLYU.EDU.HK
Department of Applied Mathematics
The Hong Kong Polytechnic University
Hong Kong, China

Xiangyu Wang XIANGYUW@GOOGLE.COM
Google LLC
1600 Amphitheatre Pkwy
Mountain view, CA 94043, USA

Chenlei Leng C.LENG@WARWICK.AC.UK
Department of Statistics
University of Warwick
and Alan Turing Institute
Coventry, CV4 7AL, UK

Editor: Saharon Rosset

Abstract

Quadratic discriminant analysis (QDA) is a standard tool for classification due to its simplicity and flexibility. Because the number of its parameters scales quadratically with the number of the variables, QDA is not practical, however, when the dimensionality is relatively large. To address this, we propose a novel procedure named DA-QDA for QDA in analyzing high-dimensional data. Formulated in a simple and coherent framework, DA-QDA aims to directly estimate the key quantities in the Bayes discriminant function including quadratic interactions and a linear index of the variables for classification. Under appropriate sparsity assumptions, we establish consistency results for estimating the interactions and the linear index, and further demonstrate that the misclassification rate of our procedure converges to the optimal Bayes risk, even when the dimensionality is exponentially high with respect to the sample size. An efficient algorithm based on the alternating direction method of multipliers (ADMM) is developed for finding interactions, which is much faster than its competitor in the literature. The promising performance of DA-QDA is illustrated via extensive simulation studies and the analysis of four real datasets.

Keywords: Bayes Risk, Consistency, High Dimensional Data, Linear Discriminant Analysis, Quadratic Discriminant Analysis, Sparsity

1. Introduction

Classification is a central topic in statistical learning and data analysis. Due to its simplicity for producing quadratic decision boundaries, quadratic discriminant analysis (QDA) has become an important technique for classification, adding an extra layer of flexibility to the linear discriminant analysis (LDA); see Hastie et al. (2009). Despite its usefulness, the number of the parameters needed by QDA scales squarely with that of the variables, making it quickly inapplicable for problems with large or even moderate dimensionality. This problem is extremely eminent in the era of big data, as one often encounters datasets with the dimensionality larger, often times substantially larger, than the sample size. This paper aims to develop a novel classification approach named DA-QDA, short for Direct Approach for QDA to make QDA useful for analyzing ultra-high dimensional data.

For ease of presentation, we focus on binary problems where observations are from two classes. Suppose that the observations from class 1 follow $X \sim N(\mu_1, \Sigma_1)$ and those from class 2 satisfy $Y \sim N(\mu_2, \Sigma_2)$, where $\mu_k \in \mathbb{R}^p, k = 1, 2$ are the mean vectors and $\Sigma_k \in \mathbb{R}^{p\times p}, k = 1, 2$ are the two covariance matrices. Compared with LDA, it is assumed that $\Sigma_1 \neq \Sigma_2$ in QDA, which gives rise to a class boundary that is quadratic in terms of the variables. Bayes' rule classifies a new observation z to class 1 if $\pi_1 f(z|\mu_1, \Sigma_1) > \pi_2 f(z|\mu_2, \Sigma_2)$, where $f(z|\mu, \Sigma)$ is the probability density function of a multivariate normal distribution with mean μ and variance Σ, and π_1 and π_2 are the two prior probabilities. Following simple algebra, the Bayes discriminant function for a new observation z is seen as

$$D(z) = (z - \mu)^T \Omega (z - \mu) + \delta^T (z - \mu) + \eta,$$

where $\mu = (\mu_1 + \mu_2)/2$ is the mean of the two centroids, $\Omega = \Sigma_2^{-1} - \Sigma_1^{-1}$ is the difference of the two precision matrices, $\delta = (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2)$, and $\eta = 2\log(\pi_1/\pi_2) + \frac{1}{4}(\mu_1 - \mu_2)^T \Omega(\mu_1 - \mu_2) + \log|\Sigma_2| - \log|\Sigma_1|$; see for example Anderson (2003). Note that the discriminant function becomes that of LDA when $\Sigma_1 = \Sigma_2 = \Sigma$. Completely analogous to a two-way interaction model in linear regression, δ in $D(z)$ can be seen as a linear index of the variables whose nonzero entries play the role of main effects, whereas the nonzero entries in Ω can be understood as interactions of second-order between the variables. Although there are other ways to represent the discriminant function, $D(z)$ is used as it is a quadratic function of $z - \mu$, making the discriminant function location-invariant with respect to the coordinates. For easy reference, we shall call subsequently the parameters Ω, δ, μ, and η in the Bayes discriminant function collectively as Bayes components.

1.1 Our contributions

We highlight the main contributions of this paper as follows.

1. DA-QDA is the first *direct* approach for sparse QDA in a high dimensional setup. That is, Ω, δ, μ, and η in the Bayes discriminant function are directly estimated with only sparse assumptions on Ω and δ but not on other intermediate quantities;

2. For estimating Ω, an intermediate step of DA-QDA and a problem of interest in its own right, we develop a new algorithm which is much more computationally and memory efficient than its competitor. See Section 2.1;

3. We develop new theory to show the theoretical attractiveness of the DA-QDA. In particular, the theory for estimating δ is new. See Section 3;

4. The problem of finding the right intercept η is of considerable interest but a general theory on estimated η is lacking (Hastie et al., 2009). We provide a first theory for the convergence property of our estimated η. See Section 3.4;

5. In extensive simulation study and real data analysis, the DA-QDA approach outperforms many of its competitors, especially when variables under considerable interact. See Section 4.

1.2 Literature review

As more and more modern datasets are high-dimensional, the problem of classification in this context has received increasing attention as the usual practice of using empirical estimates for the Bayes components is no longer applicable. Bickel and Levina (2004) first highlighted that LDA is equivalent to random guessing in the worst case scenario when the dimensionality is larger than the sample size. Scientifically and practically in many problems, however, the components in the Bayes discriminant function can be assumed sparse. In the problem we study in this paper, loosely speaking, the notion of sparsity entertains that the two-way interaction representation of the model only admits a small number of main effects and interactions. In the past few years, a plethora of methods built on suitable sparsity assumptions have been proposed to estimate the main effects as in LDA; see for example Shao et al. (2011), Cai and Liu (2011), Fan et al. (2012), Mai et al. (2012), Mai and Zou (2013), and Jiang et al. (2015). Other related linear methods for high-dimensional classification can be found in Leng (2008), Witten and Tibshirani (2011), Pan et al. (2016), Mai et al. (2015), among others.

As pointed out by Fan et al. (2015b) and Sun and Zhao (2015), it has been increasingly recognized that the assumption of a common covariance matrix across different classes, needed by LDA, can be restrictive in many practical problems. The extra layer of flexibility offered by QDA that deals with two-way variable interactions makes it extremely attractive for such problems. Li and Shao (2015) studied sparse QDA by making sparsity assumptions on $\mu_2 - \mu_1, \Sigma_1, \Sigma_2$ and $\Sigma_1 - \Sigma_2$ and proposed their sparse estimates. The assumptions made are not directly on the key quantities needed in the discriminant function $D(z)$. In addition, good estimates of these four quantities do not necessarily translate to better classification, a phenomenon similarly argued and observed by Cai and Liu (2011) and Mai et al. (2012) for LDA. Fan et al. (2015b) proposed a screening method to identify interactions when Ω admits a two block sparse structure after permutation, before applying penalized logistic regression on the identified interactions and all the main effects to estimate a sparser model. Their method cannot deal with problems where the support of Ω is in general positions. Further, the use of a separate second-step penalized logistic regression to determine important interactions and main effects is less appealing from a methodological perspective. Fan et al. (2015a) suggested a Rayleigh quotient based method for which all the fourth cross-moments of the predictors have to be estimated. Despite all these efforts, a direct yet simple approach for QDA with less stringent assumptions than in Li and Shao (2015) for high-dimensional analysis is missing.

The proposed DA-QDA approach in this paper aims to overcome the difficulties mentioned above. In particular, compared with Li and Shao (2015), we only make sparsity assumptions on Ω and δ and estimate these two quantities directly in DA-QDA. Compared to Fan et al. (2015b), we allow the interactions in Ω in general positions, without resorting to a second stage approach for interactions and main effects selection. Compared with Fan et al. (2015a), we operate directly on QDA for which only second cross-moments of the variables are needed.

DA-QDA can also be understood as a novel attempt to select interactions in the discriminant function that correspond to the nonzero entries in Ω. The problem of interaction selection is a problem of its own importance and has been studied extensively recently for regression problems. See, for example, Hao and Zhang (2014) and references therein. The problem of estimating Ω alone has also attracted attention lately in a different context. To understand how the structure of a network differs between different conditions and to find the common structures of two different Gaussian graphical models, Zhao et al. (2014) proposed a direct approach for estimating Ω by formulating their procedure via the Dantzig selector. A severe limitation is that their linear programming procedure needs to deal with $O(p^2)$ constraints, and the memory requirement by the large constraint matrix is of the order $O(p^4)$. As a result, an iteration of the algorithm in Zhao et al. (2014) requires $O(sp^4)$ computations, where s is the cardinality of the support of Ω. Apparently, their method does not scale well to high dimensional data. In Zhao et al. (2014), problems with maximum size $p = 120$ were attempted and it was reported that a long time was needed to run their method. In contrast, we use a lasso formulation and develop a new algorithm based on the alternating direction methods of multipliers (ADMM) for estimating Ω. The memory requirement of our algorithm is of the order $O(p^2)$ and its computational cost is of the order $O(p^3)$ per iteration, enabling DA-QDA to easily handle much larger problems.

The rest of the paper is organized as follows. Section 2 outlines the main DA-QDA methodology for estimating Ω and δ. A novel algorithm based on ADMM for estimating Ω is developed. Section 3 investigates the theory of DA-QDA and provides various consistency results for estimating Ω, δ, and η, as well as establishing the consistency of the misclassification risk relative to the Bayes risk. Section 4 presents extensive numerical studies and analysis of four real datasets. Comparison with other classification methods demonstrates that DA-QDA is very competitive in estimating the sparsity pattern and the parameters of interest. We provide a short discussion and outline future directions of research in Section 5. All the proofs are relegated to the Appendix.

2. DA-QDA Methodology

To obtain an estimator for the Bayes discriminant function $D(z)$, we propose direct estimators for the two of its Bayes components $\Omega = \Sigma_2^{-1} - \Sigma_1^{-1}$ and $\delta = (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2)$ under appropriate sparsity assumptions. Given data $X_j,\ j = 1,..,n_1$ from class 1 and $Y_k,\ k = 1,...,n_2$ from class 2, we can estimate μ_i and $\Sigma_i,\ i = 1, 2$, via their sample versions as

$$\hat{\mu}_1 = \frac{1}{n_1}\sum_{j=1}^{n_1} X_j,\ \hat{\mu}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2} Y_j;$$

$$\hat{\Sigma}_1 = \frac{1}{n_1}\sum_{j=1}^{n_1}(X_j - \hat{\mu}_1)(X_j - \hat{\mu}_1)^T,\ \hat{\Sigma}_2 = \frac{1}{n_2}\sum_{j=1}^{n_2}(Y_j - \hat{\mu}_2)(Y_j - \hat{\mu}_2)^T.$$

When $p >> \max\{n_1, n_2\}$, $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ are degenerate and cannot be directly used for estimating Ω. Denote the DA-QDA estimates of Ω as $\hat{\Omega}$ and δ as $\hat{\delta}$ which will be obtained as

in (2) and (8) respectively. For a given scalar η, our DA-QDA procedure classifies a new observation z in to class 1 if

$$\left[z-\frac{\hat{\mu}_1+\hat{\mu}_2}{2}\right]^T\hat{\Omega}\left[z-\frac{\hat{\mu}_1+\hat{\mu}_2}{2}\right]+\hat{\delta}^T\left[z-\frac{\hat{\mu}_1+\hat{\mu}_2}{2}\right]+\eta>0, \tag{1}$$

and classifies z into class 2 otherwise. From (1), we emphasize again that the nonzero entries in $\hat{\Omega}$ are the interactions of the variables that contribute to the classification rule, while the nonzero entries in $\hat{\delta}$ are the main effects of the variables that are used for classification. In the linear discriminant analysis when $\Sigma_1=\Sigma_2$, the rule in (1) becomes the LDA rule which is linear in the variables. As $\eta=2\log(\pi_1/\pi_2)+\frac{1}{4}(\mu_1-\mu_2)^T\Omega(\mu_1-\mu_2)+\log|\Sigma_2|-\log|\Sigma_1|$ is a scalar, we can choose η as $\hat{\eta}$ using a simple grid search, bypassing the need to estimate the determinants of Σ_1 and Σ_2. This is the strategy implemented in Section 4 and its analytical justification is provided in Section 3.4. Thus in the following, we shall focus on the estimation of Ω and δ, under certain sparsity assumptions on these two quantities.

2.1 Estimating Ω

Recall $\Omega=\Sigma_2^{-1}-\Sigma_1^{-1}$. It may be attempting to first estimate Σ_1^{-1} and Σ_2^{-1} as intermediate quantities before taking their difference. It is known, however, that accurate estimation of a covariance matrix or its inverse can be difficult in general in high dimensions unless additional assumptions are imposed (cf. Bickel and Levina (2008)). Because Ω is the quantity of interest that appears in the Bayes' rule, we propose to estimate it directly. To proceed, we note that $\Sigma_2\Omega\Sigma_1=\Sigma_1\Omega\Sigma_2=\Sigma_1-\Sigma_2$. If we define a loss function as $Tr\left(\Omega^T\Sigma_1\Omega\Sigma_2\right)/2-Tr\left(\Omega(\Sigma_1-\Sigma_2)\right)$, the loss function is minimized when Ω satisfies $\Sigma_2\Omega\Sigma_1=\Sigma_1-\Sigma_2$ or $\Omega=\Sigma_2^{-1}-\Sigma_1^{-1}$. This simple observation motivates the following penalized loss formulation for estimating Ω by replacing $\Sigma_j, j=1,2$ by their empirical estimates as

$$\hat{\Omega}=\arg\min_{\Omega\in R^{p\times p}}\ \frac{1}{2}Tr\left(\Omega^T\hat{\Sigma}_1\Omega\hat{\Sigma}_2\right)-Tr\left(\Omega(\hat{\Sigma}_1-\hat{\Sigma}_2)\right)+\lambda\|\Omega\|_1, \tag{2}$$

where $\|\Omega\|_1$ is the ℓ_1 penalty of the vectorized Ω to encourage sparsity and λ is the tuning parameter. To obtain a symmetric estimator for Ω, we may simply use $\hat{\Omega}_0=\frac{1}{2}(\hat{\Omega}+\hat{\Omega}^T)$ after $\hat{\Omega}$ is obtained. Because the second derivative of the above loss function is $\hat{\Sigma}_2\otimes\hat{\Sigma}_1$ which is nonnegative definite, the formulation in (2) is a convex problem and can be solved by a convex optimization algorithm.

We now develop an ADMM algorithm to solve for $\hat{\Omega}$ in (2) (Boyd et al., 2011; Zhang and Zou, 2014). First write the optimization problem in (2) as

$$\min_{\Omega\in R^{p\times p}}\ \frac{1}{2}Tr\left(\Omega^T\hat{\Sigma}_1\Omega\hat{\Sigma}_2\right)-Tr\left(\Omega(\hat{\Sigma}_1-\hat{\Sigma}_2)\right)+\lambda\|\Psi\|_1,\ s.t.\ \Psi=\Omega. \tag{3}$$

From this, we can form the augmented Lagrangian as

$$\begin{aligned}L(\Omega,\Psi,\Lambda) &= \frac{1}{2}Tr\left(\Omega^T\hat{\Sigma}_1\Omega\hat{\Sigma}_2\right)-Tr\left(\Omega(\hat{\Sigma}_1-\hat{\Sigma}_2)\right)+\lambda\|\Psi\|_1\\ &\quad +Tr\left(\Lambda(\Omega-\Psi)\right)+\frac{\rho}{2}\|\Omega-\Psi\|_F^2,\end{aligned}$$

where $\|\cdot\|_F$ is the Frobenius norm of a matrix and ρ is a parameter in the ADMM algorithm. See Section 4 for more details. Given the current estimate $\Omega^k,\Psi^k,\Lambda^k$, we update successively

$$\Omega^{k+1}=\arg\min_{\Omega\in R^{p\times p}}L(\Omega,\Psi^k,\Lambda^k), \tag{4}$$

$$\Psi^{k+1}=\arg\min_{\Psi\in R^{p\times p}}L(\Omega^{k+1},\Psi,\Lambda^k), \tag{5}$$

$$\Lambda^{k+1}=\Lambda^k+\rho(\Omega^{k+1}-\Psi^{k+1}). \tag{6}$$

Write $\hat{\Sigma}_i=U_iD_iU_i^T$ as the eigenvalue decomposition of $\hat{\Sigma}_i$ where $D_i=\mathrm{diag}(d_{i1},\cdots,d_{ip})$, $i=1,2$. Denote $A^k=(\hat{\Sigma}_1-\hat{\Sigma}_2)-\Lambda^k+\rho\Psi^k$ and organize the diagonals of $(D_2\otimes D_1+\rho I)^{-1}$ in a matrix B, where $B_{jk}=1/(d_{1j}d_{2k}+\rho)$. The following proposition provides explicit solutions for (4) and (5) which ensures efficient updation of our algorithm in each step.

Proposition 1 *Given Ψ^k, Λ^k, ρ and λ, the solution for (4) is given as:*

$$\Omega^{k+1}=U_1[B\circ(U_1^TA^kU_2)]U_2^T;$$

Given Ω^{k+1},Λ^k and ρ, the solution for (5) is given as:

$$\Psi^{k+1}=S(\Omega^{k+1}+\frac{\Lambda^k}{\rho},\frac{\lambda}{\rho}), \tag{7}$$

where S is known as the soft-thresholding operator on a matrix. Namely, the (i,j) entry of $S(A,b)$ for a matrix $A=(a_{ij})$ is $sign(a_{ij})(|a_{ij}|-b)_+$ where $(c)_+=c$ for $c>0$ and $(c)_+=0$ otherwise.

Note that for a given ρ, when updating Ω, we only need to update A^k which involves simple matrix subtraction, and then use matrix multiplication. Therefore the update in (4) can be efficiently implemented. Following is a brief derivation on how we obtain the explicit solutions given in Proposition 1. For (4), note that the derivative of L with respect to Ω is

$$\hat{\Sigma}_1\Omega\hat{\Sigma}_2-(\hat{\Sigma}_1-\hat{\Sigma}_2)+\Lambda^k+\rho(\Omega-\Psi^k)=(\hat{\Sigma}_1\Omega\hat{\Sigma}_2+\rho\Omega)-(\hat{\Sigma}_1-\hat{\Sigma}_2)+\Lambda^k-\rho\Psi^k,$$

which can be written as

$$(\hat{\Sigma}_2\otimes\hat{\Sigma}_1+\rho I)vec(\Omega)=vec\left((\hat{\Sigma}_1-\hat{\Sigma}_2)-\Lambda^k+\rho\Psi^k\right),$$

where vec is the vector operator. We have

$$vec(\Omega)=(\hat{\Sigma}_2\otimes\hat{\Sigma}_1+\rho I)^{-1}vec\left((\hat{\Sigma}_1-\hat{\Sigma}_2)-\Lambda^k+\rho\Psi^k\right).$$

Using the equality $vec(AXB)=(B^T\otimes A)vec(X)$, and

$$\begin{aligned}(\hat{\Sigma}_2\otimes\hat{\Sigma}_1+\rho I)^{-1} &= [(U_2\otimes U_1)(D_2\otimes D_1+\rho I)(U_2^T\otimes U_1^T)]^{-1}\\ &= (U_2\otimes U_1)(D_2\otimes D_1+\rho I)^{-1}(U_2^T\otimes U_1^T),\end{aligned}$$

we have

$$\begin{aligned}
& (\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 + \rho I)^{-1} vec(A^k) \\
= \; & [(U_2 \otimes U_1)(D_2 \otimes D_1 + \rho I)(U_2^T \otimes U_1^T)]^{-1} vec(A^k) \\
= \; & (U_2 \otimes U_1)(D_2 \otimes D_1 + \rho I)^{-1}(U_2^T \otimes U_1^T) vec(A^k) \\
= \; & (U_2 \otimes U_1)(D_2 \otimes D_1 + \rho I)^{-1} vec(U_1^T A^k U_2) \\
= \; & (U_2 \otimes U_1) vec\left(B \circ (U_1^T A^k U_2)\right) \\
= \; & vec\left(U_1[B \circ (U_1^T A^k U_2)]U_2^T\right)
\end{aligned}$$

where $\circ$ is the Hadamard product. Therefore,

$$\Omega = U_1[B \circ (U_1^T A^k U_2)]U_2^T.$$

Next we examine (5). Ignoring terms that are independent of Ψ, we just need to minimize

$$\frac{\rho}{2} Tr(\Psi^T \Psi) - \rho Tr((\Omega^{k+1})^T \Psi) - Tr((\Lambda^k)^T \Psi) + \lambda \|\Psi\|_1,$$

and the solution can be easily seen as (7). Again, the update for Γ can be efficiently implemented.

Our algorithm can be now summarized as following.

1. Initialize Ω, Ψ and Λ. Fix ρ. Compute SVD $\hat{\Sigma}_1 = U_1 D_1 U_1^T$ and $\hat{\Sigma}_2 = U_2 D_2 U_2^T$, and compute B where $B_{jk} = 1/(d_{1j} d_{2k} + \rho)$. Repeat steps 2-4 until convergence;

2. Compute $A = (\hat{\Sigma}_1 - \hat{\Sigma}_2) - \Lambda + \rho\Psi$. Then update Ω as $\Omega = U_1[B \circ (U_1^T A U_2)]U_2^T$;

3. Update Ψ by soft-thresholding $\Omega + \frac{\Lambda}{\rho}$ elementwise by $\frac{\lambda}{\rho}$;

4. Update Λ by $\Lambda \leftarrow \Lambda + \rho(\Omega - \Psi)$.

Note that the algorithm involves singular value decomposition of $\hat{\Sigma}_1$ and $\hat{\Sigma}_2$ only once. The rest of the algorithm only involves matrix addition and multiplication. Thus, the algorithm is extremely efficient. Compared with Zhao et al. (2014) whose algorithm has computational complexity of the order at least $O(p^4)$ and a memory requirement of $O(p^4)$, our algorithm has a memory requirement of the order $O(p^2)$ and computational complexity of $O(p^3)$. As a result, our method can handle much larger problems. As a first order method for convex problems, the convergence of ADMM algorithms is in general of rate $O(k^{-1})$, where k is the number of iterations. Convergence analysis of ADMM algorithms under different assumptions has been well established in some very recent optimization literatures; see for example, Nishihara et al. (2015), Hong and Luo (2017) and Chen et al. (2017). By verifying the assumptions in Hong and Luo (2017), we can established similar linear convergence results for our algorithm; see Lemma 1 in the Appendix for more details.

2.2 The linear index δ

After having estimated Ω as $\hat{\Omega}$, we discuss the estimation of the linear index $\delta = (\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2)$. We develop a procedure that avoids estimating Σ_1^{-1} and Σ_2^{-1}. First note that

$$\Sigma_1 \delta = \Sigma_1(\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2) = 2(\mu_1 - \mu_2) + \Sigma_1 \Omega(\mu_1 - \mu_2),$$
$$\Sigma_2 \delta = \Sigma_2(\Sigma_1^{-1} + \Sigma_2^{-1})(\mu_1 - \mu_2) = 2(\mu_1 - \mu_2) - \Sigma_2 \Omega(\mu_1 - \mu_2),$$

and

$$(\Sigma_1 + \Sigma_2)\delta = 4(\mu_1 - \mu_2) + (\Sigma_1 - \Sigma_2)\Omega(\mu_1 - \mu_2).$$

The last equation is the derivative of $\delta^T(\Sigma_1 + \Sigma_2)\delta/2 - \{4(\mu_1 - \mu_2) + (\Sigma_1 - \Sigma_2)\Omega(\mu_1 - \mu_2)\}^T \delta$. Motivated by this, we estimate δ by a direct method using the lasso regularization, similar to the one in Mai et al. (2012), as

$$\hat{\delta} = \arg\min_{\delta \in R^p} \; \frac{1}{2}\delta^T(\hat{\Sigma}_1 + \hat{\Sigma}_2)\delta - \hat{\gamma}^T \delta + \lambda_\delta \|\delta\|_1, \tag{8}$$

where $\hat{\gamma} = 4(\hat{\mu}_1 - \hat{\mu}_2) + (\hat{\Sigma}_1 - \hat{\Sigma}_2)\hat{\Omega}(\hat{\mu}_1 - \hat{\mu}_2)$, $\|\cdot\|_1$ is the vector ℓ_1 penalty and λ_δ is a tuning parameter. The optimization in (8) is a standard lasso problem and is easy to solve using existing lasso algorithms. We remark that (8) is much more challenging to analyze theoretically than the method in Mai et al. (2012), since the accuracy of $\hat{\Omega}$ as an estimator of Ω has to be carefully quantified in $\hat{\gamma}$.

We emphasize that our framework is extremely flexible and can accommodate additional constraints. As a concrete example, let's consider enforcing the so-called strong heredity principle in that an interaction is present unless the corresponding main effects are both present, i.e. if $\Omega_{jk} \neq 0$ then $\delta_j \neq 0$ and $\delta_k \neq 0$; see for example Hao and Zhang (2016). Denote $\mathcal{I} \subset \{1, \ldots, p\}$ as the set such that for any $j, k \in \mathcal{I}$ there exists some $\hat{\Omega}_{jk} \neq 0$. We can change the penalty in (8) as $\|\delta_{\mathcal{I}^C}\|_1$ such that the variables in $\mathcal{I}$ are not penalized. Due to space limitation, this line of research will not be studied in the current paper.

3. Theory

We show that our method can consistently select the true nonzero interaction terms in Ω and the true nonzero terms in δ. In addition, we provide explicit upper bounds for the estimation error under l_∞ norm. For classification, we further show that the misclassification rate of our DA-QDA rule converges to the optimal Bayes risk under some sparsity assumptions. For simplicity in this section we assume that $n_1 \asymp n_2$ and write $n = \min\{n_1, n_2\} - 1$. Instead of assuming $\mu_2 - \mu_1$, Σ_1, Σ_2 and $\Sigma_1 - \Sigma_2$ to be sparse as in Li and Shao (2015), we only assume that Ω and δ are sparse. For the estimation of Ω, the rate in Corollary 1 is similar to the one in Theorem 3 of Zhao et al. (2014). However, as we pointed out previously, our method is computationally much more efficient and scales better to large-dimensional problems. In addition, our work is the first direct estimation approach for sparse QDA. Importantly, the results for estimating δ are new.

Note that when estimating δ as in (8), we have used $\hat{\Omega}$ as a plug-in estimator for Ω. Consequently, from Corollaries 1 and 2, the error rate of $\hat{\delta}$ in estimating δ is a factor times of that of $\hat{\Omega}$ in estimating Ω. However, in the DA-QDA discriminant function defined as

in (1), $\hat{\Omega}$ appears in the first term which is a product of three components while $\hat{\delta}$ appears in the second term which is a product of two components. As a consequence, the overall estimation error rates of these two terms become equal. This implies that even though the estimating error of $\hat{\Omega}$ might aggregate in the estimation of δ, it does not affect the convergence rate of the classification error at all. Below we provide theory for estimating Ω, δ, and η, as well as quantifying the overall misclassification error rate.

3.1 Theory for estimating Ω

We first introduce some notation. We assume that $\Omega = (\Omega_{ij})_{1\le i,j\le p}$ is sparse with support $S = \{(i,j) : \Omega_{ij} \neq 0\}$ and we use S^c to denote the complement of S. Let d be the maximum node degree in Ω. For a vector $x = (x_1, \ldots, x_p)^T$, the l_q norm is defined as $|x|_q = (\sum_{i=1}^p |x_i|^q)^{1/q}$ for any $1 \le q < \infty$ and the l_∞ norm is defined as $|x|_\infty = \max_{1\le i\le p} |x_i|$. For any matrix $M = (m_{ij})_{1\le i,j\le p}$, its entrywise l_1 norm is defined as $||M||_1 = \sum_{1\le i,j\le p} |m_{ij}|$ and its entrywise l_∞ norm is written as $||M||_\infty = \max_{1\le i,j\le p} |m_{ij}|$. We use $|||M|||_{1,\infty} = \max_{1\le i\le p} \sum_{j=1}^p |m_{ij}|$ to denote the l_1/l_∞ norm induced matrix-operator norm. We denote $\Gamma = \Sigma_2 \otimes \Sigma_1$ and $\hat{\Gamma} = \hat{\Sigma}_2 \otimes \hat{\Sigma}_1$. Write $\Sigma_k = (\sigma_{kij})_{1\le i,j\le p}$, $\hat{\Sigma}_k = (\hat{\sigma}_{kij})_{1\le i,j\le p}$ for $k = 1, 2$. By the definition of Kronecker product, Γ is a $p^2 \times p^2$ matrix indexed by vertex pairs in that $\Gamma_{(i,j),(k,l)} = \sigma_{1ik}\sigma_{2jl}$. Denote $\Delta_i = \hat{\Sigma}_i - \Sigma_i$ for $i = 1, 2$, $\Delta_\Gamma = \hat{\Gamma} - \Gamma, \Delta_{\Gamma^T} = \hat{\Gamma}^T - \Gamma^T$, and $\epsilon_i = ||\Delta_i||_\infty$, $\epsilon = \max\{\epsilon_1, \epsilon_2\}$. $B = \max\{||\Sigma_1||_\infty, ||\Sigma_2||_\infty\}$, $B_\Sigma = \max\{||\Sigma_1||_{1,\infty}, ||\Sigma_2||_{1,\infty}\}$ and $B_\Gamma = ||\Gamma_{S,S}^{-1}||_{1,\infty}$, $B_{\Gamma^T} = ||(\Gamma_{S,S}^T)^{-1}||_{1,\infty}$, $B_{\Gamma,\Gamma^T} = \max\{B_\Gamma, B_{\Gamma^T}\}$.

To establish the model selection consistency of our estimator, we assume the following irrepresentability condition:

$$\alpha = 1 - \max_{e\in S^c} |\Gamma_{e,S}\Gamma_{S,S}^{-1}|_1 > 0.$$

This condition was first introduced by Zhao and Yu (2006) and Zou (2006) to establish the model selection consistency of the lasso. The following theorem gives the model selection consistency and the rate of convergence for the estimation of Ω.

Theorem 1 *Assume that $\alpha > 0$ and $d^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} \to 0$. For any $c > 2$, by choosing $\lambda = \kappa_1 d^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}$ for some large enough constant $\kappa_1 > 0$, we have with probability greater than $1 - p^{2-c}$,*

(i) $\hat{\Omega}_{S^c} = 0$;

(ii) there exists a large enough constant $\kappa_2 > 0$ such that

$$||\hat{\Omega} - \Omega||_\infty < \kappa_2 d^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}.$$

Theorem 1 states that if the irrepresentability condition is satisfied, the support of Ω is estimated consistently, and the rate of convergence of estimating Ω under l_∞ norm is of order $O\left(d^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}\right)$, which depends on the sparsity of Σ_1, Σ_2 and their Kronecker product. For example, our assumption $d^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} \to 0$ implies that

$B = \max\{||\Sigma_1||_\infty, ||\Sigma_2||_\infty\}$ can diverge in a rate of $o(d^{-1}B_\Sigma^{-1}B_{\Gamma,\Gamma^T}^{-1}(\frac{n}{\log p})^{1/4})$. From the proof of Theorem 1, we have the following corollary.

Corollary 1 *Assume that $\alpha > 0$, $B_\Sigma < \infty$ and $B_{\Gamma,\Gamma^T} < \infty$. For any constant $c > 2$, choosing $\lambda = Cd^2\sqrt{\frac{\log p}{n}}$ for some constant $C > 0$, if $d^2\sqrt{\frac{\log p}{n}} \to 0$, we have with probability greater than $1 - p^{2-c}$, $\hat{\Omega}_{S^c} = 0$ and*

$$||\hat{\Omega} - \Omega||_\infty = O\left(d^2\sqrt{\frac{\log p}{n}}\right).$$

Similar to Condition 2 in Zhao et al. (2014), the assumption $B_{\Gamma,\Gamma^T} < \infty$ in Corollary 1 is closely related to the mutual incoherence property introduced in Donoho and Huo (2001). In fact, it holds when imposing the usual mutual incoherence condition on the inverses of the submatrices (indexed by S) of Σ_1 and Σ_2. Since d is the maximum node degree in Ω, the number of nonzero entries in Ω is of order $O(dp)$. the rate $O\left(d^2\sqrt{\frac{\log p}{n}}\right)$ we obtained in Corollary 1 is better than the rate in Theorem 3 of Zhao et al. (2014). However, in the case where only O(d) covariates and some of their interactions are important, our rate is the same as the one in Zhao et al. (2014).

3.2 Theory for estimating δ

Let $D = \{i : \delta_i \neq 0\}$ be the support of δ and let d_δ be its cardinality. Denote $A_1 = ||\Omega||_{1,\infty}$, $A_2 = ||(\Omega^{-1})_{\cdot,D}||_{1,\infty}$, $\epsilon_\mu = \max\{|\mu_1 - \hat{\mu}_1|_\infty, |\mu_2 - \hat{\mu}_2|_\infty\}$. We define $A_\Sigma = ||\Sigma_{D,D}^{-1}||_{1,\infty}$ where $\Sigma = (\Sigma_1 + \Sigma_2)/2$ and write $\hat{\Sigma} = (\hat{\Sigma}_1 + \hat{\Sigma}_2)/2$, $\gamma = 4(\mu_1 - \mu_2) + (\Sigma_1 - \Sigma_2)\Omega(\mu_1 - \mu_2)$, $\Delta_\mu = \mu_1 - \mu_2$, $\hat{\Delta}_\mu = \hat{\mu}_1 - \hat{\mu}_2$, $A_\gamma = ||\gamma||_\infty$ and $||\hat{\Omega} - \Omega||_\infty = \epsilon_\Omega$.

To establish the model selection consistency of our estimator $\hat{\delta}$, we assume the following irrepresentability condition:

$$\alpha_\delta = 1 - \max_{e\in D^c} |\Sigma_{e,D}\Sigma_{D,D}^{-1}|_1 > 0.$$

Let $d_0 = \max\{d, d_\delta\}$. The following theorem gives the model selection consistency and the rate of convergence for the estimation of δ.

Theorem 2 *Assume that $\alpha_\delta > 0$, $|\Omega(\mu_1 - \mu_2)|_1 = O(d_0^2)$, $\|\Omega\|_\infty < \infty$, $A_\gamma < \infty$ and $d_0^3A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} \to 0$. Under the same assumptions in Theorem 1, for any $c > 2$, by choosing $\lambda_\delta = \kappa_3 d_0^3A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}$, for some large enough constant κ_3, we have, with probability greater than $1 - p^{2-c}$,*

(i) $\hat{\delta}_{D^c} = 0$;

(ii) there exists a large enough constant $\kappa_4 > 0$ such that,

$$||\hat{\delta} - \delta||_\infty < \kappa_4 d_0^3A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}.$$

From Theorem 2 and Corollary 1 we immediately have:

Corollary 2 *Suppose the assumptions of Corollary 1 and Theorem 2 hold and assume that* $A_\Sigma < \infty$. *For any constant* $c > 2$, *by choosing* $\lambda_\delta = Cd_0^3\sqrt{\frac{\log p}{n}}$ *for some large enough constant* $C > 0$, *we have with probability greater than* $1 - p^{2-c}$,

$$||\hat{\delta} - \delta||_\infty = O\left(d_0^3\sqrt{\frac{\log p}{n}}\right).$$

When $\Sigma_1 = \Sigma_2$, δ reduces to $2\Sigma_1^{-1}(\mu_1 - \mu_2)$, which is proportional to the direct discriminant variable β in Mai et al. (2012), Cai and Liu (2011) and Fan et al. (2012), and variables in $D = \{i : \delta_i \neq 0\}$ are linear discriminative features contributing to the Bayes rule. From the proof of Theorem 2 and the rate given in Theorem 2, we can see that when $A_\gamma < \infty$, $||\hat{\delta} - \delta||_\infty$ is of order $O(A_\Sigma\lambda_\delta)$. This is consistent to the result obtained in Theorem 1 of Mai et al. (2012) for the $\Sigma_1 = \Sigma_2$ case.

3.3 Misclassification rate

In this subsection, we study the asymptotic behavior of the misclassification rate for a given η and postpone the theory when η is estimated to Section 3.4. Let $R(i|j)$ and $R_n(i|j)$ be the probabilities that a new observation from class j is misclassified to class i by Bayes' rule and the DA-QDA rule respectively. Suppose $2\log(\pi_1/\pi_2) = \eta - \frac{1}{4}(\mu_1 - \mu_2)^T\Omega(\mu_1 - \mu_2) - \log|\Sigma_2| + \log|\Sigma_1|$. The optimal Bayes risk is given as

$$R = \pi_1 R(2|1) + \pi_2 R(1|2),$$

and the misclassification rate of the DA-QDA rule takes the following form:

$$R_n = \pi_1 R_n(2|1) + \pi_2 R_n(1|2).$$

Suppose $z_i \sim N(\mu_i, \Sigma_i)$ for $i = 1, 2$. Denote the density of $(z_i - \mu)^T\Omega(z_i - \mu) + \delta^T(z_i - \mu) + \eta$ as $F_i(z)$. For any constant c, define

$$u_c = \max\{\operatorname{ess\,sup}_{z\in[-c,c]}F_i(z), i = 1, 2\},$$

where ess sup denotes the essential supremum which is defined as supremum on almost everywhere of the support, i.e., except on a set of measure zero. Let $s := \|S\|_0$ be the number of nonzero elements in Ω. The following theorem establishes upper bounds for the misclassification rate difference between R_n and R.

Theorem 3 *Assuming that there exist constants* $C_\mu > 0, C_\Sigma > 1$ *such that* $\max\{|\mu_1|_\infty, |\mu_2|_\infty\} \leq C_\mu$, *and* $C_\Sigma^{-1} \leq \min\{\lambda_p(\Sigma_1), \lambda_p(\Sigma_2)\} \leq \max\{\lambda_1(\Sigma_1), \lambda_1(\Sigma_2)\} \leq C_\Sigma$ *where* $\lambda_i(\Sigma_j)$ *denotes the ith largest eigenvalue of* Σ_j. *Under the assumptions of Theorems 1 and 2, we have:*

(i) if $sd_0^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} + d_0^2A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} \to 0$ *and there exist positive constants* c, U_c *such that* $u_c \leq U_c < \infty$, *then*

$$R_n - R = O_p\left(sd_0^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} + d_0^2A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}\right);$$

(ii) if $\left(sd_0^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}} + d_0^2A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\sqrt{\frac{\log p}{n}}\right)(1 + u_c) \to 0$ *for some positive constant* c, *then with probability greater than* $1 - 3p^{2-c}$ *for some constant* $c > 2$,

$$R_n - R = O\left((1 + u_c) \times \left(sd_0^2B^2B_\Sigma^2B_{\Gamma,\Gamma^T}^2\log p\sqrt{\frac{\log p}{n}} + d_0^2A_\Sigma^2B^2B_\Sigma^3B_{\Gamma,\Gamma^T}^2\frac{\log p}{\sqrt{n}}\right)\right).$$

Theorem 3 (i) indicates that under appropriate sparsity assumptions, our DA-QDA rule is optimal in that its misclassification rate converges to the optimal Bayes risk in probability. The second statement of Theorem 3 states that under stronger conditions, R_n converges to R with overwhelming probability. From Corollary 1 and Corollary 2 and the above theorem, we immediately have:

Corollary 3 *Under the assumptions of Corollary 1, Corollary 2 and Theorem 3, we have,*

(i) if $sd_0^2\sqrt{\frac{\log p}{n}} \to 0$ *and there exist positive constants* c, U_c *such that* $u_c \leq U_c < \infty$, *then*

$$R_n - R = O_p\left(sd_0^2\sqrt{\frac{\log p}{n}}\right);$$

(ii) if $(1 + u_c)sd_0^2\sqrt{\frac{\log^3 p}{n}} \to 0$ *for some constant* $c > 0$, *then with probability greater than* $1 - 3p^{2-c}$ *for some constant* $c > 2$,

$$R_n - R = O\left(sd_0^2(1 + u_c)\sqrt{\frac{\log^3 p}{n}}\right).$$

We remark that the assumption $u_c \leq U_c$ for some constants c and U_c is similar to Condition (C4) in Li and Shao (2015), and our assumption is weaker in that we only assume the densities $F_i(z)$ is bounded in a neighborhood of zero while Condition (C4) in Li and Shao (2015) states that the densities are bounded everywhere.

3.4 Choice of η

The question of choosing the scalar η is critical for classification but receives little attention in existing literature; see Mai et al. (2012) for a detailed discussion for the LDA case. In this section, we propose to choose η by minimizing the in-sample misclassification error and establish analytical results for the estimation of η and the misclassification rate. With some abuse of notation, let (z_i, l_i) be our data where $z_i's$ are the covariates and $l_i's$ are the labels, i.e., $l_i \in \{0, 1\}$. To obtain $\hat{\eta}$, we seek to minimize the in-sample misclassification error given $\hat{\mu}, \hat{\delta}$ and $\hat{\Omega}$:

$$\hat{\eta} = arg\min_e \frac{1}{n}\sum_{i=1}^n |I\{\hat{D}(z_i, e) > 0\} - l_i|,$$

where $\hat{D}(z, e) = (z - \hat{\mu})^T\hat{\Omega}(z - \hat{\mu}) + \hat{\delta}^T(z - \hat{\mu}) + e$. We write $d(z) = (z - \mu)^T\Omega(z - \mu) + \delta^T(z - \mu)$ and $\hat{d}(z) = (z - \hat{\mu})^T\hat{\Omega}(z - \hat{\mu}) + \hat{\delta}^T(z - \hat{\mu})$ and hence we have

$$\hat{D}(z_i, e) = \hat{d}(z_i) + e.$$

Then the object function becomes

$$\frac{1}{n}\sum_{i=1}^{n}|I\{\hat{d}(z_i)+e>0\}-l_i|.$$

Without loss of generality, we can assume $\hat{d}(z_1)<\hat{d}(z_2)<\cdots<\hat{d}(z_n)$. For any e we define the index $k(e)$ to be the largest $\hat{d}(z_k)$ that satisfies $\hat{d}(z_k)<-e<\hat{d}(z_{k+1})$. Thus, the optimization can be further simplified as

$$\hat{\eta}=arg\min_{e}\frac{1}{n}\Big[\sum_{i=1}^{k(e)}l_i+\sum_{i=k(e)+1}^{n}(1-l_i)\Big].$$

Solving the above problem is simple. One just needs to compute the values of the object function for $k=0,1,2,\cdots,n$ and find the index k^* that minimizes its value. The optimal $\hat{\eta}$ can then be found as any value satisfying

$$\hat{\eta}\in(-\hat{d}(z_{k^*+1}),\ \ -\hat{d}(z_{k^*})).$$

Next we establish the asymptotic results for $\hat{\eta}$ and the misclassification rate. For a given e, we use $R(d,e)$ and $R(\hat{d},e)$ to denote the misclassification rate associated with discriminant function $D(z,e)=d(z)+e$ and discriminant function $\hat{D}(z,e)=\hat{d}(z)+e$ respectively. Analogously, the in-sample misclassification rate of $D(z,e)$ and $\hat{D}(z,e)$ are denoted as $R_n(d,e)$ and $R_n(\hat{d},e)$. From the optimality of the Bayes rule, we know that $\eta=2\log(\pi_1/\pi_2)+\frac{1}{4}(\mu_1-\mu_2)^T\Omega(\mu_1-\mu_2)+\log|\Sigma_2|-\log|\Sigma_1|$ is the unique minimizer of $R(d,\cdot)$ and we denote the corresponding optimal Bayes misclassification rate as $R=R(d,\eta)$. On the other hand, $\hat{\eta}$ is a minimizer of $R_n(\hat{d},e)$. In order to make the estimation problem feasible, we assume that there is exists a constant c_η such that $|\eta|\le c_\eta<\infty$. The following proposition indicates that, although the 0-1 loss used for computing the misclassification rate is neither continuous nor convex, the misclassification rate has a desirable property.

Proposition 2 *$R(d,e)$ is strictly monotone increasing in $e\in[\eta,\infty)$ and strictly monotone decreasing in $e\in(-\infty,\eta]$.*

From Proposition 2 and following Theorem 5.7 of Van der Vaart (2000), we establish the following theorem, which indicates that the estimator $\hat{\eta}$ is consistent and the resulting misclassification rate using the estimated rule $\hat{D}(z,\hat{\eta})$ tends to the optimal Bayes misclassification rate in probability.

Theorem 4 *Let $\hat{\eta}$ be a minimizer of $R_n(\hat{d},e)$. Under the assumptions of Theorem 3, we have:*

(i) $\hat{\eta}\to\eta$ in probability;

(ii) $R_n(\hat{d},\hat{\eta})\to R$ in probability.

4. Numerical Study

In this section, we provide extensive numerical evidence to show the empirical performance of DA-QDA by comparing it to its competitors, including the sparse QDA (sQDA, Li and Shao (2015)), the innovated interaction screening for sparse quadratic discriminant analysis (IIS-SQDA, Fan et al. (2015b)), penalized logistic regression with only main effects considered (PLR), penalized logistic regression with all interaction terms (PLR2), the direct approach for sparse LDA (DSDA, Mai et al. (2012)), the conventional LDA (LDA), the conventional QDA (QDA) and the oracle procedure (Oracle). The oracle procedure uses the true underlying model and serves as the optimal risk bound for comparison. We evaluate all methods via nine synthetic datasets and four real datasets. In addition, we also include L1-regularized SVM (L-SVM) and kernel SVM (K-SVM, with Gaussian kernel) performance as a benchmark in the real data analysis.

To fit DA-QDA, we employ ADMM to estimate Ω and the coordinate-wise descent algorithm (Friedman et al., 2010) to fit δ. Once Ω and δ are given, we then find the value of η by a simple linear search, minimizing the in-sample misclassification error. The rate parameter ρ in ADMM is set according to the optimal criterion suggested by Ghadimi et al. (2015). The other two tuning parameters, λ for estimating Ω and λ_δ for estimating δ, are chosen by 5-fold cross-validation, where the loss function is chosen to be the out-of-sample misclassification rate. To reduce the searching complexity, we are currently using a searching path rather than grid based tuning to avoid redundant computation and in-memory parallelism to distribute the computational tasks. It is worth noting that the calculation for each individual tuning pair (λ, λ_δ) can be made completely independent such that it is possible to distribute the entire calculation to multiple threads in a parallel fashion. One can also tune the parameters using the objective functions in (2) and (8) separately. However, we found that this strategy did not often lead to better classification results than tuning them jointly. This is possibly due to the complex shape of the misclassification surface as a function of these two tuning parameters. We implemented sQDA in Matlab with the leave-one-out-cross-validation (Li and Shao, 2015) to tune the three parameters. We employ Matlab's built-in function *fitcdiscr* to fit LDA and QDA and the R package *dsda* (Mai et al., 2012) to fit DSDA. For PLR, PLR2 and the second stage fit of IIS-SQDA which is a penalized logistic regression, we use the *glmnet* package and set $\alpha=0.5$ as the elastic net parameter. Other values of α was tried but did not change the result much. The other tuning parameter in *glmnet* is chosen by 10-fold cross-validation to minimize out-of-sample classification error. For the first stage of *IIS-SQDA* which is a screening step, we adopt the oracle-assisted approach proposed in Fan et al. (2015b), i.e., using the true Σ_1 and Σ_2 to compute the transformed variables used for screening as discussed in Fan et al. (2015b). To seek an appropriate screening size, we preserve the top $10, 30$ or 50 variables for each experiment to form interaction terms and report the best result (smallest misclassification error) for *IIS-SQDA*.

4.1 Synthetic data

For synthetic data, we use the same setup in Fan et al. (2015b). Observations are simulated from $N(u_1,\Sigma_1)$ and $N(u_2,\Sigma_2)$ where $u_2=0$. Recall $\Omega_1=\Sigma_1^{-1}$ and $\Omega_2=\Sigma_2^{-1}$. We set $u_1=\Sigma_1\beta$ for $\beta=(0.6,0.8,0,\cdots,0)^T$. We consider three different dimensions $p=50,\ 200,$ or 500 with $n_1=n_2=100$. The parameters Ω_1,Ω_2 and β are set as follows.

- Model 1: This model is Model 3 in Fan et al. (2015b) where Ω_1 is a band matrix with $(\Omega_1)_{ii}=1$ and $\Omega_{ij}=0.3$ for $|i-j|=1$. We set $\Omega_2=\Omega_1+\Omega$, where Ω is a symmetric

and sparse matrix with $\Omega_{10,\ 10} = -0.3758, \Omega_{10,\ 30} = 0.0616, \Omega_{10,\ 50} = 0.2037, \Omega_{30,\ 30} = -0.5482, \Omega_{30,\ 50} = 0.0286$, and $\Omega_{50,\ 50} = -0.4614$. The other 3 nonzero entries in the lower triangle of Ω are determined by symmetry.

- Model 2: We set $(\Omega_1)_{ij} = 0.5^{|i-j|}$ and let $\Omega_2 = \Omega_1 + \Omega$, where $\Omega = I_p$.
- Model 3: Ω_1 is the same as Model 2 and $\Omega_2 = \Omega_1$.
- Model 4: Ω_1 is the same as Model 2 and Ω is a band matrix defined as $(\Omega)_{ii} = 1$ and $(\Omega)_{ij} = 0.5$ for $|i-j| = 1$. Let $\Omega_2 = \Omega_1 + \Omega$.
- Model 5: $\Omega_1 = I_p$ and $\Omega_2 = \Omega_1 + \Omega$ where Ω is a random sparse symmetric matrix with conditional number 10 and non-zero density $n_1/p^2 \times 0.7$ (generated by *sprandsym* in Matlab).
- Models 6 and 7: These are Cases 9 and 10 in Srivastava et al. (2007). For Models 6 and 7, we generate the covariance by the following process: We first sample an $p \times p$ matrix R_1 where each entry is an i.i.d uniform random variable between $[0, 1]$. Then Ω_1 is defined to be $\Omega_1 = (R_1^T R_1)^{-1}$ which is a dense matrix. Matrix Ω_2 is generated similarly. For Model 6 the means are set as $u_1 = u_2 = 0$ while for Model 7, u_1 and u_2 are generated by random sampling over uniform distributions.
- Models 8 and 9: These are Cases 10 and 11 in Srivastava et al. (2007) where we generate the same $p \times p$ matrix R_1 as in Model 6. Then Ω_1 is defined to be $\Omega_1 = (R_1^T R_1 R_1^T R_1)^{-1}$ representing the ellipsoidal covariances, which often have only one strong eigenvalue and many relatively smaller ones. Matrix Ω_2 is similarly generated. The means are set as $u_1 = u_2 = 0$ for Model 8 and generated by random sampling for Model 9.

Model 1 is a model where Ω is a sparse two-block matrix after permutation. This is a model that favors IIS-SQDA. In Model 2, the difference between Ω_1 and Ω_2 is a diagonal matrix, and IIS-SQDA is expected to underperform as its screening step for identifying variables that are involved in interaction would retain all the variables. Model 3 is obviously a model that favors the linear discriminant analysis (LDA) as $\Omega = 0$, and in particular favors the sparse LDA (DSDA). This model is simulated to test whether methods designed for sparse QDA work satisfactorily in situations where LDA works the best. In Model 4, the difference matrix Ω is a tridiagonal matrix where the screening step of IIS-SQDA is expected to underperform. Finally, in Model 5, Ω admits a random sparse structure having $0.7n_1 = 70$ nonzero entries regardless of the dimension p. In Model 6 - 9, the covariance matrices are dense and so are Ω and δ. These cases are to test DA-QDA in scenarios where the sparse assumption fails to hold and the features are highly correlated. Model 6 and 8 are two difficult cases for linear methods as the means of the two classes are also the same. Our implementation of IIS-SQDA is applied only to Models 2 - 9 while the results for Model 1 are directly cited from Fan et al. (2015b).

For models 1 - 5, we simulate 100 synthetic datasets for each model and record for each method under comparison: 1) The misclassification rate (MR), 2) The false positives for main effects and interactions (FP.main for δ and FP.inter for Ω), 3) The false negatives for main effects and interactions (FN.main for δ and FN.inter for Ω). The results are

summarized in Tables 1, 2, 3, 4 and 5 for the five models under consideration. For the dense models, we compare the misclassification rates over 100 replications, and the results are summarized in Table 6.

From these tables, we can observe the following phenomena.

1. For Model 1 where the setup favors IIS-SQDA, IIS-SQDA performs the best in terms of variable selection. DA-QDA performs similarly in terms of the misclassification rate. These two methods outperform LDA and QDA (when $p = 50$), and PLR, PLR2, DSDA by a large margin in classification error.
2. For Model 2, as expected, IIS-SQDA is outperformed by DA-QDA by a large margin. Interestingly, PLR2 performs the second best. Linear classifiers including PLR and DSDA perform quite badly.
3. For Model 3, DSDA, IIS-SQDA and DA-QDA perform best and similarly. It is interesting that DA-QDA performs on par with DSDA even when the model clearly favors sparse linear classifiers.
4. For Model 4, DA-QDA outperforms all other methods again by a large margin.
5. For Model 5, DA-QDA performs the best and by a large margin when p becomes large.
6. For Models 6 - 9, the ordinary QDA performs the best for all low dimensional cases, which is as expected as the covariances of the the two classes are sufficiently different. When the dimension goes higher, DA-QDA achieves a high precision for some really difficult cases which is pretty surprising, considering the matrices are now all dense. The advantage mostly comes from that DA-QDA only imposes the sparse assumption on Ω and δ instead of the original precision matrices as for sQDA. In addition, DA-QDA performs better than IIS-SQDA.

To summarize, DA-QDA achieves the smallest misclassification rate in most examples and competitive performance in selecting main and interaction effects. IIS-SQDA is the preferred approach if Ω is a two-block diagonal matrix after permutation as is the case for Model 1. PLR2 generally performs better than (sparse) linear classifiers when interactions exist.

4.2 Real data

In this section, we investigate the performance of DA-QDA by analyzing four real data sets and compare it to the other classifiers discussed in the simulation study and we also include L1-regularized SVM (L-SVM) and kernel SVM (K-SVM, with Gaussian kernel) for more comprehensive comparison.

Quora answer classifier. This is a data challenge available at `http://www.quora.com/challenges#answer_classifier`. The training data set contains 4,500 answers from QUORA which have been annotated with either "good" or "bad". For each answer, 21 features (20 of which are effective) were extracted from the original sentences. The goal of this challenge is to automatically classify a new answer based on the 20 features. Since the dimension $p = 20$ is relatively small, we can compare DA-QDA to all the methods discussed

Table 1: The means and standard errors (in parentheses) of various performance measures by different classification methods for model 1 based on 100 replications

p	Method	MR (%)	FP.main	FP.inter	FN.main	FN.inter
50	LDA	39.43 (0.15)	–	–	–	–
	QDA	43.47 (0.10)	–	–	–	–
	PLR	36.12 (0.26)	5.95 (0.93)	–	1.21 (0.04)	–
	DSDA	35.05 (0.22)	8.81 (1.06)	–	0.07 (0.03)	–
	sQDA	27.64 (0.22)	11.17 (1.49)	–	0.33 (0.05)	–
	PLR2	30.15 (0.44)	0.51 (0.14)	11.26 (2.78)	0.60 (0.05)	2.62 (0.09)
	IIS-SQDA	27.56 (0.27)	5.60 (0.82)	2.16 (0.32)	0.19 (0.04)	2.05 (0.09)
	DA-QDA	26.50 (0.28)	0.85 (0.18)	35.26 (4.72)	0.39 (0.07)	3.74 (0.14)
	Oracle	23.04 (0.09)	–	–	–	–
200	PLR	37.62 (0.34)	7.82 (1.87)	–	1.47 (0.05)	–
	DSDA	36.34 (0.30)	15.06 (3.37)	–	0.36 (0.05)	–
	sQDA	26.80 (0.21)	12.75 (2.22)	–	0.47 (0.05)	–
	PLR2	32.55 (0.53)	0.25 (0.06)	17.44 (3.63)	0.90 (0.05)	2.72 (0.08)
	IIS-SQDA	26.94 (0.31)	6.43 (1.24)	0.78 (0.17)	0.42 (0.05)	2.22 (0.08)
	DA-QDA	26.51 (0.20)	0.29 (0.07)	25.48 (2.75)	0.82 (0.08)	4.14 (0.12)
	Oracle	21.93 (0.08)	–	–	–	–
500	PLR	38.82 (0.33)	9.31 (1.99)	–	1.58 (0.05)	–
	DSDA	37.10 (0.29)	16.06 (3.02)	–	0.42 (0.05)	–
	sQDA	28.22 (0.41)	24.22 (5.04)	–	0.58 (0.05)	–
	PLR2	35.45 (0.64)	0.34 (0.09)	55.69 (12.67)	0.99 (0.05)	3.05 (0.10)
	IIS-SQDA	26.78 (0.31)	3.22 (1.09)	0.23 (0.05)	0.98 (0.02)	2.65 (0.09)
	DA-QDA	26.68 (0.27)	0.14 (0.06)	10.96 (1.38)	1.02 (0.08)	4.36 (0.09)
	Oracle	21.81 (0.09)	–	–	–	–

Table 2: The means and standard errors (in parentheses) of various performance measures by different classification methods for model 2 based on 100 replications

p	Method	MR (%)	FP.main	FP.inter	FN.main	FN.inter
50	LDA	34.53 (0.19)	–	–	–	–
	QDA	32.09 (0.25)	–	–	–	–
	PLR	31.58 (0.20)	7.51 (0.55)	–	0.07 (0.03)	–
	DSDA	29.89 (0.16)	8.52 (0.86)	–	0.16 (0.04)	–
	sQDA	30.96 (0.90)	27.33 (1.95)	–	0.24 (0.05)	–
	PLR2	5.85 (0.10)	1.14 (0.11)	45.60 (1.08)	0.14 (0.04)	14.27 (0.33)
	IIS-SQDA	5.85 (0.10)	1.14 (0.11)	45.60 (1.08)	0.14 (0.04)	14.27 (0.32)
	DA-QDA	1.84 (0.08)	4.12 (0.49)	110.10 (10.54)	0.28 (0.05)	1.28 (0.22)
	Oracle	0.65 (0.02)	–	–	–	–
200	PLR	33.34 (0.21)	10.79 (0.70)	–	0.16 (0.04)	–
	DSDA	30.37 (0.23)	11.91 (2.19)	–	0.29 (0.05)	–
	sQDA	33.28 (0.58)	101.75 (7.72)	–	0.27 (0.05)	–
	PLR2	1.73 (0.06)	0.01 (0.01)	12.68 (0.56)	1.08 (0.05)	119.95 (0.52)
	IIS-SQDA	3.98 (0.10)	2.10 (0.15)	15.76 (0.60)	0.11 (0.04)	153.47 (0.31)
	DA-QDA	0.39 (0.18)	9.03 (2.12)	724.35 (19.52)	0.21 (0.04)	6.05 (0.35)
	Oracle	0 (0)	–	–	–	–
500	PLR	34.04 (0.24)	11.17 (1.02)	–	0.30 (0.05)	–
	DSDA	30.99 (0.22)	14.61 (2.64)	–	0.44 (0.05)	–
	sQDA	36.92 (0.64)	243.9 (21.2)	–	0.35 (0.05)	–
	PLR2	1.68 (0.06)	0 (0)	5.52 (0.33)	1.19 (0.05)	401.47 (0.59)
	IIS-SQDA	4.12 (0.09)	2.74 (0.25)	8.02 (0.43)	0.12 (0.04)	451.13 (0.29)
	DA-QDA	0.16 (0.22)	24.33 (2.18)	4.81e3 (290.1)	0.52 (0.05)	58.09 (1.10)
	Oracle	0 (0)	–	–	–	–

Table 3: The means and standard errors (in parentheses) of various performance measures by different classification methods for model 3 based on 100 replications

p	Method	MR (%)	FP.main	FP.inter	FN.main	FN.inter
50	LDA	38.82 (0.19)	–	–	–	–
	QDA	47.57 (0.11)	–	–	–	–
	PLR	36.06 (0.23)	7.73 (0.58)	–	0.14 (0.03)	–
	DSDA	34.82 (0.24)	9.54 (1.09)	–	0.26 (0.04)	–
	sQDA	41.52 (0.51)	14.89 (1.69)	–	0.44 (0.05)	–
	PLR2	37.36 (0.34)	0.60 (0.10)	31.10 (3.21)	0.39 (0.06)	0 (0)
	IIS-SQDA	35.10 (0.22)	5.25 (0.46)	10.85 (0.96)	0.06 (0.02)	0 (0)
	DA-QDA	34.99 (0.58)	0.82 (0.20)	23.84 (6.69)	0.35 (0.07)	0 (0)
	Oracle	31.68 (0.10)	–	–	–	–
200	PLR	38.50 (0.31)	12.90 (1.08)	–	0.23 (0.04)	–
	DSDA	36.27 (0.28)	14.81 (2.26)	–	0.41 (0.05)	–
	sQDA	43.82 (0.53)	53.18 (6.74)	–	0.52 (0.05)	–
	PLR2	40.31 (0.45)	0.15 (0.05)	40.38 (5.05)	0.74 (0.06)	0 (0)
	IIS-SQDA	36.32 (0.25)	25.39 (0.66)	6.03 (0.50)	0 (0)	0 (0)
	DA-QDA	36.55 (0.74)	1.70 (1.38)	37.15 (16.39)	0.89 (0.09)	0 (0)
	Oracle	31.54 (0.10)	–	–	–	–
500	PLR	39.98 (0.32)	14.79 (1.41)	–	0.40 (0.05)	–
	DSDA	37.07 (0.29)	19.49 (3.65)	–	0.59 (0.05)	–
	sQDA	46.00 (0.48)	130.91 (18.08)	–	0.57 (0.05)	–
	PLR2	42.23 (0.53)	0.03 (0.02)	36.6 (4.32)	1.07 (0.06)	0 (0)
	IIS-SQDA	37.45 (0.26)	14.53 (1.38)	3.70 (0.32)	0.07 (0.26)	0 (0)
	DA-QDA	37.95 (0.76)	0.2 (0.06)	57.49 (14.74)	1.05 (0.09)	0 (0)
	Oracle	31.85 (0.12)	–	–	–	–

Table 4: The means and standard errors (in parentheses) of various performance measures by different classification methods for model 4 based on 100 replications

p	Method	MR (%)	FP.main	FP.inter	FN.main	FN.inter
50	LDA	35.58 (0.20)	–	–	–	–
	QDA	35.40 (0.20)	–	–	–	–
	PLR	32.42 (0.23)	8.03 (0.57)	–	0.03 (0.01)	–
	DSDA	31.39 (0.21)	11.02 (1.13)	–	0.09 (0.03)	–
	sQDA	40.90 (0.46)	18.36 (1.93)	–	0.45 (0.05)	–
	PLR2	22.42 (0.21)	1.88 (0.16)	81.56 (2.26)	0.06 (0.03)	123.72 (0.36)
	IIS-SQDA	21.77 (0.20)	3.42 (0.21)	58.92 (1.86)	0 (0)	125.73 (0.32)
	DA-QDA	16.91 (0.27)	0.55 (0.14)	194.98 (11.31)	0.61 (0.08)	106.51 (0.83)
	Oracle	3.22 (0.04)	–	–	–	–
200	PLR	34.93 (0.28)	12.71 (0.88)	–	0.10 (0.03)	–
	DSDA	32.64 (0.26)	15.63 (2.14)	–	0.21 (0.04)	–
	sQDA	41.68 (0.54)	64.88 (7.33)	–	0.46 (0.05)	–
	PLR2	21.82 (0.20)	0.30 (0.05)	107.80 (2.32)	0.40 (0.05)	559.23 (0.63)
	IIS-SQDA	20.15 (0.19)	6.11 (0.31)	70.76 (1.76)	0 (0)	563.33 (0.38)
	DA-QDA	9.59 (0.19)	0.31 (0.08)	297.38 (25.33)	0.82 (0.09)	498.61 (1.49)
	Oracle	0.28 (0.02)	–	–	–	–
500	PLR	37.19 (0.32)	15.68 (1.27)	–	0.32 (0.04)	–
	DSDA	33.83 (0.30)	22.90 (3.54)	–	0.45 (0.05)	–
	sQDA	43.39 (0.48)	193.04 (20.32)	–	0.46 (0.05)	–
	PLR2	23.06 (0.23)	0.05 (0.02)	114.94 (2.34)	0.79 (0.05)	1455 (0.65)
	IIS-SQDA	19.07 (0.17)	12.86 (0.42)	57.44 (1.41)	0 (0)	1459 (0.34)
	DA-QDA	4.18 (0.13)	0.20 (0.04)	298.24 (20.8)	0.42 (0.07)	1315 (2.41)
	Oracle	0 (0)	–	–	–	–

Table 5: The means and standard errors (in parentheses) of various performance measures by different classification methods for model 5 based on 100 replications

p	Method	MR (%)	FP.main	FP.inter	FN.main	FN.inter
50	LDA	39.21 (0.20)	–	–	–	–
	QDA	46.41 (0.17)	–	–	–	–
	PLR	35.76 (0.26)	6.08 (0.43)	–	0.01 (0.01)	–
	DSDA	33.73 (0.25)	8.08 (0.99)	–	0.14 (0.04)	–
	sQDA	36.76 (0.27)	9.37 (1.57)	–	0.06 (0.02)	–
	PLR2	36.62 (0.39)	1.04 (0.13)	45.83 (3.99)	0.05 (0.02)	63.69 (0.39)
	IIS-SQDA	35.56 (0.29)	8.77 (0.50)	14.85 (0.83)	0 (0)	61.18 (0.26)
	DA-QDA	34.32 (0.53)	0.52 (0.12)	39.76 (6.47)	0.58 (0.08)	59.76 (0.64)
	Oracle	32.36 (0.25)	–	–	–	–
200	PLR	37.73 (0.34)	9.68 (0.89)	–	0.40 (0.03)	–
	DSDA	34.58 (0.35)	10.87 (2.44)	–	0.11 (0.03)	–
	sQDA	26.11 (0.27)	18.35 (4.79)	–	0.21 (0.04)	–
	PLR2	37.40 (0.44)	0.32 (0.06)	66.44 (5.47)	0.31 (0.06)	194.46 (0.35)
	IIS-SQDA	33.22 (0.28)	19.87 (0.93)	6.16 (0.41)	0 (0)	191.37 (0.10)
	DA-QDA	29.35 (0.41)	0.10 (0.05)	164.24 (73.3)	1.27 (0.07)	175.8 (0.96)
	Oracle	20.09 (0.27)	–	–	–	–
500	PLR	39.13 (0.33)	14.39 (1.29)	–	0.08 (0.03)	–
	DSDA	34.76 (0.25)	9.44 (1.77)	–	0.16 (0.04)	–
	sQDA	10.17 (0.16)	22.32 (6.88)	–	0.24 (0.05)	–
	PLR2	37.44 (0.52)	0.16 (0.05)	90.78 (6.06)	0.43 (0.06)	493.48 (0.41)
	IIS-SQDA	26.57 (0.23)	19.14 (0.57)	62.00 (1.56)	0 (0)	475.49 (0.20)
	DA-QDA	23.75 (0.49)	4.03 (2.91)	507.92 (225.36)	1.59 (0.06)	459.96 (1.96)
	Oracle	4.16 (0.08)	–	–	–	–

Table 6: The means and standard errors (in parentheses) of mis-classification rate (MR %) for models 6, 7, 8, 9 based on 100 replications

p	Method	Model 6	Model 7	Model 8	Model 9
50	LDA	49.86 (0.10)	21.91 (0.37)	49.69 (0.10)	32.41 (0.42)
	QDA	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	PLR	49.97 (0.07)	24.40 (0.38)	49.77 (0.12)	40.24 (0.79)
	DSDA	49.89 (0.11)	22.84 (0.39)	49.05 (0.23)	34.31 (0.50)
	sQDA	48.14 (0.37)	32.80 (0.69)	46.68 (0.35)	35.27 (0.68)
	PLR2	18.17 (0.24)	16.85 (0.25)	7.41 (0.14)	7.28 (0.12)
	IIS-SQDA	19.90 (0.29)	18.05 (0.25)	7.60 (0.13)	7.49 (0.12)
	DA-QDA	12.58 (0.23)	11.70 (0.28)	4.26 (0.14)	4.22 (0.14)
	Oracle	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
200	PLR	50.08 (0.09)	39.86 (0.39)	49.66 (0.16)	49.83 (0.08)
	DSDA	49.96 (0.11)	37.00 (0.28)	49.33 (0.18)	48.33 (0.34)
	sQDA	50.48 (0.15)	45.33 (0.45)	48.98 (0.26)	48.49 (0.20)
	PLR2	38.76 (0.43)	36.90 (0.48)	6.80 (0.13)	7.25 (0.10)
	IIS-SQDA	45.59 (0.37)	40.93 (0.44)	8.03 (0.11)	8.50 (0.14)
	DA-QDA	31.45 (0.42)	33.01 (0.50)	4.65 (0.88)	4.07 (0.67)
	Oracle	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
500	PLR	49.90 (0.06)	47.67 (0.26)	50.15 (0.09)	50.13 (0.10)
	DSDA	50.22 (0.21)	42.71 (0.14)	49.38 (0.09)	46.70 (0.16
	sQDA	50.17 (0.08)	50.33 (0.11)	49.85 (0.12)	49.83 (0.12)
	PLR2	46.32 (0.49)	45.58 (0.46)	5.66 (0.12)	7.47 (0.06)
	IIS-SQDA	49.33 (0.44)	46.61 (0.39)	7.37 (0.15)	9.43 (0.13)
	DA-QDA	39.67 (0.16)	41.88 (0.48)	2.43 (0.12)	2.27 (0.07)
	Oracle	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)

in the simulation via 10-fold cross-validation. In particular, we randomly split the data into ten parts, fit a model to the nine parts of the data, and report the misclassification error on the part that is left out. The average misclassification errors and the standard errors for various methods are in Table 7. Interestingly, LDA performs much better than QDA, suggesting that if we stop the analysis here, we might simply have preferred to use the linear classifier LDA. However, the story becomes different if sparse models are considered. In particular, PLR, PLR2, IIS-SQDA and DA-QDA all outperform the non-sparse models significantly with DA-QDA performing the best.

Table 7: Misclassification rate (%) for the Quora answer data under 10-fold cross-validation

Method	mean	standard error
DA-QDA	16.44	0.45
LDA	18.84	0.50
QDA	30.33	0.72
PLR	17.89	0.60
DSDA	19.11	0.56
sQDA	29.59	1.57
PLR2	17.56	0.71
IIS-SQDA	17.33	0.48
L-SVM	18.13	0.83
K-SVM	25.53	0.27

Gastrointestinal Lesions This dataset (P. Mesejo et al., 2016) contains the features extracted from a database of colonoscopic videos showing three types of gastrointestinal lesions, hyperplasic, adenoma and serrated adenoma. The original task is a multi-class classification problem, which is simplified to a binary classification task aiming at identifying adenoma. The data set contains 152 samples (76 original samples each with two different light conditions) and 768 features. We select the top 200 features with the largest absolute values of the two sample t statistics and perform a 10-fold corss-validation. The average misclassification errors and the standard errors are reported in Table 8. The data is predominated by the main effects as the logistic regression achieves the best with DA-QDA as the runner-up.

Pancreatic cancer RNA-seq data The dataset (Weinstein et al., 2013) is part of the RNA-Seq (HiSeq) PANCAN data set and is a random extraction of gene expressions of patients having different types of tumor: BRCA, KIRC, COAD, LUAD and PRAD. The dataset contains 801 patients and 20531 genes. In this task, we aim to distinguish BRCA against the other cancers. Similar to the previous study, we select 500 genes with the largest absolute values of the two sample t statistics for further analysis. Since most methods achieve 0 misclassification error in 10-fold cross-validation test, to increase the difficulty, we randomly split the dataset in two equal subsets, train on one subset and test on the other. We repeat this procedure 50 times to obtain the following Table 9. The L1 regularized SVM achieves the smallest misclassification error among all the methods. Similar as the previous dataset, the difference between cancers is dominated by main factors as the LDA

Table 8: Misclassification rate (%) for gastrointestinal lesions under 10-fold cross-validation

Method	mean	standard error
DA-QDA	33.33	0.00
LDA	—	—
QDA	—	—
PLR	30.67	2.04
DSDA	53.33	0.00
sQDA	40.00	0.00
PLR2	39.33	0.67
IIS-SQDA	40.00	1.02
L-SVM	44.00	1.85
k-SVM	46.67	0.00

has already achieved a surprisingly low misclassification rate (0.22%). The dataset also shows that DA-QDA can also perform well when the difference between the covariance matrices is small.

Table 9: Misclassification rate (%) for Pancreatic cancer RNA-seq data

Method	mean	standard error
DA-QDA	0.09	0.13
LDA	0.22	0.22
QDA	—	—
PLR	0.12	0.01
DSDA	0.62	0.22
sQDA	0.29	0.19
PLR2	0.09	0.01
IIS-SQDA	0.15	0.01
L-SVM	0.02	0.01
k-SVM	37.26	0.48

Prostate cancer Taken from `ftp://stat.ethz.ch/Manuscripts/dettling/prostate.rda`, this data contains genetic expression levels for $N = 6033$ genes of 102 individuals. The first 50 are normal control subjects while the rest are prostate cancer patients. More details of the data can be found in Singh et al. (2002), Dettling (2004) and Efron (2010). The goal is to identify genes that are linked with prostate cancer and predict potential patients and the difficulty of this task lies in the interactions among genes. The existence of interactions can often complicate the analysis and produce unreliable inference if they are ignored. For example, Figure 1 displays the pair of 118^{th} and 182^{th} gene. We can see the marginal distributions of each gene does not differ too much between the patients and the normal subjects (the middle and the right panels), suggesting that their main effects may not be important for distinguishing the two classes. In the left panel of Figure 1, however, we can identify some joint pattern that distinguishes the two groups. It can be seen that

most patients are allocated in the red triangle while most normal subjects are within the blue triangle, indicating the existence of some interaction effect that might be useful for classification.

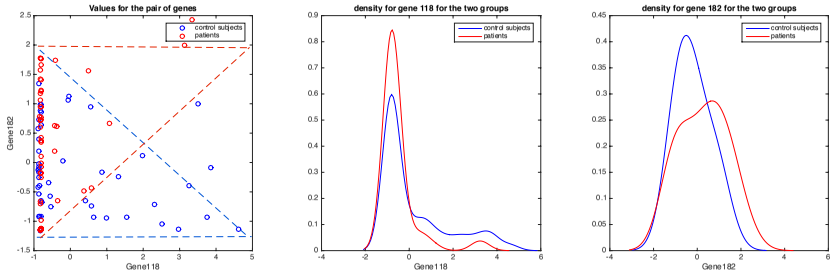


Figure 1: The plot for the gene 118 and gene 182. Left: joint scatter plot; Middle: marginal density of gene 118; Right: marginal density of gene 182.

For this data, we follow the same method in Cai and Liu (2011), retaining only the top 200 or 500 genes with the largest absolute values of the two sample t statistics. The average misclassification errors and the standard errors using 10-fold cross-validation for various methods are reported in Table 10. Note that since $p \gg n$, LDA and QDA were excluded. We can see again that DA-QDA is on par with L1 regularized SVM and outperforms all the other methods by a large margin, regardless of the number of the genes that were used for analysis.

Table 10: Misclassification rate (%) for the prostate cancer data under 10-fold cross-validation

	$p = 200$		$p = 500$	
Method	mean	std error	mean	std error
DA-QDA	0.00	0.00	1.00	1.00
LDA	—	—	—	—
QDA	—	—	—	—
PLR	11.00	2.45	16.00	2.92
DSDA	5.00	3.32	11.00	2.92
sQDA	0.00	0.00	2.00	2.00
PLR2	26.00	4.47	43.00	3.39
IIS-SQDA	11.00	2.92	18.00	2.74
L-SVM	0.00	0.00	1.00	1.00
k-SVM	48.00	9.82	63.00	2.55

5. Conclusion

We have proposed a novel method named DA-QDA for high-dimensional quadratic discriminant analysis. This is the first method aiming at directly estimating the quantities in the QDA discriminant function. The proposed framework is simple, fast to implement and enjoys excellent theoretical properties. We have demonstrated via extensive simulation and four data analyses that DA-QDA performs competitively under various circumstances.

We conclude by identifying three directions for future research. First, though the discussion of the paper is focused on binary problems, we can extend DS-QDA to handle multi-class problems as follows. When there are $k \geq 2$ classes, we can apply the DA-QDA approach to classes 1 and j, where $j = 2, \ldots, k$, in a pairwise manner. For a new sample z, denote the DA-QDA classifier between class 1 and class j as $\hat{D}_j(z)$ and suppose $D_i(z)$ is the smallest among $\{D_j(z), j = 2, \ldots, k\}$. By Bayes' rule, we can then classifier z into class i if $D_i(z) > 0$ and class 1 otherwise. Second, it is also interesting to see whether our theoretical results are optimal and in what sense. Finally, the proposed framework is extremely flexible. As a concrete example, if Ω is a two block sparse matrix after permutation as in Fan et al. (2015b), we can change the penalty $\|\Omega\|_1$ in (2) to one that encourages row sparsity, for example to $\|\Omega\|_{1,2} = \sum_{j=1}^{p} \|\Omega_{i,:}\|_2$ which is the sum of the ℓ_2 norms of the rows. It will be interesting to see how well this procedure compares with IIS-SQDA in Fan et al. (2015b). This topic is beyond the scope of the current paper and will be pursued elsewhere.

Acknowledgements

We would like to thank the two reviewers and the Action Editor Prof. Saharon Rosset for their valuable comments that have greatly improved the manuscript. Jiang's research is supported by the Hong Kong RGC grant (PolyU 253023/16P). Leng's research is partially supported by the Alan Turing Institute under the EPSRC grant EP/N510129/1.

Appendix A. Appendix A. Technical Lemmas and Proofs

A.1 Linear convergence of the ADMM algorithm

The following lemma establishes the linear convergence of our proposed ADMM algorithm in solving (3).

Lemma 1 *Given $\hat{\Sigma}_1, \hat{\Sigma}_2$ and λ, suppose that the ADMM scheme in (4)-(6) generates a solution sequence $\{\Omega^r, \Psi^r, \Lambda^r\}$. We have that $\{\Omega^r, \Psi^r\}$ converges linearly to an optimal solution of (3), and that $\|\Omega^r - \Psi^r\|_F$ converges linearly to zero.*

Proof Note that equation (3) is a special case of (1.1) of Hong and Luo (2017) with two blocks: $f(\Omega, \Psi) = f_1(\Omega) + f_2(\Psi)$ where, $f_1(\Omega) = \frac{1}{2} Tr\left(\Omega^T \hat{\Sigma}_1 \Omega \hat{\Sigma}_2\right) - Tr\left(\Omega(\hat{\Sigma}_1 - \hat{\Sigma}_2)\right)$ and $f_2(\Psi) = \lambda\|\Psi\|_1$. Note that $Tr\left(\Omega^T \hat{\Sigma}_1 \Omega \hat{\Sigma}_2\right) = vec(\Omega)^T(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1)vec(\Omega)$. Let $\hat{\Sigma}_2 \otimes \hat{\Sigma}_1 = U^T \Lambda U$ be the eigenvalue decomposition of the symmetric matrix $\hat{\Sigma}_2 \otimes \hat{\Sigma}_1$, and denote $A_1 := U^T \Lambda^{1/2} U$. Let $g_1(x) = x^T x$ be a function defined on $R^{p^2} \mapsto R$, and $h_1(x) = tr(x(\hat{\Sigma}_1 - \hat{\Sigma}_2))$, $h_2 = \lambda|x|_1$ be functions defined on $R^{p^2} \mapsto R$. We then have $f_1(\Omega) = g_1(A_1 vec(\Omega)) + h_1(vec(\Omega))$ and $f_2(\Psi) = h_2(vec(\Psi))$. Clearly given $\hat{\Sigma}_1, \hat{\Sigma}_2$ and λ, the gradient

of g_1 is uniformly Lipschitz continuous and h_1, h_2 are both polyhedral. The lemma follows immediately from Theorem 3.1 of Hong and Luo (2017). ■

A.2 Proofs of Theorem 1

We first introduce some technical lemmas and the proof of Theorem 1 will be given after these lemmas.

Lemma 2 *Suppose $\lambda_{\max}(\Sigma_k) < \epsilon_0 < \infty$ for $k = 1, 2$. There exist constants $C_0, C_1, C_2 > 0$ depending on ϵ_0 only, such that for any $|v| \leq C_0$,*

$$P(|\hat{\sigma}_{kij} - \sigma_{kij}| > v) \leq C_1 \exp(-C_2(n_k - 1)v^2) \leq C_1 \exp(-C_2 n v^2).$$

Proof Denote $X = (X_1, \ldots, X_{n_1})^T$. Let Λ be an orthogonal matrix with the last row being $(n^{-1/2}, \ldots, n^{-1/2})$ and define $Z = (z_1, \ldots, z_n) = \Lambda X$. We then have $z_1, \ldots, z_{n-1} \sim N(\mathbf{0}, \Sigma)$ and they are independent to each other. Note that $n_1 \hat{\Sigma}_1 = X^T(I_p - \mathbf{1}\mathbf{1}^T)X = Z^T \Lambda(I_p - \mathbf{1}\mathbf{1}^T)\Lambda^T Z = \sum_{i=1}^{n-1} z_i z_i^T$. This together with Lemma A.3 of Bickel and Levina (2008) prove Lemma 2. ■

Remark. Denote $\sigma^2 = \max\{\sigma_{1ii}, \sigma_{2ii}, i = 1, \ldots, p\}$. From Lemma 1 of Ravikumar et al. (2011) we can see that Lemma 2 is true for $C_1 = 4$, $C_2 = [128(1 + 4\sigma^2)^2\sigma^4]^{-1}$ and $v = 8(1 + 4\sigma^2)\sigma^2$.

Lemma 3 *Assume that,*

$$B_{\Gamma, \Gamma^T} < \frac{1}{3(d^2\epsilon_1\epsilon_2 + Bd^2(\epsilon_1 + \epsilon_2))}. \tag{9}$$

Let $R(\Delta_\Gamma) = (\Gamma_{S,S} + \Delta_\Gamma)^{-1} - \Gamma_{S,S}^{-1} + \Gamma_{S,S}^{-1}(\Delta_\Gamma)_{S,S}\Gamma_{S,S}^{-1}$. We then have

$$||R(\Delta_\Gamma)||_\infty \leq 3||(\Delta_\Gamma)_{S,S}||_\infty ||(\Delta_{\Gamma^T})_{S,S}||_{1,\infty} B_\Gamma B_{\Gamma^T}^2, \tag{10}$$

and

$$||R(\Delta_\Gamma)||_{1,\infty} \leq 3||(\Delta_\Gamma)_{S,S}||_{1,\infty} ||(\Delta_{\Gamma^T})_{S,S}||_{1,\infty} B_\Gamma B_{\Gamma^T}^2. \tag{11}$$

Moreover, we also have

$$||\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1}||_\infty \leq 3d^2(\epsilon^2 + 2B\epsilon)^2 B_\Gamma B_{\Gamma^T}^2 + (\epsilon^2 + 2B\epsilon)B_\Gamma^2, \tag{12}$$

$$||\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1}||_{1,\infty} \leq 3d^4(\epsilon^2 + 2B\epsilon)^2 B_\Gamma B_{\Gamma^T}^2 + d^2(\epsilon^2 + 2B\epsilon)B_\Gamma^2. \tag{13}$$

Proof Note that

$$\Delta_\Gamma = \Delta_2 \otimes \Delta_1 + \Delta_2 \otimes \Sigma_1 + \Sigma_2 \otimes \Delta_1.$$

Consequently by (9) we have $||\Gamma_{S,S}^{-1}||_{1,\infty}||(\Delta_\Gamma)_{S,S}||_{1,\infty} \leq 1/3$. (10) and (11) can then be proved using the same arguments as in Appendix B of Ravikumar et al. (2011). Note that

$$||(\Delta_\Gamma)_{S,S}||_\infty \leq \epsilon^2 + 2B\epsilon,$$

and

$$\max\{||(\Delta_\Gamma)_{S,S}||_{1,\infty}, ||(\Delta_{\Gamma^T})_{S,S}||_{1,\infty}\} \leq d^2(\epsilon^2 + 2B\epsilon),$$

we have

$$\begin{aligned} ||\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1}||_\infty &\leq ||R(\Delta_\Gamma)||_\infty + ||\Gamma_{S,S}^{-1}(\Delta_\Gamma)_{S,S}\Gamma_{S,S}^{-1}||_\infty \\ &\leq 3d^2(\epsilon^2+2B\epsilon)^2 B_\Gamma B_{\Gamma^T}^2 + ||(\Delta_\Gamma)_{S,S}||_\infty ||\Gamma_{S,S}^{-1}||_{1,\infty}^2 \\ &\leq 3d^2(\epsilon^2+2B\epsilon)^2 B_\Gamma B_{\Gamma^T}^2 + (\epsilon^2+2B\epsilon)B_\Gamma^2. \end{aligned}$$

This proves (12). (13) can be proved similarly. ■

Lemma 4 *Assume that (9) and the following assumptions hold:* $\alpha > 0$, $\epsilon < \min\left\{B, \frac{\alpha\lambda}{2(2-\alpha)}\right\}$ *and*

$$\begin{aligned} &3d^2\epsilon B B_{\Gamma,\Gamma^T}[1 + (B_\Sigma^2 + 3d^2\epsilon B B_{\Gamma,\Gamma^T})(9d^2\epsilon B B_{\Gamma,\Gamma^T}+1)B_{\Gamma,\Gamma^T}] \\ &\leq C_\alpha \alpha \min\{\lambda, 1\} \end{aligned} \tag{14}$$

where $C_\alpha = \frac{\alpha\lambda+2\epsilon\alpha-4\epsilon}{2B\alpha\lambda+\alpha\lambda+2\epsilon\alpha}$. *We have:*

(i) $vec(\hat{\Omega})_S = 0$.

(ii) $||\hat{\Omega} - \Omega||_\infty < 2\lambda B_{\Gamma,\Gamma^T} + 9d^2\epsilon B B_{\Gamma,\Gamma^T}^2(3d^2\epsilon B B_{\Gamma,\Gamma^T}+1)(2B+2\lambda)$.

Proof (i) Suppose $\tilde{\Omega}$ is the solution of:

$$\tilde{\Omega} = \min_{\Omega \in R^{p\times p}, \Omega_{S^c}=0} \frac{1}{2}Tr\left(\Omega^T\hat{\Sigma}_1\Omega\hat{\Sigma}_2\right) - Tr\left(\Omega(\hat{\Sigma}_1 - \hat{\Sigma}_2)\right) + \lambda\|\Omega\|_1. \tag{15}$$

We prove Lemma 4 (i) by showing that $\hat{\Omega} = \tilde{\Omega}$. Due to the convexity of (3) in the main paper, we only need to show that the derivative of (3) is zero at $\tilde{\Omega}$. Equivalently, we need to show that for any $1 \leq i, j \leq p$ we have,

$$|\hat{\Sigma}_1\tilde{\Omega}\hat{\Sigma}_2 - (\hat{\Sigma}_1 - \hat{\Sigma}_2)|_{i,j} \leq \lambda. \tag{16}$$

By taking the first derivative of (15) we obtain,

$$\{\hat{\Sigma}_1\tilde{\Omega}\hat{\Sigma}_2 - (\hat{\Sigma}_1 - \hat{\Sigma}_2) + \lambda Z\}_S = \mathbf{0}, \tag{17}$$

where $Z = (Z_{ij})_{1\leq i,j\leq p}$ with $Z_{ij} = 0$ for $(i,j) \in S^c$, $Z_{ij} = \text{sign}(\tilde{\Omega}_{ij})$ for $(i,j) \in S$ and $\tilde{\Omega}_{ij} \neq 0$, $Z_{ij} \in [-1,1]$ for $(i,j) \in S$ and $\tilde{\Omega}_{ij} = 0$. Therefore (16) is true for any $(i,j) \in S$. Using the vector operator, (17) becomes

$$\{(\hat{\Sigma}_2 \otimes \hat{\Sigma}_1)vec(\tilde{\Omega}) - vec(\hat{\Sigma}_1 - \hat{\Sigma}_2) + \lambda vec(Z)\}_S = \mathbf{0}.$$

Equivalently we have

$$vec(\tilde{\Omega})_S = \hat{\Gamma}_{S,S}^{-1}[vec(\hat{\Sigma}_1 - \hat{\Sigma}_2)_S - \lambda vec(Z)_S]. \tag{18}$$

Note that the left hand side of (16) equals $|\hat{\Sigma}_1\tilde{\Omega}\hat{\Sigma}_2 - \Sigma_1 + \Sigma_2 - (\Delta_1 - \Delta_2)|_{i,j}$. Using the vector operator and the fact that $\tilde{\Omega}_{S^c} = \mathbf{0}$, to show that (16) is true for any $e \in S^c$, we only need to show that

$$|\hat{\Gamma}_{e,S}vec(\tilde{\Omega})_S - \Gamma_{e,S}vec(\Omega)_S - vec(\Delta_1 - \Delta_2)_e| \leq \lambda. \tag{19}$$

Here we have use the fact that $\Gamma_{e,S}vec(\Omega)_S = vec(\Sigma_1 - \Sigma_2)_e$. By (18) and the fact that $vec(\Omega)_S = \Gamma_{S,S}^{-1}vec(\Sigma_1 - \Sigma_2)_S$ we have,

$$\begin{aligned} &|\hat{\Gamma}_{e,S}vec(\tilde{\Omega})_S - \Gamma_{e,S}vec(\Omega)_S - vec(\Delta_1 - \Delta_2)_e| \\ = \;& |\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}[vec(\hat{\Sigma}_1 - \hat{\Sigma}_2)_S - \lambda vec(Z)_S] \\ & -\Gamma_{e,S}\Gamma_{S,S}^{-1}vec(\Sigma_1 - \Sigma_2)_S - vec(\Delta_1 - \Delta_2)_e| \\ \leq \;& |[\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1} - \Gamma_{e,S}\Gamma_{S,S}^{-1}]vec(\Sigma_1 - \Sigma_2)_S| \\ & +|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}vec(\hat{\Sigma}_1 - \hat{\Sigma}_2 - \Sigma_1 + \Sigma_2)_S| + \|\Delta_1 - \Delta_2\|_\infty + \lambda|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1 \\ \leq \;& 2B|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1} - \Gamma_{e,S}\Gamma_{S,S}^{-1}|_1 + (\epsilon_1+\epsilon_2+\lambda)|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1 + \epsilon_1 + \epsilon_2. \end{aligned}$$

Consequently, (19) is true if

$$\begin{aligned} &\max_{e\in S^c} 2B|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1} - \Gamma_{e,S}\Gamma_{S,S}^{-1}|_1 + 2\epsilon(1 + |\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1) \leq (1-C_\alpha)\alpha\lambda, \\ &\max_{e\in S^c}|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1 \leq 1 - (1-C_\alpha)\alpha. \end{aligned} \tag{20}$$

Next we finish this proof by showing that (20) is true under the assumptions of this lemma.

By Lemma 3 we have for any $e \in S^c$,

$$\begin{aligned} &|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1} - \Gamma_{e,S}\Gamma_{S,S}^{-1}|_1 \\ \leq \;& |(\hat{\Gamma}_{e,S} - \Gamma_{e,S})\Gamma_{S,S}^{-1}|_1 + |\Gamma_{e,S}(\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1})|_1 \\ & +|(\hat{\Gamma}_{e,S} - \Gamma_{e,S})(\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1})|_1 \\ \leq \;& |\hat{\Gamma}_{e,S} - \Gamma_{e,S}|_1||\Gamma_{S,S}^{-1}||_{1,\infty} + ||\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1}||_{1,\infty}B_\Sigma^2 \\ & +|\hat{\Gamma}_{e,S} - \Gamma_{e,S}|_1||\hat{\Gamma}_{S,S}^{-1} - \Gamma_{S,S}^{-1}||_{1,\infty} \\ \leq \;& d^2(\epsilon^2+2B\epsilon)B_{\Gamma,\Gamma^T} + d^2[B_\Sigma^2 + d^2(\epsilon^2+2B\epsilon)] \\ & \times[3d^2(\epsilon^2+2B\epsilon)B_{\Gamma,\Gamma^T}+1](\epsilon^2+2B\epsilon)B_{\Gamma,\Gamma^T}^2 \\ \leq \;& 3d^2\epsilon B B_{\Gamma,\Gamma^T}[1 + (B_\Sigma^2 + 3d^2\epsilon B B_{\Gamma,\Gamma^T})(9d^2\epsilon B B_{\Gamma,\Gamma^T}+1)B_{\Gamma,\Gamma^T}] \\ \leq \;& C_\alpha\alpha\min\{\lambda,1\}. \end{aligned}$$

Consequently, $\max_{e\in S^c}|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1 \leq C_\alpha\alpha\min\{\lambda,1\} + (1-\alpha) \leq 1 - (1-C_\alpha)\alpha$, and

$$\begin{aligned} &\max_{e\in S^c} 2B|\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1} - \Gamma_{e,S}\Gamma_{S,S}^{-1}|_1 + 2\epsilon(1 + |\hat{\Gamma}_{e,S}\hat{\Gamma}_{S,S}^{-1}|_1) \\ \leq \;& 2BC_\alpha\alpha\lambda + 2\epsilon[2 - (1-C_\alpha)\alpha] \\ = \;& (1-C_\alpha)\alpha\lambda. \end{aligned}$$

(ii) From (i) we have $\hat{\Omega} = \tilde{\Omega}$. By (18) and the fact that $vec(\Omega)_S = \Gamma^{-1}_{S,S} vec(\Sigma_1 - \Sigma_2)_S$, we have

$$\begin{aligned} ||\hat{\Omega} - \Omega||_\infty &= |vec(\tilde{\Omega}) - vec(\Omega)|_\infty \\ &= |\hat{\Gamma}^{-1}_{S,S}[vec(\hat{\Sigma}_1 - \hat{\Sigma}_2)_S - \lambda vec(Z)_S] - \Gamma^{-1}_{S,S} vec(\Sigma_1 - \Sigma_2)_S|_\infty \\ &\leq \lambda||\hat{\Gamma}^{-1}_{S,S}||_{1,\infty} + |\hat{\Gamma}^{-1}_{S,S} vec(\Delta_1 - \Delta_2)_S|_\infty \\ &\quad + |(\hat{\Gamma}^{-1}_{S,S} - \Gamma^{-1}_{S,S}) vec(\Sigma_1 - \Sigma_2)|_\infty \\ &\leq (\lambda + 2\epsilon)||\hat{\Gamma}^{-1}_{S,S}||_{1,\infty} + 2B||\hat{\Gamma}^{-1}_{S,S} - \Gamma^{-1}_{S,S}||_{1,\infty} \\ &\leq (\lambda + 2\epsilon)||\Gamma^{-1}_{S,S}||_{1,\infty} + (2B + \lambda + 2\epsilon)||\hat{\Gamma}^{-1}_{S,S} - \Gamma^{-1}_{S,S}||_{1,\infty}. \end{aligned}$$

By (13) and the assumption that $2\epsilon < \alpha\lambda/(2-\alpha) < \alpha\lambda < \lambda$ we immediately have

$$\begin{aligned} & ||\hat{\Omega} - \Omega||_\infty \\ \leq\ & (\lambda + 2\epsilon)B_\Gamma + 3[d^4(\epsilon^2 + 2B\epsilon)^2 B_\Gamma B^2_{\Gamma^T} + d^2(\epsilon^2 + 2B\epsilon)B^2_\Gamma](2B + \lambda + 2\epsilon) \\ <\ & 2\lambda B_{\Gamma,\Gamma^T} + 9d^2\epsilon B B^2_{\Gamma,\Gamma^T}(3d^2\epsilon B B_{\Gamma,\Gamma^T} + 1)(2B + 2\lambda). \end{aligned}$$

■

Proof of Theorem 1

From Lemma 2 we have that with probability greater than $1 - p^{2-c}$, $\epsilon \leq \{(c\log p + \log C_1)/C_2 n\}^{1/2}$. With some abuse of notations, we denote $\epsilon = \{(c\log p + \log C_1)/C_2 n\}^{1/2}$. Choose

$$\lambda = \max\left\{8\alpha^{-1}, \frac{3(2-\alpha)(2B+1)}{1-\alpha} d^2 B B_{\Gamma,\Gamma^T}[1 + 2(B^2_\Sigma + 1/3)B_{\Gamma,\Gamma^T}]\right\} \times \sqrt{\frac{c\log p + \log C_1}{C_2 n}},$$

for some $c > 2$ and since $d^2 B^2 B^2_\Sigma B^2_{\Gamma,\Gamma^T}\sqrt{\frac{\log p}{n}} \to 0$ we can assume that the sample size n is large enough such that

$$\begin{aligned} n\ >\ & (c\log p + \log C_1) \times \max\{(C_2 \min(B^2, 1))^{-1}, 81 B^2 d^4 B^2_{\Gamma,\Gamma^T} C_2^{-1}, \\ & 9(C_2\alpha)^{-1} B^2 B^2_{\Gamma,\Gamma^T}[1 + 2(B^2_\Sigma + 1/3)B_{\Gamma,\Gamma^T}]^2\}. \end{aligned}$$

Clearly under the assumptions of Theorem 1 we have $\lambda = O\left(d^2 B^2 B^2_\Sigma B^2_{\Gamma,\Gamma^T}\sqrt{\frac{\log p}{n}}\right)$. We firstly verify that the assumptions in Lemmas 3 and 4 are true for the given λ, n and ϵ.

(i) By noticing that $\frac{8\epsilon}{\alpha} < \lambda$, $\frac{\alpha}{8} < \frac{\alpha}{4(2-\alpha)}$ and $n > (C_2 B^2)^{-1}(c\log p + \log C_1)$ we immediately have

$$\epsilon < \min\left\{B, \frac{\alpha\lambda}{4(2-\alpha)}\right\} < \min\left\{B, \frac{\alpha\lambda}{2(2-\alpha)}\right\}.$$

(ii) $n > 81 B^2 d^4 B^2_{\Gamma,\Gamma^T}(c\log p + \log C_1)/C_2$ implies $B_{\Gamma,\Gamma^T} < \frac{1}{9d^2 B\epsilon}$. Together with $\epsilon < B$ from (i) we can see that Assumption (9) holds.

(iii) Since $\epsilon < \frac{\alpha\lambda}{2(2-\alpha)}$, we have

$$C_\alpha > \frac{\alpha\lambda - 4\epsilon}{2B\alpha\lambda + \alpha\lambda} > \frac{\alpha\lambda - 2\alpha\lambda/(2-\alpha)}{2B\alpha\lambda + \alpha\lambda} = \frac{1-\alpha}{(2-\alpha)(2B+1)}.$$

Together with $B_{\Gamma,\Gamma^T} < \frac{1}{9d^2 B\epsilon}$, we have

$$\begin{aligned} & 3d^2\epsilon B B_{\Gamma,\Gamma^T}[1 + (B^2_\Sigma + 3d^2\epsilon B B_{\Gamma,\Gamma^T})(9d^2\epsilon B B_{\Gamma,\Gamma^T} + 1)B_{\Gamma,\Gamma^T}] \\ <\ & 3d^2\epsilon B B_{\Gamma,\Gamma^T}[1 + 2(B^2_\Sigma + 1/3)B_{\Gamma,\Gamma^T}] \\ \leq\ & \frac{1-\alpha}{(2-\alpha)(2B+1)}\lambda \\ <\ & C_\alpha\lambda. \end{aligned}$$

(14) is then true since $n > 9(C_2\alpha)^{-1} B^2 B^2_{\Gamma,\Gamma^T}[1 + 2(B^2_\Sigma + 1/3)B_{\Gamma,\Gamma^T}]^2(c\log p + \log C_1)$ implies

$$3\epsilon B B_{\Gamma,\Gamma^T}[1 + (B^2_\Sigma + 3d^2\epsilon B B_{\Gamma,\Gamma^T})(9d^2\epsilon B B_{\Gamma,\Gamma^T} + 1)B_{\Gamma,\Gamma^T}] \leq \alpha.$$

(i), (ii) and (iii) and Lemma 4 imply that

$$\begin{aligned} ||\hat{\Omega} - \Omega||_\infty &< 2\lambda B_{\Gamma,\Gamma^T} + 9d^2\epsilon B B^2_{\Gamma,\Gamma^T}(3d^2\epsilon B B_{\Gamma,\Gamma^T} + 1)(2B + 2\lambda) \\ &\leq 2\lambda B_{\Gamma,\Gamma^T} + 12 d^2\epsilon B B^2_{\Gamma,\Gamma^T}(2B + 2\lambda) \\ &= \frac{14}{3}\lambda B_{\Gamma,\Gamma^T} + 24 d^2\epsilon B^2 B^2_{\Gamma,\Gamma^T}. \end{aligned}$$

A.3 Proofs of Theorem 2

We first introduce some technical lemmas and the proof of Theorem 2 will be given after these lemmas.

Lemma 5 *Assume that $A_\Sigma d_\delta \epsilon < 1$ we have*

$$||\hat{\Sigma}^{-1}_{D,D} - \Sigma^{-1}_{D,D}||_{1,\infty} \leq \frac{A^2_\Sigma d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon}.$$

Proof This lemma can be easily proved using the following observation:

$$\begin{aligned} ||\hat{\Sigma}^{-1}_{D,D} - \Sigma^{-1}_{D,D}||_{1,\infty} &\leq ||\hat{\Sigma}^{-1}_{D,D}||_{1,\infty}||\hat{\Sigma}_{D,D} - \Sigma_{D,D}||_{1,\infty}||\Sigma^{-1}_{D,D}||_{1,\infty} \\ &\leq (A_\Sigma + ||\hat{\Sigma}^{-1}_{D,D} - \Sigma^{-1}_{D,D}||_{1,\infty}) d_\delta \epsilon A_\Sigma. \end{aligned}$$

■

Lemma 6

$$|\hat{\gamma} - \gamma|_\infty \leq 8\epsilon_\mu + 2(\epsilon + B_\Sigma \epsilon_\Omega A_2 + d\epsilon\epsilon_\Omega A_2)|\Omega(\mu_1 - \mu_2)|_1 + 2(B + \epsilon)(A_1 + d\epsilon_\Omega)\epsilon_\mu.$$

Proof

$$\begin{aligned}|\gamma-\hat{\gamma}|_\infty &\le 4|\hat{\Delta}_\mu-\Delta_\mu|_\infty+|(\Delta_1-\Delta_2)\Omega\Delta_\mu|_\infty+|(\hat{\Sigma}_1-\hat{\Sigma}_2)(\hat{\Omega}-\Omega)\Delta_\mu|_\infty\\ &\quad+|(\hat{\Sigma}_1-\hat{\Sigma}_2)\hat{\Omega}(\hat{\Delta}_\mu-\Delta_\mu)|_\infty.\end{aligned}$$

Lemma 6 can then be proved using the following facts:

$$|\hat{\Delta}_\mu-\Delta_\mu|_\infty\le 2\epsilon_\mu$$

$$|\mu_1-\mu_2|_1\le||(\Omega^{-1})_{\cdot,D}||_{1,\infty}|\Omega(\mu_1-\mu_2)|_1=A_2|\Omega(\mu_1-\mu_2)|_1;$$

$$|(\Delta_1-\Delta_2)\Omega\Delta_\mu|_\infty\le 2\epsilon|\Omega(\mu_1-\mu_2)|_1;$$

$$\begin{aligned}|(\hat{\Sigma}_1-\hat{\Sigma}_2)(\hat{\Omega}-\Omega)\Delta_\mu|_\infty &\le ||(\Sigma_1-\Sigma_2)(\hat{\Omega}-\Omega)||_\infty|\Delta_\mu|_1+||(\Delta_1-\Delta_2)(\hat{\Omega}-\Omega)||_\infty|\Delta_\mu|_1\\ &\le 2B_\Sigma\epsilon_\Omega|\mu_1-\mu_2|_1+2d\epsilon\epsilon_\Omega|\mu_1-\mu_2|_1;\end{aligned}$$

$$\begin{aligned}|(\hat{\Sigma}_1-\hat{\Sigma}_2)\hat{\Omega}(\hat{\Delta}_\mu-\Delta_\mu)|_\infty &\le 2(B+\epsilon)|\hat{\Omega}(\hat{\Delta}_\mu-\Delta_\mu)|_1\\ &\le 2(B+\epsilon)[||\Omega||_{1,\infty}|\hat{\Delta}_\mu-\Delta_\mu|_\infty+d\epsilon_\Omega|\hat{\Delta}_\mu-\Delta_\mu|_\infty]\\ &\le 2(B+\epsilon)(A_1+d\epsilon_\Omega)\epsilon_\mu.\end{aligned}$$

■

Proof of Theorem 2 Using similar arguments as in Lemma 2, there exists a constant $C_\epsilon>0$ such that $\max\{\epsilon,\epsilon_\mu\}\le C_\epsilon\{(c\log p+\log C_1)/C_{2\delta}n\}^{1/2}$. Similar to the proof of Theorem 1, we choose

$$\begin{aligned}\lambda_\delta &= \max\Big\{\frac{2(2-\alpha_\delta)C_\epsilon}{\alpha_\delta}[4+(2+B_\Sigma A_2)|\Omega(\mu_1-\mu_2)|_1+2B(A_1+C_3)],\\ &\quad\frac{d_\delta(5A_\Sigma+2B_\Sigma A_\Sigma^2)}{C_\delta\alpha_\delta}\times(C_3+1)\Big\}\sqrt{\frac{c\log p+\log C_1}{C_{2\delta}n}},\end{aligned}$$

where $C_3=\frac{14}{3}\max\Big\{\frac{8}{\alpha},\frac{3(2-\alpha)(2B+1)}{1-\alpha}d^2BB_{\Gamma,\Gamma^T}[1+2(B_\Sigma^2+\frac{1}{3})B_{\Gamma,\Gamma^T}]\Big\}B_{\Gamma,\Gamma^T}+24d^2B^2B_{\Gamma,\Gamma^T}^2$, and assume that n is large enough such that

$$\begin{aligned}n &> (c\log p+\log C_1)\times\max\{C_{2\delta}^{-1},2C_2^{-1}A_\Sigma^2d_\delta^2,C_2^{-1}(A_2+1)^2d^2,\\ &\quad C_2^{-1}C_\delta^{-2}\alpha_\delta^{-2}d_\delta^2(5A_\Sigma+2B_\Sigma A_\Sigma^2)^2\},\end{aligned}$$

where $0<C_\delta=\frac{\alpha_\delta\lambda_\delta-(2-\alpha_\delta)K\gamma}{\alpha_\delta\lambda_\delta(1+A_\gamma+K_\gamma)}<1$ and $C_{2\delta}=\min\{C_2,(2\sigma^2)^{-1}\}\times\min\{B^2,1\}$.

(i) Suppose $\tilde{\delta}$ is the solution of:

$$\tilde{\delta}=\min_{\delta\in R^p,\delta_{D^c}=0}\frac{1}{2}\delta^T(\hat{\Sigma}_1+\hat{\Sigma}_2)\delta-\hat{\gamma}^T\delta+\lambda_\delta\|\delta\|_1,$$

We first show that $\hat{\delta}=\tilde{\delta}$. It sufficies to show that for any $e\in D^c$,

$$|2\hat{\Sigma}_{e,D}\tilde{\delta}_D-\hat{\gamma}_e|\le\lambda_\delta.$$

By the definition of $\tilde{\delta}$ we have

$$\{2\hat{\Sigma}\tilde{\delta}-\hat{\gamma}+\lambda_\delta Z\}_D=\mathbf{0},$$

where $Z=(Z_1,\ldots,Z_p)^T$ with $Z_i=0$ for $i\in D^c$, $Z_i=\text{sign}(\tilde{\delta})$ for $i\in D$ and $\tilde{\delta}\neq 0$, $Z_i\in[-1,1]$ for $i\in D$ and $\tilde{\delta}=0$. Consequently, we have $\tilde{\delta}_D=\frac{1}{2}\hat{\Sigma}_{D,D}^{-1}(\hat{\gamma}_D-\lambda_\delta Z_D)$. Together with the fact that $\Sigma_{e,D}\Sigma_{D,D}^{-1}\gamma_D=2\Sigma_{e,D}\Sigma_{D,D}^{-1}\Sigma_{D,D}\delta_D=\gamma_e$, we have

$$\begin{aligned}|2\hat{\Sigma}_{e,D}\tilde{\delta}_D-\hat{\gamma}_e| &= |\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}(\hat{\gamma}_D-\lambda_\delta Z_D)-\hat{\gamma}_e|\\ &\le |(\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}-\Sigma_{e,D}\Sigma_{D,D}^{-1})\gamma_D|+|\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}(\hat{\gamma}_D-\gamma_D)|\\ &\quad+\lambda_\delta|\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}|_1+|\gamma_e-\hat{\gamma}_e|\\ &\le |(\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}-\Sigma_{e,D}\Sigma_{D,D}^{-1})|_1|\gamma_D|_\infty+\\ &\quad|\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}|_1|(\hat{\gamma}_D-\gamma_D)|_\infty+\lambda_\delta|\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}|_1+|\gamma_e-\hat{\gamma}_e|.\end{aligned}\tag{21}$$

For simplicity, in the following, inequalities will be derived without mentioning whether they hold "with probability greater than $1-p^{2-c}$". For example, since $n>2C_2^{-1}A_\Sigma^2d_\delta^2(c\log p+\log C_1)$, we have $A_\Sigma d_\delta\epsilon<1/2$ with probability greater than $1-p^{2-c}$ and we shall repeatedly use this inequality without mentioning it holds with probability greater than $1-p^{2-c}$. Since $n>C_{2\delta}^{-1}(c\log p+\log C_1)$, by (17) of Ravikumar et al. (2011) and Theorem 1, we also have $\epsilon_\Omega\le(C_3+1)\{(c\log p+\log C_1)/C_{2\delta}n\}^{1/2}:=\epsilon_0$.

From Lemma 5 we have,

$$\begin{aligned}&|\hat{\Sigma}_{e,D}\hat{\Sigma}_{D,D}^{-1}-\Sigma_{e,D}\Sigma_{D,D}^{-1}|_1\\ \le\ &|(\hat{\Sigma}_{e,D}-\Sigma_{e,D})\Sigma_{D,D}^{-1}|_1+|\Sigma_{e,D}(\hat{\Sigma}_{D,D}^{-1}-\Sigma_{D,D}^{-1})|_1+|(\hat{\Sigma}_{e,D}-\Sigma_{e,D})(\hat{\Sigma}_{D,D}^{-1}-\Sigma_{D,D}^{-1})|_1\\ \le\ &|\hat{\Sigma}_{e,D}-\Sigma_{e,D}|_1|\Sigma_{D,D}^{-1}|_{1,\infty}+|\Sigma_{e,D}|_1||\hat{\Sigma}_{D,D}^{-1}-\Sigma_{D,D}^{-1}||_{1,\infty}\\ &+|\hat{\Sigma}_{e,D}-\Sigma_{e,D}|_1||\hat{\Sigma}_{D,D}^{-1}-\Sigma_{D,D}^{-1})||_{1,\infty}\\ \le\ &d_\delta\epsilon A_\Sigma+d_\delta(B+\epsilon)||\hat{\Sigma}_{D,D}^{-1}-\Sigma_{D,D}^{-1}||_{1,\infty}\\ \le\ &d_\delta\epsilon A_\Sigma+\frac{d_\delta(B_\Sigma+d_\delta\epsilon)A_\Sigma^2\epsilon}{1-A_\Sigma d_\delta\epsilon}.\end{aligned}\tag{22}$$

Combining (21), (22) and Lemma 6 we have:

$$\begin{aligned}&|2\hat{\Sigma}_{e,D}\tilde{\delta}_D-\hat{\gamma}_e|\\ \le\ &\left\{\epsilon A_\Sigma+\frac{(B+d_\delta\epsilon)A_\Sigma^2\epsilon}{1-A_\Sigma d_\delta\epsilon}\right\}A_\gamma d_\delta+\left(2-\alpha_\delta+d_\delta\epsilon A_\Sigma+\frac{d_\delta(B_\Sigma+d_\delta\epsilon)A_\Sigma^2\epsilon}{1-A_\Sigma d_\delta\epsilon}\right)K_\gamma\\ &+\lambda_\delta\left(1-\alpha_\delta+d_\delta\epsilon A_\Sigma+\frac{d_\delta(B_\Sigma+d_\delta\epsilon)A_\Sigma^2\epsilon}{1-A_\Sigma d_\delta\epsilon}\right),\end{aligned}$$

where

$$K_\gamma=2\epsilon_\mu[4+(2+B_\Sigma A_2)|\Omega(\mu_1-\mu_2)|_1+2B(A_1+C_3)],$$

and we have used the fact that $\epsilon < B$, $d_\delta \epsilon A_2 < 1, d\epsilon < 1$ and hence $8\epsilon_\mu + 2(\epsilon + B_\Sigma \epsilon_\Omega A_2 + d\epsilon\epsilon_\Omega A_2)|\Omega(\mu_1 - \mu_2)|_1 + 2(B+\epsilon)(A_1 + d\epsilon_\Omega)\epsilon_\mu < K_\gamma$. Assume that

$$d_\delta \epsilon A_\Sigma + \frac{d_\delta(B_\Sigma + d_\delta\epsilon)A_\Sigma^2 \epsilon}{1 - A_\Sigma d_\delta \epsilon} \leq C_\delta \alpha_\delta \min\{\lambda_\delta, 1\}, \tag{23}$$

where $C_\delta = \frac{\alpha_\delta \lambda_\delta - (2-\alpha_\delta)K\gamma}{\alpha_\delta \lambda_\delta (1 + A_\gamma + K_\gamma)}$. It can be seen that $0 < C_\delta < 1$. We then have

$$\begin{aligned} |2\hat{\Sigma}_{e,D}\tilde{\delta}_D - \hat{\gamma}_e| &\leq \lambda_\delta A_\gamma C_\delta \alpha_\delta + (2-\alpha_\delta)K_\gamma + C_\delta \alpha_\delta \lambda_\delta K_\gamma + (1 - \alpha_\delta + C_\delta\alpha_\delta)\lambda_\delta \\ &= \lambda_\delta. \end{aligned}$$

Next we complete the Proof of this part by showing that (23) holds.

Since $n > C_2^{-1} C_\delta^{-2} \alpha_\delta^{-2} d_\delta^2 (5A_\Sigma + 2B_\Sigma A_\Sigma^2)^2 (c \log p + \log C_1)$, we have

$$d_\delta \epsilon A_\Sigma + \frac{d_\delta(B_\Sigma + d_\delta\epsilon)A_\Sigma^2 \epsilon}{1 - A_\Sigma d_\delta \epsilon} \leq d_\delta \epsilon A_\Sigma + 4 d_\delta A_\Sigma \epsilon + 2 B_\Sigma A_\Sigma^2 d_\delta \epsilon \leq C_\delta \alpha_\delta.$$

On the other hand, since $\lambda_\delta \geq \frac{d_\delta(5A_\Sigma + 2B_\Sigma A_\Sigma^2)}{C_\delta \alpha_\delta}\epsilon_0$, we have

$$d_\delta \epsilon A_\Sigma + \frac{d_\delta(B_\Sigma + \epsilon d_\delta)A_\Sigma^2 \epsilon}{1 - A_\Sigma d_\delta \epsilon} \leq \lambda_\delta C_\delta \alpha_\delta.$$

(ii) Use the fact that $A_\Sigma d_\delta \epsilon < 1/2$ we have:

$$\begin{aligned} |\hat{\delta}_D - \delta_D|_\infty &= \frac{1}{2}|\hat{\Sigma}_{D,D}^{-1}(\hat{\gamma}_D - \lambda_\delta Z_D) - \Sigma_{D,D}^{-1}\gamma_D| \\ &\leq |(\hat{\Sigma}_{D,D}^{-1} - \Sigma_{D,D}^{-1})\hat{\gamma}_D|_\infty + |\Sigma_{D,D}^{-1}(\hat{\gamma}_D - \gamma_D)|_\infty + \lambda_\delta |\hat{\Sigma}_{D,D}^{-1}|_{1,\infty} \\ &\leq \frac{A_\Sigma^2 d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon}(A_\gamma + |\hat{\gamma}_D - \gamma_D|_\infty) + A_\Sigma |\hat{\gamma}_D - \gamma_D|_\infty \\ &\quad + \lambda_\delta \left(A_\Sigma + \frac{A_\Sigma^2 d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon} \right) \\ &\leq \frac{A_\gamma A_\Sigma^2 d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon} + K_\gamma \left[\frac{A_\Sigma^2 d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon} + A_\Sigma \right] \\ &\quad + \lambda_\delta \left(A_\Sigma + \frac{A_\Sigma^2 d_\delta \epsilon}{1 - A_\Sigma d_\delta \epsilon} \right) \\ &\leq 2A_\gamma A_\Sigma^2 d_\delta \epsilon + 2K_\gamma A_\Sigma + 2\lambda_\delta A_\Sigma. \end{aligned}$$

This theorem is proved by plugging in $K_\gamma = 2\epsilon_\mu[4 + (2 + B_\Sigma A_2)|\Omega(\mu_1 - \mu_2)|_1 + 2B(A_1 + C_3)]$.

A.4 Proofs of Theorem 3

Proof (i) With some abuse of notations we write $d(z) = (z-\mu)^T \Omega (z-\mu) + \delta^T (z - \mu)$ and $\hat{d}(z) = (z - \hat{\mu})^T \hat{\Omega}(z - \hat{\mu}) + \hat{\delta}^T (z - \hat{\mu})$.

$$\begin{aligned} R_n(1|2) &= P(\hat{d}(z) + \eta > 0 | z \sim N(\mu_2, \Sigma_2)) \\ &= P(d(z) + \eta > d(z) - \hat{d}(z) | z \sim N(\mu_2, \Sigma_2)). \end{aligned}$$

Denote $z = (z_1, \ldots, z_p)^T$. Note that $z^T(\Omega - \hat{\Omega})z \leq \sum_{ij} |z_i z_j (\Omega - \hat{\Omega})_{ij}|$, by noticing that $E|z_i z_j| \leq (Ez_i^2 + Ez_j^2)/2 \leq C_\mu^2 + C_\Sigma$, we have $z^T(\Omega - \hat{\Omega})z = O_p(s||\Omega - \hat{\Omega}||_\infty)$. Using a similar argument for bounding $|z^T\Omega\mu - z\hat{\Omega}\hat{\mu}|$, $|\mu^T\Omega\mu - \hat{\mu}^T\hat{\Omega}\hat{\mu}|$, $(\delta - \hat{\delta})^T z$ and $\delta^T\mu - \hat{\delta}^T\hat{\mu}$ we obtain,

$$\begin{aligned} &|d(z) - \hat{d}(z)| \\ = \; & O_p\left(s d_0^2 B^2 B_\Sigma^2 B_{\Gamma,\Gamma^T}^2 \sqrt{\frac{\log p}{n}} + d_0^2 A_\Sigma^2 B^2 B_\Sigma^3 B_{\Gamma,\Gamma^T}^2 \sqrt{\frac{\log p}{n}} \right). \end{aligned} \tag{24}$$

From Assumption 1 and (24) and the mean value theorem, we have:

$$\begin{aligned} &R_n(1|2) - R(1|2) \\ = \; & \int_0^{d(z) - \hat{d}(z)} F_2(z) dz \\ = \; & O_p\left(s d_0^2 B^2 B_\Sigma^2 B_{\Gamma,\Gamma^T}^2 \sqrt{\frac{\log p}{n}} + d_0^2 A_\Sigma^2 B^2 B_\Sigma^3 B_{\Gamma,\Gamma^T}^2 \sqrt{\frac{\log p}{n}} \right). \end{aligned}$$

(i) is proved by noticing that the above equality is also true for $R_n(2|1) - R(2|1)$.

(ii) Let $\Phi(\cdot)$ be the cumulative distribution function of a standard normal random variable. We have any constant $C_z > 0$,

$$P(|z|_\infty > C_z \sqrt{\log p}) \leq p\left[1 - \Phi\left(\frac{C\sqrt{\log p} + C_\mu}{C_\Sigma^{1/2}} \right) \right].$$

From Lemma 11 of Liu et al. (2009) we have when p is large enough, by choosing $C_z > \sqrt{2(c-1)} C_\Sigma^{1/2}$,

$$p\left[1 - \Phi\left(\frac{C_z\sqrt{\log p} + C_\mu}{C_\Sigma^{1/2}} \right) \right] \leq p^{2-c}.$$

This together with Theorems 1 and 2 and the proof in (i), we have with probability greater than $1 - 3p^{2-c}$,

$$\begin{aligned} &|d(z) - \hat{d}(z)| \\ = \; & O_p\left(s d_0^2 B^2 B_\Sigma^2 B_{\Gamma,\Gamma^T}^2 \log p \sqrt{\frac{\log p}{n}} + d_0^2 A_\Sigma^2 B^2 B_\Sigma^3 B_{\Gamma,\Gamma^T}^2 \frac{\log p}{\sqrt{n}} \right). \end{aligned}$$

The rest of the proof is similar to that in the proof of (i). ■

A.5 Proof of Proposition 2

Proof Suppose $e_1 > e_2 \geq \eta$. Denote $c_i = |\Sigma_1\Sigma_2^{-1}|^{1/2}\exp\{\frac{1}{2}[e_i - \frac{1}{4}(\mu_1-\mu_2)^T\Omega(\mu_1-\mu_2)]\}$ for $i = 1, 2$. We have,

$$\begin{aligned} R(d, e_i) &= \pi_1 \int_{D(z,e_1)<0} f_1(z)dz + \pi_2 \int_{D(z,e_i)>0} f_2(z)dz \\ &= \int_{f_1(z)/f_2(z)<c_i^{-1}} \pi_1 f_1(z)dz + \int_{f_1(z)/f_2(z)>c_i^{-1}} \pi_2 f_2(z)dz. \end{aligned}$$

Since $e_1 > e_2 \geq \eta$, it can be easily shown that $c_1 > c_2 \geq \pi_1/\pi_2$. Consequently we have

$$R(d, e_1) - R(d, e_2) = -\int_{c_1^{-1}<f_1(z)/f_2(z)<c_2^{-1}} [\pi_1 f_1(z) - \pi_2 f_2(z)]dz > 0.$$

Therefore, $R(d, e)$ is strictly monotone increasing on $e \in [\eta, \infty)$. The second statement can be similarly proved. ■

A.6 Proofs of Theorem 4

We first introduce some technical lemmas and the proof of Theorem 4 will be given after these lemmas.

For any constant c, define $l_c = \min\{\text{ess}\inf_{z\in[-c,c]} F_i(z), i = 1, 2\}$.

Lemma 7 *For any constant $c > 0$, we have for any $-c \leq \epsilon_\eta \leq c$, $R(d, \eta + \epsilon_\eta) - R(d, \eta) \leq \pi_2 u_c|\epsilon_\eta|$ and $R(d, \eta + \epsilon_\eta) - R(d, \eta) \geq \epsilon_\eta^2 \exp(-c/2)\pi_2 l_c/4$.*

Proof Let's consider $0 \leq \epsilon_\eta \leq c$ first. Note that

$$R(d, \eta + \epsilon_\eta) - R(d, \eta) = \int_{-\epsilon_\eta<D(z,\eta)<0} [\pi_2 f_2(z) - \pi_1 f_1(z)]dz.$$

We have

$$R(d, \eta + \epsilon_\eta) - R(d, \eta) \leq \int_{-\epsilon_\eta<D(z,\eta)<0} \pi_2 f_2(z)dz = \pi_2 \int_{-\epsilon_\eta}^{0} F_2(z)dz \leq \pi_2 u_c \epsilon_\eta.$$

By noticing that $1 - \exp(-x/2) - x\exp(-c/2)/2$ is an increasing function in $[0, c]$ we have

$$\begin{aligned} R(d, \eta + \epsilon_\eta) - R(d, \eta) &\geq \int_{-\epsilon_\eta<D(z,\eta)<-\epsilon_\eta/2} [\pi_2 f_2(z) - \pi_1 f_1(z)]dz \\ &\geq \int_{-\epsilon_\eta<D(z,\eta)<-\epsilon_\eta/2} [1 - \exp(-\epsilon_\eta/2)]\pi_2 f_2(z)dz \\ &\geq \int_{-\epsilon_\eta<D(z,\eta)<-\epsilon_\eta/2} \epsilon_\eta \exp(-c/2)\pi_2 f_2(z)/2dz \\ &\geq \epsilon_\eta^2 \exp(-c/2)\pi_2 l_c/4. \end{aligned}$$

The lemma is then proved using a same argument as above for $-c \leq \epsilon_\eta < 0$. ■

Clearly 7 holds when c is set to be c_η. Noted from the proof of Lemma 7 that the bounds do not depend on η, we can claim that the bounds holds uniformly in $\epsilon_\eta \in [-c_\eta, c_\eta]$. Similarly, it can be shown that:

Lemma 8 *Suppose $\hat{d} - d = O_p(\Delta_d)$ with $\Delta_d \to 0$, we have $R(\hat{d}, e) - R(d, e) = O_p(\Delta_d u_c)$ uniformly in $e \in [-c_\eta, c_\eta]$.*

Lemma 9 *Under the assumptions of Theorem 3, $R_n(\hat{d}, e) \to R(d, e)$ in probability uniformly in $e \in [-c_\eta, c_\eta]$.*

Proof Denote all the samples in the two classes as $\{z_i, i = 1, \ldots, n_1 + n_2\}$ and denote the estimator obtained by leaving the ith sample out as $\hat{d}_{-i}$. Similarly we use $\hat{d}_{-(i,j)}$ to denote the estimator obtained by leaving the ith and jth samples out. From (24), we immediately have that for any $e \in [-c_\eta, c_\eta]$,

$$\begin{aligned} & EI\{\hat{d}(z_i) + e > 0\} - EI\{\hat{d}_{-i}(z_i) + e > 0\} \\ = \; & EI\{d(z_i) + e > 0\} - EI\{d(z_i) + e > 0\} + o(1) \\ = \; & o(1). \end{aligned}$$

Together with Lemma 8 we have

$$ER_n(\hat{d}, e) - R(d, e) = ER(\hat{d}_{-i}, e) + o(1) - R(d, e) \to 0, \tag{25}$$

uniformly in $e \in [-c_\eta, c_\eta]$. Note that

$$Var(I\{\hat{d}(z_i) + e > 0\}) \leq \frac{1}{4},$$

and for any $(i, j) \in \{(k, l) : 1 \leq k, l \leq n_1 + n_2, i \neq j\}$,

$$\begin{aligned} & Cov(I\{\hat{d}(z_i) + e > 0\}, I\{\hat{d}(z_j) + e > 0\}) \\ = \; & Cov(I\{\hat{d}_{-(i,j)}(z_i) + e > 0\}, I\{\hat{d}_{-(i,j)}(z_j) + e > 0\}) + o(1) \\ = \; & Cov(I\{d(z_i) + e > 0\}, I\{d(z_j) + e > 0\}) + o(1), \end{aligned}$$

where the last step can be obtained using (24) and Lemma 8 and the $o(1)$ term does not depend on e. Since z_i, z_j are independent, we immediately have

$$Var(R_n(\hat{d}, e)) \to 0, \tag{26}$$

uniformly in $e \in [-c_\eta, c_\eta]$. The lemma is then proved by Markov's inequality and the uniform convergence of the bias (25) and the variance (26) of $R_n(\hat{d}, e)$. ■

Proof of Theorem 4
The result that $\hat{\eta} \to \eta$ can be obtained by Proposition 2, Lemma 9 and Theorem 5.7 of Van der Vaart (2000). The second statement immediately follows from Theorem 3.

References

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*, 3rd ed. Wiley Series in Probability and Statistics. Wiley, New York, 2003.

P. Bickel and E. Levina. Some theory for fisher's linear discriminant function, 'naive bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10:989–1010, 2004.

P. Bickel and E. Levina. Covariance regularization by thresholding. *The Annals of Statistics*, 36:199–227, 2008.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3:1–122, 2011.

T. Cai and W. Liu. A direct estimation approach to sparse linear discriminant analysis. *Journal of the American Statistical Association*, 106:1566–1577, 2011.

L. Chen, D. Sun, and K. C. Toh. A note on the convergence of ADMM for linearly constrained convex optimization problems. *Computational Optimization and Applications*, 66:327–343, 2017.

M. Dettling. BagBoosting for tumor classification with gene expression data. *Bioinformatics*, 20:3583–3593, 2004.

D. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. *IEEE Transactions on Information Theory*, 47:2845–2862, 2001.

B. Efron. *Large-scale Inference: Empirical Bayes Methods for Estimation, Testing, and Prediction*. Cambridge University Press, 2010.

J. Fan, Y. Feng, and X. Tong. A ROAD to classification in high dimensional space. *Journal of the Royal Statistical Society Series B*, 74:745–771, 2012.

J. Fan, T. Ke, H. Liu, and L. Xia. QUADRO: A supervised dimension reduction method via Rayleigh quotient optimization. *The Annals of Statistics*, 43:1493–1534, 2015.

Y. Fan, Y. Kong, D. Li, and Z. Zheng. Innovated interaction screening for high-dimensional nonlinear classification. *The Annals of Statistics*, 43:1243–1272, 2015.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33:1, 2010.

E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems. *IEEE Transactions on Automatic Control*, 60:644–658, 2015.

N. Hao and H. H. Zhang. Interaction screening for ultra-high dimensional data. *Journal of the American Statistical Association*, 109:1285–1301, 2014.

N. Hao and H. H. Zhang. A note on high dimensional linear regression with interactions. *The American Statistician*, to appear, 2016.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. Springer, 2009.

M. Hong and Z. Luo. On the linear convergence of the alternating direction method of multipliers. *Mathematical Programming*, 162:165–199, 2017.

B. Jiang, Z. Chen, and C. Leng. Dynamic linear discriminant analysis for high-dimensional data. *Manuscript*, 2015.

C. Leng. Sparse optimal scoring for multiclass cancer diagnosis and biomarker detection using microarray data. *Computational Biology and Chemistry*, 32:417–425, 2008.

Q. Li and J. Shao. Sparse quadratic discriminant analysis for high dimensional data. *Statistica Sinica*, 25:457–473, 2015.

H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295-2328, 2009.

Q. Mai, Y. Yang, and H. Zou. Multiclass sparse discriminant analysis. *arXiv preprint arXiv:1504.05845*, 2015.

Q. Mai and H. Zou. A note on the equivalence of three sparse linear discriminant methods. *Technometrics*, 55:243–246, 2013.

Q. Mai, H. Zou, and M. Yuan. A direct approach to sparse discriminant analysis in ultra-high dimensions. *Biometrika*, 99:29–42, 2012.

R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, 2015. A general analysis of the convergence of ADMM. *arXiv preprint arXiv:1502.02009*, 2015.

R. Pan, H. Wang, and R. Li. Ultrahigh dimensional multi-class linear discriminant analysis by pairwise sure independence screening. *Journal of the American Statistical Association*, 111:169–179, 2016.

P. Mesejo, D, Pizarro, A. Abergel, O. Rouquette, S. Beorchia, L. Poincloux, and A. Bartoli. Computer-Aided Classification of Gastrointestinal Lesions in Regular Colonoscopy. *in IEEE Transactions on Medical Imaging*, 35(9):2051–2063, 2016.

P. Ravikumar, M. Wainwright, G. Raskutti, and B. Yu. High-dimensional covariance estimation by minimizing l1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

J. Shao, Y. Wang, X. Deng, and S. Wang. Sparse linear discriminant analysis by thresholding for high dimensional data. *The Annals of statistics,* 39:1241–1265, 2011.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd et al. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002.

S. Srivastava, M. R. Gupta, B. A. Frigyik. Bayesian Quadratic Discriminant Analysis. *Journal of Machine Learning Research*, 8:1277–1305, 2007

J. Sun and H. Zhao. The application of sparse estimation of covariance matrix to quadratic discriminant analysis. *BMC Bioinformatics*, 16:48, 2015.

A. W. Van der Vaart. *Asymptotic statistics*, Vol. 3. Cambridge University Press, 2000.

J. N. Weinstein, E. A. Collisson, G. B. Mills, K. R. M. Shaw5, B. A. Ozenberger, K. Ellrott, I. Shmulevich, C. Sander and J. M. Stuart. The cancer genome atlas pan-cancer analysis project. *Nature genetics* 45(10): 1113-1120, 2013.

D. M. Witten and R. Tibshirani. Penalized classification using Fisher's linear discriminant. *Journal of the Royal Statistical Society Series B*, 73:753–772, 2011.

T. Zhang and H. Zou. Sparse precision matrix estimation via lasso penalized D-trace loss. *Biometrika*, 101:103–120, 2014.

S. Zhao, T. Cai, and H. Li. Direct estimation of differential networks. *Biometrika*, 101:253–268, 2014.

P. Zhao and B. Yu. On model selection consistency of Lasso. *The Journal of Machine Learning Research*, 7:2541–2563, 2006.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101:1418–1429, 2006.

Journal of Machine Learning Research 19 (2018) 1-29 Submitted 9/17; Revised 3/18; Published 8/18

Distribution-Specific Hardness of Learning Neural Networks

Ohad Shamir OHAD.SHAMIR@WEIZMANN.AC.IL
Weizmann Institute of Science, Rehovot, Israel

Editor: Amir Globerson

Abstract

Although neural networks are routinely and successfully trained in practice using simple gradient-based methods, most existing theoretical results are negative, showing that learning such networks is difficult, in a worst-case sense over all data distributions. In this paper, we take a more nuanced view, and consider whether specific assumptions on the "niceness" of the input distribution, or "niceness" of the target function (e.g. in terms of smoothness, non-degeneracy, incoherence, random choice of parameters etc.), are sufficient to guarantee learnability using gradient-based methods. We provide evidence that neither class of assumptions alone is sufficient: On the one hand, for any member of a class of "nice" target functions, there are difficult input distributions. On the other hand, we identify a family of simple target functions, which are difficult to learn even if the input distribution is "nice". To prove our results, we develop some tools which may be of independent interest, such as extending Fourier-based hardness techniques developed in the context of statistical queries (Blum et al., 1994), from the Boolean cube to Euclidean space and to more general classes of functions.

Keywords: neural networks, computational hardness, distributional assumptions, gradient-based methods

1. Introduction

Artificial neural networks have seen a dramatic resurgence in recent years, and have proven to be a highly effective machine learning method in computer vision, natural language processing, and other challenging AI problems. Moreover, successfully training such networks is routinely performed using simple and scalable gradient-based methods, in particular stochastic gradient descent.

Despite this success, our theoretical understanding of the computational tractability of such methods is quite limited, with most results being negative. For example, as discussed in Livni et al. (2014), learning even depth-2 networks in a formal PAC learning framework is computationally hard in the worst case, and even if the algorithm is allowed to return arbitrary predictors. As common in such worst-case results, these are proven using rather artificial constructions, quite different than the real-world problems on which neural networks are highly successful. In particular, since the PAC framework focuses on *distribution-free* learning (where the distribution generating the examples is unknown and rather arbitrary), the hardness results rely on carefully crafted distributions, which allows one to relate the learning problem to (say) an NP-hard problem or breaking a cryptographic system. However, what if we insist on "natural" distributions? Is it possible to show that neural networks learning becomes computationally tractable? Can we show that they can be learned using the standard heuristics employed in practice, such as stochastic gradient descent?

To understand what a "natural" distribution refers to, we need to separate the distribution over examples (given as input-output pairs $(\mathbf{x}, y)$) into two components:

- *The input distribution* $p(\mathbf{x})$: "Natural" input distributions on Euclidean space tend to have properties such as smoothness, non-degeneracy, incoherence etc.
- *The target function* $h(\mathbf{x})$: In PAC learning, it is assumed that the output y equals $h(\mathbf{x})$, where h is some unkown target function from the hypothesis class we are considering. In studying neural networks, it is common to consider the class of all networks which share some fixed architecture (e.g. feedforward networks of a given depth and width). However, one may argue that the parameters of real-world networks (e.g. the weights of each neuron) are not arbitrary, but exhibit various features such as non-degeneracy or some "random like" appearance. Indeed, networks with a random structure have been shown to be more amenable to analysis in various situations (see for instance Daniely et al., 2016; Arora et al., 2014; Choromanska et al., 2015, and references therein).

Empirical evidence seems to suggest that many pairs of input distributions and target functions are computationally tractable to learn, using standard methods. However, how do we characterize these pairs? Would appropriate assumptions on one of them be sufficient to show learnability?

In this paper, we investigate these two components, and provide evidence that *neither one* of them alone is generally enough to guarantee computationally tractable learning, at least with methods resembling those used in practice. Specifically, we focus on simple, shallow ReLU networks, assume that the data can be perfectly predicted by some such network, and even allow *over-parametrization* (a.k.a. over-specification or improper learning), in the sense that we allow the learning algorithm to output a predictor which is possibly larger and more complex than the target function (this technique increases the power of the learner, and was shown to make the learning problem easier in theory and in practice, e.g. Livni et al. 2014; Safran and Shamir 2016; Soudry and Carmon 2016). Even under such favorable conditions, we show the following:

- **Hardness for "natural" target functions.** For each individual target function coming from a simple class of small, shallow ReLU networks (even if its parameters are chosen randomly or in some other oblivious way), we show that no algorithm invariant to linear transformations can successfully learn it w.r.t. all input distributions in polynomial time (this corresponds, for instance, to standard gradient-based methods together with data whitening or preconditioning). This result is based on a reduction from learning intersections of halfspaces. Although that problem is known to be hard in the worst-case over both input distributions and target functions, we essentially show that invariant algorithms as above do not "distinguish" between worst-case and average-case: If one can learn a particular target function with such an algorithm, then the algorithm can learn nearly all target functions in that class.
- **Hardness for "natural" input distributions.** We show that target functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ for any periodic ψ are generally difficult to learn using

gradient-based methods, even if the input distribution is fixed and belongs to a very broad class of smooth input distributions (including, for instance, Gaussians and mixtures of Gaussians). Note that such functions can essentially be constructed by simple shallow networks, and can be seen as an extension of generalized linear models (see McCullagh and Nelder 1989 for a survey). Unlike the previous result, which relies on a computational hardness assumption, the results here are geometric in nature, and imply that the gradient of the objective function, nearly everywhere, contains virtually no signal on the underlying target function. Therefore, any algorithm which relies on gradient information cannot learn such functions. Interestingly, the difficulty here is *not* in having a plethora of spurious local minima or saddle points—the associated stochastic optimization problem may actually have no such critical points. Instead, the objective function may exhibit properties such as *flatness* nearly everywhere, unless one is already very close to the global optimum. This highlights a potential pitfall in non-convex learning, which occurs already for a slight extension of generalized linear models, and even for "nice" input distributions.

Together, these results indicate that in order to explain the practical success of neural network learning with gradient-based methods, one would need to employ a careful combination of assumptions on both the input distribution and the target function, and that results with even a "partially" distribution-free flavor (which are common, for instance, in convex learning problems) may be difficult to attain here.

To prove our results, we develop some tools which may be of independent interest. In particular, the techniques used to prove hardness of learning functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ are based on Fourier analysis, and have some close connections to hardness results on learning Boolean functions such as parities in the well-known framework of learning from statistical queries (Kearns, 1998): In both cases, one essentially shows that the Fourier transform of the target function has very small support, and hence does not "correlate" with most functions, making it difficult to learn using certain methods. However, we consider a more general and arguably more natural class of input distributions over Euclidean space, rather than distributions on the Boolean cube. In a sense, we show that learning general periodic functions over Euclidean space is difficult (at least with gradient-based methods), for the same reasons that learning parities over the Boolean cube is difficult in the statistical queries framework. This connection has recently been formalized and extended in Song et al. (2017) (see discussion below).

1.1 Related Work

Recent years have seen quite a few papers on the theory of neural network learning. Below, we only briefly mention those most relevant to our paper.

In a very elegant work, Janzamin et al. (2015) have shown that a certain method based on tensor decompositions allows one to provably learn simple neural networks by a combination of assumptions on the input distribution and the target function. However, a drawback of their method is that it requires rather precise knowledge of the input distribution and its derivatives, which is rarely available in practice. In contrast, our focus is on algorithms which do not utilize such knowledge. Other works which show computationally-efficient

learnability of certain neural networks under sufficiently strong distributional assumptions include Arora et al. (2014); Livni et al. (2014); Andoni et al. (2014); Zhang et al. (2015).

In the context of learning functions over the Boolean cube, it is known that even if we restrict ourself to a particular input distribution (as long as it satisfies some mild conditions), it is difficult to learn parity functions using statistical query algorithms (Kearns, 1998; Blum et al., 1994), which also include gradient-based methods (Feldman et al., 2015). Since parities can be implemented with small real-valued networks, this implies that for "most" input distributions on the Boolean cube, there are neural networks which are unlikely to be learnable with gradient-based methods. However, data provided to neural networks in practice are not in the form of Boolean vectors, but rather vectors of floating-point numbers. Moreover, some assumptions on the input distribution, such as smoothness and Gaussianity, only make sense once we consider the support to be Euclidean space rather than the Boolean cube. Perhaps these are enough to guarantee computational tractability? A contribution of this paper is to show that this is not the case, and to formally demonstrate how phenomena similar to the Boolean case also occurs in Euclidean space, using appropriate target functions and distributions.

Related to the above, Song et al. (2017) recently showed that statistical query algorithms indeed cannot learn certain neural networks, using target functions similar to those we consider in Sec. 4, and for log-concave input distributions[1]. In contrast, our result for such target functions is specific to gradient-based methods, but applies to a different large family of distributions, not necessarily log-concave. Furthermore, we note that the challenges in fitting ridge functions $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ for certain ψ (when $\mathbf{x}$ is standard Gaussian and one attempts to fit the target function using a function of the same form) was also studied in Donoho and Johnstone (1989).

Finally, we note that Klivans and Kothari (2014) provides improper-learning hardness results, which hold even for a standard Gaussian distribution on Euclidean space, and for any algorithm. However, unlike our paper, their focus is on hardness of agnostic learning (where the target function is arbitrary and does not have to correspond to a given class), the results are specific to the standard Gaussian distribution, and the proofs are based on a reduction from the Boolean case.

The paper is structured as follows: In Sec. 2, we formally present some notation and concepts used throughout the paper. In Sec. 3, we provide our hardness results for natural target functions, and in Sec. 4, we provide our hardness results for natural input distributions. All proofs are presented in Sec. 5.

2. Preliminaries

We generally let bold-faced letters denote vectors. Given a complex-valued number $z = a + ib$, we let $\overline{z} = a - ib$ denote its complex conjugate, and $|z| = \sqrt{a^2 + b^2}$ denote its modulus. Given a function f, we let ∇f denote its gradient and $\nabla^2 f$ denote its Hessian (assuming they exist).

Neural Networks. Our results focus on learning predictors which can be described by simple and shallow (depth 2 or 3) neural networks. A standard feedforward neural network

1. The arXiv technical report on which our paper is based was published in September 2016, whereas their arXiv technical report was published in July 2017.

is composed of neurons, each of which computes the mapping $\mathbf{x} \mapsto \sigma(\mathbf{w}^\top \mathbf{x} + b)$, where $\mathbf{w}, b$ are parameters and σ is a scalar activation function, for example the popular ReLU function $[z]_+ = \max\{0, z\}$. These neurons are arranged in parallel in layers, so the output of each layer can be compactly represented as $\mathbf{x} \mapsto \sigma(W^\top \mathbf{x} + \mathbf{b})$, where W is a matrix (each column corresponding to the parameter vector of one of the neurons), $\mathbf{b}$ is a vector, and σ applies an activation function on the coordinates of $W^\top \mathbf{x}$. In vanilla feedforward networks, such layers are connected to each other, so given an input $\mathbf{x}$, the output equals

$$\sigma_k(W_k^\top \sigma_{k-1}(W_{k-1}^\top \dots \sigma_2(W_2^\top \sigma_1(W_1^\top \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2) \dots + \mathbf{b}_{k-1}) + \mathbf{b}_k),$$

where W_i, b_i, σ_i are parameter of the i-th layer. The number of layers k is denoted as the depth of the network, and the maximal number of columns in W_i is denoted as the width of the network. For simplicity, in this paper we focus on networks which output a real-valued number, and measure our performance with respect to the squared loss (that is, given an input-output example $(\mathbf{x}, y)$, where $\mathbf{x}$ is a vector and $y \in \mathbb{R}$, the loss of a predictor p on the example is $(p(\mathbf{x}) - y)^2$).

Gradient-Based Methods. Gradient-based methods are a class of optimization algorithms for solving problems of the form $\min_{\mathbf{w} \in \mathcal{W}} F(\mathbf{w})$ (for some given function F and assuming $\mathbf{w}$ is a vector in Euclidean space), based on computing $\nabla F(\mathbf{w})$ of approximations of $\nabla F(\mathbf{w})$ at various points $\mathbf{w}$. Perhaps the simplest such algorithm is gradient descent, which initializes deterministically or randomly at some point $\mathbf{w}_1$, and iteratively performs updates of the form $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta_t \nabla F(\mathbf{w}_t)$, where $\eta_t > 0$ is a step size parameter. In the context of statistical supervised learning problems, we are usually interested in solving problems of the form $\min_{\mathbf{w} \in \mathcal{W}} \mathbb{E}_{\mathbf{x} \sim \mathcal{D}}[\ell(f(\mathbf{w}, \mathbf{x}), h(\mathbf{x}))]$, where $\{\mathbf{x} \mapsto f(\mathbf{w}, \mathbf{x}) : \mathbf{w} \in \mathcal{W}\}$ is some class of predictors, h is a target function, and ℓ is some loss function. Since the distribution $\mathcal{D}$ is generally unknown, one cannot compute the gradient of this function w.r.t. $\mathbf{w}$ directly, but can still compute approximations, e.g. by sampling one $\mathbf{x}$ at random and computing the gradient (or sub-gradient) of $\ell(f(\mathbf{w}, \mathbf{x}), h(\mathbf{x}))$. The same approach can be used to solve empirical approximations of the above, i.e. $\min_{\mathbf{w} \in \mathcal{W}} \frac{1}{m} \sum_{i=1}^m \ell(f(\mathbf{w}, \mathbf{x}_i), h(\mathbf{x}_i))$ for some data set $\{(\mathbf{x}_i, h(\mathbf{x}_i))\}_{i=1}^m$. These are generally known as stochastic gradient methods, and are one of the most popular and scalable machine learning methods in practice.

PAC Learning. For the results of Sec. 3, we will rely on the following standard definition of PAC learning with respect to Boolean functions: Given a hypothesis class $\mathcal{H}$ of functions from $\{0,1\}^d$ to $\{0,1\}$, we say that a learning algorithm PAC-learns $\mathcal{H}$ if for any $\epsilon \in (0,1)$, any distribution $\mathcal{D}$ over $\{0,1\}^d$, and any $h^\star \in \mathcal{H}$, if the algorithm is given oracle access to i.i.d. samples $(\mathbf{x}, h^\star(\mathbf{x}))$ where $\mathbf{x}$ is sampled according to $\mathcal{D}$, then in time $\mathrm{poly}(d, 1/\epsilon)$, the algorithm returns a function $f : \{0,1\}^d \mapsto \{0,1\}$ (which can be evaluated in $\mathrm{poly}(d)$ time) such that $\Pr_{\mathbf{x} \sim \mathcal{D}}(f(\mathbf{x}) \neq h^\star(\mathbf{x})) \leq \epsilon$ with high probability (for our purposes, it will be enough to consider any constant close to 1). Note that in the definition above, we allow f not to belong to the hypothesis class $\mathcal{H}$. This is often denoted as "improper" learning, and allows the learning algorithm more power than in "proper" learning, where f must be a member of $\mathcal{H}$.

3. Natural Target Functions

In this section, we consider simple target functions parameterized by vectors $\mathbf{w}_1, \dots, \mathbf{w}_n$, of the form

$$\mathbf{x} \;\mapsto\; \left[\sum_{i=1}^n [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+\right]_+ - \left[\sum_{i=1}^n [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+ - 1\right]_+ ,$$

where $[z]_+ = \max\{0, z\}$ is the ReLU function, which correspond to standard depth-3 ReLU networks with $2n$ neurons in the first layer and 2 neurons in the second layer. Equivalently, these functions can also be written as

$$\mathbf{x} \mapsto \left[\sum_{i=1}^n [\langle \mathbf{w}_i, \mathbf{x} \rangle]_+\right]_{[0,1]} ,$$

where $[z]_{[0,1]} = \min\{1, \max\{0, z\}\}$ is the clipping operation on the interval $[0,1]$. For the rest of this section, we will use the latter formulation for convenience. Letting $W = [\mathbf{w}_1, \dots, \mathbf{w}_n]$, we can write such predictors as $\mathbf{x} \mapsto h(W^\top \mathbf{x})$ for an appropriate fixed function h. Our goal would be to show that *individually* for any such target function (regardless of how W is chosen, as long as it has full column rank), and any polynomial-time learning algorithm satisfying some conditions, there exists an input distribution on which it must fail.

We begin by noting that some algorithmic assumption is *necessary* to get such a target-function-specific result. Indeed, if we fix the target function in advance, we can always "learn" by the following algorithm: return the target function, regardless of the training data. To avoid such trivial scenarios, we will consider algorithms which exhibit certain natural invariances to the coordinate system used. One very natural invariance is with respect to orthogonal transformations: For example, if we rotate the input instances $\mathbf{x}_i$ in a fixed manner, then an orthogonally-invariant algorithm will return a predictor which still makes the same predictions on those instances. Formally, this invariance is defined as follows:

Definition 1 *Let $\mathcal{A}$ be an algorithm which inputs a data set $(\{\mathbf{x}_i, y_i\})_{i=1}^m$ (where $\mathbf{x}_i \in \mathbb{R}^d$) and outputs a predictor $\mathbf{x} \mapsto f(W^\top \mathbf{x})$ (for some function f and matrix W dependent on the data set). We say that $\mathcal{A}$ is* orthogonally-invariant*, if for any orthogonal matrix $M \in \mathbb{R}^{d \times d}$, if we feed the algorithm with $\{M\mathbf{x}_i, y_i\}_{i=1}^m$, the algorithm returns a predictor $\mathbf{x} \mapsto f(W_M^\top \mathbf{x})$, where f is the same as before and W_M is such that $W_M^\top M \mathbf{x}_i = W^\top \mathbf{x}_i$ for all $\mathbf{x}_i$.*

Remark 2 *The definition as stated refers to deterministic algorithms. For stochastic algorithms, we will understand orthogonal invariance to mean orthogonal invariance conditioned on any realization of the algorithm's random coin flips.*

For example, standard gradient and stochastic gradient descent methods for optimizing W (possibly with coordinate-oblivious regularization, such as L_2 regularization) can be easily shown to be orthogonally-invariant[2]. However, for our results we will need to make

2. Essentially, this is because the gradient of any function $g(W^\top \mathbf{x}) = g(\langle \mathbf{w}_1, \mathbf{x} \rangle, \dots, \langle \mathbf{w}_k, \mathbf{x} \rangle)$ w.r.t. any $\mathbf{w}_i$ is proportional to $\mathbf{x}$. Thus, if we multiply $\mathbf{x}$ by an orthogonal M, the gradient also gets multiplied by M. Since $M^\top M = I$, the inner products of instances $\mathbf{x}$ and gradients remain the same. Therefore, by induction, it can be shown that any algorithm which operates by incrementally updating some iterate by linear combinations of gradients will be rotationally invariant.

a somewhat stronger invariance assumption, namely invariance to general invertible linear transformations of the data (not necessarily just orthogonal). This is formally defined as follows:

Definition 3 *An algorithm $\mathcal{A}$ is* linearly-invariant*, if it satisfies Definition 1 for any invertible matrix $M \in \mathbb{R}^{d\times d}$ (rather than just orthogonal ones).*

One well-known example of such an algorithm (which is also invariant to affine transformations) is the Newton method (Boyd and Vandenberghe, 2004). More relevant to our purposes, linear invariance occurs whenever an orthogonally-invariant algorithm preconditions or "whitens" the data so that its covariance has a fixed structure (e.g. the identity matrix, possibly after a dimensionality reduction if the data is rank-deficient). For example, even though gradient descent methods are not linearly invariant, they become so if we precede them by such a preconditioning step. This is formalized in the following theorem:

Theorem 4 *Let $\mathcal{A}$ be any algorithm which given $\{\mathbf{x}_i, y_i\}_{i=1}^m$, computes the whitening matrix $P = D^{-1}U^\top$ (where $X = [\mathbf{x}_1\ \mathbf{x}_2\ \ldots\ \mathbf{x}_m]$, $X = UDV^\top$ is a thin[3] SVD decomposition of X), feeds $\{P\mathbf{x}_i, y_i\}_{i=1}^m$ to an orthogonally-invariant algorithm, and given the output predictor $\mathbf{x} \mapsto f(W^\top\mathbf{x})$, returns the predictor $\mathbf{x} \mapsto f((P^\top W)^\top\mathbf{x})$. Then $\mathcal{A}$ is linearly-invariant.*

It is easily verified that the covariance matrix of the transformed instances $P\mathbf{x}_1, \ldots, P\mathbf{x}_m$ is the $r \times r$ identity matrix (where $r = \mathrm{Rank}(X)$), so this is indeed a whitening transform. We note that whitening is a very common preprocessing heuristic, and even when not done explicitly, scalable approximate whitening and preconditioning methods are very common and widely recognized as useful for training neural networks (for example, Adagrad and batch normalization, see Duchi et al. 2011 and Ioffe and Szegedy 2015).

To show our result, we rely on a reduction from a PAC-learning problem known to be computationally hard, namely learning intersections of halfspaces. These are Boolean predictors parameterized by $\mathbf{w}_1, \ldots, \mathbf{w}_n \in \mathbb{R}^d$ and $b_1, \ldots, b_n \in \mathbb{R}$, which compute a mapping of the form

$$\mathbf{x} \ \to \ \bigwedge_{i=1}^{n} \left(\langle \mathbf{w}_i, \mathbf{x}\rangle \geq b_i\right)$$

(where we let 1 correspond to 'true' and 0 to 'false'). The problem of PAC-learning intersections of halfspaces over the Boolean cube ($\mathbf{x} \in \{0,1\}^d$) has been well-studied. In particular, two known hardness results are the following:

- Klivans and Sherstov (2009) show that under a certain well-studied cryptographic assumption (hardness of finding unique shortest vectors in a high-dimensional lattice), no algorithm can PAC-learn intersection of $n_d = d^\delta$ halfspaces (where δ is any positive constant), even if the coordinates of $\mathbf{w}_i$ and b_i are all integers, and $\max_i \|(\mathbf{w}_i, b_i)\| \leq \mathrm{poly}(d)$.

3. That is, if X is of size $d \times m$, then U is of size $d \times \mathrm{Rank}(X)$, D is of size $\mathrm{Rank}(X) \times \mathrm{Rank}(X)$, and V is of size $m \times \mathrm{Rank}(X)$.

- Daniely and Shalev-Shwartz (2016) show that under an assumption related to the hardness of refuting random K-SAT formulas, no algorithm can PAC-learn intersections of $n_d = \omega(\log(d))$ halfspaces (as $d \to \infty$), even if the coordinates of $\mathbf{w}_i$ and b_i are all integers, and $\max_i \|(\mathbf{w}_i, b_i)\| = \mathcal{O}(d)$.

In the theorem below, we will use the result of Daniely and Shalev-Shwartz (2016), which applies to an intersection of a smaller number of halfspaces, and with smaller norms. However, similar results can be shown using Klivans and Sherstov (2009), at the cost of worse polynomial dependencies on d.

The main result of this section is the following:

Theorem 5 *Consider any network $h(W_\star^\top\mathbf{x}) = [\sum_{i=1}^{n_d}[\langle \mathbf{w}_i^\star, \mathbf{x}\rangle]_+]_{[0,1]}$ (where the columns of $W_\star$ are $\mathbf{w}_1^\star \ldots \mathbf{w}_n^\star$), which satisfies the following:*

- *$n_d = \omega(\log(d))$ as $d \to \infty$*
- *$\max_i \|\mathbf{w}_i^\star\| = \mathcal{O}(d)$*
- *$\mathbf{w}_1^\star \ldots \mathbf{w}_n^\star$ are linearly independent, so the smallest singular value $s_{\min}(W_\star)$ of $W_\star$ is strictly positive.*

Then under the assumption stated in Daniely and Shalev-Shwartz (2016), there is no linearly-invariant algorithm which for any $\epsilon > 0$ and any distribution $\mathcal{D}$ over vectors of norm $\frac{\mathcal{O}(d\sqrt{dn_d})}{\min\{1, s_{\min}(W_\star)\}}$, given only access to samples $(\mathbf{x}, h(W_\star^\top\mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, runs in time $\mathrm{poly}(d, 1/\epsilon)$ and returns with high probability a predictor $\mathbf{x} \mapsto f(W^\top\mathbf{x})$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(W^\top\mathbf{x}) - h(W_\star^\top\mathbf{x})\right)^2\right] \leq \epsilon.$$

Note that the result holds even if the returned predictor $f(W^\top\mathbf{x})$ has a different structure than $h(W_\star^\top\mathbf{x})$, and W is of a larger size than $W_\star$. Thus, it applies even if the algorithm implements over-parametrization and attempts to train a network larger than $h(W_\star^\top\mathbf{x})$.

The proof (which is provided in Sec. 5) can be sketched as follows: First, the hardness assumption for learning intersection of halfspaces is shown to imply hardness of learning networks $\mathbf{x} \mapsto h(W^\top\mathbf{x})$ as described above (and even if W has linearly independent columns—a restriction which will be important later). However, this only implies that no algorithm can learn $\mathbf{x} \mapsto h(W^\top\mathbf{x})$ for *all* W and all input distributions $\mathcal{D}$. In contrast, we want to show that learning would be difficult even for *some fixed* $W_\star$. To do so, we show that if an algorithm is linearly invariant, then the ability to learn with respect to some W and all distributions $\mathcal{D}$ means that we can learn with respect to all W and all $\mathcal{D}$. Roughly speaking, we argue that for linearly-invariant algorithms, "average-case" and "worst-case" hardness are the same here. Intuitively, this is because given some arbitrary $W, \mathcal{D}$, we can create a different input distribution $\tilde{\mathcal{D}}$, so that $W, \tilde{\mathcal{D}}$ "look like" $W_\star, \mathcal{D}$ under some linear transformation (see Figure 1 for an illustration). Therefore, a linearly-invariant algorithm which succeeds on one will also succeed on the other.

A bit more formally, let us fix some $W_\star$ (with linearly independent columns), and suppose we have a linearly-invariant algorithm which can successfully learn $\mathbf{x} \mapsto h(W_\star^\top\mathbf{x})$ with

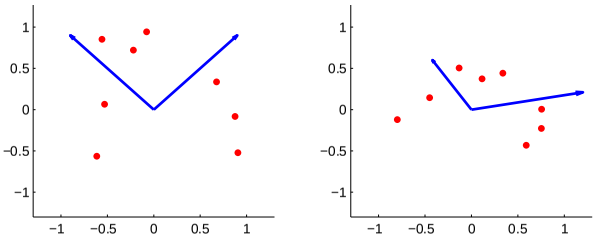


Figure 1: Correspondence between $W_\star, \mathcal{D}$ (left figure) and $W, \tilde{\mathcal{D}}$ (right figure). Arrows correspond to columns of $W_\star$ and W, and dots correspond to the support of $\mathcal{D}$ and $\tilde{\mathcal{D}}$. $\tilde{\mathcal{D}}$ is constructed so that the same linear transformation mapping $W_\star$ to W also maps $\mathcal{D}$ to $\tilde{\mathcal{D}}$.

respect to any input distribution. Let $W, \mathcal{D}$ be some other matrix and distribution with respect to which we wish to learn (where W has full column rank and is of the same size as $W_\star$). Then it can be shown that there is an invertible matrix M such that $W = M^\top W_\star$. Since the algorithm successfully learns $\mathbf{x} \mapsto h(W_\star^\top \mathbf{x})$ with respect to any input distribution, it would also successfully learn if we use the input distribution $\tilde{\mathcal{D}}$ defined by sampling $\mathbf{x} \sim \mathcal{D}$ and returning $M\mathbf{x}$. This means that the algorithm would succesfully learn from data distributed as

$$(\mathbf{x}, h(W_\star^\top \mathbf{x}))\ ,\ \mathbf{x} \sim \tilde{\mathcal{D}} \iff (M\mathbf{x}, h(W_\star^\top (M\mathbf{x})))\ ,\ \mathbf{x} \sim \mathcal{D} \iff (M\mathbf{x}, h(W^\top \mathbf{x}))\ ,\ \mathbf{x} \sim \mathcal{D}.$$

Since the algorithm is linearly-invariant, it can be shown that this implies successful learning from $(\mathbf{x}, h(W^\top \mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, as required.

In the sketch above, we have ignored some technical issues. For example, we need to be careful that M has a bounded spectral norm, so that it induces a linear transformation which does not distort norms by too much (as all our arguments apply for input distributions supported on a bounded domain). A second issue is that if we apply a linearly-invariant algorithm on a data set transformed by M, then the invariance is only with respect to the data, not necessarily with respect to new instances $\mathbf{x}$ sampled from the same distribution (and this restriction is necessary for results such as Thm. 4 to hold without further assumptions). However, it can be shown that if the data set is large enough, invariance will still occur with high probability over the sampling of $\mathbf{x}$, which is sufficient for our purposes.

4. Natural Input Distributions

In this section, we consider the difficulty of gradient-based methods to learn certain target functions, even with respect to smooth, well-behaved distributions over $\mathbb{R}^d$. Specifically, we will consider functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$, where $\mathbf{w}^\star$ is a vector of bounded norm and ψ is a periodic function. Note that if ψ is continuous and piecewise linear, then $\psi(\mathbf{w}^{\star\top}\mathbf{x})$ can be implemented by a depth-2 neural ReLU network on any bounded subset

of the domain. More generally, any continuous periodic function can be approximated arbitrarily well by such networks.

Our formal results rely on Fourier analysis and are a bit technical. Hence, we precede them with an informal description, outlining the main ideas and techniques, and presenting a specific case study which may be of independent interest (Subsection 4.1). The formal results are presented in Subsection 4.2.

4.1 Informal Description of Results and Techniques

Consider a target function of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$, and an input distribution with density function $\varphi^2(\cdot)$, where φ is some non-negative function (we consider the density as the square of some function in order to simplify notation later on). Suppose we attempt to learn this target function (with respect to the squared loss) using *some* hypothesis class, which can be parameterized by a bounded-norm vector $\mathbf{v}$ in some subset $\mathcal{V}$ of an Euclidean space (not necessarily of the same dimensionality as $\mathbf{w}^\star$), so each predictor in the class can be written as $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$ for some fixed mapping f. Thus, our goal is essentially to solve the stochastic optimization problem

$$\min_{\mathbf{v}:\mathbf{v}\in\mathcal{V}} \mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[\left(f(\mathbf{v},\mathbf{x}) - \psi(\mathbf{w}^{\star\top}\mathbf{x})\right)^2\right]. \tag{1}$$

In this section, we study the geometry of this objective function, and show that under mild conditions on f, and assuming the norm of $\mathbf{w}^\star$ is reasonably large, the gradient of the objective function with respect to $\mathbf{w}$ is almost independent of $\mathbf{w}^\star$, in the following sense: If we fix any $\mathbf{w}$ and choose $\mathbf{w}^\star$ uniformly at random, the gradient will be extremely concentrated around a fixed value which is independent of $\mathbf{w}^\star$ (e.g. exponentially small in $\|\mathbf{w}^\star\|^2$ for a Gaussian or a mixture of Gaussians). Therefore, assuming $\|\mathbf{w}^\star\|$ is reasonably large, any standard gradient-based method will follow a trajectory nearly independent of $\mathbf{w}^\star$. In fact, in practice we do not even have access to exact gradients of Eq. (1), but only to noisy and biased versions of it (e.g. if we perform stochastic gradient descent, and certainly if we use finite-precision computations). In that case, the noise will completely obliterate the exponentially small signal about $\mathbf{w}^\star$ in the gradients, and will make the trajectory essentially independent of $\mathbf{w}^\star$. As a result, assuming ψ and the distribution is such that the function $\psi(\mathbf{w}^{\star\top}\mathbf{x})$ is sensitive to the direction of $\mathbf{w}^\star$, it follows that these methods will fail to optimize Eq. (1) successfully. Finally, we note that in practice, it is common to solve not Eq. (1) directly, but rather its empirical approximation with respect to some fixed finite training set. Still, by concentration of measure, this empirical objective would converge to the one in Eq. (1) given enough data, so the same issues will occur.

An important feature of our results is that they make virtually no structural assumptions on the predictors $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$. In particular, they can represent arbitrary classes of neural networks (as well as other predictor classes). Thus, our results imply that target functions of the form $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$, where ψ is periodic, would be difficult to learn using gradient-based methods, even if we allow improper learning and consider predictor classes of a different structure.

To explain how such results are attained, let us study a concrete special case (not necessarily in the context of neural networks). Consider the target function $\mathbf{x} \mapsto \cos(2\pi\mathbf{w}^{\star\top}\mathbf{x})$,

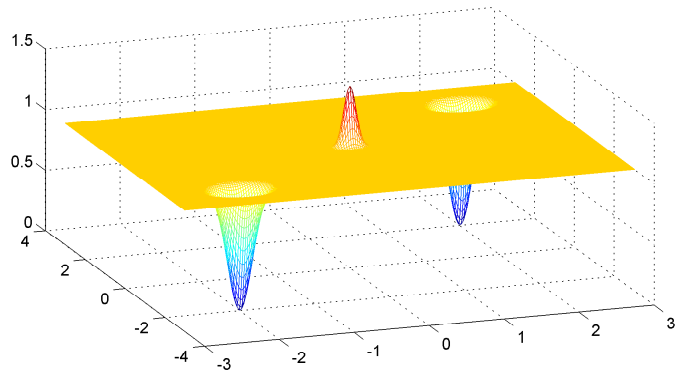


Figure 2: Graphical depiction of the objective function in Eq. (2), in 2 dimensions and where $\mathbf{w}^\star = (2,2)$.

and the hypothesis class (parameterized by $\mathbf{w}$) of functions $\mathbf{x} \mapsto \cos(2\pi \mathbf{w}^\top \mathbf{x})$. Thus, Eq. (1) takes the form

$$\min_{\mathbf{w}} \mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[\left(\cos(2\pi \mathbf{w}^\top \mathbf{x}) - \cos(2\pi \mathbf{w}^{\star\top}\mathbf{x})\right)^2\right]. \tag{2}$$

Furthermore, suppose the input distribution φ^2 is a standard Gaussian on $\mathbb{R}^d$. In two dimensions and for $\mathbf{w}^\star = (2,2)$, the objective function in Eq. (1) turns out to have the form illustrated in Figure 2. This objective function has only three critical points: A global maximum at $\mathbf{0}$, and two global minima at $\mathbf{w}^\star$ and $-\mathbf{w}^\star$. Nevertheless, it would be difficult to optimize using gradient-based methods, since it is extremely *flat* everywhere except close to the critical points. As we will see shortly, the same phenomenon occurs in higher dimensions. In high dimensions, if the direction of $\mathbf{w}^\star$ is chosen randomly, we will be overwhelmingly likely to initialize far from the global minima, and hence will start in a flat plateau in which most gradient-based methods will stall[4].

We now turn to explain why Eq. (2) has the form shown in Figure 2. This will also help to illustrate our proof techniques, which apply much more generally. The main idea is to analyze the Fourier transform of Eq. (2). Letting $\cos_{\mathbf{w}}$ denote the function $\mathbf{x} \mapsto \cos(2\pi \mathbf{w}^\top \mathbf{x})$, we can write Eq. (2) as

$$\int \left(\cos(2\pi \mathbf{w}^\top \mathbf{x}) - \cos(2\pi \mathbf{w}^{\star\top}\mathbf{x})\right)^2 \varphi^2(\mathbf{x})d\mathbf{x} \;=\; \left\|\cos_{\mathbf{w}}\cdot\varphi - \cos_{\mathbf{w}^\star}\cdot\varphi\right\|^2,$$

where $\|\cdot\|$ is the standard norm over the space $L^2(\mathbb{R}^d)$ of square integrable functions. By standard properties of the Fourier transform (as described in Sec. 2), this squared norm

4. Although there are techniques to overcome flatness (e.g. by normalizing the gradient, see Nesterov 1984; Hazan et al. 2015), in our case the normalization factor will be huge and require extremely precise gradient information, which as discussed earlier, is unrealistic here.

of a function equals the squared norm of the function's Fourier transform, which equals in turn

$$\left\|\widehat{\cos_{\mathbf{w}}\cdot\varphi} - \widehat{\cos_{\mathbf{w}}\cdot\varphi}\right\|^2 \;=\; \left\|\widehat{\cos_{\mathbf{w}}} * \hat{\varphi} - \widehat{\cos_{\mathbf{w}^\star}} * \hat{\varphi}\right\|^2.$$

$\widehat{\cos_{\mathbf{w}}}(\boldsymbol{\xi})$ can be shown to equal $\frac{1}{2}\left(\delta(\boldsymbol{\xi}-\mathbf{w}) + \delta(\boldsymbol{\xi}+\mathbf{w})\right)$, where $\delta(\cdot)$ is Dirac's delta function (a "generalized" function which satisfies $\delta(\mathbf{z}) = 0$ for all $\mathbf{z} \neq \mathbf{0}$, and $\int \delta(\mathbf{z})d\mathbf{z} = 1$). Plugging this into the above and simplifying, we get

$$\begin{aligned}&\frac{1}{4}\left\|\hat{\varphi}(\cdot-\mathbf{w}) + \hat{\varphi}(\cdot+\mathbf{w}) - \hat{\varphi}(\cdot-\mathbf{w}^\star) - \hat{\varphi}(\cdot+\mathbf{w}^\star)\right\|^2 \\ &= \frac{1}{4}\int_{\boldsymbol{\xi}} \left|\hat{\varphi}(\boldsymbol{\xi}-\mathbf{w}) + \hat{\varphi}(\boldsymbol{\xi}+\mathbf{w}) - \hat{\varphi}(\boldsymbol{\xi}-\mathbf{w}^\star) - \hat{\varphi}(\boldsymbol{\xi}+\mathbf{w}^\star)\right|^2 d\boldsymbol{\xi}, \end{aligned} \tag{3}$$

where $\hat{\varphi}(\cdot - \mathbf{w})$ stands for the function $\mathbf{x} \mapsto \hat{\varphi}(\mathbf{x}-\mathbf{w})$, etc. If φ^2 is a standard Gaussian, $\hat{\varphi}(\boldsymbol{\xi})$ can be shown to equal the Gaussian-like function $(4\pi)^{d/2}a^{-\|\boldsymbol{\xi}\|^2}$ where $a = \exp(4\pi^2)$. Plugging back, the expression above is proportional to

$$\int_{\boldsymbol{\xi}} \left(\left(a^{-\|\boldsymbol{\xi}-\mathbf{w}\|^2} + a^{-\|\boldsymbol{\xi}+\mathbf{w}\|^2}\right) - \left(a^{-\|\boldsymbol{\xi}-\mathbf{w}^\star\|^2} + a^{-\|\boldsymbol{\xi}+\mathbf{w}^\star\|^2}\right)\right)^2 d\boldsymbol{\xi}. \tag{4}$$

The expression in each inner parenthesis can be viewed as a mixture of two Gaussian-like functions, with centers at $\mathbf{w}, -\mathbf{w}$ (or $\mathbf{w}^\star, -\mathbf{w}^\star$). Thus, if $\mathbf{w}$ is far from $\mathbf{w}^\star$, these two mixtures will have nearly disjoint support, and Eq. (4) will have nearly the same value regardless of $\mathbf{w}$—in other words, it is very flat. Since this equation is nothing more than a re-formulation of the original objective function in Eq. (2) (up to a constant), we get a similar behavior for Eq. (2) as well.

This behavior extends, however, much more generally than the specific objective in Eq. (2). First of all, we can replace the standard Gaussian distribution φ^2 by any distribution such that $\hat{\varphi}$ has a localized support. This would still imply that Eq. (3) refers to the difference of two functions with nearly disjoint support, and the same flatness phenomenon will occur. Second, we can replace the cos function by any periodic function ψ. By properties of the Fourier transform of periodic functions, we still get localized functions in the Fourier domain (more precisely, the Fourier transform will be localized around integer multiples of $\mathbf{w}$, up to scaling). Finally, instead of considering hypothesis classes of predictors $\mathbf{x} \mapsto \psi(\mathbf{w}^\top \mathbf{x})$ similar to the target function, we can consider quite arbitrary mappings $\mathbf{x} \mapsto f(\mathbf{w}, \mathbf{x})$. Even though this function may no longer be localized in the Fourier domain, it is enough that only the target function $\mathbf{x} \mapsto \psi(\mathbf{w}^{\star\top}\mathbf{x})$ will be localized: That implies that regardless how f looks like, under a random choice of $\mathbf{w}^\star$, only a minuscule portion of the L_2 mass of f overlaps with the target function, hence getting sufficient signal on $\mathbf{w}^\star$ will be difficult.

As mentioned in the introduction, these techniques and observations are closely related to hardness results in the statistical queries literature, and can indeed be applied to that framework as shown recently in Song et al. (2017).

4.2 Formal Results

We now turn to provide a more formal statement of our results. The distributions we will consider consist of arbitrary mixtures of densities, whose square roots have Fourier transforms with rapidly decaying tails. More precisely, we have the following definition:

Definition 6 *Let $\epsilon(r)$ be some function from $[0,\infty)$ to $[0,1]$. A function $\varphi^2 : \mathbb{R}^d \to \mathbb{R}$ is* $\epsilon(r)$ Fourier-concentrated *if its square root φ belongs to $L^2(\mathbb{R}^d)$, and satisfies*

$$\|\hat{\varphi} \cdot \mathbf{1}_{\geq r}\| \;\leq\; \|\hat{\varphi}\|\,\epsilon(r),$$

where $\mathbf{1}_{\geq r}$ is the indicator function of $\{\mathbf{x} : \|\mathbf{x}\| \geq r\}$.

A canonical example is Gaussian distributions: Given a (non-degenerate, zero-mean) Gaussian density function φ^2 with covariance matrix Σ, its square root φ is proportional to a Gaussian with covariance 2Σ, and its Fourier transform $\hat{\varphi}$ is well-known to be proportional to a Gaussian with covariance $(2\Sigma)^{-1}$. By standard Gaussian concentration results, it follows that φ^2 is Fourier-concentrated with $\epsilon(r) = \exp(-\Omega(\lambda_{\min} r^2))$ where $\lambda_{\min}$ is the minimal eigenvalue of Σ. A similar bound can be shown when the Gaussian has some arbitrary mean. More generally, it is well-known that smooth functions (differentiable to sufficiently high order with integrable derivatives) have Fourier transforms with rapidly decaying tails. For example, if we consider the broad class of *Schwartz* functions (characterized by having values and all derivatives decaying faster than polynomially in r), then the Fourier transform of any such function is also a Schwartz function, which implies super-polynomial decay of $\epsilon(r)$ (see for instance Hunter and Nachtergaele 2001, Chapter 11 and Proposition 11.25).

We now formally state our main result for this section. We consider *any* predictor of the form $\mathbf{x} \mapsto f(\mathbf{v}, \mathbf{x})$, where f is some fixed function and $\mathbf{v}$ is a parameter vector coming from some domain $\mathcal{V}$, which we will assume w.l.o.g. to be a subset of some Euclidean space[5] (for example, f can represent a network of a given architecture, with weights specified by $\mathbf{v}$). When learning f based on data coming from an underlying distribution, we are essentially attempting to solve the optimization problem

$$\min_{\mathbf{v}\in\mathcal{V}} F_{\mathbf{w}^\star}(\mathbf{v}) := \mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[\left(f(\mathbf{v},\mathbf{x}) - \psi(\mathbf{w}^{\star\top}\mathbf{x})\right)^2\right].$$

Assuming that F is differentiable w.r.t. $\mathbf{w}$, any gradient-based method to solve this problem proceeds by computing (or approximating) $\nabla F_{\mathbf{w}^\star}(\mathbf{v})$ at various points $\mathbf{v}$. However, the following theorem shows that at *any* $\mathbf{v}$, and *regardless* of the type of predictor or network one is attempting to train, the gradient at $\mathbf{v}$ is virtually independent of the underlying target function, and hence provides very little signal:

Theorem 7 *Suppose that*

- *$\psi : \mathbb{R} \to [-1,+1]$ is a periodic function of period 1, which has bounded variation on every finite interval.*
- *φ^2 is a density function on $\mathbb{R}^d$, which can be written as a (possibly infinite) mixture $\varphi^2 = \sum_i \alpha_i \varphi_i^2$, where each φ_i^2 is $\epsilon(r)$ Fourier-concentrated.*
- *At some fixed $\mathbf{v}$, $\mathbb{E}_{\mathbf{x}\sim\varphi^2}\left\|\frac{\partial}{\partial \mathbf{v}} f(\mathbf{v},\mathbf{x})\right\|^2 \leq G_{\mathbf{v}}$ for some $G_{\mathbf{v}}$.*

5. More generally, our analysis is applicable to any separable Hilbert space.

Then for some universal positive constants c_1, c_2, c_3, if $d \geq c_1$, and if $\mathbf{w}^\star \in \mathbb{R}^d$ is a vector of norm $2r$ chosen uniformly at random, then

$$Var_{\mathbf{w}^\star}\left(\nabla F_{\mathbf{w}^\star}(\mathbf{v})\right) := \mathbb{E}_{\mathbf{w}^\star}\left\|\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]\right\|^2 \leq\; c_2 G_{\mathbf{v}}\left(\exp(-c_3 d) + \sum_{n=1}^{\infty}\epsilon(nr)\right).$$

Note that bounded variation is weaker than, say, Lipschitz continuity. Also, we note that the mixture assumption is used in a rather limited sense: most of the proof is devoted to establishing the bound w.r.t. Fourier-concentrated densities, and the result is then extended to mixtures by Jensen's inequality (see the proof for details).

Assuming $\epsilon(r)$ decays rapidly with r—say, exponentially in r^2 as is the case for a mixture of Gaussians—we get that the bound in the theorem is on the order of $\exp(-\Omega(\min\{d, r^2\}))$. Overall, the theorem implies that if r, d are moderately large, the gradient of $F_{\mathbf{w}^\star}$ at any point $\mathbf{v}$ is extremely concentrated around a fixed value, independent of $\mathbf{w}^\star$. This implies that gradient-based methods, which attempt to optimize $F_{\mathbf{w}^\star}$ via gradient information, are unlikely to succeed.

One way to formalize this argument is to consider any iterative algorithm (possibly randomized), which relies on an *ε-approximate* gradient oracle to optimize $F_{\mathbf{w}^\star}$: At every iteration t, the algorithm chooses a point $\mathbf{v}_t \in \mathcal{V}$, and receives a vector $\mathbf{g}_t$ such that $|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_t) - \mathbf{g}_t| \leq \varepsilon$. More formally, we can define such oracles and algorithms as follows:

Definition 8 (Approximate Gradient Oracle) *$O_{F,\epsilon}$ is an ε-approximate gradient oracle w.r.t. the function F on a domain $\mathcal{V}$, if given any input $\mathbf{v} \in \mathcal{V}$, it returns a fixed vector $\mathbf{g}$ such that $|\nabla F(\mathbf{v}) - \mathbf{g}| \leq \varepsilon$.*

Definition 9 (Approximate Gradient-Based Method) *Given a domain $\mathcal{V}$, we say that an algorithm A is a T-iterations, ε-approximate gradient-based method, if there exists some (deterministic or randomized) $\mathbf{v}_1 \in \mathcal{V}$ and (deterministic or randomized) functions $\{f_i : \mathcal{V}^i \to \mathcal{V}\}_{i=1}^{T-1}$, such that for any function F on $\mathcal{V}$, the output $\mathbf{v}_T \in \mathcal{V}$ of A given F can be written recursively as $\mathbf{v}_{t+1} = f_t(O_{F,\epsilon}(\mathbf{v}_1), \ldots, O_{F,\epsilon}(\mathbf{v}_t))$ for some approximate gradient oracle $O_{F,\epsilon}$.*

In our case, we will be interested in ε such that ε^3 is on the order of the bound in Thm. 7. Since the bound is extremely small for moderate d, r (say, smaller than machine precision), our definition of approximate gradient-based methods are a realistic model of gradient-based methods on finite-precision machines, even if one attempts to compute the gradients accurately. The following theorem implies that if the number of iterations is not extremely large (on the order of $1/\varepsilon$, e.g. $\exp(\Omega(\min\{d, r^2\})$ iterations for Gaussian mixtures), then with high probability, an approximate gradient-based method will return the same predictor independent of $\mathbf{w}^\star$. However, since the objective function $F_{\mathbf{w}^\star}$ is generally highly sensitive to the choice of $\mathbf{w}^\star$, this means that no such method can train a reasonable predictor.

Theorem 10 *Assume the conditions of Thm. 7, and let*

$$\varepsilon = \sqrt[3]{c_2(\sup_{\mathbf{v}\in\mathcal{V}} G_{\mathbf{v}})\left(\exp(-c_3 d) + \sum_{n=1}^{\infty}\epsilon(nr)\right)}$$

be the cube root of the bound specified there (uniformly over all $\mathbf{v} \in \mathcal{V}$). Then there exist a choice of approximate gradient oracles $\{O_{F_{\mathbf{w}^*,\varepsilon}} : \|\mathbf{w}^*\| = 2r\}$, such that the following holds for any $p \in (0,1)$ and any $\lfloor p/\varepsilon \rfloor$-iterations, ε-approximate gradient-based method A: Conditioned on an event which holds with probability $1-p$ over the random choice of $\mathbf{w}^\star$, the distribution of the output of A given $F_{\mathbf{w}^*}$ is fixed independent of $\mathbf{w}^*$.

5. Proofs

5.1 Proof of Thm. 4

Let P_M denote the whitening matrix employed if we transform the instances X by some invertible $d \times d$ matrix M (that is, X becomes MX), and P the whitening matrix employed for the original data.

Using the same notation as in the theorem, it is easily verified that $PX = V^\top$, and $P_M MX = V_M^\top$, where $U_M D_M V_M^\top$ is an SVD decomposition of the matrix MX. Since both $V^\top$ and $V_M^\top$ are $\mathrm{Rank}(X) \times m$ matrices with rows consisting of orthonormal vectors, they are related by an orthogonal transformation (i.e. there is an orthogonal matrix R_M such that $R_M V^\top = V_M^\top$). Therefore, $R_M PX = P_M MX$. Since the data is fed to an orthogonally-invariant algorithm, its output W_M satisfies $W_M^\top P_M MX = W^\top PX$. This in turn implies $W_M^\top R_M PX = W^\top PX$, and hence $W_M^\top R_M V^\top = W^\top V^\top$. Multiplying both sides on the right by V and taking a transpose, we get that $R_M^\top W_M = W$, and hence $W_M = R_M W$. In words, W and W_M are the same up to an orthogonal transformation R_M depending on M. Therefore,

$$(P_M^\top W_M)^\top MX = W_M^\top P_M MX = W^\top R_M^\top R_M PX = W^\top PX = (P^\top W)^\top X,$$

so we see that the returned predictor makes the same predictions over the data set, independent of the transformation matrix M.

5.2 Proof of Thm. 5

We start with the following auxiliary theorem, which reduces the hardness result of Daniely and Shalev-Shwartz (2016) to one about neural networks of the type we discuss here:

Theorem 11 *Under the assumption stated in Daniely and Shalev-Shwartz (2016), the following holds for any $n_d = \omega(\log(d))$ (as $d \to \infty$):*

There is no algorithm running in time poly$(d, 1/\epsilon)$, which for any distribution $\mathcal{D}$ on $\{0,1\}^d$, and any $h(W^\top \mathbf{x}) = [\sum_{i=1}^{n_d} [\langle \mathbf{w}_i, \mathbf{x}\rangle]_+]_{[0,1]}$ (where $W = [\mathbf{w}_1, \mathbf{w}_2 \dots \mathbf{w}_{n_d}]$ and $\max_i \|\mathbf{w}_i\|$ is $\mathcal{O}(d)$), given only access to samples $(\mathbf{x}, h(W^\top \mathbf{x}))$ where $\mathbf{x} \sim \mathcal{D}$, returns with high probability a function f such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] \le \epsilon.$$

Proof Suppose for the sake of contradiction that there exists an algorithm $\mathcal{A}$ which for any distribution and $h(W^\top \mathbf{x})$ as described in the theorem, returns a function f such that $\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}[(f(\mathbf{x}) - h(W^\top \mathbf{x}))^2] \le \epsilon$ with high probability.

In particular, let us focus on distributions $\mathcal{D}$ supported on $\{0,1\}^{d-1} \times \{1\}$. For these distributions, we argue that any intersection of halfspaces on $\mathbb{R}^{d-1}$ specified by $\mathbf{w}_1, \dots, \mathbf{w}_{n_d} \in \mathbb{R}^{d-1}$ with integer coordinates, and integer $b_1, \dots, b_n$, can be specified as $\mathbf{x} \mapsto 1 - h(W^\top \mathbf{x})$ for some function h as described in the theorem statement. To see this, note that for any $\mathbf{w}_i, b_i$ and $\mathbf{x} = (\mathbf{x}', 1)$ in the support of $\mathcal{D}$, $[\langle(-\mathbf{w}_i, b_i), \mathbf{x}\rangle]_+ = [-\langle \mathbf{w}_i, \mathbf{x}'\rangle + b_i]_+$ is a non-negative integer, hence

$$\left[\sum_{i=1}^{n_d} [\langle(-\mathbf{w}_i, b_i), \mathbf{x}\rangle]_+\right]_{[0,1]} = \left[\sum_{i=1}^{n_d} [-\langle \mathbf{w}_i, \mathbf{x}'\rangle + b_i]_+\right]_{[0,1]} = \bigvee_{i=1}^{n_d} \left(-\langle \mathbf{w}_i, \mathbf{x}'\rangle + b_i > 0\right)$$
$$= \bigvee_{i=1}^{n_d} \left(\langle \mathbf{w}_i, \mathbf{x}'\rangle < b_i\right) = \neg\left(\bigwedge_{i=1}^{n_d} \left(\langle \mathbf{w}_i, \mathbf{x}'\rangle \ge b_i\right)\right).$$

Therefore, for any distribution over examples labelled by an intersection of halfspaces $\mathbf{x} \mapsto 1 - h(W^\top \mathbf{x})$ (with integer-valued coordinates and bounded norms), by feeding $\mathcal{A}$ with $\{\mathbf{x}_i, 1 - y_i\}_{i=1}^m$, the algorithm returns a function f, such that with high probability, $\mathbb{E}_{\mathbf{x}}\left(f(\mathbf{x}) - h(W^\top \mathbf{x})\right)^2 \le \epsilon$, and therefore

$$\mathbb{E}_{\mathbf{x}}\left(\left(1 - f(\mathbf{x})\right) - \left(1 - h(W^\top \mathbf{x})\right)\right)^2 \le \epsilon.$$

In particular, if we consider the Boolean function $\tilde{f}(\mathbf{x}) = 1 - \mathrm{rnd}(f(\mathbf{x}))$, where $\mathrm{rnd}(z) = 0$ if $z \le 1/2$ and $\mathrm{rnd}(z) = 1$ if $z > 1/2$, we argue that $\Pr_{\mathbf{x}}(\tilde{f}(\mathbf{x}) \ne 1 - h(W^\top \mathbf{x})) \le 8\epsilon$. Since ϵ is arbitrary, and $1 - h(W^\top \mathbf{x})$ specifies an intersection of halfspaces, this would contradict the hardness result of Daniely and Shalev-Shwartz (2016), and therefore prove the theorem. This argument follows from the following chain of inequalities, where $\mathbf{1}$ denotes the indicator function:

$$\begin{aligned}
\Pr\left(\tilde{f}(\mathbf{x}) \ne g(\mathbf{x})\right) &= \Pr(f(\mathbf{x}) > 1/2 \ \wedge \ g(\mathbf{x}) = 1) + \Pr(f(\mathbf{x}) \le 1/2 \ \wedge \ g(\mathbf{x}) = 0) \\
&= \mathbb{E}\left[\mathbf{1}\left(f(\mathbf{x}) > 1/2 \ \wedge \ g(\mathbf{x}) = 1\right)\right] + \mathbb{E}\left[\mathbf{1}\left(f(\mathbf{x}) \le 1/2 \ \wedge \ g(\mathbf{x}) = 0\right)\right] \\
&\le \mathbb{E}\left[4\left(\left(1 - f(\mathbf{x})\right) - g(\mathbf{x})\right)^2\right] + \mathbb{E}\left[4\left(\left(1 - f(\mathbf{x})\right) - g(\mathbf{x})\right)^2\right] \\
&\le 8 \cdot \mathbb{E}\left[\left(\left(1 - f(\mathbf{x})\right) - g(\mathbf{x})\right)^2\right] \ \le \ 8\epsilon.
\end{aligned}$$

■

Proposition 12 *Thm. 11 holds even if we restrict $\mathbf{w}_1, \dots, \mathbf{w}_{n_d}$ to be linearly independent, with $s_{\min}(W) \ge 1$.*

Proof Suppose for the sake of contradiction that there exists an algorithm $\mathcal{A}$ which succeeds for any W as stated above. We will describe how to use $\mathcal{A}$ to get an algorithm which succeeds for any W as described in Thm. 11, hence reaching a contradiction.

Specifically, suppose we have access to samples $(\mathbf{x}, h(W^\top \mathbf{x}))$, where $\mathbf{x}$ is supported on $\{0,1\}^d$, and where W is any matrix as described in Thm. 11. We do the following: We

map every $\mathbf{x}$ to $\tilde{\mathbf{x}} \in \{0,1\}^{d+n_d}$ by $\tilde{\mathbf{x}} = (\mathbf{x}, 0, \ldots, 0)$, run $\mathcal{A}$ on the transformed samples $(\tilde{\mathbf{x}}, h(W^\top \mathbf{x}))$ to get some predictor $\tilde{f} : \{0,1\}^{d+n_d} \mapsto \mathbb{R}$, and return the predictor $f(\mathbf{x}) = \tilde{f}((\mathbf{x}, 0, \ldots, 0))$.

To see why this reduction works, we note that the mapping $\mathbf{x} \mapsto \tilde{\mathbf{x}}$ we have defined, where $\mathbf{x}$ is distributed according to $\mathbf{x}$, induces a distribution $\tilde{\mathcal{D}}$ on $\{0,1\}^{d+n_d}$. Let $\tilde{W}$ be the $(d+n_d)\times n_d$ matrix $[W; I_{n_d}]$ (that is, we add another $n_d \times n_d$ unit matrix below W). We have $\tilde{W}^\top \tilde{W} = W^\top W + I_{n_d}$, so the minimal eigenvalue of $\tilde{W}^\top \tilde{W}$ is at least 1, hence $s_{\min}(\tilde{W}) \geq 1$, so $\tilde{W}$ satisfies the conditions in the proposition. Moreover, the norm of each column of $\tilde{W}$ is larger than the norm of the corresponding column in W by at most 1, so the norm constraint in Thm. 11 still holds. Finally, $\tilde{W}\tilde{\mathbf{x}} = W\mathbf{x}$ for all $\mathbf{x}$, and therefore $h_{\tilde{W}}(\tilde{\mathbf{x}}) = h_W(\mathbf{x})$. Thus, the distribution of $(\tilde{\mathbf{x}}, h(W^\top \mathbf{x})) = (\tilde{\mathbf{x}}, h(\tilde{W}^\top \mathbf{x}))$ (which is used to feed the algorithm $\mathcal{A}$) is a valid distribution corresponding to the conditions of the proposition and Thm. 11 (only in dimension $d + n_d \leq 2d$ instead of d), so $\mathcal{A}$ returns with high probability a predictor $\tilde{f}$ such that

$$\mathbb{E}_{\tilde{\mathbf{x}}\sim\tilde{\mathcal{D}}}\left[\left(\tilde{f}(\tilde{\mathbf{x}}) - h(\tilde{W}^\top \tilde{\mathbf{x}})\right)^2\right] \leq \epsilon.$$

However, $\tilde{f}(\tilde{\mathbf{x}}) = f(\mathbf{x})$, $h(\tilde{W}^\top \tilde{\mathbf{x}}) = h(W^\top \mathbf{x})$, so the returned predictor f satisfies

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\mathbf{x}) - h(W^\top \mathbf{x})\right)^2\right] \leq \epsilon.$$

This contradicts Thm. 11, which states that no efficient algorithm can return such a predictor for *any* sufficiently large dimension d and norm bound $\mathcal{O}(d)$. ■

In the definitions of orthogonal invariance and linear invariance, we only required the invariance to hold with respect to instances $\mathbf{x}_i$ in the data set. A stronger condition is that the invariance is satisfied for any $\mathbf{x} \in \mathbb{R}^d$. However, the following lemma shows that invariance w.r.t. a data set sampled i.i.d. from some distribution is sufficient to imply invariance w.r.t. "nearly all" $\mathbf{x}$ (under the same distribution):

Lemma 13 *Suppose the data set $\{\mathbf{x}_i, y_i\}_{i=1}^m$ is sampled i.i.d. from some distribution (where $\mathbf{x}_i \in \mathbb{R}^d$), then the following holds with probability at least $1-\delta$ for any $\delta \in (0,1)$: For any invertible M and linearly-invariant algorithm (or orthogonal M and orthogonally-invariant algorithm), conditioned on the algorithm's internal randomness, the returned matrices W and W_M (with respect to the original data and the data transformed by M respectively) satisfy*

$$\Pr_{\mathbf{x}}(W_M^\top M\mathbf{x} \neq W^\top \mathbf{x}) \leq \frac{d}{\delta(m+1)}.$$

Proof It is enough to prove that with probability at least $1-\delta$ over the sampling of $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\Pr_{\mathbf{x}}(\mathbf{x} \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)) \leq \frac{d}{\delta(m+1)}. \tag{5}$$

This is because the event $W_M^\top M\mathbf{x}_i = W^\top \mathbf{x}_i$ for all i means that $W_M^\top M\mathbf{x} = W^\top \mathbf{x}$ for any $\mathbf{x}$ in the span of $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Let $\mathbf{x}_1, \ldots, \mathbf{x}_{m+1}$ be sampled i.i.d. according to $\mathcal{D}$. Considering probabilities over this sample, we have

$$\sum_{j=1}^{m+1} \Pr\left(\mathbf{x}_j \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})\right) \;=\; \mathbb{E}\left[\sum_{j=1}^{m+1} \mathbf{1}\left(\mathbf{x}_j \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_{j-1})\right)\right] \leq d, \tag{6}$$

where the latter inequality is because each $\mathbf{x}_j$ is a d-dimensional vector, hence the number of times we can get a vector not in the span of the previous ones is at most d. Moreover, since the vectors are sampled i.i.d, we have

$$\Pr(\mathbf{x}_{j+1} \notin \mathrm{span}(\mathbf{x}_1 \ldots \mathbf{x}_j)) \leq \Pr(\mathbf{x}_{j+1} \notin \mathrm{span}(\mathbf{x}_1 \ldots \mathbf{x}_{j-1})) = \Pr(\mathbf{x}_j \notin \mathrm{span}(\mathbf{x}_1 \ldots \mathbf{x}_{j-1})),$$

so the probabilities in Eq. (6) monotonically decrease with j. Thus, Eq. (6) implies

$$(m+1)\Pr\left(\mathbf{x}_{m+1} \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq d \quad\Rightarrow\quad \Pr\left(\mathbf{x}_{m+1} \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq \frac{d}{m+1}.$$

This is equivalent to

$$\mathbb{E}_{\mathbf{x}_1,\ldots,\mathbf{x}_m\sim\mathcal{D}}\left[\Pr_{\mathbf{x}_{m+1}}\left(\mathbf{x}_{m+1} \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m) | \mathbf{x}_1, \ldots, \mathbf{x}_m\right)\right] \leq \frac{d}{m+1},$$

so by Markov's inequality, with probability at least $1-\delta$ over the sampling of $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\Pr_{\mathbf{x}_{m+1}}\left(\mathbf{x}_{m+1} \notin \mathrm{span}(\mathbf{x}_1, \ldots, \mathbf{x}_m)\right) \leq \frac{d}{\delta(m+1)}.$$

Since $\mathbf{x}_{m+1}$ is sampled independently, Eq. (5) and hence the lemma follows. ■

With these results in hand, we can finally turn to prove Thm. 5. Suppose for the sake of contradiction that there exists an efficient linearly-invariant algorithm $\mathcal{A}$, which for any distribution $\mathcal{D}$ supported on vectors of norm $\mathcal{O}\left(d\sqrt{2dn_d}\right) / \min\{1, s_{\min}(W_\star)\}$, returns w.h.p. a predictor $\mathbf{x} \mapsto f(\tilde{W}^\top \mathbf{x})$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}^\star}\left[\left(f(\tilde{W}^\top \mathbf{x}) - h(W_\star^\top \mathbf{x})\right)^2\right] \leq \epsilon.$$

We will show that the very same algorithm, if given $\mathrm{poly}(d, 1/\epsilon)$ samples, can successfully learn w.r.t. *any* $d \times n_d$ matrix W and any distribution $\mathcal{D}$ satisfying Proposition 12 and Thm. 11, contradicting those results.

In what follows, we assume without loss of generality that f maps to $[0,1]$: If that is not the case, we can simply consider the predictor $\hat{f}(W^\top \mathbf{x})$, where $\hat{f}(z) = \max\{0, \min\{1, f(z)\}\}$, and note that since h returns values in $[0,1]$, then for any input $\mathbf{x}$,

$$((\hat{f}(W^\top \mathbf{x}) - h(W_\star^\top \mathbf{x}))^2 \;\leq\; (f(W^\top \mathbf{x}) - h(W_\star^\top \mathbf{x}))^2,$$

hence the expected squared loss of $\mathbf{x} \mapsto \hat{f}(W^\top \mathbf{x})$ is only smaller than $\mathbf{x} \mapsto f(W^\top \mathbf{x})$

Indeed, let W and $\mathcal{D}$ be an arbitrary matrix and distribution which satisfy the conditions of both Thm. 11 and 12 (namely, $\mathcal{D}$ is a distribution on $\{0,1\}^d$, and $W = [\mathbf{w}_1, \ldots, \mathbf{w}_{n_d}]$

satisfies $\max_i \|\mathbf{w}_i\| \leq \mathcal{O}(d)$ as well as $s_{\min}(W) \geq 1$). We first argue that there exists a $d \times d$ invertible matrix M such that

$$W = M^\top W_\star \quad , \quad \|M\| = \frac{\mathcal{O}(d\sqrt{2n_d})}{\min\{1, s_{\min}(W_\star)\}}. \tag{7}$$

To see this, note that W and $W_\star$ are of the same size and our conditions imply that both of them have full column rank. Thus, we can simply augment them to invertible $d \times d$ matrices $[W\ \hat{W}]$ and $[W_\star\ \hat{W}_\star]$, where the columns of $\hat{W}$ (respectively $\hat{W}_\star$) are an orthonormal basis for the subspace orthogonal to the column space of W (respectively $W_\star$), and choosing $M^\top = [W\ \hat{W}][W_\star\ \hat{W}_\star]^{-1}$. Thus,

$$\|M\| \leq \left\|[W\ \hat{W}]\right\| \cdot \left\|[W_\star\ \hat{W}_\star]^{-1}\right\|. \tag{8}$$

$\left\|[W\ \hat{W}]\right\|$ can be upper bounded by the Frobenius norm, which by the assumption on W from Thm. 11 and the fact that $\hat{W}$ has orthogonal columns, is $\sqrt{\mathcal{O}(d)^2 \cdot n_d + 1 \cdot (d - n_d)} = \mathcal{O}(\sqrt{d^2 n_d}) = \mathcal{O}(d\sqrt{n_d})$. Also, $\left\|[W_\star\ \hat{W}_\star]^{-1}\right\|$ can be upper bounded by the inverse square root of the smallest eigenvalue of

$$[W_\star\ \hat{W}_\star]^\top [W_\star\ \hat{W}_\star] \ = \ \begin{bmatrix} W_\star^\top W_\star & 0 \\ 0 & \hat{W}_\star^\top \hat{W}_\star \end{bmatrix} \ = \ \begin{bmatrix} W_\star^\top W_\star & 0 \\ 0 & I \end{bmatrix} \tag{9}$$

(where I is the unit matrix), which equals $1/\min\{1, s_{\min}(W_\star)\}$. Plugging these bounds into Eq. (8), we get Eq. (7).

Now, consider the following: Suppose we run the algorithm $\mathcal{A}$ using the data points $(M\mathbf{x}_i, h(W_\star^\top(M\mathbf{x}_i)))$, $i = 1, 2, \ldots, m$, where $\mathbf{x}_i$ is sampled from $\mathcal{D}$. Since $\mathbf{x}_i \in \{0,1\}^d$, and $\|\mathbf{x}_i\| \leq \sqrt{d}$, it follows from Eq. (7) that $M\mathbf{x}_i$ is always of norm at most $\|M\|\,\|\mathbf{x}_i\| \leq \|M\|\sqrt{d} = \mathcal{O}(d\sqrt{dn_d})/\min\{1, s_{\min}(W_\star)\}$, and the outputs correspond to the network specified by $W_\star$. Therefore, by assumption, the algorithm $\mathcal{A}$ would return w.h.p. a matrix $\tilde{W}_M$ such that

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_M^\top(M\mathbf{x})) - h(W_\star^\top(M\mathbf{x}))\right)^2\right] \leq \epsilon.$$

By Eq. (7), $W_\star^\top M = (M^\top W_\star)^\top = W^\top$, so this is equivalent to

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f((\tilde{W}_M^\top M\mathbf{x}) - h(W^\top\mathbf{x}))\right)^2\right] \leq \epsilon. \tag{10}$$

Let $\tilde{W}_I$ be the matrix returned by $\mathcal{A}$ if we had fed it with the samples $\{(\mathbf{x}_i, h(W^\top\mathbf{x}_i))\}_{i=1}^m$ (or equivalently, $\{(\mathbf{x}_i, h(W_\star^\top(M\mathbf{x}_i)))\}_{i=1}^m$)[6]. Let $E_\mathbf{x}$ be the event (conditioned on the samples used by the algorithm) that a freshly sampled $\mathbf{x} \sim \mathcal{D}$ satisfies $\tilde{W}_M^\top M\mathbf{x} = \tilde{W}_I^\top\mathbf{x}$. By Lemma 13, w.h.p. over the samples fed to the algorithm, $\Pr_\mathbf{x}(E_\mathbf{x}) = 1 - \mathcal{O}(d/m)$. Therefore,

6. Note that if the algorithm is stochastic, both $\tilde{W}_M$ and $\tilde{W}_I$ are not fixed given the data, but also depend on the algorithm's internal randomness. However, the proof will still follow by conditioning on any possible realization of this randomness.

w.h.p. over the samples $\mathbf{x}_1, \ldots, \mathbf{x}_m$,

$$\begin{aligned}
&\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_I^\top\mathbf{x}) - h(W^\top\mathbf{x})\right)^2\right] \\
&= \Pr_\mathbf{x}(E_\mathbf{x}) \cdot \mathbb{E}_\mathbf{x}\left[\left(f(\tilde{W}_I^\top\mathbf{x}) - h(W^\top\mathbf{x})\right)^2 \middle| E_\mathbf{x}\right] + \Pr_\mathbf{x}(\neg E_\mathbf{x}) \cdot \mathbb{E}_\mathbf{x}\left[\left(f(\tilde{W}_I^\top\mathbf{x}) - h(W^\top\mathbf{x})\right)^2 \middle| \neg E_\mathbf{x}\right] \\
&\leq \Pr_\mathbf{x}(E_\mathbf{x}) \cdot \mathbb{E}_\mathbf{x}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top\mathbf{x})\right)^2 \middle| E_\mathbf{x}\right] + \Pr_\mathbf{x}(\neg E_\mathbf{x}) \cdot 1 \\
&= \mathbb{E}_\mathbf{x}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top\mathbf{x})\right)^2 \mathbf{1}(E_\mathbf{x})\right] + \mathcal{O}\left(\frac{d}{m}\right) \\
&\leq \mathbb{E}_\mathbf{x}\left[\left(f(\tilde{W}_M^\top M\mathbf{x}) - h(W^\top\mathbf{x})\right)^2\right] + \mathcal{O}\left(\frac{d}{m}\right) \quad \leq \quad \epsilon + \mathcal{O}\left(\frac{d}{m}\right),
\end{aligned}$$

where we used the facts that both f and h map to $[0,1]$, a union bound and Eq. (10). Now, recall that $\tilde{W}_I$ refers to the output of the algorithm, given samples $\{(\mathbf{x}_i, h(W^\top\mathbf{x}_i))\}_{i=1}^m$ where $m = \text{poly}(d, 1/\epsilon)$. Thus, we have shown that w.h.p., as long as the algorithm is fed with $m \geq d/\epsilon$ samples[7], the algorithm returns $\tilde{W}_I$ which satisfies

$$\mathbb{E}_{\mathbf{x}\sim\mathcal{D}}\left[\left(f(\tilde{W}_I^\top\mathbf{x}) - h(W^\top\mathbf{x})\right)^2\right] \ = \ \mathcal{O}(\epsilon).$$

This means that the algorithm succesfully learns the hypothesis $\mathbf{x} \mapsto h(W^\top\mathbf{x})$ with respect to the distribution $\mathcal{D}$. Since ϵ is arbitrarily small and $W, \mathcal{D}$ were chosen arbitrarily, the result follows.

5.3 Proof of Thm. 7

To prove the theorem, we will require some tools from Fourier analysis on Euclidean space. We will consider functions from $\mathbb{R}^d$ to the reals $\mathbb{R}$ or complex numbers $\mathbb{C}$, and view them as elements in the Hilbert space $L^2(\mathbb{R}^d)$ of square integrable functions, equipped with the inner product

$$\langle f, g\rangle = \int_\mathbf{x} f(\mathbf{x}) \cdot \overline{g(\mathbf{x})} d\mathbf{x}$$

and the norm $\|f\| = \sqrt{\langle f, f\rangle}$. We use fg or $f \cdot g$ as shorthand for the function $\mathbf{x} \mapsto f(\mathbf{x})g(\mathbf{x})$. Any function $f \in L^2(\mathbb{R}^d)$ has a Fourier transform $\hat{f} \in L^2(\mathbb{R}^d)$, which for absolutely integrable functions can be defined as

$$\hat{f}(\mathbf{w}) = \int \exp\left(-2\pi i\mathbf{x}^\top\mathbf{w}\right) f(\mathbf{x}) d\mathbf{x}, \tag{11}$$

where $\exp(iz) = \cos(z) + i \cdot \sin(z)$, i being the imaginary unit. In the proofs, we will use the following well-known properties of the Fourier transform:

- Linearity: For scalars a, b and functions f, g, $\widehat{af + bg} = a\hat{f} + b\hat{g}$.

7. Even if the algorithm does not require that many samples, we can still artificially add more samples—these are merely used to ensure that its linear invariance is with respect to a sufficiently large data set.

- Isometry: $\langle f, g\rangle = \langle \hat{f}, \hat{g}\rangle$ and $\|f\| = \|\hat{f}\|$.
- Convolution: $\widehat{fg} = \hat{f} * \hat{g}$, where $*$ denotes the convolution operation: $(f*g)(\mathbf{w}) = \int f(\mathbf{z}) \cdot g(\mathbf{w}-\mathbf{z})\, d\mathbf{z}$.

We now turn to prove the theorem. First, we note that for any function q, $\mathbb{E}_{\mathbf{x}\sim\varphi^2}[q(\mathbf{x})] = \sum_i \alpha_i \cdot \mathbb{E}_{\mathbf{x}\sim\varphi_i^2}[q(\mathbf{x})]$. Thus, it is enough to prove the bound in the theorem when φ^2 consists of a single element whose square root is Fourier-concentrated. For a mixture $\varphi^2 = \sum_i \alpha_i \varphi_i^2$, the result follows by applying the bound for each φ_i^2 individually, and using Jensen's inequality.

To simplify notation a bit, we let $\psi_{\mathbf{w}}(\cdot)$ stand for the function $\psi(\langle \mathbf{w}, \cdot\rangle)$, and let $h(\cdot - \mathbf{v})$ (where $\mathbf{v}$ is some vector and h is a function on $\mathbb{R}^d$) stand for the function $\mathbf{x} \mapsto h(\mathbf{x}-\mathbf{v})$. Also, we will use several times the fact that for any two $L^2(\mathbb{R}^d)$ functions h_1, h_2,

$$\langle h_1(\cdot-\mathbf{v}), h_2(\cdot-\mathbf{v})\rangle \;=\; \int h_1(\mathbf{x}-\mathbf{v})\overline{h_2(\mathbf{x}-\mathbf{v})}d\mathbf{x} \;=\; \int h_1(\mathbf{x})\overline{h_2(\mathbf{x})}d\mathbf{x} = \langle h_1, h_2\rangle\,.$$

In other words, inner products (and hence also norms) in $L^2(\mathbb{R}^d)$ are invariant to a shift in coordinates.

The proof is a combination of a few lemmas, presented below.

Lemma 14 *For any $\mathbf{w}$, it holds that $\psi_{\mathbf{w}}\varphi \in L^2(\mathbb{R}^d)$, and satisfies*

$$\widehat{\psi_{\mathbf{w}}\varphi}(\mathbf{x}) \;=\; \sum_{z\in\mathbb{Z}} a_z \cdot \hat{\varphi}(\mathbf{x} - z\mathbf{w})$$

for any $\mathbf{x}$, where $\mathbb{Z}$ is the set of integers and a_z are complex-valued coefficients (corresponding to the Fourier series expansion of ψ, hence depending only on ψ) which satisfy $\sum_{z\in\mathbb{Z}} |a_z|^2 \le 1$.

Proof First, we note that $\psi_{\mathbf{w}}\varphi \in L^2(\mathbb{R}^d)$, since both $\psi_{\mathbf{w}}$ and φ are locally integrable[8] by the theorem's conditions, and satisfy

$$\|\psi_{\mathbf{w}}\varphi\|^2 = \int \psi_{\mathbf{w}}^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} = \int \psi^2(\mathbf{w}^\top\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \le \int \varphi^2(\mathbf{x})d\mathbf{x} = 1 < \infty.$$

As a result, $\widehat{\psi_{\mathbf{w}}\varphi}$ exists as a function in $L^2(\mathbb{R}^d)$. Since ψ is a function of bounded variation, it is equal everywhere to its Fourier series expansion:

$$\psi(x) = \sum_{z\in\mathbb{Z}} a_z \exp\left(2\pi i z x\right),$$

where i is the imaginary unit (note that since ψ is real-valued, the imaginary components eventually cancel out, but it will be more convenient for us to represent the Fourier series in this compact form). By Parseval's identity, $\sum_z |a_z|^2 = \int_{-1/2}^{1/2} \psi^2(x)dx$, which is at most 1 (since $\psi(x) \in [-1,+1]$).

8. Namely, integrable on any compact subset of their domain.

Based on this equation, we have

$$\psi_{\mathbf{w}}(\mathbf{x}) = \psi(\mathbf{w}^\top\mathbf{x}) = \sum_{z\in\mathbb{Z}} a_z \exp\left(2\pi i z \mathbf{w}^\top\mathbf{x}\right).$$

We now wish to compute the Fourier transform of the above[9]. First, we note that the Fourier transform of $\exp(2\pi i \langle \mathbf{v}, \cdot\rangle)$ is given by $\delta(\cdot - \mathbf{v})$, where δ is the Dirac delta function (a so-called generalized function which satisfies $\delta(\mathbf{x}) = 0$ for all $\mathbf{x} \neq \mathbf{0}$, and $\int \delta(\mathbf{x})d\mathbf{x} = 1$). Based on this and the linearity of the Fourier transform, we have that

$$\hat{\psi}_{\mathbf{w}}(\mathbf{x}) \;=\; \sum_{z\in\mathbb{Z}} a_z \cdot \delta\left(\mathbf{x} - z\mathbf{w}\right),$$

and therefore, by the convolution property of the Fourier transform, we have

$$\begin{aligned}\widehat{\psi_{\mathbf{w}}\varphi}(\mathbf{x}) = \left(\hat{\psi}_{\mathbf{w}} * \hat{\varphi}\right)(\mathbf{x}) &= \int \hat{\psi}_{\mathbf{w}}(\mathbf{z}) \cdot \hat{\varphi}(\mathbf{x}-\mathbf{z})d\mathbf{z} \\ &= \sum_{z\in\mathbb{Z}} a_z \cdot \int \delta(\mathbf{z} - z\mathbf{w}) \cdot \hat{\varphi}(\mathbf{x}-\mathbf{z})d\mathbf{z} \\ &= \sum_{z\in\mathbb{Z}} a_z \cdot \hat{\varphi}(\mathbf{x} - z\mathbf{w})\end{aligned}$$

as required. ■

Lemma 15 *For any distinct integers $z_1 \neq z_2$ and any $\mathbf{w}$ such that $\|\mathbf{w}\| = 2r$, it holds that*

$$\langle\; |\hat{\varphi}(\cdot - z_1\mathbf{w})| \;,\; |\hat{\varphi}(\cdot - z_2\mathbf{w})| \;\rangle \;\le\; 2\cdot\epsilon(|z_1 - z_2|r).$$

Proof Let $\Delta = |z_2 - z_1|r$, and $\mathbf{v} = (z_2 - z_1)\mathbf{w}$, so $\mathbf{v}$ is a vector of norm 2Δ. Since the inner product is invariant to shifting the coordinates, we can assume without loss of generality that $z_1 = 0$, and our goal is to bound $\langle|\hat{\varphi}|, |\hat{\varphi}(\cdot - \mathbf{v})|\rangle$.

Using the convention that $\mathbf{1}_{\le\Delta}$ is the indicator of $\{\mathbf{x} : \|\mathbf{x}\| \le \Delta\}$, and $\mathbf{1}_{>\Delta}$ is the indicator for its complement, we have

$$\begin{aligned}\langle|\hat{\varphi}|\,, |\hat{\varphi}(\cdot-\mathbf{v})|\rangle &= \langle|\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot-\mathbf{v})|\,\mathbf{1}_{\le\Delta}\rangle + \langle|\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot-\mathbf{v})|\,\mathbf{1}_{>\Delta}\rangle \\ &= \langle|\hat{\varphi}| \;,\; |\hat{\varphi}(\cdot-\mathbf{v})|\,\mathbf{1}_{\le\Delta}\rangle + \langle|\hat{\varphi}|\,\mathbf{1}_{>\Delta} \;,\; |\hat{\varphi}(\cdot-\mathbf{v})|\rangle \\ &\le \|\hat{\varphi}\|\,\|\hat{\varphi}(\cdot-\mathbf{v})\mathbf{1}_{\le\Delta}\| \;+\; \|\hat{\varphi}\mathbf{1}_{>\Delta}\|\,\|\hat{\varphi}(\cdot-\mathbf{v})\|\,,\end{aligned}$$

where in the last step we used Cauchy-Schwartz. Using the fact that norms and inner products are invariant to coordinate shifting, the above is at most

$$\begin{aligned}&\|\hat{\varphi}\|\,\|\hat{\varphi}\mathbf{1}_{\le\Delta}(\cdot+\mathbf{v})\| + \|\hat{\varphi}\mathbf{1}_{>\Delta}\|\,\|\hat{\varphi}\| \\ &\qquad = \|\hat{\varphi}\|\left(\sqrt{\int |\hat{\varphi}(\mathbf{x})|^2\mathbf{1}_{\|\mathbf{x}+\mathbf{v}\|\le\Delta}d\mathbf{x}} + \sqrt{\int |\hat{\varphi}(\mathbf{x})|^2\mathbf{1}_{>\Delta}(\mathbf{x})d\mathbf{x}}\right).\end{aligned}$$

9. Strictly speaking, this function does not have a Fourier transform in the sense of Eq. (11), since the associated integrals do not converge. However, the function still has a well-defined Fourier transform in the more general sense of a generalized function or distribution (see Hunter and Nachtergaele 2001 for a survey). In the derivation below, we will simply rely on some standard formulas from the Fourier analysis literature, and refer to Hunter and Nachtergaele (2001) for their formal justifications.

By the triangle inequality and the assumption $\|\mathbf{v}\| = 2\Delta$, the event $\|\mathbf{x}+\mathbf{v}\| \leq \Delta$ implies $\|\mathbf{x}\| \geq \Delta$. Therefore, the above can be upper bounded by

$$2\left\|\hat{\varphi}\right\| \sqrt{\int |\hat{\varphi}(\mathbf{x})|^2 \mathbf{1}_{\geq\Delta}(\mathbf{x}) d\mathbf{x}} \;=\; 2\left\|\hat{\varphi}\right\| \cdot \left\|\hat{\varphi} \cdot \mathbf{1}_{\geq\Delta}\right\|.$$

Since φ is Fourier-concentrated, this is at most $2\epsilon(\Delta)\left\|\hat{\varphi}\right\|^2 = 2\epsilon(\Delta)\left\|\varphi\right\|^2 = 2\epsilon(\Delta)$, where we use the isometry of the Fourier transform and the assumption that $\|\varphi\|^2 = \int \varphi^2(\mathbf{x})d\mathbf{x} = 1$. Plugging back the definition of Δ, the result follows. ■

Lemma 16 *It holds that*

$$\sum_{z_1 \neq z_2 \in \mathbb{Z}} |a_{z_1}| \cdot |a_{z_2}| \cdot \epsilon(r|z_1 - z_2|) \;\leq\; 2\sum_{n=1}^{\infty} \epsilon(nr)$$

Proof For simplicity, define $\epsilon'(x) = \epsilon(x)$ for all $x > 0$, and $\epsilon(0) = 0$. Then the expression in the lemma equals

$$\begin{aligned}
&\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}| \cdot |a_{z_2}| \cdot \epsilon'(|z_1 - z_2|r) \\
&= \sum_{z_1,z_2\in\mathbb{Z}} \left(|a_{z_1}|\sqrt{\epsilon'(|z_1 - z_2|r)}\right)\left(|a_{z_2}|\sqrt{\epsilon'(|z_1 - z_2|r)}\right) \\
&\leq \sqrt{\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}|^2 \epsilon'(|z_1 - z_2|r)} \sqrt{\sum_{z_1,z_2\in\mathbb{Z}} |a_{z_2}|^2 \epsilon'(|z_1 - z_2|r)} \\
&= \sum_{z_1,z_2\in\mathbb{Z}} |a_{z_1}|^2 \epsilon'(|z_1 - z_2|r)
\end{aligned}$$

where in the last step we used the fact that the two inner square roots are the same up to a different indexing. Recalling the definition of ϵ' and that $\sum_z |a_z|^2 \leq 1$, the above is at most

$$\begin{aligned}
&\sum_{z_1\in\mathbb{Z}} |a_{z_1}|^2 \sum_{z_2\in\mathbb{Z}} \epsilon'(|z_1 - z_2|r) \;\leq\; \sup_{z_1\in\mathbb{Z}} \sum_{z_2\in\mathbb{Z}} \epsilon'(|z_1 - z_2|r) \\
&= \left(\epsilon'(0) + 2\sum_{n=1}^{\infty} \epsilon'(nr)\right) \;=\; 2\sum_{n=1}^{\infty} \epsilon(nr)\,.
\end{aligned}$$

■

Lemma 17 *For any $g \in L^2(\mathbb{R}^d)$, if $d \geq 40$, and we sample $\mathbf{w}$ uniformly at random from $\{\mathbf{w} : \|\mathbf{w}\| = 2r\}$, it holds that*

$$\mathbb{E}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0 \left\langle g, \hat{\varphi}\right\rangle\right)^2\right] \;\leq\; 10\left\|g\right\|^2 \left(\exp(-d/20) + \sum_{n=1}^{\infty} \epsilon(nr)\right)$$

where a_0 is the coefficient from Lemma 14.

Proof By symmetry, given any function f of $\mathbf{w}$, the expectation $\mathbb{E}_{\mathbf{w}}[f(\mathbf{w})]$ (where $\mathbf{w}$ is uniform on a sphere) can be equivalently written as $\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\mathbb{E}_U[f(U\mathbf{w})] = \mathbb{E}_U\mathbb{E}_{\mathbf{w}\in\mathcal{W}}[f(U\mathbf{w})]$, where U is a rotation matrix chosen uniformly at random (so that for any $\mathbf{w}$, $U\mathbf{w}$ is uniformly distributed on the sphere of radius $\|\mathbf{w}\|$), and $\mathbb{E}_{\mathbf{w}\in\mathcal{W}}$ refers to a uniform distribution of $\mathbf{w}$ over some finite set $\mathcal{W}$ of vectors of norm $2r$. In particular, we will choose $\mathcal{W} = \{\mathbf{w}_1, \ldots, \mathbf{w}_{\lceil\exp(d/20)\rceil}\}$ which satisfies the following:

$$\forall i \;\; \|\mathbf{w}_i\| = 2r \;\;,\;\; \forall i \neq j \;\; |\mathbf{w}_i^\top \mathbf{w}_j| < 2r^2. \tag{12}$$

The existence of such a set follows from standard concentration of measure arguments[10].

Thus, our goal is to bound $\mathbb{E}_U\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\mathbb{E}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right]$. In fact, we will prove the bound stated in the lemma for any U, and will focus on $U = I$ without loss of generality (the argument for other U is exactly the same). First, by applying Lemma 14, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left(\left\langle g, \widehat{\psi_{\mathbf{w}}\varphi}\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right] &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left(\left\langle g, \sum_{z\in\mathbb{Z}} a_z \hat{\varphi}(\cdot - z\mathbf{w})\right\rangle - a_0\left\langle g, \hat{\varphi}\right\rangle\right)^2\right] \\
&= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \sum_{z\in\mathbb{Z}\setminus\{0\}} a_z \hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right].
\end{aligned} \tag{13}$$

For any $\mathbf{w} \in \mathcal{W}$, let

$$A_{\mathbf{w}} = \{\mathbf{x} \in \mathbb{R}^d \;:\; \exists z \in \mathbb{Z}\setminus\{0\} \; s.t. \; \|\mathbf{x} - z\mathbf{w}\| < r\}.$$

In words, each $A_{\mathbf{w}}$ corresponds to the union of open balls of radius r around $\pm\mathbf{w}, \pm 2\mathbf{w}, \pm 3\mathbf{w}\ldots$. An important property of these sets is that they are disjoint: $A_{\mathbf{w}} \cap A_{\mathbf{w}'} = \emptyset$ for any distinct $\mathbf{w}, \mathbf{w}' \in \mathcal{W}$. To see why, note that if there was some $\mathbf{x}$ in both of them, it would imply $\|\mathbf{x} - z_1\mathbf{w}\| < r$ and $\|\mathbf{x} - z_2\mathbf{w}'\| < r$ for some non-zero $z_1, z_2 \in \mathbb{Z}$, hence $\|z_1\mathbf{w} - z_2\mathbf{w}'\| < 2r$ by the triangle inequality. Squaring both sides and performing some simple manipulations (using the facts that $\|\mathbf{w}\| = \|\mathbf{w}'\| = 2r$ and $|z_1|, |z_2| \geq 1$), we would get

$$2|z_1 z_2| \cdot \left|\mathbf{w}^\top \mathbf{w}'\right| > 4r^2(z_1^2 + z_2^2 - 1) \geq 2r^2(z_1^2 + z_2^2) \;\;\Rightarrow\;\; \left|\mathbf{w}^\top \mathbf{w}'\right| \geq r^2\left(\left|\frac{z_1}{z_2}\right| + \left|\frac{z_2}{z_1}\right|\right) \geq 2r^2,$$

where we used the fact that $x + 1/x \geq 2$ for all $x > 0$. This contradicts the assumption on $\mathcal{W}$ (see Eq. 12), and establishes that $\{A_{\mathbf{w}}\}_{\mathbf{w}\in\mathcal{W}}$ are indeed disjoint sets.

We now continue by analyzing Eq. (13). Letting $\mathbf{1}_{A_{\mathbf{w}}}$ be the indicator function to the set $A_{\mathbf{w}}$, and $\mathbf{1}_{A_{\mathbf{w}}^C}$ be the indicator of its complement, and recalling that $(a+b)^2 \leq 2(a^2+b^2)$, we can upper bound Eq. (13) by

$$2 \cdot \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}} \sum_{z\in\mathbb{Z}\setminus\{0\}} a_z \hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right] + 2 \cdot \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_{\mathbf{w}}^C} \sum_{z\in\mathbb{Z}\setminus\{0\}} a_z \hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right]. \tag{14}$$

10. For any two vectors $\mathbf{w}, \mathbf{w}'$ picked uniformly at random from a ball of radius $2r$, $\Pr(|\mathbf{w}^\top \mathbf{w}'| \geq 2r^2) = 2\Pr\left(\left(\frac{1}{2r}\mathbf{w}\right)^\top \left(\frac{1}{2r}\mathbf{w}'\right) > \frac{1}{2}\right) \leq 2\exp\left(-\frac{d}{8}\right)$ (see Boucheron et al. 2013, Section 7.2), so by a union bound, the probability that Eq. (12) is not satisfied is at most $2\lceil\exp(d/20)\rceil^2 \exp(-d/8)$. This can be verified to be strictly less than 1 if $d \geq 40$, hence such a set exists under the lemma's conditions.

We consider each expectation separately. Starting with the first one, we have

$$\begin{aligned}
\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_\mathbf{w}}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right] &= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_\mathbf{w}}\left(\widehat{\psi_\mathbf{w}\varphi} - a_0\hat{\varphi}\right)\right\rangle^2\right] \\
&= \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle \mathbf{1}_{A_\mathbf{w}}g, \widehat{\psi_\mathbf{w}\varphi} - a_0\hat{\varphi}\right\rangle^2\right] \le \mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\|\mathbf{1}_{A_\mathbf{w}}g\right\|^2\left\|\widehat{\psi_\mathbf{w}\varphi} - a_0\hat{\varphi}\right\|^2\right] \\
&\le 2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\|\mathbf{1}_{A_\mathbf{w}}g\right\|^2\left(\left\|\widehat{\psi_\mathbf{w}\varphi}\right\|^2 + \|a_0\hat{\varphi}\|^2\right)\right] \\
&= 2\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\|\mathbf{1}_{A_\mathbf{w}}g\right\|^2\left(\|\psi_\mathbf{w}\varphi\|^2 + |a_0|^2\cdot\|\hat{\varphi}\|^2\right)\right].
\end{aligned}$$

Since we have $\|\hat{\varphi}\| = \|\varphi\| = 1$, $|a_0|^2 \le \sum_z |a_z|^2 \le 1$, and $\|\psi_\mathbf{w}\varphi\|^2 = \int \psi_\mathbf{w}^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \le \int \varphi^2(\mathbf{x})d\mathbf{x} = 1$, the above is at most

$$\begin{aligned}
4\cdot\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\|\mathbf{1}_{A_\mathbf{w}}g\right\|^2\right] &\le \frac{4}{|\mathcal{W}|}\sum_{\mathbf{w}\in\mathcal{W}}\int 1_{A_\mathbf{w}}(\mathbf{x})|g(\mathbf{x})|^2 d\mathbf{x} \\
&= \frac{4}{|\mathcal{W}|}\int\left(\sum_{\mathbf{w}\in\mathcal{W}} 1_{A_\mathbf{w}}(\mathbf{x})\right)|g(\mathbf{x})|^2 d\mathbf{x}.
\end{aligned}$$

Since $A_\mathbf{w}$ are disjoint sets, $\sum_{\mathbf{w}\in\mathcal{W}} \mathbf{1}_{A_\mathbf{w}}(\mathbf{x}) \le 1$ for any $\mathbf{x}$, so the above is at most

$$\frac{4}{|\mathcal{W}|}\int |g(\mathbf{x})|^2 d\mathbf{x} \le 4\exp(-d/20)\,\|g\|^2. \tag{15}$$

We now turn to analyze the second expectation in Eq. (14), namely

$$\mathbb{E}_{\mathbf{w}\in\mathcal{W}}\left[\left\langle g, \mathbf{1}_{A_\mathbf{w}^C}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\rangle^2\right].$$

We will upper bound the expression deterministically for any $\mathbf{w}$, so we may drop the expectation. Applying Cauchy-Schwartz, it is at most

$$\|g\|^2\cdot\left\|\mathbf{1}_{A_\mathbf{w}^C}\sum_{z\in\mathbb{Z}\setminus\{0\}} a_z\hat{\varphi}(\cdot - z\mathbf{w})\right\|^2 = \|g\|^2\left(\sum_{z_1,z_2\in\mathbb{Z}\setminus\{0\}} a_{z_1}\overline{a_{z_2}}\left\langle \mathbf{1}_{A_\mathbf{w}^C}\hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w})\right\rangle\right). \tag{16}$$

We now divide the terms in the sum above to two cases:

- If $z_1 = z_2$, then $\left\langle \mathbf{1}_{A_\mathbf{w}^C}\hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w})\right\rangle$ equals

$$\int \mathbf{1}_{A_\mathbf{w}^C}(\mathbf{x})|\hat{\varphi}(\mathbf{x} - z_1\mathbf{w})|^2 d\mathbf{x}. = \int \mathbf{1}_{A_\mathbf{w}^C}(\mathbf{x} + z_1\mathbf{w})|\hat{\varphi}(\mathbf{x})|^2 d\mathbf{x},$$

and by definition of $A_\mathbf{w}^C$ and the assumption $z_1 \ne 0$, we have $\mathbf{1}_{A_\mathbf{w}^C}(\mathbf{x} + z_1\mathbf{w}) = 1$ only if $\|\mathbf{x}\| \ge r$. Therefore, as φ is Fourier-concentrated, the above is at most

$$\int_{\mathbf{x}:\|\mathbf{x}\|\ge r} |\hat{\varphi}(\mathbf{x})|^2 d\mathbf{x} \le \epsilon^2(r)\cdot\|\hat{\varphi}\|^2 = \epsilon^2(r)\cdot\|\varphi\|^2 = \epsilon^2(r).$$

- If $z_1 \ne z_2$, then by Lemma 15,

$$\left\langle \mathbf{1}_{A_\mathbf{w}^C}\hat{\varphi}(\cdot - z_1\mathbf{w}), \hat{\varphi}(\cdot - z_2\mathbf{w})\right\rangle \le \langle|\hat{\varphi}(\cdot - z_1\mathbf{w})|, |\hat{\varphi}(\cdot - z_2\mathbf{w})|\rangle \le 2\epsilon(|z_1 - z_2|r).$$

Plugging these two cases back into Eq. (16), we get the upper bound

$$\|g\|^2\left(\sum_{z\in\mathbb{Z}\setminus\{0\}} |a_z|^2\epsilon^2(r) + 2\sum_{z_1\ne z_2\in\mathbb{Z}} |a_{z_1}|\cdot|a_{z_2}|\cdot\epsilon(|z_1 - z_2|r)\right).$$

Noting that $\sum_z |a_z|^2 \le 1$, and applying Lemma 16, the above is at most

$$\|g\|^2\left(\epsilon^2(r) + 4\sum_{n=1}^\infty \epsilon(nr)\right) \le 5\,\|g\|^2\sum_{n=1}^\infty \epsilon(nr),$$

where we used the fact that $\epsilon^2(r) \le \epsilon(r) \le \sum_{n=1}^\infty \epsilon(nr)$. Recalling this is an upper bound on the second expectation in Eq. (14), and that the first expectation is upper bounded as in Eq. (15), we get that Eq. (14) (and hence the expression in the lemma statement) is at most

$$10\,\|g\|^2\left(\exp(-d/20) + \sum_{n=1}^\infty \epsilon(nr)\right)$$

as required. ■

With these lemmas in hand, we can now turn to prove the theorem. We have that

$$\mathrm{Var}_{\mathbf{w}^\star}\left[\nabla F_{\mathbf{w}^\star}(\mathbf{v})\right] = \mathbb{E}_{\mathbf{w}^\star}\left\|\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]\right\|^2 \le \mathbb{E}_{\mathbf{w}^\star}\left\|\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbf{p}\right\|^2$$

for any vector $\mathbf{p}$ which is not dependent of $\mathbf{w}^\star$ (this $\mathbf{p}$ will be determined later). Recalling the definition of the objective function F, and letting $\mathbf{g}(\mathbf{x}) = (g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots) = \frac{\partial}{\partial\mathbf{v}}f(\mathbf{v},\mathbf{x})$, the above equals

$$\begin{aligned}
&\mathbb{E}_{\mathbf{w}^\star}\left\|\mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[\left(f(\mathbf{v},\mathbf{x}) - \psi(\mathbf{w}^{\star\top}\mathbf{x})\right)\mathbf{g}(\mathbf{x})\right] - \mathbf{p}\right\|^2 \\
&= \sum_i \mathbb{E}_{\mathbf{w}^\star}\left(\mathbb{E}_{\mathbf{x}\sim\varphi^2}\left[f(\mathbf{v},\mathbf{x})g_i(\mathbf{x}) - \psi(\mathbf{w}^{\star\top}\mathbf{x})g_i(\mathbf{x})\right] - p_i\right)^2 \\
&= \sum_i \mathbb{E}_{\mathbf{w}^\star}\left(\langle\varphi g_i, \varphi f(\mathbf{v},\cdot)\rangle - \langle\varphi g_i, \varphi\psi_{\mathbf{w}^\star}\rangle - p_i\right)^2
\end{aligned}$$

Let us now choose $\mathbf{p}$ so that $p_i = \langle\varphi g_i, \varphi f(\mathbf{v},\cdot)\rangle - \langle\varphi g_i, a_0\varphi\rangle$ (note that this choice is indeed independent of $\mathbf{w}^\star$). Plugging back and applying Lemma 17 (using the L^2 function $\widehat{\varphi g_i}$ for each i), we get

$$\begin{aligned}
\sum_i \mathbb{E}_{\mathbf{w}^\star}\left(\langle\varphi g_i, \varphi\psi_{\mathbf{w}^\star}\rangle - \langle\varphi g_i, a_0\varphi\rangle\right)^2 &= \sum_i \mathbb{E}_{\mathbf{w}^\star}\left(\left\langle\widehat{\varphi g_i}, \widehat{\varphi\psi_{\mathbf{w}^\star}}\right\rangle - \langle\widehat{\varphi g_i}, a_0\widehat{\varphi}\rangle\right)^2 \\
&\le 10\sum_i \|\widehat{\varphi g_i}\|^2\left(\exp(-d/20) + \sum_{n=1}^\infty \epsilon(nr)\right),
\end{aligned}$$

and since

$$\sum_{i=1}^{d} \|\varphi g_i\|^2 \;=\; \sum_{i=1}^{d} \int g_i^2(\mathbf{x})\varphi^2(\mathbf{x})d\mathbf{x} \;=\; \int \|\mathbf{g}(\mathbf{x})\|^2 \varphi^2(\mathbf{x}) \;=\; \mathbb{E}_{\mathbf{x}\sim\varphi^2} \|\mathbf{g}(\mathbf{x})\|^2 \;\leq\; G_{\mathbf{v}}^2,$$

the theorem follows.

5.4 Proof of Thm. 10

For any $\mathbf{w}^*$, we define the oracle $O_{F_{\mathbf{w}^*},\epsilon}$ as follows:

$$O_{F_{\mathbf{w}^*},\epsilon}(\mathbf{v}) = \begin{cases} \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})] & \text{if } |\nabla F_{\mathbf{w}^\star}(\mathbf{v}) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]| \leq \varepsilon \\ \nabla F_{\mathbf{w}^\star}(\mathbf{v}) & \text{otherwise} \end{cases},$$

where the expectation is with respect to a uniform choice of $\mathbf{w}^*$ over vectors of norm $2r$. This is an approximate gradient oracle by definition.

Below, we will prove the theorem statement assuming that the algorithm A is deterministic: Namely, that there is some event holding with probability at least $1-p$ (over the choice of $\mathbf{w}^*$), such that the algorithm's output is some fixed vector $\bar{\mathbf{v}}$ independent of $\mathbf{w}^*$. This can be extended to randomized A, by considering any possible realization of A's random coin flips: Formally, let A_c be the deterministic algorithm, derived from A by fixing its random coin flips to some fixed sequence c. The deterministic proof implies that there is some event E_c and a fixed $\bar{\mathbf{v}}_c$ (independent of $\mathbf{w}^*$) such that $\Pr_{\mathbf{w}^*}(E_c) \geq 1-p$ and $\Pr_{\mathbf{w}^*}(A_C(F_{\mathbf{w}^*}) = \bar{\mathbf{v}}_c | E_c) = 1$ (where we use $A_c(F)$ as shorthand for A_C given the function F). But now, consider the joint probability space over $\mathbf{w}^*$ and random coin flips C, and define the event E as

$$\bigvee_c (C = c \;\wedge\; E_c)$$

(namely, that the event E_c occured for the corresponding realization c of the coin flips). Since this is a disjunction of disjoint events, we have that

$$\Pr(E) = \sum_c \Pr(C = c \wedge E_c) = \sum_c \Pr(C = c)\Pr(E_c | C = c) \geq \sum_c \Pr(C = c)\cdot(1-p) = 1-p,$$

and moreover, if E occurs, then the distribution of $A_C(F_{\mathbf{w}^*})$ is identical to $\bar{\mathbf{v}}_C$ (depending only on the random coin flips C, but not on $\mathbf{w}^*$), which is exactly what the theorem states.

We now return to the proof, assuming A is deterministic. It is enough to show that with probability at least $1-p$, the oracle will only return responses of the form $\mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]$, which is clearly independent of $\mathbf{w}^\star$. Since the algorithm's output can depend on $\mathbf{w}^\star$ only through the oracle responses, this will prove the required result.

The (deterministic) algorithm's first point $\mathbf{v}_1$ is fixed before receiving any information from the oracle, and is therefore independent of $\mathbf{w}^\star$. By Thm. 7, we have that $\mathrm{Var}_{\mathbf{w}^\star}(\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1)) \leq \varepsilon^3$, which by Chebyshev's inequality, implies that

$$\Pr\left(|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v}_1)]| > \varepsilon\right) \;\leq\; \varepsilon,$$

where the probability is over the choice of $\mathbf{w}^\star$. Assuming the event above does not occur, the oracle returns $\mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v})]$, which does not depend on the actual choice of $\mathbf{w}^\star$. This

means that the next point $\mathbf{v}_2$ chosen by the algorithm is again sampled from some fixed distribution $\mathcal{D}_2$, which might depend on the algorithm's internal randomness, but not on $\mathbf{w}^\star$. Again by Thm. 7 and Chebyshev's inequality,

$$\Pr\left(|\nabla F_{\mathbf{w}^\star}(\mathbf{v}_2) - \mathbb{E}_{\mathbf{w}^\star}[\nabla F_{\mathbf{w}^\star}(\mathbf{v}_2)]| > \varepsilon\right) \;\leq\; \varepsilon.$$

Repeating this argument and applying a union bound, it follows that as long as the number of iterations T satisfies $T\varepsilon \leq p$ (or equivalently $T \leq p/\varepsilon$), the oracle reveals no information whatsoever on the choice of $\mathbf{w}^\star$ all point chosen by the algorithm (and hence also its output) are independent of $\mathbf{w}^\star$ as required.

Acknowledgments

This research is supported in part by an FP7 Marie Curie CIG grant, Israel Science Foundation grant 425/13, and the Intel ICRI-CI Institute.

References

Alexandr Andoni, Rina Panigrahy, Gregory Valiant, and Li Zhang. Learning polynomials with neural networks. In *ICML*, 2014.

Sanjeev Arora, Aditya Bhaskara, Rong Ge, and Tengyu Ma. Provable bounds for learning some deep representations. In *ICML*, 2014.

Avrim Blum, Merrick Furst, Jeffrey Jackson, Michael Kearns, Yishay Mansour, and Steven Rudich. Weakly learning dnf and characterizing statistical query learning using fourier analysis. In *STOC*, 1994.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford university press, 2013.

Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

Anna Choromanska, Mikael Henaff, Michael Mathieu, Gérard Ben Arous, and Yann LeCun. The loss surfaces of multilayer networks. In *AISTATS*, 2015.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *COLT*, 2016.

Amit Daniely, Roy Frostig, and Yoram Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *arXiv preprint arXiv:1602.05897*, 2016.

David Donoho and Iain Johnstone. Projection-based approximation and a duality with kernel methods. *The Annals of Statistics*, pages 58–106, 1989.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul): 2121–2159, 2011.

Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for stochastic convex optimization. *arXiv preprint arXiv:1512.09170*, 2015.

Elad Hazan, Kfir Levy, and Shai Shalev-Shwartz. Beyond convexity: Stochastic quasi-convex optimization. In *NIPS*, 2015.

John K. Hunter and Bruno Nachtergaele. *Applied analysis*. World Scientific Publishing, 2001.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.

Majid Janzamin, Hanie Sedghi, and Anima Anandkumar. Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods. *arXiv preprint arXiv:1506.08473*, 2015.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

Adam Klivans and Alexander Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009.

Adam R. Klivans and Pravesh Kothari. Embedding hard learning problems into gaussian space. In *APPROX/RANDOM*, 2014.

Roi Livni, Shai Shalev-Shwartz, and Ohad Shamir. On the computational efficiency of training neural networks. In *NIPS*, 2014.

Peter McCullagh and John Nelder. *Generalized linear models*. CRC press, 1989.

Yurii Nesterov. Minimization methods for nonsmooth convex and quasiconvex functions. *Matekon*, 29:519–531, 1984.

Itay Safran and Ohad Shamir. On the quality of the initial basin in overspecified neural networks. In *ICML*, 2016.

Le Song, Santosh Vempala, John Wilmes, and Bo Xie. On the complexity of learning neural networks. *arXiv preprint arXiv:1707.04615*, 2017.

Daniel Soudry and Yair Carmon. No bad local minima: Data independent training error guarantees for multilayer neural networks. *arXiv preprint arXiv:1605.08361*, 2016.

Yuchen Zhang, Jason Lee, Martin Wainwright, and Michael Jordan. Learning halfspaces and neural networks with random initialization. *arXiv preprint arXiv:1511.07948*, 2015.

Journal of Machine Learning Research 19 (2018) 1-50 Submitted 10/17; Revised 5/18; Published 8/18

Goodness-of-Fit Tests for Random Partitions via Symmetric Polynomials

Chao Gao CHAOGAO@GALTON.UCHICAGO.EDU
Department of Statistics
University of Chicago
Chicago, IL 60637, USA

Editor: Animashree Anandkumar

Abstract

We consider goodness-of-fit tests with i.i.d. samples generated from a categorical distribution $(p_1, ..., p_k)$. For a given $(q_1, ..., q_k)$, we test the null hypothesis whether $p_j = q_{\pi(j)}$ for some label permutation π. The uncertainty of label permutation implies that the null hypothesis is composite instead of being singular. In this paper, we construct a testing procedure using statistics that are defined as indefinite integrals of some symmetric polynomials. This method is aimed directly at the invariance of the problem, and avoids the need of matching the unknown labels. The asymptotic distribution of the testing statistic is shown to be chi-squared, and its power is proved to be nearly optimal under a local alternative hypothesis. Various degenerate structures of the null hypothesis are carefully analyzed in the paper. A two-sample version of the test is also studied.

Keywords: hypothesis testing, elementary symmetric polynomials, Lagrange interpolating polynomials, Vandermonde matrix, minimax optimality

1. Introduction

Consider a categorical distribution parameterized by $(p_1, ..., p_k)$. We have i.i.d. observations $X_1, ..., X_n$ that follow $\mathbb{P}(X_i = j) = p_j$. A classical goodness-of-fit testing problem is to test whether or not $p_j = q_j$ for $j \in [k]$, where $q_1, ..., q_k$ are some given numbers. One solution is given by the famous Pearson's chi-squared test (**?**). In this traditional formulation, it is assumed that the labels $(1, ..., k)$ of $(p_1, ..., p_k)$ correspond to those of $(q_1, ..., q_k)$, so that p_j can be directly compared with q_j for each $j \in [k]$. However, this assumption is not satisfied in some interesting applications. We give three examples below:

1. *Clustering models.* In a typical probabilistic setting of cluster analysis, the event $\{X_i = j\}$ means that the ith item belongs to the jth cluster, and p_j is the population frequency of the jth cluster. Here, the cluster label j does not carry any real meaning, and is present only for notational convenience. In a cluster analysis setting, the underlying object of interest is the partition of the n items instead of the cluster labels. In other words, what really matters to statisticians is the value of $\mathbb{I}\{X_i = X_{i'}\}$ (the indicator function of the event) for every pair $i \neq i'$. Therefore, a clustering model with population frequency $(p_1, ..., p_k)$ is equivalent to that with $(p_{\pi(1)}, ..., p_{\pi(k)})$ with some permutation π.

2. *Word frequency analysis.* Consider two text corpora of two different languages. The word frequencies are denoted by $(p_1, ..., p_k)$ and $(q_1, ..., q_k)$, respectively. An interesting problem in comparative linguistics is to study whether the two languages share common features by comparing $(p_1, ..., p_k)$ with $(q_1, ..., q_k)$. For languages that are not necessarily etymologically related, the correspondence between words of the two languages are usually unclear or unknown. Therefore, a reasonable comparison of word frequencies between two languages can be conducted through comparing $(p_1, ..., p_k)$ with a reordered vector $(q_{\pi(1)}, ..., q_{\pi(k)})$ for some permutation π.

3. *Simple substitution cipher.* In cryptography, a simple substitution cypher changes every character in a message to a different character systematically. Let $\{1, ..., k\}$ be a finite alphabet of characters, and $(Y_1, ..., Y_n)$ denote a message to be encrypted. A simple substitution cypher is defined by a permutation σ on the alphabet $\{1, ..., k\}$. This results in the encrypted message $(X_1, ..., X_n)$ with $X_i = \sigma(Y_i)$ for each $i \in [n]$. Suppose each Y_i is independently distributed by $\mathbb{P}(Y_i = j) = q_j$. Then, each X_i independently follows $\mathbb{P}(X_i = j) = p_j$, where $p_j = q_{\pi(j)}$ with $\pi = \sigma^{-1}$. If only the encrypted message is observed, inference of the probability vector $(q_1, ..., q_k)$ is only possible up to an unknown permutation.

Inspired by the above examples, in this paper, we consider a twist of the traditional formulation of the hypothesis testing problem. We consider the following null hypothesis:

$$H_0 : p_j = q_{\pi(j)}, \quad \text{for some } \pi \in S_k, \tag{1}$$

where $(q_1, ..., q_k)$ is a known vector and S_k is the set of all permutations of $[k]$. This null hypothesis implies that the labels $1, ..., k$ do not have any meaning. For example, the vectors $\left(\frac{1}{2}, \frac{1}{4}, \frac{1}{4}\right)$ and $\left(\frac{1}{4}, \frac{1}{4}, \frac{1}{2}\right)$ are considered equivalent. Given i.i.d. observations $X_1, ..., X_n$, one can immediately define summary statistics $n_j = \sum_{i=1}^n \mathbb{I}\{X_i = j\}$ for $j \in [k]$, which are sufficient. Since the labels of $n_1, ..., n_k$ are irrelevant, these sufficient statistics result in a random partition of the integer n. There are two ways to code such a random partition (**?**): (i) by the order statistics $n_{(1)} \geq n_{(2)} \geq ... \geq n_{(k)}$; (ii) by the numbers of terms of various sizes $m_l = \sum_{j=1}^k \mathbb{I}\{n_j = l\}$ for $l \in [n]$. It is easy to see that $\sum_{l=1}^n m_l = k$ and $\sum_{l=1}^n l m_l = n$. These two representations are equivalent because one can be derived from the other.

Inference of the probability vector $(p_1, ..., p_k)$ up to a label permutation using random partitions have been extensively studied in Bayesian statistics. The problem serves as a foundation for random partition models, cluster analysis and species distribution modeling. Priors that induce various exchangeable properties have been developed for the equivalent class $\{(p_{\pi(1)}, ..., p_{\pi(k)}) : \pi \in S_k\}$. See **????????** and references therein. In this paper, we take a frequentist point of view that is complementary to the Bayesian literature, and we do not treat the equivalent class $\{(p_{\pi(1)}, ..., p_{\pi(k)}) : \pi \in S_k\}$ as random. The theory of hypothesis testing is developed within a frequentist decision-theoretic framework.

With the unknown permutation π in the null hypothesis, the classical chi-squared test by Pearson does not work anymore. Our idea of the test is based on the following class of statistics:

$$\left\{\sum_{j=1}^k f(n_j) : f \in \mathcal{F}\right\}, \tag{2}$$

where $\mathcal{F}$ is the class of all measurable functions. For each $f \in \mathcal{F}$, the distribution of $\sum_{j=1}^k f(n_j)$ is identical for $p_j = q_{\pi(j)}$ with any $\pi \in S_k$. This is because (2) is a class of statistics that are invariant to the label permutation π. That is, $\sum_{j=1}^k f(n_j) = \sum_{j=1}^k f(n_{\pi(j)})$ for any $\pi \in S_k$. Moreover, it is easy to see that these statistics are all functions of the random partition because $\sum_{j=1}^k f(n_j) = \sum_{j=1}^k f(n_{(j)})$.

Choosing an appropriate class of f's is important. We propose to use k functions $f_1, ..., f_k$ that satisfy the *identifiability* and the *orthogonality* conditions. The identifiability condition requires that the k equations $\sum_{j=1}^k f_l(p_j) = \sum_{j=1}^k f_l(q_j)$ for $l \in [k]$ hold if and only if $p_j = q_{\pi(j)}$ for some $\pi \in S_k$. With this condition, testing whether the null hypothesis holds is equivalent to testing whether the k equations hold. The orthogonality condition requires that the k vectors $(f_l'(q_1), ..., f_l'(q_k))^T$ for $l \in [k]$ are orthogonal to each other. Intuitively speaking, this condition ensures that the information carried by the k statistics $\sum_{j=1}^k f_l(n_j)$ for $l \in [k]$ are mutually exclusive, which is a key ingredient that leads to optimal power under a local alternative.

In this paper, we choose $f_1, ..., f_k$ to be indefinite integrals of Lagrange interpolation polynomials. The choice of these polynomials satisfies the above-mentioned *identifiability* and *orthogonality* conditions. We prove that the testing statistic constructed from the k functions is asymptotically distributed by a chi-squared distribution. Moreover, we show that the power of the test is nearly optimal under a local alternative hypothesis within a decision-theoretic framework.

Our approach that uses symmetric polynomials bypasses the problem of unknown permutation π. It falls into the general umbrella of methods of moments, which are commonly used for problems that impose equivalence relations to the signals through the action of a group of transformations. For example, various method-of-moments techniques have been applied to problems including Gaussian mixture models (**?**), mixed membership models (**?**), dictionary learning (**?**), topic models (**??**) and multi-reference alignment (**?**). Recently, this idea was also applied to the problems of network testing by **??**, where the group action there is row and column permutations of the adjacency matrix of a random network.

The rest of the paper is organized as follows. In Section 2, we introduce definitions of some useful symmetric polynomials and the related Vandermonde matrix. Before getting into the testing problem for random partitions, we first solve an easier version of the problem with Gaussian observations in Section 3 and Section 4. The test using random partitions is given in Section 5. The optimality of our test is discussed in Section 6. In Section 8, we consider a two-sample version of the problem. Numerical experiments of the proposed testing procedures are given in Section 7. Finally, Section 9 is a discussion section, where we briefly analyze the property of the test on the boundary of degeneracy and discuss some open problems. The proofs of all results in the paper are given in Section 10.

We close this section by introducing the notation used in the paper. For $a, b \in \mathbb{R}$, let $a \vee b = \max(a, b)$ and $a \wedge b = \min(a, b)$. For an integer m, $[m]$ denotes the set $\{1, 2, ..., m\}$. Given a set S, $|S|$ denotes its cardinality, and $\mathbb{I}_S$ is the associated indicator function. We use $\mathbb{P}$ and $\mathbb{E}$ to denote generic probability and expectation whose distribution is determined from the context. The noncentral chi-squared distribution with degrees of freedom k and noncentrality parameter δ^2 is denoted as χ^2_{k,δ^2}. We will also use χ^2_{k,δ^2} for the associated random variables.

2. Symmetric Polynomials and Vandermonde Matrix

Define a polynomial with roots $\mu_1, ..., \mu_k \in \mathbb{R}$ by

$$f(t) = \prod_{j=1}^k (t - \mu_j).$$

It can be organized as

$$f(t) = \sum_{j=0}^k (-1)^{k-j} e_{k-j}(\mu_1, ..., \mu_k) t^j. \tag{3}$$

The coefficient before t^j is $(-1)^{k-j} e_{k-j}(\mu_1, ..., \mu_k)$, and $e_{k-j}(\mu_1, ..., \mu_k)$ is called the elementary symmetric polynomial. For $l \in \{1, ..., k\}$, the lth elementary symmetric polynomial is

$$e_l(\mu_1, ..., \mu_k) = \sum_{1 \le j_1 < \cdots < j_l \le k} \mu_{j_1} \cdots \mu_{j_l}.$$

When $l = 0$, we use the convention $e_0(\mu_1, ..., \mu_k) = 1$.

The elementary symmetric polynomials can be efficiently calculated through Newton's identities. Define the lth power sum

$$p_l(\mu_1, ..., \mu_k) = \sum_{j=1}^k \mu_j^l. \tag{4}$$

Newton's identities can be summarized through the formula

$$e_l(\mu_1, ..., \mu_k) = \frac{1}{l} \sum_{j=1}^l (-1)^{j-1} e_{l-j}(\mu_1, ..., \mu_k) p_j(\mu_1, ..., \mu_k), \tag{5}$$

for $l = 1, ..., k$.

Finally, we introduce an interesting relation between elementary symmetric polynomials and Vandermonde matrix (see Chapter 0.9.11 of **?**). Given $\mu_1, ..., \mu_k$ that take k distinct values, define a matrix $E(\mu_1, ..., \mu_k) \in \mathbb{R}^{k \times k}$, whose (j, l)th entry is

$$(-1)^{j-1} \frac{e_{k-j}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k)}{\prod_{j \in [k] \setminus \{l\}} (\mu_j - \mu_l)}. \tag{6}$$

The Vandermonde matrix $V(\mu_1, ..., \mu_k) \in \mathbb{R}^{k \times k}$ has μ_j^{l-1} on its (j, l)th entry. Interestingly, we have

$$E(\mu_1, ..., \mu_k) V(\mu_1, ..., \mu_k) = V(\mu_1, ..., \mu_k) E(\mu_1, ..., \mu_k) = I_k. \tag{7}$$

This relation implies a formula for the determinant of $E(\mu_1, ..., \mu_k)$:

$$\det(E(\mu_1, ..., \mu_k)) = \frac{1}{\det(V(\mu_1, ..., \mu_k))} = \frac{1}{\prod_{1 \le j < l \le k} (\mu_l - \mu_j)}. \tag{8}$$

3. The Gaussian Case

Before working with categorical distributions, we first study data generated from a Gaussian distribution. This allows us to grasp the mathematical essence of the problem without dealing with the dependence and heteroskedasticity of categorical distributions. We consider a Gaussian random vector $X \sim N(\theta, n^{-1}I_k)$. The mean vector $\theta \in \mathbb{R}^k$ consists of k numbers $\theta_1, ..., \theta_k$. Throughout the paper, we assume $k \geq 2$ and it is a constant that does not vary with n. We would like to test whether the k numbers are identical to $\mu_1, ..., \mu_k$ after some permutation of labels. To be rigorous, introduce a distance between two vectors θ and μ,

$$\ell(\theta, \mu) = \min_{\pi \in S_k} \sqrt{\sum_{j=1}^{k} (\theta_j - \mu_{\pi(j)})^2},$$

where S_k is the set of all permutations on $[k]$. Then, the hypothesis testing problem is

$$H_0 : \ell(\theta, \mu) = 0, \quad H_1 : \ell(\theta, \mu) > 0.$$

Throughout this section, we assume $\min_{j \neq l} |\mu_j - \mu_l| > 0$. The case $\min_{j \neq l} |\mu_j - \mu_l| = 0$ is degenerate and will be studied in the next section.

We use the notation μ_π to denote a k-dimensional vector whose jth entry is $\mu_{\pi(j)}$. Then, the null hypothesis can also be written as

$$\theta \in \{\mu_\pi : \pi \in S_k\}.$$

In other words, there is an equivalent class of probability distributions $\{N(\mu_\pi, n^{-1}I_k) : \pi \in S_k\}$. Thus, it is natural to consider summary statistics whose distributions are invariant under this equivalent class. This leads to the class of summary statistics

$$\left\{ \sum_{j=1}^{k} f(X_j) : f \in \mathcal{F} \right\},$$

where $\mathcal{F}$ is the set of all measurable functions. For $X \sim N(\theta, n^{-1}I_k)$, it is easy to see that the distribution of $\sum_{j=1}^{k} f(X_j)$ only depends on the equivalent class $\{N(\theta_k, n^{-1}I_k) : \pi \in S_k\}$. This fact holds for an arbitrary $f \in \mathcal{F}$.

Since the degree of freedom of the null hypothesis is k, our strategy is to construct a testing procedure based on $\left\{\sum_{j=1}^{k} f_l(X_j) : l \in [k]\right\}$. In other words, we need to choose the k functions $f_1(\cdot), ..., f_k(\cdot)$. The following two conditions are proposed:

1. **Identifiability**. Assume $\min_{j \neq l} |\mu_j - \mu_l| > 0$. Then the equations

$$\sum_{j=1}^{k} f_l(\theta_j) = \sum_{j=1}^{k} f_l(\mu_j), \quad l = 1, ..., k, \tag{9}$$

hold, if and only if $\ell(\theta, \mu) = 0$.

2. **Orthogonality**. Assume $\min_{j \neq l} |\mu_j - \mu_l| > 0$. Then for any $l, h \in [h]$,

$$\sum_{j=1}^{k} f_l'(\mu_j) f_h'(\mu_j) = \begin{cases} 1, & l = h, \\ 0, & l \neq h. \end{cases}$$

We give a few remarks regarding the two conditions. The first condition of identifiability is natural. It is required by the structure of the problem, and is necessary for the test to have power under the alternative hypothesis. The second condition implies information independence among the k summary statistics.

The k functions we propose to satisfy the two conditions are

$$f_l(t) = \frac{\int \prod_{j \in [k] \setminus \{l\}} (t - \mu_j)}{\prod_{j \in [k] \setminus \{l\}} (\mu_l - \mu_j)}, \quad l = 1, ..., k. \tag{10}$$

The derivatives $f_l'(t) = \frac{\prod_{j \in [k] \setminus \{l\}} (t - \mu_j)}{\prod_{j \in [k] \setminus \{l\}} (\mu_l - \mu_j)}$ are called Lagrange interpolating polynomials, and it is easy to check that the second condition of orthogonality holds. Now we check the first condition of identifiability. By (3), we have

$$\prod_{j \in [k] \setminus \{l\}} (t - \mu_j) = \sum_{j=0}^{k-1} (-1)^{k-1-j} e_{k-1-j}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k) t^j.$$

This implies

$$f_l(t) = \sum_{j=1}^{k} (-1)^{k-j} \frac{e_{k-j}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k)}{\prod_{j \in [k] \setminus \{l\}} (\mu_l - \mu_j)} \frac{t^j}{j}. \tag{11}$$

Therefore, the equations (9) can be written as

$$\sum_{j=1}^{k} (-1)^{k-j} \frac{e_{k-j}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k)}{\prod_{j \in [k] \setminus \{l\}} (\mu_l - \mu_j)} \Delta_j = 0, \quad l = 1, ..., k.$$

where $\Delta_j = \frac{1}{j} \sum_{h=1}^{k} \theta_h^j - \frac{1}{j} \sum_{h=1}^{k} \mu_h^j$. In view of the definition of the matrix $E(\mu_1, ..., \mu_k)$ in (6), we have a compact organization of the equations

$$E(\mu_1, ..., \mu_k)\Delta = 0.$$

When the assumption $\min_{j \neq l} |\mu_j - \mu_l| > 0$ holds, the matrix $E(\mu_1, ..., \mu_k)$ has full rank and is invertible according to (7) and (8), which immediately implies $\Delta = 0$. Equivalently,

$$p_j(\theta_1, ..., \theta_k) = p_j(\mu_1, ..., \mu_k), \quad j = 1, ..., k.$$

The definition of the power sum $p_j(\cdots)$ is given in (4). By Newton's identities (5), we have

$$e_j(\theta_1, ..., \theta_k) = e_j(\mu_1, ..., \mu_k), \quad j = 1, ..., k.$$

Finally, the relation between elementary symmetric polynomials and roots in (3) implies that $\prod_{j=1}^{k} (t - \theta_j)$ and $\prod_{j=1}^{k} (t - \mu_j)$ are the same polynomials. Hence, we obtain the conclusion $\ell(\theta, \mu) = 0$. The other direction trivially holds. This verifies the condition of identifiability for the functions $f_1, ..., f_k$.

Remark 1 *The computation of the statistic $\sum_{j=1}^k f_l(X_j)$ for each $l \in [k]$ is straightforward, thanks to the formula (11). According to (11), we can write*

$$\sum_{j=1}^k f_l(X_j) = \sum_{i=1}^k (-1)^{k-i} \frac{e_{k-i}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k)}{\prod_{i \in [k] \setminus \{l\}} (\mu_l - \mu_i)} \frac{\sum_{j=1}^k X_j^i}{i}.$$

In other words, $\sum_{j=1}^k f_l(X_j)$ is a linear combination of empirical moments $\{\sum_{j=1}^k X_j^i : i \in [k]\}$. To compute the elementary symmetric polynomial $e_{k-i}(\mu_1, ..., \mu_{l-1}, \mu_{l+1}, ..., \mu_k)$ in the coefficient, one can recursively apply Newton's identities (5). The overall complexity of computing $\sum_{j=1}^k f_l(X_j)$ requires $O(k^2)$ products.

We propose the testing statistic

$$T = n \sum_{l=1}^k \left(\sum_{j=1}^k f_l(X_j) - \sum_{j=1}^k f_l(\mu_j) \right)^2. \tag{12}$$

When the value of T is large, the equations (9) are unlikely to hold. Thus, the null hypothesis will be rejected when T exceeds some threshold. The asymptotic distribution of T can be derived under the null hypothesis.

Condition A 1 *Assume $\mu_1, ..., \mu_k$ are k different numbers that do not vary with n.*

Some possible extensions beyond Condition A will be discussed in Section 6.

Theorem 2 *Under Condition A, $T \rightsquigarrow \chi_k^2$ as $n \to \infty$ under the null hypothesis.*

For a chi-squared random variable χ_k^2, define a number $\chi_k^2(\alpha)$ such that

$$\mathbb{P}\left(\chi_k^2 \le \chi_k^2(\alpha)\right) = 1 - \alpha.$$

Then, a testing function is

$$\phi_\alpha = \mathbb{I}\left\{T > \chi_k^2(\alpha)\right\}.$$

By Theorem 2, its asymptotic Type-1 error is α. The next result characterizes the regime where the asymptotic power of the test tends to 1. It is a consequence of the fact that the functions $f_1, ..., f_k$ satisfy the identifiability condition.

Theorem 3 *Under Condition A, the following two statements are equivalent*

1. $\lim_{n\to\infty} \sqrt{n}\ell(\theta, \mu) = \infty$;
2. $\lim_{n\to\infty} \mathbb{P}_\theta\left(T > \chi_k^2(\alpha)\right) = 1$, *for any constant* $\alpha \in (0, 1)$,

where the probability $\mathbb{P}_\theta$ denotes $N(\theta, n^{-1} I_k)$.

Theorem 3 shows that $\lim_{n\to\infty} \sqrt{n}\ell(\theta, \mu) = \infty$ is the necessary and sufficient condition for the asymptotic power of the test to be one. For a local alternative such that $\sqrt{n}\ell(\theta, \mu) = O(1)$, the test will have a non-trivial power between 0 and 1. This contiguous regime will be studied in Section 6.

4. Degeneracy of the Problem

In the last section, we construct a chi-squared test under the assumption that $\min_{j\neq l} |\mu_j - \mu_l| > 0$. When $\min_{j\neq l} |\mu_j - \mu_l| = 0$, the identifiability condition of the functions $f_1, ..., f_k$ defined in (10) does not hold. We need to construct summary statistics based on new functions in this degenerate case.

Assume there is a partition of the set $[k]$. That is, for some $d \le k$, we have $\cup_{h=1}^d \mathcal{C}_h = [k]$, and for any $g, h \in [d]$ such that $g \neq h$, $\mathcal{C}_g \cap \mathcal{C}_h = \varnothing$. We assume

$$\mu_j = \nu_h, \text{ for all } j \in \mathcal{C}_h.$$

Moreover, we require that $\min_{g\neq h} |\nu_g - \nu_h| > 0$. To motivate the appropriate functions that we will propose, we consider two extreme cases. The first case is when $d = k$. Then, the condition $\min_{j\neq l} |\mu_j - \mu_l| > 0$ still holds, and we can still use the functions $f_1, ..., f_k$ defined in (10). The second case is when $d = 1$. This implies $\mu_1 = \mu_2 = \cdots = \mu_k = \nu_1$. Then, we can use the function

$$g(t) = (t - \nu_1)^2. \tag{13}$$

This leads to an obvious chi-squared statistic $T_g = n \sum_{j=1}^k g(X_j)$.

For a general d, we need to borrow ideas from both extreme cases. We define functions $f_1, ..., f_d$ that are modifications from (10). Define

$$f_h(t) = \frac{\int \prod_{g\in[d]\setminus\{h\}} (t - \nu_g)}{\prod_{g\in[d]\setminus\{h\}} (\nu_h - \nu_g)}, \quad h = 1, ..., d. \tag{14}$$

We also need another function to ensure identifiability. Define

$$g(t) = \frac{\prod_{g=1}^d (t - \nu_g)^2}{\sum_{g=1}^d \prod_{h\in[d]\setminus\{g\}} (t - \nu_h)^2}. \tag{15}$$

The function $g(t)$ is well defined when $d \ge 2$. When $d = 1$, we use the definition given by (13). The following proposition shows that the functions $f_1, ..., f_d, g$ together ensure identifiability via the equations

$$\sum_{j=1}^k f_h(\theta_j) = \sum_{j=1}^k f_h(\mu_j), \quad h = 1, ..., d, \tag{16}$$

and

$$\sum_{j=1}^k g(\theta_j) = \sum_{j=1}^k g(\mu_j). \tag{17}$$

Proposition 4 *Assume $\min_{g\neq h} |\nu_g - \nu_h| > 0$. We have the following conclusions.*

1. *When $d = 1$, the equation (17) holds if and only if $\ell(\theta, \mu) = 0$.*
2. *When $2 \le d \le k - 1$, the equations (16) and (17) hold if and only if $\ell(\theta, \mu) = 0$.*
3. *When $d = k$, the equation (16) holds if and only if $\ell(\theta, \mu) = 0$.*

The first conclusion of the above proposition is obvious. The last conclusion is proved in Section 3. Here we show the second conclusion. Using a similar argument that we used in Section 3, the equations (16) can be written as

$$E(\nu_1, ..., \nu_d)\Delta = 0,$$

where $\Delta \in \mathbb{R}^d$ is a vector with the hth entry being $\Delta_h = \frac{1}{h}\sum_{j=1}^k \theta_j^h - \frac{1}{h}\sum_{j=1}^k \mu_j^h$. In other words, we have

$$\sum_{j=1}^k \theta_j^h = \sum_{j=1}^k \mu_j^h, \quad \text{for } h = 1, ..., d.$$

The equation (17) immediately implies that each θ_j only takes value in the set $\{\nu_1, ..., \nu_d\}$. Therefore, there exists a partition $[k] = \cup_{h=1}^d \mathcal{D}_h$ such that $\mathcal{D}_g \cap \mathcal{D}_h = \varnothing$ for all $g \neq h$, and $\theta_j = \nu_g$ for all $j \in \mathcal{D}_g$. This leads to

$$\sum_{j=1}^k \theta_j^h = \sum_{g=1}^d |\mathcal{D}_g| \nu_g^h.$$

We also have

$$\sum_{j=1}^k \mu_j^h = \sum_{g=1}^d |\mathcal{C}_g| \nu_g^h.$$

Hence, we obtain the equations

$$\sum_{g=1}^d |\mathcal{D}_g| \nu_g^h = \sum_{g=1}^d |\mathcal{C}_g| \nu_g^h, \quad \text{for } h = 0, 1, ..., d-1.$$

The equation for $h = 0$ holds because $\sum_{g=1}^d |\mathcal{D}_g| = \sum_{g=1}^d |\mathcal{C}_g| = k$. Again, with matrix notation, these equations can be written as $V(\nu_1, ..., \nu_d)r = 0$, where $V(\nu_1, ..., \nu_d)$ is the Vandermonde matrix, and $r \in \mathbb{R}^d$ is a vector with its gth entry being $r_g = |\mathcal{D}_g| - |\mathcal{C}_g|$. When $\min_{g \neq h} |\nu_g - \nu_h| > 0$ holds, $V(\nu_1, ..., \nu_d)$ has full rank, which leads to $|\mathcal{D}_g| = |\mathcal{C}_g|$ for all $g = 1, ..., d$. Finally, we can conclude that $\ell(\theta, \mu) = 0$. The other direction is obvious.

The above proof actually shows that the function f_d is not needed when $d \leq k - 1$. The equation with $h = d$ in (16) is redundant for identifiability. The second conclusion of Proposition 4 would still hold without it. However, we still keep it for the convenience of analyzing the proposed test.

We propose two testing statistics. Define

$$T_f = n \sum_{h=1}^d \frac{1}{|\mathcal{C}_h|} \left(\sum_{j=1}^k f_h(X_j) - \sum_{j=1}^k f_h(\mu_j) \right)^2, \tag{18}$$

and

$$T_g = n \sum_{j=1}^k g(X_j). \tag{19}$$

We present asymptotic distributions of T_f and T_g. Since the case $d = 1$ is trivial, we only present results for $d \geq 2$.

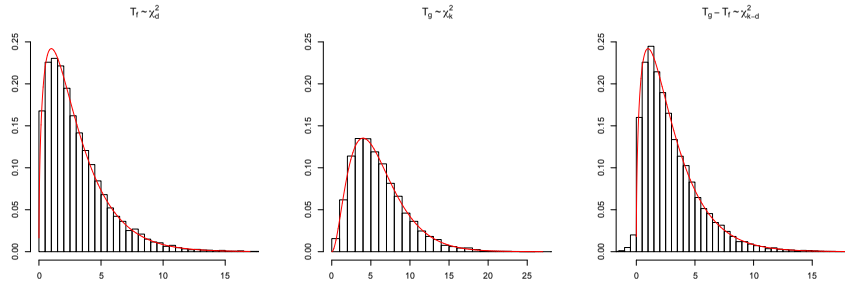


Figure 1: histograms of testing statistics with $\mu = (1, 3, 3, 3, 5, 5)$, $k = 6$, $d = 3$ and $n = 200$.

Condition B 1 *Assume $\mu_1, ..., \mu_k$ are k numbers that do not vary with n. Moreover, $\mu_j = \nu_h$ for all $j \in \mathcal{C}_h$, $h \in [d]$.*

Theorem 5 *Under Condition B, $T_f \rightsquigarrow \chi^2_d$, $T_g \rightsquigarrow \chi^2_k$ and $T_g - T_f \rightsquigarrow \chi^2_{k-d}$ as $n \to \infty$ under the null hypothesis.*

Interestingly, Theorem 5 exhibits an analysis-of-variance type of result. The three statistics all exhibit asymptotic chi-square distributions (see Figure 1). The statistic T_g dominates T_f in probability under the null hypothesis. An analogous analysis-of-variance type of result continues to hold under a local alternative (see Theorem 23). We define the testing function as

$$\phi_\alpha = \mathbb{I}\{T_f > \chi^2_d(\alpha)\} \vee \mathbb{I}\{T_g > \chi^2_k(\alpha)\},$$

where we use the notation $a \vee b$ for $\max(a, b)$. Since under the null hypothesis, $T_g \geq T_f$ in probability, the asymptotic Type-1 error is just the probability of the event $\{T_g > \chi^2_k(\alpha)\}$, which tends to α as $n \to \infty$. In fact, as we will show later in Section 6, the behavior of the testing function mainly depends on the statistic T_g in the contiguous neighborhood of the null hypothesis. The statistic T_f helps to ensure that the testing function has asymptotic power 1 as soon as $\sqrt{n}\ell(\theta, \mu) \to \infty$. Without T_f, the identifiability property of the test established in Proposition 4 would break down, and the test would lose power outside of the contiguous neighborhood of the null hypothesis. The next theorem rigorously establishes this fact.

Theorem 6 *Under Condition B, the following two statements are equivalent*

1. $\lim_{n\to\infty} \sqrt{n}\ell(\theta, \mu) = \infty$;
2. $\lim_{n\to\infty} \mathbb{P}_\theta \left(T_f > \chi^2_d(\alpha) \text{ or } T_g > \chi^2_k(\alpha) \right) = 1$, *for any constant* $\alpha \in (0, 1)$,

where the probability $\mathbb{P}_\theta$ denotes $N(\theta, n^{-1}I_k)$.

5. The Case of Categorical Distribution

Now we are ready to transfer our wisdom from Gaussian distribution to categorical distribution. Consider i.i.d. observations $X_1, ..., X_n$ from a categorical distribution $(p_1, ..., p_k)$. To be specific, for each $i \in [n]$ and $j \in [k]$, $\mathbb{P}(X_i = j) = p_j$. Throughout this section, we use $\mathbb{P}_p$ to denote the probability distribution of $X_1, ..., X_n$. We would like to test whether the k numbers $p_1, ..., p_k$ are identical to some given $q_1, ...q_k$ after a permutation of labels. Introduce a distance between the two vectors p and q,

$$\ell(p, q) = \min_{\pi \in S_k} 2\sqrt{\sum_{j=1}^{k} (\sqrt{p_j} - \sqrt{q_{\pi(j)}})^2}. \tag{20}$$

The hypothesis testing problem is

$$H_0 : \ell(p, q) = 0, \quad H_1 : \ell(p, q) > 0.$$

For each $j \in [k]$, define

$$\hat{p}_j = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{X_i = j\}.$$

Pearson's chi-squared test (**?**) is defined as $\chi^2 = n \sum_{j=1}^{k} \frac{(\hat{p}_j - q_j)^2}{q_j}$, which is asymptotically distributed as χ^2_{k-1} when $p = q$. However, this test only works when the null hypothesis is simple. Here, our null hypothesis is composite, and there is uncertainty from the underlying permutation of the labels.

Our idea is to borrow the solution for the Gaussian case in Section 3. Intuitively, the vector $(\hat{p}_1, ..., \hat{p}_k)$ is asymptotically Gaussian after some normalization. However, the normalization step brings extra difficulty for this problem. In the definition of Pearson's chi-squared test, each $\hat{p}_j$ is normalized by $\sqrt{q_j}$ because of the heteroskedasticity and dependence structure of the vector $(\hat{p}_1, ..., \hat{p}_k)$. This normalization does not work in our setting because the underlying label is not given, and we do not know which $\sqrt{q_j}$ to use. To overcome this issue, we adopt the technique of variance-stabilizing transformation (**?**), and directly work with $\sqrt{\hat{p}_j}$.

This leads to a modification of the definition of the function $f_l(\cdot)$, and the new definition is given by

$$f_l(t) = \frac{\int \prod_{j \in [k] \setminus \{l\}} (\sqrt{t} - \sqrt{q_j})}{\prod_{j \in [k] \setminus \{l\}} (\sqrt{q_l} - \sqrt{q_j})}, \quad l = 1, ..., k. \tag{21}$$

The testing statistic is

$$T = 4n \sum_{l=1}^{k} \left(\sum_{j=1}^{k} f_l(\hat{p}_j) - \sum_{j=1}^{k} f_l(q_j) \right)^2. \tag{22}$$

Similar to the discussion in Section 3, when the value of T is large, the equations (9) are unlikely to hold. Thus, the null hypothesis will be rejected when T exceeds some threshold. The asymptotic distribution of T can be derived under the null hypothesis.

Condition C 1 *Assume $q_1, ..., q_k$ are k different numbers in $(0, 1)$ that do not vary with n.*

Some possible extensions beyond Condition C will be discussed in Section 6.

Theorem 7 *Under Condition C, $T \rightsquigarrow \chi^2_{k-1}$ as $n \to \infty$ under the null hypothesis.*

For a chi-squared random variable χ^2_{k-1}, define a number $\chi^2_{k-1}(\alpha)$ such that

$$\mathbb{P}\left(\chi^2_{k-1} \le \chi^2_{k-1}(\alpha)\right) = 1 - \alpha.$$

Then, a testing function is

$$\phi_\alpha = \mathbb{I}\left\{T > \chi^2_{k-1}(\alpha)\right\}.$$

By Theorem 7, its asymptotic Type-1 error is α. The next result characterizes the regime where the asymptotic power of the test tends to 1. It is a consequence of the fact that the functions $f_1, ..., f_k$ satisfy the identifiability condition, though with slightly different definitions.

Theorem 8 *Under Condition C, the following two statements are equivalent*

1. $\lim_{n\to\infty} \sqrt{n}\ell(p, q) = \infty$;
2. $\lim_{n\to\infty} \mathbb{P}_p\left(T > \chi^2_{k-1}(\alpha)\right) = 1$, *for any constant* $\alpha \in (0, 1)$,

where the probability $\mathbb{P}_p$ is defined in the beginning of this section.

Next, we study the degenerate case where the k numbers $q_1, ..., q_k$ only take d values. There is a partition $[k] = \cup_{h=1}^{d} \mathcal{C}_h$ such that for any $g \neq h$, $\mathcal{C}_g \cap \mathcal{C}_h = \varnothing$. We assume the following condition.

Condition D 1 *Asume $q_1, ..., q_k$ are k numbers in $(0, 1)$ that do not vary with n. Moreover, $q_j = r_h$ for all $j \in \mathcal{C}_h$, $h \in [d]$.*

The approach we take is similar to that in Section 4, assisted with the technique of variance-stabilizing transformation. Define

$$f_h(t) = \frac{\int \prod_{g \in [d] \setminus \{h\}} (\sqrt{t} - \sqrt{r_g})}{\prod_{g \in [d] \setminus \{h\}} (\sqrt{r_h} - \sqrt{r_g})}, \quad h = 1, ..., d, \tag{23}$$

and

$$g(t) = \frac{\prod_{g=1}^{d} (\sqrt{t} - \sqrt{r_g})^2}{\sum_{g=1}^{d} \prod_{h \in [d] \setminus \{g\}} (\sqrt{t} - \sqrt{r_h})^2}. \tag{24}$$

Then, define the testing statistics

$$T_f = 4n \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j=1}^{k} f_h(\hat{p}_j) - \sum_{j=1}^{k} f_h(q_j) \right)^2, \tag{25}$$

and

$$T_g = 4n \sum_{j=1}^{k} g(\hat{p}_j). \tag{26}$$

The properties of T_f and T_g are given by the following theorem. Again, the case $d = 1$ is trivial, and we only present results for $d \ge 2$.

Theorem 9 *Under Condition D, $T_g \rightsquigarrow \chi^2_{k-1}$, $T_f \rightsquigarrow \chi^2_{d-1}$ and $T_g - T_f \rightsquigarrow \chi^2_{k-d}$ as $n \to \infty$ under the null hypothesis.*

We define the testing function

$$\phi_\alpha = \mathbb{I}\{T_f > \chi^2_{d-1}(\alpha)\} \vee \mathbb{I}\{T_g > \chi^2_{k-1}(\alpha)\}.$$

By Theorem 9, the Type-1 error of this test converges to α. Though T_f is dominated by T_g under the null hypothesis, both are needed to ensure the power goes to 1 under the alternative.

Theorem 10 *Under Condition D, the following two statements are equivalent*

1. $\lim_{n\to\infty} \sqrt{n}\ell(p,q) = \infty$;
2. $\lim_{n\to\infty} \mathbb{P}_p\left(T_f > \chi^2_{d-1}(\alpha) \text{ or } T_g > \chi^2_{k-1}(\alpha)\right) = 1$, *for any constant* $\alpha \in (0,1)$,

where the probability $\mathbb{P}_p$ is defined in the beginning of this section.

6. Optimality of the Test

In this section, we study the optimality issue of the testing problem from a decision-theoretic perspective. The goal is to understand the fundamental limit of the problem and establish optimality results of the proposed testing procedures. We propose to study the setting where a null hypothesis is tested against a local alternative. This leads to a nontrivial power function and a precise asymptotic characterization of the minimax risk of the test. Depending on whether the data generating process is Gaussian or categorical, and whether the null hypothesis is degenerate or not, the optimality of the test will be studied in four different cases.

6.1 Gaussian Distribution: Non-Degenerate Case

We first consider the non-degenerate situation. That is, we assume that $\mu_1, ..., \mu_k$ are k different numbers. In Section 3, we impose the assumption that the k numbers $\mu_1, ..., \mu_k$ do not depend on n. This assumption can be made significantly weaker. For two indices j and l that are not equal, define

$$\eta_{jl} = \frac{1}{\mu_j - \mu_l} \prod_{h\in[k]\setminus\{j,l\}} \frac{\mu_l - \mu_h}{\mu_j - \mu_h}.$$

It characterizes the relative difference between μ_j and μ_l in the background of the set $\{\mu_1, ..., \mu_k\}$.

Condition M1 1 *Assume* $\lim_{n\to\infty} \max_{j\neq l} \frac{|\eta_{jl}|}{\sqrt{n}} = 0$.

To understand Condition M1, we can interpret $|\eta_{jl}| + |\eta_{lj}|$ as approximately the inverse distance between μ_j and μ_l. Therefore, we allow the possibility that $|\mu_j - \mu_l|$ converges to 0, but not as fast as $n^{-1/2}$. Otherwise, the data cannot tell the difference between $\mu_j \neq \mu_l$ and $\mu_j = \mu_l$, which is equivalent to the degenerate case. Recall that the number k is assumed to be a constant that does not vary with n throughout the paper.

Consider the testing problem

$$H_0 : \theta \in \Theta_0 = \{\theta : \ell(\theta, \mu) = 0\}, \quad H_1 : \theta \in \Theta_\delta = \left\{\theta : \ell(\theta, \mu) = \frac{\delta}{\sqrt{n}}\right\}. \tag{27}$$

That is, we test the null hypothesis against its contiguous alternative. The choice of H_1 ensures a non-trivial asymptotic power. We measure the testing error via the minimax risk function

$$R_n(k,\delta) = \inf_{0\le\phi\le 1} \left\{ \sup_{\theta\in\Theta_0} \mathbb{P}_\theta \phi + \sup_{\theta\in\Theta_\delta} \mathbb{P}_\theta(1-\phi) \right\}.$$

The probability symbol $\mathbb{P}_\theta$ stands for $N(\theta, n^{-1}I_k)$. Throughout the paper, we assume k and δ are fixed constants independent of n.

We first present the lower bound.

Theorem 11 *Under Condition M1, for sufficiently large n, we have*

$$R_n(k,\delta) \ge \inf_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \le t) \right).$$

Theorem 11 gives the benchmark of the problem. Using the proposed testing statistic T defined in (12), we can achieve this benchmark.

Theorem 12 *Consider the testing procedure $\phi = \mathbb{I}\{T > t^*\}$, where T is defined in (12), and*

$$t^* = \operatorname*{argmin}_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \le t) \right).$$

Under Condition M1, we have

$$\sup_{\theta\in\Theta_0} \mathbb{P}_\theta \phi + \sup_{\theta\in\Theta_\delta} \mathbb{P}_\theta(1-\phi) \le (1+o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \le t) \right),$$

as $n \to \infty$.

Theorem 12 characterizes both Type-1 and Type-2 error of the test $\phi = \mathbb{I}\{T > t^*\}$. The conclusion holds for any local alternative with $\delta \in (0,\infty)$. It complements the result of Theorem 3. Combining Theorem 11 and Theorem 12, we conclude that the minimax testing error has the following asymptotic formula

$$R_n(k,\delta) = (1+o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \le t) \right),$$

and this error can be achieved by the test $\phi = \mathbb{I}\{T > t^*\}$ with some carefully chosen t^* only depending on k and δ.

6.2 Gaussian Distribution: Degenerate Case

Now we consider situations of degeneracy. In Section 4, it is assumed that $\mu_1, ..., \mu_k$ only take d different values. This assumption can be relaxed. Here, we assume the k numbers $\mu_1, ..., \mu_k$ can be approximately clustered into d groups. Given d different numbers $\nu_1, ..., \nu_d$, for any pair $g \neq h$, define

$$\bar{\eta}_{gh} = \frac{1}{\nu_g - \nu_h} \prod_{l \in [k] \setminus \{g,h\}} \frac{\nu_h - \nu_l}{\nu_g - \nu_l}. \tag{28}$$

Condition M2 1 *Assume* $\lim_{n\to\infty} \max_{g\neq h} \frac{|\bar{\eta}_{gh}|}{\sqrt{n}} = 0$ *and there is a partition* $\mathcal{C}_1, ..., \mathcal{C}_d$ *of* $[k]$, *such that* $\limsup_{n\to\infty} \max_{1\leq g\leq d} \max_{j\in\mathcal{C}_g} \sqrt{n}|\mu_j - \nu_g| = 0$.

This condition says that $\mu_1, ..., \mu_k$ can be approximately clustered into d groups. The within-group distance is of a smaller order than $n^{-1/2}$, and the between-group distance is of a larger order than $n^{-1/2}$.

Consider the same local testing problem (27). The lower bound of the degenerate setting is given by the following theorem.

Theorem 13 *Under Condition M2,* $n \to \infty$, *we have*

$$R_n(k, \delta) \geq (1 + o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \leq t) \right).$$

This lower bound can be achieved asymptotically using the testing statistics T_f and T_g defined in (18) and (19).

Theorem 14 *Consider the testing procedure* $\phi = \mathbb{I}\{T_f > t^*\} \vee \mathbb{I}\{T_g > t^*\}$, *where* T_f *and* T_g *are defined in (18) and (19), and*

$$t^* = \operatorname*{argmin}_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \leq t) \right).$$

Under Condition M_2, *we have*

$$\sup_{\theta\in\Theta_0} \mathbb{P}_\theta \phi + \sup_{\theta\in\Theta_\delta} \mathbb{P}_\theta(1-\phi) \leq (1 + o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_k > t\right) + \mathbb{P}(\chi^2_{k,\delta^2} \leq t) \right),$$

as $n \to \infty$.

Theorem 14 shows that the test $\phi = \mathbb{I}\{T_f > t^*\} \vee \mathbb{I}\{T_g > t^*\}$ achieves the optimal error asymptotically under a local alternative. As we will show in Theorem 23, $T_g \geq T_f$ in probability under a local alternative that $\sqrt{n}\ell(\theta, \mu) = \delta \in (0, \infty)$. Therefore, the test $\phi = \mathbb{I}\{T_f > t^*\} \vee \mathbb{I}\{T_g > t^*\}$ is asymptotically equivalent to $\mathbb{I}\{T_g > t^*\}$, and the latter only uses T_g. Though the role of the statistic T_f is negligible for a local alternative, we have already shown in Theorem 6 that as soon as $\sqrt{n}\ell(\theta, \mu) \to \infty$, the effect of using T_f starts to kick in and it is necessary to use both T_f and T_g for the asymptotic power to approach one.

6.3 Categorical Distribution: Non-Degenerate Case

We study the fundamental limit of testing for the categorical distribution. In Section 5, we assume $q_1, ..., q_k$ are k different numbers that do not depend on n. In this section, we consider a condition that is significantly weaker. Define

$$\zeta_{jl} = \frac{1}{\sqrt{q_j} - \sqrt{q_l}} \prod_{h \in [k] \setminus \{j,l\}} \frac{\sqrt{q_l} - \sqrt{q_h}}{\sqrt{q_j} - \sqrt{q_h}}.$$

Similar to the definition of η_{jl}, ζ_{jl} characterizes the relative difference between $\sqrt{q_j}$ and $\sqrt{q_l}$ in the background of the set $\{\sqrt{q_1}, ..., \sqrt{q_k}\}$.

Condition M3 1 *Assume* $\lim_{n\to\infty} \max_{j\neq l} \frac{|\zeta_{jl}|}{\sqrt{n}} = 0$ *and* $\min_{1\leq j\leq k} nq_j(1 - q_j) \to \infty$.

Compared with Condition M1, the extra requirement $\min_{1\leq j\leq k} nq_j(1 - q_j) \to \infty$ in Condition M3 ensures that each q_j is bounded away from 0 and 1 with a gap at least of order n^{-1}. If this extra requirement does not hold, q_j would be asymptotically equivalent to 0 or 1, which results in a degenerate variance.

Consider the testing problem

$$H_0 : p \in \mathcal{P}_0 = \{p : \ell(p, q) = 0\}, \quad H_1 : p \in \mathcal{P}_\delta = \left\{ p : \ell(p, q) = \frac{\delta}{\sqrt{n}} \right\}. \tag{29}$$

Recall that the distance $\ell(\cdot, \cdot)$ is defined in (20).

We present the lower bound.

Theorem 15 *Under Condition M3, as* $n \to \infty$, *we have*

$$R_n(k, \delta) \geq (1 + o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \inf_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2} + \delta_2^2 \leq t) \right).$$

Theorem 15 gives the benchmark of the problem. Using the testing statistic T defined in (22), we can achieve this benchmark.

Theorem 16 *Consider the testing procedure* $\phi = \mathbb{I}\{T > t^*\}$, *where* T *is defined in (22), and*

$$t^* = \operatorname*{argmin}_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \sup_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2} + \delta_2^2 \leq t) \right).$$

Under Condition M3, we have

$$\sup_{\theta\in\mathcal{P}_0} \mathbb{P}_\theta \phi + \sup_{\theta\in\mathcal{P}_\delta} \mathbb{P}_\theta(1-\phi) \leq (1+o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \sup_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2} + \delta_2^2 \leq t) \right),$$

as $n \to \infty$.

The upper bound given by Theorem 16 does not exactly match the lower bound given by Theorem 15. The difference lies in the Type-2 error. For the lower bound, we get $\inf_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2}+\delta_2^2 \leq t)$, while for the upper bound, it is $\sup_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2}+\delta_2^2 \leq t)$. These two quantities are close, because for any δ_1 and δ_2 that satisfy $\delta_1^2 + \delta_2^2 = \delta^2$, the expectation of $\chi^2_{k-1,\delta_1^2} + \delta_2^2$ is always $k - 1 + \delta^2$. Therefore, the test using the statistic T is nearly optimal.

6.4 Categorical Distribution: Degenerate Case

Finally, we study the categorical distribution with the presence of degeneracy. In Section 5, we consider the situation where $q_1, ..., q_k$ take d different values. Here, we propose a much weaker condition. Given d different numbers $r_1, ..., r_d \in (0,1)$, for any pair $g \neq h$, define

$$\bar{\zeta}_{gh} = \frac{1}{\sqrt{r_g} - \sqrt{r_h}} \prod_{l \in [k] \setminus \{g,h\}} \frac{\sqrt{r_h} - \sqrt{r_l}}{\sqrt{r_g} - \sqrt{r_l}}.$$

Condition M4 1 *Assume* $\lim_{n\to\infty} \max_{j \neq l} \frac{|\bar{\zeta}_{jl}|}{\sqrt{n}} = 0$, $\min_{1 \leq j \leq k} nq_j(1-q_j) \to \infty$, *and there is a partition* $\mathcal{C}_1, ..., \mathcal{C}_d$ *of* $[k]$, *such that* $\limsup_{n\to\infty} \max_{1 \leq g \leq d} \max_{j \in \mathcal{C}_g} \sqrt{n}|\sqrt{q_j} - \sqrt{r_g}| = 0$.

Condition M4 has the same interpretation as Condition M2. The extra requirement $\min_{1 \leq j \leq k} nq_j(1-q_j) \to \infty$ is also needed in Condition M3 to prevent the variance from being degenerate.

The lower bound of the local testing problem (29) is given by the next theorem.

Theorem 17 *Under Condition M4, as* $n \to \infty$, *we have*

$$R_n(k, \delta) \geq (1 + o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \inf_{\{\delta_1, \delta_2 : \delta_1^2 + \delta_2^2 = \delta^2\}} \mathbb{P}(\chi^2_{k-1, \delta_1^2} + \delta_2^2 \leq t) \right).$$

For the matching upper bound, we can use the proposed testing statistics T_f and T_g defined in (25) and (26).

Theorem 18 *Consider the testing procedure* $\phi = \mathbb{I}\{T_f > t^*\} \vee \mathbb{I}\{T_g > t^*\}$, *where* T_f *and* T_g *are defined in (25) and (26), and*

$$t^* = \operatorname*{argmin}_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \sup_{\{\delta_1, \delta_2 : \delta_1^2 + \delta_2^2 = \delta^2\}} \mathbb{P}(\chi^2_{k-1, \delta_1^2} + \delta_2^2 \leq t) \right).$$

Under Condition M4, we have

$$\sup_{\theta \in \mathcal{P}_0} \mathbb{P}_\theta \phi + \sup_{\theta \in \mathcal{P}_\delta} \mathbb{P}_\theta(1-\phi) \leq (1+o(1)) \inf_{t>0} \left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \sup_{\{\delta_1, \delta_2 : \delta_1^2 + \delta_2^2 = \delta^2\}} \mathbb{P}(\chi^2_{k-1, \delta_1^2} + \delta_2^2 \leq t) \right),$$

as $n \to \infty$.

7. Numerical Studies

In this section, we conduct numerical experiments to verify the theoretical properties of the proposed testing procedures. In each of the following scenarios, we compute power functions of α-level tests for $\alpha = 0.05$ with various sample sizes.

Scenario 1. Consider $X \sim N(\theta, n^{-1}I_k)$, and we test the null hypothesis $\ell(\theta, \mu) = 0$ with μ specified as $\mu = (1, 2, 3, 4, 5)$.

Scenario 2. Consider $X \sim N(\theta, n^{-1}I_k)$, and we test the null hypothesis $\ell(\theta, \mu) = 0$ with μ specified as $\mu = (1, 3, 3, 3, 5, 5)$.

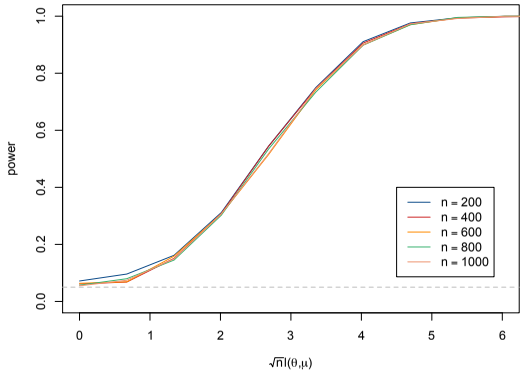


Figure 2: Power Curve of Scenario 1

Scenario 3. Consider $X_1, ..., X_n \sim (p_1, ..., p_k)$, and we test the null hypothesis $\ell(p, q) = 0$ with q specified as $q = (0.1, 0.2, 0.3, 0.4)$.

Scenario 4. Consider $X_1, ..., X_n \sim (p_1, ..., p_k)$, and we test the null hypothesis $\ell(p, q) = 0$ with q specified as $q = (0.1, 0.1, 0.4, 0.4)$.

Scenario 5. Consider $X_1, ..., X_n \sim (p_1, ..., p_k)$ and $Y_1, ..., Y_m \sim (q_1, ..., q_k)$, and we test the null hypothesis $\ell(p, q) = 0$. We set $p = (0.1, 0.1, 0.4, 0.4)$ and q to be local perturbations of p in a $O(n^{-1/2})$ neighborhood of p.

The numerical results of the five scenarios are summarized in Figures 2-6. The power curves are plotted in the contiguous regimes where $\ell(\theta, \mu) = O(n^{-1/2})$ or $\ell(p, q) = O(n^{-1/2})$. The grey dashed lines correspond to the nominal 0.05 level of the tests.

In Scenario 1, we vary θ in a local $O(n^{-1/2})$ neighborhood of the null hypothesis μ. It is clear that the power function is increasing with respect to $\sqrt{n}\ell(\theta, \mu)$. Moreover, with different sample sizes, the curves match well with each other. This verifies the conclusion of Theorem 3 that the magnitude of $\sqrt{n}\ell(\theta, \mu)$ asymptotically determines the power of the test. We observe in Figure 2 that the power is very close to 1 when $\sqrt{n}\ell(\theta, \mu)$ is greater than 6. The value of the power at $\sqrt{n}\ell(\theta, \mu) = 0$ corresponds to the Type-1 error in the null hypothesis. The actually Type-1 error is slightly greater than the nominal 0.05 level, but the approximations are reasonable for relatively large sample sizes.

Scenario 2 considers a harder null hypothesis with a degenerate $\mu = (1, 3, 3, 3, 5, 5)$. A 0.05-level test studied in Section 4 requires both testing statistics T_f and T_g. Similar to what is observed in Scenario 1, Figure 3 shows that the power approaches 1 at about $\sqrt{n}\ell(\theta, \mu) = 7$, which is predicted by Theorem 6. However, when $\sqrt{n}\ell(\theta, \mu) = 0$, the actual Type-1 errors are consistently larger than the nominal level 0.05, especially when the sample sizes are relatively small.

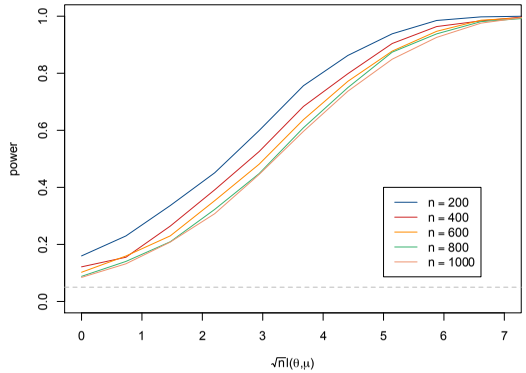


Figure 3: Power Curve of Scenario 2

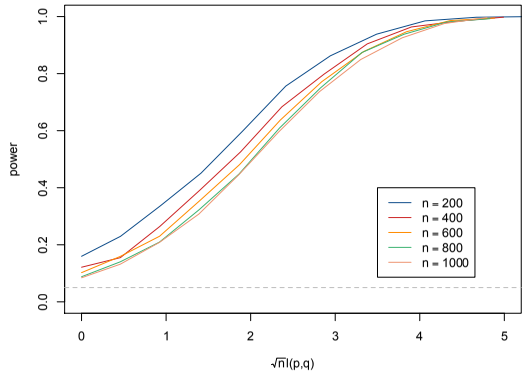


Figure 4: Power Curve of Scenario 3

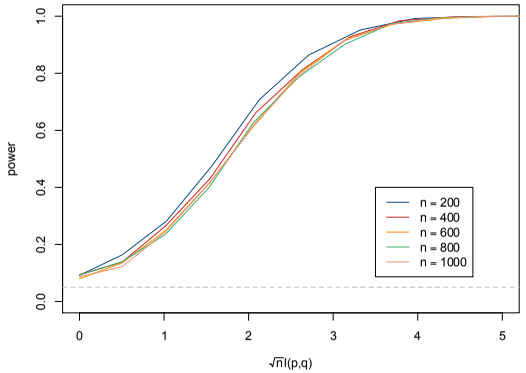


Figure 5: Power Curve of Scenario 4

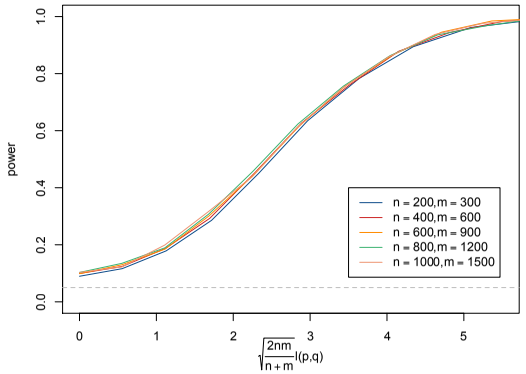


Figure 6: Power Curve of Scenario 5

Scenario 3 and Scenario 4 consider categorical distributions with a non-degenerate null $q = (0.1, 0.2, 0.3, 0.4)$ and a degenerate null $q = (0.1, 0.1, 0.4, 0.4)$, respectively. Again, Theorem 8 and Theorem 10 are verified by Figure 4 and Figure 5. The actual Type-1 errors are also larger than the nominal one, and are closer to 0.05 with larger sample sizes.

Finally in Scenario 5, we consider experiments of the two-sample test. According to the definitions of the testing statistics in (30) and (31), it is $\sqrt{\frac{2nm}{n+m}}\ell(p,q)$ that determines the power function asymptotically. Figure 6 shows that different power curves well match each other as functions of $\sqrt{\frac{2nm}{n+m}}\ell(p,q)$. As is predicted by Theorem 20, the power is close to 1 at a reasonably large value of $\sqrt{\frac{2nm}{n+m}}\ell(p,q)$.

A common theme in the above numerical results is that the actual Type-1 errors are always larger than the nominal one. We will give an explanation of this phenomenon in Section 9. Roughly speaking, whenever the null exhibits an ambiguous clustering structure, the asymptotic distribution of the testing statistic under the null is a noncentral chi-square distribution. Though a larger sample size helps to make the noncentrality parameter vanish (Figures 2-6), it still results in an estimate of Type-1 error that is too optimistic with a finite sample size. There are potentially two ways to overcome this difficulty. One is to appeal to a second-order correction, and the other is to estimate the noncentrality parameter in the null distribution. We leave this interesting topic as a future project.

8. Two-Sample Test

Consider two categorical distributions $(p_1, ..., p_k)$ and $(q_1, ..., q_k)$. Suppose we observe i.i.d. observations $X_1, ..., X_n$ from $(p_1, ..., p_k)$ and i.i.d. observations $Y_1, ..., Y_m$ from $(q_1, ..., q_k)$. We assume that $X_1, ..., X_n$ are independent of $Y_1, ..., Y_m$. The hypothesis testing problem we study in this section is

$$H_0 : \ell(p,q) = 0, \quad H_1 : \ell(p,q) > 0,$$

where the distance $\ell(\cdot,\cdot)$ is defined in (20). The two-sample testing problem is harder than the one-sample version that we have just studied. The major difficulty is that the definitions of the functions (23) and (24) all depend on the values of $(p_1, ..., p_k)$ and $(q_1, ..., q_k)$ under the null hypothesis, which is not available anymore in the two-sample scenario.

Our idea is to estimate the unknown $(p_1, ..., p_k)$ and $(q_1, ..., q_k)$ from the data, and then construct data-driven versions of (23) and (24).

For each $j \in [k]$, define $\hat{p}_j = \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{X_i = j\}$. Next, we will apply a variable clustering procedure to $(\hat{p}_1, ..., \hat{p}_k)$. The goal is to find a partition $\underline{\mathcal{C}}_1, ..., \underline{\mathcal{C}}_{\underline{d}}$ of $[k]$ according to

$$j \sim l \quad \text{if} \quad \sqrt{n}|\sqrt{\hat{p}_j} - \sqrt{\hat{p}_l}| \le \lambda_n.$$

Algorithmically, one can first sort the vector $(\hat{p}_1, ..., \hat{p}_k)$, and then find the partition sequentially. There exists a permutation $\sigma \in S_k$, such that we can rank the empirical frequencies as $\hat{p}_{\sigma(1)} \le \hat{p}_{\sigma(2)} \le ... \le \hat{p}_{\sigma(k)}$. Let $\hat{j}_1$ be the largest j such that $\sqrt{n}|\sqrt{\hat{p}_{\sigma(j)}} - \sqrt{\hat{p}_{\sigma(1)}}| \le \lambda_n$, where λ_n is some threshold to be specified later. Then, the first cluster is defined as $\underline{\mathcal{C}}_1 = \left\{\sigma(1), \sigma(2), ..., \sigma(\hat{j}_1)\right\}$. Similarly, we can define the second cluster as $\underline{\mathcal{C}}_2 = \{\sigma(\hat{j}_1 + 1), ..., \sigma(\hat{j}_2)\}$, where $\hat{j}_2$ is the largest j such that $\sqrt{n}|\sqrt{\hat{p}_{\sigma(j)}} - \sqrt{\hat{p}_{\sigma(\hat{j}_1)}}| \le \lambda_n$. We continue this operation until we obtain a partition $\underline{\mathcal{C}}_1, ..., \underline{\mathcal{C}}_{\underline{d}}$ of $[k]$. Here, $\underline{d}$ is the number of clusters estimated from the data. Now, for each $g \in [\underline{d}]$, we find the center of the cluster by $\sqrt{\underline{r}_g} = \frac{1}{|\underline{\mathcal{C}}_g|}\sum_{j\in\underline{\mathcal{C}}_g}\sqrt{\hat{p}_j}$. With the numbers $\underline{r}_1, ..., \underline{r}_{\underline{d}}$, we define

$$\underline{f}_h(t) = \frac{\int \prod_{g\in[\underline{d}]\setminus\{h\}}(\sqrt{t} - \sqrt{\underline{r}_g})}{\prod_{g\in[\underline{d}]\setminus\{h\}}(\sqrt{\underline{r}_h} - \sqrt{\underline{r}_g})}, \quad h = 1, ..., \underline{d},$$

and

$$\underline{g}(t) = \frac{\prod_{g=1}^{\underline{d}}(\sqrt{t} - \sqrt{\underline{r}_g})^2}{\sum_{g=1}^{\underline{d}}\prod_{h\in[\underline{d}]\setminus\{g\}}(\sqrt{t} - \sqrt{\underline{r}_h})^2}.$$

We repeat the above procedure on the observations $Y_1, ..., Y_m$. For each $j \in [k]$, define $\hat{q}_j = \frac{1}{m}\sum_{i=1}^m \mathbb{I}\{Y_i = j\}$. Then, apply the same variable clustering procedure on $(\hat{q}_1, ..., \hat{q}_k)$, and we obtain a partition $\overline{\mathcal{C}}_1, ..., \overline{\mathcal{C}}_{\overline{d}}$ of $[k]$. For each $g \in [\overline{d}]$, define $\sqrt{\overline{r}_g} = \frac{1}{|\overline{\mathcal{C}}_g|}\sum_{j\in\overline{\mathcal{C}}_g}\sqrt{\hat{q}_j}$. Analogous definitions of $\underline{f}_h$'s and $\underline{g}$ are given by

$$\overline{f}_h(t) = \frac{\int \prod_{g\in[\overline{d}]\setminus\{h\}}(\sqrt{t} - \sqrt{\overline{r}_g})}{\prod_{g\in[\overline{d}]\setminus\{h\}}(\sqrt{\overline{r}_h} - \sqrt{\overline{r}_g})}, \quad h = 1, ..., \overline{d},$$

and

$$\overline{g}(t) = \frac{\prod_{g=1}^{\overline{d}}(\sqrt{t} - \sqrt{\overline{r}_g})^2}{\sum_{g=1}^{\overline{d}}\prod_{h\in[\overline{d}]\setminus\{g\}}(\sqrt{t} - \sqrt{\overline{r}_h})^2}.$$

Now we can define testing statistics for this problem:

$$\begin{aligned} T_f &= \frac{2nm}{n+m}\sum_{h=1}^{\underline{d}}\frac{1}{|\underline{\mathcal{C}}_h|}\left(\sum_{j=1}^k \underline{f}_h(\hat{p}_j) - \sum_{j=1}^k \underline{f}_h(\hat{q}_j)\right)^2 \\ &\quad + \frac{2nm}{n+m}\sum_{h=1}^{\overline{d}}\frac{1}{|\overline{\mathcal{C}}_h|}\left(\sum_{j=1}^k \overline{f}_h(\hat{p}_j) - \sum_{j=1}^k \overline{f}_h(\hat{q}_j)\right)^2, \end{aligned} \tag{30}$$

and

$$T_g = \frac{2nm}{n+m}\left(\sum_{j=1}^k \underline{g}(\hat{q}_j) + \sum_{j=1}^k \overline{g}(\hat{p}_j)\right). \tag{31}$$

The asymptotic distributions of the testing statistics under the null distribution are given below.

Condition E 1 *Asume $q_1, ..., q_k$ are k numbers in $(0,1)$ that do not vary with n. Moreover, there exists a $d \ge 2$ and a partition $[k] = \cup_{h=1}^d \mathcal{C}_h$, such that $q_j = r_h$ for all $j \in \mathcal{C}_h$, $h \in [d]$.*

Theorem 19 *Assume λ_n is a diverging sequence that satisfies $\lambda_n = o(\sqrt{n})$ and we also assume $\frac{m}{n+m} \to \beta \in (0,1)$. Under Condition E, we have*

$$\begin{aligned} T_g &\rightsquigarrow \frac{1}{2}\beta\mathcal{X}_1 + \frac{1}{2}(1-\beta)\mathcal{X}_2 + \mathcal{X}_3, \\ T_f &\rightsquigarrow \mathcal{X}_3, \\ T_g - T_f &\rightsquigarrow \frac{1}{2}\beta\mathcal{X}_1 + \frac{1}{2}(1-\beta)\mathcal{X}_2, \end{aligned}$$

as $n \to \infty$ under the null hypothesis, where $\mathcal{X}_1$, $\mathcal{X}_2$ and $\mathcal{X}_3$ are independent random variables distributed as χ^2_{k-d}, χ^2_{k-d} and χ^2_{d-1}, respectively.

Let $\mathcal{X}(\alpha)$ be the number that satisfies $\mathbb{P}\left(\frac{1}{2}\beta\mathcal{X}_1 + \frac{1}{2}(1-\beta)\mathcal{X}_2 + \mathcal{X}_3 > \mathcal{X}(\alpha)\right) = \alpha$. We define the testing function as

$$\phi_\alpha = \mathbb{I}\{T_f > \mathcal{X}(\alpha)\} \vee \mathbb{I}\{T_g > \mathcal{X}(\alpha)\}.$$

Theorem 19 implies that this test has asymptotic Type-1 error α. The next result characterizes the power behavior of the test.

Theorem 20 *Assume λ_n is a diverging sequence that satisfies $\lambda_n = o(\sqrt{n})$ and we also assume $\frac{m}{n+m} \to \beta \in (0,1)$. Under Condition E, the following two statements are equivalent*

1. $\lim_{n\to\infty} \sqrt{n}\ell(p,q) = \infty$;
2. $\lim_{n\to\infty} \mathbb{P}_{p,q}\left(T_f > \mathcal{X}(\alpha) \text{ or } T_g > \mathcal{X}(\alpha)\right) = 1$, *for any constant* $\alpha \in (0,1)$,

where the probability $\mathbb{P}_{p,q}$ stands for the joint distribution of $X_1, ..., X_n, Y_1, ..., Y_m$.

Theorem 20 assumes Condition E. That is, $q_1, ..., q_k$ are fixed numbers do that depend on n, and $p_1, ..., p_k$ are allowed to vary with n. One can also assume an analogous condition for $p_1, ..., p_k$ as fixed numbers that satisfy Condition E, and allow $q_1, ..., q_k$ to vary with n.

9. Discussion and Future Directions

The testing procedures that we propose and analyze in this paper critically depend on the structure of null hypothesis. In Section 3, the mean vector $(\mu_1, ..., \mu_k)^T$ is assumed to consist of k distinct numbers, and the testing statistic is constructed based on the functions $f_1, ..., f_k$ defined in (10). In Section 4, we assume $(\mu_1, ..., \mu_k)^T$ consists of k numbers that take values in $\{\nu_1, ..., \nu_d\}$ for some $d \le k$. For this degenerate setting, we use the functions $f_1, ..., f_d$ and g defined in (14) and (15) to construct the testing statistics.

Much weaker assumptions are considered in Section 6. In Section 6.1, we allow $|\mu_j - \mu_l|$ to converge to 0, but require the difference should be of a larger order than $n^{-1/2}$ for every $j \neq l$. This extends the assumption in Section 3 that $\mu_1, ..., \mu_k$ are k distinct numbers that do not vary with n. In Section 6.2, we consider the setting where $|\nu_g - \nu_h|$ is of a larger order than $n^{-1/2}$ for every $g \neq h$, and $|\mu_j - \nu_h|$ is of a smaller order than $n^{-1/2}$ for every $j \in \mathcal{C}_h$. This setting extends the assumption used in Section 4. It turns out that the asymptotic distributions of the proposed testing statistics (Theorem 2 and Theorem 5) are still valid under these more general conditions (see Theorem 22 and Theorem 23 in Section 10.1).

However, the conditions in Section 6.1 and Section 6.2 still do not cover all situations. By requiring the within-cluster distance to be of a smaller order than $n^{-1/2}$ and the between-cluster distance to be of a larger order than $n^{-1/2}$, the numbers $\mu_1, ..., \mu_k$ enjoy an approximately exact clustering structure, because for each $j \neq l$, we either have $\sqrt{n}|\mu_j - \mu_l| \to 0$ or $\sqrt{n}|\mu_j - \mu_l| \to \infty$, depending on whether j and l are in the same cluster or not. A possible situation $\sqrt{n}|\mu_j - \mu_l| \asymp 1$ is excluded.

In this section, we discuss a situation where the clustering structure of the numbers $\mu_1, ..., \mu_k$ is ambiguous. Consider a partition $\mathcal{C}_1, ..., \mathcal{C}_d$ of $[k]$. Define $\nu_h = \frac{1}{|\mathcal{C}_h|}\sum_{j\in\mathcal{C}_g} \mu_j$. Instead of assuming the within-cluster distance is of a smaller order than $n^{-1/2}$, we consider the situation where $n\sum_{h=1}^d \sum_{j\in\mathcal{C}_h}(\mu_j - \nu_h)^2$ is of a constant order. Moreover, we also assume the between-cluster distance $|\nu_g - \nu_h|$ is of a larger order than $n^{-1/2}$ for every $g \neq h$. This is without loss of generality, because if there is some $g \neq h$, such that $|\nu_g - \nu_h| = O(n^{-1/2})$, then $\mathcal{C}_g$ and $\mathcal{C}_h$ can be combined into a single cluster. Recall the definition of $\bar{\eta}_{gh}$ in (28), we formalize this ambiguous clustering structure into the following condition.

Condition M2' 1 *For the partition $\mathcal{C}_1, ..., \mathcal{C}_d$ and clustering centers $\nu_1, ..., \nu_d$ defined above, assume $\lim_{n\to\infty} \max_{g\neq h} \frac{|\bar{\eta}_{gh}|}{\sqrt{n}} = 0$, and $\tau^2 = \lim_{n\to\infty} n\sum_{h=1}^d \sum_{j\in\mathcal{C}_h}(\mu_j - \nu_h)^2 \in [0, \infty)$.*

Note that Condition M2 is a special case of Condition M2' when $\tau^2 = 0$. The next theorem gives the asymptotic distribution of the testing statistics T_f and T_g defined in (18) and (19) under the null hypothesis.

Theorem 21 *Assume Condition M2' holds. Then we have $T_g \rightsquigarrow \chi^2_{k,\tau^2}$, $T_f \rightsquigarrow \chi^2_d$ and $T_g - T_f \rightsquigarrow \chi^2_{k-d,\tau^2}$ as $n \to \infty$ under the null hypothesis $X \sim N(\mu, n^{-1}I_k)$.*

It is interesting to see that the asymptotic distribution of T_g is a noncentral chi-squared distribution even under the null hypothesis. The noncentrality parameter τ^2 characterizes the within-cluster distance of $\mu_1, ..., \mu_k$ with respect to the partition $\mathcal{C}_1, ..., \mathcal{C}_d$. Theorem 21 is reduced to Theorem 5 when $\tau^2 = 0$.

Define a number $\chi^2_{k,\tau^2}(\alpha)$ that satisfies $\mathbb{P}(\chi^2_{k,\tau^2} \le \chi^2_{k,\tau^2}(\alpha)) = 1 - \alpha$. Then, an α-level testing function is $\phi_\alpha = \mathbb{I}\{T_g > \chi^2_{k,\tau^2}(\alpha)\} \vee \mathbb{I}\{T_f > \chi^2_{k,\tau^2}(\alpha)\}$. Compared with the null hypothesis where $\tau^2 = 0$, a nonzero τ^2 requires a higher rejection level. This means the test will have less power under a contiguous alternative, compared with the situation where $\tau^2 = 0$. Suppose $\tilde{\tau}^2 = \lim_{n\to\infty} n\sum_{h=1}^d \sum_{j\in\mathcal{C}_h}(\theta_j - \nu_h)^2 \in (0, \infty)$. Then, one can also show that $T_g \rightsquigarrow \chi^2_{k,\tilde{\tau}^2}$ under the alternative $X \sim N(\theta, n^{-1}I_k)$. Therefore, the test ϕ_α starts to have power when $\tilde{\tau}^2$ exceeds τ^2. When $\tilde{\tau}^2$ is close to or even smaller then τ^2, this test will not have any power under the alternative. On the other hand, outside of the contiguous regime where $\sqrt{n}\ell(\theta, \mu) \to \infty$, we must have $\tilde{\tau}^2 = \infty$, and then the test will have asymptotic power 1.

From what we have just discussed, we can see that the structure of $\mu_1, ..., \mu_k$ plays a critical role on the solution of the problem. The discussion also applies to the case of categorical distributions and we can obtain similar conclusions there. Theorem 21 characterizes the asymptotic distribution of the testing statistics when $\mu_1, ..., \mu_k$ exhibit an ambiguous clustering structure, right on the edge of degeneracy. This results in a non-trivial behavior

of the power function. Exact characterization of optimality of the testing problem (as what we have done in Section 6) on the edge of degeneracy remains open, and we shall consider this problem as a future project.

Finally, we discussed a list of open problems that can be viewed as natural extensions of the results in the paper.

1. *Growing or infinite support size.* The paper focuses on the case where k is a fixed integer that does not depend on n. The case with a growing k or even $k = \infty$ is of potential importance in many high-dimensional data analysis situations. This requires new techniques because for a probability vector $p = (p_1, ..., p_k)$ with a growing or an infinite k, many p_j's have extremely small values.

2. *Testing a parametric family with permutation invariance.* An extension to the null hypothesis (1) is

$$H_0 : p(j) = f_\lambda(\pi(j)) \quad \text{for some } \lambda \in \Lambda \text{ and some } \pi \in S_k.$$

   Here, $\{f_\lambda(j)\}$ is a discrete distribution with an unknown parameter $\lambda \in \Lambda$. An example is Poisson(λ). Without the permutation $\pi \in S_k$, the null hypothesis becomes $p = f_\lambda$ for some $\lambda \in \Lambda$, which is a classical goodness-of-fit test of a parametric family **??**.

3. *Non-asymptotic study of minimax separation.* This paper considers testing procedures that enjoy asymptotic optimality (Section 6). An important theoretical problem is to understand the minimax separation ρ^* for which one can consistently test the null $\ell(p, q) = 0$ against the alternative $\ell(p, q) > \rho$ if and only if $\rho > \rho^*$. With the permutation invariance, the null hypothesis is a non-convex set, which is in contrast to a convex case that was recently studied by **?**.

4. *Other group invariance.* Permutation invariance is a special case of group invariance. A more general question is to consider a null hypothesis that is invariant with respect to other group actions. A recent work **?** considered a group of cyclic shifts. It would be interesting to understand the method of invariance in a general group theoretic framework.

10. Proofs

In this section, we present the proofs of all results in the paper. In Section 10.1, we derive the asymptotic distributions of the proposed testing statistics in various settings. These results are used to derive Theorem 2, Theorem 5, Theorem 7, Theorem 9, Theorem 19 and Theorem 21. Then, in Section 10.2, we analyze the powers of the proposed tests, which include the proofs of Theorem 3, Theorem 6, Theorem 8, Theorem 10 and Theorem 20. Finally, in Section 10.3, we give proofs of all results in Section 6.

10.1 Asymptotic Distribution of the Testing Statistics

We first present and prove four theorems of the proposed testing statistics in various settings.

Theorem 22 *In addition to Condition M1, assume*

$$\lim_{n\to\infty} \sqrt{n}\ell(\theta, \mu) = \delta \in [0, \infty).$$

Then, as n tends to infinity, $T \rightsquigarrow \chi^2_{k,\delta^2}$.

Proof We first calculate the derivatives of $f_l(t)$. The first derivative is

$$f_l'(t) = \frac{\prod_{j\in[k]\setminus\{l\}}(t - \mu_j)}{\prod_{j\in[k]\setminus\{l\}}(\mu_l - \mu_j)}.$$

Therefore, $f_l'(\mu_l) = 1$. For any $j \neq l$, we give a bound for $\sup_{|t-\mu_j|\le n^{-1/2}\epsilon} |f_l'(t)|$. The following inequality is useful.

$$|\eta_{lh}| + |\eta_{hl}| = \frac{1}{|\mu_l - \mu_h|}\left(\prod_{j\in[k]\setminus\{l,h\}}\left|\frac{\mu_l - \mu_j}{\mu_h - \mu_j}\right| + \prod_{j\in[k]\setminus\{l,h\}}\left|\frac{\mu_h - \mu_j}{\mu_l - \mu_j}\right|\right) \ge \frac{2}{|\mu_l - \mu_h|}. \quad (32)$$

Note that

$$\begin{aligned}
|f_l'(t)| &= \frac{|t - \mu_j|}{|\mu_l - \mu_j|}\left|\prod_{h\in[k]\setminus\{l,j\}}\frac{t - \mu_h}{\mu_l - \mu_h}\right| \\
&= \frac{|t - \mu_j|}{|\mu_l - \mu_j|}\left|\prod_{h\in[k]\setminus\{l,j\}}\left(\frac{t - \mu_j}{\mu_l - \mu_h} + \frac{\mu_j - \mu_h}{\mu_l - \mu_h}\right)\right| \\
&\le \frac{|t - \mu_j|}{|\mu_l - \mu_j|}2^{k-2}\left(\left|\prod_{h\in[k]\setminus\{l,j\}}\left(\frac{t - \mu_j}{\mu_l - \mu_h}\right)\right| + \left|\prod_{h\in[k]\setminus\{l,j\}}\left(\frac{\mu_j - \mu_h}{\mu_l - \mu_h}\right)\right|\right) \\
&\le 2^{k-2}|t - \mu_j|^{k-1}\left(\frac{|\eta_{lj}| + |\eta_{jl}|}{2}\right)\prod_{h\in[k]\setminus\{l,j\}}\left(\frac{|\eta_{lh}| + |\eta_{hl}|}{2}\right) + 2^{k-2}|t - \mu_j||\eta_{lj}|.
\end{aligned}$$

Therefore, we have the bound

$$\kappa_1(\epsilon) = \max_{j\neq l}\sup_{|t-\mu_j|\le n^{-1/2}\epsilon}|f_l'(t)| \le 2^{k-2}\left[\max_{j\neq l}\left(\frac{\epsilon|\eta_{jl}|}{\sqrt{n}}\right)^{k-2} + \max_{j\neq l}\left(\frac{\epsilon|\eta_{jl}|}{\sqrt{n}}\right)\right]. \quad (33)$$

The above bound is useful for $k \ge 3$. For $k = 2$, it is easy to see

$$\kappa_1(\epsilon) = \max_{j\neq l}\sup_{|t-\mu_j|\le n^{-1/2}\epsilon}|f_l'(t)| \le \max_{j\neq l}\left(\frac{\epsilon|\eta_{jl}|}{\sqrt{n}}\right). \quad (34)$$

The second derivative of $f_l(t)$ is

$$f_l''(t) = \sum_{j\in[k]\setminus\{l\}}\frac{1}{(\mu_l - \mu_j)}\prod_{h\in[k]\setminus\{l,j\}}\frac{t - \mu_h}{\mu_l - \mu_h}.$$

We give a bound for $\sup_{|t-\mu_l|\le n^{-1/2}\epsilon}|f_l''(t)|$. Similar calculation gives

$$\kappa_2(\epsilon)=\max_{1\le l\le k}\sup_{|t-\mu_l|\le n^{-1/2}\epsilon}|f_l''(t)|\le k\max_{j\neq l}|\eta_{jl}|\max_{j\neq l}\left(1+\frac{\epsilon|\eta_{jl}|}{\sqrt{n}}\right)^{k-2}. \tag{35}$$

Now we are ready to derive the asymptotic distribution of T. We write the observation as $X_j=\theta_j+n^{-1/2}Z_j$, with $Z_j\sim N(0,1)$ independently. The condition $\lim_{n\to\infty}\sqrt{n}\ell(\theta,\mu)=\delta$ implies that there is some n_0, such that for any $n>n_0$, we have

$$n\ell^2(\theta,\mu)\le C_n\delta^2,$$

where C_n is an sequence that tends to infinity arbitrarily slowly. In particular, we require that C_n satisfies $C_n\to\infty$ and $\frac{C_n^{3/2}\max_{j\neq l}|\eta_{jl}|}{\sqrt{n}}\to 0$. The existence of such sequence C_n is guaranteed by the assumption $\frac{\max_{j\neq l}|\eta_{jl}|}{\sqrt{n}}\to 0$. Thus, there exists a $\pi\in S_k$, possibly depending on n, such that

$$\max_{1\le j\le k}(\theta_j-\mu_{\pi(j)})^2\le\frac{C_n\delta^2}{n}.$$

Since k does not depend on n, $\max_{1\le j\le k}Z_j^2\le C_n$ with probability that goes to 1. By triangle inequality,

$$\max_{1\le j\le k}|X_j-\mu_{\pi(j)}|\le\frac{\sqrt{C_n}(1+\sqrt{\delta^2})}{\sqrt{n}}, \tag{36}$$

with probability that goes to 1. We use Taylor expansion. For j such that $\pi(j)=l$, we have

$$f_l(X_j)-f_l(\mu_{\pi(j)})=(X_j-\mu_{\pi(j)})+\frac{1}{2}f_l''(\xi_{jl})(X_j-\mu_{\pi(j)})^2,$$

where we have used the fact that $f_l'(\mu_l)=1$. For j such that $\pi(j)\neq l$, we have

$$f_l(X_j)-f_l(\mu_{\pi(j)})=f_l'(\xi_{jl})(X_j-\mu_{\pi(j)}).$$

Therefore,

$$\begin{aligned}&\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|\\ \le\;&\frac{1}{2}|f_l''(\xi_{\pi^{-1}(l)l})|(X_{\pi^{-1}(l)}-\mu_l)^2+\sum_{j\neq\pi^{-1}(l)}|f_l'(\xi_{jl})||X_j-\mu_{\pi(j)}|.\end{aligned}$$

The number ξ_{jl} is between X_j and $\mu_{\pi(j)}$, which implies

$$\max_{j,l}|\xi_{jl}-\mu_{\pi(j)}|\le\max_{1\le j\le k}|X_j-\mu_{\pi(j)}|\le\frac{\sqrt{C_n}(1+\sqrt{\delta^2})}{\sqrt{n}}. \tag{37}$$

Using the bounds (33), (34), (35), (36) and (37), we have

$$\begin{aligned}&\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|\\ \le\;&\frac{1}{2}\frac{C_n(1+\sqrt{\delta^2})^2}{n}\kappa_2\left(\sqrt{C_n}(1+\sqrt{\delta^2})\right)\\ &+(k-1)\kappa_1\left(\sqrt{C_n}(1+\sqrt{\delta^2})\right)\frac{\sqrt{C_n}(1+\sqrt{\delta^2})}{\sqrt{n}}.\end{aligned} \tag{38}$$

Therefore,

$$\begin{aligned}&\left|T-n\sum_{l=1}^k(X_{\pi^{-1}(l)}-\mu_l)^2\right|\\ \le\;&n\sum_{l=1}^k\left|\left(\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)\right)^2-(X_{\pi^{-1}(l)}-\mu_l)^2\right|\\ \le\;&2n\sum_{l=1}^k|X_{\pi^{-1}(l)}-\mu_l|\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|\\ &+n\sum_{l=1}^k\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|^2\\ \le\;&2k\sqrt{n}\sqrt{C_n}(1+\sqrt{\delta^2})\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|\\ &+kn\left|\sum_{j=1}^k f_l(X_j)-\sum_{j=1}^k f_l(\mu_j)-(X_{\pi^{-1}(l)}-\mu_l)\right|^2.\end{aligned}$$

By (38), the bound for $\left|T-n\sum_{l=1}^k(X_{\pi^{-1}(l)}-\mu_l)^2\right|$ is of order $\frac{C_n^{3/2}\max_{j\neq l}|\eta_{jl}|}{\sqrt{n}}\to 0$. Finally, it is easy to see that

$$n\sum_{l=1}^k(X_{\pi^{-1}(l)}-\mu_l)^2\sim\chi^2_{k,\delta_n^2},$$

where $\delta_n^2=n\|\theta-\mu_\pi\|^2\to\delta^2$. Therefore, T converges to χ^2_{k,δ^2} in distribution. ■

Theorem 23 *In addition to Condition M2, assume*

$$\lim_{n\to\infty}\sqrt{n}\ell(\theta,\mu)=\delta\in[0,\infty).$$

Then, as n tends to infinity, $T_g\rightsquigarrow\chi^2_{k,\delta^2}$, and $T_g\ge T_f$ in probability. Moreover, if $\delta^2=0$, we have $T_g\rightsquigarrow\chi^2_k$, $T_f\rightsquigarrow\chi^2_d$ and $T_g-T_f\rightsquigarrow\chi^2_{k-d}$.

Proof The case $d = 1$ is obvious. We only prove the case $d \geq 2$. Similar to the inequality (32), we have $|\bar{\eta}_{gh}| + |\bar{\eta}_{hg}| \geq \frac{2}{|\nu_g - \nu_h|}$. By Condition M2, we have

$$\frac{\max_{1\leq g\leq d} \max_{j\in\mathcal{C}_g} |\mu_j - \nu_g|}{\min_{g\neq h} |\nu_g - \nu_h|} = o(1).$$

The observation is $X_j = \theta_j + n^{-1/2} Z_j$ with $Z_j \sim N(0,1)$ independently. Use the notation $L = \max_{1\leq g\leq d} \max_{j\in\mathcal{C}_g} \sqrt{n}|\mu_j - \nu_g| = o(1)$. Under the assumption of the theorem, there exists a sequence C_n that satisfies $C_n \to \infty$, $C_n^2 L \to 0$ and $\frac{C_n^{3/2} \max_{g\neq h} |\bar{\eta}_{gh}|}{\sqrt{n}} \to 0$, such that $\max_{1\leq j\leq k} Z_j^2 \leq C_n$ with probability tending to 1. Similar to the bound (36), the assumption $\lim_{n\to\infty} \sqrt{n}\ell(\theta,\mu) = \delta < \infty$ implies the existence of $\pi \in S_k$ such that

$$\max_{1\leq j\leq k} |X_j - \mu_{\pi(j)}| \leq \frac{\sqrt{C_n}(1+\sqrt{\delta^2})}{\sqrt{n}}.$$

We first study the asymptotic distribution of T_g. Note that

$$\max_{1\leq g\leq d} \max_{j\in\mathcal{C}_g} |X_{\pi^{-1}(j)} - \nu_g| \leq \frac{\sqrt{C_n}(1+\sqrt{\delta^2}) + L}{\sqrt{n}}. \tag{39}$$

Together with Condition M2 and the choice of C_n, we can immediately deduce

$$\frac{\max_{1\leq g\leq d} \max_{j\in\mathcal{C}_g} |X_{\pi^{-1}(j)} - \nu_g|}{\min_{g\neq h} |\nu_g - \nu_h|} \leq \max_{g\neq h} |\bar{\eta}_{gh}| \frac{\sqrt{C_n}(1+\sqrt{\delta^2}) + L}{\sqrt{n}} = o(1).$$

The function $g(t)$ can be written as

$$\frac{1}{g(t)} = \sum_{g=1}^{d} \frac{1}{(t-\nu_g)^2}.$$

For each $j \in \mathcal{C}_g$, we have

$$\frac{(X_{\pi^{-1}(j)} - \nu_g)^2}{g(X_{\pi^{-1}(j)})} = 1 + \sum_{h\in[d]\setminus\{g\}} \frac{(X_{\pi^{-1}(j)} - \nu_g)^2}{(X_{\pi^{-1}(j)} - \nu_h)^2}.$$

Thus,

$$\left| \frac{g(X_{\pi^{-1}(j)})}{(X_{\pi^{-1}(j)} - \nu_g)^2} - 1 \right| \leq \frac{\sum_{h\in[d]\setminus\{g\}} \frac{(X_{\pi^{-1}(j)} - \nu_g)^2}{(X_{\pi^{-1}(j)} - \nu_h)^2}}{1 + \sum_{h\in[d]\setminus\{g\}} \frac{(X_{\pi^{-1}(j)} - \nu_g)^2}{(X_{\pi^{-1}(j)} - \nu_h)^2}} \leq \sum_{h\in[d]\setminus\{g\}} \frac{(X_{\pi^{-1}(j)} - \nu_g)^2}{(X_{\pi^{-1}(j)} - \nu_h)^2}, \tag{40}$$

where the bound on the right hand side above can be bounded by

$$\sum_{h\in[d]\setminus\{g\}} \frac{2(X_{\pi^{-1}(j)} - \nu_g)^2}{(\nu_g - \nu_h)^2 - 2(X_{\pi^{-1}(j)} - \nu_h)^2} \leq 4d \max_{g\neq h} |\bar{\eta}_{gh}| \frac{\sqrt{C_n}(1+\sqrt{\delta^2}) + L}{\sqrt{n}}.$$

Together with (39), we have

$$\begin{aligned} & |g(X_{\pi^{-1}(j)}) - (X_{\pi^{-1}(j)} - \nu_g)^2| \\ = \; & (X_{\pi^{-1}(j)} - \nu_g)^2 \left| \frac{g(X_{\pi^{-1}(j)})}{(X_{\pi^{-1}(j)} - \nu_g)^2} - 1 \right| \\ \leq \; & 4d \max_{g\neq h} |\bar{\eta}_{gh}| \left(\frac{\sqrt{C_n}(1+\sqrt{\delta^2}) + L}{\sqrt{n}} \right)^3. \end{aligned}$$

Therefore

$$\begin{aligned} & \left| T_g - n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \nu_h)^2 \right| \\ \leq \; & n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} \left| g(X_{\pi^{-1}(j)}) - (X_{\pi^{-1}(j)} - \nu_g)^2 \right| \\ \leq \; & 4kd \max_{g\neq h} |\bar{\eta}_{gh}| \frac{\left(\sqrt{C_n}(1+\sqrt{\delta^2}) + L\right)^3}{\sqrt{n}} = o(1). \end{aligned} \tag{41}$$

For each $j \in \mathcal{C}_h$,

$$\begin{aligned} & n|(X_{\pi^{-1}(j)} - \nu_h)^2 - (X_{\pi^{-1}(j)} - \mu_j)^2| \\ \leq \; & n|\nu_h - \mu_j||X_{\pi^{-1}(j)} - \nu_h + X_{\pi^{-1}(j)} - \mu_j| \\ \leq \; & L\left(2\sqrt{C_n}(1+\sqrt{\delta^2}) + L\right) = o(1). \end{aligned}$$

Thus,

$$\left| T_g - n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j)^2 \right| \tag{42}$$

has a bound that tends to 0. Observe that

$$n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j)^2 \sim \chi^2_{k,\delta_n^2},$$

where

$$\delta_n^2 = n \sum_{j=1}^{k} (\theta_j - \mu_{\pi(j)})^2 \rightsquigarrow \delta^2.$$

Thus, $T_g \rightsquigarrow \chi^2_{k,\delta^2}$.

Next we derive the asymptotic distribution of T_f. Similar to (33), (34) and (35), we also have

$$\kappa_1(\epsilon) = \max_{g\neq h} \sup_{|t-\nu_g|\leq n^{-1/2}\epsilon} |f_h'(t)| \leq 2^{d-2} \left[\max_{g\neq h} \left(\frac{\epsilon|\bar{\eta}_{gh}|}{\sqrt{n}} \right)^{d-2} + \max_{g\neq h} \left(\frac{\epsilon|\bar{\eta}_{gh}|}{\sqrt{n}} \right) \right], \tag{43}$$

for $d \geq 3$,

$$\kappa_1(\epsilon) = \max_{g \neq h} \sup_{|t-\nu_g| \leq n^{-1/2}\epsilon} |f_h'(t)| \leq \max_{g \neq h} \left(\frac{\epsilon |\bar{\eta}_{gh}|}{\sqrt{n}} \right), \tag{44}$$

for $d = 2$, and

$$\kappa_2(\epsilon) = \max_{1 \leq h \leq d} \sup_{|t-\nu_h| \leq n^{-1/2}\epsilon} |f_h''(t)| \leq d \max_{g \neq h} |\bar{\eta}_{gh}| \max_{g \neq h} \left(1 + \frac{\epsilon |\bar{\eta}_{gh}|}{\sqrt{n}} \right)^{d-2}. \tag{45}$$

For any $j \in \mathcal{C}_g$,

$$\begin{aligned} & f_g(X_{\pi^{-1}(j)}) - f_g(\mu_j) \\ = \; & f_g(X_{\pi^{-1}(j)}) - f_g(\nu_g) + f_g(\nu_g) - f_g(\mu_j) \\ = \; & f_g'(\nu_g)(X_{\pi^{-1}(j)} - \mu_j) + \frac{1}{2} f_g''(\xi_{jg})(X_{\pi^{-1}(j)} - \nu_g)^2 - \frac{1}{2} f_g''(\bar{\xi}_{jg})(\mu_j - \nu_g)^2. \end{aligned}$$

For any $j \in \mathcal{C}_h$ with any $h \neq g$,

$$f_g(X_{\pi^{-1}(j)}) - f_g(\mu_j) = f_g'(\tilde{\xi}_{jg})(X_{\pi^{-1}(j)} - \mu_j).$$

By the fact that $f_g'(\nu_g) = 1$, we have

$$\begin{aligned} & \left| \sum_{j=1}^k f_g(X_j) - \sum_{j=1}^k f_g(\mu_j) - \sum_{j \in \mathcal{C}_g} (X_{\pi^{-1}(j)} - \mu_j) \right| \\ \leq \; & \frac{1}{2} \sum_{j \in \mathcal{C}_g} |f_g''(\xi_{jg})| (X_{\pi^{-1}(j)} - \nu_g)^2 + \frac{1}{2} \sum_{j \in \mathcal{C}_g} |f_g''(\bar{\xi}_{jg})| (\mu_j - \nu_g)^2 \\ & \sum_{h \in [d] \setminus \{g\}} \sum_{j \in \mathcal{C}_h} |f_g'(\tilde{\xi}_{jg})| |X_{\pi^{-1}(j)} - \mu_j|. \end{aligned}$$

The number ξ_{jg} is between $X_{\pi^{-1}(j)}$ and ν_g, the number $\bar{\xi}_{jg}$ is between μ_j and ν_g, and the number $\tilde{\xi}_{jg}$ is between $X_{\pi^{-1}(j)}$ and μ_j. Thus,

$$|\xi_{jg} - \nu_g| \leq |X_{\pi^{-1}(j)} - \nu_g| \leq \frac{\sqrt{C_n}(1 + \sqrt{\delta^2}) + L}{\sqrt{n}},$$

$$|\bar{\xi}_{jg} - \nu_g| \leq |\mu_j - \nu_g| \leq \frac{L}{\sqrt{n}},$$

and

$$|\tilde{\xi}_{jg} - \mu_j| \leq |X_{\pi^{-1}(j)} - \mu_j| \leq \frac{\sqrt{C_n}(1 + \sqrt{\delta^2})}{\sqrt{n}}.$$

Using the bounds (43), (44) and (45), we can deduce

$$\begin{aligned} & \left| \sum_{j=1}^k f_g(X_j) - \sum_{j=1}^k f_g(\mu_j) - \sum_{j \in \mathcal{C}_g} (X_{\pi^{-1}(j)} - \mu_j) \right| \\ \leq \; & k \frac{(\sqrt{C_n}(1 + \sqrt{\delta^2}) + L)^2}{n} \kappa_2 \left(\sqrt{C_n}(1 + \sqrt{\delta^2}) + L \right) \\ & + k \kappa_1 \left(\sqrt{C_n}(1 + \sqrt{\delta^2}) \right) \frac{\sqrt{C_n}(1 + \sqrt{\delta^2})}{\sqrt{n}}. \end{aligned}$$

Similar to the proof of Theorem (32), we can show that

$$\left| T_f - n \sum_{h=1}^d \frac{1}{|\mathcal{C}_h|} \left(\sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \right| \tag{46}$$

has a bound of order $\frac{C_n^{3/2} \max_{j \neq l} |\eta_{jl}|}{\sqrt{n}} \to 0$. Note that when $\delta^2 = 0$,

$$n \sum_{h=1}^d \frac{1}{|\mathcal{C}_h|} \left(\sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \sim \chi_d^2.$$

Thus, $T_f \rightsquigarrow \chi_d^2$.

Finally, we derive the asymptotic distribution for $T_g - T_f$. The bounds for (42) and (46) imply that

$$\left| T_g - T_f - n \sum_{h=1}^d \sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j)^2 + n \sum_{h=1}^d \frac{1}{|\mathcal{C}_h|} \left(\sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \right|$$

has a bound that tends to zero. Thus, the asymptotic distribution of $T_g - T_f$ is the same as that of

$$\begin{aligned} & n \sum_{h=1}^d \sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j)^2 - n \sum_{h=1}^d \frac{1}{|\mathcal{C}_h|} \left(\sum_{j \in \mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \\ = \; & n \sum_{h=1}^d \sum_{j \in \mathcal{C}_h} \left(X_{\pi^{-1}(j)} - \frac{1}{|\mathcal{C}_h|} \sum_{j \in \mathcal{C}_h} X_{\pi^{-1}(j)} - \left(\mu_j - \frac{1}{|\mathcal{C}_h|} \sum_{j \in \mathcal{C}_h} \mu_j \right) \right)^2, \end{aligned}$$

which is χ_{k-d}^2 when $\delta^2 = 0$. Therefore, $T_g - T_f \rightsquigarrow \chi_{k-d}^2$. Without the condition $\delta^2 = 0$, we can still claim $T_g \geq T_f$ in probability. ■

Theorem 24 *For $\pi = \operatorname{argmin}_{\pi \in S_k} \|\sqrt{p} - \sqrt{q_\pi}\|$, define*

$$\delta_1^2 = 4n \sum_{l=1}^k (1 - p_l) \left(\sqrt{p_l} - \sqrt{q_{\pi(l)}} \right)^2, \tag{47}$$

and

$$\delta_2^2 = 4n \sum_{l=1}^k p_l \left(\sqrt{p_l} - \sqrt{q_{\pi(l)}} \right)^2. \tag{48}$$

Assume $\limsup_{n \to \infty} (\delta_1^2 + \delta_2^2) < \infty$. Then, under Condition M3, $T - \delta_2^2 \rightsquigarrow \chi_{k-1, \delta_1^2}^2$, as n tends to infinity.

Proof The proof is almost the same as that of Theorem 22, and therefore we will omit some overlapping details. Largely speaking, we can replace the t, μ_j, θ_j, X_j by $\sqrt{t}, \sqrt{q_j}, \sqrt{p_j}, \sqrt{\hat{p}_j}$, and most parts in the proof of Theorem 22 will go through. Here are a few different details. We write $\sqrt{\hat{p}_j} = \sqrt{p}_j + n^{-1/2}Z_j/2$, with $Z_j = 2\sqrt{n}(\sqrt{\hat{p}_j} - \sqrt{p}_j)$. Condition M3 implies that $\max_{1\le j\le k} Z_j^2 = O_P(1)$. Thus, the inequality (36) in the proof of Theorem 22 can be replaced by $\max_{1\le j\le k} |\sqrt{\hat{p}_j} - \sqrt{q_{\pi(j)}}| \le \frac{\sqrt{C_n}(1+\sqrt{\delta^2})}{\sqrt{n}}$. Then, following the same argument in the proof of Theorem 22, we have

$$\left| T - 4n\sum_{l=1}^{k} (\sqrt{\hat{p}_l} - \sqrt{q_{\pi(l)}})^2 \right| = o_P(1),$$

and it is sufficient to study the asymptotic distribution of $4n\sum_{l=1}^{k}(\sqrt{\hat{p}_l} - \sqrt{q_{\pi(l)}})^2$. Let Δ be a vector with the lth entry being $2\sqrt{n}(\sqrt{\hat{p}_l} - \sqrt{q_{\pi(l)}})$. Then, we have $4n\sum_{l=1}^{k}(\sqrt{\hat{p}_l} - \sqrt{q_{\pi(l)}})^2 = \|Z + \Delta\|^2$. Under Condition M3, $Z \rightsquigarrow N(0, I_k - \sqrt{p}\sqrt{p}^T)$ by Lindeberg's central limit theorem together with an argument of delta's method. Therefore, there exists a random vector W that satisfies $W \rightsquigarrow N(0, I_k)$ and $Z = (I_k - \sqrt{p}\sqrt{p}^T)W$. This gives

$$\begin{aligned} \|Z + \Delta\|^2 &= \|(I_k - \sqrt{p}\sqrt{p}^T)W + (I_k - \sqrt{p}\sqrt{p}^T)\Delta + \sqrt{p}\sqrt{p}^T\Delta\|^2 \\ &= \|(I_k - \sqrt{p}\sqrt{p}^T)W + (I_k - \sqrt{p}\sqrt{p}^T)\Delta\|^2 + \|\sqrt{p}\sqrt{p}^T\Delta\|^2, \end{aligned}$$

where $\|(I_k - \sqrt{p}\sqrt{p}^T)W + (I_k - \sqrt{p}\sqrt{p}^T)\Delta\|^2 \rightsquigarrow \chi^2_{k-1,\delta_1^2}$ and $\|\sqrt{p}\sqrt{p}^T\Delta\|^2 = \delta_2^2$. ■

Theorem 25 *For $\pi = \mathrm{argmin}_{\pi\in S_k} \|\sqrt{p} - \sqrt{q_\pi}\|$, define*

$$\delta_1^2 = 4n\sum_{l=1}^{k}(1 - p_l)\left(\sqrt{p_l} - \sqrt{q_{\pi(l)}}\right)^2,$$

and

$$\delta_2^2 = 4n\sum_{l=1}^{k} p_l\left(\sqrt{p_l} - \sqrt{q_{\pi(l)}}\right)^2.$$

Assume $\limsup_{n\to\infty}(\delta_1^2 + \delta_2^2) < \infty$. Then, under Condition M4, $T_g - \delta_2^2 \rightsquigarrow \chi^2_{k-1,\delta_1^2}$, as n tends to infinity. Moreover, $T_g \ge T_f$ in probability. Furthermore, when $\delta_1^2 + \delta_2^2 = 0$, $T_g \rightsquigarrow \chi^2_{k-1}$, $T_f \rightsquigarrow \chi^2_{d-1}$ and $T_g - T_f \rightsquigarrow \chi^2_{k-d}$.

Proof The proof is largely the same as that of Theorem 24. We only need to replace the $t, \mu_j, \theta_j, \nu_h, X_j$ in the proof of Theorem 24 by $\sqrt{t}, \sqrt{q_j}, \sqrt{p_j}, \sqrt{r_h}, \sqrt{\hat{p}_j}$. Then, by the same argument, we have

$$\left| T_g - 4n\sum_{h=1}^{d}\sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})^2 \right| = o_P(1),$$

and

$$\left| T_g - T_f - 4n\sum_{h=1}^{d}\sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})^2 + 4n\sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h}(\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})\right)^2 \right| = o_P(1).$$

The same argument in the proof of Theorem 24 implies that $T_g - \delta_2^2 \rightsquigarrow \chi^2_{k-1,\delta_1^2}$. The conclusion $T_g \ge T_f$ in probability can be deduced by

$$\begin{aligned} & 4n\sum_{h=1}^{d}\sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})^2 - 4n\sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h}(\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})\right)^2 \\ = \; & 4n\sum_{h=1}^{d}\sum_{j\in\mathcal{C}_h}\left(\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j} - \frac{1}{|\mathcal{C}_h|}\sum_{j\in\mathcal{C}_h}(\sqrt{\hat{p}_{\pi^{-1}(j)}} - \sqrt{q_j})\right)^2 \ge 0. \end{aligned}$$

Now we derive the results under the null distribution. Recall the definition of Z_j in the proof of Theorem 24. The asymptotic distributions of T_g, T_f and $T_g - T_f$ are the same of those of

$$\sum_{j=1}^{k} Z_j^2, \quad \sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h} Z_j\right)^2, \quad \sum_{j=1}^{k} Z_j^2 - \sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h} Z_j\right)^2,$$

respectively under the null hypothesis. According to the argument in the proof of Theorem 24, $Z = (I_k - \sqrt{q}\sqrt{q}^T)W$ with $W \rightsquigarrow N(0, I_k)$. Therefore, $\sum_{j=1}^{k} Z_j^2 \rightsquigarrow \chi^2_{k-1}$.

Define a $k \times d$ matrix Q with $Q_{jh} = \frac{1}{\sqrt{|\mathcal{C}_h|}}$ if $j \in \mathcal{C}_h$ and $Q_{jh} = 0$ if $j \notin \mathcal{C}_h$. It is easy to see that QQ^T is a projection matrix and $Q^TQ = I_d$. Define a vector $\gamma \in \mathbb{R}^d$ whose hth entry is $\gamma_h = \sqrt{|\mathcal{C}_h|r_h}$. It is easy to see that γ is a unit vector. Moreover, we have $\sqrt{q} = Q\gamma$. With the new notation, we get

$$\sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h} Z_j\right)^2 = \|Q^TZ\|^2.$$

The covariance of Q^TZ is

$$Q^T(I_k - \sqrt{q}\sqrt{q}^T)Q = I_d - \gamma\gamma^T.$$

Therefore, $\|Q^TZ\|^2 \rightsquigarrow \chi^2_{d-1}$. Finally,

$$\sum_{j=1}^{k} Z_j^2 - \sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j\in\mathcal{C}_h} Z_j\right)^2 = \|Z\|^2 - \|Q^TZ\|^2 = Z^T(I_k - QQ^T)Z = W^T(I_k - QQ^T)W.$$

Therefore, its asymptotic distribution is χ^2_{k-d}. ■

The results of Theorem 2, Theorem 5, Theorem 7 and Theorem 9 are special cases of Theorem 22, Theorem 23, Theorem 24 and Theorem 25. Next, we give proofs of Theorem 19 and Theorem 21.

Proof [Proof of Theorem 19] Without loss of generality, we can assume that $p_1 = q_1 \leq p_2 = q_2 \leq ... \leq p_k = q_k$. This is just to simplify the notation. In general, such a rearrangement can always be done with extra notation of permutations. Then, $\mathcal{C}_g = \{j_g+1, j_g+2, ..., j_{g+1}\}$ for $g \in [d]$. According to the assumption, $\min_{g\neq h} \min_{j\in\mathcal{C}_g} \min_{l\in\mathcal{C}_h} \sqrt{n}|\sqrt{p_j} - \sqrt{p_l}| = o(1)$. Moreover, it is easy to see that $\max_{j\in[k]} \sqrt{n}|\sqrt{\hat{p}_j} - \sqrt{p_j}| = O_P(1)$ and $\max_{j\in[k]} \sqrt{m}|\sqrt{\hat{q}_j} - \sqrt{q_j}| = O_P(1)$. This leads to the conclusion

$$\mathbb{P}\left(\underline{\mathcal{C}}_g = \overline{\mathcal{C}}_g = \mathcal{C}_g \text{ for all } g \in [d] \text{ and } \underline{d} = \overline{d} = d\right) \to 1,$$

under Condition E.

From now on, the analysis is on the event $\{\underline{\mathcal{C}}_g = \overline{\mathcal{C}}_g = \mathcal{C}_g \text{ for all } g \in [d] \text{ and } \underline{d} = \overline{d} = d\}$. Define $\underline{Z}_j = 2\sqrt{n}(\sqrt{\hat{p}_j} - \sqrt{p_j})$ and $\overline{Z}_j = 2\sqrt{m}(\sqrt{\hat{q}_j} - \sqrt{q_j})$ for $j \in [k]$. The definition implies that $\max_{j\in[k]} |\underline{Z}_j| = O_P(1)$ and $\max_{j\in[k]} |\overline{Z}_j| = O_P(1)$. The definitions of $\underline{r}_g$ and $\overline{r}_g$ give

$$2\sqrt{n}(\sqrt{\underline{r}_g} - \sqrt{r_g}) = \frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g} \underline{Z}_j \quad \text{and} \quad 2\sqrt{m}(\sqrt{\overline{r}_g} - \sqrt{r_g}) = \frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g} \overline{Z}_j.$$

Given that $p_j = q_j = r_g$ for all $j \in \mathcal{C}_g$, we have $\sqrt{n}|\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g}| = O_P(1)$ and $\sqrt{n}|\sqrt{\hat{p}_j} - \sqrt{\overline{r}_g}| = O_P(1)$ for all $j \in \mathcal{C}_g$. We also have $|\sqrt{\hat{q}_j} - \sqrt{\underline{r}_h}|^{-1} = O_P(1)$ and $|\sqrt{\hat{p}_j} - \sqrt{\overline{r}_h}|^{-1} = O_P(1)$ for all $j \in \mathcal{C}_g$ and $h \neq g$.

We first analyze $\underline{g}(t)$. By its definition,

$$\frac{1}{\underline{g}(t)} = \sum_{h=1}^{d} \frac{1}{(\sqrt{t} - \sqrt{\underline{r}_h})^2}.$$

Thus, for any $j \in \mathcal{C}_g$,

$$\frac{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g})^2}{\underline{g}(\hat{q}_j)} = 1 + \sum_{h\in[d]\setminus\{g\}} \frac{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g})^2}{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_h})^2}.$$

Similar to the argument in (40), we get

$$\left|\frac{\underline{g}(\hat{q}_j)}{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g})^2} - 1\right| \leq \sum_{h\in[d]\setminus\{g\}} \frac{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g})^2}{(\sqrt{\hat{q}_j} - \sqrt{\underline{r}_h})^2} = O_P(n^{-1}).$$

With some rearragangements, we get

$$\left|\frac{2nm}{n+m}\sum_{j\in[k]} \underline{g}(\hat{q}_j) - \frac{2nm}{n+m}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g} (\sqrt{\hat{q}_j} - \sqrt{\underline{r}_g})^2\right| = o_P(1).$$

A similar argument also gives

$$\left|\frac{2nm}{n+m}\sum_{j\in[k]} \overline{g}(\hat{p}_j) - \frac{2nm}{n+m}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g} (\sqrt{\hat{p}_j} - \sqrt{\overline{r}_g})^2\right| = o_P(1).$$

Therefore, we obtain the following approximation

$$\left|T_g - \frac{2nm}{m+n}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{n}}\underline{Z}_j - \frac{1}{2\sqrt{m}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2 - \frac{2nm}{m+n}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{m}}\overline{Z}_j - \frac{1}{2\sqrt{n}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j\right)^2\right| = o_P(1).$$

Since

$$\begin{aligned}&\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{n}}\underline{Z}_j - \frac{1}{2\sqrt{m}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2\\ =\ &\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{n}}\underline{Z}_j - \frac{1}{2\sqrt{n}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j\right)^2 + |\mathcal{C}_g|\left(\frac{1}{2\sqrt{n}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j - \frac{1}{2\sqrt{m}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2,\end{aligned}$$

and

$$\begin{aligned}&\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{m}}\overline{Z}_j - \frac{1}{2\sqrt{n}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j\right)^2\\ =\ &\sum_{j\in\mathcal{C}_g}\left(\frac{1}{2\sqrt{m}}\overline{Z}_j - \frac{1}{2\sqrt{m}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2 + |\mathcal{C}_g|\left(\frac{1}{2\sqrt{n}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j - \frac{1}{2\sqrt{m}}\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2,\end{aligned}$$

we have

$$\left|T_g - \frac{m}{2(n+m)}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g}\left(\underline{Z}_j - \frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\underline{Z}_j\right)^2 - \frac{n}{2(n+m)}\sum_{g=1}^{d}\sum_{j\in\mathcal{C}_g}\left(\overline{Z}_j - \frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\overline{Z}_j\right)^2 - \sum_{g=1}^{d}|\mathcal{C}_g|\left(\frac{1}{|\mathcal{C}_g|}\sum_{j\in\mathcal{C}_g}\left(\sqrt{\frac{m}{m+n}}\underline{Z}_j - \sqrt{\frac{n}{m+n}}\overline{Z}_j\right)\right)^2\right| = o_P(1). \tag{49}$$

Next, we analyze $\underline{f}_h(t)$. By its definition,

$$\frac{d\underline{f}_h(t)}{d\sqrt{t}} = \frac{\prod_{g\in[d]\setminus\{h\}}(\sqrt{t} - \sqrt{\underline{r}_g})}{\prod_{g\in[d]\setminus\{h\}}(\sqrt{\underline{r}_h} - \sqrt{\underline{r}_g})}.$$

Therefore, we have

$$\max_{g\in[d]}\sup_{\sqrt{n}|\sqrt{t}-\sqrt{r_g}|\leq\lambda_n}\left|\frac{d\underline{f}_g(t)}{d\sqrt{t}} - 1\right| = o_P(1) \quad \text{and} \quad \max_{g\in[d]\setminus\{h\}}\sup_{\sqrt{n}|\sqrt{t}-\sqrt{r_g}|\leq\lambda_n}\left|\frac{d\underline{f}_h(t)}{d\sqrt{t}}\right| = o_P(1).$$

Using Taylor expansion, we get

$$\sum_{j=1}^{k} \underline{f}_h(\hat{p}_j) - \sum_{j=1}^{k} \underline{f}_h(\hat{q}_j) = \sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_j} - \sqrt{\hat{q}_j}) + o_P(1) \sum_{j=1}^{k} |\sqrt{\hat{p}_j} - \sqrt{\hat{q}_j}|.$$

Then, we have

$$\left| \frac{2nm}{n+m} \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j=1}^{k} \underline{f}_h(\hat{p}_j) - \sum_{j=1}^{k} \underline{f}_h(\hat{q}_j) \right)^2 - \frac{2nm}{n+m} \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_j} - \sqrt{\hat{q}_j}) \right)^2 \right| = o_P(1).$$

The same argument also leads to

$$\left| \frac{2nm}{n+m} \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j=1}^{k} \overline{f}_h(\hat{p}_j) - \sum_{j=1}^{k} \overline{f}_h(\hat{q}_j) \right)^2 - \frac{2nm}{n+m} \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j\in\mathcal{C}_h} (\sqrt{\hat{p}_j} - \sqrt{\hat{q}_j}) \right)^2 \right| = o_P(1).$$

Hence, we have the following approximation,

$$\left| T_f - \sum_{g=1}^{d} |\mathcal{C}_g| \left(\frac{1}{|\mathcal{C}_g|} \sum_{j\in\mathcal{C}_g} \left(\sqrt{\frac{m}{m+n}} \underline{Z}_j - \sqrt{\frac{n}{m+n}} \overline{Z}_j \right) \right)^2 \right| = o_P(1). \tag{50}$$

According to the argument in the proof of Theorem 24, $\underline{Z} = (I_k - \sqrt{p}\sqrt{p}^T)\underline{W}$ with $\underline{W} \rightsquigarrow N(0, I_k)$. Similarly, we also have $\overline{Z} = (I_k - \sqrt{q}\sqrt{q}^T)\overline{W}$ with $\overline{W} \rightsquigarrow N(0, I_k)$. Note that $\overline{W}$ is independent of $\underline{W}$. Recall the definition of the matrix Q and the vector γ in the proof of Theorem 25. Then,

$$\begin{aligned}
\sum_{g=1}^{d} \sum_{j\in\mathcal{C}_g} \left(\underline{Z}_j - \frac{1}{|\mathcal{C}_g|} \sum_{j\in\mathcal{C}_g} \underline{Z}_j \right)^2 &= \underline{Z}^T (I_k - QQ^T) \underline{Z}, \\
\sum_{g=1}^{d} \sum_{j\in\mathcal{C}_g} \left(\overline{Z}_j - \frac{1}{|\mathcal{C}_g|} \sum_{j\in\mathcal{C}_g} \overline{Z}_j \right)^2 &= \overline{Z}^T (I_k - QQ^T) \overline{Z}, \\
\sum_{g=1}^{d} |\mathcal{C}_g| \left(\frac{1}{|\mathcal{C}_g|} \sum_{j\in\mathcal{C}_g} \left(\sqrt{\frac{m}{m+n}} \underline{Z}_j - \sqrt{\frac{n}{m+n}} \overline{Z}_j \right) \right)^2 &= \left\| Q^T \left(\sqrt{\frac{m}{m+n}} \underline{Z} - \sqrt{\frac{n}{m+n}} \overline{Z} \right) \right\|^2.
\end{aligned}$$

Furthermore, we have

$$\begin{aligned}
\underline{Z}^T (I_k - QQ^T) \underline{Z} &= \underline{W}^T (I_k - QQ^T) \underline{W}, \\
\overline{Z}^T (I_k - QQ^T) \overline{Z} &= \overline{W}^T (I_k - QQ^T) \overline{W}, \\
\left\| Q^T \left(\sqrt{\frac{m}{m+n}} \underline{Z} - \sqrt{\frac{n}{m+n}} \overline{Z} \right) \right\|^2 &= \left\| (I_k - \gamma\gamma^T) Q^T \left(\sqrt{\frac{m}{m+n}} \underline{W} - \sqrt{\frac{n}{m+n}} \overline{W} \right) \right\|^2.
\end{aligned}$$

Therefore, the three terms above are asymptotically independent, and their asymptotic distributions are χ^2_{k-d}, χ^2_{k-d} and χ^2_{d-1} under the null, respectively. ■

Proof [Proof of Theorem 21] We will borrow notation and arguments used in the proof of Theorem 23. For example, we keep using the notation $L = \max_{1\le g\le d} \max_{j\in\mathcal{C}_g} \sqrt{n}|\mu_j - \nu_g|$. However, under Condition M2', we have $L = O(1)$ instead of $L = o(1)$. Let C_n be a diverging sequence that satisfies $C_n \to \infty$ and $\frac{C_n^{3/2} \max_{g\neq h} |\overline{n}_{gh}|}{\sqrt{n}} \to 0$. Then, we can use the same analysis in the proof of Theorem 23 that leads to (41) and (46). Note that the only difference is $L = O(1)$, and it will not affect the conclusions of (41) and (46). We still have

$$\left| T_g - n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \nu_h)^2 \right| = o_P(1),$$

and

$$\left| T_f - n \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \right| = o_P(1).$$

By the fact that

$$\begin{aligned}
& n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \nu_h)^2 - n \sum_{h=1}^{d} \frac{1}{|\mathcal{C}_h|} \left(\sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \mu_j) \right)^2 \\
= \; & n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \nu_h)^2 - n \sum_{h=1}^{d} |\mathcal{C}_h| \left(\frac{1}{|\mathcal{C}_h|} \sum_{j\in\mathcal{C}_h} (X_{\pi^{-1}(j)} - \nu_h) \right)^2 \\
= \; & n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} \left(X_{\pi^{-1}(j)} - \frac{1}{|\mathcal{C}_h|} \sum_{j\in\mathcal{C}_h} X_{\pi^{-1}(j)} \right)^2,
\end{aligned}$$

we also have

$$\left| T_g - T_f - n \sum_{h=1}^{d} \sum_{j\in\mathcal{C}_h} \left(X_{\pi^{-1}(j)} - \frac{1}{|\mathcal{C}_h|} \sum_{j\in\mathcal{C}_h} X_{\pi^{-1}(j)} \right)^2 \right| = o_P(1).$$

Therefore, under the null hypothesis $X \sim N(\mu, n^{-1} I_k)$, we have $T_g \rightsquigarrow \chi^2_{k,\tau^2}$, $T_f \rightsquigarrow \chi^2_d$ and $T_g - T_f \rightsquigarrow \chi^2_{k-d,\tau^2}$. ■

10.2 Power Analysis

In this section, we give proofs of Theorem 3, Theorem 6, Theorem 8, Theorem 10 and Theorem 20.

Proof [Proof of Theorem 3] We first assume $n\ell(\theta, \mu)^2 \to \infty$ and derive $T \to \infty$ in probability. Note that for each $\pi \in S_k$,

$$n \sum_{j=1}^{k} (\theta_j - \mu_{\pi(j)})^2 \le 2n \sum_{j=1}^{k} (X_j - \theta_j)^2 + 2n \sum_{j=1}^{k} (X_j - \mu_{\pi(j)})^2.$$

Therefore,

$$n\ell(\theta,\mu)^2 \le 2\sum_{j=1}^{k} Z_j^2 + 2n\ell(X,\mu)^2,$$

where $Z_j \sim N(0,1)$. The fact that $2\sum_{j=1}^{k} Z_j^2 = O_P(1)$ and the assumption $n\ell(\theta,\mu)^2 \to \infty$ implies that $n\ell(X,\mu)^2 \to \infty$ in probability. Suppose we can show $T = O_P(1)$ implies $n\ell(X,\mu)^2 = O_P(1)$, then $n\ell(X,\mu)^2 \to \infty$ in probability must implies $T \to \infty$ in probability.

Now we suppose a bound $T \le B = O(1)$, and it is sufficient to derive a bound for $n\ell(X,\mu)^2$. For each $j = 1,...,k$, we shorthand the power sums $p_j(X_1,...,X_k)$ and $p_j(\mu_1,...,\mu_k)$ by $p_j(X)$ and $p_j(\mu)$. Similarly, the elementary symmetric polynomials $e_j(X_1,...,X_k)$ and $e_j(\mu_1,...,\mu_k)$ are shorthanded by $e_j(X)$ and $e_j(\mu)$. Define a vector $\Delta \in \mathbb{R}^k$ with the jth entry being $\Delta_j = \frac{1}{j}\sum_{h=1}^{k} X_h^j - \frac{1}{j}\sum_{h=1}^{k} \mu_h^j$. Recall the definition of the matrix $E(\mu_1,...,\mu_k)$. Then,

$$T = n\|E(\mu_1,...,\mu_k)\Delta\|^2.$$

We use $\lambda_{\max}(\cdot)$ and $\lambda_{\min}(\cdot)$ to denote the largest and the smallest eigenvalues. By the fact that $V(\mu_1,...,\mu_k)E(\mu_1,...,\mu_k) = I_k$, we have

$$T \ge n\lambda_{\min}(E(\mu_1,...,\mu_k)^T E(\mu_1,...,\mu_k))\|\Delta\|^2 \ge \frac{n\|\Delta\|^2}{\lambda_{\max}(V(\mu_1,...,\mu_k)^T V(\mu_1,...,\mu_k))}.$$

The bound $T \le B$ then leads to

$$\|\Delta\|^2 \le \frac{\lambda_{\max}(V(\mu_1,...,\mu_k)^T V(\mu_1,...,\mu_k))B}{n} = O\left(\frac{B}{n}\right). \tag{51}$$

Therefore, $|p_j(X) - p_j(\mu)|^2 = O(B/n)$ for each $j \in [k]$. By Newton's identities, we can deduce $|e_j(X) - e_j(\mu)|^2 = O(B/n)$ for each $j \in [k]$. Define

$$f(t) = \prod_{j=1}^{k}(t-\mu_j), \quad \hat{f}(t) = \prod_{j=1}^{k}(t-X_j).$$

The relation between the two polynomials and the elementary symmetric polynomials is given in (3). Using (3), we give a bound for $|f(X_l)|$.

$$|f(X_l)| = |f(X_l) - \hat{f}(X_l)| \le \sum_{j=0}^{k} |e_{k-j}(X) - e_{k-j}(\mu)||X_l|^j.$$

Since $|X_l|^2 \le p_2(X) \le p_2(\mu) + |p_2(X) - p_2(\mu)| = O(1)$, we have $|f(X_l)|^2 = O(B/n)$. The following proposition is useful and will be proved in the end.

Proposition 26 *For any $\mu_1,...,\mu_k$, we have*

$$|f(t)| \ge \min_{1\le j\le k}|t-\mu_j| \prod_{1\le j<l\le k}\left|\frac{\mu_j-\mu_l}{2}\right|.$$

By this inequality, we have

$$\max_{1\le l\le k}\min_{1\le j\le k}(X_l-\mu_j)^2 = O\left(\frac{B}{n}\right). \tag{52}$$

Therefore, there exists a sequence $\sigma(1),...,\sigma(k)$ such that

$$\max_{1\le j\le k}(X_j-\mu_{\sigma(j)})^2 = O\left(\frac{B}{n}\right).$$

Since

$$\prod_{j=1}^{k}|t-\mu_{\sigma(j)}| \le 2^k\prod_{j=1}^{k}|t-X_j| + 2^k\prod_{j=1}^{k}|X_j-\mu_{\sigma(j)}| = 2^k|\hat{f}(t)| + O\left(\left(\frac{B}{n}\right)^{k/2}\right),$$

and

$$|\hat{f}(\mu_l)| = |\hat{f}(\mu_l) - f(\mu_l)| \le \sum_{j=0}^{k}|e_{k-j}(X) - e_{k-j}(\mu)||\mu_l|^j = O\left(\sqrt{\frac{B}{n}}\right),$$

we have

$$\prod_{j=1}^{k}|\mu_l-\mu_{\sigma(j)}| = O\left(\sqrt{\frac{B}{n}}\right),$$

which holds for every $l = 1,...,k$. The fact that $\mu_1,...,\mu_k$ are k different fixed number implies σ must be an element of S_k. Hence, the bound (52) implies $n\ell(X,\mu)^2 = O(B)$, and the proof of one direction is complete.

For the other direction, it is sufficient to show that $n\ell(\theta,\mu) = O(1)$ implies $T = O_P(1)$. This can be shown using the same argument in the proof of Theorem 22. ■

Proof [Proof of Proposition 26] We first consider the case $k = 2$, where $f(t) = (t-\mu_1)(t-\mu_2)$. Suppose $|t-\mu_1| \le |t-\mu_2|$, then $|t-\mu_2| \ge \frac{|\mu_1-\mu_2|}{2}$. Thus, $|f(t)| \ge \frac{|\mu_1-\mu_2|}{2}\min\{|t-\mu_1|, |t-\mu_2|\}$. The same argument also works when $|t-\mu_1| > |t-\mu_2|$. When $k = 3$,

$$\begin{aligned} |f(t)| &\ge |t-\mu_3|\frac{|\mu_1-\mu_2|}{2}\min\{|t-\mu_1|, |t-\mu_2|\} \\ &= \frac{|\mu_1-\mu_2|}{2}\min\left\{|t-\mu_1||t-\mu_3|, |t-\mu_2||t-\mu_3|\right\}. \end{aligned}$$

The inequality for $k = 2$ can be used to lower bound both $|t-\mu_1||t-\mu_3|$ and $|t-\mu_2||t-\mu_3|$. This gives the desired result for $k = 3$. A standard mathematical induction argument leads to inequality for all k. ■

Proof [Proof of Theorem 6] According to the argument that we have used in the proof of Theorem 3, we need to show $T_f = O_P(1)$ and $T_g = O_P(1)$ imply $n\ell(X,\mu)^2 = O_P(1)$ for the proof of the first direction.

Suppose $T_f \le B_1 = O(1)$ and $T_g \le B_2 = O(1)$. It is sufficient to derive a bound for $n\ell(X,\mu)^2$. We first derive an inequality for $g(t)$. Since

$$\max_{1\le g\le d}\prod_{h\in[d]\setminus\{g\}}(t-\nu_h)^2 \le \sum_{g=1}^{d}\prod_{h\in[d]\setminus\{g\}}(t-\nu_h)^2 \le d\max_{1\le g\le d}\prod_{h\in[d]\setminus\{g\}}(t-\nu_h)^2,$$

we have

$$\frac{1}{d}\min_{1\le g\le d}(t-\nu_g)^2 \le g(t) \le \min_{1\le g\le d}(t-\nu_g)^2. \tag{53}$$

Therefore, $T_g \le B_2$ implies that $\sum_{j=1}^{k}\min_{1\le g\le d}(X_j-\nu_g)^2 \le \frac{dB_2}{n}$. This implies the existence of a sequence $\sigma(1),...,\sigma(k)$ such that $\max_{1\le j\le k}(X_j-\nu_{\sigma(j)})^2 \le \frac{dB_2}{n} = O(B_2/n)$. It further implies $\max_{1\le h\le d}\max_{1\le j\le k}|X_j^h-\nu_{\sigma(j)}^h| = O(\sqrt{B_2/n})$. Define $\hat{\mathcal{C}}_g = \{j\in[k]:\sigma(j)=g\}$ for each $g\in[d]$. Then

$$\sum_{h=1}^{d-1}\left(\sum_{j=1}^{k}X_j^h - \sum_{g=1}^{d}|\hat{\mathcal{C}}_g|\nu_g^h\right)^2 = O\left(\frac{B_2}{n}\right).$$

Using the same argument in deriving (51), we can also get the bound

$$\sum_{h=1}^{d-1}\left(\sum_{j=1}^{k}X_j^h - \sum_{g=1}^{d}|\mathcal{C}_g|\nu_g^h\right)^2 = O\left(\frac{B_1}{n}\right).$$

The inequalities in the last two displays, together with the equality $\sum_{g=1}^{d}|\hat{\mathcal{C}}_g| = \sum_{g=1}^{d}|\mathcal{C}_g|$, give

$$\sum_{h=0}^{d-1}\left(\sum_{g=1}^{d}|\hat{\mathcal{C}}_g|\nu_g^h - \sum_{g=1}^{d}|\mathcal{C}_g|\nu_g^h\right)^2 = O\left(\frac{B_1+B_2}{n}\right).$$

Define a vector $r\in\mathbb{R}^d$, with its gth entry being $|\hat{\mathcal{C}}_g|-|\mathcal{C}_g|$. Then,

$$\sum_{h=0}^{d-1}\left(\sum_{g=1}^{d}|\hat{\mathcal{C}}_g|\nu_g^h - \sum_{g=1}^{d}|\mathcal{C}_g|\nu_g^h\right)^2 = \|V(\nu_1,...,\nu_d)r\|^2 \ge \lambda_{\min}(V(\nu_1,...,\nu_d)^TV(\nu_1,...,\nu_d))\|r\|^2.$$

When $\nu_1,...,\nu_d$ are d different numbers, we have $\lambda_{\min}(V(\nu_1,...,\nu_d)^TV(\nu_1,...,\nu_d))>0$, and thus $\|r\|^2 = O\left(\frac{B_1+B_2}{n}\right)$. Since $\|r\|^2$ is an integer, we must have $\|r\|^2=0$, which gives $|\hat{\mathcal{C}}_g| = |\mathcal{C}_g|$ for any $g\in[d]$. From this we can deduce that $n\ell(X,\mu)^2 = O(B_2)$.

For the other direction, it is sufficient to show that $n\ell(\theta,\mu)^2 = O(1)$ implies $T_f = O_P(1)$ and $T_g = O_P(1)$. This can be shown using the same argument in the proof of Theorem 23. ■

Proof [Proofs of Theorem 8 and Theorem 10] The proofs are the same as those of Theorem 3 and Theorem 6. ■

Proof [Proof of Theorem 20] First of all, we have $\max_{j\in[k]}\sqrt{n}|\sqrt{\hat{q}_j}-\sqrt{q_j}| = O_P(1)$. This gives that $\mathbb{P}(\overline{\mathcal{C}}_g = \mathcal{C}_g \text{ for all } g\in[d] \text{ and } \overline{d}=d)\to 1$. From now on, the analysis is on the event $\{\overline{\mathcal{C}}_g = \mathcal{C}_g \text{ for all } g\in[d] \text{ and } \overline{d}=d\}$. Since we also have $\max_{j\in[k]}\sqrt{n}|\sqrt{\hat{p}_j}-\sqrt{p_j}| = O_P(1)$, the statement $n\ell(p,q)^2\to\infty$ is equivalent to $n\ell(\hat{p},q)^2\to\infty$ in probability. Therefore, we only need to establish the equivalence between $n\ell(\hat{p},q)^2\to\infty$ in probability and the power of the test goes to one.

In the first direction of the proof, we suppose that $T_f \le B_1 = O_P(1)$ and $T_g \le B_2 = O_P(1)$, and we will show $n\ell(\hat{p},q)^2 = O_P(1)$. The bound $T_g \le B_2 = O_P(1)$ implies that

$$\frac{2nm}{n+m}\sum_{j=1}^{k}\overline{g}(\hat{p}_j) \le B_2.$$

By the definition of $\overline{g}(\cdot)$, we have $\overline{g}(t) \ge d^{-1}\min_{g\in[d]}(\sqrt{t}-\sqrt{r_g})^2$. This implies the bound

$$\sum_{j=1}^{k}\min_{g\in[d]}\left(\sqrt{\hat{p}_j}-\sqrt{r_g}\right)^2 \le \frac{dB_2(n+m)}{nm} = O_P(n^{-1}).$$

Since $\max_{g\in[d]}\sqrt{n}|\sqrt{r_g}-\sqrt{r_g}| = O_P(1)$, we deduce

$$\sum_{j=1}^{k}\min_{g\in[d]}\left(\sqrt{\hat{p}_j}-\sqrt{r_g}\right)^2 = O_P(n^{-1}).$$

Then, there must exist $\sigma(1),...,\sigma(k)$ such that $\max_{j\in[k]}|\sqrt{\hat{p}_j}-\sqrt{r_{\sigma(j)}}|^2 = O_P(n^{-1})$. It further implies that $\max_{h\in[d]}\max_{j\in[k]}|(\sqrt{\hat{p}_j})^h-(\sqrt{r_{\sigma(j)}})^h| = O_P(n^{-1/2})$. Define $\hat{\mathcal{C}}_g = \{j\in[k]:\sigma(j)=g\}$ for each $g\in[d]$. Then, we have

$$\sum_{h=1}^{d-1}\left(\sum_{j=1}^{k}(\sqrt{\hat{p}_j})^h - \sum_{g=1}^{d}|\hat{\mathcal{C}}_g|(\sqrt{r_g})^h\right)^2 = O_P(n^{-1}). \tag{54}$$

Note that

$$\begin{aligned}
T_f &\ge \frac{2nm}{n+m}\sum_{h=1}^{d}\frac{1}{|\mathcal{C}_h|}\left(\sum_{j=1}^{k}\overline{f}_h(\hat{p}_j)-\sum_{j=1}^{k}\overline{f}_h(\hat{q}_j)\right)^2\\
&\ge \frac{2nm}{d(n+m)}\sum_{h=1}^{d}\left(\sum_{j=1}^{k}\overline{f}_h(\hat{p}_j)-\sum_{j=1}^{k}\overline{f}_h(\hat{q}_j)\right)^2\\
&= \frac{2nm}{d(n+m)}\|E(\sqrt{r}_1,...,\sqrt{r}_d)\Delta\|^2,
\end{aligned}$$

where Δ is a d-dimensional vector with $\Delta_h = \frac{1}{h}\sum_{j=1}^{k}(\sqrt{\hat{p}_j})^h - \frac{1}{h}\sum_{j=1}^{k}(\sqrt{\hat{q}_j})^h$. Thus, the bound $T_f \le B_1 = O_P(1)$ implies that

$$\|E(\sqrt{r}_1,...,\sqrt{r}_d)\Delta\|^2 = O_P(n^{-1}).$$

Since $\lambda_{\min}(E(\sqrt{r_1},...,\sqrt{r_d})^T E(\sqrt{r_1},...,\sqrt{r_d}))$ is a positive constant that is bounded away from 0, and

$$|\lambda_{\min}(E(\sqrt{\bar{r}_1},...,\sqrt{\bar{r}_d})^T E(\sqrt{\bar{r}_1},...,\sqrt{\bar{r}_d})) - \lambda_{\min}(E(\sqrt{r_1},...,\sqrt{r_d})^T E(\sqrt{r_1},...,\sqrt{r_d}))| = o_P(1),$$

we have $\|\Delta\|^2 = O_P(n^{-1})$, which further leads to

$$\sum_{h=1}^{d-1}\left(\sum_{j=1}^{k}(\sqrt{\hat{p}_j})^h - \sum_{g=1}^{d}|\mathcal{C}_g|(\sqrt{r_g})^h\right)^2 = O_P(n^{-1}), \tag{55}$$

by using the fact that $\max_{j\in[k]}\sqrt{n}|\sqrt{\hat{q}_j}-\sqrt{q_j}| = O_P(1)$ and Condition E. The two inequalities (54) and (55), together with the fact that $\sum_{g=1}^d|\hat{\mathcal{C}}_g| = \sum_{g=1}^d|\mathcal{C}_g|$, imply

$$\sum_{h=0}^{d-1}\left(\sum_{g=1}^{d}|\hat{\mathcal{C}}_g|(\sqrt{r_g})^h - \sum_{g=1}^{d}|\mathcal{C}_g|(\sqrt{r_g})^h\right)^2 = O_P(n^{-1}).$$

The same argument used in the proof of Theorem 6 implies that $|\hat{\mathcal{C}}_g| = |\mathcal{C}_g|$ for all $g \in [d]$. Therefore, together with $\max_{j\in[k]}|\sqrt{\hat{p}_j} - \sqrt{r_{\sigma(j)}}|^2 = O_P(n^{-1})$, we obtain the conclusion $n\ell(\hat{p}, q) = O_P(1)$.

For the other direction, when $n\ell(p,q)^2 = O(1)$, the approximations (49) and (50) in the proofs of Theorem 19 hold with bounds at the order of $O_P(1)$. This leads to $T_f = O_P(1)$ and $T_g = O_P(1)$. ■

10.3 Minimax Upper and Lower Bounds

In this section, we prove all results in Section 6. We first give proofs for the lower bounds, and then for the upper bounds.

Proof [Proof of Theorem 11] We first observe an inequality $|\eta_{jl}|+|\eta_{lj}| \geq \frac{2}{|\mu_l-\mu_j|}$, which has been derived in the proof of Theorem 22. Thus, Condition M1 implies $\sqrt{n}|\mu_j - \mu_l| \to \infty$ for any $j \neq l$. Consider the set

$$\bar{\Theta}_\delta = \left\{\theta : \|\theta - \mu\| = \frac{\delta}{\sqrt{n}}\right\}.$$

For each $\theta \in \bar{\Theta}_\delta$, $|\theta_j - \mu_j|^2 \leq \frac{\delta^2}{n}$, which implies μ_j is the closest element to θ_j in the set $\{\mu_1,...,\mu_k\}$. Therefore, $\ell(\theta,\mu) = \|\theta-\mu\| = \delta/\sqrt{n}$, which implies $\bar{\Theta}_\delta \subset \Theta_\delta$. This gives the lower bound

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1}\left\{\mathbb{P}_\mu\phi + \sup_{\theta\in\bar{\Theta}_\delta}\mathbb{P}_\theta(1-\phi)\right\}.$$

Consider the uniform distribution Π on $\bar{\Theta}_\delta$. Then,

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1}\left\{\mathbb{P}_\mu\phi + \int\mathbb{P}_\theta(1-\phi)d\Pi(\theta)\right\}.$$

By Neyman-Pearson lemma, the optimal testing function ϕ is given by

$$\phi = \left\{\frac{d\int\mathbb{P}_\theta d\Pi(\theta)}{d\mathbb{P}_\mu} > 1\right\}.$$

Using the property of Π, we have

$$\begin{aligned}\frac{d\int\mathbb{P}_\theta d\Pi(\theta)}{d\mathbb{P}_\mu} &= \int\frac{d\mathbb{P}_\theta}{d\mathbb{P}_\mu}d\Pi(\theta)\\ &= \int\exp\left(-\frac{n}{2}\|\theta-\mu\|^2 + n\langle X-\mu,\theta-\mu\rangle\right)d\Pi(\theta)\\ &= e^{-\delta^2/2}\int\exp\left(n\langle X-\mu,\theta-\mu\rangle\right)d\Pi(\theta).\end{aligned}$$

Let $\bar{\Pi}$ be the uniform distribution on the unit sphere $\{\theta : \|\theta\| = 1\}$, and then we have

$$\int\exp\left(n\langle X-\mu,\theta-\mu\rangle\right)d\Pi(\theta) = \int\exp\left(\delta\sqrt{n}\langle X-\mu,\theta\rangle\right)d\bar{\Pi}(\theta).$$

Let f be the marginal density of the first coordinate of $\theta \sim \bar{\Pi}$. Then, $f(t) \propto (1-t^2)^{\frac{k-3}{2}}$. The uniformity of $\bar{\Pi}$ implies that

$$\int\exp\left(\delta\sqrt{n}\langle X-\mu,\theta\rangle\right)d\bar{\Pi}(\theta) = \frac{\int_{-1}^1\exp\left(\delta\sqrt{n}\|X-\mu\|t\right)(1-t^2)^{\frac{k-3}{2}}dt}{\int_{-1}^1(1-t^2)^{\frac{k-3}{2}}dt}. \tag{56}$$

Therefore, we can write the quantity in the above display as $F(\sqrt{n}\|X-\mu\|)$. Since

$$F'(x) = \frac{\int_0^1(e^{\delta xt}-e^{-\delta xt})\delta t(1-t^2)^{\frac{k-3}{2}}dt}{\int_{-1}^1(1-t^2)^{\frac{k-3}{2}}dt} > 0, \text{ for } x > 0,$$

the testing statistic $\frac{d\int\mathbb{P}_\theta d\Pi(\theta)}{d\mathbb{P}_\mu}$ is an increasing function of $\|X-\mu\|^2$. This implies

$$\phi = \left\{n\|X-\mu\|^2 \geq t\right\},$$

for some $t > 0$. Note that $n\|X-\mu\|^2 \sim \chi^2_k$ under $\mathbb{P}_\mu$, and $n\|X-\mu\|^2 \sim \chi^2_{k,\delta^2}$ under any $\mathbb{P}_\theta$ with $\theta \in \bar{\Theta}_\delta$. Hence,

$$R_n(k,\delta) \geq \inf_{t>0}\left\{\mathbb{P}(\chi^2_k \geq t) + \mathbb{P}(\chi^2_{k,\delta^2} < t)\right\}.$$

This completes the proof. ■

Proof [Proof of Theorem 13] Since $|\bar{\eta}_{gh}| + |\bar{\eta}_{hg}| \geq \frac{2}{|\nu_g-\nu_h|}$, it is implied by Condition M2 that $\sqrt{n}|\nu_g - \nu_h| \to \infty$ for any $g \neq h$. Moreover, for any $j \in \mathcal{C}_h$, $|\mu_j - \nu_h| = o(n^{-1/2})$. Under these assumptions, for any θ such that $\|\theta-\mu\| = \frac{\delta}{\sqrt{n}}$, there exists a $\pi \in S_k$ that depends

on θ and $\|\theta_\pi - \mu\| = \ell(\theta,\mu) = \frac{\delta}{\sqrt{n}}(1+\epsilon_\theta)$. Moreover, $|\epsilon_\theta| = o(1)$ uniformly over all θ that satisfies $\|\theta - \mu\| = \frac{\delta}{\sqrt{n}}$. Define

$$\theta' = \mu + \frac{1}{1+\epsilon_\theta}(\theta - \mu). \tag{57}$$

Then, $\|\theta' - \mu\| = \frac{\delta_\theta}{\sqrt{n}}$ and $\ell(\theta',\mu) = \frac{\delta}{\sqrt{n}}$, where $\delta_\theta = \frac{\delta}{1+\epsilon_\theta}$. We use the notation R to denote the operator $R : \theta \mapsto R(\theta) = \theta'$ defined by (57). By the definition, a useful property is $\frac{R(\theta)-\mu}{\|R(\theta)-\mu\|} = \frac{\theta-\mu}{\|\theta-\mu\|}$. Consider the set

$$\bar{\Theta}_\delta = \left\{ R(\theta) : \|\theta - \mu\| = \frac{\delta}{\sqrt{n}} \right\}.$$

This definition immediately implies $\bar{\Theta}_\delta \subset \Theta_\delta$. Note that each element in $\bar{\Theta}_\delta$ can be represented as

$$R(\theta) = \mu + \frac{\delta_\theta}{\sqrt{n}} \frac{\theta - \mu}{\|\theta - \mu\|}.$$

Since there is a one-to-one relation between $\frac{\theta-\mu}{\|\theta-\mu\|}$ and a unit vector v, we can also write each element in $\bar{\Theta}_\delta$ as $\mu + \frac{\delta_v}{\sqrt{n}}v$. Consider a uniform probability measure $\bar{\Pi}$ on $\{v : \|v\| = 1\}$. Then, by the same argument in the proof of Theorem 11,

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1} \left\{ \mathbb{P}_\mu \phi + \int \mathbb{P}_{\mu + \frac{\delta_v}{\sqrt{n}}v}(1-\phi) d\bar{\Pi}(v) \right\},$$

and the likelihood ratio is

$$\mathcal{L} = \int \frac{d\mathbb{P}_{\mu+\frac{\delta_v}{\sqrt{n}}v}}{d\mathbb{P}_\mu} d\bar{\Pi}(v) = \int \exp\left(-\delta_v^2/2 + \delta_v\sqrt{n}\left\langle X - \mu, v\right\rangle\right) d\bar{\Pi}(v).$$

Under the assumption, there exist δ_- and δ_+ such that $\delta_- \leq \delta_v \leq \delta_+$ for all v and $\delta_-/\delta = 1 + o(1)$ and $\delta_+/\delta = 1 + o(1)$. We introduce the upper and lower brackets of $\mathcal{L}$ as

$$\begin{aligned}
\mathcal{L}_- &= \min\left\{ \int \exp\left(-\delta_+^2/2 + \delta_-\sqrt{n}\left\langle X-\mu, v\right\rangle\right) d\bar{\Pi}(v), \right. \\
&\qquad \left. \int \exp\left(-\delta_+^2/2 + \delta_+\sqrt{n}\left\langle X-\mu, v\right\rangle\right) d\bar{\Pi}(v) \right\}, \\
\mathcal{L}_+ &= \min\left\{ \int \exp\left(-\delta_-^2/2 + \delta_-\sqrt{n}\left\langle X-\mu, v\right\rangle\right) d\bar{\Pi}(v), \right. \\
&\qquad \left. \int \exp\left(-\delta_-^2/2 + \delta_+\sqrt{n}\left\langle X-\mu, v\right\rangle\right) d\bar{\Pi}(v) \right\}.
\end{aligned}$$

The definitions imply $\mathcal{L}_- \leq \mathcal{L} \leq \mathcal{L}_+$. Define the function

$$F_\delta(x) = \frac{\int_{-1}^1 \exp(\delta x t)(1-t^2)^{\frac{k-3}{2}} dt}{\int_{-1}^1 (1-t^2)^{\frac{k-3}{2}} dt}. \tag{58}$$

By (56), we have

$$\begin{aligned}
\mathcal{L}_- &= e^{-\delta_+^2/2}\min\{F_{\delta_-}(\sqrt{n}\|X-\mu\|), F_{\delta_+}(\sqrt{n}\|X-\mu\|)\}, \\
\mathcal{L}_+ &= e^{-\delta_-^2/2}\max\{F_{\delta_-}(\sqrt{n}\|X-\mu\|), F_{\delta_+}(\sqrt{n}\|X-\mu\|)\}.
\end{aligned}$$

Define $\phi = \mathbb{I}\{\mathcal{L} > 1\}$, $\phi_- = \mathbb{I}\{\mathcal{L}_- > 1\}$ and $\phi_+ = \mathbb{I}\{\mathcal{L}_+ > 1\}$. We have the inequality $\phi_- \leq \phi \leq \phi_+$. For $\theta = \mathbb{E}X = \mu$, $\|\sqrt{n}(X-\mu)\|^2 \sim \chi_k^2$. Thus, let $Z \sim N(0, I_k)$, and then we have

$$\begin{aligned}
\mathbb{P}_\mu \phi &\geq \mathbb{P}_\mu(\mathcal{L}_- > 1) \\
&= \mathbb{P}\left(e^{-\delta_+^2/2}\min\{F_{\delta_-}(\|Z\|), F_{\delta_+}(\|Z\|)\} > 1\right) \\
&\to \mathbb{P}\left(e^{-\delta^2/2}F_\delta(\|Z\|) > 1\right).
\end{aligned}$$

For the alternative $\theta = \mu + \frac{\delta_v}{\sqrt{n}}v \in \bar{\Theta}_\delta$, $\|\sqrt{n}(X-\mu)\|^2 \sim \chi_{k^2,\delta_v^2}$, where $\delta_v^2 \in [\delta_-, \delta_+]$. Then,

$$\begin{aligned}
\mathbb{P}_{\mu+\frac{\delta_v}{\sqrt{n}}v}(1-\phi) &\geq \mathbb{P}_{\mu+\frac{\delta_v}{\sqrt{n}}v}(\mathcal{L}_+ \leq 1) \\
&= \mathbb{P}\left(e^{-\delta_-^2/2}\max\{F_{\delta_-}(\|Z+\delta_v v\|), F_{\delta_+}(\|Z+\delta_v v\|)\} \leq 1\right) \\
&\geq \mathbb{P}\left(e^{-\delta_-^2/2}\max\{F_{\delta_-}(\|Z+\delta_+ v\|), F_{\delta_+}(\|Z+\delta_+ v\|)\} \leq 1\right) \\
&\to \mathbb{P}\left(e^{-\delta^2/2}F_\delta(\|Z+\delta v\|) \leq 1\right)
\end{aligned}$$

Note that $\mathbb{P}\left(e^{-\delta^2/2}F_\delta(\|Z+\delta v\|) \leq 1\right)$ is independent of v. Therefore, by the fact that $F_\delta(x)$ is increasing on $x > 0$, we have

$$\begin{aligned}
R_n(k,n) &\geq \mathbb{P}_\mu(\mathcal{L}_- > 1) + \inf_{\|v\|=1} \mathbb{P}_{\mu+\frac{\delta_v}{\sqrt{n}}v}(\mathcal{L}_+ \leq 1) \\
&\geq (1+o(1))\left\{ \mathbb{P}\left(e^{-\delta^2/2}F_\delta(\|Z\|) > 1\right) + \inf_{\|v\|=1} \mathbb{P}\left(e^{-\delta^2/2}F_\delta(\|Z+\delta v\|) \leq 1\right) \right\} \\
&\geq (1+o(1)) \inf_{t>0}\left\{ \mathbb{P}(\chi_k^2 \geq t) + \mathbb{P}(\chi_{k,\delta^2}^2 < t) \right\}.
\end{aligned}$$

The proof is complete. ■

Proof [Proof of Theorem 15] Note that Condition M3 implies $\sqrt{n}|\sqrt{q_j} - \sqrt{q_l}| \to \infty$ for any $j \neq l$. Consider the set

$$\bar{\mathcal{P}}_\delta = \left\{ p : \|\sqrt{p} - \sqrt{q}\| = \frac{\delta}{\sqrt{n}} \right\}.$$

For each $p \in \bar{\mathcal{P}}_\delta$, $|\sqrt{p_j} - \sqrt{q_j}|^2 \leq \frac{\delta^2}{n}$, which implies $\sqrt{q_j}$ is the closest element to $\sqrt{p_j}$ in the set $\{\sqrt{q_1}, ..., \sqrt{q_k}\}$. Therefore, $\ell(p,q) = \|\sqrt{p} - \sqrt{q}\| = \delta/\sqrt{n}$, which implies $\bar{\mathcal{P}}_\delta \subset \mathcal{P}_\delta$. This gives the lower bound

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1} \left\{ \mathbb{P}_q \phi + \sup_{p\in\bar{\mathcal{P}}_\delta} \mathbb{P}_p(1-\phi) \right\}.$$

Let Π be the uniform distribution on the sphere $\{v : \|v - \sqrt{q}\| = \delta/\sqrt{n}\}$. Then,

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1}\left\{\mathbb{P}_q\phi + \int \mathbb{P}_p(1-\phi)d\Pi(\sqrt{p})\right\}.$$

By Neyman-Pearson lemma, the optimal testing function ϕ is given by

$$\phi = \left\{\frac{d\int \mathbb{P}_p d\Pi(\sqrt{p})}{d\mathbb{P}_q} > 1\right\}.$$

By the definition, we have

$$\mathcal{L} = \frac{d\int \mathbb{P}_p d\Pi(\sqrt{p})}{d\mathbb{P}_q} = \int \exp\left(n\sum_{j=1}^k q_j \log\frac{p_j}{q_j} + n\sum_{j=1}^k(\hat{p}_j - q_j)\log\frac{p_j}{q_j}\right)d\Pi(\sqrt{p}),$$

where $\hat{p}_j = \frac{1}{n}\sum_{i=1}^n \mathbb{I}\{X_i = j\}$. Note that $\log\sqrt{\frac{p_j}{q_j}} = \log\left(1 + \frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right)$. By Condition M3, $\max_{1\leq j\leq k}\left|\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right| = o(1)$. Therefore,

$$\max_{1\leq j\leq k}\frac{\left|\log\sqrt{\frac{p_j}{q_j}} - \frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right|}{\left|\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right|^2} = O(1), \tag{59}$$

and

$$\max_{1\leq j\leq k}\frac{\left|\log\sqrt{\frac{p_j}{q_j}} - \frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}} + \frac{1}{2}\left(\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right)^2\right|}{\left|\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right|^3} = O(1). \tag{60}$$

Since

$$\sum_{j=1}^k q_j\left[\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}} - \frac{1}{2}\left(\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}}\right)^2\right] = -\|\sqrt{p}-\sqrt{q}\|^2.$$

By (60), we have

$$\sum_{j=1}^k q_j\log\frac{p_j}{q_j} = -(1+o(1))2\|\sqrt{p}-\sqrt{q}\|^2.$$

Under Condition M3, $\hat{p}_j/p_j = 1 + o_P(1)$, and this implies $\hat{p}_j/q_j = 1 + o_P(1)$. Therefore,

$$\sum_{j=1}^k(\hat{p}_j - q_j)\frac{\sqrt{p_j}-\sqrt{q_j}}{\sqrt{q_j}} = 2(1+o_P(1))\sum_{j=1}^k(\sqrt{\hat{p}_j}-\sqrt{q_j})(\sqrt{p_j}-\sqrt{q_j}). \tag{61}$$

By (59), we have

$$\sum_{j=1}^k(\hat{p}_j - q_j)\log\frac{p_j}{q_j} = 4(1+o_P(1))\sum_{j=1}^k(\sqrt{\hat{p}_j}-\sqrt{q_j})(\sqrt{p_j}-\sqrt{q_j}). \tag{62}$$

The approximations (61) and (62) imply the existence of δ_- and δ_+ that satisfies $\delta_- = (1+o(1))\delta$, $\delta_+ = (1+o(1))\delta$. Moreover, on an event E with probability $1-o(1)$ under both null and alternative, the following inequalities hold:

$$-2\delta_+^2 \leq n\sum_{j=1}^k q_j\log\frac{p_j}{q_j} \leq -2\delta_-^2,$$

$$\sum_{j=1}^k(\hat{p}_j - q_j)\log\frac{p_j}{q_j} \leq \max\left\{\frac{4\delta_-}{\sqrt{n}}\left\langle\sqrt{\hat{p}}-\sqrt{q}, \frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}\right\rangle, \frac{4\delta_+}{\sqrt{n}}\left\langle\sqrt{\hat{p}}-\sqrt{q}, \frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}\right\rangle\right\},$$

and

$$\sum_{j=1}^k(\hat{p}_j - q_j)\log\frac{p_j}{q_j} \geq \min\left\{\frac{4\delta_-}{\sqrt{n}}\left\langle\sqrt{\hat{p}}-\sqrt{q}, \frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}\right\rangle, \frac{4\delta_+}{\sqrt{n}}\left\langle\sqrt{\hat{p}}-\sqrt{q}, \frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}\right\rangle\right\}.$$

We introduce the upper and lower brackets of the $\mathcal{L}$ as

$$\begin{aligned}
\mathcal{L}_- &= \min\left\{\int\exp\left(-2\delta_+^2 + 4\delta_-\sqrt{n}\left\langle\sqrt{\hat{p}}-\sqrt{q}, v\right\rangle\right)d\bar{\Pi}(v),\right. \\
&\qquad \left.\int\exp\left(-2\delta_+^2 + 4\delta_+\sqrt{n}\left\langle\sqrt{\hat{p}}-\sqrt{q}, v\right\rangle\right)d\bar{\Pi}(v)\right\}, \qquad (63)\\
\mathcal{L}_+ &= \max\left\{\int\exp\left(-2\delta_-^2 + 4\delta_-\sqrt{n}\left\langle\sqrt{\hat{p}}-\sqrt{q}, v\right\rangle\right)d\bar{\Pi}(v),\right. \\
&\qquad \left.\int\exp\left(-2\delta_-^2 + 4\delta_+\sqrt{n}\left\langle\sqrt{\hat{p}}-\sqrt{q}, v\right\rangle\right)d\bar{\Pi}(v)\right\}, \qquad (64)
\end{aligned}$$

where $\bar{\Pi}$ is the uniform distribution on the unit sphere $\{v : \|v\| = 1\}$. By (56), we have

$$\begin{aligned}
\mathcal{L}_- &= e^{-2\delta_+^2}\min\{F_{2\delta_-}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|), F_{2\delta_+}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|)\},\\
\mathcal{L}_+ &= e^{-2\delta_-^2}\max\{F_{2\delta_-}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|), F_{2\delta_+}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|)\}.
\end{aligned}$$

where $F_\delta(x)$ is defined in (58). Note that $4n\|\sqrt{\hat{p}}-\sqrt{q}\|^2 \rightsquigarrow \chi^2_{k-1}$ under the null and $4n\|\sqrt{\hat{p}}-\sqrt{q}\|^2 - \delta_2^2 \rightsquigarrow \chi^2_{k-1,\delta_1^2}$, with $\delta_1^2 = \delta_1(p)^2$ and $\delta_2^2 = \delta_2(p)^2$ defined in (47) and (48), under the alternative. Define $\phi = \mathbb{I}\{\mathcal{L} > 1\}$, $\phi_- = \mathbb{I}\{\mathcal{L}_- > 1\}$, $\phi_+ = \mathbb{I}\{\mathcal{L}_+ > 1\}$ and $\phi^* = \mathbb{I}\{\mathcal{L}^* > 1\}$. Then, we have the inequality $\phi_-\mathbb{I}_E \leq \phi\mathbb{I}_E \leq \phi_+\mathbb{I}_E$. For $q = p$, we have

$$\begin{aligned}
\mathbb{P}_q\phi &= \mathbb{P}_q\phi\mathbb{I}_E + \mathbb{P}_q\phi\mathbb{I}_{E^c}\\
&\geq \mathbb{P}_q\phi_-\mathbb{I}_E\\
&\geq \mathbb{P}_q(\mathcal{L}_- > 1) - \mathbb{P}_q(E^c)\\
&= \mathbb{P}_q\left(e^{-2\delta_+^2}\min\{F_{2\delta_-}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|), F_{2\delta_+}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|)\} > 1\right) - \mathbb{P}_q(E^c)\\
&\to \mathbb{P}\left(e^{-2\delta^2}F_{2\delta}\left(\sqrt{\chi^2_{k-1}}\right) > 1\right).
\end{aligned}$$

For the alternative, we have

$$\begin{aligned}
\mathbb{P}_p(1-\phi) &= \mathbb{P}_p(1-\phi)\mathbb{I}_E + \mathbb{P}_p(1-\phi)\mathbb{I}_{E^c} \\
&\leq \mathbb{P}_p(1-\phi_+)\mathbb{I}_E + \mathbb{P}_p(E^c) \\
&\leq \mathbb{P}_p(1-\phi_+) + \mathbb{P}_p(E^c) \\
&= \mathbb{P}_p\left(e^{-2\delta_-^2}\max\{F_{2\delta_-}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|), F_{2\delta_+}(2\sqrt{n}\|\sqrt{\hat{p}}-\sqrt{q}\|)\} \leq 1\right) + \mathbb{P}_p(E^c) \\
&\to \mathbb{P}\left(e^{-2\delta^2}F_{2\delta}\left(\sqrt{\chi^2_{k-1,\delta_1(p)^2}+\delta_2(p)^2}\right) \leq 1\right) \\
&\geq \inf_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}\left(e^{-2\delta^2}F_{2\delta}\left(\sqrt{\chi^2_{k-1,\delta_1^2}+\delta_2^2}\right) \leq 1\right).
\end{aligned}$$

By the fact that $F_\delta(x)$ is increasing on $x > 0$, we have

$$\begin{aligned}
R_n(k,\delta) &\geq (1+o(1))\left\{\mathbb{P}\left(e^{-2\delta^2}F_{2\delta}\left(\sqrt{\chi^2_{k-1}}\right) > 1\right) + \right. \\
&\qquad \left. \inf_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}\left(e^{-2\delta^2}F_{2\delta}\left(\sqrt{\chi^2_{k-1,\delta_1^2}+\delta_2^2}\right) \leq 1\right)\right\} \\
&\geq (1+o(1))\inf_{t>0}\left(\mathbb{P}\left(\chi^2_{k-1} > t\right) + \inf_{\{\delta_1,\delta_2:\delta_1^2+\delta_2^2=\delta^2\}} \mathbb{P}(\chi^2_{k-1,\delta_1^2}+\delta_2^2 \leq t)\right).
\end{aligned}$$

The proof is complete. ■

Proof [Proof of Theorem 17] It is implied by Condition M4 that $\sqrt{n}|\sqrt{r_g}-\sqrt{r_h}| \to \infty$ for any $g \neq h$. Moreover, for any $j \in \mathcal{C}_h$, $|\sqrt{q_j}-\sqrt{r_h}| = o(n^{-1/2})$. Under these assumptions, for any p such that $\|\sqrt{p}-\sqrt{q}\| = \frac{\delta}{\sqrt{n}}$, there exists a $\pi \in S_k$ that depends on p and $\|\sqrt{p_\pi}-\sqrt{q}\| = \ell(p,q) = \frac{\delta}{\sqrt{n}}(1+\epsilon_\theta)$. Moreover, $|\epsilon_\theta| = o(1)$ uniformly over all p that satisfies $\|\sqrt{p}-\sqrt{q}\| = \frac{\delta}{\sqrt{n}}$. Define

$$p' = \left(\sqrt{q} + \frac{1}{1+\epsilon_\theta}(\sqrt{p}-\sqrt{q})\right)^2. \tag{65}$$

Then, $\|\sqrt{p}-\sqrt{q}\| = \frac{\delta_\theta}{\sqrt{n}}$ and $\ell(p',q) = \frac{\delta}{\sqrt{n}}$, where $\delta_\theta = \frac{\delta}{1+\epsilon_\theta}$. We use the notation R to denote the operator $R : p \mapsto R(p)$ defined by (65). By the definition, a useful property is $\frac{\sqrt{R(p)}-\sqrt{q}}{\|\sqrt{R(p)}-\sqrt{q}\|} = \frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}$. Consider the set

$$\bar{\mathcal{P}}_\delta = \left\{R(p) : \|\sqrt{p}-\sqrt{q}\| = \frac{\delta}{\sqrt{n}}\right\}.$$

This definition immediately implies $\bar{\mathcal{P}}_\delta \subset \mathcal{P}_\delta$. Note that each element in $\bar{\mathcal{P}}_\delta$ can be represented as

$$R(p) = \left(\sqrt{q} + \frac{\delta_\theta}{\sqrt{n}}\frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}\right)^2.$$

Since there is a one-to-one relation between $\frac{\sqrt{p}-\sqrt{q}}{\|\sqrt{p}-\sqrt{q}\|}$ and a unit vector v, we can also write each element in $\bar{\mathcal{P}}_\delta$ as $\left(\sqrt{q}+\frac{\delta_v}{\sqrt{n}}v\right)^2$. Consider a uniform probability measure $\bar{\Pi}$ on $\{v : \|v\| = 1\}$. Then, by the same argument in the proof of Theorem 11,

$$R_n(k,\delta) \geq \inf_{0\leq\phi\leq 1}\left\{\mathbb{P}_q\phi + \int \mathbb{P}_{\left(\sqrt{q}+\frac{\delta_v}{\sqrt{n}}v\right)^2}(1-\phi)d\bar{\Pi}(v)\right\},$$

and the likelihood ratio is

$$\mathcal{L} = \int \frac{d\mathbb{P}_{\left(\sqrt{q}+\frac{\delta_v}{\sqrt{n}}v\right)^2}}{d\mathbb{P}_q}d\bar{\Pi}(v).$$

Using the same arguments in the proofs of Theorem 13 and Theorem 15, there exist δ_- and δ_+, with which we can define $\mathcal{L}_-$ and $\mathcal{L}_+$ as in (63) and (64) with the desired properties. Then, the same argument in the proof of Theorem 15 leads to the desired result. ■

Proof [Proof of Theorem 12] By studying the proof of Theorem 22, the only probabilistic argument in approximation is that $\max_{1\leq j\leq k} Z_j^2 \leq C_n$ in probability. Since this event is independent of θ, the in-probability argument can be made uniformly over $\theta \in \Theta_\delta$ and $\theta \in \Theta_0$. ■

Proof [Proof of Theorem 14] By Theorem 23, $T_g \geq T_f$ in probability. This implies that $\mathbb{P}_\theta\phi = \mathbb{P}_\theta(T_g > t^*)$ and $\mathbb{P}_\theta(1-\phi) = \mathbb{P}_\theta(T_g \leq t^*)$ under both null and alternative distributions. Then, by the same argument in the proof of Theorem 12, we obtain the desired conclusion. ■

Proof [Proofs of Theorem 16 and Theorem 18] Similar to the argument used in the proof of Theorem 12, the results directly follow the conclusions of Theorem 24 and Theorem 25. ■

Acknowledgments

The research of CG is supported in part by NSF grant DMS-1712957.

Journal of Machine Learning Research 19 (2018) 1-46 Submitted 12/17; Revised 05/18; Published 08/18

A Spectral Approach for the Design of Experiments: Design, Analysis and Algorithms

Bhavya Kailkhura kailkhura1@llnl.gov
Center for Applied Scientific Computing
Lawrence Livermore National Lab
Livermore, CA 94550, USA

Jayaraman J. Thiagarajan jjayaram@llnl.gov
Center for Applied Scientific Computing
Lawrence Livermore National Lab
Livermore, CA 94550, USA

Charvi Rastogi charvirastogi@iitb.ac.in
Department of EECS
Indian Institute of Technology
Bombay, MH 400076, India

Pramod K. Varshney varshney@syr.edu
Departments of EECS
Syracuse University
Syracuse, NY 13244, USA

Peer-Timo Bremer bremer5@llnl.gov
Center for Applied Scientific Computing
Lawrence Livermore National Lab
Livermore, CA 94550, USA

Editor: Animashree Anandkumar

Abstract

This paper proposes a new approach to construct high quality space-filling sample designs. First, we propose a novel technique to quantify the space-filling property and optimally trade-off uniformity and randomness in sample designs in arbitrary dimensions. Second, we connect the proposed metric (defined in the spatial domain) to the quality metric of the design performance (defined in the spectral domain). This connection serves as an analytic framework for evaluating the qualitative properties of space-filling designs in general. Using the theoretical insights provided by this spatial-spectral analysis, we derive the notion of optimal space-filling designs, which we refer to as space-filling spectral designs. Third, we propose an efficient estimator to evaluate the space-filling properties of sample designs in arbitrary dimensions and use it to develop an optimization framework for generating high quality space-filling designs. Finally, we carry out a detailed performance comparison on two different applications in varying dimensions: a) image reconstruction and b) surrogate modeling for several benchmark optimization functions and a physics simulation code for inertial confinement fusion (ICF). Our results clearly evidence the superiority of the proposed space-filling designs over existing approaches, particularly in high dimensions.

Keywords: design of experiments, space-filling, poisson-disk sampling, surrogate modeling, regression

1. Introduction

Exploratory analysis and inference in high dimensional parameter spaces is a ubiquitous problem in science and engineering. As a result, a wide-variety of machine learning tools and optimization techniques have been proposed to address this challenge. In its most generic formulation, one is interested in analyzing a high-dimensional function $f : \mathcal{D} \to \mathbb{R}$ defined on the d-dimensional domain $\mathcal{D}$. A typical approach for such an analysis is to first create an initial sampling $\mathcal{X} = \{\mathbf{x_i} \in \mathcal{D}\}_{i=1}^{N}$ of $\mathcal{D}$, evaluate f at all $\mathbf{x_i}$, and perform subsequent analysis and learning using only the resulting tuples $\{(\mathbf{x_i}, f(\mathbf{x_i}))\}_{i=1}^{N}$. Despite the widespread use of this approach, a critical question that still persists is: how should one obtain a high quality initial sampling $\mathcal{X}$ for which the data $f(\mathcal{X})$ is acquired or generated? This challenge is typically referred to as Design of Experiments (DoE) and solutions have been proposed as early as (Fisher, 1935) that optimized agricultural experiments. Subsequently, DoE has received significant attention from researchers in different fields (Garud et al., 2017). It is also an important *building block* for a wide variety of machine learning applications, such as, supervised machine learning, neural network training, image reconstruction, reinforcement learning, etc. (for a detailed discussion see Section 10). In several scenarios, it has been shown that success crucially depends on the quality of the initial sampling $\mathcal{X}$. Currently, a plethora of sampling solutions exist in the literature with a wide-range of assumptions and statistical guarantees; see (Garud et al., 2017; Owen, 2009) for a detailed review of related methods. Conceptually, most approaches aim to cover the sampling domain as uniformly as possible, in order to generate the so called *space-filling* experimental designs (Joseph, 2016)[1]. However, it is well know that uniformity alone does not necessarily lead to high performance. For example, optimal sphere packings lead to highly uniform designs, yet are well known to cause strong aliasing artifacts most easily perceived by the human visual system in many computer graphics applications. Instead, a common assumption is that a good design should balance uniformity and randomness[2]. Unfortunately, an exact definition for what should be considered a good space-filling design remains elusive.

Most common approaches use various scalar metrics to encapsulate different notions of ideal sampling properties. One popular metric is the discrepancy of an experimental design, defined as an appropriate ℓ_p norm of the ratio of points within all (hyper-rectangular) sub-volumes of $\mathcal{D}$ and the corresponding volume ratio. In other words, discrepancy quantifies the non-uniformity of a sample design. The most prominent examples of so called *discrepancy sequences* are Quasi-Monte Carlo (QMC) methods and their variants (Caflisch, 1998). In their classical form, discrepancy sequences are deterministic though extensions to incorporate randomess have been proposed, for example, using digital scrambling (Owen, 1995). Nevertheless, by optimizing for discrepancy these techniques focus almost exclusively on uniformity, and consequently even optimized QMC patterns can be quite structured and create

1. The term "space-filling" is used widely in the literature on design of experiments. However, in most cases, space-filling is meant in an intuitive sense and as a synonym for "evenly or uniformly spread". Later in this paper, we will provide a technical definition of space-filling property and what should be considered a good sample design.
2. In this paper, by randomness we mean that sample points are uniformly distributed over space. Here "uniform" is used in the sense that sample points follow a uniform probability distribution over the sampling region and that each location is equally likely to be selected as sample location, not in the sense that they are "evenly dispersed over the sampling region.

aliasing artifacts. Furthermore, even the fastest known strategies for evaluating popular discrepancy measures require $O(N^2d)$ operations making evaluation, let alone optimization, for discrepancy difficult even for moderate dimensions. Finally, for most discrepancy measures, the optimal achievable values are not known. This makes it difficult to determine whether a poorly performing sample design (e.g., in terms of generalization (or test error) in a regression application and reconstruction error in an image reconstruction application) is due to the insufficiency of the chosen discrepancy measure or due to ineffective optimization.

Another class of metrics to describe sample designs are based on geometric distances. These can be used directly by, for example, optimizing the maximin or minimax distance of a sample design (Schlomer et al., 2011) or indirectly by enforcing empty disk conditions. The latter is the basis for the so-called Poisson disk samples (Lagae and Dutr, 2008), which aim to generate random points such that no two samples can be closer than a given minimal distance r_{min}, i.e. enforcing an empty disc of radius r_{min} around each sample. Typically, Poisson-type samples are characterized by the *relative radius*, ρ, defined as the ratio of the minimum disk radius r_{min} and the maximum possible disk radius r_{max} for N samples to cover the sampling domain. Similar to the discrepancy sequences, maximin and minimax designs exclusively consider uniformity, are difficult to optimize for especially in higher dimensions, and often lead to very regular patterns. Poisson disk samples use ρ to trade off randomness (lower ρ values) and uniformity (higher ρ values). A popular recommendation in 2-d is to aim for $0.65 \leq \rho \leq 0.85$ as a good compromise. However, there does not exist any theoretical guidance for choosing ρ and hence, optimal values for higher dimensions are not known. As discussed in more detail in Section 2, there also exist a wide variety of techniques that combine different metrics and heuristics. For example, Latin Hypercube sampling (LHS) aims to spread the sample points uniformly by stratification, and one can potentially optimize the resulting design using maximin or minimax techniques (Jin et al., 2005).

In general, scalar metrics to evaluate the quality of a sample design tend not to be very descriptive. Especially in high dimensions different designs with, for example, the same ρ can exhibit widely different performance and for some discrepancy sequences the optimal designs converge to random samples in high dimensions (Morokoff and Caflisch, 1994; Wang and Sloan, 2008). Furthermore, one rarely knows the best achievable value of the metric, i.e. the lowest possible discrepancy, for a given problem which makes evaluating and comparing sampling designs difficult. Finally, most metrics are expensive to compute and not easily optimized. This makes it challenging in practice to create good designs in high dimensions and with large sample sizes.

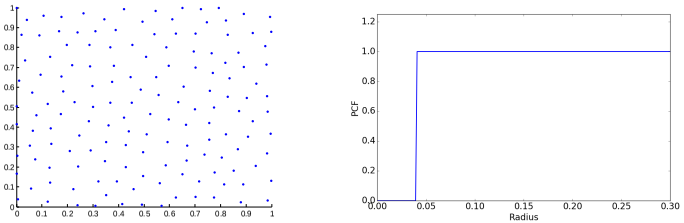
To alleviate this problem, we propose a new technique to quantify the space-filling property, which enables us to systematically trade-off uniformity and randomness, consequently producing better quality sampling designs. More specifically, we use tools from statistical mechanics to connect the qualitative performance (in the spectral domain) of a sampling pattern with its spatial properties characterized by the pair correlation function (PCF). The PCF measures the distribution of point samples as a function of distances, thus, providing a holistic view of the space-filling property (See Figure 1(b)). Furthermore, we establish the connection between the PCF and the power spectral density (PSD) via the $1-$D Hankel transform in arbitrary dimensions, thus providing a relation between the PCF and the quality metric of sampling quality to help subsequent design and analysis.

Using insights from the analysis of space-filling designs in the spectral domain, we provide design guidelines to systematically trade-off uniformity and randomness for a good sampling pattern. The analytical tractability of the PCF enables us to perform theoretical analysis in the spectral domain to derive the structure of optimal space-filling designs, referred to as *space-filling spectral design* in the rest of this paper. Next, we develop an edge corrected kernel density estimator based technique to measure the space-filling property via PCFs in arbitrary dimensions. In contrast to existing PCF estimation techniques, the proposed PCF estimator is both accurate and computationally efficient. Based on this estimator, we develop a systematic optimization framework and a novel algorithm to synthesize space-filling spectral designs. In particular, we propose to employ a weighted least-squares based gradient descent optimization, coupled with the proposed PCF estimator, to accurately match the optimal space-filling spectral design defined in terms of the PCF.

Note that there is a strong connection between the proposed space-filling spectral designs and coverage based designs such as Poisson Disk Sampling (PDS) (Gamito and Maddock, 2009). However, the major difference lies in the metric/criterion these techniques use to estimate and optimize the space-filling designs. Furthermore, existing works on PDS focus primarily on algorithmic issues, such as worst-case running times and numerical issues associated with providing high-quality implementations. However, different PDS methods often demonstrate widely different performances which raises the questions of how to evaluate the qualitative properties of different PDS patterns and how to define an optimal PDS pattern? We argue that, coverage (ρ) based metrics alone are insufficient for understanding the statistical aspects of PDS. This makes it difficult to generate high quality PDS patterns. As we will demonstrate below, existing PDS approaches largely ignore the randomness objective and instead concentrate exclusively on the coverage objective resulting in inferior sampling patterns compared to space-filling spectral designs, especially in high dimensions. Note that on the other hand, the proposed PCF based metric does not have these limitations and enables a comprehensive analysis of statistical properties of space-filling designs (including PDS), while producing higher quality sampling patterns compared to the state-of-the-art PDS approaches.

In (Kailkhura et al., 2016a), we use the PCF to understand the nature of PDS and provided theoretical bounds on the sample size of achievable PDS. Here we significantly extend our previous work and provide a more comprehensive analysis of the problem along with a novel space-filling spectral designs, an edge corrected PCF estimator, an optimization approach to synthesize the space-filling spectral designs and a detailed evaluation of the performance of the proposed sample design. The main contributions of this paper can be summarized as follows:

- We provide a novel technique to quantify the space-filling property of sample designs in arbitrary dimensions and systematically trade-off uniformity and randomness.
- We use tools from statistical mechanics to connect the qualitative performance (in the spectral domain) of a sample design with its spatial properties characterized by the PCF.
- We develop a computationally efficient edge corrected kernel density estimator based technique to estimate the space-filling property in arbitrary dimensions.



(a) (b)

Figure 1: A sample design that balances *randomness* and *uniformity*. (a) Point distribution, and (b) Pair correlation function.

- Using theoretical insights obtained via spectral analysis of point distributions, we provide design guidelines for optimal space-filling designs.
- We devise a systematic optimization framework and a gradient descent optimization algorithm to generate high quality space-filling designs.
- We demonstrate the superiority of proposed space-filling spectral samples compared to existing space-filling approaches through rigorous empirical studies on two different applications: *a*) image reconstruction and *b*) surrogate modeling on several benchmark optimization functions and an inertial confinement fusion (ICF) simulation code.

2. Related Work

In this section, we provide a brief overview of existing approaches for creating space-filling sampling patterns. Note that the prior art for this long-studied research area is too extensive to cover in detail, and hence we recommend interested readers to refer to (Garud et al., 2017; Owen, 2009) for a more comprehensive review.

2.1 Latin Hypercube Sampling

Monte-Carlo methods form an important class of techniques for space-filling sample design. However, it is well known that Monte-Carlo methods are characterized by high variance in the resulting sample distributions. Consequently, variance reduction methods are employed in practice to improve the performance of simple Monte Carlo techniques. One example is stratified sampling using the popular Latin Hypercube Sampling (LHS) (McKay, 1992; Packham, 2015). Since its inception, several variants of LHS have been proposed with the goal of achieving better space-filling properties, in addition to reducing variance. A notable improvement in this regard are techniques that achieve LHS space filling not only in one-dimensional projections, but also in higher dimensions. For example, Tang (Tang, 1993; Leary et al., 2003) introduced orthogonal-array-based Latin hypercube sampling to improve space-filling in higher dimensional subspaces. Furthermore, a variety of space-filling criteria

such as entropy, integrated mean square error, minimax and maximin distances, have been utilized for optimizing LHS (Jin et al., 2005). A particularly effective and widely adopted metric is the maximin distance criterion, which maximizes the minimal distance between points to avoid designs with points too close to one another (Morris and Mitchell, 1995). A detailed study on LHS and its variants can be found in (Koehler and Owen, 1996).

2.2 Quasi Monte Carlo Sampling

Following the success of Monte-Carlo methods, Quasi-Monte Carlo (QMC) sampling was introduced in (Halton, 1964) and since then has become the de facto solution in a wide-range of applications (Caflisch, 1998; Wang and Sloan, 2008). The core idea of QMC methods is to replace the random or pseudo-random samples in Monte-Carlo methods with well-chosen deterministic points. These deterministic points are chosen such that they are highly uniform, which can be quantified using the measure of discrepancy. Low-discrepancy sequences along with bounds on their discrepancy were introduced in the 1960's by Halton (Halton, 1964) and Sobol (Sobol, 1967), and are still in use today. However, despite their effectiveness, a critical limitation of QMC methods is that error bounds and statistical confidence bounds of the resulting designs cannot be obtained due to the deterministic nature of low-discrepancy sequences. In order to alleviate this challenge, randomized quasi-Monte Carlo (RQMC) sampling has been proposed (LEcuyer and Lemieux, 2005), and in many cases shown to be provably better than the classical QMC techniques (Owen and Tribble, 2005). This has motivated the development of other randomized quasi-Monte Carlo techniques, for example, methods based on digital scrambling (Owen, 1995).

2.3 Poisson Disk Sampling

While discrepancy-based designs have been popular among uncertainty quantification researchers, the computer graphics community has had long-standing success with coverage-based designs. In particular, Poisson disk sampling (PDS) is widely used in applications such as image/volume rendering. The authors in (Dippe and Wold, 1985; Cook, 1986) were the first to introduce PDS for turning regular aliasing patterns into featureless noise, which makes them perceptually less visible. Their work was inspired by the seminal work of Yellott *et.al.* (Yellott, 1983), who observed that the photo-receptors in the retina of monkeys and humans are distributed according to a Poisson disk distribution, thus explaining its effectiveness in imaging.

Due to the broad interest created by the initial work on PDS, a large number of approaches to generate Poisson disk distributions have been developed over the last decade (Gamito and Maddock, 2009; Ebeida et al., 2012, 2011; Ip et al., 2013; Bridson, 2007; Oztireli and Gross, 2012; Heck et al., 2013; Wei, 2008; Dunbar and Humphreys, 2006; Wei, 2010; Balzer et al., 2009; Geng et al., 2013; Yan and Wonka, 2012a, 2013; Ying et al., 2013b, 2014; Hou et al., 2013; Ying et al., 2013a; Guo et al., 2014; Wachtel et al., 2014; Xu et al., 2014; Ebeida et al., 2014; de Goes et al., 2012; Zhou et al., 2012). Most Poisson disk sample generation methods are based on dart throwing (Dippe and Wold, 1985; Cook, 1986), which attempts to generate as many darts as necessary to cover the sampling domain while not violating the Poisson disk criterion. Given the desired disk size r_{min} (or coverage ρ), dart throwing generates random samples and rejects or accepts each sample based on its distance

to the previously accepted samples. Despite its effectiveness, its primary shortcoming is the choice of termination condition, since the algorithm does not know whether or not the domain is fully covered. Hence, in practice, the algorithm has poor convergence, which in turn makes it computationally expensive. On the other hand, dart throwing is easy to implement and applicable to any sampling domain, even non-Euclidean. For example, Anirudh *et.al.* use a dart throwing technique to generate Poisson disk samples on the Grassmannian manifold of low-dimensional linear subspaces (Anirudh et al., 2017).

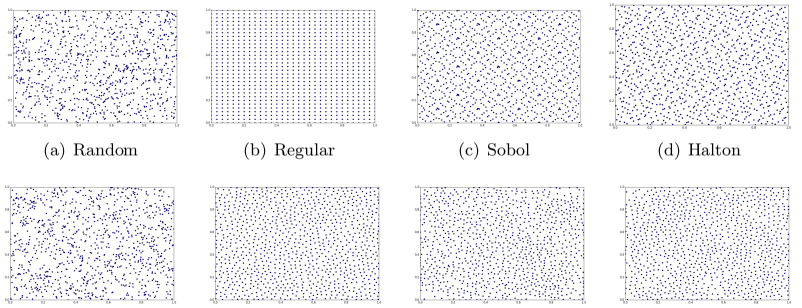
Reducing the computational complexity of PDS generation, particularly in low and moderate dimensions, has been the central focus of many existing efforts. To this end, approximate techniques that produce sample sets with characteristics similar to Poisson disk have been developed. Early examples (McCool and Fiume, 1992) are relatively simple and can be used for a wide range of sampling domains, but the gain in computational efficacy is limited. Other methods partition the space into grid cells in order to allow parallelization across the different cells and achieve linear time algorithms (Bridson, 2007). Another class of approaches, referred to as *tile-based* methods, have been developed for generating a large number of Poisson disk samples in 2-D. Broadly, these methods either start with a smaller set of samples, often obtained using other PDS techniques, and tile these samples (Wachtel et al., 2014), or alternatively use a regular tile structure for placing each sample (Ostromoukhov et al., 2004). With the aid of efficient data structures, these methods can generate a large number of samples efficiently. Unfortunately, these approximations can lead to low sample quality due to artifacts induced at tile boundaries and the inherent non-random nature of tilings. More recently, many researchers have explored the idea of partitioning the sampling space in order to avoid generating new samples that will be ultimately rejected by dart throwing. While some of these methods only work in $2-$D (Dunbar and Humphreys, 2006; Ebeida et al., 2011), the efficiency of other methods that are designed for higher dimensions (Gamito and Maddock, 2009; Ebeida et al., 2012) drops exponentially with increasing dimensions. Finally, relaxation methods that iteratively increase the Poisson disk radius of a sample set (McCool and Fiume, 1992) by re-positioning the samples also exist. However, these methods have the risk of converging to a regular pattern with tight packing unless randomness is explicitly enforced (Balzer et al., 2009; Schlomer et al., 2011).

A popular variant of PDS is the maximal PDS (MPDS) distribution, where the *maximality* constraint requires that the sample disks overlap, in the sense that they cover the whole domain leaving no room to insert an additional point. In practice, maximal PDS tends to outperform traditional PDS due to better coverage. However, algorithmically guaranteeing maximality requires expensive checks causing the resulting algorithms to be slow in moderate (2-5) and practically unfeasible in higher (7 and above) dimensions. Though strategies to alleviate this limitation have been proposed in (Ebeida et al., 2012), the inefficiency of MPDS algorithms in higher dimensions still persists. Interestingly, a common limitation of all existing MPDS approaches is that there is no direct control over the number of samples produced by the algorithm, which makes the use of these algorithms difficult in practice, since optimizing samples for a given sample budget is the most common approach.

As discussed in Section 1, the metrics used by the space-filling designs discussed above do not provide insights into how to systematically trade-off uniformity and randomness. Thereby, making the design and optimization of sampling pattern a cumbersome process. To alleviate this problem, we propose a novel metric for assessing the space-filling property

and connect the proposed metric (defined in the spatial domain) to the quality metric of design performance (defined in the spectral domain).

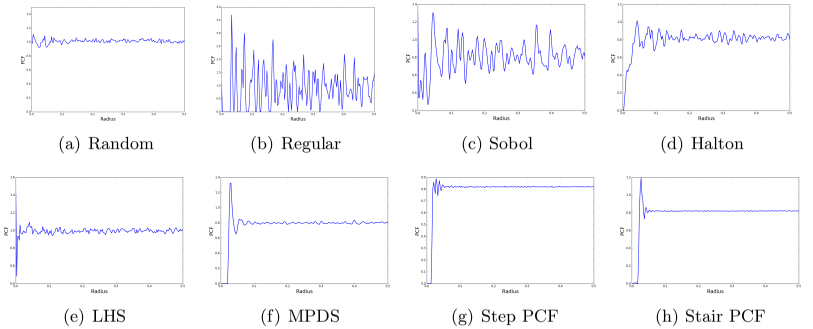
3. A Metric for Assessing Space-filling Property



(a) Random (b) Regular (c) Sobol (d) Halton

(e) LHS (f) MPDS (g) Step PCF (h) Stair PCF

Figure 2: Visualization of 2-d point distributions obtained using different sample design techniques. In all cases, the number of samples N was fixed at 1000.

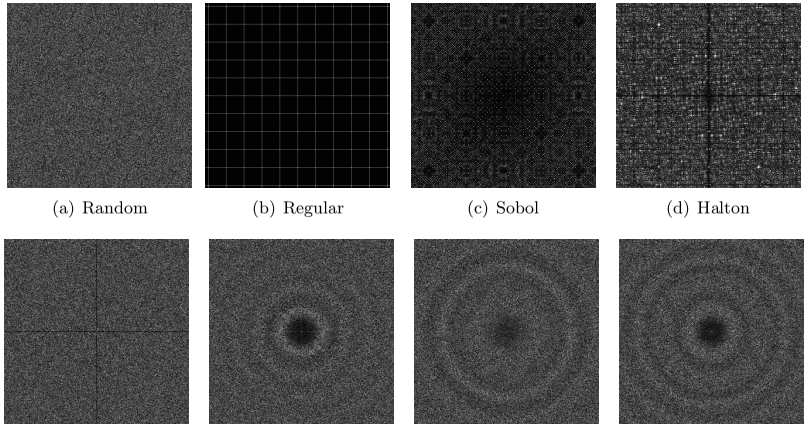


(a) Random (b) Regular (c) Sobol (d) Halton

(e) LHS (f) MPDS (g) Step PCF (h) Stair PCF

Figure 3: *Space-filling Metric*: Pair correlation functions, corresponding to the samples in Figure 2, characterize the coverage (and randomness) of point distributions obtained using different techniques.

In this section we first provide a definition of a space-filling design. Subsequently, we propose a metric to quantify space-filling properties of sample designs.



(a) Random (b) Regular (c) Sobol (d) Halton

(e) LHS (f) MPDS (g) Step PCF (h) Stair PCF

Figure 4: *Performance Quality Metric*: Power spectral density is used to characterize the effectiveness of sample designs, through the distribution of power in different frequencies.

3.1 Space-filling Designs

Without any prior knowledge of the function f of interest, a reasonable objective when creating $\mathcal{X}$ is that the samples should be random to provide an equal chance of finding features of interest, e.g., local minima in an optimization problem, anywhere in $\mathcal{D}$. However, to avoid sampling only parts of the parameter space, a second objective is to cover the space in $\mathcal{D}$ uniformly, in order to guarantee that all sufficiently large features are found. Therefore, a good space-filling design can be defined as follows:

Definition 1 *A space-filling design is a set of samples that are distributed according to a uniform probability distribution (Objective 1: Randomness) but no two samples are closer than a given minimum distance r_{min} (Objective 2: Coverage).*

Next, we describe the metric that we use to quantify the space-filling property of a sample design. The proposed metric is based on the spatial statistic, *pair correlation function* (PCF) and we will show that this metric is directly linked to the quality metric of design performance defined in the spectral domain.

3.2 Pair Correlation Function as a Space-filling Metric

In contrast to existing scalar space-filling metrics such as discrepancy, and coverage, the PCF characterizes the distribution of sample distances, thus providing a comprehensive description of the sample designs. A precise definition of the PCF can be given in terms

of the intensity λ and product density β of a point process (Illian et al., 2008; Oztireli and Gross, 2012).

Definition 2 *Let us denote the intensity of a point process $\mathcal{X}$ as $\lambda(\mathcal{X})$, which is the average number of points in an infinitesimal volume around $\mathcal{X}$. For isotropic point processes, this is a constant value. To define the product density β, let $\{B_i\}$ denote the set of infinitesimal spheres around the points, and $\{dV_i\}$ indicate the volume measures of B_i. Then, we have*[3] *$Pr(\mathbf{X_1} = \mathbf{x_1}, \cdots, \mathbf{X_N} = \mathbf{x_N}) = \beta(\mathbf{x_1}, \cdots, \mathbf{x_N}) dV_1 \cdots dV_N$ which represents the probability of having points $\mathbf{x_i}$ in the infinitesimal spheres $\{B_i\}$. In the isotropic case, for a pair of points, β depends only on the distance between the points, hence one can write $\beta(\mathbf{x_i}, \mathbf{x_j}) = \beta(||\mathbf{x_i} - \mathbf{x_j}||) = \beta(r)$ and $Pr(r) = \beta(r) dV_i dV_j$. The PCF is then defined as*

$$G(r) = \frac{\beta}{\lambda^2}. \tag{1}$$

Note that the PCF characterizes spatial properties of a sample design. However, in several cases, it is easier to link the quality metric of a sample design to its spectral properties. Therefore, we establish a connection between the spatial property of a sample design defined in PCF space to its spectral properties.

3.3 Connecting Spatial Properties and Spectral Properties of Space-filling Designs

Fourier analysis is a standard approach for understanding the qualitative properties of sampling patterns. Hence, we propose to analyze the spectral properties of sample designs, using tools such as the power spectral density, in order to assess their quality. For isotropic samples, a quality metric of interest is the radially-averaged power spectral density, which describes how the signal power is distributed over different frequencies.

Definition 3 *For a finite set of N points, $\{\mathbf{x}_j\}_{j=1}^N$, in a region with unit volume, the power spectral density of the sampling function $\sum_{j=1}^N \delta(\mathbf{x} - \mathbf{x}_j)$ is formally defined as*

$$P(\mathbf{k}) = \frac{1}{N}|S(\mathbf{k})|^2 = \frac{1}{N}\sum_{j,\ell} e^{-2\pi i \mathbf{k}.(\mathbf{x}_\ell - \mathbf{x}_j)}, \tag{2}$$

where $|.|$ denotes the l^2-norm and $S(\mathbf{k})$ denotes the Fourier transform of the sampling function.

The radially-averaged power spectral density (PSD) is denoted using $P(k)$. Next, we show that the connection between spectral properties of a d-dimensional isotropic sample design and its corresponding pair correlation function can be obtained via the d-dimensional Fourier transform or more efficiently using the 1-d Hankel transform.

3. We denote a realization of random variables $\mathbf{X}_1, \cdots, \mathbf{X}_N$ by $\mathbf{x}_1, \cdots, \mathbf{x}_N$.

Proposition 4 *For an isotropic sample design with N points, $\{\mathbf{x}_j\}_{j=1}^N$, in a d-dimensional region with unit volume, the pair correlation function $G(r)$ and radially averaged power spectral density $P(k)$ are related as follows:*

$$G(r) = 1 + \frac{V}{2\pi N} H\left[P(k) - 1\right] \tag{3}$$

where V is the volume of the sampling region and $H[.]$ denotes the 1-d Hankel transform, defined as

$$H(f(k)) = \int_0^\infty k J_0(kr) f(k) dk,$$

with $J_0(.)$ denoting the Bessel function of order zero.

Proof Note that PSD and PCF of a sample design are related via the d-dimensional Fourier transform as follows:

$$\begin{aligned} P(\mathbf{k}) &= 1 + \frac{N}{V} F\left(G(\mathbf{r}) - 1\right) \\ &= 1 + \frac{N}{V} \int_{\mathbb{R}^d} \left(G(\mathbf{r}) - 1\right) \exp(-i\mathbf{k}.\mathbf{r}) d\mathbf{r}. \end{aligned}$$

It can be shown that, for radially symmetric or isotropic functions, the above relationship simplifies to

$$P(k) = 1 + 2\pi \frac{N}{V} H\left[G(r) - 1\right].$$

Next, using the inverse property of the Hankel transform, i.e.,

$$H_0^{-1}(f(r)) = \int_0^\infty r J_0(kr) f(r) dr,$$

we have

$$G(r) = 1 + \frac{V}{2\pi N} H\left[P(k) - 1\right]. \tag{4}$$

■

Proposition 4 is important as it enables us to qualitatively understand space-filling designs by first mapping them into the PCF space constructed based on spatial distances between points and, then, evaluating and understanding spectral properties of sample designs.

In Figure 3, we show the PCF[4] of some commonly used 2-d sample designs ($N = 1000$) illustrated in Figure 2. As can be observed, both regular grid samples and QMC sequences have significant oscillations in their PCFs, which can be attributed to their structured nature. Regular grid sample design demonstrates a large disk radius r_{min} ($G(r) = 0$ for $0 \leq r \leq r_{min}$) as every sample is at least r_{min} apart from the rest of the samples, which in turn implies a better coverage. However, in practice, they perform poorly (e.g., in terms of generalization (or test error) in regression application and reconstruction error in image reconstruction application) compared to randomized sample designs and this can

4. Note that for non-isotropic sample designs, d-dimensional PCF (Illian et al., 2008) can be more descriptive.

be understood by studying their spectral properties. In contrast, random sample (Monte-Carlo) designs have a constant PCF with nearly no oscillations, since point samples are uncorrelated, thus, $P(r) = \lambda dx \lambda dy$ and theoretically have $G(r) = 1$, $\forall r$. Furthermore, the LHS design has a similar PCF as random designs with the exception of a small, yet non-zero, r_{min}.

Other variants of PDS like MPDS, Step PCF and Stair PCF designs attempt to trade-off between coverage ($G(r) = 0$ for $0 \leq r \leq r_{min}$) and randomness $G(r) = 1$, for $r > r_{min}$. Note that the Step and the Stair PCF methods are space-filling spectral designs proposed later in this paper. However, upon a careful comparison, it can be seen that MPDS has a larger peak and more oscillations in its PCF compared to the proposed designs. In fact, our empirical studies show that the amount of oscillations in the PCF of the MPDS design significantly increases with dimensions.

Next, in Figure 4, we show the corresponding PSDs of the different sample designs. It can be seen that, oscillations in PCF directly correspond to oscillations in PSDs. For example, the oscillatory behavior of the PCF for regular and QMC sequences cause a non-uniform distribution of power in their corresponding PSDs. Furthermore, the larger peak height in the PCF of MPDS implies that a large amount of power is concentrated in a small frequency band instead of power being distributed over all frequencies. In Section 5, we will analyze the effect of the shape of PCF on the performance of a sample design in detail.

It is important to note that, not every PCF (or PSD) is physically realizable by a sample design. In fact, there are two necessary mathematical conditions [5] that a sample design must satisfy to be realizable.

Definition 5 (Realizability) *A PCF can be defined to be potentially realizable through a sample design, if it satisfies the following conditions:*

- *its PCF must be non-negative, i.e., $G(r) \geq 0$, $\forall r$, and*
- *its corresponding PSD must be non-negative, i.e., $P(k) \geq 0$, $\forall k$.*

As both the PSD and the PCF characteristics are strongly tied to each other (as shown in Proposition 4), these two conditions limit the space of realizable space-filling spectral designs. The results from this section will serve as tools for qualitatively understanding and, thus, designing optimal space-filling spectral designs in the following sections.

4. Space-filling Spectral Designs

In this section, we first formalize desired characteristics of a good space-filling design, as given in Definition 1. Next, we will describe the proposed framework for creating space-filling spectral designs.

Definition 6 *A set $\mathcal{X}$ of N random samples $\{\mathbf{X_i}\}_{i=1}^N$ in a sampling domain $\mathcal{D}$ can be characterized as a space-filling design, if $\mathcal{X} = \{\mathbf{X_i} = \mathbf{x_i} \in \mathcal{D};\ i = 1, \cdots N\}$ satisfy the following two objectives:*

5. Whether or not these two conditions are not only necessary but also sufficient is still an open question (however, no counterexamples are known).

- $\forall \mathbf{X_i} \in \mathcal{X},\ \forall \triangle\mathcal{D} \subseteq \mathcal{D}\ :\ Pr(\mathbf{X_i} = \mathbf{x_i} \in \triangle\mathcal{D}) = \int_{\triangle\mathcal{D}} \mathbf{dX}$
- $\forall \mathbf{x_i}, \mathbf{x_j} \in \mathcal{X}\ :\ ||\mathbf{x_i} - \mathbf{x_j}|| \geq r_{min}$

where r_{min} is referred to as the coverage radius.

In the above definition, the first objective states that the probability of a random sample $\mathbf{X_i} \in \mathcal{X}$ falling inside a subset $\triangle\mathcal{D}$ of $\mathcal{D}$ is equal to the hyper-volume of $\triangle\mathcal{D}$ (uniform distribution). The second condition enforces the minimum distance constraint between point sample pairs for improving coverage.

A Poisson design enforces the first condition alone, in which case the number of samples that fall inside any subset $\triangle\mathcal{D} \subseteq \mathcal{D}$ obeys a discrete Poisson distribution. Though easier to implement, Poisson sampling often produces distributions where the samples are grouped into clusters and leaves holes in possibly the regions of interest. In other words, this increases the risk of missing important features, when only the samples are used for analysis. Consequently, a sample design that distributes random samples in a uniform manner across $\mathcal{D}$ is preferred, so that clustering patterns are not observed. The coverage condition explicitly eliminates the clustering behavior by preventing samples from being closer than $r_{\min}$. A space-filling design can be defined conveniently in the PCF domain and we refer to this as the space-filling spectral design, due to its direct connection to the spectral domain properties.

4.1 Defining a Space-filling Spectral Design in Spatial Domain

For Poisson design, point locations are not correlated and, therefore, $P(r) = \lambda dx \lambda dy$. This implies that for Poisson designs $G(r) = 1$. Similarly, for space-filling designs, due to the minimum distance constraint between the point sample pairs, we do not have any point samples in the region $0 \leq r < r_{\min}$. Consequently, space-filling spectral designs are defined as a step pair correlation function in the spatial domain (*Step PCF*).

Proposition 7 *Given the desired coverage radius r_{min}, a space-filling spectral design is defined in the spatial domain as*

$$G(r - r_{min}) = \begin{cases} 0 & \text{if } r < r_{min} \\ 1 & \text{if } r \geq r_{min}. \end{cases}$$

As a consequence of Proposition 4, space-filling spectral designs can equivalently be defined in the spectral domain.

4.2 Defining a Space-filling Spectral Design in Spectral Domain

We derive the power spectral density of the space-filling spectral design using the connection established in Section 3. Following our earlier notation, we denote the d-dimensional power spectral density by $P(\mathbf{k})$ and d-dimensional PCF by $G(\mathbf{r})$.

Proposition 8 *(Kailkhura et al., 2016a) Given the desired coverage radius r_{min}, a d-dimensional space-filling spectral design $\mathcal{X}$, with N sample points in a sampling domain $\mathcal{D}$ of volume V, can be defined in the PSD domain as*

$$P(k) = 1 - \frac{N}{V}\left(\frac{2\pi r_{min}}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(k r_{min})$$

where $J_{\frac{d}{2}}(.)$ is the Bessel function of order $d/2$.

Proof We know that,

$$P(\mathbf{k}) = 1 + \frac{N}{V} F\left(G(\mathbf{r}) - 1\right), \tag{5}$$

$$= 1 + \frac{N}{V}\int_{\mathbb{R}^d} (G(\mathbf{r}) - 1)\exp(-i\mathbf{k}.\mathbf{r}) d\mathbf{r}, \tag{6}$$

where $F(.)$ denotes the d-dimensional Fourier transform. Note that for the radially symmetric or isotropic functions, i.e., $G(r)$ where $r = ||\mathbf{r}||$, the above relationship simplifies to

$$P(k) = 1 + \frac{N}{V}(2\pi)^{\frac{d}{2}} k^{1-\frac{d}{2}} H_{\frac{d}{2}-1}\left(r^{\frac{d}{2}-1}(G(r) - 1)\right), \tag{7}$$

where

$$H_v(f(r))(k) = \int_0^\infty r J_v(kr) f(r) dr$$

is the 1−d Hankel transform of order v with J being the Bessel function. To derive the PSD of a step function, we first evaluate the Hankel transform of $f(r) = r^{\frac{d}{2}-1}(G(r) - 1)$ where $G(r)$ is a step function.

$$\begin{aligned} H_{\frac{d}{2}-1}\left(r^{\frac{d}{2}-1}(G(r) - 1)\right) &= \int_0^\infty r^{\frac{d}{2}} J_{\frac{d}{2}-1}(kr)\left(G(r) - 1\right) dr \\ &= -\int_0^{r_{\min}} r^{\frac{d}{2}} J_{\frac{d}{2}-1}(kr) dr \\ &= -\frac{r_{\min}^{\frac{d}{2}}}{k} J_{\frac{d}{2}}(k r_{\min}) \end{aligned}$$

Using this expression in (7), we obtain

$$P(k) = 1 - \frac{N}{V}\left(\frac{2\pi r_{\min}}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(k r_{\min}). \tag{8}$$

■

This proposition connects the spatial properties of a space-filling spectral design, defined via the PCF, to its spectral properties. The motivation for this is the fact that in several cases, it is easier to link the qualitative performance of a sample design to its spectral properties. In the next section, we will develop the relation between spectral properties and an qualitatives measure of the performance, which in turn provides us guidelines for designing better space-filling spectral sampling patterns.

5. Qualitative Analysis of Space-filling Spectral Designs

In this section, we derive insights regarding the qualitative performance of space-filling spectral designs. To this end, we analyze the impact of the shape of the PCF on the reconstruction performance. Further, for a tractable analysis, we consider the task of recovering the class of periodic functions using space-filling spectral designs and analyze the reconstruction error as a function of their spectral properties. The analysis presented in this section will clarify how the shape of the PCF of a sample design directly impacts its reconstruction performance.

5.1 Analysis of Reconstruction Error for Periodic Functions

Let us denote the Fourier transform of the sample design $\mathcal{X}$ by $\mathcal{S}$. The function to be sampled and its corresponding Fourier representation are denoted by $\mathcal{I}$ and $\hat{\mathcal{I}}(k)$ respectively. Now, the spectrum of the sampled function is given by $\hat{\mathcal{I}}_s(k) = \mathcal{S} * \hat{\mathcal{I}}(k)$. Note that a sampling pattern with a finite number of points is comprised of two components, a DC peak at the origin and a noisy remainder $\bar{\mathcal{S}}$. Thus, equivalently, we have $\hat{\mathcal{I}}_s(k) = \{N\delta(k) + \bar{\mathcal{S}}\} * \hat{\mathcal{I}}(k)$. The error introduced in the process of function reconstruction is the difference between the reconstructed and the original functions:

$$\mathcal{E}(k) = |\hat{\mathcal{I}}_s(k)/N - \mathcal{I}(k)|^2 = |\bar{\mathcal{S}} * \hat{\mathcal{I}}(k)/N|^2$$

where we have divided the R.H.S. by N to normalize the energy of $\mathcal{I}_s$. For error analysis, we focus on the low frequency content of the error term, since the high frequency components are removed during the reconstruction process.

Denoting the power spectrum without the DC component by $\bar{\mathcal{P}}(k)$, for a constant function the error simplifies to

$$\mathcal{E}(k) \propto |\bar{\mathcal{S}}(k)|^2 \propto \bar{\mathcal{P}}(k). \tag{9}$$

This, as stated above, allows for the characterization of the error in terms of the spectral properties of the sampling pattern used.

Next, we consider an important class of functions, the family of periodic functions, for further analysis. All periodic functions with a finite period can be expressed as a Fourier series[6], which is a summation of sine and cosine terms

$$\mathcal{I}(x) = a_0 + \sum_{m=1}^{M} a_m cos(2\pi m x) + \sum_{m=1}^{M} b_m sin(2\pi m x).$$

The Fourier transform of this function is equivalently a summation of pulses:

$$\hat{\mathcal{I}}(k) = a_0\delta(k) + \sum_{m=1}^{M} a_m \left(\frac{1}{2}(\delta(k-m) + \delta(k+m)\right) + \sum_{m=1}^{M} b_m \left(\frac{1}{2}(\delta(k+m) - \delta(k-m))\right).$$

6. In the subsequent analysis, the number of terms in the Fourier series, M, is an arbitrary value which can be replaced by infinity for non-differentiable/discontinuous functions. Note that the Fourier series of periodic functions that are smooth (no discontinuity and no sharp corners) is finite.

Making substitutions, $a_m + b_m = A_m$, $a_m - b_m = B_m$, we obtain

$$\mathcal{E}(k) = \frac{1}{4N^2}\left|4a_0\bar{\mathcal{S}}(k) + \sum_{m=1}^{M}\left(A_m\bar{\mathcal{S}}(k+m) + B_m\bar{\mathcal{S}}(k-m)\right)\right|^2.$$

The reconstruction error can then be upper bounded as follows:

$$\mathcal{E}(k) \leq \frac{1}{4N}\left[4a_0^2\bar{\mathcal{P}}(k) + \sum_{m=1}^{M}\left(A_m^2\bar{\mathcal{P}}(k+m) + B_m^2\bar{\mathcal{P}}(k-m)\right)\right]. \tag{10}$$

In the case of a single sinusoidal function, $cos(2\pi f x)$, using triangle inequality, this becomes (Heck et al., 2013)

$$\mathcal{E}(k) \leq \frac{1}{4N}\left[\bar{\mathcal{P}}(k+f) + \bar{\mathcal{P}}(k-f) + 2\sqrt{\bar{\mathcal{P}}(k+f)\bar{\mathcal{P}}(k-f)}\right]. \tag{11}$$

Even though this is only an upper bound and the theoretic analysis is restricted to periodic functions, we have empirically found that it accurately predicts the characteristics of the sampling error for a broad range of complex functions and provides useful guidelines (more details are provided in Section 9).

The above analysis implies that to assess the quality of the sample designs, one can analyze their spectral behavior. More specifically, the above analysis suggests that to minimize the reconstruction error (Eq. (10) and (11)): (a) the power spectra of the sample design should be close to zero, and (b) for errors to be broadband white noise (uniform over frequencies), the power spectra should be a constant. Note that in several applications, e.g., image reconstruction, most relevant information is predominantly at low frequencies. In such scenarios, this naturally leads to the following criteria for sample designs: (a) the spectrum should be close to zero for low frequencies which indicates the range of frequencies that can be represented with almost no error, (b) the spectrum should be a constant for high frequencies or contain minimal amount of oscillations in the power spectrum. However, as we will see next, there exist a trade-off between low frequency power and high frequency oscillations in power spectra.

5.2 Effect of PCF Characteristics on Sampling Performance

Based on the two criteria discussed above, we assess the effect of the shape of the PCF on the quality of space-filling designs in the spectral domain. Note that PCFs of the samples constructed in practice (Figure 2) often demonstrates the following characteristics: (a) presence of a zero-region characterized by r_{min}, (b) a large peak around r_{min}, and (c) damped oscillations. To model and analyze these characteristics, we consider the following parametric PCF family[7]

$$\begin{aligned} G(r; r_{\min}, \delta, a, c) &= G(r - r_{\min}) + (a-1)\left(G(r - r_{\min}) - G(r - r_{\min} - \delta)\right) \\ &+ \frac{a-1}{4r}\exp(-r/2)\sin(c \times r - c)G(r - r_{\min}) \end{aligned} \tag{12}$$

7. Note that there exist a broad range of parametric space-filling spectral designs. However, finding PCFs that are realizable is a nontrivial problem because the space of functions that obey the realizability conditions is not easy to parametrize.

where $G(r - r_{\min})$ is the Step function, peak width $\delta \geq 0$ and the peak height $a \geq 1$ and last term in (12) corresponds to damped oscillations. This family is a generalization of Step PCF, with additional parameterization of peak height and oscillations in the PCF.

5.2.1 Effect of Peak Height on Spectral Properties

In order to study the impact of increasing peak height in the PCF on the PSD characteristics, we conduct an empirical study. We compute the PSD of a sample design with the following parameters: $N = 195, r_{\min} = 0.02, \delta = 0.005$. Note that we vary the PCF peak height a, which actually reflects the behavior of existing coverage based PDS algorithms. As shown in Figure 5(a), increasing a results in both significantly *higher* low frequency power and *larger* high frequency oscillations. As expected, the PSD of the Step PCF (or $a = 1$) performs the best, i.e., the spectrum is close to zero for low frequencies and constant for high frequencies.

5.2.2 Effect of Disk Radius on Spectral Properties

Next, we study the importance of choosing an appropriate $r_{\min}$ (or coverage ρ) while generating sample distributions. In Figure 5(b), we show the PSD for $N = 195$ and $a = 1$, with varying disk radius values $r_{\min}$. For a fixed sample budget, as we increase the radius, we observe two contrasting changes in the PSD: (i) the spectrum tends to be close to zero at low frequencies and (ii) an increase in oscillations for high frequencies. Consequently, there is a trade-off between low frequency power and high frequency oscillations in power spectra which can be controlled by varying $r_{\min}$. However, the increase in oscillations are less significant compared to the gain in the zero-region. Furthermore, in several applications, low frequency content is more informative, and hence one may still attempt to maximize r_{min} or coverage.

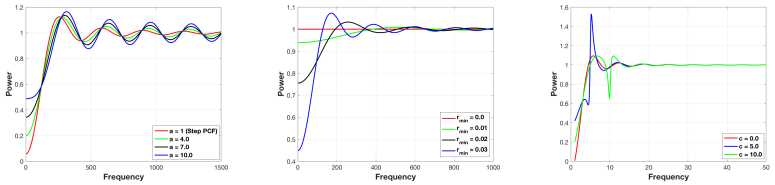
5.2.3 Effect of Oscillations on Spectral Properties

Finally, we study the effect of oscillations in the PCF on the power distribution in the spectral domain. In Figure 5(c), we plot the PSD for $a = 1$ with varying amounts of oscillations controlled via the parameter c. It can be seen that introducing oscillations in the PCF results in significantly *higher* low frequency power and *larger* high frequency oscillations. As expected, the PSD of the Step PCF (or $c = 0$) behaves the best.

In summary, the discussion in this section suggests that the PCF of an ideal space-filling spectral design should have the following three properties: (a) large r_{min}, (b) small peak height, and (c) low oscillations. Since, the Step PCF satisfies these three properties, it is expected to be a good space-filling spectral design. Next, we consider the problem of optimizing the parameter of the Step PCF design, i.e. r_{min}.

6. Optimization of Step PCF based Space-filling Spectral Designs

The proposed space-filling metric enjoys mathematical tractability and is supported by theoretical results as defined in Section 4. This enables us to obtain new insights for optimizing Step PCF based space-filling spectral designs. In particular, (a) For a fixed $r_{\min}$, we obtain the maximum number of point samples in any arbitrary dimension d, (b) For a



(a) (b) (c)

Figure 5: (a) Effect of peak height in the PCF on power spectra, (b) Effect of disk radius in the PCF on power spectra, (c) Effect of oscillations in the PCF on power spectra.

fixed sampling budget N, we derive the maximum achievable $r_{\min}$ in arbitrary dimension d.

6.1 Case 1: Fixed $r_{\min}$

The problem of finding the maximum number of point samples in a Step PCF based space-filling spectral design with a given disk radius $r_{\min}$ can be formalized as follows:

$$\begin{aligned} \text{maximize} \quad & N \\ \text{subject to} \quad & P(k) \geq 0, \ \forall k \\ & G(r - r_{\min}) \geq 0, \ \forall r, \end{aligned} \tag{13}$$

where $P(k) = 1 - \frac{N}{V}\left(\frac{2\pi r_{\min}}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(k r_{\min})$. Note that a space-filling spectral design has to satisfy realizability constraints as defined in Definition 5.

Proposition 9 *(Kailkhura et al., 2016a) For a fixed disk radius r_{min}, the maximum number of point samples possible for a realizable Step PCF based space-filling spectral design in the sampling region with volume V can be approximated as*

$$N \approx \frac{V\Gamma\left(\frac{d}{2}+1\right)}{\pi^{\frac{d}{2}} r_{min}^{d}}.$$

Proof Using the definition of the Step PCF function, the constraint $G(r - r_{\min})$ is trivially satisfied. Note that the constraint $P(k) \geq 0, \ \forall k$ is equivalent to $\min_k P(k) \geq 0$. In other

words,

$$\begin{aligned}
&\min_k 1-\rho\left(\frac{2\pi r_{\min}}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_{\min}) \geq 0 \\
\Leftrightarrow\ &\max_k \rho\left(\frac{2\pi r_{\min}}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_{\min}) \leq 1 \\
\Leftrightarrow\ &\rho\,(2\pi)^{\frac{d}{2}}\, r_{\min}^d \max_k \left(\frac{J_{\frac{d}{2}}(kr_{\min})}{(kr_{\min})^{\frac{d}{2}}}\right) \leq 1 \\
\Leftrightarrow\ &\rho\,(2\pi)^{\frac{d}{2}}\, r_{\min}^d \frac{1}{2^{\frac{d}{2}}\Gamma\left(\frac{d}{2}+1\right)} \lesssim 1 \qquad (14)\\
\Leftrightarrow\ &N \lesssim \frac{V\Gamma\left(\frac{d}{2}+1\right)}{(\pi)^{\frac{d}{2}}\, r_{\min}^d}
\end{aligned}$$

where, in (14) we have used the fact that $J_v(x) \approx (x/2)^v/\Gamma(v+1)$ and $\rho = N/V$. ■

Note that for the 2-dimensional case, we have $\frac{J_1(kr_{\min})}{kr_{\min}} = \text{jinc}(kr_{\min})$ where jinc(.) is the sombrero function (sometimes called besinc function or jinc function). Now using the fact that jinc(x) has the maximum value equal to $1/2$, for a fixed disk radius $r_{\min}$, the maximum number of point samples possible in a 2-d Step PCF based space-filling spectral design is given by

$$N = V/\pi(r_{\min})^2,$$

which again corroborates our bound in Proposition 9.

6.2 Case 2: Fixed N

Alternately, we can also derive the bound for the disk radius of Step PCF with a fixed sampling budget N as follows:

$$\begin{aligned}
&\text{maximize} && r_{\min} \\
&\text{subject to} && P(k) \geq 0,\ \forall k \\
& && G(r - r_{\min}) \geq 0,\ \forall r
\end{aligned} \qquad (15)$$

Proposition 10 *(Kailkhura et al., 2016a) For a fixed sampling budget N, the maximum possible disk radius r_{min} for a realizable Step PCF based space-filling spectral design in the sampling region with volume V can be approximated as*

$$r_{min} \approx \sqrt[d]{\frac{V\Gamma\left(\frac{d}{2}+1\right)}{\pi^{\frac{d}{2}}N}}.$$

Proof The proof is similar to the one in Proposition 9. ■

6.3 Relative Radius of Step PCF

As mentioned before, the current literature characterizes coverage by the fraction ρ of the maximum possible radius r_{max} for N samples to cover the sampling domain, such that $r_{min} = \rho r_{max}$. The maximum possible disk radius is achieved by the deterministic hexagonal lattice (Schreiber, 1943) and can be approximated in a d dimensional sampling region as $r_{max} \approx \sqrt[d]{\frac{A_d}{C_d N}}$. Here, A_d is the hypervolume of the sampling domain and $C_d = V_d/r^d$ with V_d being the hypervolume of a hypersphere with radius r. Note that a uniformly distributed point set can have a relative radius of 0, and the relative radius of a hexagonal lattice equals 1 (in 2-d). Next, we derive a closed-form expression for the relative radius of Step PCF based design.

Proposition 11 *For a fixed sampling budget N, the maximum relative radius ρ for Step PCF based space-filling spectral design in the sampling region with volume V is given by $\rho = \frac{1}{2\sqrt[d]{\eta_d}}$ where η_d is maximal density of a sphere packing in d-dimensions.*

Proof Let us denote by $r_{max} = \arg\min_r \eta_d$, then, the maximal density of a sphere packing with N samples in d-dimensions is given by

$$\eta_d = \frac{N\pi^{d/2}}{\Gamma(1+\frac{d}{2})}\frac{r_{max}^d}{V} \qquad (16)$$

$$\Leftrightarrow\ \eta_d = \left(\frac{r_{max}}{r_{min}}\right)^d \qquad (17)$$

$$\Leftrightarrow\ \rho = \frac{1}{2\sqrt[d]{\eta_d}} \qquad (18)$$

where equality in (17) uses Proposition (10). ■

For $d = 2$ and 3, the relative radius simplifies to:

$$\rho = 0.5\sqrt[2]{\frac{\pi\sqrt{3}}{6}},\ \textbf{for } d = 2, \text{ and}$$

$$\rho = 0.5\sqrt[3]{\frac{\pi\sqrt{2}}{6}},\ \textbf{for } d = 3.$$

Note that finding the maximal density of a sphere packing for an arbitrary high dimension (except in $d = 2, 3$ and recently in $8, 24$ (viazovska, 2017; Cohn et al., 2017)) is an open problem. Note that best known packings are often lattices, thus, we use the best known lattices to be an approximation of r_{max} in our analysis[8].

In Figure 6, we plot the relative radius $\rho = r_{min}/r_{max}$ of Step PCF for different dimensions d. It is interesting to notice that the relative radius of Step PCF based designs

8. We use relative radius as a metric only for analysis and not for design optimization.

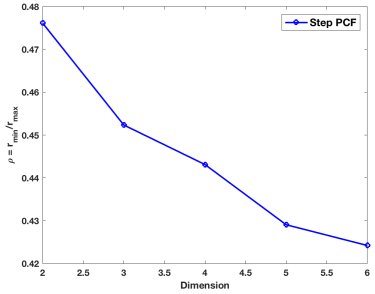


Figure 6: Relative radius $\rho = r_{min}/r_{max}$ of Step PCF based space-filling spectral design for different dimensions d.

increases as the dimension d increases, i.e., Step PCF based designs approach a more regular pattern. Further, note that for a fixed sampling budget both r_{min} and r_{max} increase as the number of dimensions increases. The Step PCF based designs maintain randomness by keeping the PCF flat, but this comes at a cost: the disk radius r_{min} of these patterns is very small (as can be seen from Figure 6). For several applications, covering the space better (by trading-off randomness) is more important. In the next section, we will propose a new class of space-filling spectral designs that can achieve a much higher r_{min} at the small cost of compromising randomness by introducing a single peak into an otherwise flat PCF.

7. Space-filling Spectral Designs with Improved Coverage

To improve the coverage of Step PCF base space-filling spectral design, in this section, we propose a novel space-filling spectral design which systematically trades-off randomness with coverage of the resulting samples. Note that the randomness property can be relaxed either by increasing the peak height of the PCF, or by increasing the amounts oscillations in the PCF (as discussed in Section 5.2). For simplicity[9], we adopt the former strategy and use only the peak height parameter. More specifically, as an alternative to Step PCF, we design the following generalization which we refer as the Stair PCF design.

7.1 Stair PCF based Space-filling Spectral Design

Now, we define the proposed *Stair* PCF based space-filling design and quantify the gains achieved in the coverage characteristics (*i.e.* r_{min}).

Stair PCF in the Spatial Domain: The Stair PCF construction is defined as follows:

9. In our initial experiments, we found that increasing the peak height alone is sufficient for trading-off randomness to maximize coverage, and performs better than trading-off randomness by increasing oscillations in the PCF.

$$G(r; r_0, r_1, P_0) = f(r - r_1) + P_0 \left(f(r - r_0) - f(r - r_1)\right), \tag{19}$$
$$\text{with } f(r - r_0) = \left\{ \begin{array}{ll} 0 & \text{if } r \leq r_0 \\ 1 & \text{if } r > r_0 \end{array} \right\},$$
$$\text{where } r_0 \leq r_1 \text{ and } P_0 \geq 1.$$

This family of space-filling spectral designs has three interesting properties:

- except for a single peak in the region $r_0 \leq r \leq r_1$, the PCF is flat, thus, does not compromise randomness entirely,
- both the height and width of the peak can be optimized to maximize coverage,
- the Step PCF based spectral design can be derived as as a special case of this construction.

Now, the problem boils down to finding the combinations of the three parameters (r_0, r_1, P_0) that are realizable and yield a good sample design (discussed in Section 7.2). A representative example of Stair PCF is shown in Figure (7(a)).

Stair PCF in the Spectral Domain: Following the analysis in the earlier sections, we derive the power spectral density of Stair PCF based space-filling spectral designs.

Proposition 12 *The power spectral density of a Stair PCF based space-filling spectral designs, $G(r; r_0, r_1, P_0)$, with N samples in the sampling region with volume V is given by*

$$P(k) = 1 - \frac{N}{V} P_0 \left(\frac{2\pi r_0}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_0) - \frac{N}{V}(1 - P_0) \left(\frac{2\pi r_1}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_1).$$

Proof Using results from Section 4.2, we have

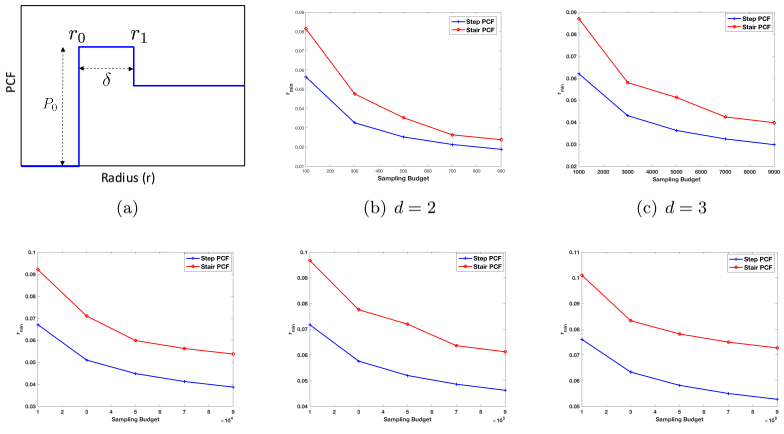
$$P(k) = 1 + \frac{N}{V} (2\pi)^{\frac{d}{2}} k^{1-\frac{d}{2}} H_{\frac{d}{2}-1} \left(r^{\frac{d}{2}-1}(G(r) - 1)\right). \tag{20}$$

To derive the PSD of a Stair function, we first evaluate the Hankel transform of $f(r) = (G(r) - 1)$ where $G(r)$ is a Stair function.

$$\begin{aligned} H_{\frac{d}{2}-1} \left(r^{\frac{d}{2}-1}(G(r) - 1)\right) &= \int_0^\infty r^{\frac{d}{2}} J_{\frac{d}{2}-1}(kr) \left(G(r) - 1\right) dr \\ &= -P_0 \int_0^{r_0} r^{\frac{d}{2}} J_{\frac{d}{2}-1}(kr) dr - (1 - P_0) \int_0^{r_1} r^{\frac{d}{2}} J_{\frac{d}{2}-1}(kr) dr \\ &= -P_0 \frac{r_0^{\frac{d}{2}}}{k} J_{\frac{d}{2}}(kr_0) - (1 - P_0) \frac{r_1^{\frac{d}{2}}}{k} J_{\frac{d}{2}}(kr_1) \end{aligned}$$

Using this expression in (20),

$$P(k) = 1 - \frac{N}{V} P_0 \left(\frac{2\pi r_0}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_0) - \frac{N}{V}(1 - P_0) \left(\frac{2\pi r_1}{k}\right)^{\frac{d}{2}} J_{\frac{d}{2}}(kr_1). \tag{21}$$



(a) (b) $d = 2$ (c) $d = 3$

(d) $d = 4$ (e) $d = 5$ (f) $d = 6$

Figure 7: (a) Pair correlation function of Stair PCF based designs, (b)-(f) Maximum Disk Radius For Step and Stair PCF for dimensions 2 to 6.

■

Next, we empirically evaluate the gain in coverage achieved by Stair PCF based designs compared to the Step PCF based designs.

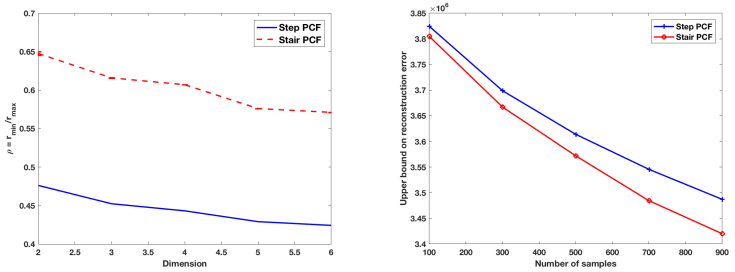
7.2 Coverage Gain with Stair PCF

Ideally, the optimal Stair PCF should be obtained by simultaneously maximizing r_0 ($:= r_{min}$) and minimizing P_0. Furthermore, not all PCFs in the Stair PCF family are realizable. Due to the realizability conditions, the parameters cannot be adjusted independently. The main challenge, therefore, is to find the combinations of the three parameters (r_0, r_1, P_0) that is realizable and yield a good sample design. Unlike Step PCF, the closed form expression for the optimal parameters (r_0, r_1, P_0) are difficult to obtain, and, therefore, we explore this family of PCF patterns empirically by searching configurations for which:

- the disk radius r_0 is as high as possible, and
- the PCF is flat with minimal increase in the peak height P_0.

7.2.1 Disk Radius r_{min} vs. Sample Budget N

In this section, we show the increase in coverage (or r_{min}) obtained by compromising randomness by increasing peak height in the PCF. We constrain the peak height to be below $P_0 \leq 1.5$ and analyze the gain in r_{min} due to this small compromise in randomness. Furthermore, we assume that $r_{min}^{step} \leq r_0 \leq 2 \times r_{min}^{step}$ and $r_0 \leq r_1 \leq 1.5 \times r_0$. In Figures 7(b)



(a) (b)

Figure 8: (a) Gain in the relative radius ρ achieved with the Stair PCF constructions, in comparison to the Step PCF constructions; (b) Upper bound on the reconstruction error of Step and Stair PCF based constructions.

through 7(f), we compare the maximum r_{min} achieved by the Step and Stair PCF designs, for varying sample sizes in dimensions 2 to 6. It can be seen that introducing a small peak in the PCF results in a significant increase in the coverage. This gain can be observed for all sampling budgets in all dimensions. Furthermore, as expected, for low sampling budgets maximal gain is observed, and should decrease with increasing N as the r_{min} for both the families will asymptotically (in N) converge to zero.

7.2.2 Relative Radius ρ vs. Dimension d

In this section, we study the increase in relative radius ρ due to the introduction of a peak in the PCF. Again, we assume that $P_0 \leq 1.5$, $r_{min}^{step} \leq r_0 \leq 2 \times r_{min}^{step}$ and $r_0 \leq r_1 \leq 1.5 \times r_0$. In Figure 8(a), we show the maximum $\rho = r_{min}/r_{max}$ achieved by the Step and Stair PCFs for different dimensions d. For Stair PCF, we do not have a closed form expression of ρ, thus, we obtain the maximum achievable r_{min} empirically for various sampling budgets and plot the mean (with standard deviation) behavior of the ρ. It can be seen that introducing a small peak in the PCF results in a significant increase in the relative radius. This gain can be observed at all sampling budgets in all dimensions. This also corroborates the recommendation of using $0.65 \leq \rho \leq 0.85$ in practice for coverage based designs and suggests that in higher dimensions ρ should be higher.

7.2.3 Analysis of Reconstruction Error Upper Bound

We also assess the reconstruction quality of the Step and Stair PCF based spectral designs, on the class of periodic functions considered in Section 5.1, for varying sampling budgets. Here, we consider the setup where $0 \leq k \leq 1000$ and $0 \leq f \leq 1000$. In Figure 8(b), we plot the average reconstruction error upper bounds as given in (11) for Step and Stair PCF. As expected, for both sample designs, the reconstruction error decreases with an

increase in the sampling budget. More interestingly, the reconstruction error of Stair PCF is lower compared to the reconstruction error of Step PCF, thus showing the effectiveness of increased coverage in sample designs.

8. Synthesis of Space-filling Spectral Designs

In this section, we describe the proposed approach for synthesizing sample designs that match the optimal (Stair or Step) PCF characteristics. Existing approaches for PCF matching such as (Oztireli and Gross, 2012; Kailkhura et al., 2016b) rely on kernel density estimators to evaluate the PCF of a point set. A practical limitation of these approaches is the lack of an efficient PCF estimator in high dimensions. More specifically, these estimators are biased due to lack of an appropriate edge correction strategy. This bias in PCF estimation arises due to the fact that sample hyper-spheres used in calculating point-pattern statistics may fall partially outside the study region and will produce a biased estimate of the PCF unless a correction is applied. The effect of this bias is barely noticeable in 2 dimensions and hence existing PCF matching algorithms have ignored this. However, this problem becomes severe in higher dimensions, thus, making the matching algorithm highly inaccurate. To address this crucial limitation, we introduce an edge corrected estimator for computing the PCF of sample designs in arbitrary dimensions. Following this, we describe a gradient descent based optimization technique to synthesize samples that match the desired PCF.

8.1 PCF Estimation in High Dimensions with Edge Correction

In order to create an unbiased PCF estimator, we propose to employ an edge corrected kernel density estimator, defined as follows:

$$\hat{G}(r) = \frac{V_W}{\gamma_W}\frac{V_W}{N}\frac{1}{S_E(N-1)}\sum_{i=1}^{N}\sum_{\substack{j=1\\ i\neq j}}^{N} k\left(r - |x_i - x_j|\right) \tag{22}$$

where $k(.)$ denotes the kernel function; here we use the classical Gaussian kernel

$$k(z) = \frac{1}{\sqrt{\pi}\sigma}\exp\left(-\frac{z^2}{2\sigma^2}\right). \tag{23}$$

In the above expression, V_W indicates the volume of the sampling region. When the sampling region is a hyper-cube with length 1, we have $V_W = 1$. Let S_E denote the area of hyper-sphere with radius r which is given by

$$S_E = \frac{dr^{d-1}\pi^{\frac{d}{2}}}{\Gamma(1+\frac{d}{2})}.$$

Also, we denote the surface area of the sampling region by S_W, which is expressed as

$$S_W = r^{d-1}\sin^{d-2}\phi_1 \sin^{d-3}\phi_2 \cdots \sin\phi_{d-2}.$$

The term $\frac{V_W}{\gamma_W}$ performs edge correction to handle the unboundedness of the estimator, where γ_W is an isotropic set covariance function given by

$$\gamma_W = \frac{1}{S_E}\int\limits_{\substack{0\leq\phi_{d-1}\leq 2\pi\\ 0\leq\phi_i\leq\pi, i=1 \text{ to } d-2}} S_W \gamma d\phi_1 \cdots d\phi_{d-1} \tag{24}$$

where $\gamma = \prod_{p=1}^{d}(1-|x^p|)$ with

$$\begin{aligned}
x^1 &= r\cos\phi_1\\
x^2 &= r\sin\phi_1\cos\phi_2\\
x^3 &= r\sin\phi_1\sin\phi_2\cos\phi_3\\
&\vdots\\
x^{d-1} &= r\sin\phi_1\cdots\sin\phi_{d-2}\cos\phi_{d-1}\\
x^d &= r\sin\phi_1\cdots\sin\phi_{d-2}\sin\phi_{d-1}.
\end{aligned}$$

In Figure 9(a), we show that by using an approximate edge correction factor (using the same factor as $d = 2$), the PCF is wrongly estimated. Moreover, as the dimension increases, the estimated PCF moves farther away from the true PCF very quickly.

Note that the calculation of the correct edge correction factor requires the evaluation of a multi-dimensional integral which is computationally expensive in high dimensions. In this paper, we provide a closed form approximation of γ_W (using polynomial regression of order 2) in different dimensions $d = 2$ to 6 when $r \leq 1.0$. More specifically, we have the following approximation $\hat{\gamma_W} = 1 - a_1 r + a_2 r^2$ where a_1 and a_2 are as given below.

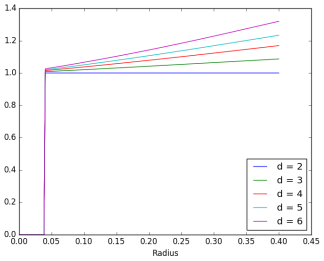
Dimension	$d=2$	$d=3$	$d=4$	$d=5$	$d=6$
a_1	$4/\pi$	1.47	1.63	1.75	1.89
a_2	$1/\pi$	0.54	0.72	0.87	1.04

It can be observed from Figures 9(b) through 9(f) that the proposed approximations are quite tight.

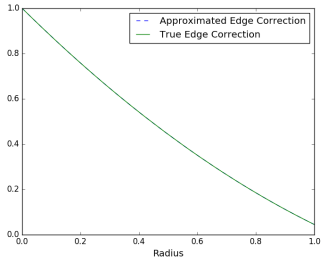
8.2 Synthesis Algorithm

The underlying idea of the proposed algorithm is to iteratively transform an initial random input sample design such that its PCF matches the target PCF. In particular, we propose a non-linear least squares formulation to optimize for the desired space-filling properties. Let us denote the target PCF by $G^*(r)$. We discretize the radius r into m points $\{r_j\}_{j=1}^{m}$ and minimize the sum of the weighted squares of errors between the target PCF $G^*(r_j)$ and the curve-fit function (kernel density estimator of PCF) $G(r_j)$ over m points. This scalar-valued goodness-of-fit measure is referred to as the *chi-squared error* criterion and can be posed as a non-linear weighted least squares problem as follows.

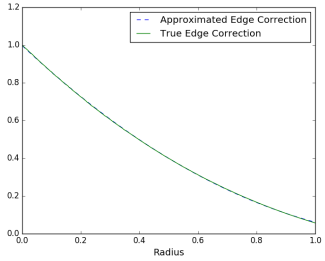
$$\arg\min \sum_{j=1}^{M}\left(\frac{G(r_j) - G^*(r_j)}{w_j}\right)^2,$$



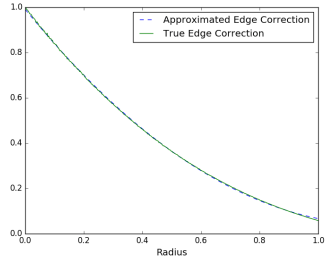
(a)



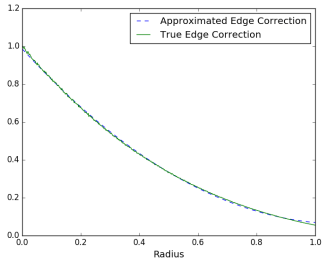
(b) $d = 2$



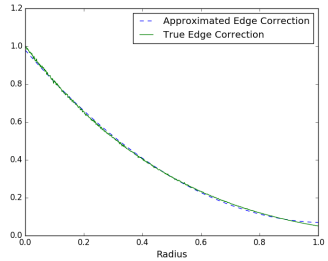
(c) $d = 3$



(d) $d = 4$



(e) $d = 5$



(f) $d = 6$

Figure 9: (a) Incorrect PCF estimation due to the use of an approximate edge correction factor, (b)-(f) Effectiveness of the approximation edge correction, obtained using polynomial regression, in comparison to the true edge correction from the evaluation of a multi-dimensional integral, for dimensions 2 to 6.

where w_j indicates the weight (importance) assigned to the fitting error at radius r_j. This optimization problem can be efficiently solved using a variant of gradient descent algorithm (discussed next), that in our experience converges quickly. In the simplest cases of uniform weights the solution tends to produce a higher fitting error at lower radii r_j. To address this challenge we use a non-uniform distribution for the weights $\{w_j\}$. These weights are initialized to be uniform and are updated in an adaptive fashion in the gradient descent iterations. The weight w_j at gradient descent iteration $t+1$ is given by (Kailkhura et al., 2016b):

$$w_j = \frac{1}{|G^t(r_j) - G^*(r_j)|}$$

where $G^t(r_j)$ is the value of the PCF at radius r_j during the gradient descent iteration t. Note that PCF matching is a highly non-convex problem. We found that the following trick further helps solve PCF matching problem more efficiently.

8.2.1 One Sided PCF Smoothing

We propose to perform one sided smoothing of the target PCF which is given as follows:

$$\hat{G}^*(r) = \begin{cases} (cr)^b & \text{if } r < r_{\min} \\ 1 & \text{if } r \geq r_{\min}. \end{cases}$$

where c is some pre-specified constant and $b > 1$ is the smoothing constant obtained via cross-validation. More specifically, we add polynomial noise in the low radius region of the PCF. This can also be interpreted as polynomial approximation of the PCF in the low radii regime. We have noticed that sometimes adding a controlled amount of Gaussian noise instead of polynomial noise also improves the performance.

8.2.2 Edge Corrected Gradient Descent

The non-linear least squares problem is solved iteratively using gradient descent. Starting with a random point set $X = \{x_i\}_{i=1}^N$, we iteratively update x_i in the negative gradient direction of the objective function. At each iteration k, this can be formally stated as

$$x_i^{k+1} = x_i^k - \lambda \frac{\Delta_i}{|\Delta_i|},$$

where λ is the step size and $\Delta_i = \{\Delta_i^k\}_{k=1}^d$ in the normalized edge corrected gradient is given by

$$\Delta_i^p = \sum_{i \neq l} \frac{(x_l^p - x_i^p)}{|x_l - x_i|} \sum_{j=1}^{m} \frac{G(r_j)^k - G^*(r_j)}{w_j(1 - a_1 r_j + a_2 r_j^2) r_j^{d-1}} \left(|x_l - x_i| - r_j\right) k\left(r_j - |x_i - x_l|\right). \tag{25}$$

We re-evaluate the PCF $G(r_j)^k$ of the updated point set after each iteration using the unbiased estimator from the previous section.

The pseudocode of the algorithm is provided in Algorithm 1.

Algorithm 1 Space-filling Spectral Sample Design using PCF Matching Algorithm

Input: Number of samples N, dimension d, Smoothed target PCF $\hat{G}^*(r_j)$, weights w_j, step size λ, edge correction factors (a_1, a_2)

$\mathbf{X} \leftarrow \text{Random}(N, d)$ ▷ Initial random sample design

$G \leftarrow \text{PCF}(\mathbf{X})$ ▷ Calculate initial PCF using Eq. (22)

for $t = 1$ to T **do** ▷ Total T gradient descent iterations

  for $i = 1$ to N **do** ▷ Update each sample at a time

    $\Delta_i^p \leftarrow \dfrac{\partial}{\partial x_i^p} \sum_{j=1}^{M} \left(\dfrac{G^t(r_j) - G^*(r_j)}{w_j} \right)^2$ for $p \in \{1, \cdots, d\}$ ▷ Calculate gradients using (25)

    $x_i^{t+1} \leftarrow x_i^t - \lambda \dfrac{\Delta_i}{|\Delta_i|}$ ▷ Update the sample

    $G^t \leftarrow \text{PCF}(\mathbf{X})$ ▷ Update the PCF

    $w_j \leftarrow \dfrac{1}{|G^t(r_j) - \hat{G}^*(r_j)|}$ ▷ Update weights

return X ▷ Space-filling Spectral Samples

In Figure 10, we compare the behavior of the proposed PCF matching algorithm with and without the one sided PCF smoothing. The target PCF is designed using a Step PCF design with r_{min} as given in Proposition 10. PCF matching is carried out with varying sampling budget, $N = 100,\ 200,\ 400,\ 600,\ 800$ for $d = 2,\ 3,\ 4,\ 5,\ 6$, respectively. The variances of the Gaussian kernel were set at $\sigma^2 = 0.0065,\ 0.007,\ 0.01,\ 0.01,\ 0.01$ for $d = 2,\ 3,\ 4,\ 5,\ 6$, respectively and the step size for the gradient descent algorithm was fixed at 0.001. The value of b was obtained using cross-validation. The initial point set was generated randomly (uniform) in the unit hyper-cube and matching was carried for 100 gradient descent iterations. It can be observed that the proposed algorithm produces an accurate fit to the target, and that the smoothing actually leads to improved performance.

In Figure 11, we demonstrate the synthesis of a Stair PCF based spectral design, using parameters $P_0 = 1.2, \delta = 0.025$. Similar to the previous case, PCF matching is carried out with varying sampling budget, $N = 100,\ 200,\ 400,\ 600,\ 800$, for $d = 2,\ 3,\ 4,\ 5,\ 6$ respectively. The variances of the Gaussian kernel were set at $\sigma^2 = 0.0065,\ 0.007,\ 0.01,\ 0.01,\ 0.01$ for $d = 2,\ 3,\ 4,\ 5,\ 6$, respectively and the step size for the gradient descent algorithm was fixed at 0.001. We found that matching the Stair PCF is more challenging for a gradient descent optimization compared to the Step PCF. When a random point set is used for initialization, reaching convergence takes much longer. However, choosing the initial point set intelligently improves the quality of matching significantly. In all our experiments, we used the maximal PDS (Ebeida et al., 2012) to initialize the optimization and matching was carried for 100 gradient descent iterations. We observed that another reasonable choice for the initialization is a regular grid sample, and interestingly in most cases it matches the performance of the MPDS initialization. Furthermore, one sided PCF smoothing does not provide significant improvements in this case, particularly in higher dimensions.

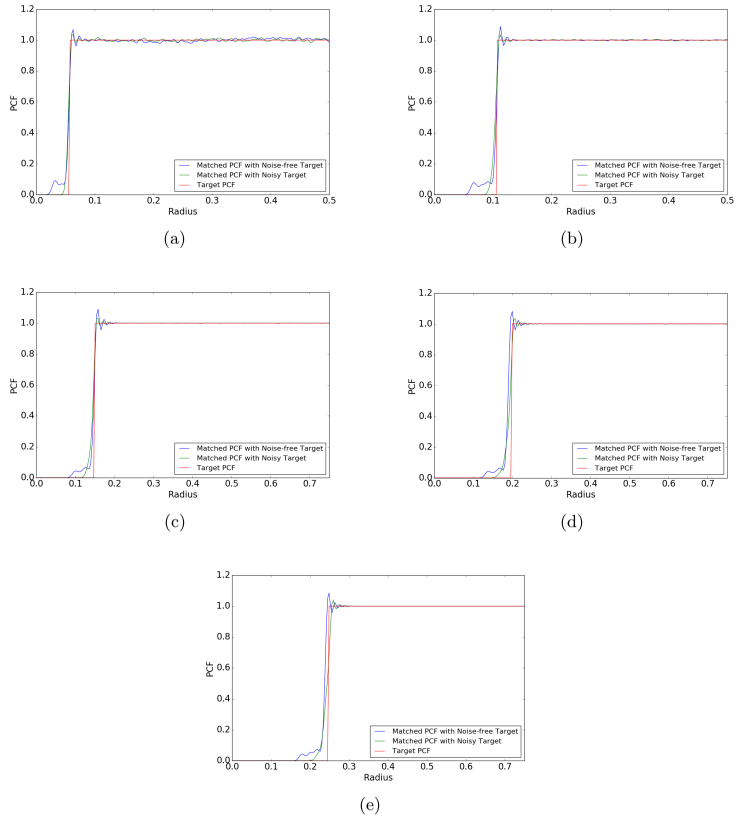


Figure 10: Step PCF synthesis using one sided PCF smoothing technique. (a) $d = 2$ (b) $d = 3$ (c) $d = 4$ (d) $d = 5$ (e) $d = 6$.

9. Experiments

In this section, we evaluate the qualitative performance of proposed space-filling spectral designs and present comparisons to popularly adopted space-filling designs, such as LHS, QMC and MPDS. Note that currently there does not exist any PDS synthesis approach which can generate sample sets with a desired size N while achieving user-specified spatial characteristics (e.g. relative radius). In all PDS synthesis approaches, there is no control over the number of samples generated by the algorithm which makes the use of these al-

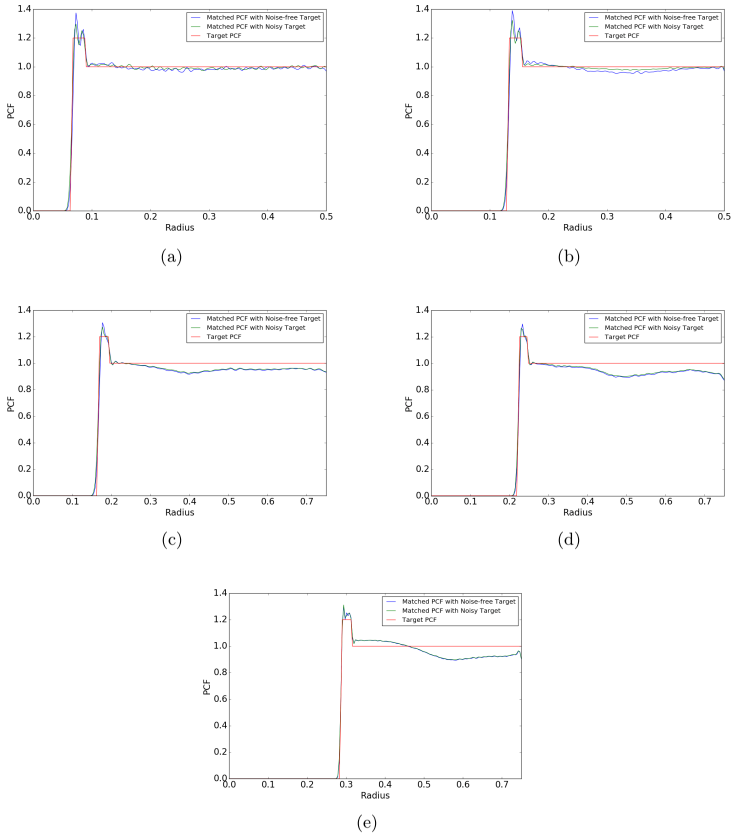


Figure 11: Stair PCF synthesis using one sided PCF smoothing technique. (a) $d = 2$ (b) $d = 3$ (c) $d = 4$ (d) $d = 5$ (e) $d = 6$.

gorithms difficult in practice. However, the proposed approach can control both N and r_{min} simultaneously. For our qualitative comparison, we perform three empirical studies, in dimensions 2 to 6 : (a) image reconstruction, (b) regression on several benchmark optimization functions, and (c) surrogate modeling for an inertial confinement fusion (ICF) simulation code.

Table 1: Impact of different space-filling designs on image reconstruction performance. In all cases, we show the reconstructed images and their PSNR values.

α	Sobol	Halton	LHS	MPDS	Step	Stair
0.001	14.82 dB	15.75 dB	14.29 dB	16.90 dB	16.00 dB	16.46 dB
0.002	12.10 dB	12.34 dB	11.39 dB	13.16 dB	12.58 dB	12.96 dB
0.003	11.22 dB	11.38 dB	10.75 dB	11.88 dB	11.55 dB	11.78 dB
0.004	10.61 dB	10.67 dB	10.34 dB	10.92 dB	10.80 dB	10.90 dB
0.005	9.80 dB	9.70 dB	9.59 dB	9.82 dB	9.83 dB	9.84 dB
0.006	9.09 dB	8.97 dB	8.97 dB	8.96 dB	9.08 dB	9.02 dB
0.007	8.49 dB	8.39 dB	8.51 dB	8.32 dB	8.49 dB	8.42 dB

9.1 Image Reconstruction

In this experiment, we consider the problem of designing sample distributions for image reconstruction. More specifically, we consider the commonly used zone plate test function:

$$z(r) = (1 + \cos(\alpha r^2))/2,$$

with varying levels of complexity (or frequency content) α. Note that we choose the zone plate for our study over natural images, since it shows the response for a wide range of frequencies and aliasing effects that are not masked by image features. For all zone plate renderings in this paper, we have tiled toroidal sets of 1000 2-dimensional points over the image-plane and utilized a Lanczos filter with a support of width 4 for resampling. Further, we also report the peak signal-to-noise ratio (PSNR) as a quantitative error measure:

$$\text{PSNR} = 20 \log_{10} \frac{1}{\text{MSE}},$$

where MSE is the mean squared error. However, it is well known in the image processing community that PSNR can be a weak surrogate for visual quality (as we will see later) and, therefore, we also show the reconstructed images.

Table 1 illustrates the reconstructions obtained using different space-filling designs, for varying values of α. It can be observed from the results that the QMC sequences produce a large amount of aliasing artifacts in the high frequency regions, which can be directly linked to the oscillations in their corresponding PCFs. On the other hand, LHS design recovers a small amount of low-frequencies, and maps most of the frequencies to white noise due to its small r_{min} and near-constant PCF. In contrast, sample designs which attempt to trade-off between coverage and randomness properties, i.e., MPDS and the proposed spectral space-filling designs (as seen in Figure 3), have superior reconstruction quality. These designs reduce the aliasing artifacts, have cleaner low frequency content (upper left corner)) and map all high frequencies (bottom right corner) to white noise. More interestingly, we see that for low complexity cases, i.e., lower α, the MPDS performs the best followed by the proposed Stair and Step PCF respectively. For moderately complex images, the Stair PCF performs the best followed by the Step and the MPDS. Finally, for highly complex images, the Step PCF performs the best followed by the Stair and the MPDS. These observations corroborate our discussion in Section 5.2 that an increase in r_{min} (coverage) in the PCF results in an increase in the range of low frequencies that can be recovered without aliasing, and equivalently reduction in the amount of oscillations (or an increase in randomness) in the PCF leads to reduced oscillations in the PSD, which in turn indicates a systematic mapping of high frequency content to white noise. Note that when $\alpha = 0.007$, both LHS and Sobol designs have PSNR greater than (or equal to) the PSNR of Step PCF design. However, the quality of the reconstructed image by Step PCF is far superior compared to the one by LHS and Sobol designs. This further corroborates our claim on PSNR being a weak surrogate and justifies the use of reconstructed images itself as a performance metric.

9.2 Regression Modeling for Benchmark Optimization Functions

In this study, we consider the problem of fitting regression models to analytical functions and perform a comparative study of different sample designs, in terms of their generalization

performance. More specifically, we consider a set of benchmark analytic functions between dimensions 2 and 6, that are commonly used in global optimization tests (Jamil and Yang, 2013). They are chosen due to their diversity in terms of their complexity and shapes. We compare the performance of proposed space-filling spectral designs (Step, Stair) with coverage based designs (MPDS), low-discrepancy designs (Halton and Sobol), latin hypercube sampling and random sampling. Appendix A lists the set of functions used in our experiments. In each case, we fit a random forest regressor with 30 trees and repeated for 20 independent realizations of sample designs. We evaluate the generalization performance on 10^6 regular grid based test samples. Finally, we report mean (horizontal lines) and standard deviation (vertical lines) of 3 popular quality metrics (over 20 realizations) to quantify the performance of the resulting regression models: mean squared error (MSE), relative average absolute error (AAE), and the R^2-statistic. The metrics are defined as follows:

$$MSE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}, \tag{26}$$

$$AAE(\mathbf{y}, \hat{\mathbf{y}}) = \frac{\sum_{i=1}^{N} |y_i - \hat{y}_i|}{N * STD(\mathbf{y})}, \tag{27}$$

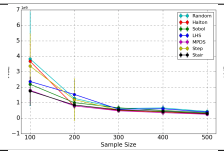
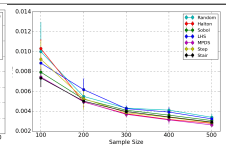
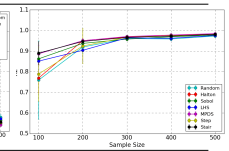
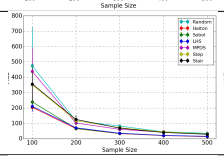
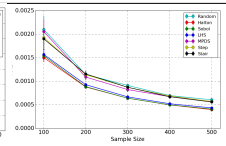
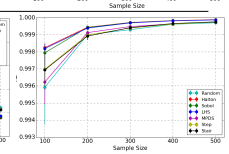
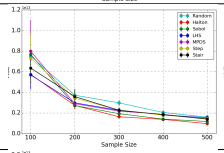
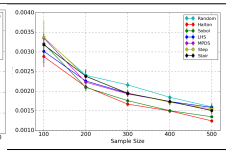
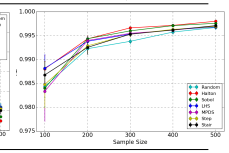
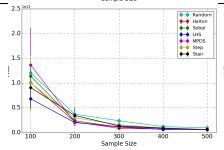
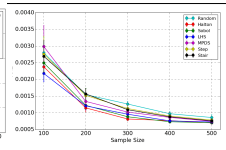
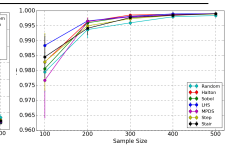
$$R^2(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{N} (y_i - MEAN(\mathbf{y}))^2} \tag{28}$$

where $\mathbf{y} = f(\mathbf{x})$ are the true function values and $\hat{\mathbf{y}}$ are the predicted values.

Tables 2 through 6 show the performance of different space-filling designs for various analytic functions in dimensions 2 to 6, respectively[10]. We see that, for $d = 2$ (Table 2), LHS and Halton sequences perform better compared to the rest of the sample designs on most of the test functions. However, on some functions, e.g., GoldsteinPrice, Stair PCF and MPDS perform better. Therefore, none of the sample designs consistently guarantee superior performance. For $d = 3$ (Table 3), we see that Stair PCF design and MPDS (followed by Sobol sequences) perform consistently better compared to the rest of the approaches. As we go higher in dimensions, i.e., $d > 3$, we notice a significant gain in the performance of Stair PCF based space-filling spectral designs. Interesting, the amount of performance gain of Stair PCF based design increases as we go higher in dimensions. The reason for the poor regression performance of QMC sequences and LHS for $d > 3$ is due to their poor space-filling properties in high dimensions (Wang and Sloan, 2008). In comparison, both space-filling spectral designs and MPDS have good space-filling properties. We found that Stair PCF design and MPDS have similar coverage characteristics (r_{min}). However, the difference in their performance can be attributed to the fact that MPDS designs have significantly more oscillations in their PCF compared to an equivalent Stair PCF based space-filling spectral design, i.e. violation of the randomness objective.

10. Non-monotonicity of the error curves represents over-fitting and is more prominent with conventional sample designs.

Table 2: Impact of sample design on generalization performance of regression models fit to benchmark analytical functions in 2 dimensions. LHS and Halton sequences perform slightly better compared to rest of the sample designs.

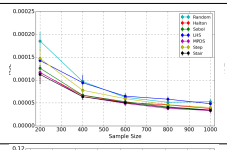
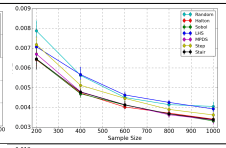
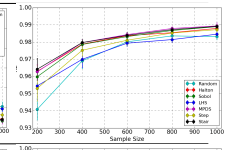
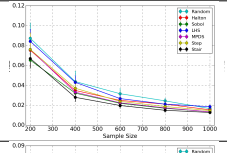
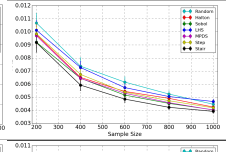
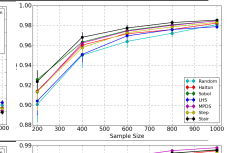
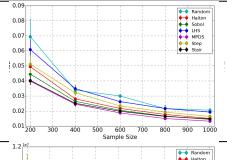
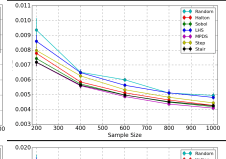
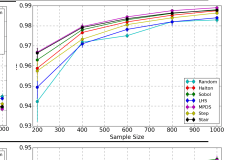
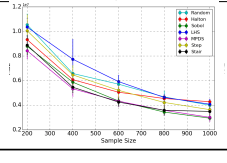
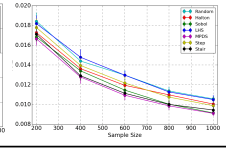
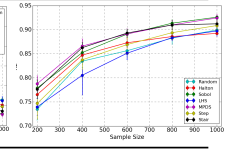
Function	MSE	AAE	R2-Statistic
GoldsteinPrice			
Chichinadze			
Rosenbrock			
Cube			

9.3 Surrogate Model Design for an Inertial Confinement Fusion (ICF) Simulator

In this subsection, we consider the problem of designing surrogate models for an inertial confinement fusion (ICF) simulator developed at the National Ignition Facility (NIF). The NIF is aimed at demonstrating inertial confinement fusion (ICF), that is, thermonuclear ignition and energy gain in a laboratory setting. The goal is to focus 192 beams of the most energetic laser built so far onto a tiny capsule containing frozen deuterium. Under the right conditions, the resulting pressure will collapse the target to the point of ignition where hydrogen starts to fuse and produce massive amounts of energy, effectively creating a small star which can be harnessed for energy production. Though significant progress has been made, the ultimate goal of "ignition" has not yet been reached.

NIF employs an adaptive pipeline: perform experiments, use post-shot simulations to understand the experimental results, and design new experiments with parameter settings that are expected to improve performance. From an analysis viewpoint, the goal is to search the parameter space to find the region that leads to near-optimal performance. The dataset

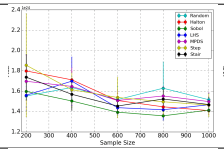
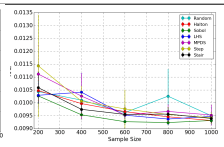
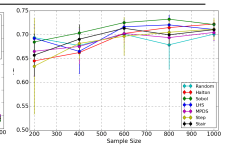
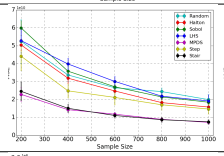
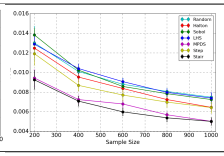
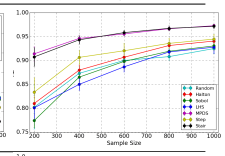
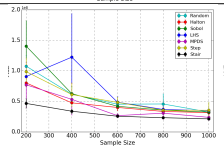
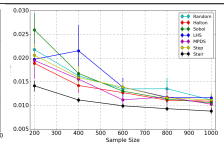
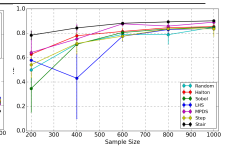
Table 3: Impact of sample design on generalization performance of regression models fit to benchmark analytical functions in 3 dimensions. While the Stair PCF and MPDS designs are consistently better than the other methods, the amount of performance gain is minimal.

Function	MSE	AAE	R2-Statistic
BoxBetts			
Hartmann3			
Wolfe			
Helical Valley			

considered here is a so called engineering or macro-physics simulation ensemble in which an implosion is simulated using different input parameters, such as, laser power, pulse shape etc. From these simulations, scientists extract a set of drivers, physical quantities believed to determine the behavior of the resulting implosion. These drivers are then analyzed with respect to the energy yield to better understand how to optimize future experiments. As one can expect, the success of this pipeline heavily depends on the quality of samples used for post-shot simulations.

We use the NIF 1-d HYDRA simulator (Marinak et al., 2001) and compare the performance of proposed space-filling spectral designs with existing approaches (random, LHS, Halton and Sobol and MPDS). For each simulation run, a large number of output quantities, such as peak velocity, yield, etc., are computed, and subsequently used to describe the resulting implosion. We vary the number of input parameters between 2 and 6, and fix the remaining variables to their default values. In each case, we fit a random forest regressor with 30 trees and repeated for 20 independent realizations of sample designs. We evaluated

Table 4: Impact of sample design on generalization performance of regression models fit to benchmark analytical functions in 4 dimensions. Stair PCF and MPDS designs demonstrate appreciable gains over popular sample design choices.

Function	MSE	AAE	R2-Statistic
DeVilliersGlasser01			
Colville			
Powell			

the reconstruction performance on 10^5 regular grid based test samples using the metrics in the previous experiment.

Table 7 shows the regression performance of the different sample designs for various output quantities in dimensions 2 to 6. We observe that regression error patterns are consistent with our observations in Section 9.2. The proposed Stair PCF based design consistently performs the best (followed by MPDS) for $d \geq 3$. Furthermore, the performance gain with the Stair PCF based design improves as we go higher in dimensions. This performance gain can be credited to their ability to achieve better space-filling properties in high dimensions by intelligently balancing the trade-off between coverage and randomness, and the effectiveness of the proposed metric (PCF) adopted for design and optimization.

10. Applications of Design of Experiments in Machine Learning Problems

In addition to the experiments presented in the paper, the proposed experiment design methodologies have a broader impact on several classical machine learning (ML) formulations. In its simplest form, the proposed sample designs can be used to create training data in supervised learning problems. In particular, optimized sample designs can provide significant performance gains in application areas where efficient data acquisition is required, e.g. machine learning for scientific data analysis (Karpatne et al., 2017). One such use case was considered in Section 9.3 of this paper for inertial confinement fusion (ICF) studies. We demonstrated that supervised models learned using samples from the proposed exper-

Table 5: Impact of sample design on generalization performance of regression models fit to benchmark analytical functions in 5 dimensions. In higher dimensions, conventional methods such as the LHS and QMC perform very poorly, while Stair PCF design significantly outperforms all competing methods, because of improved trade-off between coverage and randomness properties.

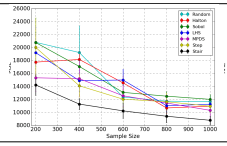
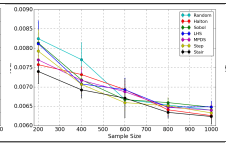
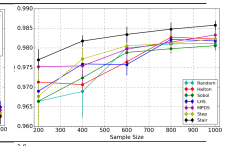
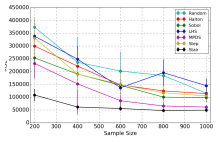
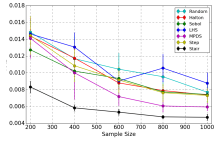
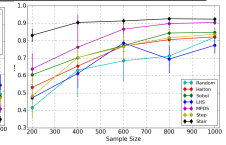
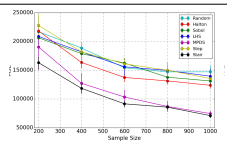
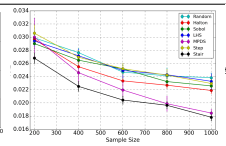
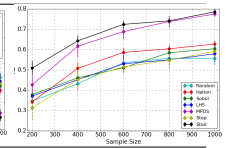
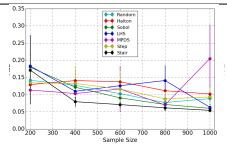
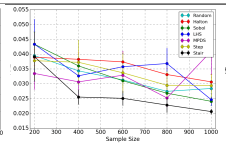
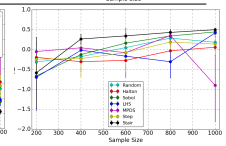
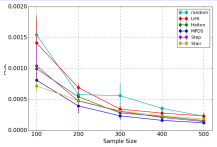
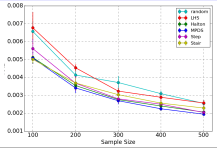
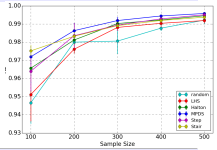
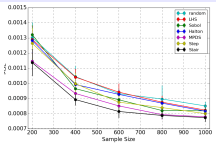
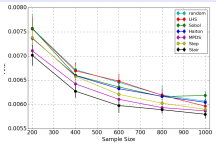
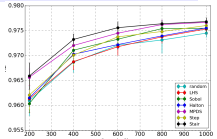
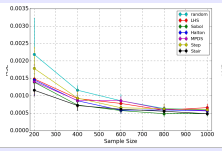
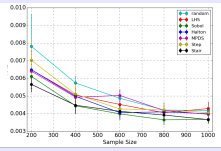
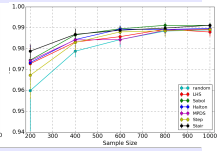
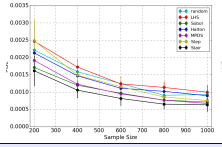
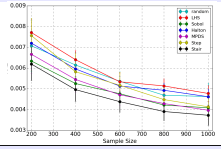
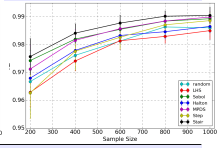
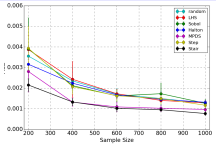
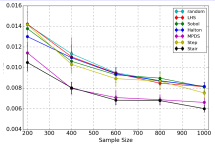
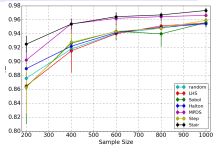
Function	MSE	AAE	R2-Statistic
Dolan			
BiggsExp05			

Table 6: Impact of sample design on generalization performance of regression models fit to benchmark analytical functions in 6 dimensions. Even with highly complex functions such as the *Hartmann6*, the proposed Stair PCF based spectral space-filling design produces more accurate regression models, thus evidencing the importance of improved space-filling characteristics.

Function	MSE	AAE	R2-Statistic
Trid			
Hartmann6			

iment design produce highly generalizable models, i.e., have significantly lower prediction error for unseen input conditions. Furthermore, utilizing the notion of space-filling spectral designs to incorporate the prior information on signal structure (e.g., low dimensionality, sparsity) will be particularly useful for image analysis and computer vision tasks. Though not discussed, the analytical framework developed in this paper, can be extended beyond

Table 7: Performance of surrogate models for the NIF 1−d HYDRA simulator using different sample design techniques, with varying number of input parameters. While the conventional sample designs achieve reasonable performance in low dimensions, the proposed Stair PCF based design is consistently superior as the dimension of the input space grows.

Function	MSE	AAE	R2-Statistic
Parameter Space Dimension = 2			
PEAK fusion power			
Parameter Space Dimension = 3			
Radiation energy			
Parameter Space Dimension = 4			
MINradius shock			
Parameter Space Dimension = 5			
MINradius shock			
Parameter Space Dimension = 6			
MAXpressure			

Euclidean spaces to non-linear, embedded manifolds. For example, Anirudh et.al. (Anirudh et al., 2017) proposed a dart throwing technique to generate Poisson disk samples on the

Grassmannian manifold of low-dimensional linear subspaces. Despite the effectiveness of such heuristic techniques over other randomized sampling strategies, generating space-filling spectral designs on embedded manifolds is challenging since it is non-trivial to create equivalent definitions of the PSD and the PCF metrics for non-Euclidean domains.

Furthermore, there is a connection between the sample design problem considered in this paper and the classic ML task of active learning. In many practical scenarios, it is possible to use information gleaned from previous observations to improve the sampling process. As more samples are obtained, one can learn how to improve the sampling process by deciding where to sample next. These sampling feedback techniques are more generally known as adaptive sampling in the statistics literature. Note that, several popular design of experiment techniques have been extended to adaptive sampling scenarios (Yan and Wonka, 2012b). A natural extension of our work is towards builds importance sampling techniques, guided by spectral properties. In addition to these conventional applications, more recently, optimized sample designs have been used to improve the convergence characteristics of neural network training process. Several efforts are currently being undertaken for effective mini-batch sampling and studying their effect on the convergence rate of training algorithms. Similarly, one could develop improved mini-batch sampling strategies through the analytical framework of space-filling spectral designs. Similarly, hyper-parameter optimization in deep learning is another application area where optimized sample design can be very useful (Bergstra and Bengio, 2012). Finally, space-filling spectral designs are also applicable in reinforcement learning (Sutton and Barto, 1998) and Bayesian optimization (Snoek et al., 2012) where a key requirement is to effectively balance between exploration and exploitation.

11. Conclusion and Future Directions

In this work, we considered the problem of constructing high quality space-filling designs. We proposed the use of pair correlation function (PCF) to quantify the space-filling property and systematically traded-off coverage and randomness in sample designs in arbitrary dimensions. Next, we linked PCF to the power spectral density (PSD) to analyze the qualitative measure of the design performance. Using the insights provided by this spatial-spectral analysis, we proposed novel space-filling spectral designs. We also provided an efficient PCF estimator to evaluate the space-filling properties of sample designs in arbitrary dimensions. Next, we devised a gradient descent based optimization algorithm to generate high quality space-filling designs. Superiority of proposed space-filling spectral designs were shown on two different applications in 2 to 6 dimensions: a) image reconstruction and b) surrogate modeling on several benchmark optimization functions and an inertial confinement fusion (ICF) simulation code. There are still many interesting questions that remain to be explored in the future work such as an analysis of the problem for non-linear manifolds. Note that some analytical methodologies used in this paper are certainly exploitable for studying and designing space-filling designs in different manifolds. Other questions such as PCF parameterizations for other variants of space-filling designs, adaptive and importance sampling, and optimization approaches to synthesize them can also be investigated.

Acknowledgments

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. LLNL-JRNL-743060

Appendix A. Benchmark Optimization Functions

A.1 2 Dimensional Functions

Rosenbrock: $\sum_{p=1}^{2}\left(100((x^p)^2 - x^{p+1}) + (x^p - 1)^2\right)$
Cube: $100(x^2 - (x^1)^3)^2 + (1 - x^1)^2$
Chichinadze: $(x^1)^2 - 12x^1 + 8\sin(2.5\pi x^1) + 10\cos(0.5\pi x^1) + 11 - 0.2\frac{\sqrt{5}}{\exp(0.5(x^2 - 0.5)^2)}$
GoldsteinPrice: $\left(1 + (x^1 + x^2 + 1)^2(19 - 14x^1 + 3(x^1)^2 - 14x^2 + 6x^1x^2 + 3(x^2)^2)\right)$
$\left(30 + (2x^1 - 3x^2)^2(18 - 32x^1 + 12(x^1)^2 + 48x^2 - 36x^1x^2 + 27(x^2)^2)\right)$

A.2 3 Dimensional Functions

BoxBetts: $\sum_{p=1}^{3} g(x^i)^2; g(x) = \exp(-0.1(p+1)x^1) - \exp(-0.1(p+1)x^2) - (\exp(-0.1(p+1)) - \exp(-(p+1))x^3)$
HelicalValley: $100(x^3 - 10\psi(x^1, x^2))^2 + (\sqrt{(x^1)^2 + (x^2)^2} - 1)^2 + (x^3)^2; 2\pi\psi(x^1, x^2) = \tan^{-1}(x^2/x^1)$ **if** $x^1 \geq 0$, **and** $\pi + \tan^{-1}(x^2/x^1)$ **otherwise.**
Wolfe: $\frac{4}{3}((x^1)^2 + (x^2)^2 - x^1x^2)^{0.75} + x^3$
Hartmann3: $-\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{3} a_{ij}(x^j - p_{ij})^2)$

A.3 4 Dimensional Functions

DeVilliersGlasser01: $\sum_{i=1}^{24}\left(x^1(x^2)^{0.1(i-1)}\sin(x^3(0.1(i-1)) + x^4) - y_i\right)^2$
Powell: $(x^3 + 10x^1)^2 + 5(x^2 - x^4)^2 + (x^1 - 2x^2)^4 + 10(x^3 - x^4)^4$
Colville: $(x^1-1)^2+100((x^1)^2-x^2)^2+10.1(x^2-1)^2+(x^3-1)^2+90((x^3)^2-x^4)^2+10.1(x^4-1)^2 + 19.8\frac{x^4-1}{x^2}$

A.4 5 Dimensional Functions

BiggsExp05: $\sum_{i=1}^{11}(x^3e^{-t_ix^1} - x^4e^{-t_ix^2} + 3e^{-t_ix^5} - y_i)^2;\ t_i = 0.1i;\ y_i = e^{-t_i} - 5e^{-10t_i} + 3e^{-4t_i}$
Dolan: $|(x61 + 1.7x^2)\sin(x^1) - 1.5x^3 - 0.1x^4\cos(x^5 - x^1) + 0.2(x^5)^2 - x^2 - 1|$

A.5 6 Dimensional Functions

Trid: $\sum_{p=1}^{6}(x^p - 1)^2 - \sum_{p=2}^{6} x^p x^{p-1}$
Hartmann6: $-\sum_{i=1}^{4} c_i \exp(-\sum_{j=1}^{6} a_{ij}(x^j - p_{ij})^2)$

References

R. Anirudh, B. Kailkhura, J. J. Thiagarajan, and P. T. Bremer. Poisson disk sampling on the grassmannnian: Applications in subspace optimization. *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 1(1):690–698, July 2017. ISSN 2160-7516.

Michael Balzer, Thomas Schlomer, and Oliver Deussen. Capacity-constrained point distributions: A variant of lloyd's method. *ACM Trans. Graph.*, 28(3):86:1–86:8, July 2009. ISSN 0730-0301.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

Robert Bridson. Fast poisson disk sampling in arbitrary dimensions. *ACM SIGGRAPH 2007 Sketches*, 2007.

Russel E. Caflisch. Monte carlo and quasi-monte carlo methods. *Acta Numerica*, 7:149, 1998.

Henry Cohn, Abhinav Kumar, Stephen D Miller, Danylo Radchenko, and Maryna Viazovska. The sphere packing problem in dimension 24. *Annals of Mathematics*, 185(3): 1017–1033, 2017.

Robert L. Cook. Stochastic sampling in computer graphics. *ACM Trans. Graph.*, 5(1): 51–72, January 1986. ISSN 0730-0301.

Fernando de Goes, Katherine Breeden, Victor Ostromoukhov, and Mathieu Desbrun. Blue noise through optimal transport. *ACM Trans. Graph.*, 31(6):171:1–171:11, November 2012. ISSN 0730-0301.

Mark A. Z. Dippe and Erling Henry Wold. Antialiasing through stochastic sampling. *SIGGRAPH Comput. Graph.*, 19(3):69–78, July 1985. ISSN 0097-8930.

Daniel Dunbar and Greg Humphreys. A spatial data structure for fast poisson-disk sample generation. *ACM Trans. Graph.*, 25(3):503–508, July 2006. ISSN 0730-0301.

Mohamed S. Ebeida, Andrew A. Davidson, Anjul Patney, Patrick M. Knupp, Scott A. Mitchell, and John D. Owens. Efficient maximal poisson-disk sampling. *ACM Trans. Graph.*, 30(4):49:1–49:12, July 2011. ISSN 0730-0301.

Mohamed S. Ebeida, Scott A. Mitchell, Anjul Patney, Andrew A. Davidson, and John D. Owens. A simple algorithm for maximal poisson-disk sampling in high dimensions. *Computer Graphics Forum*, 31(2pt4):785–794, 2012. ISSN 1467-8659.

Mohamed S. Ebeida, Anjul Patney, Scott A. Mitchell, Keith R. Dalbey, Andrew A. Davidson, and John D. Owens. K-d darts: Sampling by k-dimensional flat searches. *ACM Trans. Graph.*, 33(1):3:1–3:16, February 2014. ISSN 0730-0301.

Ronald A Fisher. The design of experiments. 1935. *Oliver and Boyd, Edinburgh*, 1935.

Manuel N. Gamito and Steve C. Maddock. Accurate multidimensional poisson-disk sampling. *ACM Trans. Graph.*, 29(1):8:1–8:19, December 2009. ISSN 0730-0301.

Sushant S. Garud, Iftekhar A. Karimi, and Markus Kraft. Design of computer experiments: A review. *Computers and Chemical Engineering*, 106(Supplement C):71 – 95, 2017. ISSN 0098-1354. ESCAPE-26.

Bo Geng, HuiJuan Zhang, Heng Wang, and GuoPing Wang. Approximate poisson disk sampling on mesh. *Science China Information Sciences*, 56(9):1–12, 2013. ISSN 1674-733X.

Jianwei Guo, Dong-Ming Yan, Guanbo Bao, Weiming Dong, Xiaopeng Zhang, and Peter Wonka. Efficient triangulation of poisson-disk sampled point sets. *The Visual Computer*, 30(6-8):773–785, 2014. ISSN 0178-2789.

J. H. Halton. Algorithm 247: Radical-inverse quasi-random point sequence. *Commun. ACM*, 7(12):701–702, December 1964. ISSN 0001-0782.

Daniel Heck, Thomas Schlomer, and Oliver Deussen. Blue noise sampling with controlled aliasing. *ACM Trans. Graph.*, 32(3):25:1–25:12, July 2013. ISSN 0730-0301.

Wenguang Hou, Xuming Zhang, Xin Li, Xudong Lai, and Mingyue Ding. Poisson disk sampling in geodesic metric for dem simplification. *International Journal of Applied Earth Observation and Geoinformation*, 23(1):264 – 272, 2013. ISSN 0303-2434.

Janine Illian, Antti Penttinen, Helga Stoyan, and Dietrich Stoyan. *Statistical analysis and modelling of spatial point patterns*, volume 70. John Wiley and Sons, 2008.

Cheuk Yiu Ip, M. Adil Yalin, David Luebke, and Amitabh Varshney. Pixelpie: Maximal poisson-disk sampling with rasterization. *Proceedings of the 5th High-Performance Graphics Conference*, pages 17–26, 2013.

Momin Jamil and Xin-She Yang. A literature survey of benchmark functions for global optimisation problems. *International Journal of Mathematical Modelling and Numerical Optimisation*, 4(2):150–194, 2013.

Ruichen Jin, Wei Chen, and Agus Sudjianto. An efficient algorithm for constructing optimal design of computer experiments. *Journal of Statistical Planning and Inference*, 134(1): 268 – 287, 2005. ISSN 0378-3758.

V Roshan Joseph. Space-filling designs for computer experiments: A review. *Quality Engineering*, 28(1):28–35, 2016.

B. Kailkhura, J. J. Thiagarajan, P. T. Bremer, and P. K. Varshney. Theoretical guarantees for poisson disk sampling using pair correlation function. *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2589–2593, March 2016a.

Bhavya Kailkhura, Jayaraman J. Thiagarajan, Peer-Timo Bremer, and Pramod K. Varshney. Stair blue noise sampling. *ACM Trans. Graph.*, 35(6):248:1–248:10, November 2016b. ISSN 0730-0301.

Anuj Karpatne, Gowtham Atluri, James H Faghmous, Michael Steinbach, Arindam Banerjee, Auroop Ganguly, Shashi Shekhar, Nagiza Samatova, and Vipin Kumar. Theory-guided data science: A new paradigm for scientific discovery from data. *IEEE Transactions on Knowledge and Data Engineering*, 29(10):2318–2331, 2017.

JR Koehler and AB Owen. Computer experiments. *Handbook of statistics*, 13:261–308, 1996.

Ares Lagae and Philip Dutr. A comparison of methods for generating poisson disk distributions. *Computer Graphics Forum*, 27(1):114–129, 2008. ISSN 1467-8659.

Stephen Leary, Atul Bhaskar, and Andy Keane. Optimal orthogonal-array-based latin hypercubes. *Journal of Applied Statistics*, 30(5):585–598, 2003.

Pierre LEcuyer and Christiane Lemieux. Recent advances in randomized quasi-monte carlo methods. *Modeling uncertainty*, pages 419–474, 2005.

MM Marinak, GD Kerbel, NA Gentile, O Jones, D Munro, S Pollaine, TR Dittrich, and SW Haan. Three-dimensional hydra simulations of national ignition facility targets. *Physics of Plasmas*, 8(5):2275–2280, 2001.

Michael McCool and Eugene Fiume. Hierarchical poisson disk sampling distributions. *Proceedings of the Conference on Graphics Interface '92*, pages 94–105, 1992.

Michael D. McKay. Latin hypercube sampling as a tool in uncertainty analysis of computer models. *Proceedings of the 24th Conference on Winter Simulation*, pages 557–564, 1992.

William J. Morokoff and Russel E. Caflisch. Quasi-random sequences and their discrepancies. *SIAM J. Sci. Comput*, 15:1251–1279, 1994.

Max D. Morris and Toby J. Mitchell. Exploratory designs for computational experiments. *Journal of Statistical Planning and Inference*, 43(3):381 – 402, 1995. ISSN 0378-3758.

Victor Ostromoukhov, Charles Donohue, and Pierre-Marc Jodoin. Fast hierarchical importance sampling with blue noise properties. *ACM Trans. Graph.*, 23(3):488–495, August 2004. ISSN 0730-0301.

Art B. Owen. *Randomly Permuted (t,m,s)-Nets and (t, s)-Sequences*. Springer New York, New York, NY, 1995. ISBN 978-1-4612-2552-2.

Art B Owen. Monte carlo and quasi-monte carlo for statistics. *Monte Carlo and Quasi-Monte Carlo Methods 2008*, pages 3–18, 2009.

Art B Owen and Seth D Tribble. A quasi-monte carlo metropolis algorithm. *Proceedings of the National Academy of Sciences of the United States of America*, 102(25):8844–8849, 2005.

A. Cengiz Oztireli and Markus Gross. Analysis and synthesis of point distributions based on pair correlation. *ACM Trans. Graph.*, 31(6):170:1–170:10, November 2012. ISSN 0730-0301.

N. Packham. Combining latin hypercube sampling with other variance reduction techniques. *Wilmott*, 2015(76):60–69, 2015. ISSN 1541-8286.

Thomas Schlomer, Daniel Heck, and Oliver Deussen. Farthest-point optimized point sets with maximized minimum distance. *Proceedings of the ACM SIGGRAPH Symposium on High Performance Graphics*, pages 135–142, 2011.

Edwin W. Schreiber. Mathematical snapshots*. *School Science and Mathematics*, 43(9): 795–799, 1943. ISSN 1949-8594.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, pages 2951–2959, 2012.

I.M Sobol. On the distribution of points in a cube and the approximate evaluation of integrals. *USSR Computational Mathematics and Mathematical Physics*, 7(4):86 – 112, 1967. ISSN 0041-5553.

Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Boxin Tang. Orthogonal array-based latin hypercubes. *Journal of the American statistical association*, 88(424):1392–1397, 1993.

maryna s viazovska. the sphere packing problem in dimension 8. *annals of mathematics*, 185(3):991–1015, 2017.

Florent Wachtel, Adrien Pilleboue, David Coeurjolly, Katherine Breeden, Gurprit Singh, Gael Cathelin, Fernando de Goes, Mathieu Desbrun, and Victor Ostromoukhov. Fast tile-based adaptive sampling with user-specified fourier spectra. *ACM Trans. Graph.*, 33 (4):56:1–56:11, July 2014. ISSN 0730-0301.

Xiaoqun Wang and Ian H. Sloan. Low discrepancy sequences in high dimensions: How well are their projections distributed? *Journal of Computational and Applied Mathematics*, 213(2):366 – 386, 2008. ISSN 0377-0427.

Li-Yi Wei. Parallel poisson disk sampling. *ACM Trans. Graph.*, 27(3):20:1–20:9, August 2008. ISSN 0730-0301.

Li-Yi Wei. Multi-class blue noise sampling. *ACM Trans. Graph.*, 29(4):79:1–79:8, July 2010. ISSN 0730-0301.

Qingtong Xu, Jing Wang, and Xuandong An. A pipeline for surface reconstruction of 3-dimentional point cloud. *Audio, Language and Image Processing (ICALIP), 2014 International Conference on*, pages 822–826, July 2014.

Dong-Ming Yan and Peter Wonka. Adaptive maximal poisson-disk sampling on surfaces. *SIGGRAPH Asia 2012 Technical Briefs*, pages 21:1–21:4, 2012a.

Dong-Ming Yan and Peter Wonka. Adaptive maximal poisson-disk sampling on surfaces. *SIGGRAPH Asia 2012 Technical Briefs*, page 21, 2012b.

Dong-Ming Yan and Peter Wonka. Gap processing for adaptive maximal poisson-disk sampling. *ACM Trans. Graph.*, 32(5):148:1–148:15, October 2013. ISSN 0730-0301.

JI Yellott. Spectral consequences of photoreceptor sampling in the rhesus retina. *Science*, 221(4608):382–385, 1983. ISSN 0036-8075.

Xiang Ying, Zhenhua Li, and Ying He. A parallel algorithm for improving the maximal property of poisson disk sampling in r2 and r3. *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 179–179, 2013a.

Xiang Ying, Shi-Qing Xin, Qian Sun, and Ying He. An intrinsic algorithm for parallel poisson disk sampling on arbitrary surfaces. *Visualization and Computer Graphics, IEEE Transactions on*, 19(9):1425–1437, Sept 2013b. ISSN 1077-2626.

Xiang Ying, Zhenhua Li, and Ying He. A parallel algorithm for improving the maximal property of poisson disk sampling. *Computer-Aided Design*, 46(1):37 – 44, 2014. ISSN 0010-4485.

Yahan Zhou, Haibin Huang, Li-Yi Wei, and Rui Wang. Point sampling with general noise spectrum. *ACM Trans. Graph.*, 31(4):76:1–76:11, July 2012. ISSN 0730-0301.

Journal of Machine Learning Research 19 (2018) 1-49 Submitted 7/16; Revised 8/18; Published 9/18

Kernel Density Estimation for Dynamical Systems

Hanyuan Hang HANS2017@RUC.EDU.CN
Institute of Statistics and Big Data
Renmin University of China
100872 Beijing, China

Ingo Steinwart INGO.STEINWART@MATHEMATIK.UNI-STUTTGART.DE
Institute for Stochastics and Applications
University of Stuttgart
70569 Stuttgart, Germany

Yunlong Feng YLFENG@ALBANY.EDU
Department of Mathematics and Statistics
State University of New York
The University at Albany
Albany, New York 12222, USA

Johan A.K. Suykens JOHAN.SUYKENS@ESAT.KULEUVEN.BE
Department of Electrical Engineering, ESAT-STADIUS, KU Leuven
Kasteelpark Arenberg 10, Leuven, B-3001, Belgium

Editor: John Shawe-Taylor

Abstract

We study the density estimation problem with observations generated by certain dynamical systems that admit a unique underlying invariant Lebesgue density. Observations drawn from dynamical systems are not independent and moreover, usual mixing concepts may not be appropriate for measuring the dependence among these observations. By employing the $\mathcal{C}$-mixing concept to measure the dependence, we conduct statistical analysis on the consistency and convergence of the kernel density estimator. Our main results are as follows: First, we show that with properly chosen bandwidth, the kernel density estimator is universally consistent under L_1-norm; Second, we establish convergence rates for the estimator with respect to several classes of dynamical systems under L_1-norm. In the analysis, the density function f is only assumed to be Hölder continuous or pointwise Hölder controllable which is a weak assumption in the literature of nonparametric density estimation and also more realistic in the dynamical system context. Last but not least, we prove that the same convergence rates of the estimator under L_∞-norm and L_1-norm can be achieved when the density function is Hölder continuous, compactly supported, and bounded. The bandwidth selection problem of the kernel density estimator for dynamical system is also discussed in our study via numerical simulations.

Keywords: Kernel density estimation, dynamical system, dependent observations, $\mathcal{C}$-mixing process, universal consistency, convergence rates, covering number, learning theory

1. Introduction

Dynamical systems are now ubiquitous and are vital in modeling complex systems, especially when they admit recurrence relations. Statistical inference for dynamical systems has drawn continuous attention across various fields, the topics of which include parameter estimation, invariant measure estimation, forecasting, noise detection, among others. For instance, in the statistics and machine learning community, the statistical inference for certain dynamical systems have been recently studied in Suykens et al. (1995); Suykens and Vandewalle (2000); Suykens et al. (2002); Zoeter and Heskes (2005); Anghel and Steinwart (2007); Steinwart and Anghel (2009); Deisenroth and Mohamed (2012); McGoff et al. (2015a); Hang and Steinwart (2017), just to name a few. We refer the reader to a recent survey in McGoff et al. (2015b) for a general depiction of this topic. The purpose of this study is to investigate the density estimation problem for dynamical systems via a classical nonparametric approach, i.e., kernel density estimation.

The commonly considered density estimation problem can be stated as follows. Let x_1, x_2, $\ldots, x_n$ be observations drawn independently from an unknown distribution P on $\mathbb{R}^d$ with the density f. Density estimation is concerned with the estimation of the underlying density f. Accurate estimation of the density is important for many machine learning tasks such as regression, classification, and clustering problems and also plays an important role in many real-world applications. Nonparametric density estimators are popular since weaker assumptions are applied to the underlying probability distribution. Typical nonparametric density estimators include the histogram and kernel density estimator. In this study, we are interested in the latter one, namely, *kernel density estimator*, which is also termed as *Parzen-Rosenblatt estimator* (Parzen, 1962; Rosenblatt, 1956) and takes the following form

$$f_n(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{1}$$

Here, $h := h_n > 0$ is a bandwidth parameter and K is a smoothing kernel. In the literature, point-wise and uniform consistency and convergence rates of the estimator f_n to the unknown truth density f under various distance measurements, e.g., L_1, L_2, and L_∞, have been established by resorting to the regularity assumptions on the smoothing kernel $\mathcal{K}$, the density f, and the decay of the bandwidth sequence $\{h_n\}$. Besides the theoretical concerns on the consistency and convergence rates, another practical issue one usually needs to address is the choice of the bandwidth parameter h_n, which is also called the *smoothing parameter*. It plays a crucial role in the bias-variance trade-off in kernel density estimation. In the literature, approaches to choosing the smoothing parameter include least-squares cross-validation (Bowman, 1984; Rudemo, 1982), biased cross-validation (Scott and Terrell, 1987), plug-in method (Park and Marron, 1990; Sheather and Jones, 1991), the double kernel method (Devroye, 1989), and also the method based on a discrepancy principle (Eggermont and LaRiccia, 2001). We refer the reader to Jones et al. (1996a) for a general overview and to Wand and Jones (1994); Cao et al. (1994); Jones et al. (1996b); Devroye (1997) for more detailed reviews.

Note that studies on the kernel density estimator (1) mentioned above heavily rely on the assumption that the observations are drawn in an i.i.d fashion. In the literature of statistics and machine learning, it is commonly accepted that the i.i.d assumption on the

given data can be very much restrictive in real-world applications. Having realized this, researchers turn to weaken this i.i.d assumption by assuming that the observations are weakly dependent under various notions of weakly dependence which include α-mixing, β-mixing, and ϕ-mixing (Bradley, 2005). There has been a flurry of work to attack this problem with theoretical and practical concerns, see e.g., Györfi (1981); Masry (1983, 1986); Robinson (1983); Masry and Györfi (1987); Tran (1989b); Györfi and Masry (1990); Tran (1989a); Hart and Vieu (1990); Yu (1993) and Hall et al. (1995), under the above notions of dependence. These studies were conducted under various notions of sample dependence. In fact, as Györfi and Lugosi (1992) pointed out, for samples that are ergodic, kernel density estimation is not universally consistent under the usual conditions. A counter example was devised there showing the existence of an ergodic sequence of uniformly distributed random variables based on which the kernel density estimation almost surely does not tend to zero in the L_1 sense. On the other hand, the assumed correlation among the observations complicates the kernel density estimation problem from a technical as well as practical view and also brings inherent barriers. This is because, more frequently, the analysis on the consistency and convergence rates of the kernel density estimator (1) is proceeded by decomposing the error term into bias and variance terms, which correspond to data-free and data-dependent error terms, respectively. The data-free error term can be tackled by using techniques from the approximation theory while the data-dependent error term is usually dealt with by exploiting arguments from the empirical process theory such as concentration inequalities. As a result, due to the existence of dependence among observations and various notions of the dependence measurement, the techniques, and results concerning the data-dependent error term are in general not universally applicable. On the other hand, it has been also pointed out that the bandwidth selection in kernel density estimation under dependence also departures from the independent case, see e.g., Hart and Vieu (1990); Hall et al. (1995).

In fact, when the observations $x_1, x_2, \ldots, x_n \in \mathbb{R}^d$ are generated by certain ergodic measure-preserving dynamical systems, the problem of kernel density estimation can be even more involved. To explain, let us consider a discrete-time ergodic measure-preserving dynamical system described by the sequence $(T^n)_{n\geq 1}$ of iterates of an unknown map $T : \Omega \to \Omega$ with $\Omega \subset \mathbb{R}^d$ and a unique invariant measure P which possesses a density f with respect to the Lebesgue measure (rigorous definitions will be given in the sequel). That is, we have

$$x_i = T^i(x_0), \quad i = 1, 2, \ldots, n, \tag{2}$$

where x_0 is the initial state. It is noticed that in this case the usual mixing concepts are not general enough to characterize the dependence among observations generated by (2) (Maume-Deschamps, 2006; Steinwart and Anghel, 2009; Hang and Steinwart, 2017). On the other hand, existing theoretical studies on the consistency or convergence rates of the kernel density estimator for i.i.d. observations frequently assume that the density function f is sufficiently smooth, e.g., first-order or even second-order smoothness. However, more often than not, this requirement can be stringent in the dynamical system context. It is well-known that (see e.g., Liverani (1995); Baladi (2000)) piecewise expanding maps (or Lasota-Yorke maps) admit a density f which only belongs to the space BV, i.e., functions

of bounded variation. Typical examples are the Gauss map in Example 3 and the β-maps in Example 1 (see Subsection 2.2).

In this study, the kernel density estimation problem with observations generated by dynamical systems (2) is approached by making use of a more general concept for measuring the dependence of observations, namely, the so-called $\mathcal{C}$-mixing process (refer to Section 2 for the definition). Proposed in Maume-Deschamps (2006) and recently investigated in Hang and Steinwart (2017), the $\mathcal{C}$-mixing concept is shown to be more general and powerful in measuring dependence among observations generated by dynamical systems and can accommodate a large class of dynamical systems. There, a Bernstein-type exponential inequality for $\mathcal{C}$-mixing processes was established and its applications to some learning schemes were explored.

Our main purpose in this paper is to conduct some theoretical analysis and practical implementations on the kernel density estimator for dynamical systems. The primary concern is the consistency and convergence rates of the kernel density estimator (1) with observations generated by dynamical systems (2). The consistency and convergence analysis is conducted under L_1-norm, and L_∞-norm, respectively. We show that under mild assumptions on the smoothing kernel, with properly chosen bandwidth, the estimator is universally consistent under L_1-norm. When the probability distribution P possesses a polynomial or exponential decay outside of a radius-r ball in its support, under the Hölder continuity assumptions on the kernel function and the density, we obtain almost optimal convergence rates under L_1-norm. Moreover, when the probability distribution P is compactly supported, which is a frequently encountered setting in the dynamical system context, we prove that stronger convergence results of the estimator can be developed, i.e., convergence results under L_∞-norm which are shown to be of the same order with its L_1-norm convergence rates. Finally, with regard to the practical implementation of the estimator, we also discuss the bandwidth selection problem by performing numerical comparisons among several typical existing selectors that include least squares cross-validation and its variants for dependent observations, and the double kernel method. We show that the double kernel bandwidth selector proposed in Devroye (1989) can in general work well. Moreover, according to our numerical experiments, we find that bandwidth selection for kernel density estimator of dynamical systems is usually ad-hoc in the sense that its performance may depend on the considered dynamical system.

The rest of this paper is organized as follows. Section 2 is a warm-up section for the introduction of some notations, definitions and assumptions that are related to the kernel density estimation problem and dynamical systems. We provide our main results on the consistency and convergence rates of the kernel density estimator in Section 3. Some comments and discussions concerning the main results will be also provided in this section. The main analysis on bounding error terms is presented in Section 4. We discuss the bandwidth selection problem in Section 5 and provide some numerical simulations. All the proofs of Section 3 and Section 4 can be found in Section 6. We end this paper in Section 7.

2. Preliminaries

2.1 Notations

Throughout this paper, λ^d is denoted as the Lebesgue measure on $\mathbb{R}^d$ and $\|\cdot\|$ is an arbitrary norm on $\mathbb{R}^d$. We denote B_r as the centered ball of $\mathbb{R}^d$ with radius r, that is,

$$B_r := \{x = (x_1, \ldots, x_d) \in \mathbb{R}^d : \|x\| \leq r\}.$$

Recall that for $1 \leq p < \infty$, the ℓ_p^d-norm is defined as $\|x\|_{\ell_p^d} := (x_1^p + \cdots + x_d^p)^{1/p}$, and the ℓ_∞^d-norm is defined as $\|x\|_{\ell_\infty^d} := \max_{i=1,\ldots,d} |x_i|$. Let $(\Omega, \mathcal{A}, \mu)$ be a probability space. We denote $L_p(\mu)$ as the space of (equivalence classes of) measurable functions $g : \Omega \to \mathbb{R}$ with finite L_p-norm $\|g\|_p$. Then $L_p(\mu)$ together with $\|g\|_p$ forms a Banach space. Moreover, if $\mathcal{A}' \subset \mathcal{A}$ is a sub-σ-algebra, then $L_p(\mathcal{A}', \mu)$ denotes the space of all $\mathcal{A}'$-measurable functions $g \in L_p(\mu)$. Finally, for a Banach space E, we write B_E for its closed unit ball.

In what follows, the notation $a_n \lesssim b_n$ means that there exists a positive constant c such that $a_n \leq c\, b_n$, for all $n \in \mathbb{N}$. With a slight abuse of notation, in this paper, c, c' and C are used interchangeably for positive constants while their values may vary across different lemmas, propositions, theorems, and corollaries.

2.2 Dynamical Systems and $\mathcal{C}$-mixing Processes

In this subsection, we first introduce the dynamical systems of interest, namely, ergodic measure-preserving dynamical systems. Mathematically, an *ergodic measure-preserving dynamical system* is a system $(\Omega, \mathcal{A}, \mu, T)$ with a mapping $T : \Omega \to \Omega$ that is measure-preserving, i.e., $\mu(A) = \mu(T^{-1}A)$ for all $A \in \mathcal{A}$, and ergodic, i.e., $T^{-1}A = A$ implies $\mu(A) = 0$ or 1. In this study, we are confined to the dynamical systems in which Ω is a subset of $\mathbb{R}^d$, μ is a probability measure that is absolutely continuous with respect to the Lebesgue measure λ^d and admits a unique invariant Lebesgue density f. In order to include a larger variety of density functions that commonly appear in dynamical systems, we introduce a new measurement of continuity that is a generalization of the α-Hölder continuity, which is defined as follows:

Definition 1 (Pointwise α-Hölder Controllable) *A density function $f : \mathbb{R}^d \to [0, \infty)$ is called pointwise α-Hölder controllable, if for λ^d-almost all $x \in \mathbb{R}^d$ there exists a constant $c(x) \geq 0$ and a radius $r(x) > 0$ such that for all $x' \in \mathbb{R}^d$ with $\|x'\| < r(x)$ we have*

$$|f(x + x') - f(x)| \leq c(x)\|x'\|^\alpha .$$

Moreover, f is called uniformly pointwise α-Hölder controllable, if

$$r_0 := \operatorname{ess\,inf}_{x \in \Omega} r(x) > 0 .$$

Note that the an α-Hölder continuous function can be recognized as a special case of the α-Hölder controllable functions with $c(x)$ and $r(x)$ being some universal constant $c > 0$ and $r > 0$.

In our study, it is assumed that the observations $x_1, x_2, \cdots, x_n$ are generated by the discrete-time dynamical system (2). Below we list several typical examples of discrete-time dynamical systems that satisfy the above assumptions (Lasota and Mackey, 1985):

Example 1 (β-Map) *For $\beta > 1$, the β-map is defined as*

$$T(x) = \beta x \mod 1, \quad x \in (0, 1),$$

with a unique invariant Lebesgue density given by

$$f(x) = c_\beta \sum_{i \geq 0} \beta^{-(i+1)} \mathbf{1}_{[0, T^i(1)]}(x),$$

where c_β is a constant chosen such that f has integral 1.

Example 2 (Logistic Map) *The Logistic map defined by*

$$T(x) = 4x(1 - x), \quad x \in (0, 1)$$

admits the unique invariant Lebesgue density

$$f(x) = \frac{1}{\pi\sqrt{x(1-x)}} \cdot \mathbf{1}_{(0,1)}(x), \quad x \in \mathbb{R}.$$

Moreover, for all $\alpha \in (0, 1/2)$, the density f is α-Hölder controllable with

$$c(x) := \begin{cases} 0 & \text{if } x < 0 \\ x^{-1/2-\alpha} & \text{if } 0 < x < 1/4 \\ 4 & \text{if } 1/4 \leq x \leq 3/4 \\ (1-x)^{-1/2-\alpha} & \text{if } 3/4 \leq x < 1 \\ 0 & \text{if } x > 1 . \end{cases}$$

and

$$r(x) := \begin{cases} -x & \text{if } x < 0 \\ x/2 & \text{if } 0 < x < 1/4 \\ 1/4 & \text{if } 1/4 \leq x \leq 3/4 \\ 1 - x/2 & \text{if } 3/4 \leq x < 1 \\ x - 1 & \text{if } x > 1 . \end{cases} \tag{3}$$

Example 3 (Gauss Map) *The Gauss map is defined by*

$$T(x) = \frac{1}{x} \mod 1, \quad x \in (0, 1),$$

with a unique invariant Lebesgue density

$$f(x) = \frac{1}{\log 2} \cdot \frac{1}{1+x} \cdot \mathbf{1}_{[0,1]}(x), \quad x \in \mathbb{R}.$$

Moreover, since the restriction $f_{|[0,1]}$ is Lipschitz continuous with Lipschitz constant $\frac{1}{\log 2}$, it is easy to check that f is pointwise 1-Hölder controllable with $r(x)$ given by (3) and $c(x) := \frac{1}{\log 2} \cdot \mathbf{1}_{(0,1)}(x)$.

We now introduce the notion for measuring the dependence among observations from dynamical systems, namely, $\mathcal{C}$-mixing process. To this end, let us assume that $(X, \mathcal{B})$ is a measurable space with $X \subset \mathbb{R}^d$. Let $\mathcal{X} := (X_n)_{n\geq 1}$ be an X-valued stochastic process on $(\Omega, \mathcal{A}, \mu)$, and for $1 \leq i \leq j \leq \infty$, denote by $\mathcal{A}_i^j$ the σ-algebra generated by $(X_i, \ldots, X_j)$. Let $\Gamma : \Omega \to X$ be a measurable map. μ_Γ is denoted as the Γ-image measure of μ, which is defined as $\mu_\Gamma(B) := \mu(\Gamma^{-1}(B))$, $B \subset X$ measurable. The process $\mathcal{X}$ is called *stationary* if $\mu_{(X_{i_1+j},\ldots,X_{i_n+j})} = \mu_{(X_{i_1},\ldots,X_{i_n})}$ for all $n, j, i_1, \ldots, i_n \geq 1$. Denote $P := \mu_{X_1}$. Moreover, for $\psi, \varphi \in L_1(\mu)$ satisfying $\psi\varphi \in L_1(\mu)$, we denote the correlation of ψ and φ by

$$\operatorname{cor}(\psi, \varphi) := \int_\Omega \psi\, \varphi \,\mathrm{d}\mu - \int_\Omega \psi \,\mathrm{d}\mu \cdot \int_\Omega \varphi \,\mathrm{d}\mu\,.$$

It is shown that several dependency coefficients for $\mathcal{X}$ can be expressed in terms of such correlations for restricted sets of functions ψ and φ. In order to introduce the notion, we also need to define a new norm which introduces restrictions on ψ and φ considered here. Throughout this paper, $\mathcal{C}(X)$ is denoted as a subspace of bounded measurable functions $g : X \to \mathbb{R}$ and that we have a semi-norm $|||\cdot|||$ on $\mathcal{C}(X)$. For $g \in \mathcal{C}(X)$, we define the $\mathcal{C}$-norm $\|\cdot\|_\mathcal{C}$ by

$$\|g\|_\mathcal{C} := \|g\|_\infty + |||g|||. \tag{4}$$

Additionally, we need to introduce the following restrictions on the semi-norm $|||\cdot|||$.

Assumption A *We assume that the following two restrictions on the semi-norm $|||\cdot|||$ hold:*

(i) $|||g||| = 0$ *for all constant functions* $g \in \mathcal{C}(X)$*;*

(ii) $|||e^g||| \leq \left\|e^g\right\|_\infty |||g|||, \qquad g \in \mathcal{C}(X).$

Note that the first constraint on the semi-norm in Assumption A implies its shift invariance on $\mathbb{R}$ while the inequality constraint can be viewed as an abstract *chain rule* if one views the semi-norm as a norm describing aspects of the smoothness of g. In fact, it is easy to show that the following function classes, which are probably also the most frequently considered in the dynamical system context, satisfy Condition (i) in Assumption A. Moreover, they also satisfy Condition (ii) in Assumption A, as shown in Hang and Steinwart (2017):

- $L_\infty(X)$: The class of bounded functions on X;
- $BV(X)$: The class of bounded variation functions on X;
- $C_{b,\alpha}(X)$: The class of bounded and α-Hölder continuous functions on X;
- $\operatorname{Lip}(X)$: The class of Lipschitz continuous functions on X;
- $C^1(X)$: The class of continuously differentiable functions on X.

The corresponding semi-norms are

$$\begin{aligned}
|||g|||_{L_\infty(X)} &:= 0,\\
|||g|||_{BV(X)} &:= \|g\|_{BV(X)},\\
|||g|||_{C_{b,\alpha}(X)} &:= |g|_\alpha := \sup_{x\neq x'} \frac{|g(x) - g(x')|}{|x - x'|^\alpha},\\
|||g|||_{\operatorname{Lip}(X)} &:= |g|_1,\\
|||g|||_{C^1(X)} &:= \sum_{i=1}^d \left\| \frac{\partial g}{\partial x_i} \right\|_\infty.
\end{aligned}$$

Throughout this paper, we assume that for all $r \geq 1$, there exists a function $g \in \mathcal{C}$ and a constant K such that $\mathbf{1}_{B^c_{r/2}} \leq g \leq 1$ and $|||g||| \leq K$. It is easily to see that this holds for function sets $\mathcal{C} = \operatorname{Lip}$, $C_{b,\alpha}$, C^1 etc.

Definition 2 ($\mathcal{C}$-mixing Process) *Let $(\Omega, \mathcal{A}, \mu)$ be a probability space, $(X, \mathcal{B})$ be a measurable space, $\mathcal{X} := (X_i)_{i\geq 1}$ be an X-valued, stationary process on Ω, and $\|\cdot\|_\mathcal{C}$ be defined by (4) for some semi-norm $|||\cdot|||$. Then, for $n \geq 1$, we define the $\mathcal{C}$-mixing coefficients by*

$$\phi_\mathcal{C}(\mathcal{X}, n) := \sup\left\{\operatorname{cor}(\psi, g(X_{k+n})) : k \geq 1, \psi \in B_{L_1(\mathcal{A}_1^k,\mu)}, g \in B_{\mathcal{C}(X)}\right\},$$

and the time-reversed $\mathcal{C}$-mixing coefficients by

$$\phi_{\mathcal{C},rev}(\mathcal{X}, n) := \sup\left\{\operatorname{cor}(g(X_k), \varphi) : k \geq 1, g \in B_{\mathcal{C}(X)}, \varphi \in B_{L_1(\mathcal{A}_{k+n}^\infty,\mu)}\right\}.$$

Let $(d_n)_{n\geq 1}$ be a strictly positive sequence converging to 0. We say that $\mathcal{X}$ ***is*** **(time-reversed) $\mathcal{C}$-mixing** *with rate $(d_n)_{n\geq 1}$, if we have $\phi_{\mathcal{C},rev}(\mathcal{X}, n) \leq d_n$ for all $n \geq 1$. Moreover, if $(d_n)_{n\geq 1}$ is of the form*

$$d_n := c_0 \exp\left(-bn^\gamma\right), \qquad n \geq 1,$$

for some constants $c_0 > 0$, $b > 0$, and $\gamma > 0$, then $\mathcal{X}$ is called **geometrically *(time-reversed) $\mathcal{C}$-mixing***.

Remark 3 *In Definition 2, if $|||\cdot||| = 0$, we obtain the classical ϕ-mixing coefficients. If $|||\cdot||| \neq 0$, the resulting $\mathcal{C}$-norm satisfies $\|\cdot\|_\mathcal{C} \geq \|\cdot\|_\infty$ and therefore, the mixing coefficients admit fewer functions. Thus, the considered functions must be "smoother" than the ones in the ϕ-mixing case and therefore statistical changes of small spatial nature in x do not have such a large impact on h, if h is smooth. In other words, even if the trajectory $x_1, \ldots, x_n$ stays in a certain region for a while, this does not impact the empirical average $\frac{1}{n}\sum_{i=1} h(x_i)$ as much as it would for non-smooth h. As a result, the concentration properties in this case hold similarly as in the i.i.d. case.*

From the above definition, we see that a $\mathcal{C}$-mixing process is defined in association with an underlying function space. For the above listed function spaces, i.e., $L_\infty(X)$, $BV(X)$, $C_{b,\alpha}(X)$, $\operatorname{Lip}(X)$, and $C^1(X)$, the increase of the smoothness enlarges the class of the

associated stochastic processes, as illustrated in Hang and Steinwart (2017). Note that the classical ϕ-mixing process is essentially a $\mathcal{C}$-mixing process associated with the function space $L_\infty(X)$. Note also that not all α-mixing processes are $\mathcal{C}$-mixing, and vice versa. We refer the reader to Hang and Steinwart (2017) for the relations among α-, ϕ- and $\mathcal{C}$-mixing processes.

On the other hand, under the above notations and definitions, from Theorem 4.7 in Maume-Deschamps (2006), we know that Logistic map in Example 2 is geometrically time-reversed $\mathcal{C}$-mixing with $\mathcal{C} = \mathrm{Lip}(0,1)$ while Theorem 4.4 in Maume-Deschamps (2006) (see also Chapter 3 in Baladi (2000)) indicates that Gauss map in Example 3 is geometrically time-reversed $\mathcal{C}$-mixing with $\mathcal{C} = BV(0,1)$. Example 1 is also geometrically time-reversed $\mathcal{C}$-mixing with $\mathcal{C} = BV(0,1)$ according to Maume-Deschamps (2006). For more examples of geometrically time-reversed $\mathcal{C}$-mixing dynamical systems, the reader is referred to Section 2 in Hang and Steinwart (2017). Besides, there also exist several high-dimensional examples. For instance, piecewise expanding maps (Baladi, 2000, Chapter 3), hyperbolic and Smale's Axiom A diffeomorphisms (Baladi, 2000, Chapter 4), and Anosov diffeomorphisms (Baladi, 2001; Lasota and Mackey, 1985).

2.3 Kernel Density Estimators for Dynamical Systems

In this subsection, we formulate kernel density estimators for dynamical systems that admit a unique underlying invariant Lebesgue density. The existence and uniqueness of an invariant measure (and the corrsponding invariant density) and smooth invariant measure is a classical problem in the theory of dynamical systems (Katok and Hasselblatt, 1995; Baladi, 2000; Lasota and Mackey, 1985). Perhaps the first existence theorem for continuous maps goes back to Krylov and Bogolyubov, see e.g. (Katok and Hasselblatt, 1995, Theorem 4.1.1). Then Lasota and Yorke (1973) proved the existence theorem for piecewise expanding maps. Since then, results concerning the existence for many other dynamical systems such as uniformly hyperbolic attractors and nonuniformly hyperbolic uni-modal maps have been established, see e.g. Baladi (2000). From the discussion after Theorem 2.1 in Baladi (2000) we know that mixing (thus ergodic) implies the uniqueness among all absolutely continuous invariant measures and smoothness ensures the existence of the invariant measure.

For the smoothing kernel K in the kernel density estimator, in this paper we consider its following general form, namely, d-dimensional smoothing kernel:

Definition 4 *A bounded, monotonically decreasing function $K : [0,\infty) \to [0,\infty)$ is a* ***d-dimensional smoothing kernel*** *if*

$$\int_{\mathbb{R}^d} K(\|x\|)\,\mathrm{d}x =: \kappa \in (0,\infty). \tag{5}$$

The choice of the norm in Definition 4 does not matter since all norms on $\mathbb{R}^d$ are equivalent. To see this, let $\|\cdot\|'$ be another norm on $\mathbb{R}^d$ satisfying $\kappa \in (0,\infty)$. From the equivalence of the two norms on $\mathbb{R}^d$, one can find a positive constant c such that $\|x\| \le c\|x\|'$ holds for all $x \in \mathbb{R}$. Therefore, easily we have

$$\int_{\mathbb{R}^d} K(\|x\|')\,\mathrm{d}x \le \int_{\mathbb{R}^d} K\left(\|x\|/c\right)\,\mathrm{d}x = c^d \int_{\mathbb{R}^d} K(\|x\|)\,\mathrm{d}x < \infty.$$

In what follows, without loss of generality, we assume that the constant κ in Definition 4 equals to 1.

Lemma 5 *A bounded, monotonically decreasing function $K : [0,\infty) \to [0,\infty)$ is a d-dimensional smoothing kernel if and only if*

$$\int_0^\infty K(r) r^{d-1}\,\mathrm{d}r \in (0,\infty).$$

Proof From the above discussions, it suffices to consider the integration constraint for the kernel function K with respect to the Euclidean norm $\|\cdot\|_{\ell_2^d}$. We thus have

$$\int_{\mathbb{R}^d} K\big(\|x\|_{\ell_2^d}\big)\,\mathrm{d}x = d\tau_d \int_0^\infty K(r) r^{d-1}\,\mathrm{d}r,$$

where $\tau_d = \pi^{d/2}/\Gamma\left(\frac{d}{2}+1\right)$ is the volume of the unit ball $B_{\ell_2^d}$ of the Euclidean space ℓ_2^d. This completes the proof of Lemma 5. ∎

Let $r \in [0,+\infty)$ and denote $\mathbf{1}_A$ as the indicator function. Several common examples of d-dimensional smoothing kernels $K(r)$ include the Naive kernel $\mathbf{1}_{[0,1]}(r)$, the Triangle kernel $(1-r)\mathbf{1}_{[0,1]}(r)$, the Epanechnikov kernel $(1-r^2)\mathbf{1}_{[0,1]}(r)$, and the Gaussian kernel e^{-r^2}. In this paper, we are interested in the kernels that satisfy the following restrictions on their shape and regularity:

Assumption B *For a fixed function space $\mathcal{C}(X)$, we make the following assumptions on the d-dimensional smoothing kernel K:*

(i) *K is Hölder continuous with exponent β with $\beta \in (0,1]$;*

(ii) *$\kappa_\upsilon := \int_0^\infty K(r) r^{\upsilon+d-1}\,\mathrm{d}r < \infty$ for some $\upsilon \in (0,\infty)$;*

(iii) *For some $R > 0$, K satisfies $K(r) = 0$ for all $r > R$;*

(iv) *For all $x \in \mathbb{R}^d$, we have $K(\|x - \cdot\|/h) \in \mathcal{C}(X)$ and there exists a function $\varphi : (0,\infty) \to (0,\infty)$ such that*

$$\sup_{x\in\mathbb{R}^d} |||K(\|x-\cdot\|/h)||| \le \varphi(h).$$

It is easy to verify that for $\mathcal{C} = \mathrm{Lip}$, Assumption B is met for the Triangle kernel, the Epanechnikov kernel, and the Gaussian kernel. Particularly, Condition (iv) holds for all these kernels with $|||\cdot|||$ being the Lipschitz norm and $\varphi(h) \le \mathcal{O}(h^{-1})$. Moreover, as we shall see below, not all the conditions in Assumption B are required for the analysis conducted in this study and conditions assumed on the kernel will be specified explicitly.

We now show that given a d-dimensional smoothing kernel K as in Definition 4, one can easily construct a probability density on $\mathbb{R}^d$.

Definition 6 (K-Smoothing of a Measure) *Let K be a d-dimensional smoothing kernel and Q be a probability measure on $\mathbb{R}^d$. Then, for $h > 0$,*

$$f_{Q,h}(x) := f_{Q,K,h}(x) := h^{-d} \int_{\mathbb{R}^d} K\left(\|x - x'\|/h\right) \mathrm{d}Q(x'), \qquad x \in \mathbb{R}^d,$$

*is called a K-**smoothing of** Q.*

It is not difficult to see that $f_{Q,h}$ defines a probability density on $\mathbb{R}^d$ since Fubini's theorem yields that

$$\begin{aligned}\int_{\mathbb{R}^d} f_{Q,h}(x)\,\mathrm{d}x &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} h^{-d} K\left(\|x - x'\|/h\right) \mathrm{d}Q(x')\,\mathrm{d}x \\ &= \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} K(\|x\|)\,\mathrm{d}x\,\mathrm{d}Q(x') = 1.\end{aligned}$$

Let us denote $K_h : \mathbb{R}^d \to [0, +\infty)$ as

$$K_h(x) := h^{-d} K\left(\|x\|/h\right),\ x \in \mathbb{R}^d. \tag{6}$$

Note that K_h also induces a density function on $\mathbb{R}^d$ since there holds $\|K_h\|_1 = 1$.

For the sake of notational simplification, in what follows, we introduce the convolution operator $*$. Under this notation, we then see that $f_{Q,h}$ is the density of the measure that is the convolution of the measure Q and $\nu_h = K_h\,\mathrm{d}\lambda^d$. Recalling that P is a probability measure on $\mathbb{R}^d$ with the corresponding density function f, by taking $Q := P$ with $\mathrm{d}P = f\,\mathrm{d}\lambda^d$, we have

$$f_{P,h} = K_h * f = f * K_h = K_h * \mathrm{d}P. \tag{7}$$

Since $K_h \in L_\infty(\mathbb{R}^d)$ and $f \in L_1(\mathbb{R}^d)$, from Proposition 8.8 in Folland (1999), we know that $f_{P,h}$ is uniformly continuous and bounded. Specifically, when Q is the empirical measure $D_n = \frac{1}{n}\sum_{i=1}^n \delta_{x_i}$, the *kernel density estimator for dynamical systems* in this study can be expressed as

$$f_{D_n,h}(x) = K_h * \mathrm{d}D_n(x) = n^{-1}h^{-d}\sum_{i=1}^n K\left(\|x - x_i\|/h\right). \tag{8}$$

From now on, for notational simplicity, we will suppress the subscript n of D_n and denote $D := D_n$, e.g., $f_{D,h} := f_{D_n,h}$.

3. Main Results and Statements

In this section, we present main results on the consistency and convergence rates of $f_{D,h}$ to the true density f under L_1-norm and also L_∞-norm for some special cases. We also present some comments and discussions on the obtained main results.

Recall that $f_{D,h}$ is a nonparametric density estimator and so the criterion that measures its goodness-of-fit matters, which, for instance, includes L_1-distance, L_2-distance, and L_∞-distance. In the literature of kernel density estimation, probably the most frequently

employed criterion is the L_2-distance of the difference between $f_{D,h}$ and f, since it entails an exact bias-variance decomposition and can be analyzed relatively easily by using Taylor expansion involved arguments. However, it is argued in Devroye and Györfi (1985) (see also Devroye and Lugosi (2001)) that L_1-distance could be a more reasonable choice since: it is invariant under monotone transformations; it is always well-defined as a metric on the space of density functions; it is also proportional to the total variation metric and so leads to better visualization of the closeness to the true density function than L_2-distance. As to the L_∞-distance, it measures the worst-case goodness-of-fit of the estimator.

3.1 Results on Consistency

We first present results on the consistency property of the kernel density estimator $f_{D,h}$ in the sense of L_1-norm. A kernel density estimator $f_{D,h}$ is said to be *consistent* in the sense of L_1-norm if $f_{D,h}$ converges to f almost surely under L_1-norm.

Theorem 7 *Let K be a d-dimensional smoothing kernel that satisfies Conditions (i) and (iv) in Assumption B. Let $\mathcal{X} := (X_n)_{n\geq 1}$ be an X-valued stationary geometrically (time-reversed) $\mathcal{C}$-mixing process on $(\Omega, \mathcal{A}, \mu)$ with $\|\cdot\|_{\mathcal{C}}$ being defined for some semi-norm $|||\cdot|||$ that satisfies Assumption A. If*

$$h_n \to 0 \quad \textit{and} \quad \frac{nh_n^d}{(\log n)^{(2+\gamma)/\gamma}} \to \infty, \quad \textit{as} \quad n \to \infty,$$

then the kernel density estimator f_{D,h_n} is universally consistent in the sense of L_1-norm.

The consistency result in Theorem 7 is independent of the probability distribution P and is therefore of the universal type. In particular, these results apply to the examples provided in Section 2.2, i.e., β-map, Logistic map, and Gauss map.

3.2 Results on Convergence Rates under L_1-Norm

We now show that if certain tail assumptions on P are imposed, convergence rates can be obtained under L_1-norm. Here, we consider three different situations, namely, the tail of the probability distribution P has a polynomial decay, exponential decay and disappears, respectively.

Theorem 8 *Let K be a d-dimensional smoothing kernel that satisfies Assumption B. Assume that the density f is α-Hölder continuous with $\alpha \leq \beta$. Let $\mathcal{X} := (X_n)_{n\geq 1}$ be an X-valued stationary geometrically (time-reversed) $\mathcal{C}$-mixing process on $(\Omega, \mathcal{A}, \mu)$ with $\|\cdot\|_{\mathcal{C}}$ being defined for some semi-norm $|||\cdot|||$ that satisfies Assumption A. We consider the following cases:*

(i) $P\left(B_r^c\right) \lesssim r^{-\eta d}$ *for some $\eta > 0$ and for all $r \geq 1$;*

(ii) $P\left(B_r^c\right) \lesssim e^{-ar^\eta}$ *for some $a > 0$, $\eta > 0$ and for all $r \geq 1$;*

(iii) $P\left(B_{r_0}^c\right) = 0$ *for some $r_0 \geq 1$.*

For the above cases, if $n \geq n_0$ with n_0 given in the proof, and the sequences h_n are of the following forms:

(i) $h_n = \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{1+\eta}{(1+\eta)(2\alpha+d)-\alpha}}$;

(ii) $h_n = \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{1}{2\alpha+d}} (\log n)^{-\frac{d}{\gamma}\cdot\frac{1}{2\alpha+d}}$;

(iii) $h_n = \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2\alpha+d}}$;

then with probability μ *at least* $1-\frac{1}{n}$*, there holds*

$$\|f_{D,h_n} - f\|_1 \le \varepsilon_n,$$

where the convergence rates

(i) $\varepsilon_n \lesssim \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{\alpha\eta}{(1+\eta)(2\alpha+d)-\alpha}}$;

(ii) $\varepsilon_n \lesssim \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{\alpha}{2\alpha+d}} (\log n)^{\frac{d}{\gamma}\cdot\frac{\alpha+d}{2\alpha+d}}$;

(iii) $\varepsilon_n \lesssim \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{\alpha}{2\alpha+d}}$.

Notice that the above Theorem 8 holds only when the underlying density function f is α-Hölder continuous. However, as shown in Examples 2 and 3, instead of being α-Hölder continuous, density functions of commonly used dynamical systems often only satisfy a weaker continuity condition such as the pointwise α-Hölder controllable condition. Therefore, for this case, we are encouraged to establish convergence rates under certain tail condition of the probability distribution. Unfortunately, as will be shown in Proposition 11 later, in general, we are not able to give explicit expressions of the convergence rates as in Theorem 8. Nevertheless, for certain dynamical systems mentioned in this paper, we are still able to derive convergence rates explicitly as follows:

Example 4 *Consider the Gauss map from Example 3 with the resulting density*

$$f(x) = \frac{1}{\log 2}\cdot\frac{1}{1+x}, \quad x \in (0,1)$$

and functions $r(\cdot)$ *and* $c(\cdot)$ *as specified in Example 3. Then, for* $\Omega := (0,1)$*, if we pick a smoothing kernel with* $K(r) = 0$ *for all* $r > 1$ *and* $h_n = \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{1/(2+d)}$*, then we obtain the convergence rate*

$$\left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2+d}}.$$

Example 5 *Consider the logistic map of Example 2 with resulting density*

$$f(x) = \frac{1}{\pi\sqrt{x(1-x)}}, \quad x \in (0,1),$$

an $\alpha \in (0, 1/2)$ *and the corresponding functions* $r(\cdot)$ *and* $c(\cdot)$ *specified in Example 2. As in Example 4, we choose* $\Omega := (0,1)$*, and pick a smoothing kernel with* $K(r) = 0$ *for all* $r > 1$*, then we obtain approximately the convergence rate*

$$\left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2+2d}}.$$

Both of the above-mentioned two densities are not continuous at the end points 0 and 1, but $f_{P,h}$ turns out to be continuous everywhere. Therefore, uniform approximation is not be possible. However, as shown in the above examples, the uniform approximation can be achieved if we remove both of the neighbourhood around the critical points 0 and 1. This phenomenon can be apparently observed from Figures 1 and 2 in Section 5.

3.3 Results on Convergence Rates under L_∞-Norm

We now state our main results on the convergence of $f_{D,h}$ to f under L_∞-norm.

Theorem 9 *Let* $\mathcal{X} := (X_n)_{n\ge1}$ *be an* X*-valued stationary geometrically (time-reversed)* $\mathcal{C}$*-mixing process on* $(\Omega, \mathcal{A}, \mu)$ *with* $\|\cdot\|_{\mathcal{C}}$ *being defined for some semi-norm* $|||\cdot|||$ *that satisfies Assumption A.*

(i) Let K *be a* d*-dimensional smoothing kernel that satisfies Conditions* (i) *and* (iv) *in Assumption B. Assume that there exists a constant* $r_0 \ge 1$ *such that* $\Omega \subset B_{r_0} \subset \mathbb{R}^d$ *and the density function* f *is* α*-Hölder continuous with* $\alpha \le \beta$ *and* $\|f\|_\infty < \infty$*.*

(ii) Let K *be a* d*-dimensional smoothing kernel that satisfies Conditions* (iii) *and* (iv) *in Assumption B, and* f *is pointwise* α*-Hölder controllable. Fix an* $\Omega \subset \mathbb{R}^d$ *with* $\{x \in \mathbb{R}^d : f(x) > 0 \text{ and } r(x) \text{ exists}\} \subset \Omega$ *and define* $\Omega^{+hR} := \{x \in \mathbb{R}^d : \inf_{x''\in\Omega}\|x - x''\| \le hR\}$*. Moreover, we define* $X_h^* := \{x \in \mathbb{R}^d : r(x) > hR\}$ *and assume that function* $x \mapsto c(x)$ *is bounded on* $X_h^* \cap \Omega^{+hR}$*.*

Then, for both cases (i) *and* (ii)*, all* $n \ge n_0^*$ *with* n_0^* *that will be given in the proof, by choosing*

$$h_n = \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2\alpha+d}},$$

with probability μ *at least* $1-\frac{1}{n}$*, there holds*

$$\|f_{D,h_n} - f\|_\infty \lesssim \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{\alpha}{2\alpha+d}}. \tag{9}$$

In Theorems 8 and 9, one needs to ensure that $n \ge n_0$ with n_0 and $n \ge n_0^*$ being specified later. One may also note that due to the involvement of the term $\varphi(h_n)$, the numbers n_0 and n_0^* depend on the h_n. However, recalling that for the Triangle kernel, the Epanechnikov kernel, and the Gaussian kernel, we have $\varphi(h_n) \le \mathcal{O}(h_n^{-1})$, which, together with the choices of h_n in Theorems 8 and 9, implies that n_0 and n_0^* are well-defined. It should be also remarked that in the scenario where the density function f is compactly supported and bounded, the convergence rate of $f_{D,h}$ to f is not only obtainable, but also the same with that derived under L_1-norm. This is indeed an interesting observation since convergence under L_∞-norm implies convergence under L_1-norm.

3.4 Comments and Discussions

This section presents some comments on the obtained theoretical results on the consistency and convergence rates of $f_{D,h}$ and compares them with related findings in the literature.

We highlight that in our analysis the density function f is only assumed to be Hölder continuous. As pointed out in the introduction, in the context of dynamical systems, this seems to be more than a reasonable assumption. On the other hand, the consistency and the convergence results obtained in our study, are of type "with high probability" due to the use of the Bernstein-type exponential inequality that takes into account the variance information of the random variables. From our analysis and the obtained theoretical results, one can also easily observe the influence of the dependence among observations. For instance, from Theorem 7 we see that with increasing dependence among observations (corresponding to smaller γ), in order to ensure the universal consistency of f_{D,h_n}, the decay of h_n (with respect to n^{-1}) is required to be faster. This is in fact also the case if we look at results on the convergence rates in Theorems 8 and 9. Moreover, the influence of the dependence among observations is also indicated there. That is, an increase of the dependence among observations may slow down the convergence of $f_{D,h}$ in the sense of both L_1-norm and L_∞-norm. It is also interesting to note that when γ tends to infinity, which corresponds to the case where observations can be roughly treated as independent ones, meaningful convergence rates can be also deduced. It turns out that, up to a logarithmic factor, the established convergence rates (9) under L_∞-norm, namely, $\mathcal{O}(((\log n)^{(2+\gamma)/\gamma}/n)^{\alpha/(2\alpha+d)})$, match the optimal rates in the i.i.d. case, see, e.g., Khas′minskii (1979) and Stone (1983).

As mentioned in the introduction, there exist several studies in the literature that address the kernel density estimation problem for dynamical systems. For example, Bosq and Guégan (1995) conducted some first studies and showed the point-wise consistency and the convergence (in expectation) of the kernel density estimator. The convergence rates obtained in their study are of the type $\mathcal{O}(n^{-4/(4+2d)})$, which are conducted in terms of the variance of $f_{D,h}$. The notion they used for measuring the dependence among observations is α-mixing coefficient (see A_3 in Bosq and Guégan (1995)). Considering the density estimation problem for one-dimensional dynamical systems, Prieur (2001) presented some studies on the kernel density estimator $f_{D,h}$ by developing a central limit theorem and apply it to bound the variance of the estimator. Further some studies on the kernel density estimation of the invariant Lebesgue density for dynamical systems were conducted in Blanke et al. (2003). By considering both dynamical noise and observational noise, point-wise convergence of the estimator $f_{D,h}$ in expectation was established, i.e., the convergence of $\mathbb{E}f_{D,h}(x) - f(x)$ for any $x \in \mathbb{R}^d$. Note further that these results rely on the second-order smoothness and boundedness of f. Therefore, the second-order smoothness assumption on the density function together with the point-wise convergence in expectation makes it different from our work. In particular, under the additional assumption on the tail of the noise distribution, the convergence of $\mathbb{E}(f_{D,h}(x) - f(x))^2$ for any fixed $x \in \mathbb{R}^d$ is of the order $\mathcal{O}(n^{-2/(2+\beta d)})$ with $\beta \geq 1$. Concerning the convergence of $f_{D,h}$ in a dynamical system setup, Maume-Deschamps (2006) also presented some interesting studies which in some sense also motivated our work here. By using also the $\mathcal{C}$-mixing concept as adopted in our study to measure the dependence among observations from dynamical systems, she presented the point-wise convergence of $f_{D,h}$ with the help of Hoeffding-type exponential inequality (see Proposition 3.1 in Maume-Deschamps (2006)). The assumption applied on f is that it is bounded from below and also α-Hölder continuous (more precisely, f is assumed to be α-*regular*, see Assumption 2.3 in Maume-Deschamps (2006)). Hence, from the above discussions, we suggest that the work we present in this study is essentially different from that in Maume-Deschamps (2006).

4. Error Analysis

We conduct error analysis for the kernel density estimator $f_{D,h}$ in this section by establishing its consistency and convergence rates, which are stated in the above section in terms of the L_1-distance and L_∞-distance. The downside of using L_1-distance is that it does not admit an exact bias-variance decomposition and the usual Taylor expansion involved techniques for error estimation may not apply directly. Nonetheless, if we introduce the intermediate estimator $f_{P,h}$ in (7), obviously the following inequality holds

$$\|f_{D,h} - f\|_1 \leq \|f_{D,h} - f_{P,h}\|_1 + \|f_{P,h} - f\|_1. \tag{10}$$

The consistency and convergence analysis in our study will be mainly conducted in the L_1 sense with the help of inequality (10). Besides, for some specific case, i.e., when the density f is compactly supported, we are also concerned with the consistency and convergence of $f_{D,h}$ to f under L_∞-norm. In this case, there also holds the following inequality

$$\|f_{D,h} - f\|_\infty \leq \|f_{D,h} - f_{P,h}\|_\infty + \|f_{P,h} - f\|_\infty. \tag{11}$$

It is easy to see that the first error term on the right-hand side of (10) or (11) is stochastic due to the empirical measure D while the second one is deterministic because of its sampling-free nature. Loosely speaking, the first error term corresponds to the variance of the estimator $f_{D,h}$, while the second one can be treated as its bias although (10) or (11) is not an exact error decomposition. In our study, we proceed with the consistency and convergence analysis on $f_{D,h}$ by bounding the two error terms, respectively.

4.1 Bounding the Deterministic Error Term

Our first theoretical result on bounding the deterministic error term shows that, given a d-dimensional kernel K, the L_1-distance between its K-smooth of the measure P, i.e., $f_{P,h}$, and f can be arbitrarily small by choosing the bandwidth appropriately. Moreover, under mild assumptions on the regularity of f and K, the L_∞-distance between the two quantities possesses a polynomial decay with respect to the bandwidth h.

Proposition 10 *Let K be a d-dimensional smoothing kernel.*

(i) For any $\varepsilon > 0$, there exists $0 < h_\varepsilon \leq 1$ such that for any $h \in (0, h_\varepsilon]$ we have

$$\|f_{P,h} - f\|_1 \leq \varepsilon.$$

(ii) If K satisfies Condition (ii) in Assumption B and f is α-Hölder continuous with $\alpha \leq \upsilon$, then there exists a constant $c > 0$ such that for all $h > 0$ we have

$$\|f_{P,h} - f\|_\infty \leq ch^\alpha.$$

(iii) If K satisfies Condition (ii) in Assumption B and f is uniformly pointwise α-Hölder controllable with $\alpha \leq \upsilon$, then there exists a constant $c > 0$ such that for all $h > 0$ and λ^d-almost all $x \in \mathbb{R}^d$ we have

$$|f_{P,h}(x) - f(x)| \leq c\, c(x) h^{\alpha} + c f(x) h^{\upsilon} + \int_{B^c_{r_0/(2h)}} K(\|x'\|)\, f(x + hx')\, \mathrm{d}x' .$$

(iv) Assume that K satisfies Condition (iii) in Assumption B and f is pointwise α-Hölder controllable. We fix an $\Omega \subset \mathbb{R}^d$ such that

$$\{x \in \mathbb{R}^d : f(x) > 0 \text{ and } r(x) \text{ exists } \} \subset \Omega ,$$

and we define $\Omega^{+hR} := \{x \in \mathbb{R}^d : \inf_{x'' \in \Omega} \|x - x''\| \leq hR\}$. For λ^d-almost all $x \notin \Omega^{+hR}$ we then have $|f_{P,h}(x) - f(x)| = 0$. Moreover, there exists a constant $c > 0$ such that for all $x \in \Omega^{+hR}$ for which $r(x)$ exists we have

$$|f_{P,h}(x) - f(x)| \leq c\, c(x) h^{\alpha} + c \cdot \|K\|_{\infty} f(x) \left| R - \frac{r(x)}{h} \right|_+ + \|K\|_{\infty} h^{-d} \int_{r(x) \leq \|x'\| \leq hR} f(x + x')\, \mathrm{d}x' .$$

We now show that the L_1-distance between $f_{P,h}$ and f can be upper bounded by their difference (in the sense of L_∞-distance) on a compact domain of $\mathbb{R}^d$ together with their difference (in the sense of L_1-distance) outside this domain. As we shall see later, this observation will entail us to consider different classes of the true density f. The following result is crucial in our subsequent analysis on the consistency and convergence rates of $f_{D,h}$.

Proposition 11 *Assume that K is a d-dimensional smoothing kernel that satisfies Condition (ii) in Assumption B.*

(i) There exists a constant $c_1 > 0$ such that for all $h \leq 1$ and $r \geq 1$, we have

$$\|f_{P,h} - f\|_1 \leq c_1 r^d \|f_{P,h} - f\|_{\infty} + c_1 P(B^c_{r/2}) + c_1 (h/r)^{\upsilon}.$$

(ii) Assume that f is uniformly pointwise α-Hölder controllable with $\alpha \leq \upsilon$. For $r > 0$ we define

$$H(r) := \int_{B_r} c(x)\, \mathrm{d}x.$$

Then there exists a constant $c_2 > 0$ such that for all $h > 0$ and $r \geq 1$ we have

$$\|f_{P,h} - f\|_1 \leq c_2 h^{\alpha} H(r) + c_2 P(B^c_{r/2}) + c_2 h^{\upsilon} .$$

(iii) Assume that K satisfies Condition (iii) in Assumption B and f is pointwise α-Hölder controllable. Furthermore, we pick an Ω as in Proposition 10. Then there exists a constant $c_3 > 0$ such that for all $h > 0$ and $r > 0$ we have

$$\|f_{P,h} - f\|_1 \leq c_3\, H(\infty)\, h^{\alpha} + c_3 P\big(\{x : r(x) \leq hR\}\big) + c_3\, h^{-d} \int_{\Omega^{+hR}} \int_{r(x) \leq \|x'\| \leq hR} f(x + x')\, \mathrm{d}x'\, \mathrm{d}x .$$

(iv) Let K, f, and Ω satisfy the conditions in (iii). We define $X^*_h := \{x \in \mathbb{R}^d : r(x) > hR\}$ and assume that function $x \mapsto c(x)$ is bounded on $X^*_h \cap \Omega^{hR}$, then there is another constant $c_4 > 0$ such that for all $h > 0$ we have

$$\sup_{x \in X^*_h} |f_{P,h}(x) - f(x)| \leq c_4 \sup_{x \in X^*_h \cap \Omega^{hR}} |c(x)| \cdot h^{\alpha} .$$

4.2 Bounding the Stochastic Error Term

We now proceed with the estimation of the stochastic error term $\|f_{D,h} - f_{P,h}\|_1$ by establishing probabilistic oracle inequalities. For the sake of readability, let us start with an overview of the analysis conducted in this subsection for bounding the stochastic error term.

4.2.1 An Overview of the Analysis

In this study, the stochastic error term is tackled by using capacity-involved arguments and the Bernstein-type inequality established in Hang and Steinwart (2017). In the sequel, for any fixed $x \in \Omega \subset \mathbb{R}^d$, we write

$$k_{x,h} := h^{-d} K(\|x - \cdot\|/h), \tag{12}$$

and we further denote the centered random variable $\widetilde{k}_{x,h}$ on Ω as

$$\widetilde{k}_{x,h} := k_{x,h} - \mathbb{E}_P k_{x,h}. \tag{13}$$

It thus follows that

$$\mathbb{E}_D \widetilde{k}_{x,h} = \mathbb{E}_D k_{x,h} - \mathbb{E}_P k_{x,h} = f_{D,h}(x) - f_{P,h}(x),$$

and consequently we have

$$\|f_{D,h} - f_{P,h}\|_1 = \int_{\mathbb{R}^d} |\mathbb{E}_D \widetilde{k}_{x,h}|\, \mathrm{d}x,$$

and

$$\|f_{D,h} - f_{P,h}\|_{\infty} = \sup_{x \in \Omega} |\mathbb{E}_D \widetilde{k}_{x,h}|.$$

As a result, in order to bound $\|f_{D,h} - f_{P,h}\|_1$, it suffices to bound the supremum of the empirical process $\mathbb{E}_D \widetilde{k}_{x,h}$ indexed by $x \in \mathbb{R}^d$. For any $r > 0$, there holds

$$\|f_{D,h} - f_{P,h}\|_1 = \int_{B_r} |\mathbb{E}_D \widetilde{k}_{x,h}|\, \mathrm{d}x + \int_{B^c_r} |\mathbb{E}_D \widetilde{k}_{x,h}|\, \mathrm{d}x.$$

The second term of the right-hand side of the above equality can be similarly dealt with as in the proof of Proposition 11. In order to bound the first term, we define $\widetilde{\mathcal{K}}_{h,r}$ as the function set of $\widetilde{k}_{x,h}$ that corresponds to x which lies on a radius-r ball of $\mathbb{R}^d$:

$$\widetilde{\mathcal{K}}_{h,r} := \big\{\widetilde{k}_{x,h} : x \in B_r\big\} \subset L_\infty(\mathbb{R}^d).$$

The idea here is to apply capacity-involved arguments and the Bernstein-type exponential inequality in Hang and Steinwart (2017) to the function set $\widetilde{\mathcal{K}}_{h,r}$ and the associated empirical process $\mathbb{E}_D \widetilde{k}_{x,h}$. The difference between $f_{D,h}$ and $f_{P,h}$ under the L_∞-norm can be bounded analogously. Therefore, to further our analysis, we first need to bound the capacity of $\widetilde{\mathcal{K}}_{h,r}$ in terms of covering numbers.

4.2.2 Bounding the Capacity of the Function Set $\widetilde{\mathcal{K}}_{h,r}$

Definition 12 (Covering Number) *Let (X,d) be a metric space and $A \subset X$. For $\varepsilon > 0$, the ε-**covering number** of A is denoted as*

$$\mathcal{N}(A,d,\varepsilon) := \min\left\{ n \geq 1 : \exists\, x_1, \cdots, x_n \in X \text{ such that } A \subset \bigcup_{i=1}^{n} B_d(x_i,\varepsilon) \right\},$$

where $B_d(x,\varepsilon) := \{x' \in X : d(x,x') \leq \varepsilon\}$.

For any fixed $r \geq 1$, we consider the function set

$$\mathcal{K}_{h,r} := \{k_{x,h} : x \in B_r\} \subset L_\infty(\mathbb{R}^d).$$

The following proposition provides an estimate of the covering number of $\mathcal{K}_{h,r}$.

Proposition 13 *Let K be a d-dimensional smoothing kernel that satisfies Condition (i) in Assumption B and $h \in (0,1]$. Then there exists a positive constant c' such that for all $\varepsilon \in (0,1]$, we have*

$$\mathcal{N}(\mathcal{K}_{h,r}, \|\cdot\|_\infty, \varepsilon) \leq c' r^d h^{-d-\frac{d^2}{\beta}} \varepsilon^{-\frac{d}{\beta}}.$$

4.2.3 Oracle Inequalities under L_1-Norm, and L_∞-Norm

We now establish oracle inequalities for the kernel density estimator (8) under L_1-norm, and L_∞-norm, respectively. These oracle inequalities will be crucial in establishing the consistency and convergence results of the estimator. Recall that the considered kernel density estimation problem is based on samples from an X-valued $\mathcal{C}$-mixing process which is associated with an underlying function class $\mathcal{C}(X)$. As shown below, the established oracle inequality holds without further restrictions on the support of the density function.

Proposition 14 *Suppose that Assumption B holds. Let $\mathcal{X} := (X_n)_{n\geq 1}$ be an X-valued stationary geometrically (time-reversed) $\mathcal{C}$-mixing process on $(\Omega,\mathcal{A},\mu)$ with $\|\cdot\|_{\mathcal{C}}$ being defined for some semi-norm $|||\cdot|||$ that satisfies Assumption A. Then for all $0 < h \leq 1$, $r \geq 1$ and $\tau \geq 1$, there exists an $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$, with probability μ at least $1 - 3e^{-\tau}$, there holds*

$$\|f_{D,h} - f_{P,h}\|_1 \lesssim \sqrt{\frac{(\log n)^{2/\gamma} r^d (\tau + \log \frac{nr}{h})}{h^d n}} + \frac{(\log n)^{2/\gamma} r^d (\tau + \log \frac{nr}{h})}{h^d n} + P(B_{r/4}^c) + \sqrt{\frac{32\tau(\log n)^{2/\gamma}}{n}} + \left(\frac{h}{r}\right)^\beta.$$

Here n_0 will be given explicitly in the proof.

Our next result shows that when the density function f is compactly supported and bounded, an oracle inequality under L_∞-norm can be also derived.

Proposition 15 *Let K be a d-dimensional kernel function that satisfies Conditions (i) and (iii) in Assumption B. Let $\mathcal{X} := (X_n)_{n\geq 1}$ be an X-valued stationary geometrically (time-reversed) $\mathcal{C}$-mixing process on $(\Omega,\mathcal{A},\mu)$ with $\|\cdot\|_{\mathcal{C}}$ being defined for some semi-norm $|||\cdot|||$ that satisfies Assumption A. Assume that there exists a constant $r_0 \geq 1$ such that $\Omega \subset B_{r_0} \subset \mathbb{R}^d$ and the density function f satisfies $\|f\|_\infty < \infty$. Then for all $0 < h \leq 1$ and $\tau > 0$, there exists an $n_0^* \in \mathbb{N}$ such that for all $n \geq n_0^*$, with probability μ at least $1 - e^{-\tau}$, there holds*

$$\|f_{D,h} - f_{P,h}\|_\infty \lesssim \sqrt{\frac{\|f\|_\infty (\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)(\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n}.$$

Here n_0^* will be given explicitly in the proof.

In Proposition 15, the kernel K is only required to satisfy Conditions (i) and (iii) in Assumption B whereas the condition that $\int_0^\infty K(r) r^{\beta+d-1}\,\mathrm{d}r < \infty$ for some $\beta > 0$ is not needed. This is again due to the compact support assumption of the density function f as stated in Proposition 15.

5. Bandwidth Selection and Simulation Studies

This section discusses the model selection problem of the kernel density estimator (8) by performing numerical simulation studies. In the context of kernel density estimation, model selection is mainly referred to the choice of the smoothing kernel K and the selection of the kernel bandwidth h, which are of crucial importance for the practical implementation of the data-driven density estimator. According to our experimental experience and also the empirical observations reported in Maume-Deschamps (2006), it seems that the choice of the kernel or the noise does not have a significant influence on the performance of the estimator. Therefore, our emphasis will be placed on the bandwidth selection problem in our simulation studies.

5.1 Several Bandwidth Selectors

In the literature of kernel density estimation, various bandwidth selectors have been proposed, several typical examples of which have been alluded to in the introduction. When turning to the case with dependent observations, the bandwidth selection problem has been also drawing much attention, see e.g., Hart and Vieu (1990); Chu and Marron (1991); Hall et al. (1995); Yao and Tong (1998). Among existing bandwidth selectors, probably the most frequently employed ones are based on the cross-validation ideas. For cross-validation bandwidth selectors, one tries to minimize the integrated squared error (ISE) of the empirical estimator $f_{D,h}$ where

$$\mathrm{ISE}(h) := \int (f_{D,h} - f)^2 = \int f_{D,h}^2 - 2\int f_{D,h} \cdot \int f + \int f^2.$$

Note that on the right-hand side of the above equality, the last term $\int f^2$ is independent of h and so the minimization of $\mathrm{ISE}(h)$ is equivalent to minimize

$$\int f_{D,h}^2 - 2\int f_{D,h} \cdot \int f.$$

It is shown that with i.i.d observations, an unbiased estimator of the above quantity, which is termed as least squares cross-validation (LSCV), is given as follows:

$$\mathrm{LSCV}(h) := \int f_{D,h}^2 - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i,h}(x_i), \tag{14}$$

where the leave-one-out density estimator $\hat{f}_{-i,h}$ is defined as

$$\hat{f}_{-i,h}(x) := \frac{1}{n-1}\sum_{j\neq i}^{n} K_h(x - x_j).$$

When the observations are dependent, it is shown that cross-validation can produce much under-smoothed estimates, see e.g., Hart and Wehrly (1986); Hart and Vieu (1990). Observing this, Hart and Vieu (1990) proposed the modified least squares cross-validation (MLSCV), which is defined as follows

$$\mathrm{MLSCV}(h) := \int f_{D,h}^2 - \frac{2}{n}\sum_{i=1}^{n} \hat{f}_{-i,h,l_n}(x_i), \tag{15}$$

where l_n is set to 1 or 2 as suggested in Hart and Vieu (1990) and

$$\hat{f}_{-i,h,l_n}(x) := \frac{1}{\#\{j : |j-i| > l_n\}} \sum_{|j-i|>l_n} K_h(x - x_j).$$

The underlying intuition of proposing MLSCV is that when estimating the density of a fixed point, ignoring observations in the vicinity of this point may be help in reducing the influence of dependence among observations. However, when turning to the L_1 point of view, the above bandwidth selectors may not work well due to the use of the least squares criterion. Alternatively, Devroye (1989) proposed the double kernel bandwidth selector that minimizes the following quantity

$$\mathrm{DKM}(h) := \int |f_{D,h,K} - f_{D,h,L}|, \tag{16}$$

where $f_{D,h,K}$ and $f_{D,h,L}$ are kernel density estimators based on the kernels K and L, respectively. Some rigorous theoretical treatments on the effectiveness of the above bandwidth selector were made in Devroye (1989).

Our purpose in simulation studies is to conduct empirical comparisons among the above bandwidth selectors in the dynamical system context instead of proposing new approaches.

5.2 Experimental Setup

In our experiments, observations $x_1, \cdots, x_n$ are generated from the following model[1]

$$\begin{cases} \tilde{x}_i = T^i(x_0), \\ x_i = \tilde{x}_i + \varepsilon_i, \end{cases} \quad i = 1, \cdots, n, \tag{17}$$

1. Note that here the observational noise is assumed for the considered dynamical system (17), which differs from (2) and can be a more realistic setup from an empirical and experimental viewpoint. In fact, it is observed also in Maume-Deschamps (2006) that the influence of low SNR noise is not obvious in density estimation. We therefore adopt this setup in our experiments. All the observations reported in this experimental section apply to the noiseless case (2).

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, σ is set to 0.01 and the initial state x_0 is randomly generated based on the density f. For the map T in (17), we choose Logistic map in Example 2 and Gauss map in Example 3. We vary the sample size among $\{5\times 10^2, 10^3, 5\times 10^3, 10^4\}$, implement bandwidth selection procedures over 20 replications and select the bandwidth from a grid of values in the interval $[h_L, h_U]$ with 100 equispaced points. Here, h_L is set as the minimum distance between consecutive points x_i, $i = 1, \cdots, n$ (Devroye and Lugosi, 1997), while h_U is chosen according to the *maximal smoothing principle* proposed in Terrell (1990). Throughout our experiments, we use the Gaussian kernel for the kernel density estimators.

In our experiments, we conduct comparisons among the above-mentioned bandwidth selectors which are, respectively, denoted as follows:

- LSCV: the least squares cross-validation given in (14);
- MLSCV-1: the modified least squares cross-validation in (15) with $l_n = 1$;
- MLSCV-2: the modified least squares cross-validation in (15) with $l_n = 2$;
- DKM: the double kernel method defined in (16) where the two kernels used here are the Epanechnikov kernel and the Triangle kernel, respectively.

In the experiments, due to the known density functions for Logistic map and Gauss map, and in accordance with our previous analysis from the L_1 point of view, the criterion of comparing different selected bandwidths is the following absolute mean error (AME):

$$\mathrm{AME}(h) = \frac{1}{m}\sum_{i=1}^{m} |f_{D,h}(u_i) - f(u_i)|,$$

where $u_1, \cdots, u_m$ are m equispaced points in the interval $[0, 1]$ and m is set to 10000. We also compare the selected bandwidth with the one that has the minimum absolute mean error which serves as a ***baseline*** method in our experiments.

5.3 Simulation Results and Observations

The AMEs of the above bandwidth selectors for Logistic map in Example 2 and Gauss map in Example 3 over 20 replications are averaged and recorded in Tables 1 and 2 below.

In Figs. 1 and 2, we also plot the kernel density estimators for Logistic map in Example 2 and Gauss map in Example 3 with different bandwidths and their true density functions with different sample sizes. The sample size of each panel, in Figs. 1 and 2, from up to bottom, is 10^3, 10^4, and 10^5, respectively. In each panel, the densely dashed black curve represents the true density, the dotted blue curve is the estimated density function with the bandwidth selected by the baseline method while the solid red curve stands for the estimated density with the bandwidth selected by the double kernel method. All density functions in Figs. 1 and 2 are plotted with 100 equispaced points in the interval $(0, 1)$.

From Tables 1 and 2, and Figs. 1 and 2, we see that the true density functions of Logistic map and Gauss map can be approximated well with enough observations and the double kernel method works slightly better than the other three methods for the two dynamical systems. In fact, according to our experimental experience, we find that the bandwidth selector of the kernel density estimator for a dynamical system is usually ad-hoc. That

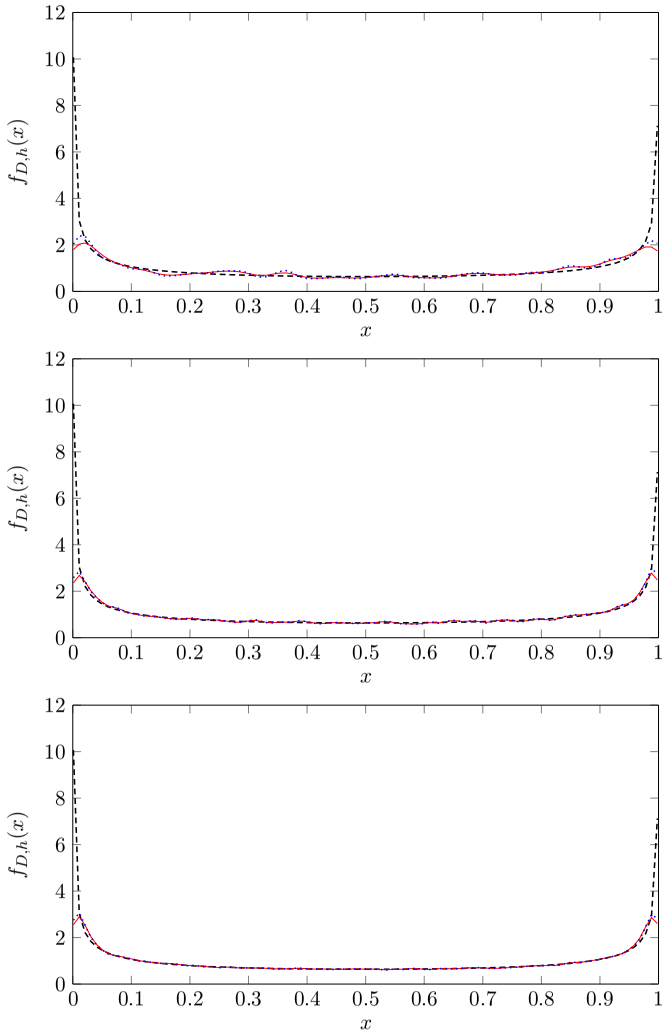


Figure 1: Plots of the kernel density estimators $f_{D,h}$ for Logistic map in Example 2 with different bandwidths and its true density with different sample sizes. The sample size of each panel, from up to bottom, is 10^3, 10^4, and 10^5, respectively. In each panel, the dashed black curve represents the true density of Logistic map, the dotted blue curve is the estimated density of Logistic map with the bandwidth selected by the baseline method while the solid red curve stands for the estimated density of Logistic map with the bandwidth selected by the double kernel method. All curves are plotted with 100 equispaced points in the interval $(0, 1)$.

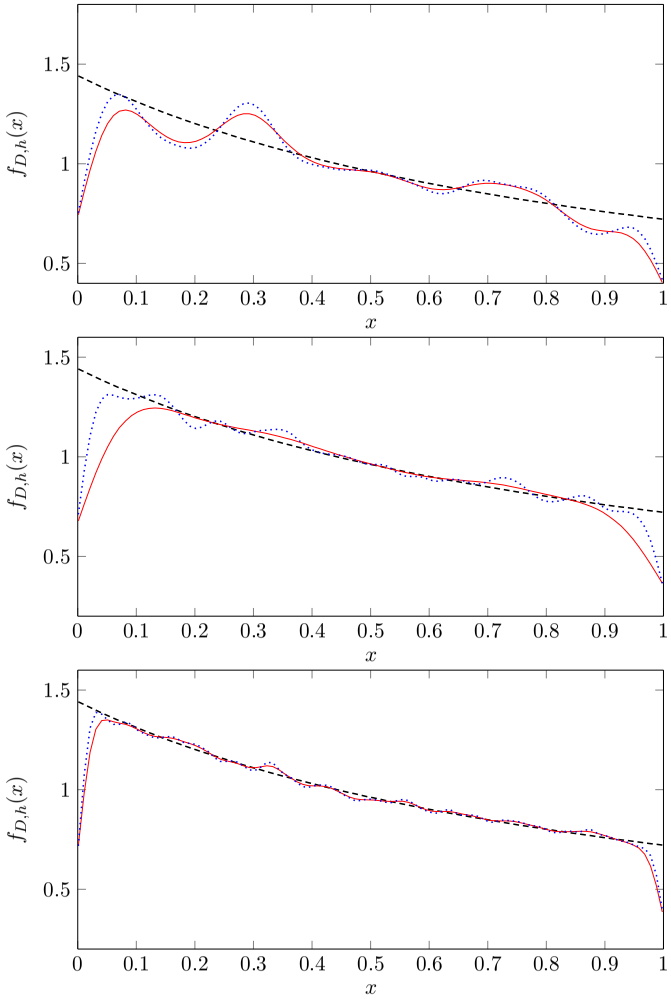


Figure 2: Plots of the kernel density estimators $f_{D,h}$ for Gauss map in Example 3 with different bandwidths and its true density with different sample sizes. The sample size of each panel, from up to bottom, is 10^3, 10^4, and 10^5, respectively. In each panel, the dashed black curve represents the true density of Gauss map, the dotted blue curve is the estimated density of Gauss map with the bandwidth selected by the baseline method while the solid red curve stands for the estimated density of Gauss map with the bandwidth selected by the double kernel method. All curves are plotted with 100 equispaced points in the interval $(0, 1)$.

Table 1: The AMEs of Different Bandwidth Selectors for Logistic Map in Example 2

sample size	LSCV	MLSCV-1	MLSCV-2	DKM	Baseline
5×10^2	.3372	.3369	.3372	.3117	.3013
1×10^3	.2994	.2994	.2994	.2804	.2770
5×10^3	.2422	.2422	.2422	.2340	.2326
1×10^4	.2235	.2235	.2235	.2220	.2192

Table 2: The AMEs of Different Bandwidth Selectors for Gauss Map in Example 3

sample size	LSCV	MLSCV-1	MLSCV-2	DKM	Baseline
5×10^2	.1027	.1026	.1059	.1181	.0941
1×10^3	.0925	.0933	.0926	.0925	.0878
5×10^3	.0626	.0626	.0626	.0586	.0585
1×10^4	.0454	.0454	.0454	.0440	.0439

is, for existing bandwidth selectors, there seems no a universal optimal one that can be applicable to all dynamical systems and outperforms the others. Therefore, further exploration and insights on the bandwidth selection problem in the dynamical system context certainly deserve future study. On the other hand, we also notice that due to the presence of dependence among observations generated by dynamical systems, the sample size usually needs to be large enough to approximate the density function well. This can be also seen from the plotted density functions in Figs. 1 and 2 with varying sample sizes.

Aside from the above observations, not surprisingly, from Figs. 1 and 2, we also observe the *boundary effect* (Gasser et al., 1985) from the kernel density estimators for dynamical systems, which seems to be even more significant than the i.i.d case. From a practical implementation view, some special studies are arguably called for addressing this problem.

6. Proofs

The following lemma is needed for the proof of Example 2.

Lemma 16 *For $x, h \geq 0$ with $x \geq 2h$ and $p \in (0,1)$ we have*

$$x - h \geq 2^{p-1} x^{1-p} h^p \,.$$

Proof [of Lemma 16] For $c := 2^{p-1}$, $x \geq 0$, and $h \geq 0$ we define $g_h(x) := cx^{1-p}h^p - x + h$, so that our goal is to show $g_h(x) \leq 0$ for all $x \geq 2h$. To this end, we first note that

$$g_h(2h) = c(2h)^{1-p}h^p - h = c2^{1-p}h^{1-p}h^p - h = 0\,,$$

that is, the assertion is true for $x = 2h$. To show the inequality for $x > 2h$, we note that $g_h'(x) = (1-p)cx^{-p}h^p - 1$. A simple calculation then shows that g_h' has its only zero at $x^* := ((1-p)c)^{1/p}h$. Moreover, since $g_h''(x) = (1-p)(-p)cx^{-1-p}h^p < 0$ we see that g_h has a global maximum at x^* and no (local) minimum. Consequently, g_h is decreasing on $[x^*, \infty)$,

so that it suffices to show that $x^* \leq 2h$. The latter, however is equivalent to $(1-p)c \leq 2^p$, and by the definition of c, this inequality is satisfied if and only if $(1-p)/2 \leq 1$. Since the latter is obviously true, we obtain the assertion. ∎

Proof [of Example 2] Here we only need to show that the density f is α-controllable with the specified functions. Moreover, this is obvious for $x < 0$ and $x > 1$. Moreover, the first and second derivatives of f on $(1/8, 7/8)$ are

$$f'(x) = -\frac{1-2x}{2\pi(x(1-x))^{3/2}} \qquad \text{and} \qquad f''(x) = \frac{8x^2 - 8x + 3}{4\pi(x(1-x))^{5/2}}\,.$$

Clearly, this gives $f''(x) \geq 1$ for all $x \in (1/8, 7/8)$ and hence f' is increasing on this interval. Using the symmetry of f' around $x = 1/2$ we then obtain

$$\sup_{x \in (1/8, 7/8} |f'(x)| = |f'(1/8)| = \frac{3}{8\pi} \cdot \Big(\frac{64}{7}\Big)^{3/2} \leq 4\,.$$

Consequently, the restriction $f_{|[1/8,7/8]}$ is Lipschitz continuous with constant not exceeding 4. This shows the assertion for $x \in [1/4, 3/4]$. Therefore, it remains to consider the cases $0 < x < 1/4$ and $3/4 < x < 1$, and due to the symmetry of f we may actually restrict our considerations to $0 < x < 1/4$. Let us therefore fix an $x \in (0, 1/4)$. For h with $|h| < r(x)$, that is $h \in (-x/2, x/2)$, we then find

$$\pi|f(x+h) - f(x)| = \frac{\big|\sqrt{x(1-x)} - \sqrt{(x+h)(1-x-h)}\big|}{\sqrt{x(1-x)(1-x-h)(x+h)}}\,. \tag{18}$$

Moreover, we have $1 - x > 3/4$ and $1 - x - h > 5/8$ and hence we find

$$\frac{1}{\sqrt{x(x+h)(1-x-h)(1-x)}} \leq \sqrt{\frac{32}{15}} \cdot \frac{1}{\sqrt{x(x+h)}}\,. \tag{19}$$

Our next goal is to bound the numerator in (18). To this end, we define

$$g(x,h) := \sqrt{x(1-x)} - \sqrt{(x+h)(1-x-h)}\,, \qquad h, x \in [0, 1/4]\,.$$

Note that the function $t \mapsto \sqrt{t(1-t)}$ is increasing on $[0, 1/2]$ and hence we have $g(x,h) \geq 0$ if $h \in [-1/4, 0]$ and $g(x,h) \leq 0$ if $h \in [0, 1/4]$. Now, our goal is to establish

$$\big|g(x,h)\big| \leq |h|^{1/2}\,, \qquad x \in [0, 1/4],\ h \in (-x/2, 1/4] \tag{20}$$

In the case $h \in [0, 1/4]$ our preliminary considerations on g show that (20) is equivalent to

$$(x+h)(1-x-h) \leq h + x(1-x) + 2\sqrt{hx(1-x)}\,.$$

Moreover, simple algebraic transformations show that the latter is equivalent to $-2xh - h^2 \leq 2\sqrt{hx(1-x)}$, which is obviously true. This shows (20) for $h \in [0, 1/4]$. Now, in the case $h \in (-x/2, 0]$ we first observe that we have

$$\big|g(x,h)\big| = \big|g(x+h, -h)\big|\,.$$

Let us write $\tilde{x} := x + h$ and $\tilde{h} := -h$. Then we have $\tilde{h} \in [0, x/2) \subset [0, 1/4]$ and $\tilde{x} \in (x/2, x] \subset (0, 1/4]$. Using (20) in the already proven case $\tilde{h} \in [0, 1/4]$ we then find

$$\bigl|g(x,h)\bigr| = \bigl|g(\tilde{x},\tilde{h})\bigr| \leq |\tilde{h}|^{1/2} = |h|^{1/2}\,,$$

which finishes the proof of (20).

Now combining (19) with (20) we find

$$\pi\,|f(x+h) - f(x)| \leq \sqrt{\frac{32}{15}} \cdot \frac{\sqrt{|h|}}{\sqrt{x(x+h)}}.$$

Let us first consider the case $h \in (-x/2, 0]$. Then Lemma 16 applied to $p := 2\varepsilon$ and $\tilde{h} := -h$ gives

$$\frac{1}{\sqrt{x+h}} = \frac{1}{\sqrt{x-\tilde{h}}} \leq 2^{(1-p)/2} x^{(p-1)/2} \tilde{h}^{-p/2} = 2^{1/2-\varepsilon} x^{-1/2+\varepsilon} |h|^{-\varepsilon}$$

Inserting this inequality in the previous inequality thus shows

$$|f(x+h) - f(x)| \leq \frac{1}{\pi} \cdot \sqrt{\frac{64}{15}} \cdot x^{-1+\varepsilon}\,|h|^{1/2-\varepsilon} \leq x^{-1+\varepsilon}\,|h|^{1/2-\varepsilon}\,. \tag{21}$$

Moreover, for $h \in [0, x/2)$ we have $x+h \geq x-h$, and hence the previous considerations actually give (21) for all $h \in (-x/2, x/2)$. Since for $\varepsilon := 1/2-\alpha$ we have both $-1+\varepsilon = -1/2-\alpha$ and $1/2-\varepsilon = \alpha$, we then obtain the assertion. ∎

The following lemma, which will be used several times in the sequel, supplies the key to the proof of Propositions 11 and 10.

Lemma 17 *Assume that K is a d-dimensional smoothing kernel that satisfies Condition (ii) of Assumption B. Then there exists a constant $c_1 > 0$ such that for all $r > 0$ we have*

$$\int_{B_r^c} K(\|x\|)\,\mathrm{d}x \leq c_1 r^{-\upsilon}\,. \tag{22}$$

Moreover, for all probability measures Q on $\mathbb{R}^d$ and all $h > 0$ and $r > 0$ the functions $k_{x,h}$, which are defined in (12) for all $x \in \mathbb{R}^d$, satisfy

$$\int_{B_r^c} \mathbb{E}_Q k_{x,h}\,\mathrm{d}x \leq \kappa\, Q(B_{r/2}^c) + c_1 2^\upsilon\,(h/r)^\upsilon\,. \tag{23}$$

Proof [of Lemma 17] For the proof of (22) we fix a constant $\tilde{c} \geq 1$ such that we have $\tilde{c}^{-1}\,\|\cdot\|_{\ell_2^d} \leq \|\cdot\| \leq \tilde{c}\,\|\cdot\|_{\ell_2^d}$. Since K is monotonically decreasing, we then find

$$\begin{aligned}
\int_{B_r^c} K(\|x\|)\,\mathrm{d}x \leq \int_{B_r^c} K(\tilde{c}^{-1}\,\|x\|_{\ell_2^d})\,\mathrm{d}x &\leq \int_{\tilde{c}^{-1} r}^\infty K(\tilde{c}^{-1}t) t^{d-1}\,\mathrm{d}t \\
&= \tilde{c}^d \int_{\tilde{c}^{-2}\,r}^\infty K(t) t^{d-1}\,\mathrm{d}t \\
&\leq \tilde{c}^{d+2\upsilon} \int_{\tilde{c}^{-2}\,r}^\infty K(t) r^{-\upsilon} t^{d+\upsilon-1}\,\mathrm{d}t \\
&\leq \tilde{c}^{d+2\upsilon}\,\kappa_\upsilon\, r^{-\upsilon}\,,
\end{aligned}$$

where in the last step we used Condition (ii) of Assumption B.

For the proof of (23) we first observe that for $t_0 > 0$, we have

$$\begin{aligned}
\int_{B_r^c} \mathbb{E}_Q k_{x,h}\,\mathrm{d}x &= \int_{B_r^c} \int_{\mathbb{R}^d} h^{-d} K\left(\|x - x'\|/h\right)\,\mathrm{d}Q(x')\,\mathrm{d}x \\
&= \int_{\mathbb{R}^d} \int_{\mathbb{R}^d} K(\|x\|) \mathbf{1}_{B_r^c}(hx + x')\,\mathrm{d}x\,\mathrm{d}Q(x') \\
&= \int_{\mathbb{R}^d} K(\|x\|) \int_{\mathbb{R}^d} \mathbf{1}_{B_r^c}(hx + x')\,\mathrm{d}Q(x')\,\mathrm{d}x \\
&\leq \int_{B_{t_0}} K(\|x\|) \int_{\mathbb{R}^d} \mathbf{1}_{B_r^c}(hx + x')\,\mathrm{d}Q(x')\mathrm{d}x + \int_{B_{t_0}^c} K(\|x\|)\,\mathrm{d}x.
\end{aligned}$$

Moreover, it is easy to see that $\mathbf{1}_{B_r^c}(hx + x') = 1$ if and only if $\|hx + x'\| \geq r$. Now we set $t_0 := \frac{r}{2h}$. In this case, if we additionally have $x \in B_{t_0}$, then $\|x'\| \geq r - h\|x\| \geq r - ht_0 = r/2$. Using (22) we thus find

$$\begin{aligned}
\int_{B_r^c} \mathbb{E}_Q k_{x,h}\,\mathrm{d}x &\leq \int_{B_{t_0}} K(\|x\|) Q(B_{r/2}^c)\,\mathrm{d}x + \int_{B_{t_0}^c} K(\|x\|)\,\mathrm{d}x \\
&\leq \kappa\, Q(B_{r/2}^c) + c_1\, t_0^{-\upsilon}\,.
\end{aligned} \tag{24}$$

By $t_0 = \frac{r}{2h}$ we then find the assertion. ∎

The following technical lemma is needed in the proof of Proposition 10.

Lemma 18 *For all $0 \leq a \leq b$ and $d \geq 1$ we have*

$$b^d - a^d \leq d \cdot b^{d-1} \cdot (b - a)\,. \tag{25}$$

Proof [of Lemma 18] In the case $b = 0$ there is nothing to prove. Moreover, in the case $b > 0$, we first observe by deviding by b^d that (25) is equivalent to

$$1 - \left(\frac{a}{b}\right)^d \leq d \cdot \left(1 - \frac{a}{b}\right).$$

Consequently, it suffices to show

$$1 - t^d \leq d\,(1 - t)\,, \qquad t \in [0, 1]\,. \tag{26}$$

To this end, we define $h(t) = d\,(1-t) - 1 + t^d$ for $t \in [0, 1]$. This gives $h'(t) = -d + dt^{d-1}$ and a simple check then shows $h'(t) \leq 0$ for all $t \in [0, 1]$. Moreover, we have $h(1) = 0$ and combining both we thus find $h(t) \geq 0$ for all $t \in [0, 1]$. This shows (26). ∎

Proof [of Proposition 10] *(i).* Since the space of continuous and compactly supported functions $C_c(\mathbb{R}^d)$ is dense in $L_1(\mathbb{R}^d)$, we can find $\bar{f} \in C_c(\mathbb{R}^d)$ such that

$$\|f - \bar{f}\|_1 \leq \varepsilon/3,\ \ \forall \varepsilon > 0.$$

Therefore, for any $\varepsilon > 0$, we have

$$\begin{aligned}\|f_{P,h} - f\|_1 &= \int_{\mathbb{R}^d} |f * K_h - f| \,\mathrm{d}x \\ &\le \int_{\mathbb{R}^d} |f * K_h - \bar{f} * K_h| \,\mathrm{d}x + \int_{\mathbb{R}^d} |\bar{f} * K_h - \bar{f}| \,\mathrm{d}x + \int_{\mathbb{R}^d} |f - \bar{f}| \,\mathrm{d}x \qquad (27) \\ &\le \frac{2\varepsilon}{3} + \int_{\mathbb{R}^d} |\bar{f} * K_h - \bar{f}| \,\mathrm{d}x,\end{aligned}$$

where K_h is defined in (6) and the last inequality follows from the fact that

$$\|f * K_h - \bar{f} * K_h\|_1 \le \|f - \bar{f}\|_1 \le \varepsilon/3.$$

The above inequality is due to Young's inequality (8.7) in Folland (1999). Moreover, there exist a constant $M > 0$ such that $\operatorname{supp}(\bar{f}) \subset B_M$ and a constant $r > 0$ such that

$$\int_{B_r^c} K(\|x\|) \,\mathrm{d}x \le \frac{\varepsilon}{9\|\bar{f}\|_1}.$$

Now we define $L : \mathbb{R}^d \to [0, \infty)$ by

$$L(x) := \mathbf{1}_{[-r,r]}(\|x\|) K(\|x\|)$$

and $L_h : \mathbb{R}^d \to [0, \infty)$ by

$$L_h(x) := h^{-d} L(x/h).$$

Then we have

$$\begin{aligned}\int_{\mathbb{R}^d} |\bar{f} * K_h - \bar{f}| \,\mathrm{d}x &\le \int_{\mathbb{R}^d} |\bar{f} * K_h - \bar{f} * L_h| \,\mathrm{d}x + \int_{\mathbb{R}^d} |\bar{f} * L_h - \bar{f}| \,\mathrm{d}x \\ &\le \|\bar{f}\|_1 \|K_h - L_h\|_1 + \int_{\mathbb{R}^d} \left| \bar{f} * L_h - \bar{f} \int_{\mathbb{R}^d} L_h \,\mathrm{d}x \right| \mathrm{d}x \\ &\quad + \int_{\mathbb{R}^d} \left| \bar{f} * \int_{\mathbb{R}^d} (L_h - K_h) \,\mathrm{d}x \right| \mathrm{d}x \\ &\le 2\|\bar{f}\|_1 \|K_h - L_h\|_1 + \int_{\mathbb{R}^d} \left| \bar{f} * L_h - \bar{f} \int_{\mathbb{R}^d} L_h \,\mathrm{d}x \right| \mathrm{d}x.\end{aligned}$$

Moreover, we have

$$\begin{aligned}\|K_h - L_h\|_1 &= \int_{\mathbb{R}^d} \frac{1}{h^d} \left| \mathbf{1}_{[-r,r]}\left(\frac{\|x\|}{h}\right) K\left(\frac{\|x\|}{h}\right) - K\left(\frac{\|x\|}{h}\right) \right| \mathrm{d}x \\ &= \int_{\mathbb{R}^d} \left| \mathbf{1}_{[-r,r]}(\|x\|) K(\|x\|) - K(\|x\|) \right| \mathrm{d}x \\ &= \int_{B_r^c} K(\|x\|) \,\mathrm{d}x \le \frac{\varepsilon}{9\|\bar{f}\|_1}.\end{aligned}$$

Finally, for $h \le 1$, we have

$$\begin{aligned}\int_{\mathbb{R}^d} \left| \bar{f} * L_h - \bar{f} \int_{\mathbb{R}^d} L_h \,\mathrm{d}x \right| \mathrm{d}x &= \int_{\mathbb{R}^d} \left| \int_{\mathbb{R}^d} \big(\bar{f}(x - x') - \bar{f}(x)\big) L_h(x') \,\mathrm{d}x' \right| \mathrm{d}x \\ &\le \int_{B_{r+M}} \int_{\mathbb{R}^d} |\bar{f}(x - x') - \bar{f}(x)| L_h(x') \,\mathrm{d}x' \,\mathrm{d}x.\end{aligned}$$

Since $\bar{f}$ is uniformly continuous, there exists a constant $h_\varepsilon > 0$ such that for all $h \le h_\varepsilon$ and $\|x'\| \le rh$, we have

$$|\bar{f}(x - x') - \bar{f}(x)| \le \varepsilon' := \frac{\varepsilon}{9(r + M)^d \lambda^d(B_1)}.$$

Consequently we obtain

$$\int_{\mathbb{R}^d} |\bar{f}(x - x') - \bar{f}(x)| L_h(x') \,\mathrm{d}x' \le \varepsilon' \int_{B_{rh}} L_h(x') \,\mathrm{d}x' \le \varepsilon' \int_{\mathbb{R}^d} K_h \,\mathrm{d}x = \varepsilon'.$$

Therefore, we obtain

$$\int_{\mathbb{R}^d} \left| \bar{f} * L_h - \bar{f} \int_{\mathbb{R}^d} L_h \,\mathrm{d}x \right| \mathrm{d}x \le \int_{B_{r+M}} \varepsilon' \,\mathrm{d}x = \frac{\varepsilon}{9} \qquad (28)$$

and consequently the assertion can be proved by combining estimates in (27) and (28).

(ii). The α-Hölder continuity of f tells us that for any $x \in \mathbb{R}^d$, there holds

$$\begin{aligned}|f_{P,h}(x) - f(x)| &= \left| \frac{1}{h^d} \int_{\mathbb{R}^d} K\left(\frac{\|x - x'\|}{h}\right) f(x') \,\mathrm{d}x' - f(x) \right| \\ &= \left| \int_{\mathbb{R}^d} K(\|x'\|) f(x + hx') \,\mathrm{d}x' - f(x) \right| \\ &= \left| \int_{\mathbb{R}^d} K(\|x'\|) \big(f(x + hx') - f(x)\big) \,\mathrm{d}x' \right| \\ &\le c_1 \int_{\mathbb{R}^d} K(\|x'\|) \big(h\|x'\|\big)^\alpha \,\mathrm{d}x' \\ &\le c_2 \int_{\mathbb{R}^d} K\big(\|x'\|_{\ell_2^d}\big) h^\alpha \|x'\|_{\ell_2^d}^\alpha \,\mathrm{d}x' \\ &\le c_3 h^\alpha \int_0^\infty K(r) r^{\alpha + d - 1} \,\mathrm{d}r \\ &\le c_3 \kappa_\alpha h^\alpha,\end{aligned}$$

where $c_1, c_2, c_3 > 0$ are suitable constants.

(iii). For fixed $h > 0$ we define $r := \frac{r_0}{2h}$. A quick calculation then yields $hr = r_0/2 < r_0$. Following the proof of *(ii)* and using that f is uniformly pointwise α-Hölder controllable,

we thus find for λ^d-almost all $x \in \mathbb{R}^d$:

$$\begin{aligned}
|f_{P,h}(x) - f(x)| &= \left| \int_{\mathbb{R}^d} K(\|x'\|) \big(f(x+hx') - f(x)\big)\, \mathrm{d}x' \right| \\
&\le \int_{B_r} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' + \int_{B_r^c} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \\
&\le c(x) \int_{B_r} K(\|x'\|) \big(h\|x'\|\big)^\alpha \, \mathrm{d}x' + f(x) \int_{B_r^c} K(\|x'\|)\, \mathrm{d}x' \\
&\quad + \int_{B_r^c} K(\|x'\|)\, f(x+hx')\, \mathrm{d}x' \\
&\le c_2\, c(x) h^\alpha + c_1 f(x) r^{-\upsilon} + \int_{B_r^c} K(\|x'\|)\, f(x+hx')\, \mathrm{d}x'
\end{aligned}$$

where the first term is estimated similarly to the first part of the proof of (22), and the second term is directly estimated by (22). The definition of r then yields the result.

(iv). Let us fix an $h > 0$. Following the proof of *(ii)* and using that f is pointwise α-Hölder controllable, we then obtain for $x \in \mathbb{R}^d$ with $r(x) > hR$:

$$\begin{aligned}
|f_{P,h}(x) - f(x)| &= \left| \int_{\mathbb{R}^d} K(\|x'\|) \big(f(x+hx') - f(x)\big)\, \mathrm{d}x' \right| \\
&\le \int_{B_R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \qquad (29) \\
&\le c(x) h^\alpha \int_{B_R} K(\|x'\|)\, \|x'\|^\alpha \, \mathrm{d}x' \\
&\le c \cdot c(x) \cdot h^\alpha \,, \qquad (30)
\end{aligned}$$

where c is a suitable constant bounding the integral in the second to last line. In particular, this shows the assertion for all $x \in \Omega^{+hR}$ for which $r(x)$ exists and satisfies $r(x) > hR$.

Let us now consider the case $x \in \Omega^{+hR}$ with $r(x) \le hR$. Here, (29) and a repetition of the estimates around (30) give

$$\begin{aligned}
|f_{P,h}(x) - f(x)| &\le \int_{B_R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \\
&= \int_{\|x'\| < \frac{r(x)}{h}} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \\
&\quad + \int_{\frac{r(x)}{h} \le \|x'\| \le R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \\
&\le c \cdot c(x) \cdot h^\alpha + \int_{\frac{r(x)}{h} \le \|x'\| \le R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \,.
\end{aligned}$$

To bound the second integral, we first observe that

$$\begin{aligned}
&\int_{\frac{r(x)}{h} \le \|x'\| \le R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' \\
&\le \int_{\frac{r(x)}{h} \le \|x'\| \le R} K(\|x'\|)\, f(x+hx')\, \mathrm{d}x' + \int_{\frac{r(x)}{h} \le \|x'\| \le R} K(\|x'\|)\, f(x)\, \mathrm{d}x' \\
&\le \|K\|_\infty h^{-d} \int_{r(x) \le \|x'\| \le hR} f(x+x')\, \mathrm{d}x' + \|K\|_\infty f(x) \int_{\frac{r(x)}{h} \le \|x'\| \le R} 1\, \mathrm{d}x' \,.
\end{aligned}$$

It thus remains to bound the second integral. To this end, observe that for $C := \mathrm{vol}_d(B_{\|\cdot\|})$ we have

$$\int_{\frac{r(x)}{h} \le \|x'\| \le R} 1\, \mathrm{d}x' = C\left(R^d - \left(\frac{r(x)}{h}\right)^d\right) \le dCR^{d-1}\left(R - \frac{r(x)}{h}\right),$$

where the last inequality is due to Lemma 18. Combining all estimates yields the assertion for all $x \in \Omega^{+hR}$ for which $r(x)$ exists and satisfies $r(x) \le hR$.

Let us finally consider the case of an $x \notin \Omega^{+hR}$ for which $r(x)$ exists. Then we obviously have $x \notin \Omega$ and since $r(x)$ exists we conclude that $f(x) = 0$. Moreover, the definition of Ω^{+hR} gives $\|x - x''\| > hR$ for all $x'' \in \Omega$. Let us now fix an x' with $\|x'\| \le R$. Assume we had $x + hx' \in \Omega$. For $x'' := x + hx'$ we would then find $hR < \|x - x''\| = \|hx'\| \le hR$, which is impossible. Consequently, we have $x + hx' \notin \Omega$, and thus we obtain $f(x+hx') = 0$ for λ^d-almost all x' with $\|x'\| \le R$. Combining our considerations with (29) leads to

$$|f_{P,h}(x) - f(x)| \le \int_{\|x'\| \le R} K(\|x'\|) \big|f(x+hx') - f(x)\big|\, \mathrm{d}x' = 0\,,$$

and hence we have shown the assertion. ∎

Proof [of Proposition 11] *(i).* We decompose $\|f_{P,h} - f\|_1$ as follows

$$\begin{aligned}
\|f_{P,h} - f\|_1 &= \int_{B_r} |f_{P,h}(x) - f(x)|\, \mathrm{d}x + \int_{B_r^c} |f_{P,h}(x) - f(x)|\, \mathrm{d}x \qquad (31) \\
&\le \lambda^d(B_r) \|f_{P,h} - f\|_\infty + \int_{B_r^c} \mathbb{E}_P k_{x,h}\, \mathrm{d}x + \int_{B_r^c} f\, \mathrm{d}x \\
&\le \lambda^d(B_1)\, r^d\, \|f_{P,h} - f\|_\infty + \int_{B_r^c} \mathbb{E}_P k_{x,h}\, \mathrm{d}x + P\left(B_r^c\right). \qquad (32)
\end{aligned}$$

Combining (32) and (24), we obtain the desired assertion.

(ii). Since f is pointwise α-Hölder controllable, *(iii)* of Proposition 10 tells us that

$$\begin{aligned}
\int_{B_r} |f_{P,h}(x) - f(x)|\, \mathrm{d}x &\le c h^\alpha \cdot \int_{B_r} c(x)\, \mathrm{d}x + c h^\upsilon + \int_{B_r} \int_{B_{r_0/(2h)}^c} K(\|x'\|)\, f(x+hx')\, \mathrm{d}x'\, \mathrm{d}x \\
&= c\, H(r)\, h^\alpha + c h^\upsilon + \int_{B_{r_0/(2h)}^c} K(\|x'\|) \int_{B_r} f(x+hx')\, \mathrm{d}x\, \mathrm{d}x' \\
&\le c\, H(r)\, h^\alpha + c h^\upsilon + \int_{B_{r_0/(2h)}^c} K(\|x'\|)\, \mathrm{d}x' \\
&\le c\, H(r)\, h^\alpha + \tilde{c}\, h^\upsilon \,,
\end{aligned}$$

where in the last step we used (22). Combining this estimate with the decomposition (31) and the inequality (24) then yields the assertion.

(iii). In this case, *(iv)* of Proposition 10 gives

$$\begin{aligned}\int_{\mathbb{R}^d} |f_{P,h}(x) - f(x)|\,\mathrm{d}x &= \int_{\Omega^{+hR}} |f_{P,h}(x) - f(x)|\,\mathrm{d}x \\ &\le c\,h^\alpha \int_{\Omega^{+hR}} c(x)\,\mathrm{d}x + c\cdot\|K\|_\infty \int_{\Omega^{+hR}} f(x)\left|R - \frac{r(x)}{h}\right|_+ \mathrm{d}x \\ &\quad + \|K\|_\infty h^{-d} \int_{\Omega^{+hR}} \int_{r(x)\le\|x'\|\le hR} f(x+x')\,\mathrm{d}x'\,\mathrm{d}x\,.\end{aligned}$$

Moreover, we have

$$\begin{aligned}\int_{\Omega^{+hR}} f(x)\left|R - \frac{r(x)}{h}\right|_+ \mathrm{d}x &\le \int_{\mathbb{R}^d} f(x)\left|R - \frac{r(x)}{h}\right|_+ \mathrm{d}x \\ &= \int_{r(x)\le hR} \Big(R - \frac{r(x)}{h}\Big)\,P(dx) \\ &\le R\cdot P(\{x : r(x)\le hR\})\,,\end{aligned}$$

which, together with our previous estimate gives the assertion.

(iv). Let us consider the partition $X_h^* = \big(X_h^*\cap\Omega^{+hR}\big)\cup\big(X_h^*\cap(\mathbb{R}^d\setminus\Omega^{+hR})\big)$. For $x\in X_h^*\cap(\mathbb{R}^d\setminus\Omega^{+hR})$, the first assertion of *(iv)* of Proposition 10 then shows $|f_{P,h}(x)-f(x)|=0$, and hece the assertion is true for these x. Moreover, for $X_h^*\cap\Omega^{+hR}$, the second assertion of *(iv)* of Proposition 10 gives

$$\begin{aligned}|f_{P,h}(x) - f(x)| &\le c\,c(x)h^\alpha + c\cdot\|K\|_\infty f(x)\left|R - \frac{r(x)}{h}\right|_+ \\ &\quad + \|K\|_\infty h^{-d}\int_{r(x)\le\|x'\|\le hR} f(x+x')\,\mathrm{d}x'\,,\end{aligned}$$

and since the second and third term vanish on X_h^* we obtain the assertion. ■

To prove Proposition 13, we need the following lemmas.

Lemma 19 *Let (X,d) and (Y,e) be metric spaces and $T: X\to Y$ be an α-Hölder continuous function with constant c. Then, for $A\subset X$ and all $\varepsilon>0$ we have*

$$\mathcal{N}(T(A), e, c\varepsilon^\alpha) \le \mathcal{N}(A,d,\varepsilon).$$

Proof [of Lemma 19] Let $x_1,\dots,x_n$ be an ε-net of A, that is, $A\subset\bigcup_{i=1}^n B_d(x_i,\varepsilon)$. For $i=1,\cdots,n$, we set $y_i := T(x_i)$. Now, it only suffices to show that this gives a $c\varepsilon^\alpha$-net of $T(A)$.

In fact, supposing that $y\in T(B_d(x_i,\varepsilon))$, then there exists $x\in B_d(x_i,\varepsilon)$ such that $T(x)=y$. This implies

$$e(T(x),T(x_i)) \le cd^\alpha(x,x_i) \le c\varepsilon^\alpha.$$

Therefore, we have $T(B_d(x_i,\varepsilon))\subset B_e(y_i,c\varepsilon^\alpha)$. That is, $y_1,\dots,y_n$ is a $c\varepsilon^\alpha$-net of $T(A)$. This completes the proof of Lemma 19. ■

Remark 20 *We remark that when X is a Banach space with the norm $\|\cdot\|$, then for any $c>0$ there holds*

$$\mathcal{N}(cA,\|\cdot\|,\varepsilon) = \mathcal{N}\left(A,\|\cdot\|,\varepsilon/c\right).$$

Lemma 21 *Let $\|\cdot\|'$ be another norm on $\mathbb{R}^d$. Then for all $\varepsilon\in(0,1]$ we have*

$$\mathcal{N}(B_1,\|\cdot\|',\varepsilon) \lesssim \varepsilon^{-d}.$$

Proof [of Lemma 21] It is a straightforward conclusion of Proposition 1.3.1 in Carl and Stephani (1990) and Lemma 6.21 in Steinwart and Christmann (2008). ■

Lemma 22 *Let K be a d-dimensional smoothing kernel that satisfies Conditions (i) in Assumption B. Let $h>0$ be the bandwidth parameter, and $k_{x,h}$ be defined in (12) for any $x\in\mathbb{R}^d$. Then we have*

$$\sup_{y\in\mathbb{R}^d}|k_{x,h}(y)-k_{x',h}(y)| \le \frac{c}{h^{\beta+d}}\|x-x'\|^\beta,\ \ x,x'\in\mathbb{R}^d,$$

where c is a positive constant.

Proof [of Lemma 22] From the definition of $k_{x,h}$ and the fact that K is a d-dimensional β-Hölder continuous kernel, we have

$$\begin{aligned}|k_{x,h}(y)-k_{x',h}(y)| &= \frac{1}{h^d}\left|K\Big(\frac{\|x-y\|}{h}\Big) - K\Big(\frac{\|x'-y\|}{h}\Big)\right| \\ &\le \frac{c}{h^d}\left|\frac{\|x-y\|}{h} - \frac{\|x'-y\|}{h}\right|^\beta \\ &\le ch^{-(\beta+d)}\|x-x'\|^\beta,\end{aligned}$$

where c is a positive constant. The desired conclusion is thus obtained. ■

Proof [of Proposition 13] Lemma 22 reveals that $\mathcal{K}_{h,r}$ is the image of Hölder continuous map $B_r\to L_\infty(\mathbb{R}^d)$ with the constant $ch^{-(\beta+d)}$. By Lemmas 19 and 21 we obtain

$$\begin{aligned}\mathcal{N}(\mathcal{K}_{h,r},\|\cdot\|_\infty,\varepsilon) &\le \mathcal{N}\left(B_r,\|\cdot\|,\Big(\frac{\varepsilon h^{\beta+d}}{c}\Big)^{1/\beta}\right) \\ &= \mathcal{N}\left(B_1,\|\cdot\|,\Big(\frac{\varepsilon h^{\beta+d}}{cr^\beta}\Big)^{1/\beta}\right) \\ &\le c'\left(\frac{\varepsilon h^{\beta+d}}{r^\beta}\right)^{-d/\beta},\end{aligned}$$

where c' is a constant independent of ε. This completes the proof of Proposition 13. ■

The following Bernstein-type exponential inequality, which was developed recently in Hang and Steinwart (2017), will serve as one of the main ingredients in the consistency and convergence analysis of the kernel density estimator (8). It can be stated in the following general form:

Theorem 23 (Bernstein Inequality (Hang and Steinwart, 2017)) *Assume that $\mathcal{X} := (X_n)_{n\geq 1}$ is an X-valued stationary geometrically (time-reversed) $\mathcal{C}$-mixing process on $(\Omega, \mathcal{A}, \mu)$ with $\|\cdot\|_{\mathcal{C}}$ be defined by (4) for some semi-norm $|||\cdot|||$ satisfying Condition (ii) in Assumption A, and $P := \mu_{X_1}$. Moreover, let $g : X \to \mathbb{R}$ be a function such that $g \in \mathcal{C}(X)$ with $\mathbb{E}_P g = 0$ and assume that there exist some $A > 0$, $B > 0$, and $\sigma \geq 0$ such that $|||g||| \leq A$, $\|g\|_\infty \leq B$, and $\mathbb{E}_P g^2 \leq \sigma^2$. Then, for all $\tau > 0$, $k \in \mathbb{N}$, and*

$$n \geq n_0 := \max\left\{\min\left\{m \geq 3 : m \geq \left(\frac{808c_0(3A+B)}{B}\right)^{\frac{1}{k}} \text{ and } \frac{m}{(\log m)^{\frac{2}{\gamma}}} \geq 4\right\}, e^{\frac{k+1}{b}}\right\},$$

with probability μ at least $1 - 4e^{-\tau}$, there holds

$$\left|\frac{1}{n}\sum_{i=1}^n g(X_i)\right| \leq \sqrt{\frac{8(\log n)^{\frac{2}{\gamma}}\sigma^2\tau}{n}} + \frac{8(\log n)^{\frac{2}{\gamma}}B\tau}{3n}.$$

Proof [of Proposition 14] Let the notations $k_{x,h}$ and $\widetilde{k}_{x,h}$ be defined in (12) and (13), respectively, that is, $k_{x,h} := h^{-d}K(\|x - \cdot\|/h)$, and $\widetilde{k}_{x,h} := k_{x,h} - \mathbb{E}_P k_{x,h}$. We first assume that $x \in \mathbb{R}^d$ is fixed and then estimate $\mathbb{E}_D\widetilde{k}_{x,h}$ by using Bernstein's inequality in Theorem 23. For this purpose, we shall verify the following conditions: Obviously, we have $\mathbb{E}_P\widetilde{k}_{x,h} = 0$. Moreover, simple estimates yield

$$\|\widetilde{k}_{x,h}\|_\infty \leq 2\|k_{x,h}\|_\infty \leq 2h^{-d}\|K\|_\infty \leq 2h^{-d}K(0)$$

and

$$\mathbb{E}_P\widetilde{k}^2_{x,h} \leq \mathbb{E}_P k^2_{x,h} = \int_{\mathbb{R}^d} k^2_{x,h}(x')\mathrm{d}P(x').$$

Finally, the first condition in Assumption A and Condition (iv) in Assumption B imply

$$|||\widetilde{k}_{x,h}||| \leq |||k_{x,h}||| \leq h^{-d}\sup_{x\in\mathbb{R}^d}|||K\left(\|x - \cdot\|/h\right)||| \leq h^{-d}\varphi(h).$$

Now we can apply the Bernstein-type inequality in Theorem 23 and obtain that for $n \geq n_1$, for any fixed $x \in \mathbb{R}^d$, with probability μ at most $4e^{-\tau}$, there holds

$$|\mathbb{E}_D\widetilde{k}_{x,h}| \geq \sqrt{\frac{8\tau(\log n)^{2/\gamma}\int_{\mathbb{R}^d} k^2_{x,h}(x')\mathrm{d}P(x')}{n}} + \frac{16\tau(\log n)^{2/\gamma}K(0)}{3h^d n}, \tag{33}$$

where

$$n_1 := \max\left\{\min\left\{m \geq 3 : m \geq \left(\frac{808c_0(3h^{-d}\varphi(h) + K(0))}{2K(0)}\right)^{\frac{1}{d+1}} \text{ and } \frac{m}{(\log m)^{\frac{2}{\gamma}}} \geq 4\right\}, e^{\frac{d+1}{b}}\right\}. \tag{34}$$

Consider the function set $\widetilde{\mathcal{K}}_{h,r} := \{\widetilde{k}_{x,h} : x \in B_r\}$. We choose $y_1, \ldots, y_m \in B_r$ such that $\{k_{y_1,h}, \ldots, k_{y_m,h}\}$ is a minimal $\varepsilon/2$-net of $\mathcal{K}_{h,r} = \{k_{x,h} : x \in B_r\}$ with respect to $\|\cdot\|_\infty$. Noticing the following relation

$$\|\widetilde{k}_{x,h} - \widetilde{k}_{y_j,h}\|_\infty \leq 2\|k_{x,h} - k_{y_j,h}\|_\infty \leq \varepsilon,$$

we know that $\{\widetilde{k}_{y_1,h}, \ldots, \widetilde{k}_{y_m,h}\}$ is an ε-net of $\widetilde{\mathcal{K}}_{h,r}$ with respect to $\|\cdot\|_\infty$. Note that here we have $m = \mathcal{N}(\mathcal{K}_{h,r}, \|\cdot\|_\infty, \frac{\varepsilon}{2})$, since the net is minimal. From Proposition 13, we know that there exists a positive constant c independent of ε such that $\log m \leq c\log\frac{r}{h\varepsilon}$. From the estimate in (33) and a union bound argument, with probability μ at least $1 - 4me^{-\tau}$, the following estimate holds

$$\sup_{j=1,\ldots,m}|\mathbb{E}_D\widetilde{k}_{y_j,h}| \leq \sqrt{\frac{8\tau(\log n)^{2/\gamma}\int_{\mathbb{R}^d} k^2_{y_j,h}(x')\mathrm{d}P(x')}{n}} + \frac{16\tau(\log n)^{2/\gamma}K(0)}{h^d n}.$$

By a simple variable transformation, we see that with probability μ at least $1 - e^{-\tau}$, there holds

$$\sup_{j=1,\ldots,m}|\mathbb{E}_D\widetilde{k}_{y_j,h}| \leq \sqrt{\frac{8(\log n)^{2/\gamma}\int_{\mathbb{R}^d} k^2_{y_j,h}(x')\mathrm{d}P(x')(\tau + \log(4m))}{n}} + \frac{16(\log n)^{2/\gamma}K(0)(\tau + \log(4m))}{h^d n}.$$

Recalling that $\{k_{y_1,h}, \ldots, k_{y_m,h}\}$ is an $\varepsilon/2$-net of $\mathcal{K}_{h,r}$, this implies that, for any $x \in B_r$, there exists y_j such that $\|k_{x,h} - k_{y_j,h}\|_\infty \leq \varepsilon/2$. Then we have

$$\begin{aligned}\left||\mathbb{E}_D\widetilde{k}_{x,h}| - |\mathbb{E}_D\widetilde{k}_{y_j,h}|\right| &\leq \left|\mathbb{E}_D\widetilde{k}_{x,h} - \mathbb{E}_D\widetilde{k}_{y_j,h}\right| \\ &\leq |\mathbb{E}_D k_{x,h} - \mathbb{E}_D k_{y_j,h}| + |\mathbb{E}_P k_{x,h} - \mathbb{E}_P k_{y_j,h}| \\ &\leq \|k_{x,h} - k_{y_j,h}\|_{L_1(D)} + \|k_{x,h} - k_{y_j,h}\|_{L_1(P)} \\ &\leq \varepsilon,\end{aligned}$$

and consequently

$$|\mathbb{E}_D\widetilde{k}_{x,h}| \leq |\mathbb{E}_D\widetilde{k}_{y_j,h}| + \varepsilon. \tag{35}$$

By setting $a := 8(\log n)^{2/\gamma}(\tau + \log(4m))/n$, we have

$$\begin{aligned}\left|\sqrt{a\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')} - \sqrt{a\int_{\mathbb{R}^d} k^2_{y_j,h}(x')\,\mathrm{d}P(x')}\right| &= \left|\|\sqrt{a}k_{x,h}\|_{L_2(P)} - \|\sqrt{a}k_{y_j,h}\|_{L_2(P)}\right| \\ &\leq \sqrt{a}\|k_{x,h} - k_{y_j,h}\|_{L_2(P)} \\ &\leq \sqrt{a}\varepsilon/2.\end{aligned}$$

This together with inequality (35) implies that for any $x \in B_r$, there holds

$$\begin{aligned}|\mathbb{E}_D\widetilde{k}_{x,h}| &\le |\mathbb{E}_D\widetilde{k}_{y_j,h}| + 2\varepsilon \\ &\le \sqrt{a\int_{\mathbb{R}^d} k^2_{y_j,h}(x')\,\mathrm{d}P(x')} + \frac{2aK(0)}{h^d} + \varepsilon \\ &\le \sqrt{a\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')} + \frac{\sqrt{a}\varepsilon}{2} + \frac{2aK(0)}{h^d} + \varepsilon.\end{aligned}$$

Consequently we have

$$\begin{aligned}\int_{B_r} |\mathbb{E}_D\widetilde{k}_{x,h}|\,\mathrm{d}x \le \int_{B_r} \sqrt{a\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')}\,\mathrm{d}x \\ + r^d\lambda^d(B_1)\cdot\frac{2aK(0)}{h^d} + r^d\lambda^d(B_1)(\sqrt{a}/2+1)\varepsilon.\end{aligned}$$

Now recall that for $E \subset \mathbb{R}^d$ and $g : E \to \mathbb{R}$, Hölder's inequality implies

$$\|g\|_{\frac{1}{2}} = \left(\int_{\mathbb{R}^d} |\mathbf{1}_E|^{\frac{1}{2}}|g|^{\frac{1}{2}}\,\mathrm{d}x\right)^2 \le \int_{\mathbb{R}^d} |\mathbf{1}_E|\,\mathrm{d}x \int_{\mathbb{R}^d} |g|\,\mathrm{d}x = \mu(E)\cdot\|g\|_1.$$

This tells us that

$$\int_{B_r}\sqrt{a\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')}\,\mathrm{d}x \le \sqrt{\mu(B_r)}\cdot\sqrt{\int_{B_r} a\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')\,\mathrm{d}x}.$$

Moreover, there holds

$$\begin{aligned}\int_{B_r}\int_{\mathbb{R}^d} k^2_{x,h}(x')\,\mathrm{d}P(x')\,\mu(\mathrm{d}x) &= \int_{\mathbb{R}^d}\int_{B_r} h^{-2d}K^2\left(\|x-x'\|/h\right)\,\mathrm{d}x\,\mathrm{d}P(x') \\ &\le \int_{\mathbb{R}^d}\int_{\mathbb{R}^d} h^{-2d}K^2\left(\|x\|/h\right)\,\mathrm{d}x\,\mathrm{d}P(x') \\ &= h^{-d}\int_{\mathbb{R}^d} K^2(\|x\|)\,\mathrm{d}x \\ &\le K(0)h^{-d}.\end{aligned}$$

We now set $\varepsilon = \frac{1}{n}$ and obtain $\log(4m) \le c\log\frac{nr}{h}$. Thus we have

$$\begin{aligned}\int_{B_r}|\mathbb{E}_D\widetilde{k}_{x,h}|\,\mathrm{d}x &\lesssim \sqrt{\frac{(\log n)^{2/\gamma}r^d(\tau+\log(4m))}{h^d n}} + \frac{(\log n)^{2/\gamma}r^d(\tau+\log(4m))}{h^d n} \\ &\quad + \sqrt{\frac{(\log n)^{2/\gamma}(\tau+\log(4m))}{n}}\cdot\frac{r^d}{n} \\ &\lesssim \sqrt{\frac{(\log n)^{2/\gamma}r^d(\tau+\log\frac{nr}{h})}{h^d n}} + \frac{(\log n)^{2/\gamma}r^d(\tau+\log\frac{nr}{h})}{h^d n}.\end{aligned} \tag{36}$$

Now we need to estimate the corresponding integral over B_r^c. By definition we have

$$\int_{B_r^c}|\mathbb{E}_D\widetilde{k}_{x,h}|\,\mathrm{d}x \le \int_{B_r^c}\mathbb{E}_D k_{x,h}\,\mathrm{d}x + \int_{B_r^c}\mathbb{E}_P k_{x,h}\,\mathrm{d}x.$$

From Lemma 17 we obtain

$$\int_{B_r^c}\mathbb{E}_D k_{x,h}\,\mathrm{d}x \lesssim D(B^c_{r/2}) + \left(\frac{h}{r}\right)^\beta = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{B^c_{r/2}}(x_i) + \left(\frac{h}{r}\right)^\beta,$$

and

$$\int_{B_r^c}\mathbb{E}_P k_{x,h}\,\mathrm{d}x \lesssim P(B^c_{r/2}) + \left(\frac{h}{r}\right)^\beta.$$

Since $r \ge 1$, we can construct a function g with $\mathbf{1}_{B^c_{r/2}} \le g \le \mathbf{1}_{B^c_{r/4}}$ and there exists a function $\psi(r)$ such that $\|\|g\|\| \le \psi(r)$. Applying Bernstein inequality in Theorem 23 with respect to this function g, it is easy to see that when $n \ge n_2$, with probability μ at least $1-2e^{-\tau}$, there holds

$$\mathbb{E}_D g - \mathbb{E}_P g \le \sqrt{\frac{8\tau(\log n)^{2/\gamma}}{n}} + \frac{8\tau(\log n)^{2/\gamma}}{3n},$$

where

$$n_2 := \max\left\{\min\left\{m \ge 3 : m^2 \ge 808c_0(3\psi(r)+1) \text{ and } \frac{m}{(\log m)^{\frac{2}{\gamma}}} \ge 4\right\}, e^{\frac{3}{b}}\right\}.$$

This implies that with probability μ at least $1-2e^{-\tau}$, there holds

$$\begin{aligned}D(B^c_{r/2}) = \frac{1}{n}\sum_{i=1}^n \mathbf{1}_{B^c_{r/2}}(x_i) &\le \mathbb{E}_D g \\ &\le \mathbb{E}_P g + \sqrt{\frac{8\tau(\log n)^{2/\gamma}}{n}} + \frac{8\tau(\log n)^{2/\gamma}}{3n} \\ &\le \mathbb{E}_P\mathbf{1}_{B^c_{r/4}}(x_i) + \sqrt{\frac{8\tau(\log n)^{2/\gamma}}{n}} + \frac{8\tau(\log n)^{2/\gamma}}{3n}\end{aligned}$$

and consequently we obtain

$$\int_{B_r^c}|\mathbb{E}_D\widetilde{k}_{x,h}|\,\mathrm{d}x \lesssim P(B^c_{r/4}) + \sqrt{\frac{32\tau(\log n)^{2/\gamma}}{n}} + \left(\frac{h}{r}\right)^\beta. \tag{37}$$

By combining estimates in (36) and (37), and taking $n_0 = \max\{n_1, n_2\}$, we have accomplished the proof of Proposition 14. ∎

Remark 24 *Let us briefly discuss the choice of the function $\psi(r)$ in the proof of Proposition 14. For example, in the case $\mathcal{C}(X) = \mathrm{Lip}(\mathbb{R})$, we can choose*

$$g(x) := \begin{cases} 1, & \text{for } |x| > r, \\ 0, & \text{for } |x| < r/4, \\ -\frac{4x}{3r} - \frac{1}{3}, & \text{for } -r \le x \le -r/4, \\ \frac{4x}{3r} - \frac{1}{3}, & \text{for } r/4 \le x \le r. \end{cases}$$

Then we have $\|\|g\|\| \le \frac{4}{3r} \le 4/3$ and therefore, n_2 is well-defined. Moreover, it is easily seen that even for smoother underlying functions classes like C^1 we can construct a function g such that $\|\|g\|\| < \infty$.

Proof [of Proposition 15] Recalling the definitions of $k_{x,h}$ and $\widetilde{k}_{x,h}$ given in (12) and (13), we have

$$\|f_{D,h} - f_{P,h}\|_\infty = \sup_{x\in\Omega} |\mathbb{E}_D \widetilde{k}_{x,h}|.$$

To prove the assertion, we first estimate $\mathbb{E}_D f_{x,h}$ for fixed $x \in \mathbb{R}^d$ using the Bernstein inequality in Theorem 23. For this purpose, we first verify the following conditions: Obviously, we have $\mathbb{E}_P \widetilde{k}_{x,h} = 0$. Then, simple estimates imply

$$\|\widetilde{k}_{x,h}\|_\infty \le 2\|k_{x,h}\|_\infty \le 2h^{-d}\|K\|_\infty \le 2h^{-d}K(0)$$

and

$$\mathbb{E}_P \widetilde{k}^2_{x,h} \le \mathbb{E}_P k^2_{x,h} = h^{-d} \int_{\mathbb{R}^d} K^2\left(\|x - x'\|/h\right) f(x') h^{-d}\,\mathrm{d}x' \lesssim \|f\|_\infty h^{-d}.$$

Finally, the first condition in Assumption A and Condition (iv) in Assumption B yield

$$\|\|\widetilde{k}_{x,h}\|\| \le \|\|k_{x,h}\|\| \le h^{-d} \sup_{x\in\mathbb{R}^d} \|\|K\left(\|x - \cdot\|/h\right)\|\| \le h^{-d}\varphi(h).$$

Therefore, we can apply the Bernstein inequality in Theorem 23 and obtain that for $n \ge n_0^*$, for any fixed $x \in \mathbb{R}^d$, with probability μ at least $1 - 4e^{-\tau}$, there holds

$$|\mathbb{E}_D \widetilde{k}_{x,h}| \lesssim \sqrt{\frac{\tau\|f\|_\infty (\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)\tau(\log n)^{2/\gamma}}{3h^d n}, \tag{38}$$

where

$$n_0^* := \max\left\{ \min\left\{ m \ge 3 : m \ge \left(\frac{808c_0(3h^{-d}\varphi(h) + K(0))}{2K(0)}\right)^{\frac{1}{d+1}} \text{ and } \frac{m}{(\log m)^{\frac{2}{\gamma}}} \ge 4 \right\}, e^{\frac{d+1}{b}} \right\}. \tag{39}$$

Let us consider the following function set

$$\mathcal{K}'_{h,r_0} := \left\{ \widetilde{k}_{x,h} : x \in B_{r_0} \right\}$$

and choose $y_1, \ldots, y_m \in B_{r_0}$ such that $\{k_{y_1,h}, \ldots, k_{y_m,h}\}$ is a minimal $\varepsilon/2$-net of $\mathcal{K}_{h,r_0}$ with respect to $\|\cdot\|_\infty$ and $m = \mathcal{N}(\mathcal{K}_{h,r_0}, \|\cdot\|_\infty, \frac{\varepsilon}{2})$. As in the proof of Proposition 14, one can show that $\widetilde{k}_{y_1,h}, \ldots, \widetilde{k}_{y_m,h}$ is an ε-net of $\mathcal{K}'_{h,r_0}$. Again from Proposition 13 we know that there holds $\log(4m) \lesssim \log \frac{r_0}{h\varepsilon}$. This in connection with (38) implies that the following union bound

$$\sup_{j=1,..,m} |\mathbb{E}_D \widetilde{k}_{y_j,h}| \lesssim \sqrt{\frac{\|f\|_\infty(\tau + \log(4m))(\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)(\tau + \log(4m))(\log n)^{2/\gamma}}{h^d n}$$

holds with probability μ at least $1 - e^{-\tau}$. For any $x \in B_{r_0}$, there exists a y_j such that $\|k_{x,h} - k_{y_j,h}\|_\infty \le \varepsilon$. Then we have

$$\begin{aligned} \left||\mathbb{E}_D \widetilde{k}_{x,h}| - |\mathbb{E}_D \widetilde{k}_{y_j,h}|\right| &\le |\mathbb{E}_D \widetilde{k}_{x,h} - \mathbb{E}_D \widetilde{k}_{y_j,h}| \\ &\le |\mathbb{E}_D k_{x,h} - \mathbb{E}_D k_{y_j,h}| + |\mathbb{E}_P k_{x,h} - \mathbb{E}_P k_{y_j,h}| \\ &\le \|k_{x,h} - k_{y_j,h}\|_{L_1(D)} + \|k_{x,h} - k_{y_j,h}\|_{L_1(P)} \\ &\le \varepsilon, \end{aligned}$$

and consequently with probability μ at least $1 - e^{-\tau}$, there holds

$$\begin{aligned} |\mathbb{E}_D \widetilde{k}_{x,h}| &\le |\mathbb{E}_D \widetilde{k}_{y_j,h}| + \varepsilon \\ &\lesssim \sqrt{\frac{\|f\|_\infty(\tau + \log(4m))(\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)(\tau + \log(4m))(\log n)^{2/\gamma}}{h^d n} + \varepsilon \end{aligned}$$

for any $x \in B_{r_0}$. By setting $\varepsilon = \frac{1}{n}$, we obtain $\log(4m) \lesssim \log \frac{nr_0}{h}$. Thus, with probability μ at least $1 - e^{-\tau}$, we have

$$\begin{aligned} |\mathbb{E}_D \widetilde{k}_{x,h}| &\lesssim \sqrt{\frac{\|f\|_\infty(\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)(\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n} + \frac{1}{n} \\ &\lesssim \sqrt{\frac{\|f\|_\infty(\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n}} + \frac{K(0)(\tau + \log(\frac{nr_0}{h}))(\log n)^{2/\gamma}}{h^d n}. \end{aligned}$$

By taking the supremum of the left hand side of the above inequality over x, we complete the proof of Proposition 15. ∎

Proof [of Theorem 7] Without loss of generality, we assume that $h_n \le 1$. Since $h_n \to 0$, Proposition 10 implies that $\|f_{P,h} - f\|_1 \le \varepsilon$. We set

$$r_n := \left(\frac{nh_n^d}{(\log n)^{(2+2\gamma)/\gamma}}\right)^{1/d} \to \infty \tag{40}$$

and we can also assume w.l.o.g that $r_n \ge 2$. Moreover, there exists a constant n'_1 such that

$$P(B^c_{r_n/2}) \le \varepsilon, \ \forall\, n \ge n'_1.$$

For any $0 < \delta < 1$, we select $\tau := \log(1/\delta)$. Then there exists a constant n_2' such that $\log \frac{nr_n}{h_n} \geq \tau$ for all $n \geq n_2'$. On the other hand, with the above choice of r_n, we have

$$\log \frac{nr_n}{h_n} \leq \log\left(\frac{n^{1/d}h_n}{(\log n)^{(2+2\gamma)/\gamma}} \cdot \frac{n}{h_n}\right) \leq (1+d^{-1})\log n \lesssim \log n.$$

Thus, for all $n \geq \max\{n_1', n_2'\}$, we have

$$\frac{(\log n)^{2/\gamma} r_n^d \log(\frac{nr_n}{h_n})}{nh_n^d} \lesssim \frac{(\log n)^{2/\gamma} r_n^d \log n}{nh_n^d} = \frac{1}{\log n} \to 0.$$

Thus, following from Proposition 14, when n is sufficient large, for any $\varepsilon > 0$, with probability μ at least $1 - 3\delta$, there holds

$$\|f_{D,h_n} - f\|_1 \lesssim \varepsilon.$$

Therefore, with properly chosen δ, one can show that f_{D,h_n} converges to f under L_1-norm almost surely. We have completed the proof of Theorem 7. ■

Proof [of Theorem 8] *(i)* Combining the estimates in Proposition 14 and Proposition 11, we know that with probability μ at least $1 - 2e^{-\tau}$, there holds

$$\begin{aligned}\|f_{D,h} - f\|_1 &\lesssim \sqrt{\frac{(\log n)^{2/\gamma} r^d (\tau + \log(\frac{nr}{h_n}))}{h_n^d n}} + \frac{(\log n)^{2/\gamma} r^d (\tau + \log(\frac{nr}{h_n}))}{h_n^d n} \\ &\quad + \frac{\tau(\log n)^{2/\gamma}}{n} + P\big(B_r^c\big) + r^d h_n^\alpha + \left(\frac{h_n}{r}\right)^\beta \\ &\lesssim \sqrt{\frac{(\log n)^{2/\gamma} r^d (\tau + \log(\frac{nr}{h_n}))}{h_n^d n}} + \frac{\tau(\log n)^{2/\gamma}}{n} + P\big(B_r^c\big) + r^d h_n^\alpha + \left(\frac{h_n}{r}\right)^\beta.\end{aligned}$$

Let $\tau := \log n$ and later we will see from the choices of h_n and r_n that there exists some constant c such that $\log(\frac{nr}{h})$ can be bounded by $c \log n$. Therefore, with probability μ at least $1 - \frac{1}{n}$ there holds

$$\begin{aligned}\|f_{D,h} - f\|_1 &\lesssim \sqrt{\frac{r^d (\log n)^{(2+\gamma)/\gamma}}{h_n^d n}} + r^{-\eta d} + r^d h_n^\alpha \\ &\lesssim r^d \left(\frac{\log n}{nr^d}\right)^{\frac{\alpha}{2\alpha+d}} + r^{-\eta d} \\ &\lesssim \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{\alpha\eta}{(1+\eta)(2\alpha+d)-\alpha}},\end{aligned}$$

by choosing

$$h_n = \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{1+\eta}{(1+\eta)(2\alpha+d)-\alpha}} \quad \text{and} \quad r := r_n = \left(\frac{n}{(\log n)^{(2+\gamma)/\gamma}}\right)^{\frac{\alpha}{d(1+\eta)(2\alpha+d)-\alpha d}}.$$

(ii) Similar to case *(i)*, one can show that with probability μ at least $1 - \frac{1}{n}$ there holds

$$\begin{aligned}\|f_{D,h} - f\|_1 &\lesssim \sqrt{\frac{r^d (\log n)^{(2+\gamma)/\gamma}}{h_n^d n}} + e^{-ar^\eta} + r^d h_n^\alpha \\ &\lesssim r^d \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{nr^d}\right)^{\frac{\alpha}{2\alpha+d}} + e^{-ar^\eta} \\ &\lesssim \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{\alpha}{2\alpha+d}} (\log n)^{\frac{d}{\eta}\cdot\frac{\alpha+d}{2\alpha+d}},\end{aligned}$$

by choosing

$$h_n = \left(\frac{(\log n)^{(2+\gamma)/\gamma}}{n}\right)^{\frac{1}{2\alpha+d}} (\log n)^{-\frac{d}{\eta}\cdot\frac{1}{2\alpha+d}} \quad \text{and} \quad r_n = (\log n)^{\frac{1}{\eta}}.$$

(iii) From Proposition 11 we see that with confidence $1 - \frac{1}{n}$, there holds

$$\|f_{D,h} - f_{P,h}\|_1 \lesssim \sqrt{\frac{r_0^d (\log n)^{(2+\gamma)/\gamma}}{h_n^d n}} + h_n^\alpha \lesssim \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{\alpha}{2\alpha+d}},$$

where h_n is chosen as

$$h_n = \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2\alpha+d}}.$$

The proof of Theorem 8 is completed. ■

Proof [of Theorem 9] The desired estimate is an easy consequence if we combine the estimates in Proposition 15 and Proposition 10 *(ii)* for case (i), Proposition 15 and Proposition 11 *(iv)* for case (ii), respectively, and choose

$$h_n = \left((\log n)^{(2+\gamma)/\gamma}/n\right)^{\frac{1}{2\alpha+d}}.$$

We omit the details of the proof here. ■

To prove the initial Examples 4 and 5, we need the following lemma.

Lemma 25 *Let $g : \mathbb{R} \to [0,\infty)$ be an Lebesgue integrable function with $f(x) = 0$ for all $x \notin [0,1]$ and $f(x) \leq cx^{-\gamma}$ for some constants $c > 0$ and $\gamma \in [0,1)$ and all $x \in (0, 1/2]$. We define $r(x) := |x|/2$ for $x \in \mathbb{R}$. Then for all $h \in (0, 1/8]$ we have the following two inequalities.*

$$\int_{r(x)\leq h} f(x)\,\mathrm{d}x \leq \frac{2c}{1-\gamma} h^{1-\gamma}$$

$$h^{-1}\int_{-h}^{1/2}\int_{r(x)\leq|x'|\leq h} f(x+x')\,\mathrm{d}x'\,\mathrm{d}x \leq \frac{10c}{1-\gamma} h^{1-\gamma}.$$

Proof [of Lemma 25] The first inequality simply follows from

$$\int_{r(x)\le h} f(x)\,\mathrm{d}x = \int_0^{2h} f(x)\,\mathrm{d}x \le c\int_0^{2h} x^{-\gamma}\,\mathrm{d}x \le \frac{2c}{1-\gamma}h^{1-\gamma}\,.$$

For the proof of the second inequality, we first observe that, for $x\in[0,2h]$, we have

$$\begin{aligned}\int\limits_{r(x)\le|x'|\le h} f(x+x')\,\mathrm{d}x' &= \int\limits_{x\le 2|x'|\le 2h} f(x+x')\,\mathrm{d}x'\\ &= \int\limits_{-h}^{-x/2} f(x+x')\,\mathrm{d}x' + \int\limits_{x/2}^{h} f(x+x')\,\mathrm{d}x'\\ &= \int\limits_{-h+x}^{x/2} f(x')\,\mathrm{d}x' + \int\limits_{3x/2}^{h+x} f(x')\,\mathrm{d}x'. \end{aligned} \tag{41}$$

Let us consider the first term in the last equation. For the case where $x\in[0,h)$, we have

$$\int_{-h+x}^{x/2} f(x')\mathrm{d}x' = \int_0^{x/2} f(x')\mathrm{d}x' \le c\int_0^{x/2}(x')^{-\gamma}\mathrm{d}x' = \frac{c}{1-\gamma}\left(\frac{x}{2}\right)^{1-\gamma} \le \frac{c}{1-\gamma}h^{1-\gamma},$$

and for the case where $x\in(h,2h]$, we have

$$\int_{-h+x}^{x/2} f(x')\mathrm{d}x' \le c\int_{-h+x}^{x/2}(x')^{-\gamma}\mathrm{d}x' = \frac{c}{1-\gamma}\left[\left(\frac{x}{2}\right)^{1-\gamma} - (x-h)^{1-\gamma}\right] \le \frac{c}{1-\gamma}h^{1-\gamma}.$$

Therefore, for all $x\in[0,2h]$, we obtain

$$\int_{-h+x}^{x/2} f(x')\mathrm{d}x' \le \frac{c}{1-\gamma}h^{1-\gamma}. \tag{42}$$

On the other hand, for the second term, there holds

$$\begin{aligned}\int_{3x/2}^{h+x} f(x')\mathrm{d}x' \le c\int_{3x/2}^{h+x}(x')^{-\gamma}\mathrm{d}x' &= \frac{c}{1-\gamma}\left[(h+x)^{1-\gamma} - \left(\frac{3x}{2}\right)^{1-\gamma}\right]\\ &\le \frac{c}{1-\gamma}(3h)^{1-\gamma} \le \frac{3c}{1-\gamma}h^{1-\gamma}. \end{aligned} \tag{43}$$

Combining the estimates (41), (42), and (43), we obtain

$$\int\limits_{r(x)\le|x'|\le h} f(x+x')\,\mathrm{d}x' \le \frac{4c}{1-\gamma}h^{1-\gamma}\,,$$

while for $x\in(2h,1/2]$ the integral vanishes since $\{x' : r(x)\le|x'|\le h\}=\emptyset$. Moreover, for $x\in[-h,0)$, we have

$$\begin{aligned}\int\limits_{r(x)\le|x'|\le h} f(x+x')\,\mathrm{d}x' = \int\limits_{-x\le 2|x'|\le 2h} f(x+x')\,\mathrm{d}x' &= \int\limits_{-2h\le 2x'\le x} f(x+x')\,\mathrm{d}x' + \int\limits_{-x}^{2h} f(x+x')\,\mathrm{d}x'\\ &= \int_{-x}^{2h} f(x+x')\,\mathrm{d}x'\\ &\le c\int_0^{2h+x}(x')^{-\gamma}\,\mathrm{d}x'\\ &\le \frac{2c}{1-\gamma}h^{1-\gamma}\,. \end{aligned}$$

We these estimates we then conclude that

$$\int_{-h}^{1/2}\int_{r(x)\le|x'|\le h} f(x+x')\,\mathrm{d}x'\,\mathrm{d}x = \int_{-h}^{2h}\int_{r(x)\le|x'|\le h} f(x+x')\,\mathrm{d}x'\,\mathrm{d}x \tag{44}$$

$$\le \frac{10c}{1-\gamma}h^{2-\gamma}\,, \tag{45}$$

and hence we have shown the assertion. ■

Proof [of Example 4] We can choose $R=1$, $\Omega^{+hR}=\Omega^{+h}=[-h,1+h]$, and

$$X_h^* = (-\infty,-h)\cup(2h,1-2h)\cup(1+h,\infty)\,.$$

Moreover, we have already seen in Example 3 that we can pick $\alpha=1$ with bounded function $x\mapsto c(x)$. Consequently, *(iv)* of Proposition 11 shows

$$\sup_{x\in X_h^*}|f_{P,h}(x)-f(x)| \le c\,h$$

for some constant $c>0$ and all $h>0$. Moreover, Lemma 25 applied to both critical points $x=0$ and $x=1$ with $\gamma=0$ in combination with *(iii)* of Proposition 11 shows

$$\|f_{P,h}-f\|_1 \le c\,h$$

for another constant $c>0$ and all $h\in(0,1/8]$. Proposition 14 then yields the assertion. ■

Proof [of Example 5] We again choose $R=1$, $\Omega^{+hR}=\Omega^{+h}=[-h,1+h]$, and

$$H(\infty) = \int_{\mathbb{R}} c(x)\,\mathrm{d}x < \infty$$

Consequently, Lemma 25 applied to both critical points $x=0$ and $x=1$ with $\gamma=1/2$ in combination with *(iii)* of Proposition 11 shows

$$\|f_{P,h}-f\|_1 \le c\,h^{\alpha}$$

for some constant $c>0$ and all $h\in(0,1/8]$. Again, Proposition 14 then yields the desired result. ■

7. Conclusion

In the present paper, we studied the kernel density estimation problem for dynamical systems admitting a unique invariant Lebesgue density by using the $\mathcal{C}$-mixing coefficient to measure the dependence among observations. The main results presented in this paper are the consistency and convergence rates of the kernel density estimator in the sense of L_1-norm and L_∞-norm. With a properly chosen bandwidth, we showed that the kernel density estimator is universally consistent. Under mild assumptions on the kernel function and the density function, we established convergence rates for the estimator. For instance, when the density function is bounded and compactly supported, both L_1-norm and L_∞-norm convergence rates with the same order can be achieved for general geometrically time-reversed $\mathcal{C}$-mixing dynamical systems. The convergence mentioned here is of type "with high probability" due to the use of a Bernstein-type exponential inequality and this makes the present study different from the existing related studies. We also discussed the model selection problem of the kernel density estimation in the dynamical system context by carrying out numerical experiments.

Acknowledgments

The authors are grateful to Professor László Györfi, the reviewers, and the action editor for helpful comments that helped improve the quality and the presentation of this paper. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC AdG A-DATADRIVE-B (290923). Hanyuan Hang's research is supported by the National Natural Science Foundation of China (Project No.: 11731011). This paper reflects only the authors' views, the Union is not liable for any use that may be made of the contained information. Research Council KUL: GOA/10/09 MaNet, CoE PFV/10/002 (OPTEC), BIL12/11T; PhD/Postdoc grants. Flemish Government: FWO: projects: G.0377.12 (Structured systems), G.088114N (Tensor based data similarity); PhD/Postdoc grants. IWT: projects: SBO POM (100031); PhD/Postdoc grants. iMinds Medical Information Technologies SBO 2014. Belgian Federal Science Policy Office: IUAP P7/19 (DYSCO, Dynamical systems, control and optimization, 2012-2017). The corresponding author is Yunlong Feng.

References

Marian Anghel and Ingo Steinwart. Forecasting the evolution of dynamical systems from noisy observations. *arXiv preprint arXiv:0707.4146*, 2007.

Viviane Baladi. *Positive Transfer Operators and Decay of Correlations*, volume 16 of *Advanced Series in Nonlinear Dynamics*. World Scientific Publishing Co., Inc., River Edge, NJ, 2000.

Viviane Baladi. Decay of correlations. In *Smooth Ergodic Theory and its Applications (Seattle, WA, 1999)*, volume 69 of *Proc. Sympos. Pure Math.*, pages 297–325. Amer. Math. Soc., Providence, RI, 2001.

Delphine Blanke, Denis Bosq, and Dominique Guégan. Modelization and nonparametric estimation for dynamical systems with noise. *Statistical Inference for Stochastic Processes*, 6(3):267–290, 2003.

Denis Bosq and Dominique Guégan. Nonparametric estimation of the chaotic function and the invariant measure of a dynamical system. *Statistics & Probability Letters*, 25(3):201–212, 1995.

Adrian W. Bowman. An alternative method of cross-validation for the smoothing of density estimates. *Biometrika*, 71(2):353–360, 1984.

Richard C. Bradley. Basic properties of strong mixing conditions. A survey and some open questions. *Probability Surveys*, 2(2):107–144, 2005.

Ricardo Cao, Antonio Cuevas, and Wensceslao González Manteiga. A comparative study of several smoothing methods in density estimation. *Computational Statistics & Data Analysis*, 17(2):153–176, 1994.

Bernd Carl and Irmtraud Stephani. *Entropy, Compactness and the Approximation of Operators*. Cambridge University Press, Cambridge, 1990.

Chih-Kang Chu and James S. Marron. Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics*, 19(4):1906–1918, 1991.

Marc Deisenroth and Shakir Mohamed. Expectation propagation in Gaussian process dynamical systems. In *Advances in Neural Information Processing Systems*, pages 2609–2617. NIPS Foundation, 2012.

Luc Devroye. The double kernel method in density estimation. *Annales de l'Institut Henri Poincaré (B) Probabilités et Statistiques*, 25(4):533–580, 1989.

Luc Devroye. Universal smoothing factor selection in density estimation: theory and practice. *Test*, 6(2):223–320, 1997.

Luc Devroye and László Györfi. *Nonparametric Density Estimation: The L_1 View*, volume 119. John Wiley & Sons Incorporated, 1985.

Luc Devroye and Gábor Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and Yatracos classes. *The Annals of Statistics*, 25(6):2626–2637, 1997.

Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer Science & Business Media, 2001.

Paul Eggermont and Vince LaRiccia. *Maximum Penalized Likelihood Estimation: Volume I: Density Estimation*. Springer, New York, 2001.

Gerald B. Folland. *Real Analysis*. John Wiley & Sons, New York, 1999.

Theo Gasser, Hans-Georg Müller, and Volker Mammitzsch. Kernels for nonparametric curve estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 47(2):238–252, 1985.

László Györfi. Strong consistent density estimate from ergodic sample. *Journal of Multivariate Analysis*, 11(1):81–84, 1981.

László Györfi and Gábor Lugosi. Kernel density estimation from ergodic sample is not universally consistent. *Computational statistics & data analysis*, 14(4):437–442, 1992.

László Györfi and Elias Masry. The L_1 and L_2 strong consistency of recursive kernel density estimation from dependent samples. *IEEE Transactions on Information Theory*, 36(3):531–539, 1990.

Peter Hall, Soumendra Nath Lahiri, and Jörg Polzehl. On bandwidth choice in nonparametric regression with both short-and long-range dependent errors. *The Annals of Statistics*, 23(6): 1921–1936, 1995.

Hanyuan Hang and Ingo Steinwart. A Bernstein-type inequality for some mixing processes and dynamical systems with an application to learning. *The Annals of Statistics*, 45(2):708–743, 2017.

Jeffrey D. Hart and Philippe Vieu. Data-driven bandwidth choice for density estimation based on dependent data. *The Annals of Statistics*, 18(2):873–890, 1990.

Jeffrey D. Hart and Thomas E. Wehrly. Kernel regression estimation using repeated measurements data. *Journal of the American Statistical Association*, 81(396):1080–1088, 1986.

Michael C. Jones, James S. Marron, and Simon J. Sheather. A brief survey of bandwidth selection for density estimation. *Journal of the American Statistical Association*, 91(433):401–407, 1996a.

Michael C. Jones, James S. Marron, and Simon J. Sheather. Progress in data-based bandwidth selection for kernel density estimation. *Computational Statistics*, 11(3):337–381, 1996b.

Anatole Katok and Boris Hasselblatt. *Introduction to the Modern Theory of Dynamical Systems*, volume 54 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1995.

Rafail Z. Khas′minskii. A lower bound on the risks of non-parametric estimates of densities in the uniform metric. *Theory of Probability & Its Applications*, 23(4):794–798, 1979.

Andrzej Lasota and Michael C. Mackey. *Probabilistic Properties of Deterministic Systems*. Cambridge University Press, 1985.

Andrzej Lasota and James A. Yorke. On the existence of invariant measures for piecewise monotonic transformations. *Transactions of the American Mathematical Society*, 186:481–488, 1973.

Carlangelo Liverani. Decay of correlations for piecewise expanding maps. *Journal of Statistical Physics*, 78(3):1111–1129, 1995.

Elias Masry. Probability density estimation from sampled data. *Information Theory, IEEE Transactions on*, 29(5):696–709, 1983.

Elias Masry. Recursive probability density estimation for weakly dependent stationary processes. *Information Theory, IEEE Transactions on*, 32(2):254–267, 1986.

Elias Masry and László Györfi. Strong consistency and rates for recursive probability density estimators of stationary processes. *Journal of Multivariate Analysis*, 22(1):79–93, 1987.

Véronique Maume-Deschamps. Exponential inequalities and functional estimation for weak dependent data: applications to dynamical systems. *Stochastics and Dynamics*, 6(4):535–560, 2006.

Kevin McGoff, Sayan Mukherjee, Andrew Nobel, and Natesh Pillai. Consistency of maximum likelihood estimation for some dynamical systems. *The Annals of Statistics*, 43(1):1–29, 2015a.

Kevin McGoff, Sayan Mukherjee, and Natesh Pillai. Statistical inference for dynamical systems: a review. *Statistics Surveys*, 9:209–252, 2015b.

Byeong U. Park and James S. Marron. Comparison of data-driven bandwidth selectors. *Journal of the American Statistical Association*, 85(409):66–72, 1990.

Emanuel Parzen. On estimation of a probability density function and mode. *The Annals of Mathematical Statistics*, 33(3):1065–1076, 1962.

Clémentine Prieur. Density estimation for one-dimensional dynamical systems. *ESAIM: Probability and Statistics*, 5:51–76, 2001.

Peter M. Robinson. Nonparametric estimators for time series. *Journal of Time Series Analysis*, 4 (3):185–207, 1983.

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. *The Annals of Mathematical Statistics*, 27(3):832–837, 1956.

Mats Rudemo. Empirical choice of histograms and kernel density estimators. *Scandinavian Journal of Statistics*, 9:65–78, 1982.

David W. Scott and George R. Terrell. Biased and unbiased cross-validation in density estimation. *Journal of the American Statistical Association*, 82(400):1131–1146, 1987.

Simon J. Sheather and Michael C. Jones. A reliable data-based bandwidth selection method for kernel density estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 53(3):683–690, 1991.

Ingo Steinwart and Marian Anghel. Consistency of support vector machines for forecasting the evolution of an unknown ergodic dynamical system from observations with unknown noise. *The Annals of Statistics*, 37(2):841–875, 2009.

Ingo Steinwart and Andreas Christmann. *Support Vector Machines*. Springer, New York, 2008.

Charles J. Stone. Optimal uniform rate of convergence for nonparametric estimators of a density function or its derivatives. In *Recent Advances in Statistics*, pages 393–406. Academic Press, New York, 1983.

Johan A.K. Suykens and Joos Vandewalle. Recurrent least squares support vector machines. *Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on*, 47(7):1109–1114, 2000.

Johan A.K. Suykens, Joos Vandewalle, and Bart De Moor. *Artificial Neural Networks for Modelling and Control of Non-Linear Systems*. Springer Science & Business Media, 1995.

Johan A.K. Suykens, Tony Van Gestel, Jos De Brabanter, Bart De Moor, and Joos Vandewalle. *Least Squares Support Vector Machines*. World Scientific, Singapore, 2002.

George R. Terrell. The maximal smoothing principle in density estimation. *Journal of the American Statistical Association*, 85(410):470–477, 1990.

Lanh Tat Tran. The L_1 convergence of kernel density estimates under dependence. *The Canadian Journal of Statistics*, 17(2):197–208, 1989a.

Lanh Tat Tran. Recursive density estimation under dependence. *Information Theory, IEEE Transactions on*, 35(5):1103–1108, 1989b.

Matt P. Wand and Chris M. Jones. *Kernel Smoothing*. Chapman & Hall, London, 1994.

Qiwei Yao and Howell Tong. Cross-validatory bandwidth selections for regression estimation based on dependent data. *Journal of Statistical Planning and Inference*, 68(2):387–415, 1998.

Bin Yu. Density estimation in the L^∞-norm for dependent data with applications to the Gibbs sampler. *The Annals of Statistics*, 21(2):711–735, 1993.

Onno Zoeter and Tom Heskes. Change point problems in linear dynamical systems. *The Journal of Machine Learning Research*, 6:1999–2026, 2005.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 09/16; Revised 01/18; Published 09/18

Invariant Models for Causal Transfer Learning

Mateo Rojas-Carulla mr597@cam.ac.uk
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Department of Engineering
Univ. of Cambridge, United Kingdom

Bernhard Schölkopf bs@tuebingen.mpg.de
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Richard Turner ret26@cam.ac.uk
Department of Engineering
Univ. of Cambridge, United Kingdom

Jonas Peters* jonas.peters@math.ku.dk
Department of Mathematical Sciences
Univ. of Copenhagen, Denmark

Editor: Massimiliano Pontil

Abstract

Methods of transfer learning try to combine knowledge from several related tasks (or domains) to improve performance on a test task. Inspired by causal methodology, we relax the usual covariate shift assumption and assume that it holds true for a *subset* of predictor variables: the conditional distribution of the target variable given this subset of predictors is invariant over all tasks. We show how this assumption can be motivated from ideas in the field of causality. We focus on the problem of Domain Generalization, in which no examples from the test task are observed. We prove that in an adversarial setting using this subset for prediction is optimal in Domain Generalization; we further provide examples, in which the tasks are sufficiently diverse and the estimator therefore outperforms pooling the data, even on average. If examples from the test task are available, we also provide a method to transfer knowledge from the training tasks and exploit all available features for prediction. However, we provide no guarantees for this method. We introduce a practical method which allows for automatic inference of the above subset and provide corresponding code. We present results on synthetic data sets and a gene deletion data set.

Keywords: Transfer learning, Multi-task learning, Causality, Domain adaptation, Domain generalization.

1. Introduction

Standard approaches to supervised learning assume that training and test data can be modeled as an i.i.d. sample from a distribution $\mathbb{P} := \mathbb{P}^{(\mathbf{X},Y)}$. The inputs $\mathbf{X}$ are often vectorial, and the outputs Y may be labels (classification) or continuous values (regression). The i.i.d. setting is theoretically well understood and yields remarkable predictive accuracy in problems such as image classification, speech recognition and machine translation (e.g., Schmidhuber, 2015; Krizhevsky et al., 2012). However, many real world problems do not fit into this setting. The field of transfer learning attempts to address the scenario in which distributions may change between training and testing. We focus on two different problems within transfer learning: domain generalization and multi-task learning. We begin by describing these two problems, followed by a discussion of existing assumptions made to address the problem of knowledge transfer, as well as the new assumption we assay in this paper.

1.1 Domain generalization and multi-task learning

Assume that we want to predict a target $Y \in \mathbb{R}$ from some predictor variable $\mathbf{X} \in \mathbb{R}^p$. Consider D training (or source) tasks[1] $\mathbb{P}^1, \dots, \mathbb{P}^D$ where each $\mathbb{P}^k$ represents a probability distribution generating data $(\mathbf{X}^k, Y^k) \sim \mathbb{P}^k$. At training time, we observe a sample $\left(\mathbf{X}_i^k, Y_i^k\right)_{i=1}^{n_k}$ for each source task $k \in \{1, \dots, D\}$; at test time, we want to predict the target values of an unlabeled sample from the task T of interest. We wish to learn a map $f : \mathbb{R}^p \to \mathbb{R}$ with small expected squared loss $\mathcal{E}_{\mathbb{P}^T}(f) = \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T}(Y^T - f(\mathbf{X}^T))^2$ on the test task T.

In domain generalization (DG) (e.g., Muandet et al., 2013), we have $T = D + 1$, that is, we are interested in using information from the source tasks in order to predict Y^{D+1} from $\mathbf{X}^{D+1}$ in a related yet unobserved test task $\mathbb{P}^{D+1}$. To beat simple baseline techniques, regularity conditions on the differences of the tasks are required. Indeed, if the test task differs significantly from the source tasks, we may run into the problem of negative transfer (Pan and Yang, 2010) and DG becomes impossible (Ben-David et al., 2010).

If examples from the test task are available during training (e.g., Pan and Yang, 2010; Baxter, 2000), we refer to the problem as asymmetric multi-task learning (AMTL). If the objective is to improve performance in all the training tasks (e.g., Caruana, 1997), we call the problem symmetric multi-task learning (SMTL), see Table 1 for a summary of these settings. In multi-task learning (MTL), which includes both AMTL and SMTL, if infinitely many labeled data are available from the test task, it is impossible to beat a method that learns on the test task and ignores the training tasks.

1.2 Prior work

A first family of methods assumes that **covariate shift** holds (e.g., Quionero-Candela et al., 2009; Schweikert et al., 2009). This states that for all $k \in \{1, \dots, D, T\}$, the conditional distributions $Y^k \,|\, \mathbf{X}^k$ are **invariant** between tasks. Therefore, the differences in the joint distribution of $\mathbf{X}^k$ and Y^k originate from a difference in the marginal distribution of $\mathbf{X}^k$. Under covariate shift, for instance, if an unlabeled sample from the test task is available at training in the DG setting, the training sample can be re-weighted via importance sampling (Gretton et al., 2009; Shimodaira, 2000; Sugiyama et al., 2008) so that it becomes representative of the test task.

∗. Most of this work was done while JP was at the Max Planck Institute for Intelligent Systems in Tübingen.

1. In this work, we use the expression "task" and "domain" interchangeably.

method	training data from	test domain
Domain Generalization (DG)	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D)$	$T := D + 1$
	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D), \mathbf{X}^{D+1}$	
Asymm. Multi-Task Learning (AMTL)	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D)$	$T := D$
	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D), \mathbf{X}^D$	
Symm. Multi-Task Learning (SMTL)	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D)$	all
	$(\mathbf{X}^1, Y^1), \ldots, (\mathbf{X}^D, Y^D), \mathbf{X}^1, \ldots, \mathbf{X}^D$	

Table 1: Taxonomy for domain generalization (DG) and multi-task learning (AMTL and SMTL). Each problem can either be used without (first line) or with (second line) additional unlabeled data.

Another line of work focuses on **sharing parameters** between tasks. This idea originates in the hierarchical Bayesian literature (Bonilla et al., 2007; Gao et al., 2008). For instance, Lawrence and Platt (2004) introduce a model for MTL in which the mapping f_k in each task $k \in \{1, \ldots, D, T\}$ is drawn independently from a common Gaussian Process (GP), and the likelihood of the latent functions depends on a shared parameter θ. A similar approach is introduced by Evgeniou and Pontil (2004): they consider an SVM with weight vector $w^k = w_0 + v^k$, where w_0 is shared across tasks and v^k is task specific. This allows for tasks to be similar (in which case v^k does not have a significant contribution to predictions) or quite different. Daumé III et al. (2010) use a related approach for MTL when there is one source and one target task. Their method relies on the idea of augmented feature space, which they obtain using two features maps $\Phi^s(\mathbf{X}^s) = (\mathbf{X}^s, \mathbf{X}^s, 0)$ for the source examples and $\Phi^t(\mathbf{X}^t) = (\mathbf{X}^t, 0, \mathbf{X}^t)$ for the target examples. They then train a classifier using these augmented features. Moreover, they propose a way of using available unlabeled data from the target task at training.

An alternative family of methods is based on learning a set of **common features** for all tasks (Argyriou et al., 2007a; Romera-Paredes et al., 2012; Argyriou et al., 2007b; Raina et al., 2007). For instance, Argyriou et al. (2007a,b) propose to learn a set of low dimensional features shared between tasks using L^1 regularization, and then learn all tasks independently using these features. In Raina et al. (2007), the authors construct a similar set of features using L^1 regularization but make use of only unlabeled examples. Chen et al. (2012) proposes to build shared feature mappings which are robust to noise by using autoencoders.

Finally, the assumption introduced in this paper is based on a **causal** view on domain adaptation and transfer.

Schölkopf et al. (2012) relate multi-task learning with the independence between cause and mechanism. This notion is closely related to exogeneity (Zhang et al., 2015b), which roughly states that a causal mechanism mapping a cause X to Y should not depend on the distribution of X. Additionally, Zhang et al. (2013) consider the problem of target and conditional shift when the target variable is causal for the features. They assume that there exists a linear mapping between the covariates in different tasks, and the parameters of this

mapping only depend on the distribution of the target variable. Moreover, Zhang et al. (2015a) argue that the availability of multiple domains is sufficient to drop this previous assumption when the distribution of Y^k and the conditional $\mathbf{X}^k \,|\, Y^k$ change independently. The conditional in the test task can then be written as a linear mixture of the conditionals in the source domains. The concept of invariant conditionals and exogeneity can also be used for causal discovery (Peters et al., 2016; Zhang et al., 2015b; Peters et al., 2017).

1.3 Contribution

Taking into account causal knowledge, **our approach** to DG and MTL assumes that covariate shift holds only for a subset of the features. From the point of view of causal modeling (Pearl, 2009), assuming invariance of conditionals makes sense if the conditionals represent causal mechanisms (e.g., Hoover, 1990), see Section 2.3 for details. Intuitively, we expect that a causal mechanism is a property of the physical world, and it does not depend on what we feed into it. If the input (which in this case coincides with the covariates) shifts, the mechanism should thus remain invariant (Hoover, 1990; Janzing and Schölkopf, 2010; Peters et al., 2016). In the anticausal direction, however, a shift of the input usually leads to a changing conditional (Schölkopf et al., 2012). In practice, prediction problems are often not causal — we should allow for the possibility that the set of predictors contains variables that are causal, anticausal, or confounded, i.e., statistically dependent variables without a directed causal link with the target variable. We thus expect that there is a *subset* S^* of predictors, referred to as an **invariant set**, for which the covariate shift assumption holds true, i.e., the conditionals of output given predictor $Y^k \,|\, \mathbf{X}^k_{S^*}$ are invariant across $k \in \{1, \ldots, D, T\}$. If S^* is a strict subset of all predictors, this relaxes full covariate shift. We prove that knowing S^* leads to robust properties for DG. Once an invariant set is known, traditional methods for covariate shift can be applied as a black box, see Figure 1. In the MTL setting, when labeled or unlabeled examples from the test task are available during training, we might not want to discard the features outside of S^* for prediction. Hence, we also propose a method to leverage the knowledge of the invariant set S^* and the available examples from the test task in order to outperform a method that learns only on the test task.

Finally, note that in this work, we concentrate on the linear setting, keeping in mind that this has specific implications for covariate shift.

1.4 Organization of the paper

Section 2 formally describes our approach and its underlying assumptions; in particular, we assume that an invariant set S^* is known. For DG, we prove in Section 2.1 that predicting using only features in S^* is optimal in an adversarial setting. Moreover, we present an example in which we compare our proposed estimator with pooling the training data, a standard technique for DG. In MTL, when additional labeled examples from T are available, one might want to use all available features for prediction. Section 2.2 provides a method to address this. We discuss a link to causal inference in Section 2.3. Often, an invariant set S^* is not known a priori. Section 3 presents a method for inferring an invariant set from data. Section 4 contains experiments on simulated and real data.

2. Exploiting invariant conditional distributions in transfer learning

Consider a transfer learning regression problem with source tasks $\mathbb{P}^1, \ldots, \mathbb{P}^D$, where $(\mathbf{X}^k, Y^k) \sim \mathbb{P}^k$ for $k \in \{1, \ldots, D\}$.[2] We now formulate our main assumptions.

(A1) There exists a subset $S^* \subseteq \{1, \ldots, p\}$ of predictor variables such that

$$Y^k \,|\, \mathbf{X}^k_{S^*} \overset{d}{=} Y^{k'} \,|\, \mathbf{X}^{k'}_{S^*} \quad \forall k, k' \in \{1, \ldots, D\}. \tag{1}$$

We say that S^* is an **invariant set** which leads to invariant conditionals. Here, $\overset{d}{=}$ denotes equality in distribution.

(A1') This invariance also holds in the test task T, i.e., (1) holds for all $k, k' \in \{1, \ldots, D, T\}$.

(A2) The conditional distribution of Y given an invariant set S^* is linear: there exists $\alpha \in \mathbb{R}^{|S^*|}$ and a random variable ϵ such that for all $k \in \{1, \ldots, D\}$, $[Y^k \,|\, \mathbf{X}^k_{S^*} = x] \overset{d}{=} \alpha^t x + \epsilon^k$, that is $Y^k = \alpha^t \mathbf{X}^k_{S^*} + \epsilon^k$, with $\epsilon^k \perp\!\!\!\perp \mathbf{X}^k_{S^*}$ and for all $k \in \{1, \ldots, D\}$, $\epsilon^k \overset{d}{=} \epsilon$.

Assumption (A1') is stronger than (A1) only in the DG setting, where, of course, (A1') and (A2) imply the linearity also in the test task T. While Assumption (A1) is testable from training data, see Section 3, (A1') is not. In covariate shift, one usually assumes that (A1') holds for the set of all features. Therefore, (A1') is a weaker condition than covariate shift, see Figure 1. We regard this assumption as a building block that can be combined with any method for covariate shift, applied to the subset S^*. It is known that it can be arbitrarily hard to exploit the assumption of covariate shift in practice (Ben-David et al., 2010). In a general setting, for instance, assumptions about the support of the training distributions $\mathbb{P}^1, \ldots, \mathbb{P}^D$ and the test distribution $\mathbb{P}^T$ must be made for methods such as re-weighting to be expected to work (e.g., Gretton et al., 2009). The aim of our work is not to solve the full covariate shift problem, but to elucidate a relaxation of covariate shift in which it holds given only a subset of the features. We concentrate on linear relations (A2), which circumvents the issue of overlapping supports, for example.

For the remainder of this section, we assume that we are given an invariant subset S^* that satisfies (A1) and (A2). Note that we will also require (A1') for DG. In MTL, the invariance can be tested on the labeled data available from the test task, so (A1) and (A1') are equivalent.

We show how the knowledge of S^* can be exploited for the DG problem (Section 2.1) and in the MTL case (Section 2.2). Here and below, we focus on linear regression using squared loss

$$\mathcal{E}_{\mathbb{P}^T}(\beta) = \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T}(Y^T - \beta^t \mathbf{X}^T)^2 \tag{2}$$

(the superscript T corresponds to the test task, not to be confused with the transpose, indicated by superscript t). We denote by $\mathcal{E}_{\mathbb{P}^1, \ldots, \mathbb{P}^D}(\beta)$ the squared error averaged over the training tasks $k \in \{1, \ldots, D\}$.

2. We assume throughout this work the existence of densities and that random variables have finite variance.

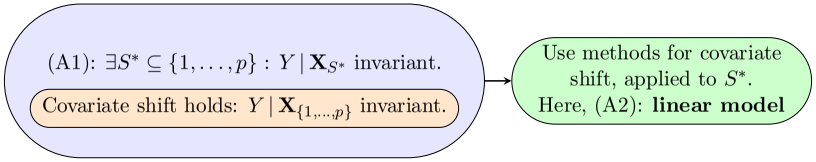


Figure 1: Assumption (A1) (blue) is a relaxation of covariate shift (orange): the covariate shift assumption is a special case of (A1) with $S^* = \{1, \ldots, p\}$. Given the invariant set S^*, methods for covariate shift can be applied.

2.1 Domain generalization (DG): no labels from the test task

We first study the DG setting in which we receive no labeled examples from the test task during training time. Throughout this subsection, we assume that additionally to (A1) and (A2), assumption (A1') holds. It is important to appreciate that (A1') is a strong assumption that is not testable on the training data: it is an assumption about the test task. We believe no nontrivial statement about DG is possible without an assumption of this type.

Now, we introduce our proposed estimator, which uses the conditional mean of the target variable given the invariant set in the training tasks. We prove that this estimator is optimal in an adversarial setting.

Proposed estimator. The optimal predictor obtained by minimizing (2) is the conditional mean

$$\beta^{opt} := \arg\min_{\beta \in \mathbb{R}^p} \mathcal{E}_{\mathbb{P}^T}(\beta), \tag{3}$$

which is not available during training time. Given an invariant set S^* satisfying (A1), (A1') and (A2), we propose to use the corresponding conditional expectation as an estimator. In other words, let $\beta^{S^*} = \arg\min_{\beta \in \mathbb{R}^{|S^*|}} (Y^1 - \beta^t \mathbf{X}^1_{S^*})^2$ be the vector obtained by minimizing the squared loss in the training tasks using only predictors in S^*. We propose as a predictor the vector $\beta^{CS(S^*)} \in \mathbb{R}^p$ obtained by adding zeros to β^{S^*} in the dimensions corresponding to covariates outside of S^*. More formally, we propose to use as a predictor

$$\begin{array}{ccc} \mathbb{R}^p & \to & \mathbb{R} \\ \mathbf{x} & \mapsto & \mathbb{E}[Y^1 \,|\, \mathbf{X}^1_{S^*} = \mathbf{x}_{S^*}] \end{array} \quad \text{and write} \quad \mathbb{E}[Y^1 \,|\, \mathbf{X}^1_{S^*} = \mathbf{x}_{S^*}] = \left(\beta^{CS(S^*)}\right)^t \mathbf{x}. \tag{4}$$

Because of (A1), the conditional expectation in (4) is the same in all training tasks. In the limit of infinitely many data, given a subset S, $\beta^{CS(S)}$ is obtained by pooling the training tasks and regressing using only features in S. In particular, $\beta^{CS} := \beta^{CS(\{1,\ldots,p\})}$ is the estimator obtained when assuming traditional covariate shift.

Optimality in an adversarial setting. In an adversarial setting, predictor (4) satisfies the following optimality condition; as for the other results, the proof is provided in Appendix A. We state and prove a more general, nonlinear version of Theorem 1 in Appendix A.1.

Theorem 1 (Adversarial) *Consider* $(\mathbf{X}^1, Y^1) \sim \mathbb{P}^1$,..., $(\mathbf{X}^D, Y^D) \sim \mathbb{P}^D$ *and an invariant set* S^* *satisfying (A1) and (A2). The proposed estimator satisfies an optimality statement over the set of distributions such that (A1') holds: we have*

$$\beta^{CS(S^*)} \in \arg\min_{\beta \in \mathbb{R}^p} \sup_{\mathbb{P}^T \in \mathcal{P}} \mathcal{E}_{\mathbb{P}^T}(\beta),$$

where $\beta^{CS(S^*)}$ *is defined in* (4) *and* $\mathcal{P}$ *contains all distributions over* $(\mathbf{X}^T, Y^T)$, $T = D + 1$, *that are absolutely continuous with respect to the same product measure* μ *and satisfy* $Y^T \,|\, \mathbf{X}^T_{S^*} \stackrel{d}{=} Y^1 \,|\, \mathbf{X}^1_{S^*}$.

Unlike the optimal predictor β^{opt}, the proposed estimator (4) can be learned from the data available in the training tasks. Given a sample $(\mathbf{X}^k_1, Y^k_1), \ldots, (\mathbf{X}^k_{n_k}, Y^k_{n_k})$ from tasks $k \in \{1, \ldots, D\}$, we can estimate the conditional mean in (4) by regressing Y^k on $\mathbf{X}^k_{S^*}$. Due to (A1), we may also pool the data over the different tasks and use

$$(\mathbf{X}^1_1, Y^1_1), \ldots, (\mathbf{X}^1_{n_1}, Y^1_{n_1}), (\mathbf{X}^2_1, Y^2_1), \ldots, (\mathbf{X}^D_{n_D}, Y^D_{n_D})$$

as a training sample for this regression.

One may also compare the proposed estimator with pooling the training tasks, a standard baseline in transfer learning which corresponds to assuming that usual covariate shift holds. Focusing on a specific example, Proposition 2 in the following paragraph shows that when the test tasks become diverse, predicting using (4) outperforms pooling on average over all tasks.

Comparison against pooling the data. We proved that the proposed estimator (4) does well on an adversarial setting, in the sense that it minimizes the largest error on a task in $\mathcal{P}$. The following result provides an example in which we can analytically compare the proposed estimator with the estimator obtained from pooling the training data, which is a benchmark in transfer learning. We prove that in this setting, the proposed estimator outperforms pooling the data on average over test tasks when the tasks become more diverse.

Let $\mathbf{X}^k_{S^*}$ be a vector of independent Gaussian variables in task k. Let the target Y^k satisfy

$$Y^k = \alpha^t \mathbf{X}^k_{S^*} + \epsilon^k, \tag{5}$$

where for each $k \in \{1, \ldots, D\}$, ϵ^k is Gaussian and independent of $\mathbf{X}^k_{S^*}$. We have $\mathbf{X}^k = (\mathbf{X}^k_{S^*}, Z^k)$, where

$$Z^k = \gamma^k Y^k + \eta^k,$$

for some $\gamma^k \in \mathbb{R}$ and where η^k is Gaussian and independent of Y^k.[3] Moreover, assume that the training tasks are balanced. We compare properties of estimator $\beta^{CS(S^*)}$ defined in Equation (4) against the least squares estimator obtained from pooling the training data. In this setting, the tasks differ in coefficients γ^k, which are randomly sampled. We prove that the squared loss averaged over unseen test tasks is always larger for the pooled approach, when coefficients γ^k are centered around zero. In the case where they are centered around a non-zero mean, we prove that when the variance between tasks (in this case, for coefficients γ^k) becomes large enough, the invariant approach also outperforms pooling the data.

3. Using the notation introduced later in Section 2.3, this corresponds to a Gaussian SEM with DAG shown in Fig. 3.

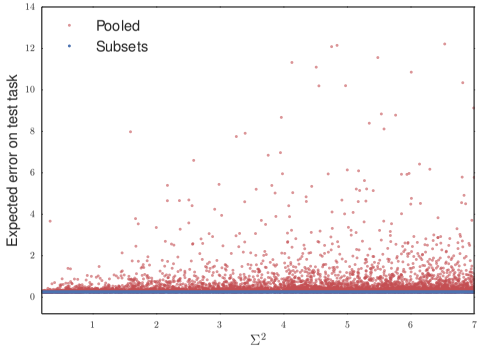


Figure 2: The figure shows expected errors for the pooled approach and the proposed method, see Equation (6). $\mu = 0$. We consider two training tasks over $10,000$ simulations. In each, we randomly sample the variance of each covariate in $\mathbf{X}$, the variance of η, and γ. σ^2 is the same in all tasks. As predicted by Proposition Proposition 2 observe that the error from the pooled approach (red) is systematically higher than the error from the prediction using only the invariant subset (blue), and both the error and its variance become large as the variance Σ^2 of coefficients γ^k increases.

Proposition 2 (Average performance) *Consider the model described previously. Moreover, assume that the tasks differ as follows: the coefficients* $\gamma^1, \ldots, \gamma^D, \gamma^T = \gamma^{D+1}$ *are i.i.d. with mean zero and variance* $\Sigma^2 > 0$. *The tasks do not differ elsewhere. In particular, the distribution of* $X^k_{S^*}$ *is the same for all tasks. Then the least squares predictor obtained from pooling the* D *training tasks* $\beta^{CS} = (\beta^{CS}_{S^*}, \beta^{CS}_Z)$ *satisfies:*

$$\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS}\right)\right) \geq \mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS(S^*)}\right)\right) = \sigma^2. \tag{6}$$

In particular, this implies the following:

$$\mathbb{E}_{\gamma^1,\ldots,\gamma^D,\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS}\right)\right) \geq \mathbb{E}_{\gamma^1,\ldots,\gamma^D,\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS(S^*)}\right)\right) = \sigma^2. \tag{7}$$

Moreover, if the coefficients $\gamma^1, \ldots, \gamma^D, \gamma^T$ *are i.i.d. with non-zero mean* μ, (6) *holds for fixed* $\gamma^1, \ldots, \gamma^D$ *if* $\Sigma^2 \geq P(\mu)$, *where* P *is a polynomial in* μ, *see Appendix A.2 for details.*

The proof of Proposition 2 can be found in Appendix A.2. Figure 2 visualizes Proposition 2 for two training tasks, it shows the expected errors for the pooled and invariant approaches, see (6), as the variance Σ^2 increases. Recall that Σ^2 corresponds to the variance of coefficients γ^k, and thus indicates how different the tasks are. The expected errors are computed using the analytic expression found in the proof of Proposition 2. As predicted by Proposition 2, the expected error of the pooled approach always exceeds the one of the proposed method (the coefficients γ^k are centered around zero), see Equation (6). As Σ^2 tends to

zero, γ^k is close to zero in all tasks, which explains the equality of both the pooled and invariant errors for the limit case Σ approaching 0. For coefficients γ^k centered around a non zero value, Equation (6) does not necessarily hold for small Σ^2.

Proposition 2 presents a setting in which the invariant approach outperforms pooling the data when the test errors are averaged over γ, i.e., $\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS}\right)\right) \geq \mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS}\right)\right)$. It is also clear to see that the equality of the distribution of ϵ^k in Equation (5) for all $k \in \{1,\dots,D\}$ leads to $\mathrm{Var}_\gamma\left(\mathcal{E}_{\mathbb{P}^T}\left(\beta^{CS(S^*)}\right)\right) = 0$, thus our invariant estimator minimizes the variance of the test errors across all related tasks.

2.2 Multi-task learning (MTL): combining invariance and task-specific information

In MTL, a labeled sample $\left(\mathbf{X}_i^T, Y_i^T\right)_{i=1}^{n_T}$ is available from the test task and the goal is to transfer knowledge from the training tasks. As before, we are given an invariant set S^* satisfying (A1) and (A2). Can we combine the invariance assumption with the new labeled sample and perform better than a method that trains only on the data in the test task? According to (A1) and (A2), the target satisfies $Y^k = \alpha^t \mathbf{X}_{S^*}^k + \epsilon^k$, where the noise ϵ^k has zero mean and finite variance, is independent of $\mathbf{X}_{S^*}^k$ and has the same distribution in the different tasks $k \in \{1,\dots,D,T\}$. Our objective is to use the knowledge gained from the training tasks to get a better estimate of β^{opt} defined in Equation (3). We describe below a way to tackle this using missing data methods.

Missing data approach In this section, we specify how we propose to tackle MTL by framing it as a missing data problem. While the idea is presented in the context of AMTL, it can be used for SMTL in the same way. In order to motivate the method, assume that for each $k \in \{1,\dots,D,T\}$, there exists another probability distribution $\mathbb{Q}^k$ with density q^k having the following properties: (i) when restricted to $(\mathbf{X}_{S^*}^k, Y^k)$, $\mathbb{Q}^k$ coincides with $\mathbb{P}^k$, (ii) the conditional $q^T(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$ coincides with $p^T(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$ on the test task and (iii) $q(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N) := q^k(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$ is the same in all tasks (which is not satisfied by $\mathbb{P}^k$, of course). The goal of learning the regression model from Y on $\mathbf{X}_{S^*}$ and $\mathbf{X}_N$ in $\mathbb{P}^T$ coincides with the task of learning the same regression model in $\mathbb{Q}^T$. Property (iii) implies that we can pool the data from all tasks $\mathbb{Q}^k$. This is not possible, of course, for the given data, which we have received from the distributions $\mathbb{P}^k$. But now assume that in all training tasks, we only have access to the marginal $(\mathbf{X}_{S^*}^k, Y^k)$ from $\mathbb{Q}^k$. Any method that addresses the regression under these constraints be used with the data available because of (i). We first prove the existence of such distributions $\mathbb{Q}^k$:

Proposition 3 (Correctness of transfer) *Let S^* be an invariant set verifying (A1) and (A2). For $k \in \{1,\dots,D,T\}$, denote by $(\mathbf{x}, y) \mapsto p^k(\mathbf{x}, y)$ the density of $\mathbb{P}^k$. Then there exists a function $q : \mathbb{R}^p \to \mathbb{R}^+$ such that for each $k \in \{1,\dots,D,T\}$, there exists a distribution $\mathbb{Q}^k$ with density q^k such that for all $(\mathbf{x}, y) \in \mathbb{R}^{d+1}$, for all $k \in \{1,\dots,D,T\}$,*

i) $q^k(\mathbf{x}_{S^*}, y) = p^k(\mathbf{x}_{S^*}, y)$,
ii) $q^T(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N) = p^T(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$,
iii) $q^k(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N) = q(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$.

The proof for Proposition 3 can be found in Appendix A.3. Following the previous intuition, for the training tasks $k \in \{1,\dots,D\}$, we hide the data of $\mathbf{X}_N^k$ and pretend the data in each

task $k \in \{1,\dots,D,T\}$ come from $\mathbb{Q}^k$. Note that some of the data are only missing for the training tasks. More precisely, $\mathbf{X}_N^k$ is missing for $k \in \{1,\dots,D\}$, while because of (i) in Proposition 3, $(\mathbf{X}_{S^*}^k, Y^k)$ is available for all tasks $k \in \{1,\dots,D,T\}$. We thus pool the data and learn a regression model of Y versus $(\mathbf{X}_{S^*}, \mathbf{X}_N)$ by maximizing the likelihood of the observed data.

We formalize the problem as follows. Let $(\mathbf{Z}_i)_{i=1}^n = (\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i}, Y_i)_{i=1}^n$ be a pooled sample of the available data from the training tasks and the test task, in which $\mathbf{X}_{N,i}$ is considered missing if $\mathbf{X}_i$ is drawn from one of the training tasks. Here, $n = \sum_{k=1}^T n_k$ is the total number of training and test examples. Denote by $\mathbf{Z}_{obs,i}$ the components of $\mathbf{Z}_i$ which are not missing. In particular, $\mathbf{Z}_{obs,i} = \mathbf{Z}_i$ if i is drawn from the test task and $\mathbf{Z}_{obs,i} = (\mathbf{X}_{S^*,i}, Y_i)$ otherwise. Moreover, let Σ be a $(p+1) \times (p+1)$ positive definite matrix, and Σ_i is the submatrix of Σ which corresponds to the observed features for example i. If example i is drawn from a training task, Σ_i is of size $(|S^*|+1) \times (|S^*|+1)$, and $(p+1) \times (p+1)$ otherwise. The log-likelihood based on the observed data for matrix Σ satisfies:

$$\ell(\Sigma) = \text{const} - \frac{1}{2}\sum_{i=1}^{n} \det\left(\Sigma_i\right) - \frac{1}{2}\mathbf{Z}_{obs,i}^T \Sigma_i^{-1} \mathbf{Z}_{obs,i}, \tag{8}$$

and our goal is to find Σ which maximizes (8). This model for the likelihood assumes that the data is multi-variate Gaussian with covariance matrix Σ.

When all data are observed, the least squares estimator β^{opt} can be seen as the result of a two step procedure. First, (8) is maximized for the sample covariance matrix. Then, one computes the conditional mean $\mathbb{E}[Y \,|\, \mathbf{X} = \mathbf{x}]$ of the estimated joint distribution of ($\mathbf{X}$,Y). In the case of missing data, however, the sample covariance matrix does no longer maximize (8), see paragraph *'A naive estimator for comparison'* below. Instead, we maximize (8) using EM.

Chapter 11 in Little and Rubin (1986) provides the update equations for optimizing Equation (8) using EM. More precisely, given an estimate Σ^r of the covariance matrix at step r, the algorithm goes as follows.

E step: For an example i, we define

$$\mathbf{Z}_i^r := \begin{cases} \mathbf{Z}_i \text{ if example } i \text{ is from the test task,} \\ (\mathbf{X}_{S^*,i}, \mathbb{E}(\mathbf{X}_N^r \,|\, \mathbf{Z}_{obs,i}), Y_i) \text{ otherwise.} \end{cases}$$

Here, we are essentially imputing the data for $\mathbf{X}_N$ in the training tasks by the conditional mean given the observed data, using the current estimate of the covariance matrix Σ^r. The conditional expectation is computed using the current estimate Σ^r and the Gaussian conditioning formula:

$$\mathbb{E}(\mathbf{X}_N^r \,|\, \mathbf{Z}_{obs,i}) = \Sigma_{NZ_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \mathbf{Z}_{obs,i},$$

where $\Sigma_{NZ_{obs}}^r$ is the submatrix of Σ^r corresponding to the cross-covariance between $\mathbf{X}_N$ and $(\mathbf{X}_{S^*}, Y)$, and $\Sigma_{Z_{obs}}^r$ is the submatrix corresponding to the covariance of $(\mathbf{X}_{S^*}, Y)$. For examples from the test task, we simply copy the example, since $\mathbb{P}^T = \mathbb{Q}^T$. Moreover, define

$$C_{N,i}^r := \begin{cases} 0 \text{ if example } i \text{ is from the test task,} \\ \mathrm{Cov}(\mathbf{X}_N^r \,|\, \mathbf{Z}_{obs,i}) = \Sigma_N^r - \Sigma_{NZ_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \Sigma_{Z_{obs}N}^r \text{ otherwise.} \end{cases}$$

M step: compute the sample covariance given the imputed data:

$$\Sigma^{r+1} = \frac{1}{n}\mathbb{E}\left(\sum_{i=1}^{n} \mathbf{Z}_i^r (\mathbf{Z}_i^r)^t \,|\, \mathbf{Z}_{obs,i}, \Sigma^r\right) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{Z}_i^r (\mathbf{Z}_i^r)^t + C_i^r,$$

where C_i^r is a $(p+1)\times(p+1)$ matrix whose submatrix corresponding to features in N is $C_{N,i}^r$, and the remaining elements are 0. The intuition for the M step is simple: we compute the sample covariance with the values imputed for $\mathbf{X}_N$. Since these values are being imputed, matrix C adds uncertainty for the corresponding values.

Once the algorithm has converged, we can read off the regression coefficient from the joint covariance matrix as $\mathbb{E}[Y \,|\, \mathbf{X}_{S^*} = \mathbf{x}_{S^*}]$. The whole procedure is initialized with the sample covariance matrix computed with the available labeled sample from T.

Incorporating unlabeled data The previous method also allows us to incorporate unlabeled data from the test task. Indeed, assume that an unlabeled sample $\mathbf{X}^T = (\mathbf{X}_{S^*}^T, \mathbf{X}_N^T)$ from the test task is also available at training time. This can be incorporated in the previous framework since the label Y can be considered to be missing (as opposed to $\mathbf{X}_N^T$ previously). We can then write $\mathbf{Z}_i^r = (\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i}, \mathbb{E}(Y_i^r \,|\, \mathbf{Z}_{obs,i}))$ for the unlabeled data, thus imputing the value of Y in in the E-step by the conditional mean given $(\mathbf{X}_{S^*,i}, \mathbf{X}_{N,i})$. The added covariance is then $C_{Y,i}^r = \mathrm{Var}(Y)^r - \Sigma_{YZ_{obs}}^r (\Sigma_{Z_{obs}}^r)^{-1} \Sigma_{Z_{obs}Y}^r$. The rest of the algorithm remains unchanged.

A naive estimator for comparison In the population setting, Proposition 5 in Appendix A.4 provides an expression for β^{opt} as a function of α and ϵ from Assumption (A2). As in the previous paragraph, one could try to estimate the covariance matrix of $(\mathbf{X}, Y)$ using the knowledge of α and ϵ from the training tasks, and then read off the regression coefficients. In the presence of a finite amount of labeled and unlabeled data from the test task, a naive approach would thus plug in the knowledge of α and ϵ as follows: the entries of $\hat{\Sigma}_{\mathbf{X},Y}$ that correspond to the covariances between $\mathbf{X}_{S^*}$ and Y are replaced with $\hat{\Sigma}_{\mathbf{X}_{S^*}} \cdot \alpha$, and the entry corresponding to the variance of Y is replaced by $\alpha^t \hat{\Sigma}_{\mathbf{X}_{S^*}} \alpha + \mathrm{Var}(\epsilon)$. This, however, often performs worse than forgetting about α and using the data in the test domain only, see Figure 5 (left). Why is this the case? The naive solution described above leads to a matrix Σ that does not only *not* maximize (8) but that often is not even positive definite. One needs to optimize over the free parameters of Σ, which corresponds to the covariance between $\mathbf{X}_N$ and Y, given the constraint of positive definiteness. For comparison, we modified the naive approach as follows. First, we find a positive definite matrix satisfying the desired constraints. In order to do this, we solve a semi-definite Program (SDP) with a trivial objective which always equals zero. Then, we maximize the likelihood (8) over the free parameters of Σ with a Nelder-Mead simplex algorithm. The constrained optimization problem can be shown to be convex in the neighborhood of the optimum (Zwiernik et al., 2017, Sec. 3) if the number of data in the test domain grows. While gradients can be computed for this problem, gradient-based methods seem to perform poorly in practice (experiments are not shown for gradient based methods).

In an idealized scenario, infinite amount of unlabeled data in the test and labeled data in the training tasks could provide us with $\Sigma_{\mathbf{X}}$, $\Sigma_{(\mathbf{X}_{S^*},Y)}$ and $\mathrm{Var}(Y)$. We could then plug in these values into Σ and optimize over the remaining parameters, see $\beta^{CS(cau+,i.d.)}$ in

Figure 5 (left). In practice, we have to estimate $\Sigma_{\mathbf{X}}$, $\Sigma_{(\mathbf{X}_{S^*},Y)}$ and $\mathrm{Var}(Y)$ from data. Thus, the EM approach mentioned above constitutes the more principled approach.

2.3 Relation to causality

In this section, we provide a brief introduction to causal notions in order to motivate our method. More specifically, we show that under some conditions, the set S^* of causal parents verifies Assumptions (A1) and (A1'). Structural equation models (SEMs) (Pearl, 2009) are one possibility to formalize causal statements. We say that a distribution over random variables $\mathbf{X} = (X_1, \ldots, X_p)$ is induced by a structural equation model with corresponding graph $\mathcal{G}$ if each variable X_j can be written as a deterministic function of its parents $\mathbf{PA}_j^{\mathcal{G}}$ (in $\mathcal{G}$) and some noise variable N_j:

$$X_j = f_j(X_{\mathbf{PA}_j^{\mathcal{G}}}, N_j)\,, \quad j = 1, \ldots, p\,. \tag{9}$$

Here, the graph is required to be acyclic and the noise variables are assumed to be jointly independent. An SEM comes with the ability to describe *interventions*. Intervening in the system corresponds to replacing one of the structural equations (9). The resulting joint distribution is called an intervention distribution. Changing the equation for variable X_j usually affects the distribution of its children for example, but never the distribution of its parents. Consider now an SEM over variables $(\mathbf{X}, Y)$. Here, we do not specify the graphical relation between Y and the other nodes: Y may or may not have children or parents. Suppose further that the different tasks $\mathbb{P}^1, \ldots, \mathbb{P}^D$ are intervention distributions of an underlying SEM with graph structure $\mathcal{G}$. If the target variable has not been intervened on, then the set $S^* := \mathbf{PA}_Y^{\mathcal{G}}$ satisfies Assumptions (A1) and (A1'). This means that as long as the interventions will not take place at the target variable, the set S^* of causal parents will satisfy Assumptions (A1) and (A1').

Recently, Peters et al. (2016) have given several sufficient conditions for the identifiability of the causal parents in the linear Gaussian framework. E.g., if the interventions take place at informative locations, or if we see sufficiently many different interventions, the set of causal parents is the *only* set S^* that satisfies Assumptions (A1) and (A1'). If there exists more than one set leading to invariant predictions, they consider the intersection of all such subsets. In this sense, seeing more environments helps for identifying the causal structure. In this work, we are interested in prediction rather than causal discovery. Therefore, we try to find a trade-off between models that predict well and invariant models that generalize well to other domains. That is, in the DG setting, we are interested in the subset which leads to invariant conditionals and minimizes the prediction error across training tasks.

If the tasks $\mathbb{P}^k$ correspond to interventions in an SEM, we may construct an extended SEM with a parent-less environment variable E that points into the intervened variables. Then, $\mathbb{P}^k$ equals the distribution of $(\mathbf{X}, Y) \,|\, E = k$, see (Peters et al., 2016, Appendix C). If the distribution of $(\mathbf{X}, Y, E)$ is Markov and faithful w.r.t. the extended graph, the smallest set S that leads to invariant conditionals and to best prediction is a subset of the Markov blanket of Y: certainly, it contains all parents of Y; if it includes a descendant of Y, this must be a child of Y (which yields better prediction and still blocks any path from Y to E); analogously, any contained ancestor of a child of Y must be a parent of that child.

3. Learning invariant conditionals

In the previous section, we have seen how a known invariant subset $S^* \subseteq \{1, \dots, p\}$ of predictors leading to invariant conditionals $Y^k \,|\, \mathbf{X}^k_{S^*}$, see Assumptions (A1) and (A1'), can be beneficial in the problems of DG and MTL. In practice, such a set S^* is often unknown. We now present a method that aims at *inferring* an invariant subset from data. Throughout this paper, we denote by S any subset of features, while S^+ is an invariant set (which is not necessarily unique) for which (A1) holds. Such a subset S^+ *does not necessarily satisfy both Assumptions (A1) and (A1').* Indeed, in DG, only (A1) is testable in the training data. More precisely, if several invariant sets which satisfy (A1) are found, and only some of them satisfy (A1'), we cannot find these from data. We therefore have to add a criterion allowing us to select among several invariant sets. The method we propose provides an estimator $\hat{S}$ for an invariant subset S^+, which is chosen as the subset satisfying Assumption (A1) which maximizes predictive accuracy on a validation set. In MTL, we still write S^+, even if we could then write S^* as (A1') becomes testable. It is summarized in Algorithm 1, code is provided in `https://github.com/mrojascarulla/causal_transfer_learning`.

3.1 Our method.

Algorithm 1: Subset search

Inputs: Sample $(\mathbf{x}_i^k, y_i^k)_{i=1}^{n_k}$ for tasks $k \in \{1, \dots, D\}$, threshold δ for independence test.

Outputs: Estimated invariant subset $\hat{S}$.

Set $S_{acc} = \{\}$, MSE $= \{\}$.
for $S \subseteq \{1, \dots, p\}$ **do**
  linearly regress Y on $\mathbf{X}_S$ and compute the residuals $R_{\beta^{CS(S)}}$ on a validation set.
  compute $H = \text{HSIC}_b\left((R_{\beta^{CS(S)},i}, K_i)_{i=1}^n\right)$ and the corresponding p-value p^* (or the p-value from an alternative test, e.g., Levene test.).
  if $p^* > \delta$ **then**
    compute $\widehat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$, the empirical estimate of $\mathcal{E}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$ on a validation set.
    S_{acc}.add(S), MSE.add($\widehat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$)
  end
end
Select $\hat{S}$ according to *RULE*, see Section 3.4.

Consider a set of D tasks, a target variable Y^k and a vector $\mathbf{X}^k$ of p predictor variables in task k. For $\beta \in \mathbb{R}^p$, we define the residual in task k as:

$$R^k_\beta = Y^k - \beta^t \mathbf{X}^k, \quad k \in \{1, \dots, D\}. \tag{10}$$

By Assumptions (A1) and (A2), there exists a subset S^+ and some vector $\beta^{CS(S^+)}$ such that for all $j \notin S^+$, $\beta_j^{CS(S^+)} = 0$ and $R^1_{\beta^{CS(S^+)}} \overset{d}{=} \dots \overset{d}{=} R^D_{\beta^{CS(S^+)}}$. Such a set S^+ is not necessarily unique. As stated in (Peters et al., 2016), the number of invariant subsets decreases as more

Algorithm 2: Greedy subset search

Inputs: Sample $(\mathbf{x}_i^k, y_i^k)_{i=1}^{n_k}$ for tasks $k \in \{1, \dots, D\}$, threshold δ for independence test.

Outputs: Estimated invariant set $\hat{S}greedy$.

Set $S_{acc} = \{\}$, $\hat{S}_{current}\{\}$, MSE $= \{\}$.
for $i \in \{1, \dots, n_{iters}\}$ **do**
  Set $stat_{min} = \infty$.
  for $S \in \mathcal{S}_{\hat{S}_{current}}$ **do**
    linearly regress Y on $\mathbf{X}_S$ and compute the residuals $R_{\beta^{CS(S)}}$ on a validation set.
    compute $H = \text{HSIC}_b\left((R_{\beta^{CS(S)},i}, K_i)_{i=1}^n\right)$ and the corresponding p-value p^* (or the p-value from an alternative test, e.g. Levene test.).
    if $p^* > \delta$ **then**
      compute $\widehat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$, the empirical estimate of $\mathcal{E}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$ on a validation set.
      S_{acc}.add(S), MSE.add($\widehat{\mathcal{E}}_{\mathbb{P}^{1,\dots,D}}(\beta^{CS(S)})$),
      set $\hat{S}_{current} = S$.
    end
    else if $H < stat_{min}$ **then**
      set $\hat{S}_{current} = S$, $stat_{min} = H$.
    end
  end
end
Select $\hat{S}$ according to *RULE*, see Section 3.4.

different tasks are observed at training time. We propose to do an exhaustive search over subsets S of predictors and statistically test for equality of the distribution of the residuals in the training tasks, see the section below. Among the accepted subsets, we select the subset $\hat{S}$ which leads to the smallest error on a validation set. This is a fundamental difference to the method proposed by Peters et al. (2016). Indeed, while our method addresses the transfer problem, Peters et al. (2016) is about causal discovery. Algorithm 1 finds an invariant subset which also leads to the lowest validation error. This subset may contain covariates which are non causal, see Section 4.3 for further details. On the other hand, Peters et al. (2016) estimate the causal parents (with coverage guarantee). Such an approach has a different purpose and performs very badly both in DG and MTL: e.g., when all tasks are identical, it uses the empty set as predictors, while our method selects the full set of predictors.

In Section 3.3, we propose two solutions for when the number of predictors p is too large for an exhaustive search: a greedy method and variable selection. While the algorithms are presented using linear regression, the extension to a nonlinear framework is straightforward. In particular, linear regression can be replaced by a nonlinear regression method.

3.2 Statistical tests for equality of distributions.

In order to test whether a subset S leads to invariant conditionals, we can use a statistical test to check whether the residuals $R^k_{\beta^{CS(S)}}$ have the same distribution in all tasks $k \in \{1, \dots, D\}$. We propose two possible methods.

For Gaussian data, one can use a Levene test (Levene, 1960) to test whether the residuals have the same variance in all tasks; their means are zero as long as an intercept is included in the regression model.

As an alternative, we propose a nonparametric D-sample test by testing whether the residuals are independent of the task index. This test is a direct application of HSIC (Gretton et al., 2007) but to our knowledge, is novel. Suppose that the index of the task can be considered as a random variable K. We consider the sample $Z = (R_{\beta^S,i}, K_i)_{i=1}^n$ as drawn from a joint distribution over residuals and task indices, where $n = \sum_{k=1}^D n_k$ and $K_i \in \{1, \dots, D\}$ is a discrete value indicating the index of the corresponding task. The residuals have the same distribution in all training tasks if and only if R_{β^S} and K are independent. Two characteristic kernels are used: a kernel κ is used for embedding the residuals and a trivial kernel d such that $d(i, j) = \delta_{ij}$ is used for K. Let therefore $\mathrm{HSIC}(R_{\beta^S}, K)$ denote the value of the HSIC (Gretton et al., 2007) between R_{β^S} and K, and let $\mathrm{HSIC}_b(Z)$ be the corresponding test statistic. A subset S is accepted if if leads to accepting the null hypothesis of independence between R_{β^S} and K at level δ.

Both in the case of the Levene test and the D-sample test, the test outputs a p-value p^*, and we accept the null H_0 if $p^* > \delta$. Among these accepted subsets, we output the set $\hat{S}$ which leads to the smallest loss on a validation set. The test level δ is given as an input to our method and allows for a trade-off between predictive accuracy and exploiting invariance. As δ tends to zero, the null is accepted for all subsets and we then select all features, which is equivalent to covariate shift. When δ approaches one, no subset is accepted as invariant. Our method then reduces to the mean prediction. In order to compute p-values, a Gamma approximation is used for the distribution of $\mathrm{HSIC}_b(Z)$ under the null.

For non-additive models, one may even apply a conditional independence test (e.g., Zhang et al., 2011; Fukumizu et al., 2008) to test whether K is independent of $Y \,|\, \mathbf{X}_S$.

3.3 Scalability to a large number of predictors

When the number of features p is large, full subset search is computationally not feasible. We propose two solutions for this scenario. If one has reasons to believe that the signal is sparse, that is the true set S^* is small, one may use a **variable selection** technique such as the Lasso (Tibshirani, 1996) as a first step. Under the assumptions described in Section 2.3, we know that the invariant set with the best prediction in the training tasks can be assumed to be a subset of the relevant features (which here equals the Markov blanket of Y). Thus, if variable screening is satisfied ,i.e., one selects all relevant variables and possibly more, the pre-selection step does not change the result of Algorithm 1 in the limit of infinitely many data. For linear models with ℓ_1 penalization, variable screening is a well studied problem, see, e.g., compatibility and β_{min} conditions (Bühlmann and van de Geer, 2011, Chapter 2.5).

Alternatively, one may perform a **greedy search** over subsets when full subset search is not feasible. Denote by $\mathcal{S}_S$ the collection of neighboring sets of a set S obtained by adding or

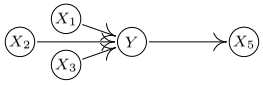


Figure 3: Example of a directed acyclic graph, see Section 2.3. If Y is not intervened on, the conditional $Y \,|\, X_1, X_2, X_3$ remains invariant.

removing exactly one predictor in S. If no subset has been accepted at a given iteration, we select the neighbor leading to the smallest test statistic. If a neighbor is accepted, we select the one which leads to the smallest training error. We start with the p subsets with only one element, and allow to add or remove a single predictor at each step, see Algorithm 2. As often for greedy methods, there is no theoretical guarantee.

3.4 Subset selection in MTL

In DG, among the accepted subsets, we select the set $\hat{S}$ which leads to the lowest validation error. In MTL, however, a labeled sample from the test task T is available at training time. Therefore, Algorithm 1 is slightly modified. First, we get all the sets for which H_0 is accepted. Then, we select the accepted set $\hat{S}$ which leads to the smallest 5 fold cross validation error. For each subset, we compute the least squares coefficients using the procedure described in Section 2.2, and measure the prediction error on the held out validation set. Using the notation of Algorithm 1, let S_{acc} be the set of subsets accepted as invariant, and let MSE be the set of their corresponding squared errors on the validation set. The following rules are used for selecting an invariant set in DG and MTL.

i) ***RULE* for DG**: Return $\hat{S} = S_{acc}[\arg\min \mathrm{MSE}]$.
ii) ***RULE* for MTL**: Define $CV_{acc} = \{\}$. For each set $S \subseteq S_{acc}$, do $CV_{acc}.\mathrm{add}(CV_S)$, where CV_S is the 5-fold cross validation error over the labeled test data obtained by optimizing (8) using EM with subset S.
Return $\hat{S} = S_{acc}[\arg\min CV_{acc}]$.

Given a set of $k \in \{1, \dots, T\}$ training tasks, a collection of sets $\hat{S}_1, \dots, \hat{S}_u$ (eventually empty) is obtained, all of which lead to accepting the null hypothesis of invariance between the training tasks in DG. Our methods use the MSE on a validation set as a criterion for selecting a subset among these u candidates. This is a design choice which is dependent on the specific application, and can be modified. For instance, if being conservative is important, the MSE may be an inappropriate choice. One may be then interested in combining confidence intervals for the accepted sets. One idea is to consider all accepted sets at the same time, one of which is, with probability $1-\alpha$, the set S^* from Assumption (A1'). These sets yield different predictions, one of which stems from S^*, again, with probability $1-\alpha$. In some settings, it might be helpful to output the whole set of predictions. If one is interested in confidence intervals, these may be combined by taking its union. Heinze-Deml et al. (2018) discuss this idea in the context of prediction under interventions.

estimator	description
$\beta^{CS(cau)}$	Linear regr. with true causal predictors (often unknown in practice).
$\beta^{CS(\hat{S})}$	Finding the invariant set $\hat{S}$ using full subset search and performing lin. regr. using predictors in $\hat{S}$. $\hat{S}greedy$ corresponds to finding the invariant set using a greedy procedure. $\hat{S}Lasso$ corresponds to doing variable selection using Lasso as a first step, then doing full subset search on the selected features.
β^{CS}	Pooling the training data and using linear regr.
$\beta^{CS(\hat{S}+)}$	Finding the invariant set $\hat{S}$ using full subset search and solve the optimization problem described in 'A naive estimator for comparison'.
$\beta^{CS(\hat{S}\sharp)}$	Finding the invariant set $\hat{S}$ using full subset search and maximizing (8) for MTL using EM.
β^{mean}	Pooling the training data and outputting the mean of the target.
β^{dom}	Linear regression using only the available labeled sample from T.
β^{MTL}	Multi-task feature learning estimator (Argyriou et al., 2007a).
β^{DICA}	DICA (Muandet et al., 2013) with rbf kernel.
β^{mDA}	Pooling the training data and an unlabeled sample from T, learning features using mSDA (Chen et al., 2012) with one layer and linear output, then using linear regr.

Table 2: Estimators used in the numerical experiments. A '+' next to a subset S corresponds to the method for MTL described in the last paragraph of Section 2.2.

4. Experiments

We compare our estimator to different methods, which are summarized in Table 2. $\beta^{CS(cau)}$ uses the ground truth for S^* when it is available, $\beta^{CS(\hat{S})}$ corresponds to full search using Algorithm 1, β^{CS} uses the pooled training data, β^{MTL} performs the Multi-task feature learning algorithm (Argyriou et al., 2007a) for the MTL setting and β^{DICA} performs DICA (Muandet et al., 2013) for DG. For DICA, which is a nonlinear method, the kernel matrices are constructed using an rbf kernel, and the length-scale of the kernel is selected according to the median heuristic. In the MTL setting, we combine the invariance with task specific information by optimizing (8) using EM, resulting in regression coefficients $\beta^{CS(\hat{S}\sharp)}$ and $\beta^{CS(cau\sharp)}$ when the ground truth is known. Finally, $\beta^{CS(cau\sharp,UL)}$ indicates that unlabeled data from T was also available. For reference, Figure 5 (left) provides results for $\beta^{CS(\hat{S}+)}$ and $\beta^{CS(cau+)}$, which correspond to the estimators obtained by solving the constrained optimization problem described in the paragraph *'A naive estimator for comparison'* of Section 2.2 ($\beta^{CS(cau+)}$ uses the ground truth for S^* and α), while β^{naive} imputes the covariance matrices but does not optimize the free parameters. $\beta^{CS(cau+,i.d.)}$ (infinite data) also assumes that we know the ground truth for the entries of the covariance matrix for the test task corresponding to the covariance of $\mathbf{X}$, the covariance between $\mathbf{X}_{S^*}$ and Y, and the variance of Y.

4.1 Synthetic data set

In this section, we generate a synthetic data set in which the causal structure of the problem is known. For all experiments, we choose $\delta = 0.05$ as a rejection level for the statistical test in Algorithms 1 and 2. Moreover, we use 40% of the training examples to fit the linear

models in Algorithms 1 and 2, and the remaining data as validation. The sensitivity to the choice of δ is discussed in Section 4.2.

Generative process of the data For each task $k \in \{1, 2, \ldots, D, T\}$, we sample a set of causal variables from a multivariate Gaussian

$$\mathbf{X}^k_{S^*} \sim \mathcal{N}(0, \Sigma^k_{S^*})$$

where the covariance matrix $\Sigma^k_{S^*}$ is drawn from a Wishart distribution $\mathcal{W}(U^k_{S^*}, p)$, where $U^k_{S^*}$ is computed as $V^k(V^k)^t$. Here, V^k is a $(|S|, |S|)$ matrix of standard Gaussian random variables.

The target variable Y^k is drawn as

$$Y^k = \alpha \mathbf{X}^k_{S^*} + \epsilon^k$$

where $\epsilon^k \sim \mathcal{N}(0, 2)$ (the standard deviation of ϵ^k is 6 for the non sparse DG experiment with 30 predictors, see the bottom of Figure 4).

We sample the remaining predictor variables as

$$\mathbf{X}^k_N = \gamma^k Y^k + \beta^k (\mathbf{X}^k_{S^*})_C + \eta^k$$

where $\eta^k \sim \mathcal{N}(0, \Sigma^k_N)$. $(\mathbf{X}^k_{S^*})_C$ is a subset of $\mathbf{X}^k_{S^*}$ of size $|C|$ which generates both the target Y^k and $\mathbf{X}_N T^k$. γ^k of size $|N|$ is computed as $\gamma^k = (1-\lambda)\gamma_0 + \lambda g^k$, where $\lambda \in [0, 1]$, γ_0 is the same in all tasks while g^k is task dependent. Both γ_0 and g^k are drawn from a standard Gaussian. Similarly to γ^k, β^k is a $(|C|, |N|)$ matrix computed as $\beta^k = (1-\lambda)\beta_0 + \lambda b^k$. Σ^k_N is sampled similarly to $\Sigma^k_{S^*}$. Finally, α is sampled from a standard Gaussian distribution.

The generative process and hyper-parameters are the same for all the experiments (DG and MTL).

Results Our goal is to linearly predict target Y^T using predictors $\mathbf{X}^T = (\mathbf{X}^T_{S^*}, \mathbf{X}^T_N)$ on the test task. Given regression coefficient β, we measure the performance in the test task using the logarithm of the empirical estimator of $\mathcal{E}_{\mathbb{P}^T}(\beta)$.

In Figure 4, we are in the DG setting (thus, no labeled examples from T are observed at training). 4000 examples per training task are available for the top left and right plots, while only 1000 examples per task are available on the bottom because of computational reasons. We report the log average empirical MSE over left out test tasks. We study both sparse and non sparse settings (in which full search is not feasible). On the upper left and upper right, we see that when more than four training tasks are available, both the full search and greedy approaches are able to recover an invariant set, and outperform pooling the data for any number of training tasks. When more than five training tasks are observed, $\beta^{CS(\hat{S})}$ performs like $\beta^{CS(cau)}$, which uses knowledge of the ground truth. On the bottom, full search is not feasible, and $\beta^{CS(\hat{S}greedy)}$ outperforms other approaches.

In Figure 5 (top left), we consider an AMTL setting, in which large amounts of labeled data (36000) from the training tasks and unlabeled data from the test task (50000) are available. Both S and N are of size 3, such that $\mathbf{X}$ is 6-dimensional. For all MTL experiments, 6 training tasks are available. We report the percentage of simulations for which the population MSE of a given approach outperforms β^{dom}. We see that $\beta^{CS(cau+,i.d.)}$ systematically outperforms β^{dom}. Moreover, $\beta^{CS(cau+)}$ and $\beta^{CS(\hat{S}+)}$ also perform well, and positive

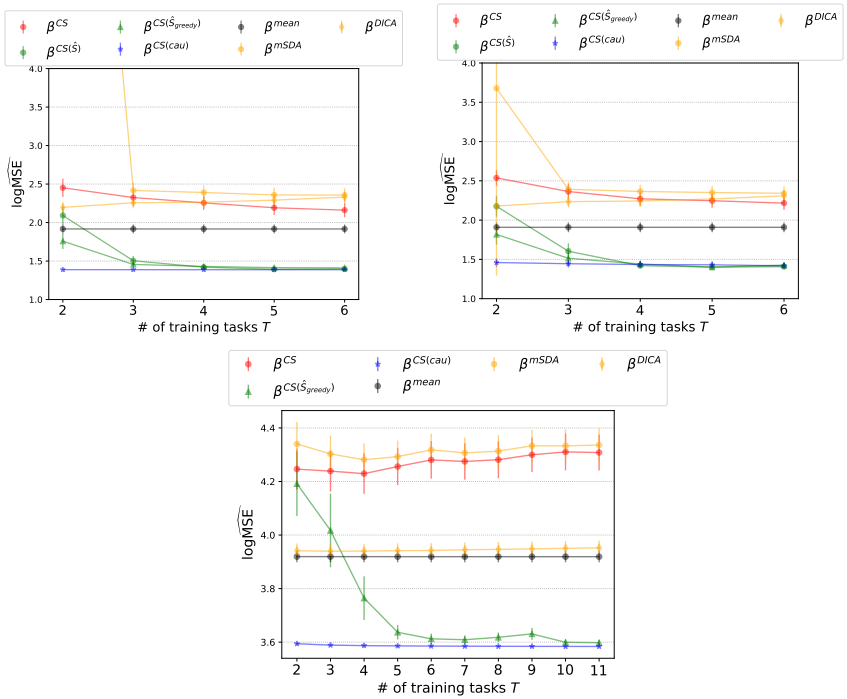


Figure 4: DG setting. Logarithm of the empirical squared error in the test task for the different estimators in the DG setting. The results show averages and 95% confidence intervals for the mean performance over 100 repetitions. We vary the number of tasks D available at training time. Upper left: both S and N are of size 3, such that $\mathbf{X}$ is 6-dimensional. $|C|$ is of size one. Upper right: 30 noise variables are added to $\mathbf{X}$. Variable selection using the Lasso is used prior to computing $\beta^{CS(\hat{S})}$, while $\beta^{CS(\hat{S}greedy)}$ uses all predictors. Bottom: both S and N are of size 15. Full search is not computationally feasible in this setting and only the greedy procedure can be used. Other methods such as β^{CS}, β^{mSDA} and β^{DICA} often perform badly, which explains why in comparison β^{mean} appears to performs well.

transfer is effective. However, a prohibitively large amount of labeled and unlabeled data is needed for these approaches, and the differences become non-significant for all methods except $\beta^{CS(cau+,i.d.)}$. This shows the limitation of this family of approaches. In a setting with only 900 examples per training task in SMTL, we plot in Figure 6 the histogram of the error difference $\Delta = \mathcal{E}(\beta^{dom}) - \mathcal{E}(\beta)$ for $\beta^{CS(cau\sharp)}$. Figure 5 (top right) corresponds to the same setting, but we vary the number of unlabeled data available (we only plot methods

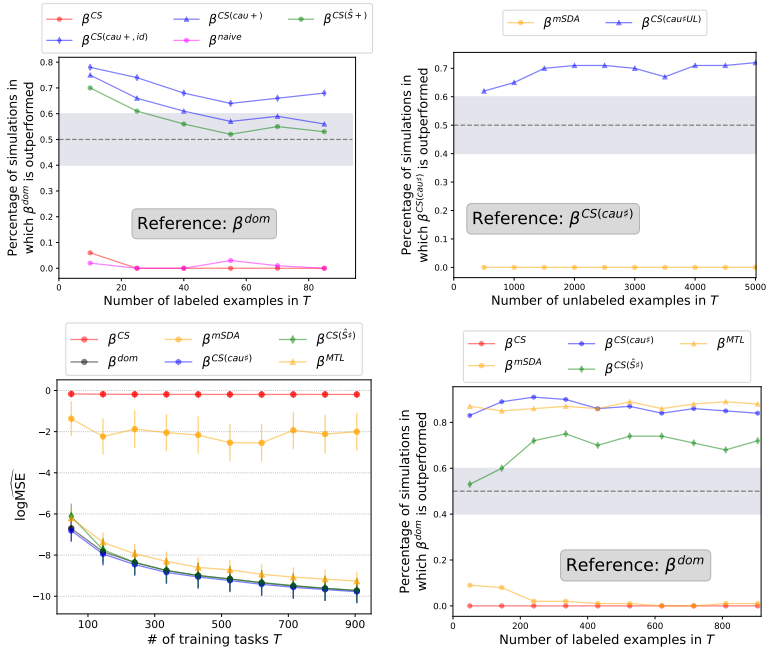


Figure 5: MTL setting. Percentage of repetitions (out of 100) for which the corresponding method outperforms β^{dom} (or $\beta^{CS(cau\sharp)}$ for the top right plot). Both S and N are of size 3, such that $\mathbf{X}$ is 6-dimensional. Upper left: AMTL setting. This plot shows that the methods $\beta^{CS(\hat{S}+)}$ and $\beta^{CS(cau+)}$ presented in Section 2.2 perform well, but a large amount of data is necessary: 50000 unlabeled examples from T and 36000 training examples are available. The naive method β^{naive} performs poorly. Upper right: in the SMTL setting, we fix the number of training data (500 per task) and vary the amount of unlabeled data available from the test task. We report the percentage of scenarios in which the corresponding method outperforms $\beta^{CS(cau\sharp)}$ this time (which uses no unlabeled data). While β^{mDA} always performs worse than $\beta^{CS(cau\sharp)}$ and does not exploit the unlabeled data, we see that $\beta^{CS(cau\sharp,UL)}$ performs better as the amount of unlabeled data increases. Bottom: SMTL setting, and we vary the number of labeled examples available in each training task. Here, significantly less labeled data was available in the training tasks (from 50 to 1000 per task). In this setting, the methods using unlabeled data were given 100 unlabeled examples. Bottom left: logarithm of the empirical squared error in the test task for different estimators. Bottom right: percentage of repetitions (out of 100) for which the corresponding method outperforms β^{dom}.

that use unlabeled data, and $\beta^{CS(cau\sharp)}$ is used as reference instead of β^{dom}). In Figure 5 (bottom) we consider an SMTL setting in which only 100 unlabeled data points are avail-

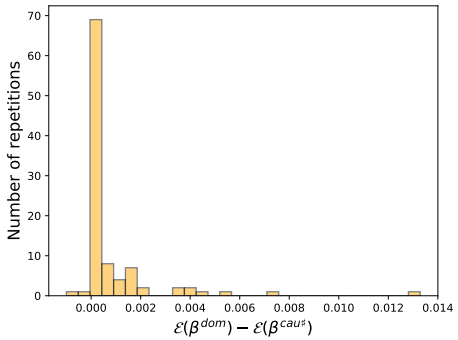


Figure 6: In the SMTL setting, 900 examples from each of the training tasks are available (this corresponds to the data point furthest to the right in the bottom plot of Figure 5). We run 100 repetitions and plot the histograms of $\Delta = \mathcal{E}(\beta^{dom}) - \mathcal{E}(\beta^{CS(cau\sharp)})$. The proposed estimator outperform β^{dom}: for a large proportion of the repetitions, $\Delta > 0$. More importantly, the distribution of Δ is heavily skewed in the positive values. In other words, when β^{dom} outperforms $\beta^{CS(cau\sharp)}$, the difference in performance is small, while the difference is often larger for the converse.

able, and only few labeled examples are available in each task. Here, we see that $\beta^{CS(cau\sharp)}$, $\beta^{CS(\hat{S}\sharp)}$ and β^{MTL} perform well, while other methods do not. In terms of MSE (bottom left), the difference in performance between the top competing methods is not statistically significant.

Time complexity The most expensive component of our method is the estimation of the invariant subset. In the DG experiment in Figure 4, with $n = 4000$ examples available for each of the 6 tasks, and $p = 6$ predictors, full subset search takes 0.067 seconds and greedy search 0.037, where the results are averaged over 100 repetitions. With $p = 10$, full search averages at 1.57 seconds, and greedy search 0.0396. With $p = 30$, where full search is not feasible, greedy search averages at 1.21 seconds. In the MTL experiment in Figure 5, the EM algorithm runs for 0.00105 seconds on average over 100 repetitions. As a reference, in MTL, linear regression averages at 0.000301 seconds and mSDA at 0.0547 seconds.

4.2 Sensitivity to the acceptance level δ

Both Algorithm 1 and its greedy version Algorithm 2 receive an acceptance level δ as input for the statistical test. In our other experiments, we chose the standard value of $\delta = 0.05$. Figure 8 shows the error on the test tasks in the DG setting for both methods for different values of δ. The setting is the same as in the left of Figure 4 for three training tasks. β^{CS} and $\beta^{CS(cau)}$ are provided as reference. For $\delta = 0$, all subsets are accepted as invariant,

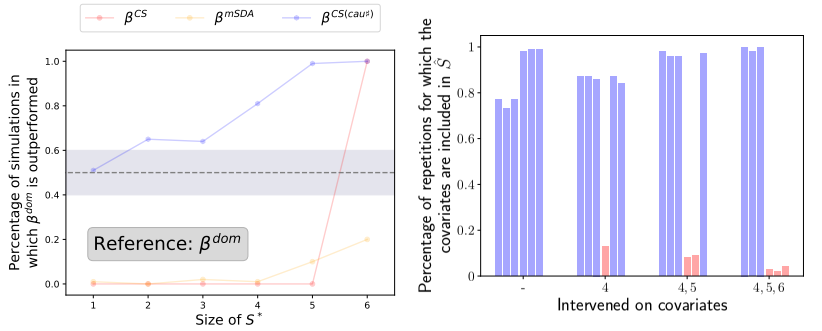


Figure 7: *Left*: SMTL setting with 6 tasks and 900 examples per task. We plot the percentage of repetitions (over 100) for which the given methods outperform β^{dom}, as a function of the size of the invariant set S^*. We see that as S^* becomes larger, more information is transferred from the training tasks, and as such the performance of $\beta^{CS(cau\sharp)}$ improves. When S^* is the full set, our method behaves like pooling the data. *Right*: Covariates selected by Algorithm 1 when the training tasks contain interventions only on some of the covariates. The bars represent the percentage of repetitions (out of 100) for which the corresponding covariates were selected. When there are no interventions in the training tasks, meaning that all the training tasks follow the same distribution, Algorithm 1 systematically selects *all* covariates for prediction. When more interventions are performed, however, the corresponding covariates (in red) are excluded in a large number of the repetitions.

thus both methods behave like pooling the data. After a critical value of δ, no subset is accepted, and both algorithms return the subset with the largest p-value.

4.3 Informativeness and subset estimation

The estimation of an invariant subset involves finding a subset for which the residuals have the same distribution across tasks. It is desirable, however, that the selected subset is one which explains the data best. This is ensured by selecting the subset which leads to the smallest error on a validation set. Therefore, some covariates in N may be included in a selected subset *if there are no interventions on this covariates in the training tasks*. More precisely, if including a covariate does not lead to a statistically measurable difference in the distribution of the residuals between the training tasks, it is advantageous in general to include it in the selected subset since the data is better explained.

We illustrate this in Figure 7 (right) in the setting previously described with $p = 6$. We estimate an invariant subset using Algorithm 1 over 100 repetitions in the following scenarios: i) all the covariates have the same distribution across tasks, ii) one, two or three covariates in N are subject to interventions between the tasks. Figure 7 (right) show the proportion of repetitions for which each covariate is included in the selected subset. We see that, as expected, covariates in N for which there are no interventions are included

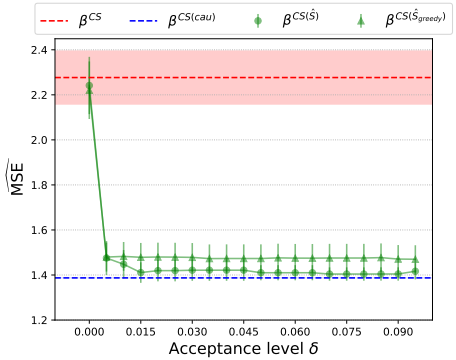


Figure 8: Logarithm of the empirical squared error in the test task in the DG setting as a function of the acceptance level of the statistical test δ in Algorithm 1. The setup corresponds to $t = 3$ in Figure 4 (left), also over 100 repetitions. For $\delta = 0$, all subsets are accepted, so the full set of predictors, which minimizes the validation squared error, is selected. Algorithm 1 then returns β^{CS}. As δ increases, no subset is accepted, and Algorithm 1 returns the subset with the largest p-value.

in the selected subset in a large portion of the repetitions, while the other covariates are excluded. This highlights that Algorithm 1 can only exclude covariates whose distribution shifts between training tasks. If being conservative is important for the problem at hand, one can modify Algorithm 1 accordingly, see the end of Section 3.4.

Moreover, in Figure 7 (left) we consider a similar setting, and we compute the performance against β^{dom} in an SMTL setting as the size of the invariant set increases. We see that as the size of the invariant set increases, the performance of $\beta^{CS(cau^\sharp)}$ improves, since more information is being transferred from the training tasks. When $p = 6$, traditional covariate shift holds, and $\beta^{CS(cau^\sharp)}$ performs on par with β^{pool}.

4.4 Gene perturbation experiment

We apply our method to gene perturbation data provided by Kemmeren et al. (2014). This data set consists of the m-RNA expression levels of $p = 6170$ genes $X_1, \ldots, X_p$ of the Saccharomyces cerevisiae (yeast). It contains both $n_{obs} = 160$ observational data points and $n_{int} = 1479$ data points from intervention experiments. In each of these interventions, one known gene (out of p genes) is deleted. In the following, we consider two different tasks. The observational sample is drawn from the first task, and the pooled n_{int} interventions are drawn from the second task.

Motivation In order to gain an intuition about the experiments we are presenting, consider Figure 9. We select as a target a gene Y out of the p genes, and our goal is to predict the activity of Y given the remaining $p-1$ genes as features. Some of these $p-1$ genes are

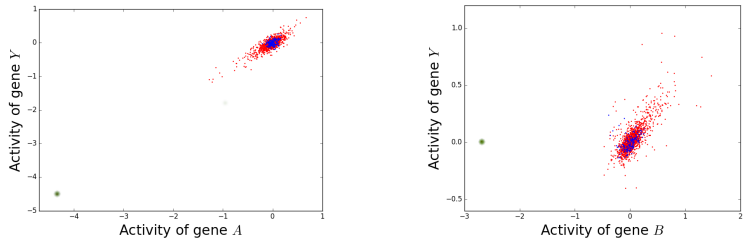


Figure 9: Example of the expression of pairs of genes, where A is causal (left) and B is non-causal (right) of target Y. The blue points are from the observational sample (task 1), the red dots are the interventional sample (task 2), and the green point corresponds to the single interventions in which A and B are intervened on respectively. On the left, a model learned on the data in red and blue would still perform well on the intervention point, which is not the case on the right.

causal of the activation of Y. For example, Figure 9 shows on the x-axis the activity of two genes (gene A on the left, gene B on the right) such that:

- The expressions of A and B are strongly correlated with the expression of Y.
- A is causal of Y (here, we use the definition of a causal effect proposed by Peters et al. (2016)).
- B is non-causal of Y (anticausal or confounded).

In Figure 9 (left), the blue points correspond to the 160 data points from the observational sample, which corresponds to the first task. The red dots are the 1478 data points from the interventional sample, except for the single data point for which A is intervened on, and constitute the second task. The plot on Figure 9 (right) is constructed analogously for B. We can indeed see that in the pooled sample from task 1 and 2, A and B are both strongly correlated with target Y.

The key difference between both plots are the green points. On Figure 9 (left), the green dot corresponds to the single intervention experiment in which gene A is intervened on. Similarly, the green dot on Figure 9 (right) is the single point in which B is intervened on. Our goal is to consider the DG setting in which the test task consists on this single intervention point.

For the causal gene A, one expects that a change in the activity of A should translate into a proportional change in the activity of Y. We observe that, in the particular example of the left plot, a linear regression model from A to Y trained only on the pooled data from tasks 1 and 2 (blue and red in Figure 9) would lead to a small prediction error on the intervened point (in green). That is, $S^* = \{A\}$ might be a good candidate for a set satisfying Assumptions (A1), (A1') and (A2). For the non-causal gene B, however, intervening on B leaves the activity of Y unchanged, and the linear model learned on the data from tasks 1 and 2 performs badly on the test point in green. In such case, a candidate set is the empty set $S^* = \{\}$, leading to prediction using the mean of the target in the training data. A model which is aiming to test in these challenging intervention points should therefore

include causal genes as features, but exclude non-causal genes. In these experiments, we aim at testing whether we can exclude non-causal genes such as B automatically.

Setup We address the problem of predicting the activity of a given gene from the remaining genes. We are looking at the following:

- We consider p different **problems**. In each problem $j \in \{1, \dots, p\}$, we aim at predicting the activity $Y = X_j$ of gene j using $(X_\ell)_{\ell \neq j}$ as features.
- In each problem $j \in \{1, \dots, p\}$, two **training tasks** $k \in \{1, 2\}$ are available. The data from the first task is the observational sample, and the data from the second task are all the n_{int} interventions (we shall subsequently remove some points for testing, see below).

The goal is now to apply our method to each of the problems and estimate an invariant subset. Due to the large number of predictors, we first select the 10 top predictor variables using the Lasso and then apply Algorithm 1 to select a set of invariant predictors $\hat{S}$, see $\beta^{\hat{S}Lasso}$ in Table 2. We denote the indices of the features selected using Lasso by $L = (L_1, \dots, L_{10})$.

The procedure is then evaluated as follows: for each problem $j \in \{1, \dots, p\}$, we first find the genes in $(X_{L_1}, \dots, X_{L_{10}})$ for which an interventional example is available. Note that this might not hold for all selected genes, since only $n_{int} < p$ interventions are available. We then iterate the following procedure (this is within the context of *the same problem*): for each gene in $(X_{L_1}, \dots, X_{L_{10}})$ for which an intervention is available,

- we put aside the example corresponding to this intervention from the training data (in the motivation example, this would correspond to the green point).
- we estimate an invariant subset $\hat{S} \subseteq L$ using Algorithm 1 with the remaining observational and interventional data.
- we test all methods on the single intervention point which was put aside.

We expect two different scenarios, as explained in the motivation paragraph above: (1) if the intervened gene is a *cause* of the target gene, it should still be a good predictor (see Section 2.3); then, it should be beneficial to have this gene included in the set of predictors $\hat{S}$. (2) if the intervened gene is anticausal or confounded (we refer to this scenario as *non-causal*), the statistical relation to the target gene might change dramatically after the intervention and therefore, one may not want to base the prediction on this gene. In order to see this effect and understand how the different approaches for DG in Table 2 handle the problem, we consider two groups of experiments.

(1) we select the target genes Y for which one of the features in L is causal for the activity of Y and for which an intervention experiment is available. 39 problems fall in this causal scenario.

(2) out of the remaining problems we chose target genes with (non-causal) predictors that have been intervened on and — in order to increase the difficulty of the problem — that are strongly correlated with the target gene. We therefore select 269 cases for which a Pearson correlation test (the null hypothesis corresponds to no correlation) outputs a p-value equal to zero.

Results Figure 10 shows box plots for the errors of the different methods for the causal problems (1) on the top left and for the non-causal problems (2) in the top right. We do not plot outliers in order to improve presentation. Figure 10 (top left) presents the causal

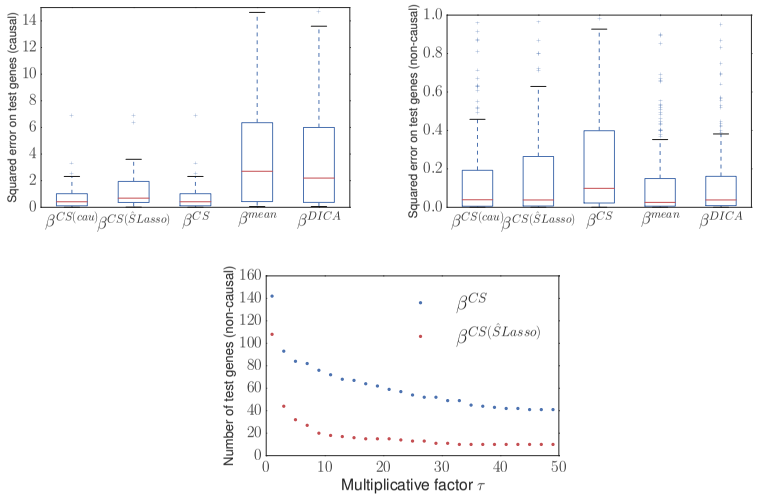


Figure 10: In the causal problems (top left), interventions are performed on causal genes. As expected, the input genes continue to be good predictors, and β^{CS} works well. In the non-causal problems (top right), one of the inputs is intervened upon and becomes a poor predictor, impairing the performance of β^{CS}. The mean predictor β^{mean} uses none of the predictors, and therefore works comparatively well in this scenario. Our proposed estimator $\beta^{CS(\hat{S})}$ provides reasonable estimates in both the causal and non-causal settings, while other methods only perform well in one of the scenarios. β^{DICA} performs similarly to β^{mean} in both scenarios, and is therefore outperformed by other methods in the causal problems (note that β^{DICA} uses all available features). Bottom: in the non-causal scenario (2), we plot the number of test genes for which the squared error for β^{CS} is larger than τ times the squared error for $\beta^{CS(\hat{S})}$, and vice-versa, where τ is plotted on the x-axis. This plot shows the number of genes for which one of the method does significantly worse than the other. By this measure, $\beta^{CS(\hat{S}Lasso)}$ outperforms β^{CS} for all values of τ.

scenario. As expected, pooling does well in this setting. Figure 10 (bottom) shows that in the non-causal problems (2), prediction using an invariant subset leads to less severe mistakes on test genes compared to pooling the tasks.

For comparison, since we know which predictors are being intervened on at test time, we included a method that makes use of causal knowledge: $\beta^{CS(cau)}$ uses all 10 predictors in the causal problems (1) and all but the intervened gene in the non-causal problems (2). In practice, this causal knowledge is often not available. We regard it as promising that the fully automated procedure $\beta^{CS(\hat{S}Lasso)}$ performs comparably to $\beta^{CS(cau)}$.

5. Conclusions and further directions

We propose a method for transfer learning that is motivated by causal modeling and exploits a set of invariant predictors. If the underlying causal structure is known and the tasks correspond to interventions on variables other than the target variable, the causal parents of the target variable constitute such a set of invariant predictors. We prove that predicting using an invariant set is optimal in an adversarial setting in DG. If the invariant structure is not known, we propose an algorithm that automatically detects an invariant subset, while also focusing on good prediction. In practice, we see that our algorithm successfully finds a set of predictors leading to invariant conditionals when enough training tasks are available. Our method can incorporate additional data from the test task (MTL) and yields good performance on synthetic data. Although an invariant set may not always exist, our experiment on real data indicates that exploiting invariance leads to methods which are robust against transfer.

As we saw in the DG and MTL experiments, $\beta^{\hat{S}}$ does not always performs as well as β^{cau}, which uses the ground truth. We believe that alternative methods for estimating the set $\hat{S}$ may close this gap. Furthermore, extending our framework to nonlinearities seems straightforward and may prove to be useful in many applications. For instance, we provide a general, nonlinear version of Theorem 1 in Appendix A. Moreover, Algorithms 1 and 2 are presented in a linear setting. However, the extension to a nonlinear framework is straightforward. In particular, the linear regression can be replaced by a nonlinear regression method. We expect that there may be feature maps leading to invariant conditionals that are different from a subset.

We expect our method to be favorable in (adversarial-like) situations with strong differences between the tasks, such as the gene experiment in Section 4.4. We also evaluated our method on the School dataset (Bakker and Heskes, 2003), but found that we do not do better than pooling the data (we also do not do worse, the results are not shown). We believe this may be due to the fact that the difference between the tasks in this dataset are not too large.

We believe, finally, that the link to causal assumptions and the exploitation of causal structure may lend itself well to proving additional theoretical results on transfer learning.

Appendix A.

In this Appendix, we provide proofs for the theoretical results in the paper, as well as an extension of Theorem 1.

A.1 A nonlinear extension of Theorem 1

The extension of Theorem 1 to a nonlinear setting is straightforward. Given a subset S^* leading to invariant predictions, the proposed predictor is defined as the conditional expectation

$$f_{S^*}: \begin{array}{ccc} \mathbb{R}^p & \to & \mathbb{R} \\ \mathbf{x} & \mapsto & = \mathbb{E}[Y^1 \,|\, \mathbf{X}^1_{S^*} = \mathbf{x}_{S^*}]. \end{array} \tag{11}$$

The following theorem states that f_{S^*} is optimal over the set of continuous functions $\mathcal{C}^0$ in an adversarial setting.

Theorem 4 *Consider D tasks $(\mathbf{X}^1, Y^1) \sim \mathbb{P}^1, \ldots, (\mathbf{X}^D, Y^D) \sim \mathbb{P}^D$ that satisfy Assumption (A1). Then the estimator f_{S^*} in (11) satisfies*

$$f_{S^*} \in \operatorname*{arg\,min}_{f \in \mathcal{C}^0} \sup_{\mathbb{P}^T \in \mathcal{P}} \mathbb{E}_{(\mathbf{X}^T, Y^T) \sim \mathbb{P}^T} \left(Y^T - f(\mathbf{X}^T)\right)^2 ,$$

where $\mathcal{P}$ contains all distributions over $(\mathbf{X}^T, Y^T)$ that are absolutely continuous with respect to the same product measure μ and satisfy $Y^T \,|\, \mathbf{X}^T_{S^*} \overset{d}{=} Y^1 \,|\, \mathbf{X}^1_{S^*}$.

Proof

Consider a function f that is possibly different from f_{S^*}, see (11). For each distribution $\mathbb{Q} \in \mathcal{P}$, we will now construct a distribution $\mathbb{P} \in \mathcal{P}$ such that

$$\int (y - f(\mathbf{x}))^2 \, d\mathbb{P} \geq \int (y - f_{S^*}(\mathbf{x}))^2 \, d\mathbb{Q} \,.$$

In this proof, we assume that the probability distributions in $\mathcal{P}$ are absolutely continuous with respect to Lebesgue measure. The extension to the case where they are absolutely continuous with respect to a same product measure μ is straightforward. Let us therefore assume that $\mathbb{Q}$ has a density $(\mathbf{x}, y) \mapsto q(\mathbf{x}, y)$. Define $\mathbb{P}$ to be the distribution that corresponds to $p(\mathbf{x}, y) := q(\mathbf{x}_{S^*}, y) \cdot q(\mathbf{x}_N)$, where $\mathbf{x}_N$ contains all components of $\mathbf{x}$ that are not in S^*. In the distribution $\mathbb{P}$, the random vector $\mathbf{X}_N$ is independent of $(\mathbf{X}_{S^*}, Y)$. But then

$$\begin{aligned}
&\int (y - f(\mathbf{x}))^2 \, d\mathbb{P} \\
&= \int_{\mathbf{x}_N} \int_{\mathbf{x}_{S^*}, y} (y - f(\mathbf{x}_{S^*}, \mathbf{x}))^2 \, p(\mathbf{x}_{S^*}, y) \, d\mathbf{x}_{S^*} \, dy \, p(\mathbf{x}_N) \, d\mathbf{x}_N \\
&\geq \int_{\mathbf{x}_N} \int_{\mathbf{x}_{S^*}, y} (y - f_{S^*}(\mathbf{x}_{S^*}))^2 \, p(\mathbf{x}_{S^*}, y) \, d\mathbf{x}_{S^*} \, dy \, p(\mathbf{x}_N) \, d\mathbf{x}_N \\
&= \int_{\mathbf{x}, y} (y - f_{S^*}(\mathbf{x}_{S^*}))^2 \, q(\mathbf{x}_{S^*}, \mathbf{x}_N, y) \, d\mathbf{x}_{S^*} \, dy \, d\mathbf{x}_N \\
&= \int (y - f_{S^*}(\mathbf{x}))^2 \, d\mathbb{Q}.
\end{aligned}$$

■

A.2 Proof of Proposition 2

We consider three variables and the following generative process: $Y^k = \alpha^t \mathbf{X}^k_{S^*} + \epsilon^k$, $Z^k = \gamma^k Y^k + \eta^k$, where $\epsilon^k \sim \mathcal{N}(0, \sigma^2)$, $\eta^k \sim \mathcal{N}(0, \sigma^2_\eta)$ and $(\mathbf{X}^k_{S^*})_j \sim \mathcal{N}(0, (\sigma_X)^2_j)$. In this model, γ^k is the parameter responsible for the difference between the tasks, while the other parameters are shared between the tasks.

At training time, D tasks are available. We first aim to obtain an explicit formula for the linear regression coefficients $\beta^{CS} = (\beta^{CS}_{S^*}, \beta^{CS}_Z)$ obtained from pooling all the training tasks together. Denote by $\mathbf{X}$, Y and Z the pooled training data. For fixed $\gamma^1, \dots, \gamma^D$, the expected loss in the training data satisfies for coefficient β verifies:

$$\begin{aligned}\mathbb{E}\left(\left(Y - (\beta_X)^t\mathbf{X} - \beta_Z Z\right)^2\right) &= \frac{1}{D}\sum_{k=1}^{D}\mathbb{E}\left(Y^k - (\beta_X)^t\mathbf{X}^k - \beta_Z Z^k\right)^2 \\ &= \beta_X^t \mathrm{diag}(\sigma_X^2)\beta_X + \frac{\beta_Z^2}{D}\left(\sigma_\eta^2 D + V_Y\overline{\gamma^2}\right) + 2(\beta_Z\frac{\bar\gamma}{D} - 1)\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X + V_Y - 2\frac{\bar\gamma}{D}V_Y\beta_Z, \end{aligned} \tag{12}$$

where $V_Y = \alpha^t\mathrm{diag}(\sigma_X^2)\alpha + \epsilon^2$. By differentiating (12) with respect to β, we obtain the following expression for the pooled coefficients:

$$\beta_Z^{CS} = \frac{\bar\gamma\sigma^2}{V_Y^2\overline{\gamma^2} + D\sigma_\eta^2 - \frac{\bar\gamma^2}{D}\alpha^t\mathrm{diag}(\sigma_X^2)\alpha} \quad \text{and} \quad \beta^{CS}_{S^*} = (1 - \frac{\bar\gamma}{D}\beta_Z^{CS})\alpha,$$

where $\overline{\gamma^2} = \sum_{k=1}^D(\gamma^k)^2$ and $\bar\gamma = \sum_{k=1}^D\gamma^k$. Consider now an unseen test task with coefficient γ^T. The expected loss on the test task using the pooled coefficients is:

$$\begin{aligned}\mathcal{E}_{\mathbb{P}^T}(\beta^{CS}) = \mathbb{E}\left((Y^T - (\beta_X^{CS})^t\mathbf{X}^T - \beta_Z^{CS}Z^T)^2\right) &= \left(\beta_X^{CS}\right)^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} + (\beta_Z^{CS})^2\left(V_Y(\gamma^T)^2 + \sigma_\eta^2\right) \\ &+ 2\beta_Z^{CS}\gamma^T\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} + V_Y \\ &- 2\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} - 2\beta_Z^{CS}V_Y\gamma^T. \end{aligned} \tag{13}$$

Therefore, the expectation with respect to γ^T is:

$$\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})\right) = (\beta_X^{CS})^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} + (\beta_Z^{CS})^2\left(V_Y\Sigma^2 + \sigma_\eta^2\right) + V_Y - 2\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS}$$

Denote by $\mathcal{E}_{\mathbb{P}^T}(\beta^S) = \sigma^2$ the expected loss when using the invariant conditional predictor $\beta^{S^*} = (\alpha, 0)$. Then:

$$\begin{aligned}&\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})\right) \geq \mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{S^*})\right) \\ &\Leftrightarrow (\beta_X^{CS})^t\mathrm{diag}(\sigma_X^2)(\beta_X^{CS}) + (\beta_Z^{CS})^2\left(V_Y\Sigma^2 + \sigma_\eta^2\right) + V_Y - 2\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} \geq \sigma^2 \\ &\Leftrightarrow (\beta_Z^{CS})^2\left(V_Y\Sigma^2 + \sigma_\eta^2\right) \geq 2\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} - (\beta_X^{CS})^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} - \alpha^t\mathrm{diag}(\sigma_X^2)\alpha \\ &\Leftrightarrow (\beta_Z^{CS})^2\left(V_Y\Sigma^2 + \sigma_\eta^2\right) \geq -\frac{\bar\gamma^2}{D^2}(\beta_Z^{CS})^2\alpha^t\mathrm{diag}(\sigma_X^2)\alpha, \end{aligned} \tag{14}$$

by replacing $\beta_X^{CS} = \alpha - \alpha\frac{\bar\gamma}{D}\beta_Z^{CS}$. This inequality holds true for any value of the variance Σ^2, and the pooled coefficient leads to larger error in expectation.

Consider now that the coefficients γ^k are fixed and centered around a non-zero value μ. Then the expectation with respect to γ^T of the loss in the test task is the following:

$$\begin{aligned}\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})\right) = (\beta_X^{CS})^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} + (\beta_Z^{CS})^2\left(V_Y(\Sigma^2 + \mu^2) + \sigma_\eta^2\right) \\ + 2\beta_Z^{CS}\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS}\mu + V_Y - 2\alpha^t\mathrm{diag}(\sigma_X^2)\beta_X^{CS} - 2\beta_Z^{CS}V_Y\mu. \end{aligned} \tag{15}$$

Then, if $\bar\gamma \neq 0$ (if $\bar\gamma = 0$, both estimators coincide):

$$\mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{CS})\right) \geq \mathbb{E}_{\gamma^T}\left(\mathcal{E}_{\mathbb{P}^T}(\beta^{S^*})\right) \Leftrightarrow \Sigma^2 \geq P(\mu), \tag{16}$$

where $P(\mu) = -\mu^2 - \frac{2}{\beta_Z^{CS}}\left(\left(1 - \frac{\bar\gamma}{D}\beta_Z^{CS}\right)\frac{\alpha^t\mathrm{diag}(\sigma_X^2)\alpha}{V_Y} - 1\right)\mu - \frac{\bar\gamma^2}{V_Y D^2}\alpha^t\mathrm{diag}(\sigma_X^2)\alpha + \frac{\sigma_\eta}{V_Y}$.

A.3 Proof of Proposition 3

Proof For $k \in \{1, \dots, D, T\}$, let $\mathbb{Q}^k$ be the probability distribution with density:

$$q^k(\mathbf{x}_{S^*}, \mathbf{x}_N, y) := p^k(\mathbf{x}_{S^*}, y)p^T(\mathbf{x}_N \,|\, \mathbf{x}_{S^*}, y). \tag{17}$$

In the test task T, we trivially have $q^T = p^T$. First, it is easy to see that q^k and p^k have the same marginal distribution over $\mathbf{x}_{S^*}$ and y. Indeed:

$$\begin{aligned}q^k(\mathbf{x}_{S^*}, y) &= \int_{\mathbb{R}^{|N|}} q^k(\mathbf{x}_{S^*}, \mathbf{x}_N, y)d\mathbf{x}_N \\ &= \int_{\mathbb{R}^{|N|}} p^k(\mathbf{x}_{S^*}, y)p^T(\mathbf{x}_N \,|\, \mathbf{x}_{S^*}, y)d\mathbf{x}_N \\ &= p^k(\mathbf{x}_{S^*}, y)\int_{\mathbb{R}^{|N|}} p^T(\mathbf{x}_N \,|\, \mathbf{x}_{S^*}, y)d\mathbf{x}_N = p^k(\mathbf{x}_{S^*}, y). \end{aligned} \tag{18}$$

Second, we prove that the conditional $q^k(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N)$ is the same in all tasks. Indeed, by applying Bayes' rule:

$$\begin{aligned}q^k(y \,|\, \mathbf{x}_{S^*}, \mathbf{x}_N) &= q^k(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})\frac{q^k(y, \mathbf{x}_{S^*})}{q^k(\mathbf{x}_{S^*}, \mathbf{x}_N)} \\ &= p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})\frac{q^k(y \,|\, \mathbf{x}_{S^*})}{q^k(\mathbf{x}_N \,|\, \mathbf{x}_{S^*})} \\ &= p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})\frac{p^k(y \,|\, \mathbf{x}_{S^*})}{\int_{\mathbb{R}} q^k(y, \mathbf{x}_N \,|\, \mathbf{x}_{S^*})dy} \\ &= p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})\frac{p^k(y \,|\, \mathbf{x}_{S^*})}{\int_{\mathbb{R}} q^k(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})q^k(y \,|\, \mathbf{x}_{S^*})dy} \\ &= p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})\frac{p^k(y \,|\, \mathbf{x}_{S^*})}{\int_{\mathbb{R}} p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})p^k(y \,|\, \mathbf{x}_{S^*})dy}. \end{aligned}$$

We have used the fact that $q^k(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*}) = p^T(\mathbf{x}_N \,|\, y, \mathbf{x}_{S^*})$, which follows from (18). Since the last equality leads to a term which is equal in all tasks (indeed, Assumption (A1) ensures that $p^k(y \,|\, \mathbf{x}_{S^*})$ is the same for all $k \in \{1, \dots, D, T\}$), we have the desired result. ■

A.4 Statement and proof of Proposition 5

In this Section, we provide an analytic expression for β^{opt} from (3) in terms of α and ϵ.

Proposition 5 *Assume that $\mathbf{X}_{S^*}$ follows an arbitrary distribution and that Assumptions (A1) and (A2) hold. Let $\gamma \in \mathbb{R}^{|N|}$ be the solution of an L^2 regression from $\mathbf{X}_N^T$ on Y^T. Therefore, we can write $\mathbf{X}_N^T = \gamma Y^T + \eta$, with η uncorrelated to Y^T, and the components of η can be correlated. Then the regression coefficients $\beta^{opt} = (\beta_{S^*}^{opt}, \beta_N^{opt})$ minimizing the expected squared loss in the test task satisfy*

$$\beta_N^{opt} = \mathbb{E}(\epsilon^2) M^{-1} \gamma \,, \tag{19}$$

$$\beta_{S^*}^{opt} = \alpha \left(1 - (\gamma^T)^t \beta_N^T\right) - \Sigma_{X,S^*}^{-1} \Sigma_{X,N} \beta_N \,, \tag{20}$$

where $M = \mathbb{E}(\epsilon^2)\gamma\gamma^t + \Sigma_N - \Sigma_{X,N}^t \Sigma_{X,S^*}^{-1} \Sigma_{X,N}$, and $\Sigma_N := \mathbb{E}(\eta\eta^t)$, $\Sigma_{X,S^*} := \mathbb{E}(\mathbf{X}_{S^*}\mathbf{X}_{S^*}^t)$, $\Sigma_{X,N} := \mathbb{E}(\mathbf{X}_{S^*}\eta^t)$ are the corresponding Gram matrices.[4]

Proof To simplify notation, we write Y^T, $\mathbf{X}_{S^*}^T$ and $\mathbf{X}_N^T$ as Y, $\mathbf{X}_{S^*}$ and $\mathbf{X}_N$. We compute the gradients of the expected squared loss after replacing the expression for Y and $\mathbf{X}_{S^*}$:

$$\begin{aligned} L &= \mathbb{E}(Y - \beta_{S^*}^t \mathbf{X}_{S^*} - \beta_N^t \mathbf{X}_N)^2 \\ &= (\alpha(1 - \gamma^t\beta_N) - \beta_{S^*})^t \Sigma_{X,S^*} (\alpha(1 - \gamma^t\beta_N) - \beta_{S^*}) \\ &\quad + (1 - \beta_N^t\gamma)^2 \mathbb{E}(\epsilon^2) + \beta_N^t \Sigma_N \beta_N - 2(\alpha(1 - \gamma^t\beta_N) - \beta_{S^*})^t \Sigma_{X,N} \beta_N \end{aligned}$$

The gradients satisfy

$$\begin{aligned} \frac{\partial L}{\partial \beta_{S^*}} &= -2\Sigma_{X,S^*}(\alpha(1 - \gamma^t\beta_N) - \beta_{S^*}) + 2\Sigma_{X,N}\beta_N \\ \frac{1}{2}\frac{\partial L}{\partial \beta_N} &= \Sigma_N \beta_N - (1 - \gamma^t\beta_N)\mathbb{E}(\epsilon^2)\gamma + \gamma\alpha^t \Sigma_{X,N}\beta_N \\ &\quad - \gamma\alpha^t \Sigma_{X,S^*}(\alpha(1 - \gamma^t\beta_N) - \beta_{S^*}) - \Sigma_{X,N}^t(\alpha(1 - \gamma^t\beta_N) - \beta_{S^*}) \end{aligned}$$

By setting these to zero, we find the stated values for $\beta_{S^*}^{opt}$ and β_N^{opt}. ■

Appendix B.

The code to reproduce the experiments in the paper can be found in `https://github.com/mrojascarulla/causal_transfer_learning`.

4. We dropped the superscript T to lighten the notation.

References

A. Argyriou, T. Evgeniou, and M. Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 41 – 48, 2007a.

A. Argyriou, M. Pontil, Y. Ying, and C. Micchelli. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 25 – 32, 2007b.

B. Bakker and T. Heskes. Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, 4:83 – 99, 2003.

J. Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12:149 – 198, 2000.

S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility theorems for domain adaptation. In *Proc. of the 13th Intern. Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 129 – 136, 2010.

E. Bonilla, K. Chai, and C. Williams. Multi-task Gaussian process prediction. In *Advances in Neural Information Processing Systems 20 (NIPS)*, pages 153 – 160, 2007.

P. Bühlmann and S. van de Geer. *Statistics for High-Dimensional Data: Methods, Theory and Applications*. Springer Series in Statistics. Springer, New York, NY, 2011.

R. Caruana. Multitask learning. *Machine Learning*, 28:41 – 75, 1997.

M. Chen, Z. Xu, K. Weinberger, and F. Sha. Marginalized denoising autoencoders for domain adaptation. In *Proc. of the 29th Intern. Conference on Machine Learning (ICML)*, pages 767 – 774, 2012.

H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proc. of the 2010 Workshop on Domain Adaptation for NLP*, pages 53 – 59, 2010.

T. Evgeniou and M. Pontil. Regularized multi-task learning. In *Proc. of the 10th Intern. Conference on Knowledge Discovery and Data Mining*, pages 109 – 117, 2004.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 489–496, 2008.

J. Gao, W. Fan, J. Jiang, and J. Han. Knowledge transfer via multiple model local structure mapping. In *Proc. of the 10th Intern. Conference on Knowledge Discovery and Data Mining*, pages 283 – 291, 2008.

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 19 (NIPS)*, pages 585 – 592, 2007.

A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3:131 – 160, 2009.

C. Heinze-Deml, J. Peters, and N. Meinshausen. Invariant causal prediction for nonlinear models. *Journal of Causal Inference*, 2018.

K. D. Hoover. The logic of causal inference. *Economics and Philosophy*, 6:207 – 234, 1990.

D. Janzing and B. Schölkopf. Causal inference using the algorithmic Markov condition. *IEEE Transactions on Information Theory*, 56(10):5168 – 5194, 2010.

P. Kemmeren, K. Sameith, L. van de Pasch, J. Benschop, T. Lenstra, T. Margaritis, E. ODuibhir, E. Apweiler, S. van Wageningen, C. Ko, et al. Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell*, 157(3):740 – 752, 2014.

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25 (NIPS)*, pages 1097 – 1105, 2012.

N. Lawrence and J. Platt. Learning to learn with the informative vector machine. In *Proc. of the 21st Intern. Conference on Machine learning (ICML)*, pages 65–72, 2004.

H. Levene. Robust tests for equality of variances. *Contributions to probability and statistics: Essays in honor of Harold Hotelling*, 2:278 – 292, 1960.

R. Little and D. Rubin. *Statistical Analysis with Missing Data*. John Wiley & Sons, Inc., 1986. ISBN 0-471-80254-9.

K. Muandet, D. Balduzzi, and B. Schölkopf. Domain generalization via invariant feature representation. In *Proc. of the 30th Intern. Conference on Machine Learning (ICML)*, pages 10 – 18, 2013.

S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345 – 1359, 2010.

J. Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge Univ. Press, USA, 2nd edition, 2009.

J. Peters, P. Bühlmann, and N. Meinshausen. Causal inference using invariant prediction: identification and confidence intervals. *Journal of the Royal Statistical Society: Series B (with discussion)*, 78(5):947 – 1012, 2016.

J. Peters, D. Janzing, and B. Schölkopf. *Elements of Causal Inference - Foundations and Learning Algorithms*. MIT Press, Cambridge, MA, USA, 2017.

J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence. *Dataset shift in machine learning*. The MIT Press, 2009.

R. Raina, A. Battle, H. Lee, B. Packer, and A. Y. Ng. Self-taught learning: Transfer learning from unlabeled data. In *Proc. of the 24th Intern. Conference on Machine Learning (ICML)*, pages 759 – 766, 2007.

B. Romera-Paredes, A. Argyriou, N. Berthouze, and M. Pontil. Exploiting unrelated tasks in multi-task learning. In *Proc. of the 15th Intern. Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 951 – 959, 2012.

J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117, 2015.

B. Schölkopf, D. Janzing, J. Peters, E. Sgouritsa, K. Zhang, and J. M. Mooij. On causal and anticausal learning. In *Proc. of the 29th Intern. Conference on Machine Learning (ICML)*, pages 1255 – 1262, 2012.

G. Schweikert, C. Widmer, B. Schölkopf, and G. Rätsch. An empirical analysis of domain adaptation algorithms for genomic sequence analysis. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1433 – 1440, 2009.

H. Shimodaira. Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference*, 90(2):227 – 244, 2000.

M. Sugiyama, S. Nakajima, H. Kashima, P. Buenau, and M. Kawanabe. Direct importance estimation with model selection and its application to covariate shift adaptation. In *Advances in Neural Information Processing Systems 21 (NIPS)*, pages 1433 – 1440, 2008.

R. Tibshirani. Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B*, 58:267 – 288, 1996.

K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proc. of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 804 – 813, 2011.

K. Zhang, B. Schölkopf, K. Muandet, and Z. Wang. Domain adaptation under target and conditional shift. In *Proc. of the 30th Intern. Conference on Machine Learning (ICML)*, pages 819 – 827, 2013.

K. Zhang, M. Gong, and B. Schölkopf. Multi-source domain adaptation: A causal view. In *Proc. of the 29th AAAI Conference on Artificial Intelligence*, pages 3150 – 3157, 2015a.

K. Zhang, J. Zhang, and B. Schölkopf. Distinguishing cause from effect based on exogeneity. In *Proc. of the 15th conference on Theoretical Aspects of Rationality and Knowledge (TARK)*, pages 261 – 271, 2015b.

P. Zwiernik, C. Uhler, and D. Richards. Maximum likelihood estimation for linear gaussian covariance models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(4):1269 – 1292, 2017.

Journal of Machine Learning Research 19 (2018) 1-42 Submitted 10/16; Revised 12/17; Published 8/18

The *xyz* algorithm for fast interaction search in high-dimensional data

Gian-Andrea Thanei G.A.THANEI@GMAIL.COM
Seminar für Statistik
ETH Zürich
8092 Zürich, Switzerland

Nicolai Meinshausen MEINSHAUSEN@STAT.MATH.ETHZ.CH
Seminar für Statistik
ETH Zürich
8092 Zürich, Switzerland

Rajen D. Shah* RDS37@CAM.AC.UK
Statistical Laboratory
University of Cambridge
Cambridge, CB3 0WB, UK

Editor: Jennifer Dy

Abstract

When performing regression on a data set with p variables, it is often of interest to go beyond using main linear effects and include interactions as products between individual variables. For small-scale problems, these interactions can be computed explicitly but this leads to a computational complexity of at least $\mathcal{O}(p^2)$ if done naively. This cost can be prohibitive if p is very large.

We introduce a new randomised algorithm that is able to discover interactions with high probability and under mild conditions has a runtime that is subquadratic in p. We show that strong interactions can be discovered in almost linear time, whilst finding weaker interactions requires $\mathcal{O}(p^\alpha)$ operations for $1 < \alpha < 2$ depending on their strength. The underlying idea is to transform interaction search into a closest pair problem which can be solved efficiently in subquadratic time. The algorithm is called *xyz* and is implemented in the language `R`. We demonstrate its efficiency for application to genome-wide association studies, where more than 10^{11} interactions can be screened in under 280 seconds with a single-core 1.2 GHz CPU.

Keywords: interactions, high-dimensional data, regression, computational tradeoffs, close pairs

1. Introduction

Given a response vector $\mathbf{Y} \in \mathbb{R}^n$ and matrix of associated predictors $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_p) \in \mathbb{R}^{n \times p}$, finding interactions is often of great interest as they may reveal important relationships and improve predictive power. When the number of variables p is large, fitting a model involving interactions can involve serious computational challenges. The simplest form of interaction search consists of screening for pairs (j, k) with high inner product between the outcome of interest $\mathbf{Y}$ and the point-wise product $\mathbf{X}_j \circ \mathbf{X}_k$:

$$\text{Keep all pairs } (j,k) \text{ for which } \mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)/n > \kappa. \tag{1}$$

This search is of complexity $\mathcal{O}(np^2)$ in a naive implementation and quickly becomes infeasible for large p. Of course one would typically be interested in maximising (absolute values of) correlations rather than dot products in (1), an optimisation problem that would be at least as computationally intensive.

Even more challenging is the task of fitting a linear regression model involving pairwise interactions:

$$Y_i = \mu + \sum_{j=1}^{p} X_{ij}\beta_j + \sum_{k=1}^{p}\sum_{j=1}^{k-1} X_{ij}X_{ik}\theta_{jk} + \varepsilon_i. \tag{2}$$

Here $\mu \in \mathbb{R}$ is the intercept and β_j and θ_{jk} contain coefficients for main effects and interactions respectively, and ε_i is random noise.

In this paper, we make several contributions to the problem of searching for interactions in high-dimensional settings.

(a) We first establish a form of equivalence between (1) and closest-pair problems (Shamos and Hoey, 1975; Agarwal et al., 1991). Assume for now that all predictors and outcomes are binary, so $X_{ij}, Y_i \in \{-1, 1\}$ (we will later relax this assumption) and define $\mathbf{Z} \in \{-1,1\}^{n \times p}$ as $Z_{ij} = Y_i X_{ij}$. Then it is straightforward to show that (1) is equivalent to

$$\text{Keep all pairs } (j,k) \text{ for which } \|\mathbf{X}_j - \mathbf{Z}_k\|_2 < \kappa' \tag{3}$$

for some κ'. This connects the search for interactions to literature in computational geometry on problems of finding closest pairs of points.

(b) We introduce the *xyz* algorithm to solve (3) based on randomly projecting each of the columns in $\mathbf{X}$ and $\mathbf{Z}$ to a one-dimensional space. By exploiting the ability to sort the resulting $2p$ points with $\mathcal{O}(p\log(p))$ computational cost, we achieve a run time that is always subquadratic in p and can even reach a linear complexity $\mathcal{O}(np)$ when κ is much larger than the quantity $|\mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)|/n$ of the bulk of the pairs (j, k). We show that our approach can be viewed as an example of locality sensitive hashing (Leskovec et al., 2014) optimised for our specific problem.

(c) We show how any method for solving (1) can be used to fit regression models with interactions (15) by building it into an algorithm for the Lasso (Tibshirani, 1996). The use of *xyz* thus leads to a procedure for applying the Lasso to all main effects and interactions with computational cost that scales subquadratically in p.

(d) We provide implementations of both the core *xyz* algorithm and its extension to the Lasso in the R package `xyz`, which is available on github (Thanei, 2016) and CRAN.

Our work here is thus related to "closest pairs of points" algorithms in computational geometry as well as an extensive literature on modelling interactions in statistics, both of which we now review.

∗. Supported by the Isaac Newton Trust Early Career Support Scheme, the Alan Turing Institute under the EPSRC grant EP/N510129/1 and EPSRC Programme Grant EP/N031938/1

1.1 Related work

A common approach to avoid the quadratic cost in p of searching over all pairs of variables (1) is to restrict the search space: one can first seek a small number of important main effects, and then only consider interactions involving these discovered main effects. More specifically, one could fit a main effects Lasso (Tibshirani, 1996) to the data first, add interactions between selected main effects to the matrix of predictors, and then run the Lasso once more on the augmented design matrix in order to produce the final model (see Wu et al. (2010) for example). Tree-based methods such as CART (Breiman et al., 1984) work in a similar fashion by aiming to identify an important main effect and then only considering interactions involving this discovered effect.

However it is quite possible for the signal to be such that main effects corresponding to important interactions are hard to detect. As a concrete example of this phenomenon, consider the setting where $\mathbf{X}$ is generated randomly with all entries independent and having the uniform distribution on $\{-1, 1\}$. Suppose the response is given by $Y_i = X_{i1}X_{i2}$, so there is no noise. Since the distribution $Y_i|X_{ij}$ is the same for all k, main effects regressions would find it challenging to select variables 1 and 2. Note that by reparametrising the model by adding one to each entry of $\mathbf{X}$ for example, we obtain $Y_i = (X_{i1} - 1)(X_{i2} - 1) = 1 - X_{i1} - X_{i2} + X_{i1}X_{i2}$. The model now respects the so-called strong hierarchical principle (Bien et al., 2013) that interactions are only present when their main effects are. The hierarchical principle is useful to impose on any fitted model. However, imposing the principle on the model does not imply that the interactions will easily be found by searching for main effects first. The difficulty of the example problem is due to interaction effects masking main effects: this is a property of the signal $\mathbb{E}(Y_i)$ and of course no reparametrisation can make the main effects any easier to find. Approaches that increase the set of interactions to be considered iteratively can help to tackle this sort of issue in practice (Bickel et al., 2010; Hao and Zhang, 2014; Friedman, 1991; Shah, 2016) as can those that randomise the search procedure (Breiman, 2001). However they cannot eliminate the problem of missing interactions, nor do these approaches offer guarantees of how likely it is that they discover an interaction.

As alluded to earlier, the pure interaction search problem (3) is related to close pairs of points problems, and more specifically the close bichromatic pairs problem in computational geometry (Agarwal et al., 1991). Most research in this area has focused on algorithms that lead to computationally optimal results in the number of points p whilst considering the dimension n to be constant. This has resulted in algorithms where the scaling of the computational complexity with n is at least of order 2^n (Shamos and Hoey, 1975). Since for meaningful statistical results one would typically require $n \gg \log(p)$, these approaches would not lead to subquadratic complexity. An exception is the so-called lightbulb algorithm (Paturi et al., 1989) which employs a similar strategy for binary data; our work here shows that this is optimal among random projection-based methods and also that it may be modified to handle continuous data and also detect interactions in high-dimensional regression settings.

In the special case where $n = p$ and $Z_{ij}, X_{ij} \in \{-1, 1\}$, (3) may be seen to be equivalent to searching for large magnitude entries in the product of square matrices $\mathbf{X}$ and $\mathbf{Z}^T$. This latter problem is amenable to fast matrix multiplication algorithms, which in theory can

deliver a subquadratic complexity of roughly $\mathcal{O}(p^{2.4}) = \mathcal{O}(np^{1.4})$ (Williams, 2012; Davie and Stothers, 2013; Le Gall, 2012). However the constants hidden in the order notation are typically very large, and practical implementations are unavailable. The Strassen algorithm (Strassen, 1969) is the only fast matrix multiplication algorithm used regularly in practice to the best of our knowledge. With a complexity of roughly $\mathcal{O}(p^{2.8}) = \mathcal{O}(np^{1.8})$, the improvement over a brute force close pairs search is only slight.

The strategy we use is most closely related to locality sensitive hashing (LSH) (Indyk and Motwani, 1998) which encompasses a family of hashing procedures such that similar points are mapped to the same bucket with high probability. A close pair search can then be conducted by searching among pairs mapped to the same bucket. In fact, our approach for solving (3) can be thought of as an example of LSH optimised for our particular problem setting. This connection is detailed in Appendix B.

A seemingly attractive alternative to the subsampling-based LSH-strategy we employ is the method of random projections which is motivated by the theoretical guarantees offered by the Johnson–Lindenstrauss Lemma (Achlioptas, 2003). Perhaps surprisingly, we can show that using random projections instead of our subsampling-based scheme leads to a quadratic run time for interaction search (see Theorem 1 and section 5.1).

An approach that bears some similarity with our procedure is that of *epiq* (Arkin et al., 2014). This works by projecting the data and then searches through a lower dimensional representation for close pairs. This appears to improve upon a naive brute force empirically but there are no proven guarantees that the run time improves on the $\mathcal{O}(np^2)$ complexity of a naive search.

The *Random Intersection Trees* algorithm of Shah and Meinshausen (2014) searches for potentially deeper interactions in data with both $\mathbf{X}$ and $\mathbf{Y}$ binary. In certain cases with strong interactions a complexity close to linear in p is achieved; however it is not clear how to generalise the approach to continuous data or embed it within a regression procedure.

The idea of Kong et al. (2016) is to first transform the data by forming $\tilde{\mathbf{Y}} = \mathbf{Y} \circ \mathbf{Y}$ and $\tilde{\mathbf{X}}_j = \mathbf{X}_j \circ \mathbf{X}_j$ for each predictor. Next $\tilde{\mathbf{X}}_j$ and $\tilde{\mathbf{Y}}$ are tested for independence using the distance correlation test. In certain settings, this can reveal important interactions with a computational cost linear in p. However, the powers of these tests depend on the distributions of the transformed variables $\tilde{\mathbf{X}}_j$. For example in the binary case when $\mathbf{X} \in \{-1, 1\}^{n \times p}$, each transformed variable will be a vector of 1's and the independence tests will be unhelpful. We will see that our proposed approach works particularly well in this setting.

1.2 Organisation of the paper

In Section 2 we consider the case where both the response $\mathbf{Y}$ and the predictors $\mathbf{X}$ are binary. We first demonstrate how (15) may be converted to a form of closest pair of points problem. We then introduce a general version of the *xyz* algorithm which solves this based on random projections. As we show in Section 2.1 there is a particular random projection distribution that is optimal for our purposes. This leads to our final version of the *xyz* algorithm which we present in Section 2.3 along with an analysis of its run time and probabilistic guarantees that it recovers important interactions. In Section 3 we extend the *xyz* algorithm to continuous data. These ideas are then used in Section 4 to

demonstrate how the *xyz* algorithm can be embedded within common algorithms for high-dimensional regression (Friedman et al., 2010) allowing high-dimensional regression models with interactions to be fitted with subquadratic complexity in p. Section 5 contains a variety of numerical experiments on real and simulated data that complement our theoretical results and demonstrate the effectiveness of our proposal in practice. We conclude with a brief discussion in Section 6 and all proofs are collected in the Appendix.

2. The *xyz* algorithm for binary data

In this section, we present a version of the *xyz* algorithm applicable in the special case where both $\mathbf{X}$ and $\mathbf{Y}$ are binary, so $X_{ij} \in \{-1, 1\}$ and $Y_i \in \{-1, 1\}$. We build up to the algorithm in stages, giving the final version in Section 2.2.

Define $\mathbf{Z} \in \{-1, 1\}^{n \times p}$ by $Z_{ij} = Y_i X_{ij}$ and

$$\gamma_{jk} = \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}_{\{Y_i = X_{ij} X_{ik}\}}. \tag{4}$$

We call γ_{jk} the interaction strength of the pair (j, k). It is easy to see that the interaction search problem (1) can be expressed in terms of either the γ_{jk} or the normalised squared distances. Indeed

$$2\gamma_{jk} - 1 = \mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)/n = \mathbf{Z}_j^T \mathbf{X}_k / n = 1 - \|\mathbf{Z}_j - \mathbf{X}_k\|_2^2/(2n). \tag{5}$$

Thus those pairs (j, k) with $\mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)/n$ large will have γ_{jk} large, and $\|\mathbf{Z}_j - \mathbf{X}_k\|_2^2$ small. This equivalence suggests that to solve (1), we can search for pairs (j, k) of columns $\mathbf{Z}_j, \mathbf{X}_k$ that are close in ℓ_2 distance. At first sight, this new problem would also appear to involve a search across all pairs, and would thus incur an $\mathcal{O}(np^2)$ cost. As mentioned in the introduction, close pair searches that avoid a quadratic cost in p incur typically an exponential cost in n. Since n would typically be much larger than $\log(p)$, such searches would be computationally infeasible.

We can however project each of the n-dimensional columns of $\mathbf{X}$ and $\mathbf{Z}$ to a lower dimensional space and then perform a close pairs search. The Johnson–Lindenstrauss Lemma, which states roughly that one can project p points into a space of dimension $\mathcal{O}(\log(p))$ and faithfully preserve distances, may appear particularly relevant here. The issue is that the projected dimension suggested by the Johnson–Lindenstrauss Lemma is still too large to allow for an efficient close pairs search. The following observation however gives some encouragement: if we had $\mathbf{Y} = \mathbf{X}_j \circ \mathbf{X}_k$ so $\mathbf{X}_j = \mathbf{Z}_k$, even a one-dimensional projection $\mathbf{R} \in \mathbb{R}^n$ will have $|\mathbf{R}^T(\mathbf{X}_j - \mathbf{Z}_k)| = 0 = \|\mathbf{X}_j - \mathbf{Z}_k\|_2$, which implies that a perfect interaction will have zero distance in the projected space. We will later see that our approach leads to a linear run time in such a case. Importantly, we are only interested in using a projection that preserves the distances between the close pairs rather than all pairs, which makes our problem very different to the setting considered in the Johnson–Lindenstrauss Lemma.

With this in mind, consider the following general strategy. First project the columns of $\mathbf{X}$ and $\mathbf{Z}$ to one-dimensional vectors x and z using a random projection $\mathbf{R}$: $\mathbf{x} = \mathbf{X}^T\mathbf{R}$, $\mathbf{z} = \mathbf{Z}^T\mathbf{R}$. Next for some threshold τ, collect all pairs (j, k) such that $|x_j - z_k| \leq \tau$ in the set E. By first sorting $\mathbf{x}$ and $\mathbf{z}$, a step requiring only $\mathcal{O}(p \log(p))$ computations (see

for example Sedgewick (1998)), this close pairs search can be shown to be very efficient. Given this set of candidate interactions, we can check for each $(j, k) \in E$ whether we have $\gamma_{jk} \geq \gamma$. The process can be repeated L times with different random projections, and one would hope that given enough repetitions, any given strong interaction would be present in one of the candidate sets $E_1, \ldots, E_L$ with high probability. This approach is summarised in Algorithm 1 which we term the general form of the *xyz* algorithm. A schematic overview is given in Figure 1.

Algorithm 1 A general form of the *xyz* algorithm.

Input: $\mathbf{X} \in \{-1, 1\}^{n \times p}$, $\mathbf{Y} \in \{-1, 1\}^n$
Parameters: $\xi = (G, L, \tau, \gamma)$. Here G is the joint distribution for the projection vector $\mathbf{R}$, L is the number of projections, and τ and γ are the thresholds for close pairs and interactions strength respectively.
Output: I set of strong interactions.

Form $\mathbf{Z}$ via $Z_{ij} = Y_i X_{ij}$ and set $I := \emptyset$.
for $l \in \{1, \ldots, L\}$ **do**
  Draw random vector $\mathbf{R} \in \mathbb{R}^n$ with distribution G and project the data using $\mathbf{R}$, to form

$$\mathbf{x} = \mathbf{X}^T\mathbf{R} \text{ and } \mathbf{z} = \mathbf{Z}^T\mathbf{R}.$$

  Collect in E_l all pairs (j, k) such that $|x_j - z_k| \leq \tau$.
  Add to I those $(j, k) \in E_l$ for which $\gamma_{jk} \geq \gamma$.
end for

There are several parameters that must be selected, and a key choice to be made is the form of the random projection $\mathbf{R}$. For the joint distribution G of $\mathbf{R}$ we consider the following general class of distributions, which includes both dense and sparse projections. We sample a random or deterministic number M of indices from the set $\{1, \ldots, n\}$, $i_1, \ldots, i_M$, either with or without replacement. Then, given a distribution $F \in \mathcal{F}$ where $\mathcal{F}$ is a class of distributions to be specified later, we form a vector $\mathbf{D} \in \mathbb{R}^M$ with independent components each distributed according to F. We then define the random projection vector $\mathbf{R}$ by

$$R_i = \sum_{m=1}^{M} D_m \mathbb{1}_{\{i_m = i\}}, \qquad i = 1, \ldots, n. \tag{6}$$

Each configuration of the *xyz* algorithm is characterised by fixing the following parameters:

(i) G, a distribution for the projection vector R which is determined through (6) by $F \in \mathcal{F}$, a distribution for the subsample size M and whether sampling is with replacement or not;

(ii) $L \in \mathbb{N}$, the number of projection steps;

(iii) $\tau \geq 0$, the close pairs threshold;

(iv) $\gamma \in (0, 1)$, the interaction strength threshold.

We will denote the collection of all possible parameter levels by Ξ. This includes the following subclasses of interest. Fix $F \in \mathcal{F}$.

(a) **Dense projections**. Let $\mathbf{R} \in \mathbb{R}^n$ have independent components distributed according to F and denote the distribution of $\mathbf{R}$ by G. This falls within our general framework above with M set to n and sampling without replacement. Let

$$\Xi_{\text{dense}} := \{\xi \in \Xi \text{ with joint distribution equal to G}\}.$$

(b) **Subsampling**. Let $\mathcal{G}_{\text{subsample}}$ be the set of distributions for R obtained through (6) when subsampling with replacement. Let

$$\Xi_{\text{subsample}} := \{\xi \in \Xi : \text{joint distribution } G \in \mathcal{G}_{\text{subsample}}\}.$$

(c) **Minimal subsampling**. Let Ξ_{minimal} be the set of all parameters in $\Xi_{\text{subsample}}$ such that the close pairs threshold is $\tau = 0$ and M takes randomly values in the set $\{m, m+1\}$ for some positive integer m.

$$\Xi_{\text{minimal}} := \{\xi \in \Xi_{\text{subsample}} \text{ with } \tau = 0 \text{ and } M \in \{m, m+1\} \text{ for some } m \in \mathbb{N}\}.$$

Note that we have suppressed the dependence of the classes above on the fixed distribution $F \in \mathcal{F}$ for notational simplicity. We define $\mathcal{F}$ to be the set of all univariate absolutely continuous and symmetric distributions with bounded density and finite third moment. The restriction to continuous distributions in $\mathcal{F}$ ensures that Ξ_{minimal} is invariant to the choice of F: when $\tau \equiv 0$, every $F \in \mathcal{F}$ with $L \in \mathbb{N}$ and the distribution for M fixed yields the same algorithm. Moreover the set of close pairs in C_l is simply the set of pairs (j,k) that have $X_{i_m j} = Z_{i_m k}$ for all $m = 1, \ldots, M$, that is the set of pairs that are equal on the subsampled rows. We note that the symmetry and boundedness of the densities in $\mathcal{F}$ and finiteness of the third moment are mainly technical conditions necessary for the theoretical developments in the following section. We will assume without loss of generality that the second moment is equal to 1. This condition places no additional restriction on Ξ since a different second moment may be absorbed into the choice of τ.

Minimal subsampling represents a very small subset of the much larger class of randomised algorithms outlined above. However, Theorem 1 below shows that minimal subsampling is essentially always at least as good as any algorithm from the wider class, which is perhaps surprising. A beneficial consequence of this result is that we only need to search for the optimal ways of selecting M and L; the threshold τ is fixed at $\tau = 0$ and the choice of the continuous distribution F is inconsequential for minimal subsampling. The choices we give in Section 2.2 yield a subquadratic run time that approaches linear in p when the interactions to be discovered are much stronger than the bulk of the remaining interactions.

2.1 Optimality of minimal subsampling

In this section, we compare the run time of the algorithms in $\xi \in \Xi_{\text{dense}}, \Xi_{\text{subsample}}$ and Ξ_{minimal} that return strong interactions with high probability. Let (j^*, k^*) be the indices of

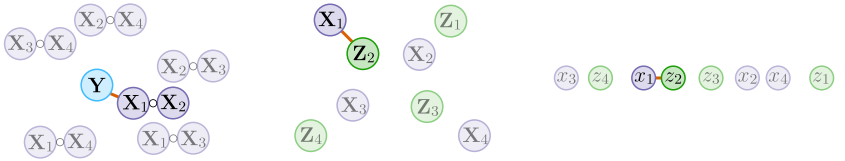


Figure 1: Illustration of the general *xyz* algorithm. The strongest interaction is the pair $(1,2)$ and $p = 4$. Panel a) illustrates the interaction search among $\mathbf{Y}$ and $\mathbf{X}_j \circ \mathbf{X}_k$, panel b) shows the closest pair problem after the transformation $Z_{ij} = X_{ij} Y_i$ and panel c) depicts the closest pair problem after the data has been projected. These are the three main steps in the *xyz* algorithm.

a strongest interaction pair, that is $\gamma_{j^*k^*} = \max_{j,k \in \{1,\ldots,p\}} \gamma_{jk}$. We will consider algorithms ξ with γ set to $\gamma_{j^*k^*}$. Define the power of $\xi \in \Xi$ as

$$\text{Power}(\xi) := \mathbb{P}_\xi((j^*, k^*) \in I).$$

For $\eta \in (0,1)$, let

$$\Xi_{\text{dense}}(\eta) = \{\xi \in \Xi_{\text{dense}} : \text{Power}(\xi) \geq \eta\},$$

and define $\Xi_{\text{subsample}}(\eta)$ and $\Xi_{\text{minimal}}(\eta)$ analogously. Note that these classes depend on the underlying $F \in \mathcal{F}$, which is considered to be fixed, and moreover that we are fixing $\gamma = \gamma_{j^*k^*}$. We consider an asymptotic regime where we have a sequence of response–predictor matrix pairs $(\mathbf{Y}^{(n)}, \mathbf{X}^{(n)}) \in \mathbb{R}^n \times \mathbb{R}^{n \times p_n}$. Write $\gamma_{jk}^{(n)}$ for the corresponding interaction strengths, and let $\gamma_1^{(n)} = \max_{j,k} \gamma_{jk}^{(n)}$. Let $f_{\boldsymbol{\gamma}^{(n)}}$ be the probability mass function corresponding to drawing an element of $\boldsymbol{\gamma}^{(n)}$ uniformly at random. Note that $f_{\boldsymbol{\gamma}^{(n)}}$ has domain $\{0, 1/n, 2/n, \ldots, 1\}$. We make the following assumptions about the sequence of interaction strength matrices $\boldsymbol{\gamma}^{(n)}$.

(A1) There exists c_0 such that $|\{(j,k) : \gamma_{jk}^{(n)} = \gamma_1^{(n)}\}| \leq c_0 p_n$.

(A2) There exists $\gamma_l > 0$, $\gamma_u < 1$ such that $\gamma_u \geq \gamma_1^{(n)} \geq \gamma_l$ for all n.

(A3) There exists $\rho < 1$ such that $f_{\boldsymbol{\gamma}^{(n)}}$ is non-increasing on $[\rho\gamma_1^{(n)}, \gamma_1^{(n)}) \cap \{0, 1/n, \ldots, 1\}$.

Assumption (A1) is rather weak: typically one would expect the maximal strength interaction to be essentially unique, while (A1) requires that at most of order p_n interactions have maximal strength. (A2) requires the maximal interaction strength to be bounded away from 0 and 1, which is the region where complexity results for the search of interactions are of interest. As mentioned earlier, if the maximal interaction strength is 1, it will always be retained in the close-pair sets C_l, whilst if its strength is too close to 0, then it is near impossible to distinguish it from the remaining interactions. (A3) ensures a certain form of separation between maximal strength interactions and the bulk of the interactions.

To aid readability, in the following we suppress the dependence of quantities on n in the notation. Given $\mathbf{X}$ and $\mathbf{Y}$, we may define $T(\xi)$ as the expected number of computational operations performed by the algorithm corresponding to ξ. We have the following result.

Theorem 1 *Given $F \in \mathcal{F}$ and $\eta \in (0,1)$, there exists n_0 such that for all $n \geq n_0$ we have*

$$\inf_{\xi \in \Xi_{minimal}(\eta)} T(\xi) = \inf_{\xi \in \Xi_{subsample}(\eta)} T(\xi), \tag{7}$$

$$\inf_{\xi \in \Xi_{minimal}(\eta)} \frac{T(\xi)}{np^2} \to 0, \tag{8}$$

and there exists $c > 0$ such that

$$\inf_{\xi \in \Xi_{dense}(\eta)} \frac{T(\xi)}{np^2} > c. \tag{9}$$

The theorem shows that the optimal run time is achieved when using minimal subsampling. The last point is surprising: setting $\mathbf{R} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$, for example, will not improve the computational complexity over the brute-force approach and dense Gaussian projections hence do not reduce the complexity of the search. This is not caused by the larger computational effort involved in computing the dense projections: indeed even if these could be computed for free this result would remain. Rather the cost stems from the fact that dense projections have a much lower power for detecting true close pairs in the projected one-dimensional space.

2.2 The final version of xyz

The optimality properties of minimal subsampling presented in the previous section suggest the approach set out in Algorithm 2, which we will refer to as the *xyz* algorithm. Here we

Algorithm 2 Final version of the *xyz* algorithm.

Input: $\mathbf{X} \in \{-1,1\}^{n \times p}$, $\mathbf{Y} \in \{-1,1\}^n$, subsample size M, number of projections L, threshold for interaction strength γ.
Output: I set of strong interactions.
1: Form $\mathbf{Z}$ via $Z_{ij} = Y_i X_{ij}$.
2: **for** $l \in \{1, \ldots, L\}$ **do**
3: Form $\mathbf{R} \in \mathbb{R}^n$ as in (6) with distribution $F = U[0,1]$ and set $\mathbf{x} = \mathbf{X}^T\mathbf{R}$, $\mathbf{z} = \mathbf{Z}^T\mathbf{R}$.
4: Find all pairs (j,k) such that $x_j = z_k$ and store these in E_l.
5: Add to I those pairs in E_l for which $\gamma_{jk} \geq \gamma$.
6: **end for**

are using a simplified version of the minimal subsampling proposal given in the previous section where we keep M fixed rather than allowing it to be random. The reason is that the potential additional gain from allowing M to be any one of two consecutive numbers with certain probabilities is minimal but necessary for Theorem 1 and so the simpler approach is preferable. We note that the uniform distribution in line 3 may be replaced with any continuous distribution to yield identical results.

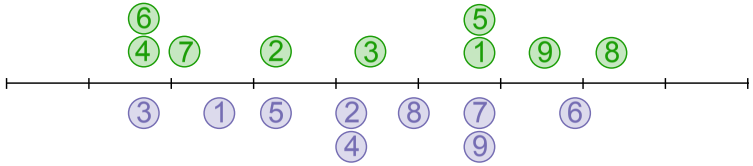


Figure 2: Illustration of an equal pairs search among components of $\mathbf{x}, \mathbf{z} \in \mathbb{R}^p$ when $p = 9$. The horizontal locations of blue and green circles numbered j give x_j and z_j respectively. Sorting of $(\mathbf{x}, \mathbf{z})$ allows traversal of the unique locations. At each of these it is checked whether points of both colours are present, and if so, the indices are recorded. Here the set of equal pairs $(\{3\} \times \{4,6\}) \cup (\{5\} \times \{2\}) \cup (\{7,9\} \times \{1,5\})$ would be returned.

To perform the equal pairs search in line 4, we sort the concatenation $(\mathbf{x}, \mathbf{z}) \in \mathbb{R}^{2p}$ to determine the unique elements of $\{x_1, \ldots, x_p, z_1, \ldots, z_p\}$. At each of these locations, we can check if there are components from both x and z lying there, and if so record their indices. This procedure, which is illustrated in Figure 2, gives us the set of equal pairs E in the form of a union of Cartesian products. The computational cost is $\mathcal{O}(p \log(p))$. This complexity is driven by the cost of sorting whilst the recording of indices is linear in p. We note, however, that looping through the set of equal pairs in order to output a list of close pairs of the form $(j_1, k_1), \ldots, (j_{|E|}, k_{|E|})$ would incur an additional cost of the size of E, though in typical usage we would have $|E| = o(p)$. Readers familiar with locality sensitive hashing (LSH) can find a short interpretation of equal pairs search as an LSH-family in the appendix. In the next section, we discuss in detail the impact of minimal subsampling on the complexity of the *xyz* algorithm and the discovery probability it attains.

2.3 Computational and statistical properties of xyz

We have the following upper bound on the expected number of computational operations performed by *xyz* (Algorithm 2) when the subsample size and number of repetitions are M and L:

$$C(M,L) := \underset{\text{(i)}}{np} + L\{\underset{\text{(ii)}}{Mp} + \underset{\text{(iii)}}{p\log(p)} + \underset{\text{(iv)}}{n\mathbb{E}_\xi(|E_1|)}\}. \tag{10}$$

The terms may be explained as follows: (i) construction of $\mathbf{Z}$; (ii) multiplying M subsampled rows of $\mathbf{X}$ and $\mathbf{Z}$ by $\mathbf{R} \in \mathbb{R}^n$; (iii) finding the equal pairs; (iv) checking whether the interactions exceed the interaction strength threshold γ. Note we have omitted a constant factor from the upper bound $C(M,L)$. There is a lower bound only differing from (10) in the equal pairs search term (iii), which is p instead of $p\log(p)$. It will be shown that (iv) is the dominating term and therefore the upper and lower bound are asymptotically equivalent, implying the bounds are tight.

An interaction with strength γ is retained in E_1 with probability γ^M. Hence it is present in the final set of interactions I with probability

$$\eta(M,L) = 1 - (1 - \gamma^M)^L. \tag{11}$$

The following result demonstrates how the *xyz* algorithm can be used to find interactions whilst incurring only a subquadratic computational cost.

Theorem 2 *Let F_Γ be the distribution function corresponding to a random draw from the set of interaction strengths $\{\gamma_{jk}\}_{j,k\in\{1,\dots,p\}}$. Given an interaction strength threshold γ, let $1-F_\Gamma(\gamma) = c_1/p$. Define $\gamma_0 = p^{-1/M}$ and let c_2 be defined by $1-F_\Gamma(\gamma_0) = c_2 p^{\log(\gamma)/\log(\gamma_0)-1}$. We assume that $\gamma_0 < \gamma$. Finally given a discovery threshold $\eta' \in [1/2, 1)$ let L be the minimal L' such that $\eta(M, L') \geq \eta'$. Ignoring constant factors we have*

$$C(M, L) \leq \log\{1/(1-\eta')\}(1 + c_1 + c_2)[\{1 + 1/\log(\gamma_0^{-1})\}\log(p) + n]p^{1+\log(\gamma)/\log(\gamma_0)}.$$

If $n \gg \log(p)$ and γ_0 is bounded away from 1 we see that the dominant term in the above is

$$cnp^{1+\log(\gamma)/\log(\gamma_0)}, \tag{12}$$

where $c = \log\{1/(1-\eta')\}(1 + c_1 + c_2)$. Typically we would expect γ to be such that $|\{\gamma_{jk} : \gamma_{jk} > \gamma\}| \sim p$ as only the largest interactions would be of interest: thus we may think of c_1 as relatively small. If M is such that γ_0 is also larger than the bulk of the interactions, we would also expect c_2 to be small. Indeed, suppose that the proportion of interactions whose strengths are larger than γ_0 is $1 - F_\Gamma(\gamma_0) = c_1'/p$. Then $c_2 = c_1'/p^{\log(\gamma)/\log(\gamma_0)} < c_1'$. As a concrete example, if $\gamma = 0.9$ and M is such that $\gamma_0 = 0.55$, the exponent in (12) is around 1.17, which is significantly smaller than the exponent of 2 that a brute-force approach would incur; see also the examples in Section 5. Note also that when $\gamma = 1$, the exponent is 1 for all $\gamma_0 < 1$: if we are only interested in interactions whose strength is as large as possible, we have a run time that is linear in p.

It is interesting to compare our results here with the run times of approaches based on fast matrix multiplication. By computing $\mathbf{X}^T\mathbf{Z}$ we may solve the interaction search problem (1). Naive matrix multiplication would require $\mathcal{O}(np^2)$ operations, but there are faster alternatives when $n = p$. The fastest known algorithm (Williams, 2012) gives a theoretical run time of $\mathcal{O}(np^{1.37})$ when $n = p$. For *xyz* to achieve such a run time when $\gamma_0 = 0.55$ for example, the target interaction strength would have to be $\gamma \geq 0.81$: a somewhat moderate interaction strength. For $\gamma > 0.81$, *xyz* is strictly better; we also note that fast matrix multiplication algorithms tend to be unstable or lack a known implementation and are therefore rarely used in practice. A further advantage is that the *xyz* algorithm has an optimal memory usage of $\mathcal{O}(np)$.

We also note that whilst Theorem 2 concerns the the discovery of any single interaction with strength at least γ, the run time required to discover a fixed number interactions with strength at least γ would only differ by a multiplicative constant. If we however want a guarantee of discovering the p strongest pairs the bound in Theorem 2 would no longer hold.

To minimise the run time in (12), we would like γ_0 to be larger than most of the interactions in order that c_2 and hence c be small, yet a smaller γ_0 yields a more favourable exponent. Thus a careful choice of M, on which γ_0 depends, is required for *xyz* to enjoy good performance. In the following we show that an optimal choice of M exists, and we discuss how this M may be estimated based on the data.

Clearly if for some pair (M, L), we find another pair (M', L') with $\eta(M', L') > \eta(M, L)$ but $C(M', L') \leq C(M, L)$, we should always use (M', L') rather than (M, L). It turns out

that there is in fact an optimal choice of M such that the parameter choice is not dominated by any others in this fashion. Define

$$M^* = \operatorname*{arg\,min}_{M\in\mathbb{N}} \left\{ -\frac{1}{\log(1-\gamma^M)}\left(Mp + p\log(p) + n\sum_{j,k}\gamma_{jk}^M\right)\right\}, \tag{13}$$

where it is implicitly assumed that the minimiser is unique. This will always be the case except for peculiar values of γ.

Proposition 3 *Let $L \in \mathbb{N}$. If $(M', L') \in \mathbb{N}^2$ has $\eta(M', L') \geq \eta(M^*, L)$, then also $C(M', L') \geq C(M^*, L)$ with the final inequality being strict if $M' \neq M^*$ and M^* is a unique minimiser.*

Thus there is a unique Pareto optimal M. Although the definition of M^* involves the moments of F_Γ, this can be estimated by sampling from $\{\gamma_{jk}\}$. We can then numerically optimise a plugin version of the objective to arrive at an approximately optimal M.

3. Interaction search on continuous data

In the previous section we demonstrated how the *xyz* algorithm can be used to efficiently solve the simplest form of interaction search (1) when both $\mathbf{X}$ and $\mathbf{Y}$ are binary. In this section we show how small modifications to the basic algorithm can allow it to do the same when $\mathbf{Y}$ is continuous, and also when $\mathbf{X}$ is continuous. We discuss the regression setting in Section 4.

3.1 Continuous Y and binary $\mathbf{X}$

We begin by considering the setting where $\mathbf{X} \in \{-1, 1\}^{n\times p}$, but where we now allow real-valued $\mathbf{Y} \in \mathbb{R}^n$. Without loss of generality, we will assume $\|\mathbf{Y}\|_1 = 1$. The approach we take is motivated by the observation that the inner product $\mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)$ can be interpreted as a weighted inner product of $\mathbf{X}_j \circ \mathbf{X}_k$ with the sign pattern of $\mathbf{Y}$, using weights $w_i = |Y_i|$.

With this in mind, we modify *xyz* in the following way. We set $\mathbf{Z}$ to be $Z_{ij} = \operatorname{sgn}(Y_i)X_{ij}$. Let $i_1, \dots, i_M \in \{1, \dots, n\}$ be i.i.d. such that $\mathbb{P}(i_s = i) = w_i$. Forming the projection vector $\mathbf{R}$ using (6), we then find the probability of (j, k) being in the equal pairs set may be computed as follows.

$$\begin{aligned}
\{\mathbb{P}(\mathbf{R}^T\mathbf{X}_j = \mathbf{R}^T\mathbf{Z}_k)\}^{1/M} &= \mathbb{P}\left(X_{i_s j} = \operatorname{sgn}(Y_{i_s})X_{i_s k} \text{ for all } s = 1, \dots, M\right) \\
&= \mathbb{P}(X_{i_1 j} = \operatorname{sgn}(Y_{i_1})X_{i_1 k}) \qquad \text{as the } i_s \text{ are i.i.d.} \\
&= \sum_{i=1}^n \mathbb{P}(X_{i_1 j} = \operatorname{sgn}(Y_{i_1})X_{i_1 k} | i_1 = i)\mathbb{P}(i_1 = i) \\
&= \sum_{i=1}^n |Y_i| \mathbb{1}_{\{X_{ij} = \operatorname{sgn}(Y_i)X_{ik}\}} \\
&= \sum_{i:\operatorname{sgn}(Y_i) = X_{ij}X_{ik}} Y_i X_{ij} X_{ik} =: \tilde{\gamma}_{jk},
\end{aligned}$$

where $\mathbb{P}$ here is with respect to the randomness of $\mathbf{R}$ (and, equivalently, the random indices $i_1, \dots, i_M$) with $\mathbf{Y}$ and $\mathbf{X}$ considered fixed. The calculation above shows that the run time

bound of Theorem 2 continues to hold in the setting with continuous $\mathbf{Y}$ provided we replace the interaction strengths γ_{jk} with their continuous analogues $\tilde{\gamma}_{jk}$.

As a simple example, consider the model

$$Y_i = X_{i1}X_{i2} + \varepsilon_i,$$

with $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$ and $\mathbf{X}$ generated randomly having each entry drawn independently from $\{-1, 1\}$ each with probability $1/2$. Then for a non-interacting pair $j \neq 1, 2$ or $k \neq 1, 2$, we have $\tilde{\gamma}_{jk} \approx 0.5$. For the pair $(1, 2)$ we calculate an interaction strength of

$$\begin{aligned}\tilde{\gamma}_{12} &= \mathbb{P}(\operatorname{sgn}(Y_{i_1}) = X_{i_1 1}X_{i_1 2}) = \mathbb{P}(\operatorname{sgn}(X_{i_1 1}X_{i_1 2} + \varepsilon_i) = X_{i_1 1}X_{i_1 2}) \\ &= \mathbb{P}(|\varepsilon_i| < 1) + \frac{1}{2}\mathbb{P}(|\varepsilon_i| > 1) = \frac{1}{2}(1 + \mathbb{P}(|\varepsilon_i| < 1)).\end{aligned}$$

Note that here that probability is over the randomness in the noise ε_i. A quick simulation gives the following table:

σ^2	0.1	0.25	0.5	1	2	5
$\tilde{\gamma}_{12}$	0.99	0.98	0.92	0.84	0.76	0.67

Using Theorem 2 and the above table we can estimate the computational complexity needed to discover the pair $(1, 2)$ given a value of σ^2.

3.2 Continuous Y and continuous X

The previous section demonstrated how resampling with non-uniform weights transforms a setup with continuous $\mathbf{Y}$ into one with binary response. If both $\mathbf{X}$ and $\mathbf{Y}$ are continuous, we continue to use the previous strategy to deal with the continuous response. For the matrix $\mathbf{X}$ with continuous predictor values we cannot use weighted resampling as the weights would depend on the interaction pair of interest. In the following we examine the effects of transformations of $\mathbf{X}$ to a binary data matrix $\tilde{\mathbf{X}}$. To allow for randomized mappings, we define the transformations via a function $g : \mathbb{R} \mapsto [0, 1]$ as

$$\mathbb{P}(\tilde{X}_{ij} = 1) = g(X_{ij}) \text{ and } 1 - \mathbb{P}(\tilde{X}_{ij} = -1) = 1 - g(X_{ij}),$$

where the transformation is always applied independently for each entry of the predictor matrix and for each subsample.

The following gives the probability of Y_i agreeing in sign with $\tilde{X}_{ij}\tilde{X}_{ik}$ when i is sampled with probability proportional to $|Y_i|$.

Proposition 4 *Given the transform $\mathbb{P}(\tilde{X}_{ij} = 1) = g(X_{ij})$ and sampling an index i_s according to $\mathbb{P}(i_s = i) = Y_i/\|\mathbf{Y}\|_1$, then the probability of a match is*

$$\mathbb{P}(\operatorname{sgn}(Y_{i_s}) = \tilde{X}_{i_s j}\tilde{X}_{i_s k}) = \frac{1}{2} + \frac{1}{2\|\mathbf{Y}\|_1}\sum_{i=1}^{n} Y_i(1 - 2g(X_{ij}))(1 - 2g(X_{ik})). \tag{14}$$

Thus we may define a continuous analogue of the interaction strength γ_{jk} based on the transform given by g as

$$\gamma^g_{jk} = \frac{1}{2} + \frac{1}{2\|\mathbf{Y}\|_1}\sum_{i=1}^{n} Y_i(1 - 2g(X_{ij}))(1 - 2g(X_{ik})).$$

These quantities may be substituted into Theorem 2 to yield the following upper bound on expected run time when using *xyz* on transformed data.

Corollary 5 *Let F_{Γ^g} be the distribution function corresponding to a random draw from the set of interaction strengths $\{\gamma^g_{jk}\}_{j,k\in\{1,\dots,p\}}$. Given an interaction strength threshold γ, let $1-F_{\Gamma^g}(\gamma) = c_1/p$. Define $\gamma_0 = p^{-1/M}$ and let c_2 be defined by $1-F_\Gamma(\gamma_0) = c_2 p^{\log(\gamma)/\log(\gamma_0)-1}$. We assume that $\gamma_0 < \gamma$. Finally given a discovery threshold $\eta' \in [1/2, 1)$ let L be the minimal L' such that $\eta(M, L') \geq \eta'$. Ignoring constant factors we have*

$$C(M, L) \leq \log\{1/(1-\eta')\}(1 + c_1 + c_2)[\{1 + 1/\log(\gamma_0^{-1})\}\log(p) + n]p^{1+\log(\gamma)/\log(\gamma_0)}.$$

The expected computational costs depends critically on the distribution of the interaction strengths F_{Γ^g}. To gain a better understanding of what impact different transformations have on this distribution and subsequently on run time we will study the following simple model for $(\mathbf{Y}, \mathbf{X}) \in \mathbb{R}^n \times \mathbb{R}^{n\times p}$:

$$Y_i = X_{ij^*}X_{ik^*} + \varepsilon_i, \qquad i = 1, \dots, n, \tag{15}$$

where the ε_i are independent and have identical sub-exponential distributions symmetric about 0 and the rows of $\mathbf{X}$ are i.i.d. We now introduce two practically useful choices of g and study their properties in the context of model (15).

The unbiased transform

A natural choice for the transform g is one that satisfies the unbiasedness requirement:

$$\mathbb{E}(\tilde{X}_{ij}) = X_{ij}. \tag{16}$$

It turns out that this requirement uniquely defines the transform, which we refer to as the *unbiased transform*.

Proposition 6 *Let $X_{ij} \in [-1, 1]$. If its transformed version $\tilde{X}_{ij}$ satisfies* (16)*, then g takes the form*

$$\mathbb{P}(\tilde{X}_{ij} = 1) = g(X_{ij}) = \frac{X_{ij} + 1}{2}.$$

Furthermore the interaction strength in (14) *is given by*

$$\mathbb{P}(\operatorname{sgn}(Y_{i_s}) = \tilde{X}_{i_s j}\tilde{X}_{i_s k}) = \gamma^g_{jk} = \frac{1}{2} + \frac{1}{2\|\mathbf{Y}\|_1}\sum_{i=1}^{n} Y_i X_{ij} X_{ik}.$$

Proposition 6 shows that γ^g_{jk} is a monotone function of the inner product $\sum_{i=1}^n Y_i X_{ij} X_{ik}$.

We remark that if the entries of $\mathbf{X}$ do not lie in $[-1, 1]$, we may divide each entry in the ith row by $\nu_i := \max_j |X_{ij}|$, and multiply Y_i by ν_i^2, for each i. Proposition 6 will then hold for the scaled versions of $\mathbf{Y}$ and $\mathbf{X}$. In order to describe the performance of the unbiased transform when applied to data generated by the model (15), we define the following quantities:

$$\mathbb{E}(|X_{ij^*}X_{ik^*}|) = m_1, \;\; \mathbb{E}(X^2_{ij^*}X^2_{ik^*}) = m_2 \text{ and } \mathbb{E}(|\varepsilon_i|) = m_\varepsilon.$$

We consider an asymptotic regime where $p = p_n$ may diverge as n tends to infinity, though we suppress this in the notation. We introduce the following assumptions.

(B1) $m_2(r_u-1) \leq \mathbb{E}(X_{ij^*}X_{ik*}X_{ij}X_{ik}) \leq m_2(1-r_u)$, for $r_u \in (0,1)$ and $\forall\, j,k \in \{1,\ldots,p\}^2$.

(B2) The noise level satisfies the bound

$$\frac{1}{1-r_u} > 1 + \frac{m_\epsilon}{m_1}.$$

(B3) Let p be such that be such that

$$\frac{\log(n)\log(p)}{n} \overset{n\to\infty}{\to} 0.$$

(B1) ensures non-interactions are not too strongly correlated to the actual interaction pair (j^*, k^*). Note that (B3) allows for high-dimensional settings with $p \gg n$.

Theorem 7 *Assume all entries of* $\mathbf{X}$ *have mean zero and lie in* $[-1,1]$ *almost surely. Further assume (B1)–(B3) hold. When* M *and* L *are as in Corollary 5 and the unbiased transform is used, we have*

$$C(M,L) = o_{\mathbb{P}}\Big(np^{1+\delta+\frac{\log(1/2+m_2/2(m_1+m_\varepsilon))}{\log(1/2+m_2(1-r_u)/2m_1)}}\Big)$$

for any $\delta > 0$. *Here* $\mathbb{P}$ *is with respect to the randomness in* $\mathbf{X}$ *and* $\boldsymbol{\varepsilon}$.

Though the run time above can often improve significantly on the worst-case quadratic run time, observe that unlike in the binary case, if there is no noise and $Y_i = X_{ij^*}X_{ik^*}$, we do not necessarily have a run time close to linear in p. For example, when $X_{ij} \overset{iid}{\sim} \text{Uniform}(-1,1)$, the interaction strength of the true interaction can be shown to equal to

$$\gamma^g_{j^*k^*} = \frac{1}{2} + \frac{\sum_{i=1}^n Y_i X_{ij^*} X_{ik^*}}{2\|\mathbf{Y}\|_1} = \frac{1}{2} + \frac{\|\mathbf{Y}\|_2^2}{2\|\mathbf{Y}\|_1} \overset{n\to\infty}{=} \frac{13}{18}.$$

Substituting this into the run time given by Theorem 2, this would result in an expected complexity of roughly $\mathcal{O}(np^{1.47})$; this is still substantially smaller than a quadratic run time, but raises the question as to whether such a loss in speed is avoidable.

Additionally, if $\mathbf{X}$ has several outlying entries, normalising the design matrix by scaling by the row-wise maximums can shrink $\gamma^g_{j^*k^*}$ towards $1/2$. To limit the impact of this normalisation, we can first cap the entries of $\mathbf{X}$ so their absolute value is bounded by some $c > 0$. Though the resulting interaction strength will not have the form given in Proposition 6, it may better discriminate between interactions of interest and noise.

Capping with $c = 1$ is closely related to applying the sign transform, which we study next.

The sign transform

We now consider the *sign transform* given by $\tilde{X}_{ij} = \text{sgn}(X_{ij})$; if there are zero cases we use a coin toss to map them to $\{-1,1\}$. For the sign transform we have $g(X_{ij}) = 2\,\text{sgn}(X_{ij}) - 1$ and so the interaction strength is given as:

$$\mathbb{P}(\text{sgn}(Y_{i_s}) = \tilde{X}_{i_s j}\tilde{X}_{i_s k}) = \gamma^g_{jk} = \frac{1}{2} + \frac{1}{2\|\mathbf{Y}\|_1}\sum_{i=1}^n Y_i \text{sgn}(X_{ij})\text{sgn}(X_{ik}).$$

The sign transform recovers the close to linear run time achieved in the binary case when a interaction is perfect as now if $Y_i = X_{ij^*}X_{ik^*}$, we have $\gamma^g_{j^*k^*} = 1$. Also the sign transform is not adversely affected by the presence of outlying entries in $\mathbf{X}$, and for our theory we can relax the assumption that the entries of $\mathbf{X}$ are in $[-1,1]$ to here only requiring that they have a subexponential distribution. To facilitate comparison with the unbiased transform, we impose assumptions analogous to (B1)–(B3):

(C1) $r_s/2 \leq \mathbb{P}(X_{ij} < 0 | X_{ik}, X_{ij^*}, X_{ik^*}) \leq 1 - r_s/2$, for $r_s \in (0,1)$ and $\forall\, j,k \in \{1,...,p\}^2$.

(C2) The noise level satisfies

$$\frac{1}{1-r_s} > 1 + \frac{m_\epsilon}{m_1}.$$

(C3) Let p be such that

$$\frac{\log(p)^5}{n} \overset{n\to\infty}{\to} 0.$$

Theorem 8 *Suppose that each entry of* $\mathbf{X}$ *has a mean-zero subexponential distribution. Further assume (C1)–(C3). When* M *and* L *are as in Corollary 5 and the sign transform is used, we have*

$$C(M,L) = o_{\mathbb{P}}\Big(np^{1+\delta+\frac{\log(1/2+m_1/2(m_1+m_\varepsilon))}{\log(1-r_s)}}\Big)$$

for any $\delta > 0$. *Here* $\mathbb{P}$ *is with respect to the randomness in* $\mathbf{X}$ *and* ε.

Both transforms yield a run time of the form $o_{\mathbb{P}}(np^\alpha)$. Comparing the exponents α we have:

unbiased transform:

$$\alpha_u = 1 + \frac{\log(1/2 + m_2/2(m_1+m_\varepsilon))}{\log(1/2 + m_2(1-r_u)/2m_1)}$$

sign transform:

$$\alpha_s = 1 + \frac{\log(1/2 + m_1/2(m_1+m_\varepsilon))}{\log(1/2 + (1-r_s)/2)}.$$

For bounded data $\mathbf{X} \in [-1,1]^{n\times p}$ and when $m_\varepsilon \ll m_1$, we have $m_1/2(m_1+m_\varepsilon) \approx 1/2$ so that $\alpha_s = 1$ whereas $\alpha_u > 1$. Hence in case of a strong signal the sign transform can give a smaller run time than the unbiased transform.

4. Application to Lasso regression

Thus far we have only considered the simple version of the interaction search problem (1) involving finding pairs of variables whose interaction has a large dot product with $\mathbf{Y}$. In this section we show how any solution to this, and in particular the *xyz* algorithm, may be used to fit the Lasso (Tibshirani, 1996) to all main effects and pairwise interactions in an efficient fashion.

Given a response $\mathbf{Y} \in \mathbb{R}^n$ and a matrix of predictors $\mathbf{X} \in \mathbb{R}^{n\times p}$, let $\mathbf{W} \in \mathbb{R}^{n\times p(p+1)/2}$ be the matrix of interactions defined by

$$\mathbf{W} = (\mathbf{X}_1 \circ \mathbf{X}_1, \mathbf{X}_1 \circ \mathbf{X}_2, \cdots, \mathbf{X}_1 \circ \mathbf{X}_p, \mathbf{X}_2 \circ \mathbf{X}_2, \mathbf{X}_2 \circ \mathbf{X}_3, \cdots, \mathbf{X}_p \circ \mathbf{X}_p).$$

We will assume that $\mathbf{Y}$ and the columns of $\mathbf{X}$ have been centred. Note that the centring of $\mathbf{X}$ means the $\mathbf{W}$ implicitly contains main effects terms. Let $\tilde{\mathbf{W}}$ be a version of $\mathbf{W}$ with centred columns. Consider the Lasso objective function

$$(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}}) = \underset{\boldsymbol{\beta}\in\mathbb{R}^p, \boldsymbol{\theta}\in\mathbb{R}^{p(p+1)/2}}{\operatorname{argmin}} \left\{ \frac{1}{2n} \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta} - \tilde{\mathbf{W}}\boldsymbol{\theta}\|_2^2 + \lambda(\|\boldsymbol{\beta}\|_1 + \|\boldsymbol{\theta}\|_1) \right\}. \tag{17}$$

Note that since the entire design matrix in the above is column-centred, any intercept term would always be zero.

In order to avoid a cost of $\mathcal{O}(np^2)$ it is necessary to avoid explicitly computing $\mathbf{W}$. To describe our approach, we first review in Algorithm 3 the active set strategy employed by several of the fastest Lasso solvers such as `glmnet` (Friedman et al., 2010). We use the notation that for a matrix $\mathbf{M}$ and a set of column indices H, $\mathbf{M}_H$ is the submatrix of $\mathbf{M}$ formed from those columns indexed by H. Similarly for a vector $\mathbf{v}$ and component indices H, $\mathbf{v}_H$ is the subvector of $\mathbf{v}$ formed from the components of $\mathbf{v}$ indexed by H.

Algorithm 3 Active set strategy for Lasso computation

Input: $\mathbf{X}$, $\mathbf{Y}$ and grid of λ values $\lambda_1 > \cdots > \lambda_L$.
Output: Lasso solutions $\hat{\boldsymbol{\beta}}_{\lambda_l}$ and $\hat{\boldsymbol{\theta}}_{\lambda_l}$ at each λ on the grid.

1: **for** $l \in \{1, \ldots, L\}$ **do**
2: If $l = 1$ set $A, B = \emptyset$; otherwise set $A = \{k : \hat{\beta}_{\lambda_{l-1},k} \neq 0\}$ and $B = \{k : \hat{\theta}_{\lambda_{l-1},k} \neq 0\}$.
3: Compute the Lasso solution $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}})$ when $\lambda = \lambda_l$ under the additional constraint that $\hat{\boldsymbol{\beta}}_{A^c} = 0$ and $\hat{\boldsymbol{\theta}}_{B^c} = 0$.
4: Let $U = \{k : |\mathbf{X}_k^T(\mathbf{Y} - \mathbf{X}_A\hat{\boldsymbol{\beta}}_A - \tilde{\mathbf{W}}_B\hat{\boldsymbol{\theta}}_B)|/n > \lambda_l\}$ and $V = \{k : |\tilde{\mathbf{W}}_k^T(\mathbf{Y} - \mathbf{X}_A\hat{\boldsymbol{\beta}}_A - \tilde{\mathbf{W}}_B\hat{\boldsymbol{\theta}}_B)|/n > \lambda_l\}$ be the set of coordinates that violate the KKT conditions when $(\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\theta}})$ is taken as a candidate solution.
5: If U and V are empty, we set $\hat{\boldsymbol{\beta}}_{\lambda_l} = \hat{\boldsymbol{\beta}}$, $\hat{\boldsymbol{\theta}}_{\lambda_l} = \hat{\boldsymbol{\theta}}$. Else we update $A = A \cup U$ and $B = B \cup V$ and return to line 3.
6: **end for**

As the sets A and B would be small, computation of the Lasso solution in line 3 is not too expensive. Instead line 4, which performs a check of the Karush–Kuhn–Tucker (KKT) conditions involving dot products of all interaction terms and the residuals, is the computational bottleneck: a naive approach would incur a cost of $\mathcal{O}(np^2)$ at this stage.

There is however a clear similarity between the KKT conditions check for the interactions and the simple interaction search problem (1). Indeed the computation of V, the set containing all interactions that violate the KKT conditions, may be expressed in the following way:

$$\text{Keep all pairs } (j,k) \text{ for which } |(\mathbf{Y} - \mathbf{X}_A\hat{\boldsymbol{\beta}}_A - \tilde{\mathbf{W}}_B\hat{\boldsymbol{\theta}}_B)^T(\mathbf{X}_j \circ \mathbf{X}_k)/n| > \lambda_l. \tag{18}$$

Note that since $\mathbf{Y} - \mathbf{X}_A\hat{\boldsymbol{\beta}}_A - \tilde{\mathbf{W}}_B\hat{\boldsymbol{\theta}}_B$ is necessarily centered, there is no need to center the interactions in (18). In order to solve (18) we can use the *xyz* algorithm, setting γ in Algorithm 2 to λ_l and $\mathbf{Y}$ to each of $\pm(\mathbf{Y} - \mathbf{X}_A\hat{\boldsymbol{\beta}}_A - \tilde{\mathbf{W}}_B\hat{\boldsymbol{\theta}}_B)$ in turn.

Precisely the same strategy of performing KKT condition checks using *xyz* can be used to accelerate computation for interaction modeling for a variety of variants of the Lasso such

as the elastic net (Zou and Hastie, 2005) and ℓ_1-penalised generalised linear models. Note also that it is straightforward to use a different scaling for the penalty on the interaction coefficients in (17), which may be helpful in practice.

5. Experiments

To test the algorithm and theory developed in the previous sections, we run a sequence of experiments on real and simulated data.

5.1 Comparison of minimal subsampling and dense projections

One of the surprising outcomes of our theoretical analysis is extent of the suboptimality of Gaussian random projections, which whilst they suffice for the conclusion of the Johnson–Lindenstrauss Lemma, are not well-suited for our purposes here (see Theorem 1). We can explicitly compute the probability of retaining an interaction of strength γ in E_1 for both dense Gaussian projections ξ_{Gauss} and minimal subsampling $\xi_{minimal}$ given an equal computational budget. We consider various values of p ranging from 10 up to 10^6 and we fix $n = 1000$. We set $L = 1$ and select other parameters of the algorithms to ensure the average size of E_1 is equal to p in the setting when all interaction strengths are equal to 0.5. Specifically we make the following choices.

- ξ_{Gauss}: the close pairs threshold $\tau \geq 0$ is the $1/p$–quantile of the distribution of $|W|$ when $W \sim N(0, 0.5n)$.
- $\xi_{minimal}$: the subsample size $M = \lceil \log(1/p)/\log(0.5) \rceil$.

We then plot the probability η of discovering an interaction of strength γ, as a function of γ for different values of p (Figure 3). For $\xi_{minimal}$, η is given in equation (11). For ξ_{Gauss}, η is the $1/p$–quantile of the distribution of $|W|$ when $W \sim N(0, n(1-\gamma))$.

5.2 Scaling

In this experiment we test how the *xyz* algorithm scales on a simple test example as we increase the dimension p. We generate data $\mathbf{X} \in \mathbb{R}^{n\times p}$ with each entry sampled independently uniformly from $\{-1, 1\}$. We do this for different values of p, ranging from 1000 to 30 000: this way for the largest p considered there are more than 400 million possible interactions. Then for each $\mathbf{X}$ we construct response vectors $\mathbf{Y}$ such that only the pair $(1, 2)$ is a strong interaction with an interaction strength taking values in $\{0.7, 0.8, 0.9\}$. Through this construction, if n is large enough, all the pairs except $(1, 2)$ will have an interaction strength around 0.5, and very few will have one above 0.55. We thus set M so that $\gamma_0 = p^{-1/M} \approx 0.55$. Since the only strong interaction is $(1, 2)$, we set $\gamma = \gamma_{12}$ Each data set configuration determined by p and γ_{12} is simulated 300 times and we measure the time it takes *xyz* to find the pair $(1, 2)$. In Figure 3 we plot the average run time against the dimension p with the different choices for γ_{12} highlighted in different colours.

Theorem 2 indicates that the run time should be of the order $np^{1+\log(\gamma)/\log(\gamma_0)}$. We see that the experimental results here are in close agreement with this prediction.

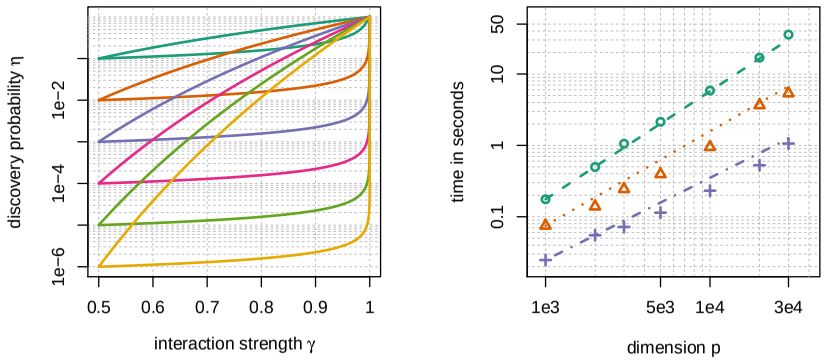


Figure 3: Left panel: Discovery probability as a function of γ for different values of $p \in \{10^1, \ldots, 10^6\}$ (colours decreasing in p from yellow $p = 10^6$ to green $p = 10$). The lower lines correspond to the dense Gaussian projections, the upper lines to minimal subsampling. It can be seen that the discovery probability for minimal subsampling is much higher (up to factor 10^4) than for Gaussian projections. Right panel: Time to discover the interaction pair as a function of the data set dimension p. Lines correspond to the theoretical prediction (with the intercept chosen based on the data points) and symbols give the actual measured run time. Colour coding: green $\gamma = 0.7$, orange $\gamma = 0.8$ and purple $\gamma = 0.9$.

5.3 Run on SNP data

In the next experiment we compare the performance of *xyz* to its closest competitors on a real data set. For each method we measure the time it takes to discover strong interactions. We consider the LURIC data set (Winkelmann et al., 2001), which contains data of patients that were hospitalised for coronary angiography. We use a preprocessed version of the data set that is made up of $n = 859$ observations and 687 253 predictors. The data set is binary. The response $\mathbf{Y}$ indicates coronary disease (1 corresponding to affected and -1 healthy) and $\mathbf{X}$ contains Single Nucleotide Polymorphisms (SNPs) which are variations of base-pairs on DNA. The response vector $\mathbf{Y}$ is strongly unbalanced: there are 681 affected cases ($Y_i = 1$) and 178 unaffected ($Y_i = -1$).

To get a contrast of the performance of *xyz* we compare it to *epiq* (Arkin et al., 2014), another method for fast high-dimensional interaction search. In order for *epiq* to detect interactions it needs to assume the model

$$Y_i = \alpha_{j^*k^*} X_{ij^*} X_{ik^*} + \varepsilon_i, \tag{19}$$

where $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$. It then searches for interactions by considering the test statistics

$$T_{jk} = (\mathbf{R}^T(\mathbf{Y} \circ \mathbf{X}_j))(\mathbf{R}^T\mathbf{X}_k)$$

where $\mathbf{R} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. These are used to try to find the pair (j^*, k^*), which is assumed to be the pair for which the inner product $\mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)$ is maximal. It is an easy calculation to show

that $\mathbb{E}(T_{jk}) = \mathbf{Y}^T(\mathbf{X}_j \circ \mathbf{X}_k)$. To maximise the inner product on the right, *epiq* considers pairs where T_{jk}^2 is large by looking at pairs where both $(\mathbf{R}^T(\mathbf{Y} \circ \mathbf{X}_j))^2$ and $(\mathbf{R}^T\mathbf{X}_k)^2$ are large. While the approach of *epiq* is somewhat related to *xyz*, there are no bounds available for the time it takes to find strong interactions.

We also compare both methods to a naive approach where we subsample a fixed number of interactions uniformly at random, and retain the strongest one. We refer to this as *naive search*.

At fixed time intervals we check for the strongest interaction found so far with all three methods. We plot the interaction strength as a function of the computational time (Figure 4). All three methods eventually discover interactions of very similar strength and it would be a hasty judgement to say whether one significantly outperforms the others. *xyz* nevertheless discovers the strongest interactions on average for a fixed run time compared to the other two approaches. To get a clearer picture we run two additional experiments on a slight modification of the LURIC data set. We implant artificial interactions where we set the strength to $\gamma_{12} = 0.8$ and another example with $\gamma_{12} = 0.9$. In these two experiments *xyz* clearly outperforms all other methods considered (Figure 4; panels 3 and 4). Besides *xyz* being the fastest at interaction search, it also offers a probabilistic guarantee that there are no strong interactions left in the data. This guarantee comes out of Theorem 2. To run *xyz* we have to calculate the optimal subsample size (13) for use of minimal subsampling:

$$M^* = \operatorname*{arg\,min}_{M \in \mathbb{N}} \left\{ -\frac{1}{\log(1 - \gamma^M)} \left(Mp + p\log(p) + n \sum_{j,k} \gamma_{jk}^M \right) \right\} = 21.$$

The sum in this optimisation can be approximated by uniformly sampling over pairs. Assume we have an interaction pair (j^*, k^*) with interaction strength $\gamma_{j^*k^*} = 0.85$ and say the rest of the pairs (j, k) have an interaction strength of no more than $\gamma_{jk} \leq 0.55$. The probability that we discover this pair in one run ($L = 1$) of the *xyz* algorithm is $\gamma_{j^*k^*}^{21}$. Therefore the probability of missing this pair after $L = 100$ runs is given by

$$(1 - \gamma_{j^*k^*}^{21})^L \approx 0.03.$$

Note that the number of possible interactions is $p(p-1)/2 \approx 10^{11}$. The whole search took 280 seconds. Naive search offers a similar guarantee, however it is extremely weak. The probability of not discovering the pair after drawing pL samples (with $L = 100$) is bounded by $[1 - 2/\{p(p-1)\}]^{Lp} \approx 0.999$. If we consider the run time guarantee from Theorem 2, the dominating term in the complexity of *xyz* in terms of p is

$$p^{1 + \frac{\log(0.85)}{\log(0.55)}} \approx p^{1.27}.$$

This may be compared to the expected run time of order p^2 for naive search, which means that *xyz* is about 30 000 times faster than naive search (when $p = 687\,253$). In the empirical comparison this factor is around 20 000.

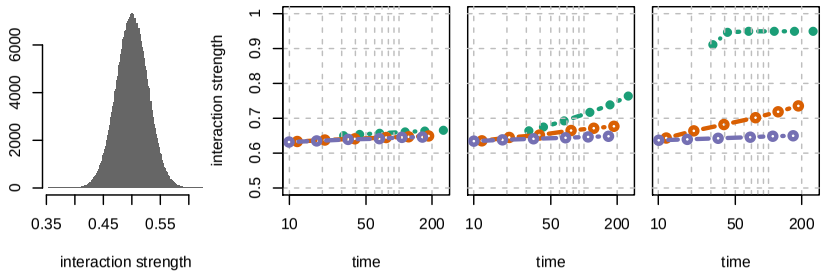


Figure 4: Left: Histogram of interaction strength of 10^6 interaction pairs, sampled at random from the more than 10^{11} existing pairs from the LURIC data set. The right three panels show the interaction strength of the discovered pairs as a function of the computation time for *xyz* (green), *epiq* (orange) and naive search (purple). The first panel gives results on the the original LURIC data set, and the second and third (rightmost) panels show results with an implanted interaction with strengths $\gamma_{12} = 0.8$ and $\gamma_{12} = 0.95$ respectively. It can be clearly seen that *xyz* outperforms its competitors by a large margin.

5.4 Regression on artificial data

In this section we demonstrate the capabilities of *xyz* in interaction search for continuous data as explained in Section 3. We simulate two different models of the form (15):

$$Y_i = \mu + \sum_{j=1}^{p} X_{ij}\beta_j + \sum_{k=1}^{p}\sum_{j=1}^{k-1} X_{ij}X_{ik}\theta_{jk} + \varepsilon_i.$$

We consider three settings. For all three settings we have $n = 1000$. We let $p \in \{250, 500, 750, 1000\}$. Each row of $\mathbf{X}$ is generated i.i.d. as $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$. The magnitudes of both the main and interaction effects are chosen uniformly from the interval $[2, 6]$ (20 main effects and 10 interaction effects) and we set $\varepsilon_i \sim \mathcal{N}(0, 1)$. The three settings we consider are as follows.

1. $\mathbf{\Sigma} = \mathbf{I} \in \mathbb{R}^{p\times p}$, we generate a hierarchical model: $\theta_{jk} \neq 0 \Rightarrow \beta_j \neq 0$ and $\beta_k \neq 0$. We first sample the main effects and then pick interaction effects uniformly from the pairs of main effects.

2. $\mathbf{\Sigma} = \mathbf{I} \in \mathbb{R}^{p\times p}$, we generate a strictly non-hierarchical model: $\theta_{jk} \neq 0 \Rightarrow \beta_j = 0$ and $\beta_k = 0$. We first sample the main effects and then pick interaction effects uniformly from all pairs excluding main effects as coordinates.

3. We repeat the setting 2 with a data set that contains strong correlations. We create a dependence structure in $\mathbf{X}$, by first generating a DAG with on average 10 edges per node. Each node is sampled so that it is a linear function of its parents plus some independent centred Gaussian noise, with a variance of 10% the variance coming from the direct parents. The resulting correlation matrix then unveils for each variable $\mathbf{X}_j$ a substantial number of variables strongly correlated to $\mathbf{X}_j$ (There is usually around 10 variables with a correlation of above 0.9). Such a correlation structure will make it easier to detect pairs of variables whose product can serve as strong predictor of $\mathbf{Y}$, even though it has not been included in the construction of $\mathbf{Y}$.

We run three different procedures to estimate the main and interaction effects.

- **Two-stage Lasso:** We fit the Lasso to the data, and then run the Lasso once more on an augmented design matrix containing interactions between all selected main effects. Complexity analysis of the Least Angle Regression (LARS) algorithm (Efron et al., 2004) suggests the computational cost would be $\mathcal{O}(np\min(n, p))$, making the procedure very efficient. However, as the results show, it struggles in situations such as that given by model 2, where a main effects regression will fail to select variables involved in strong interactions.

- **Lasso with all interactions:** Building the full interaction matrix and computing the standard Lasso on this augmented data matrix. Analysis of the LARS algorithm would suggest the computational complexity would be in the order $\mathcal{O}(np^2\min(n, p^2))$. Nevertheless, for small p, this approach is feasible.

- **xyz:** This is Algorithm 3; we set the parameter L to be $\sqrt{p}$ in order to target the strong interactions.

The experiment (seen in Figure 5) shows that *xyz* enjoys the favourable properties of both its competitors: it is as fast as the two-stage Lasso that gives an almost linear run time in p, and it is about as accurate as the estimator calculated from screening all pairs (brute-force).

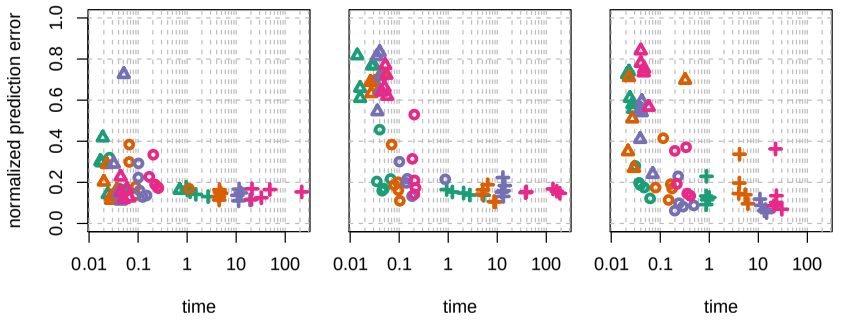


Figure 5: Normalised ℓ_2^2 prediction error as a function of time in seconds. Triangle: Two-stage Lasso. Circle: *xyz*-regression. Cross: Brute-force. The different colours correspond to different values of p: green $p = 250$, orange $p = 500$, purple $p = 750$ and pink $p = 1000$. The left panel shows the results on setting 1, center panel shows setting 2 and right panel setting 3.

5.5 Regression on real data

Here we run *xyz* regression on continuous real data sets where the ground truth is unknown. On each data set we pick at random $p = 2000$ variables and run *xyz* and the Lasso implemented in `glmnet` with all interactions included. We subsample an increasing number of variables to vary the difficulty of the regression problem. For each sample we measure the run time and the normalized out of sample squared ℓ_2^2 error:

$$\frac{\|\mathbf{Y}_{\text{test}} - \mathbf{X}_{\text{test}}\hat{\boldsymbol{\beta}} - \tilde{\mathbf{W}}_{\text{test}}\hat{\boldsymbol{\theta}}\|_2^2}{\|\mathbf{Y}_{\text{test}}\|_2^2}.$$

Experiments are run on the following three different data sets:

- **Riboflavin:** The Riboflavin production data set (Bühlmann et al., 2014) contains $n = 71$ samples and $p = 4088$ predictors (gene-expressions). The response $\mathbf{Y}$ and the design $\mathbf{X}$ are both continuous.
- **Kemmeren:** The Kemmeren (Kemmeren and et al., 2014) data set records knock-outs of $p = 6170$ genes. The data $\mathbf{X}$ is continuous. We sample $\mathbf{Y}$ randomly from the genes not present in the subsample taken from $\mathbf{X}$.
- **Climate:** The climate data set from the CNRM model from the CMIP5 model ensemble (Knutti et al., 2013) simulates the temperature of points on the northern hemisphere which is recorded in $\mathbf{X}$. The response $\mathbf{Y}$ simulates the temperature on a random position on the southern hemisphere. The data contains $n = 231$ observations.

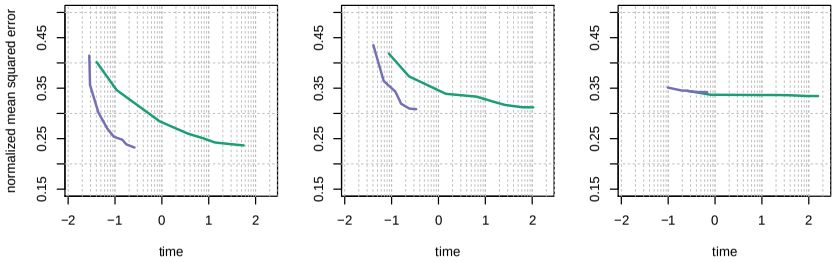


Figure 6: From left to right column the experiments correspond to Riboflavin, Kemmeren and Climate. The y-axis depicts the normalized squared error and the x-axis records the run time in seconds on the $\log_{10}$ scale. It can be seen that *xyz* (purple) offers clear computational advantages while giving similar level of prediction error to the Lasso fitted to all interactions as implemented in `glmnet` (green).

For each experiment we fix the number of runs L to $\sqrt{p}$ so the run time of *xyz* is $\mathcal{O}(np^{1.5})$. The experiments show that the *xyz* algorithm has a similar prediction performance to the Lasso applied to all interactions as implemented in `glmnet`. However *xyz* is around 100 times faster for $p = 2000$. The results of all 6 experiments can be seen in Figure 6.

6. Discussion

In this work we exploited a relationship between closest pairs of point problems and interaction search. By solving the former problem using random projections to project points down to a one-dimensional space and then sorting the resulting projected points, we were able to produce an algorithm for interaction search that enjoys a run time that is sub quadratic under mild assumptions and when used to search for very strong interactions can be almost linear. Though we have looked at interaction search in this paper, the basic engine for computing the large inner products between collections of vectors may have other interesting applications, for example in large-scale clustering problems. We hope to study such applications in future work.

Table of frequently used notation

n, p	number of observations and number of variables
$\mathbf{X}, \mathbf{Y}$	predictor matrix and response vector
$\mathbf{X}_j$	jth variable / column of $\mathbf{X}$
$\boldsymbol{\beta}, \boldsymbol{\theta}$	coefficients of main effects and interaction effects
γ_{jk}	interaction strength of the pair (j, k)
G	distribution of projection
M	subsample size
$\mathbf{R}$	projection vector
L	number of projections
τ, γ	close pairs threshold and interaction strength threshold
Ξ	set of all configurations of the *xyz* algorithm, the elements of this set are denoted by ξ
η	probability that a given interaction is present in the output of the *xyz* algorithm
$\tilde{\mathbf{X}}$	binarized version of $\mathbf{X}$
$\mathbf{W}$	predictor matrix containing all possible interaction pairs

Appendix A

Here we include proofs that were omitted earlier.

Proof of Theorem 1

In the following, we fix the following notation for convenience:

$$\Psi = \Xi_{\text{minimal}}, \qquad \Psi(\eta) = \Xi_{\text{minimal}}(\eta),$$
$$\Xi = \Xi_{\text{subsample}}, \qquad \Xi(\eta) = \Xi_{\text{subsample}}(\eta).$$

Note that both $\Psi(\eta)$ and $\Xi(\eta)$ depend on F though this is suppressed in the notation. Also define $\Xi_{\text{all}} = \Xi \cup \Xi_{\text{dense}}$ and $\Xi_{\text{all}}(\eta) = \Xi(\eta) \cup \Xi_{\text{dense}}(\eta)$. We will reference the parameters levels contained in $\xi \in \Xi_{\text{all}}$ as ξ_L and ξ_τ. If $\xi \in \Xi$ then we will write ξ_M for the distribution of the subsample size M.

If we let V denote the complexity of the search for τ-close pairs, similarly to (10) we have that

$$T(\xi) = c_1 np + L(c_2 \mathbb{E}_\xi M p + \mathbb{E}_\xi V + c_3 n \mathbb{E}_\xi |E_1|), \tag{20}$$

where c_1, c_2, c_3 are constants. Suppose $\psi \in \Psi$ and $\xi \in \Xi$ have $\mathbb{E}_\xi |E_1| = \mathbb{E}_\psi |E_1|$. Then since searching for τ-close pairs is at least as computationally difficult as finding equal pairs we know that $\mathbb{E}_\xi V \geq \mathbb{E}_\psi V$.

Similarly for $\xi \in \Xi_{\text{dense}}$ we have

$$T(\xi) = c_1 np + L(c_2 np + \mathbb{E}_\xi V + c_3 n \mathbb{E}_\xi |E_1|). \tag{21}$$

For $\xi \in \Xi_{\text{all}}$, define

$$\alpha(\xi) = \mathbb{E}_\xi |E_1| / p^2, \qquad \beta(\xi) = \mathbb{P}_\xi((j^*, k^*) \in I_1)$$

where I_1 is the set of candidate interactions I when $L = 1$. Note that

$$\mathbb{P}_\xi((j^*, k^*) \in I) = 1 - \{1 - \beta(\xi)\}^{\xi_L}.$$

Thus any $\xi \in \Xi_{\text{all}}(\eta)$ with $T(\xi)$ minimal must have ξ_L as the smallest L such that $1 - \{1 - \beta(\xi)\}^{\xi_L} \geq \eta$, whence

$$\xi_L = \lceil \log(1 - \eta) / \log\{1 - \beta(\xi)\} \rceil. \tag{22}$$

Note that $\beta(\xi)$ does not depend on ξ_L, so the above equation completely determines the optimal choice of L once other parameters have been fixed. We will therefore henceforth assume that L has been chosen this way so that the discovery probability of all the algorithms is at least η.

The proofs of (8) and (9) are contained in Lemmas 12 and 13 respectively. The proof of (7) is more involved and proceeds by establishing a Neyman–Pearson type lemma (Lemmas 10 and 11) showing that given a constraint on the 'size' α that is sufficiently small, minimal subsampling enjoys maximal 'power' β. To complete the argument, we show that any sequence of algorithms with size α remaining constant as $p \to \infty$ cannot have a subquadratic complexity, whilst Lemma 12 attests that in contrast minimal subsampling does have subquadratic complexity under the assumptions of the theorem. Several auxiliary technical lemmas are collected in Section 6

Our proofs Lemmas 10 and 11 make use of the following bound on a quantity related to the ratio of the size to the power of minimal subsampling.

Lemma 9 *Suppose $\psi \in \Psi$ has distribution for M placing mass on M and $M + 1$. Under the assumptions of Theorem 1,*

$$\frac{\alpha(\psi)}{\gamma_1^M} \leq \frac{2}{1 - \rho} \frac{1}{M + 1}.$$

Proof We have

$$\frac{\alpha(\psi)}{\gamma_1^M} \leq \frac{1}{p^2} \sum_{j,k} (\gamma_{jk}/\gamma_1)^M \leq \frac{c_0}{p} + \sum_{i=0}^{n\gamma_1 - 1} \left(\frac{i}{n\gamma_1}\right)^M f_n(i/n).$$

Now the sum on the RHS is maximised over f_n obeying constraints (A1) and (A2) in the following way. If $\rho\gamma_1 n > \gamma_1 n - 1$ then f_n places all available mass on $\gamma_1 - 1/n$. Otherwise f_n should be as close to constant as possible on $\lceil \rho\gamma_1 n \rceil / n, \ldots, (\gamma_1 n - 1)/n$, and zero below $\lceil \rho\gamma_1 n \rceil / n$. In both cases it can be seen that

$$\sum_{i=0}^{n\gamma_1 - 1} \left(\frac{i}{n\gamma_1}\right)^M f_n(i/n) \leq \frac{2}{1 - \rho} \int_{(1+\rho)/2}^1 x^M dx \leq \frac{2}{1 - \rho} \frac{1}{M + 1}.$$

■

The following Neyman–Pearson-type lemma considers only non-randomised algorithms in Ξ. In Lemma 11 we extend this result to randomised algorithms.

Lemma 10 *Let Ξ_0 be the set of $\xi \in \Xi$ such that ξ_M places mass only on a single M, so the subsample size is not randomised. There exists an α_0 independent of n such that for all $\alpha' \leq \alpha_0$, we have*

$$\sup_{\psi \in \Psi : \alpha(\psi) \leq \alpha'} \beta(\psi) = \sup_{\xi \in \Xi_0 : \alpha(\xi) \leq \alpha'} \beta(\xi).$$

Moreover the suprema are achieved.

Proof Each $\xi \in \Xi_0$ is parametrised by its close pairs threshold τ and subsample size M. Given a $\xi \in \Xi_0$ with parameter values τ and M we compute $\alpha(\xi)$ as follows. Note that by replacing the threshold τ by $\tau/2$, we may assume that $\mathbf{X}$ and $\mathbf{Z}$ have entries in $\{-1/2, 1/2\}$. Thus $\mathbf{X}_j - \mathbf{Z}_k$ has components in $\{-1, 0, 1\}$. Let J_{jk} be the number of non-zero components of $(X_{i_m j} - Z_{i_m k})_{m=1}^M$. Then $J_{jk} \sim \mathrm{Binom}(M, 1 - \gamma_{jk})$. Thus

$$\mathbb{P}\Big(\Big| \sum_{m=1}^M D_m (X_{i_m j} - Z_{i_m k}) \Big| \leq \tau \Big) = \mathbb{P}(J_{jk} = 0) + \sum_{r=1}^M \mathbb{P}\Big(\Big| \sum_{m=1}^r D_m \Big| \leq \tau \Big) \mathbb{P}(J_{jk} = r),$$

noting that $D_m \stackrel{d}{=} -D_m$. By Lemma 14 we know there exists an $a > 0$ such that for all $\tau \leq a\sqrt{M}$ the RHS is bounded below by

$$\gamma_{jk}^M + \sum_{r=r_0}^M \frac{c_1 \tau}{\sqrt{r}} \binom{M}{r} \gamma_{jk}^{M-r} (1 - \gamma_{jk})^r \tag{23}$$

for M sufficiently large. Here the constants $a, c_1 > 0$ and $r_0 \in \mathbb{N}$ depend only on F.

Consider $\tau > a\sqrt{M}$. In this case, for $r \leq M$ sufficiently large we have by Lemma 14

$$\mathbb{P}\Big(\Big| \sum_{m=1}^r D_m \Big| \leq \tau \Big) \geq \mathbb{P}\Big(\Big| \sum_{m=1}^r D_m \Big| \leq a\sqrt{r} \Big) \geq c_1 a.$$

However then for M sufficiently large,

$$\mathbb{P}(J_{jk} = 0) + \sum_{r=1}^M \mathbb{P}\Big(\Big| \sum_{m=1}^r D_m \Big| \leq \tau \Big) \mathbb{P}(J_{jk} = r) \geq c_1 a/2,$$

so $\alpha(\xi) \geq c_1 a/2$. Note also that we must have $\alpha_0 \geq \alpha(\xi) \geq \gamma_l^M$, so $M \geq \log(\alpha_0)/\log(\gamma_l)$. Thus by choosing $0 < \alpha_0 < c_1 a/2$ sufficiently small, we can rule out $\tau > a\sqrt{M}$ and so we henceforth assume that $\tau \leq a\sqrt{M}$, and that M is sufficiently large such that (23) holds for all (j, k).

We have

$$\alpha(\xi) \geq \frac{1}{p^2} \sum_{j,k} \Big\{ \gamma_{jk}^M + \tau \sum_{r=r_0}^M \frac{c_1}{\sqrt{r}} \binom{M}{r} \gamma_{jk}^{M-r} (1 - \gamma_{jk})^r \Big\}. \tag{24}$$

Similarly we have

$$\beta(\xi) \leq \gamma_1^M + \tau \sum_{r=1}^M \frac{c_2}{\sqrt{r}} \binom{M}{r} \gamma_1^{M-r} (1 - \gamma_1)^r. \tag{25}$$

Now substituting the upper bound on τ implied by (24) into (25), we get

$$\beta(\xi) \leq \gamma_1^M + Q_M \Big(\alpha(\xi) - \frac{1}{p^2} \sum_{j,k} \gamma_{jk}^M \Big)$$

where

$$Q_M = \frac{c_2 \sum_{r=1}^M r^{-1/2} \binom{M}{r} \gamma_1^{M-r} (1 - \gamma_1)^r}{c_1 p^{-2} \sum_{j,k} \sum_{r=r_0} r^{-1/2} \binom{M}{r} \gamma_{jk}^{M-r} (1 - \gamma_{jk})^r}.$$

Now by Lemma 15, for M sufficiently large and some constant Q we have

$$Q_M \leq Q \frac{\sqrt{1 - \gamma_1}}{\sum_{j,k} \sqrt{1 - \gamma_{jk}}/p^2} \leq Q.$$

Thus

$$\beta(\xi) \leq \gamma_1^M + Q \Big(\alpha(\xi) - \frac{1}{p^2} \sum_{j,k} \gamma_{jk}^M \Big) \tag{26}$$

for all M sufficiently large. Now given α_0, let M_0 be such that

$$\frac{1}{p^2} \sum_{j,k} \gamma_{jk}^{M_0} \geq \alpha_0 \geq \frac{1}{p^2} \sum_{j,k} \gamma_{jk}^{M_0+1}.$$

Consider the minimal subsampling algorithm ψ that chooses subsample size as either M_0 or $M_0 + 1$ with probabilities b and $1 - b$ such that

$$\alpha(\psi) = \frac{1}{p^2} \sum_{j,k} \{ b\gamma_{jk}^{M_0} + (1 - b)\gamma_{jk}^{M_0+1} \} = \alpha_0.$$

Then we have $\beta(\psi) = b\gamma_1^{M_0} + (1 - b)\gamma_1^{M_0+1}$. Now suppose $\xi \in \Xi_0$ has $\alpha(\xi) \leq \alpha_0$. Then in particular $M \geq M_0 + 1$. We first examine the case where $M = M_0 + 1$. Then

$$\begin{aligned}
\frac{1}{\gamma_1^{M_0}} \{ \beta(\psi) - \beta(\xi) \} &\geq b + (1 - b)\gamma_1 - \gamma_1 - \frac{Q}{\gamma_1^{M_0}} \Big(\alpha_0 - \frac{1}{p^2} \sum_{j,k} \gamma_{j,k}^{M_0+1} \Big) \\
&= b + (1 - b)\gamma_1 - \gamma_1 - \frac{aQ}{\gamma_1^{M_0}} \frac{1}{p^2} \sum_{j,k} (\gamma_{j,k}^{M_0} - \gamma_{j,k}^{M_0+1}) \\
&\geq b \Big((1 - \gamma_u) - \frac{2Q}{1 - \rho} \frac{1}{M_0 + 1} \Big),
\end{aligned}$$

using Lemma 9 in the final line. Note this is non-negative for M_0 sufficiently large. When $M \geq M_0 + 2$ we instead have

$$\frac{\beta(\xi)}{\beta(\psi)} \leq \frac{\beta(\xi)}{\gamma_1^{M_0+1}} \leq \gamma_1 + \frac{2Q}{\gamma_1(1 - \rho)} \frac{1}{M_0 + 1} \leq \gamma_u + \frac{2Q}{\gamma_l(1 - \rho)} \frac{1}{M_0 + 1} < 1$$

for M_0 sufficiently large. Recall that by making α_0 sufficiently small, we can force M_0 to be arbitrarily large. Thus the result is proved. ■

Lemma 11 *There exists an α_0 independent of n such that for all $\alpha' \leq \alpha_0$, we have*

$$\sup_{\psi\in\Psi:\alpha(\psi)\leq\alpha'} \beta(\psi) = \sup_{\xi\in\Xi:\alpha(\xi)\leq\alpha'} \beta(\xi).$$

Moreover the suprema are achieved.

Proof With a slight abuse of notation, write $\xi(M', \tau')$ for the element of $\xi \in \Xi$ that fixes $M = M'$ and $\tau = \tau'$. Using the notation of Lemma 10, define function $f : [0, 1] \to [0, 1]$ by

$$f(\alpha') = \sup_{\xi\in\Xi_0:\,\alpha(\xi)\leq\alpha'} \beta(\xi).$$

Note that for $\xi \in \Xi$ we have

$$\beta(\xi) \leq \mathbb{E}_{M\sim\xi_M} f[\alpha\{\xi(M, \xi_\tau)\}]. \tag{27}$$

Now by Lemma 10 we know there exists α_0 (depending on F) such that on $[0, \alpha_0]$, f is the linear interpolation of points

$$\left(\frac{1}{p^2}\sum_{j,k} \gamma_{j,k}^M,\, \gamma_1^M\right)_{M=1}^{\infty}.$$

We claim that f is concave on $[0, \alpha_0]$. Indeed, it suffices to show that the slopes of the successive linear interpolants are decreasing in this region, or equivalently that their reciprocals are increasing. We have

$$\frac{1}{p^2}\sum_{j,k} \frac{\gamma_{jk}^{M+1} - \gamma_{jk}^M}{\gamma_1^{M+1} - \gamma_1^M} = \frac{1}{p^2}\sum_{j,k} \left(\frac{\gamma_{j,k}}{\gamma_1}\right)^M \frac{\gamma_{jk} - 1}{\gamma_1 - 1} \tag{28}$$

which increases as M decreases, thus proving the claim.

Note also that the RHS of (28) is at most $\alpha(\psi)/\{(1 - \gamma_u)\gamma_1^M\}$ when ψ has subsample size fixed at M. Thus by Lemma 9 we see the derivatives of the linear interpolants approach infinity as they get closer to the origin. This implies the existence of an $0 < \alpha_1 < \alpha_0$ such that $-\sup\big(\partial(-f)(\alpha_1)\big) \geq \{1 - f(\alpha_1)\}/(\alpha_0 - \alpha_1)$, where $\partial(-f)(\alpha_1)$ denotes the subdifferential of the function $-f$ at α_1. We may therefore invoke Lemma 16 to conclude that for ξ with $\alpha(\xi) \leq \alpha_1$

$$\mathbb{E}_{M\sim\xi_M} f[\alpha\{\xi(M, \xi_\tau)\}] \leq f[\mathbb{E}_{M\sim\xi_M}\alpha\{\xi(M, \xi_\tau)\}] = f(\alpha(\xi)) \leq f(\alpha_1) = \max_{\psi\in\Psi:\alpha(\psi)\leq\alpha_1} \beta(\psi).$$

Combining with (27) gives the result. ■

The next lemma establishes subquadratic complexity of minimal subsampling.

Lemma 12 *Under the assumptions of Theorem 1, we have* $\inf_{\psi\in\Psi(\eta)} T(\psi)/(np^2) \to 0$.

Proof Let $\psi \in \Psi$ be such that ψ_M places all mass on M. We have that $\beta(\psi) = \gamma_1^M$. Thus using the inequality $-x \leq \log(1 - x)$ for $x \in (0, 1)$, we have

$$\psi_L \leq -\gamma_1^{-M}\log(1 - \eta).$$

Lemma 9 gives an upper bound on $\psi_L\mathbb{E}_\psi E_1$. Note that $\mathbb{E}_\psi V = \mathcal{O}(p\log(p))$. Thus ignoring constant factors, we have

$$T(\psi)/(np^2) \leq \frac{M + \log(p)}{\gamma_1^M np} + \frac{1}{M + 1}.$$

Taking $M = \big\lfloor \log(1/\sqrt{p})/\log(\gamma_1)\big\rfloor$ then ensures $T(\psi)/(np^2) \to 0$. ■

Lemma 13 *Let $\xi \in \Xi_{dense}$. There exists $c > 0$ and $n_0 \in \mathbb{N}$ such that for all $n \geq n_0$,*

$$\inf_{\xi\in\Xi_{dense}} T(\xi)/(np^2) > c.$$

Proof Each $\xi \in \Xi_{\text{dense}}$ is parametrised by its close pairs threshold τ. Given a $\xi \in \Xi_{\text{dense}}(F)$ with close pairs threshold τ we compute $\alpha(\xi)$ as follows. Similarly to Lemma 10 we may assume without loss of generality that $\mathbf{X}$ and $\mathbf{Z}$ have entries in $\{-1/2, 1/2\}$ so $\mathbf{X}_j - \mathbf{Z}_k$ has components in $\{-1, 0, 1\}$. Since $R_i \overset{d}{=} -R_i$ as $F \in \mathcal{F}$, we have

$$\mathbb{P}\bigg(\bigg|\sum_{i=1}^n R_i(X_{ij} - Z_{ik})\bigg| \leq \tau\bigg) = \mathbb{P}\bigg(\bigg|\sum_{i=1}^{n(1-\gamma_{jk})} R_i\bigg| \leq \tau\bigg).$$

We now use Lemma 14. For $n(1-\gamma_u)$ sufficiently large, when $\tau \leq a\sqrt{n}$ the RHS is bounded below by

$$\frac{c_1\tau}{\sqrt{n(1 - \gamma_{jk})}}.$$

Here constant $a, c_1 > 0$ also depend only on F. Thus

$$\alpha(\xi) \geq \frac{1}{p^2}\sum_{j,k} \frac{c_1\tau}{\sqrt{n(1 - \gamma_{jk})}} \geq c_1\tau/\sqrt{n}. \tag{29}$$

Similarly we have

$$\beta(\xi) \leq \frac{c_2\tau}{\sqrt{n(1 - \gamma_1)}}. \tag{30}$$

Note that from (29), when $\tau > a\sqrt{n}$ we have $\alpha(\xi) \geq c_1 a$. Thus from (21) we know there exists n_0 such that for all $n \geq n_0$, we have

$$\inf_{\xi\in\Xi_{\text{dense}}(\eta):\xi_\tau>a\sqrt{n}} T(\xi)/(np^2) \geq \inf_{\xi\in\Xi_{\text{dense}}(\eta):\xi_\tau>a\sqrt{n}} \xi_L\alpha(\xi) \geq \xi_L c_1 a > 0. \tag{31}$$

We therefore need only consider the case where $\tau \leq a\sqrt{n}$ and where $\alpha(\xi) \to 0$.

Substituting the upper bound on τ implied by (29) into (30), we get

$$\beta(\xi) \leq \alpha(\xi)\frac{c_2}{c_1\sqrt{1 - \gamma_u}}.$$

Note that then

$$\xi_L \geq \frac{\log(1-\eta)}{\log\{1-\alpha(\xi)c_2/(c_1\sqrt{1-\gamma_u})\}} \geq c_3\frac{\log\big(1/1-\eta\big)}{\alpha(\xi)}$$

for some $c_3 > 0$ provided $\alpha(\xi) < 1/2$ say. However this gives us

$$\inf_{\xi\in\Xi_{\text{dense}}(\eta):\xi_\tau\leq a\sqrt{n}} T(\xi)/(np^2) \geq \inf_{\xi\in\Xi_{\text{dense}}(\eta):\xi_\tau\leq a\sqrt{n}} \xi_L\alpha(\xi) \geq \min\{1/2, c_3\log\big(1/1-\eta\big)\} > 0.$$

Combined with (31) this give the result. ■

With the previous lemmas in place, we are in a position to prove (7) of Theorem 1.

Proof of Theorem 1

The proofs of (8) and (9) are contained in Lemmas 12 and 13 respectively. To show (7) we argue as follows. Given F and η, suppose for contradiction that there exists a sequence $\xi^{(1)}, \xi^{(2)}, \ldots$ and $n_1 < n_2 < \cdots$ such that (making the dependence on n of the computational time explicit)

$$\inf_{\psi\in\Psi(\eta)} T^{(n_k)}(\psi) > T^{(n_k)}(\xi^{(k)})$$

for all k. By Lemma 12, we must have $T^{(n_k)}(\xi^{(k)})/(np^2) \to 0$. This implies that $\alpha(\xi^{(k)}) \to 0$. By Lemma 11, we know that for k sufficiently large

$$\sup_{\psi\in\Psi:\alpha(\psi)=\alpha(\xi^{(k)})} \beta(\psi) \geq \beta(\xi^{(k)}).$$

Let $\psi^{(k)}$ be the maximiser of the LHS. In order for $T^{(n_k)}(\psi^{(k)}) > T^{(n_k)}(\xi^{(k)})$, it must be the case that $\mathbb{E}_{M\sim\psi^{(k)}_M} M > \mathbb{E}_{M\sim\xi^{(k)}_M} M$. However we claim that $\xi = \psi^{(k)}$ minimises $\mathbb{E}_{M\sim\xi_M} M$ among all $\xi \in \Xi$ with $\alpha(\xi) \leq \alpha(\xi^{(k)}) =: \alpha_0$, which gives a contradiction and completes the proof. Let f be the function that linearly interpolates the points

$$\Big(\frac{1}{p^2}\sum_{j,k}\gamma_{j,k}^M,\, M\Big)_{M=1}^{\infty}.$$

Note that f is decreasing. By considering the inverse of f it is clear that f is convex. With a slight abuse of notation, write $\xi(M,\tau)$ for the element of $\xi \in \Xi$ such that ξ_M places all mass on M and $\xi_\tau = \tau$. Note that

$$\mathbb{E}_{M\sim\xi_M} M = \mathbb{E}_{M\sim\xi_M} f[\alpha\{\xi(M,0)\}] \geq \mathbb{E}_{M\sim\xi_M} f[\alpha\{\xi(M,\xi_\tau)\}].$$

Now suppose ξ has $\alpha(\xi) \leq \alpha_0$. Then from the above and Jensen's inequality,

$$\mathbb{E}_{M\sim\xi_M} M \geq f\big(\mathbb{E}_{M\sim\xi_M}\alpha(\xi(M,\xi_\tau))\big) \geq f(\alpha_0) = \mathbb{E}_{M\sim\psi^{(k)}_M} M. ■$$

Proof of Theorem 2

First note that from (11) we have $L \leq \log(1-\eta')/\log(1-\gamma^M)+1$. Then using the inequality $\log(1-x) \leq -x$ for $x \in (0,1)$, we have

$$L \leq \frac{\log\{1/(1-\eta')\}+1}{\gamma^M}.$$

Note that from the definition of γ_0 we have $\gamma^{-M} = p^{\log(\gamma)/\log(\gamma_0)}$. We then see that

$$\begin{aligned}
\gamma^{-M}\mathbb{E}(E_1) &= \gamma^{-M}\sum_{j,k}\gamma_{jk}^M \\
&\leq \gamma^{-M}\Big(\sum_{j,k:\gamma_{jk}>\gamma}\gamma_{jk}^M + \sum_{j,k:\gamma_0<\gamma_{jk}\leq\gamma}\gamma_{jk}^M + \sum_{j,k:\gamma_{jk}\leq\gamma_0}\gamma_{jk}^M\Big) \\
&\leq c_1 p\gamma^{-M} + c_2 p^{1+\log(\gamma)/\log(\gamma_0)} + p^2\gamma_0^M\gamma^{-M} \\
&\leq (c_1+c_2+1)p^{1+\log(\gamma)/\log(\gamma_0)}.
\end{aligned}$$

Collecting together the terms in (10) we have

$$C(M,L) \leq np + [\log\{1/(1-\eta')\}+1][\log(p)\{1+1/\log(\gamma_0^{-1})\} + n(c_1+c_2+1)]p^{1+\log(\gamma)/\log(\gamma_0)}$$

from which the result easily follows.

Proof of Proposition 3

Let $\eta^* = \eta(M^*, L)$. Note that in order for $\eta(M', L') \geq \eta^*$ it must be the case that $L' \geq \log(1-\eta^*)/\log(1-\gamma^{M'})$. Therefore

$$\begin{aligned}
C(M',L') - np &\geq \frac{\log(1-\eta^*)}{\log(1-\gamma^{M'})}\Big(M'p + p\log(p) + n\sum_{j,k}\gamma_{jk}^{M'}\Big) \\
&\geq \min_{M\in\mathbb{N}}\frac{\log(1-\eta^*)}{\log(1-\gamma^M)}\Big(Mp + p\log(p) + n\sum_{j,k}\gamma_{jk}^M\Big) \qquad (32) \\
&= \frac{\log(1-\eta^*)}{\log(1-\gamma^{M^*})}\Big(M^*p + p\log(p) + n\sum_{j,k}\gamma_{jk}^{M^*}\Big) = C(M^*,L).
\end{aligned}$$

Moreover, the inequality leading to (32) is strict if M^* is the unique minimiser and $M' \neq M^*$.

Technical lemmas

Lemma 14 *Let $F \in \mathcal{F}$ and suppose $(R_i)_{i=1}^\infty$ is an i.i.d. sequence with $R_i \sim F$.*

Then for all $a > 0$, there exists $c_1, c_2 > 0$ and $l_0 \in \mathbb{N}$ such that for all $l \geq l_0$ and $0 \leq \tau \leq a\sqrt{l}$ we have

$$\frac{c_1\tau}{\sqrt{l}} \leq \mathbb{P}\Big(\Big|\sum_{i=1}^l R_i\Big| \leq \tau\Big) \leq \frac{c_2\tau}{\sqrt{l}}.$$

Proof Let f_l be the density of $\sum_{i=1}^l R_i/\sqrt{l}$. Note that as $\mathbb{E}(|R_1|^3) < \infty$, we must have $\mathbb{E}(R_1^2) < \infty$, so we may assume without loss of generality that $\mathbb{E}(R_1^2) = 1$. Then by Theorem 3 of Petrov (1964) we have that for sufficiently large l,

$$|f_l(t) - \phi(t)| \leq \frac{c}{\sqrt{l}(1+|t|^3)}. \tag{33}$$

Here c is a constant and $\phi(t) = e^{-t^2/2}/\sqrt{2\pi}$ is the standard normal density. Now by the mean value theorem, we have

$$2 \inf_{0\leq t\leq \tau/\sqrt{l}} \{f_l(t)\}\frac{\tau}{\sqrt{l}} \leq \mathbb{P}\Big(\big|\sum_{i=1}^l R_i\big|/\sqrt{l} \leq \tau/\sqrt{l}\Big) \leq 2 \sup_{0\leq t\leq \tau/\sqrt{l}} \{f_l(t)\}\frac{\tau}{\sqrt{l}}.$$

Thus from (33), for l sufficiently large we have

$$\mathbb{P}\Big(\big|\sum_{i=1}^l R_i\big| \leq \tau\Big) \geq \frac{\tau}{\sqrt{l}}\Big(\frac{\sqrt{2}}{\sqrt{\pi}}\exp\{-\tau^2/(2l)\} - \frac{2c}{\sqrt{l}}\Big).$$

Note that for $a > 0$ and l sufficiently large we have $\sqrt{2/\pi}e^{-a^2/2} > 2c/\sqrt{l}$, whence

$$\mathbb{P}\Big(\big|\sum_{i=1}^l R_i\big| \leq \tau\Big) \geq \frac{c_1\tau}{\sqrt{l}}$$

for $0 \leq \tau \leq a\sqrt{l}$, some $c_1 > 0$. A similar argument yields the upper bound in the final result. ■

Lemma 15 *Suppose $\gamma \in [0,1)$. For all $M \in \mathbb{N}$ we have*

$$\sum_{r=1}^M \frac{1}{\sqrt{r}}\binom{M}{r}(1-\gamma)^r\gamma^{M-r} \leq \frac{\sqrt{2}}{\sqrt{(1-\gamma)M}}. \tag{34}$$

Given $r_0 \in \mathbb{N}$ and $\gamma \in [0,1)$, there exists $c > 0$ and $M_0 \in \mathbb{N}$ such that for all $M \geq M_0$ we have

$$\sum_{r=r_0}^M \frac{1}{\sqrt{r}}\binom{M}{r}(1-\gamma)^r\gamma^{M-r} \geq \frac{c}{\sqrt{(1-\gamma)M}}. \tag{35}$$

Proof First we show the upper bound (34). Let $J \sim \text{Binomial}(M, 1-\gamma)$.

$$\begin{aligned}\sum_{r=1}^M \frac{1}{\sqrt{r}}\binom{M}{r}(1-\gamma)^r\gamma^{M-r} &\leq \sqrt{2}\sum_{r=1}^M \frac{1}{\sqrt{r+1}}\binom{M}{r}(1-\gamma)^r\gamma^{M-r} \\ &\leq \sqrt{2}\mathbb{E}(1/\sqrt{J+1}).\end{aligned}$$

Next, by Jensen's inequality we have $\mathbb{E}(1/\sqrt{J+1}) \leq \sqrt{\mathbb{E}\{1/(J+1)\}}$. We now compute $\mathbb{E}\{1/(J+1)\}$ as follows.

$$\begin{aligned}\mathbb{E}\Big(\frac{1}{J+1}\Big) &= \sum_{r=0}^M \frac{1}{r+1}\binom{M}{r}(1-\gamma)^r\gamma^{M-r} \\ &= \frac{1}{M+1}\sum_{r=0}^M \binom{M+1}{r+1}(1-\gamma)^r\gamma^{M-r} \\ &= \frac{1}{(1-\gamma)(M+1)}\sum_{r=0}^M \binom{M+1}{r+1}(1-\gamma)^{r+1}\gamma^{M-r} \\ &= \frac{1-\gamma^{M+1}}{(1-\gamma)(M+1)} \leq \frac{1}{(1-\gamma)(M+1)}.\end{aligned}$$

Putting things together gives (34).

Turning now to (35), we see that the LHS equals

$$\mathbb{E}(1/\sqrt{J}\mathbb{1}_{\{J\geq r_0\}}) = \mathbb{E}(1/\sqrt{J}|J \geq r_0)\mathbb{P}(J \geq r_0).$$

By Jensen's inequality we have

$$\mathbb{E}(1/\sqrt{J}|J \geq r_0) \geq \frac{1}{\sqrt{\mathbb{E}(J|J\geq r_0)}} = \frac{\sqrt{\mathbb{P}(J\geq r_0)}}{\sqrt{\mathbb{E}(J\mathbb{1}_{\{J\geq r_0\}})}} \geq \frac{\sqrt{\mathbb{P}(J\geq r_0)}}{\sqrt{(1-\gamma)M}}.$$

But as $M \to \infty$, $\mathbb{P}(J \geq r_0) \to 1$, which easily gives the result. ■

Lemma 16 *Let $f : [0,\infty) \to [0,1]$ be non-decreasing. Suppose there exists $0 < \alpha_1 < \alpha_0$ such that:*

(i) f is concave on $[0, \alpha_0]$;

(ii) $-\sup\big(\partial(-f)(\alpha_1)\big) \geq \{1 - f(\alpha_1)\}/(\alpha_0 - \alpha_1)$, where $\partial(-f)(\alpha_1)$ denotes the subdifferential of the function $-f$ at α_1.

Then if random variable X has $\mathbb{E}(X) \leq \alpha_0$, then $f(\mathbb{E}X) \geq \mathbb{E}f(X)$.

Proof Write $m = -\sup\big(\partial(-f)(\alpha_1)\big)$ Let function $g : [0,\infty) \to [0,\infty)$ be defined as follows.

$$g(x) = \begin{cases} f(x) & \text{if } 0 \leq x \leq \alpha_1 \\ f(\alpha_1) + m(x-\alpha_1) & \text{if } x > \alpha_1. \end{cases}$$

Note that g thus defined has $g(\alpha_0) \geq 1$. We see that g is convex and $g \geq f$. Thus if $\mathbb{E}(X) \leq \alpha_1$, by Jensen's inequality we have

$$f(\mathbb{E}X) = g(\mathbb{E}X) \geq \mathbb{E}g(X) \geq \mathbb{E}f(X).$$

■

Appendix B

Connection to LSH

Minimal subsampling as considered in Algorithm 2 is closely related to the locality-sensitive hashing (LSH) framework: Define $h(j) = \mathbf{R}^T\mathbf{X}_j$ ($\mathbf{R}$ corresponds to the minimal subsampling projection) to be the hashing function and $\mathcal{H}$ to be the family of such functions, from which we sample uniformly. Then $\mathcal{H}$ is $(\gamma, c\gamma, p_1, p_2)$-sensitive, that is:

- if $\gamma_{jk} \geq \gamma$ then $\mathbb{P}(h(j) = h(k)) \geq p_1$
- if $\gamma_{jk} \leq c\gamma$ then $\mathbb{P}(h(j) = h(k)) \leq p_2$,

where $0 < c < 1$. In the case of the minimal subsampling we have $p_1 = \gamma^M$ and $p_2 = \gamma^M c^M$. However, the typical LSH machinery cannot be applied directly to the equal pairs problem above. In our setting, we are not interested in preserving close pairs but rather the closest pairs. Theorem 1 establishes that the family $\mathcal{H}$ leads to the maximal ratio p_1/p_2 among all linear hashing families.

Appendix C

Proof of Proposition 4

Proof

$$\begin{aligned}\mathbb{P}(\mathrm{sgn}(Y_i) = \tilde{X}_{ij}\tilde{X}_{ik}) &= \frac{\mathrm{sgn}(Y_i)+1}{2}(g(X_{ij})g(X_{ik}) + (1-g(X_{ij}))(1-g(X_{ik})))\\ &+ \frac{1-\mathrm{sgn}(Y_i)}{2}(g(X_{ij})(1-g(X_{ik})) + (1-g(X_{ij}))g(X_{ik}))\\ &= \frac{1}{2} + \frac{\mathrm{sgn}(Y_i)}{2}(1-2g(X_{ij}))(1-2g(X_{ik})).\end{aligned}$$

■

Appendix D

The unbiased transform and the sign transform

Proposition 6
Proof The equation

$$\mathbb{E}[\tilde{X}_{ij}] = \mathbb{P}(\tilde{X}_{ij} = 1) - \mathbb{P}(\tilde{X}_{ij} = -1) = X_{ij},$$

implies

$$\mathbb{P}(\tilde{X}_{ij} = 1) = \frac{X_{ij}+1}{2}$$

This uniquely determines the unbiased transform. ■

Next we show two Lemmas that will be useful when proving Theorems 7 and 8.

Lemma 17 *Consider the setup of Theorem 7. Then there exists constants $C_1^\varepsilon, C_2^\varepsilon > 0$ such that defining*

$$\alpha_{n,p}^u = \alpha_{n,p}^u(t) = \left(1 + \frac{t + \log(nC_1^\varepsilon)}{C_2^\varepsilon}\right)\sqrt{2\{t + \log(4p)\}/n},$$

with probability at least $1 - 2\exp(-t)$ we have:

$$\begin{aligned}\frac{\sum_i Y_i X_{ij^*} X_{ik^*}}{\|\mathbf{Y}\|_1} &\notin \Big[-\frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}, \frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}\Big]\\ \frac{\sum_{i=1}^n Y_i X_{ij} X_{ik}}{\|\mathbf{Y}\|_1} &\in \Big[-\frac{m_2(1-r_u) + \alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}, \frac{m_2(1-r_u) + \alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}\Big] \ \forall (j,k) \neq (j^*, k^*).\end{aligned}$$

Proof First we consider a capped version of ε:

$$\varepsilon_i' = \begin{cases}\varepsilon_i \text{ if } |\varepsilon_i| \leq \sigma\\ \sigma\mathrm{sgn}(\varepsilon_i) \text{ otherwise,}\end{cases}$$

where σ is to be chosen later. We may apply Hoeffding's inequality to these bounded variables. We have to bound two terms:

$\frac{\sum_{i=1}^n Y_i X_{ij^*} X_{ik^*}}{\|\mathbf{Y}\|_1}$ from below and $\frac{\sum_{i=1}^n Y_i X_{ij} X_{ik}}{\|\mathbf{Y}\|_1}$ from above, for $(j,k) \neq (j^*, k^*)$.

Schematically the first term can be dealt with in the following way:

$$\mathbb{P}\Big(\frac{A+B}{C+D} \geq \frac{a+b}{c+d}\Big) \geq 1 - \mathbb{P}(A \leq a) - \mathbb{P}(B \leq b) - \mathbb{P}(C \geq c) - \mathbb{P}(D \geq d) \tag{36}$$

where

$$A + B = \sum_{i=1}^n (X_{ij^*}X_{ik^*})^2 + \varepsilon_i' X_{ij^*}X_{ik^*} \ \text{ and } \ C + D = \sum_{i=1}^n |X_{ij^*}X_{ik^*} + \varepsilon_i'|.$$

We deal with each term individually. Using Hoeffding's inequality we get:

$$A: \ \mathbb{P}\Big(\textstyle\sum_{i=1}^p (X_{ij^*}X_{ik^*})^2 \leq nm_2 - \delta\Big) \leq \exp(-\delta^2/2n))$$

$$B: \ \mathbb{P}\Big(\textstyle\sum_{i=1}^n \varepsilon_i' X_{ij^*}X_{ik^*} \leq -\kappa\Big) \leq \exp(-\kappa^2/2n\sigma^2)$$

$$C: \ \mathbb{P}\Big(\textstyle\sum_{i=1}^n |X_{ij^*}X_{ik^*}| \geq nm_1 + \delta\Big) \leq \exp(-\delta^2/2n)$$

$$D: \ \mathbb{P}\Big(\textstyle\sum_{i=1}^n |\varepsilon_i'| \geq nm_\varepsilon + \kappa\Big) \leq \exp(-2\kappa^2/n\sigma^2).$$

This gives us a bound of the interaction strength of the true interaction pair:

$$\begin{aligned}\mathbb{P}\Big(\frac{\sum_i Y_i X_{ij^*} X_{ik^*}}{\|\mathbf{Y}\|_1} &\geq \frac{nm_2 - \delta - \kappa}{nm_1 + nm_\varepsilon + \delta + \kappa}\Big)\\ &\geq 1 - \exp(-\delta^2/2n) - \exp(-\delta^2/2n)\\ &\quad - \exp(-\kappa^2/2n\sigma^2) - \exp(-\kappa^2/2n\sigma^2)\end{aligned}$$

Similarly we can treat the interaction strength of the non interacting pairs:

$A:$ Here we use assumption $(B1)$:

$$m_2(r_u - 1) \leq \mathbb{E}[X_{ij^*}X_{ik^*}X_{im}X_{io}] \leq m_2(1 - r_u).$$

Hence, $\mathbb{P}\Big(\sum_{i=1}^n X_{ij^*}X_{ik^*}X_{ij}X_{ik} \geq nm_2(1-r_u) + \delta\Big) \geq \exp(-\delta/2n)$.

For the rest we run the same bounds as before (using $|X_{ij^*}X_{ik^*} + \varepsilon_i'| \geq |X_{ij^*}X_{ik^*}| + \varepsilon_i'$). This yields the bound

$$\begin{aligned}\mathbb{P}\Big(\frac{\sum_{i=1}^n Y_i X_{ij}X_{ik}}{\|\mathbf{Y}\|_1} \leq \frac{nm_2(1-r_u)+\delta+\kappa}{nm_1 - \delta - \kappa}\Big) &\geq 1 - \exp(-\delta^2/2n) - \exp(-\delta^2/2n) \\ &\quad - \exp(-\kappa^2/2n\sigma^2) - \exp(-\kappa^2/2n\sigma^2)\end{aligned}$$

The above inequality needs to hold for all at most p^2 pairs that are not interactions, so that we effectively multiply the exponential terms with p^2. Another factor of 2 is multiplied in for the negative sign, as the fraction also has to be bounded away from -1. In total we thus have:

$$\begin{aligned}\frac{\sum_{i=1}^n Y_i X_{ij^*}X_{ik^*}}{\|\mathbf{Y}\|_1} &\notin \Big[-\frac{nm_2 - \delta - \kappa}{nm_1 + nm_\varepsilon + \delta + \kappa}, \frac{nm_2 - \delta - \kappa}{nm_1 + nm_\varepsilon + \delta + \kappa}\Big] \\ \frac{\sum_{i=1}^n Y_i X_{ij}X_{ik}}{\|\mathbf{Y}\|_1} &\in \Big[-\frac{nm_2(1-r_u)+\delta+\kappa}{nm_1 - \delta - \kappa}, \frac{nm_2(1-r_u)+\delta+\kappa}{nm_1 - \delta - \kappa}\Big] \ \forall (m,o) \neq (j,l)\end{aligned}$$

with probability at least $1 - \exp(-\delta^2/2n) - \exp(-\delta^2/2n) - \exp(-\kappa^2/2n\sigma^2) - \exp(-\kappa^2/2n\sigma^2)$.

Finally, let $\sigma \geq 1$, then we have to set δ and κ so that the probability is bigger than $1 - \exp(-t)$. This gives:

$$\exp(-t) = 4p\exp(-\delta^2/2n) \text{ and } \exp(-t) = 4p\exp(-\kappa^2/2n\sigma^2).$$

This gives

$$\delta = \sqrt{2n(t + \log(4p))} \text{ and } \kappa = \sqrt{2n\sigma^2(t + \log(4p))}.$$

Thus for $\alpha_{n,p}^u = \frac{\sqrt{2(t+\log(4p))(1+\sigma^2)}}{\sqrt{n}}$,

$$\begin{aligned}\frac{\sum_i Y_i X_{ij^*}X_{ik^*}}{\|\mathbf{Y}\|_1} &\notin \Big[-\frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}, \frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}\Big] \\ \frac{\sum_{i=1}^n Y_i X_{ij}X_{ik}}{\|\mathbf{Y}\|_1} &\in \Big[-\frac{m_2(1-r_u)+\alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}, \frac{m_2(1-r_u)+\alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}\Big] \ \forall (j,k) \neq (j^*,k^*)\end{aligned}$$

with probability at least $1 - \exp(-t)$.

Now we extend this result to the case of unbounded errors, that is we now assume that with high probability ε_i are bounded:

$$\mathbb{P}(\varepsilon_i = \varepsilon_i', \ \forall\, i) = 1 - \exp(-t).$$

Here we used the sub-exponential tail behavior of ε. We have $\mathbb{P}(|\varepsilon_i| \geq t) \leq C_1^\varepsilon \exp(-C_2^\varepsilon t)$. Hence we set

$$t = C_2^\varepsilon \sigma - \log(nC_1^\varepsilon) \ \Rightarrow \sigma = \frac{t + \log(nC_1^\varepsilon)}{C_2^\varepsilon}$$

Thus,

$$\alpha_{n,p}^u = \frac{\sqrt{2\{t + \log(4p)\}\{1 + (\frac{t+\log(nC_1^\varepsilon)}{C_2^\varepsilon})\}^2\}}}{\sqrt{n}}$$

with probability at least $1 - 2\exp(-t)$ we have:

$$\begin{aligned}\frac{\sum_i Y_i X_{ij^*}X_{ik^*}}{\|\mathbf{Y}\|_1} &\notin \Big[-\frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}, \frac{m_2 - \alpha_{n,p}^u}{m_1 + m_\varepsilon + \alpha_{n,p}^u}\Big] \\ \frac{\sum_{i=1}^n Y_i X_{ij}X_{ik}}{\|\mathbf{Y}\|_1} &\in \Big[-\frac{m_2(1-r_u)+\alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}, \frac{m_2(1-r_u)+\alpha_{n,p}^u}{m_1 - \alpha_{n,p}^u}\Big] \ \forall (j,k) \neq (j^*,k^*).\end{aligned}$$

■

Next we prove the equivalent result for the sign transform. The proof is very similar to the unbiased case:

Lemma 18 *Consider the setup of Theorem 8. Then there exists constants $C_1^X, C_2^X, C_1^\varepsilon, C_2^\varepsilon > 0$ such that defining*

$$\alpha_{n,p}^s = \alpha_{n,p}^s(t) = \frac{\sqrt{2(t + \log(4p))\Big(\Big(\frac{t+\log(pmC_1^X)}{C_2^X}\Big)^4 + \Big(\frac{t+\log(nC_1^\varepsilon)}{C_2^\varepsilon}\Big)^2\Big)}}{\sqrt{n}},$$

with probability at least $1 - 3\exp(-t)$ we have:

$$\begin{aligned}\frac{\sum_{i=1}^n Y_i \mathrm{sgn}(X_{ij^*}X_{ik^*})}{\|\mathbf{Y}\|_1} &\notin \Big[-\frac{m_1 - \alpha_{n,p}^s}{m_1 + m_\varepsilon + \alpha_{n,p}^s}, \frac{m_1 - \alpha_{n,p}^s}{m_1 + m_\varepsilon + \alpha_{n,p}^s}\Big] \\ \frac{\sum_{i=1}^n Y_i \mathrm{sgn}(X_{ij}X_{ik})}{\|\mathbf{Y}\|_1} &\in \Big[-\frac{m_1(1-r_s)+\alpha_{n,p}^s}{m_1 - \alpha_{n,p}^s}, \frac{m_1(1-r_s)+\alpha_{n,p}^s}{m_1 - \alpha_{n,p}^s}\Big] \ \forall\, (m,o) \neq (j^*,k^*).\end{aligned}$$

Proof First consider capped versions of the random variables of interest:

$$X_{ij}' = \begin{cases} X_{ij} \text{ if } |X_{ij}| \leq M \\ M\mathrm{sgn}(X_{ij}) \text{ otherwise}\end{cases} \quad \text{and} \quad \varepsilon_i' = \begin{cases} \varepsilon_i \text{ if } |\varepsilon_i| \leq \sigma \\ \sigma\mathrm{sgn}(\varepsilon_i) \text{ otherwise}\end{cases}$$

where M and σ are to be chosen later. Given these capped variables we can use Hoeffding's inequality as we now deal with bounded variables. We have to bound two terms:

$\frac{\sum_{i=1}^n Y_i \mathrm{sgn}(X_{ij^*}'X_{ik^*}')}{\|\mathbf{Y}\|_1}$ from below and $\frac{\sum_{i=1}^n Y_i \mathrm{sgn}(X_{ij}'X_{ik}')}{\|\mathbf{Y}\|_1}$ from above, for $(j,k) \neq (j^*,k^*)$

As in Lemma 17 equation (36):

$$A + B = \sum_{i=1}^n |X_{ij^*}'X_{ik^*}'| + \varepsilon_i' \mathrm{sgn}(X_{ij^*}'X_{ik^*}') \ \text{ and } \ C + D = \sum_{i=1}^n |X_{ij^*}'X_{ik^*}' + \varepsilon_i'|.$$

We deal with each term individually. Using Hoeffding's inequality we get:

$A:\ \mathbb{P}\Big(\sum_{i=1}^{p}|X'_{ij^*}X'_{ik^*}|\le nm_1-\delta\Big)\le\exp(-\delta^2/2nM^4))$

$B:\ \mathbb{P}\Big(\sum_{i=1}^{n}\varepsilon'_i\le-\kappa\Big)\le\exp(-\kappa^2/2n\sigma^2)$

$C:\ \mathbb{P}\Big(\sum_{i=1}^{n}|X'_{ij^*}X'_{ik^*}|\ge nm_1+\delta\Big)\le\exp(-\delta^2/2nM^4)$

$D:\ \mathbb{P}\Big(\sum_{i=1}^{n}|\varepsilon'|\ge nm'_\varepsilon+\kappa\Big)\le\exp(-2\kappa^2/n\sigma^2)$

This gives us a bound of the interaction strength of the true interaction pair:

$$\mathbb{P}\Big(\frac{\sum_i Y_i\mathrm{sgn}(X'_{ij^*}X'_{ik^*})}{\|\mathbf{Y}\|_1}\ge\frac{nm_1-\delta-\kappa}{nm_1+nm_\varepsilon+\delta+\kappa}\Big)\ge1-2\exp(-\delta^2/2nM^4)-2\exp(-\kappa^2/2n\sigma^2)$$

Similarly we can treat the interaction strength of the non interacting pairs:

$A:$ Here we use assumption $(C1)$. It implies

$$r_s/2\le\mathbb{P}(\mathrm{sgn}(X'_{ij^*}X'_{ik^*})=\mathrm{sgn}(X'_{ij}X'_{ik})|\mathbf{X})\le1-r_s/2.$$

This we use for computing the expectation:

$$\begin{aligned}\mathbb{E}[X'_{ij^*}X'_{ik^*}\mathrm{sgn}(X'_{ij}X'_{ik})]&=\mathbb{E}[\mathbb{E}[|X'_{ij^*}X'_{ik^*}|\mathrm{sgn}(X'_{ij}X'_{ik}X'_{ij^*}X'_{ik^*})]\\&=\mathbb{E}[\mathbb{E}[2|X'_{ij^*}X'_{ik^*}|\mathbf{1}_{\{\mathrm{sgn}(X'_{ij}X'_{ik}X'_{ij^*}X'_{ik^*})=1\}}|\mathbf{X}]]-\mathbb{E}[|X'_{ij^*}X'_{ik^*}|]\\&=\mathbb{E}[\mathbb{E}[2|X'_{ij^*}X'_{ik^*}||\mathbf{X}]]\mathbb{P}(\mathrm{sgn}(X'_{ij}X'_{ik}X'_{ij^*}X'_{ik^*})=1|\mathbf{X})-\mathbb{E}[|X'_{ij^*}X'_{ik^*}|]\\&=\mathbb{E}[|X'_{ij^*}X'_{ik^*}|](2\mathbb{P}(\mathrm{sgn}(X'_{ij}X'_{ik}X'_{ij^*}X'_{ik^*})=1|\mathbf{X})-1).\end{aligned}$$

Thus the expectation is given as:

$$m_1(r_s-1)\le E[X'_{ij^*}X'_{ik^*}\mathrm{sgn}(X'_{ij}X'_{ik})]\le m_1(1-r_s).$$

Hence, $\mathbb{P}\Big(\sum_{i=1}^{n}X'_{ij^*}X'_{ik^*}\mathrm{sgn}(X'_{ij}X'_{ik})\ge nm_1(1-r_s)+\delta\Big)\ge\exp(-2\delta/nM^4)$.

For the rest we use the same bounds as before (using $|X'_{ij^*}X'_{ik^*}+\varepsilon'_i|\ge|X'_{ij^*}X'_{ik^*}|+\varepsilon'_i$). This yields the bound

$$\mathbb{P}\Big(\frac{\sum_{i=1}^{n}Y_i\mathrm{sgn}(X'_{ij}X'_{ik})}{\|\mathbf{Y}\|_1}\le\frac{nm_1(1-r_s)+\delta+\kappa}{nm_1-\delta-\kappa}\Big)\ge1-\exp(-2\delta^2/nM^4)-\exp(-2\kappa^2/n\sigma^2).$$

The above inequality needs to hold for the at most p^2 pairs that are not interactions, so that we effectively multiply the exponential terms with p^2. Another factor of 2 is multiplied in for the negative sign, as the fraction also has to be bounded away from -1. In total we thus have:

$$\begin{aligned}&\frac{\sum_i Y_i\mathrm{sgn}(X'_{ij^*}X'_{ik^*})}{\|\mathbf{Y}\|_1}\notin\Big[-\frac{nm_1-\delta-\kappa}{nm_1+nm_\varepsilon+\delta+\kappa},\frac{nm_1-\delta-\kappa}{nm_1+nm_\varepsilon+\delta+\kappa}\Big]\\&\frac{\sum_{i=1}^{n}Y_i\mathrm{sgn}(X'_{ij}X'_{ik})}{\|\mathbf{Y}\|_1}\in\Big[-\frac{nm_1(1-r_s)+\delta+\kappa}{nm_1-\delta-\kappa},\frac{nm_1(1-r_s)+\delta+\kappa}{nm_1-\delta-\kappa}\Big]\ \forall(j,k)\neq(j^*,k^*)\end{aligned}$$

with probability at least $1-2p\exp(-\delta^2/2nM^4)-2p\exp(-\kappa^2/2n\sigma^2)$.

Finally we have to set δ and κ so that the probability is bigger than $1-\exp(-t)$. This gives:

$$\exp(-t)=4p\exp(-\delta^2/2nM^4)\ \text{ and }\ \exp(-t)=4p\exp(-\kappa^2/2n\sigma^2)$$

This gives

$$\delta=\sqrt{2nM^4(t+\log(4p))}\ \text{ and }\ \kappa=\sqrt{2n\sigma^2(t+\log(4p))}$$

Thus for $\alpha^s_{n,p}=\frac{\sqrt{2(t+\log(4p))(M^4+\sigma^2)}}{\sqrt{n}}$

$$\begin{aligned}&\frac{\sum_i Y_i\mathrm{sgn}(X'_{ij^*}X'_{ik^*})}{\|\mathbf{Y}\|_1}\notin\Big[-\frac{m_1-\alpha^s_{n,p}}{m_1+m_\varepsilon+\alpha^s_{n,p}},\frac{m_1-\alpha^s_{n,p}}{m_1+m_\varepsilon+\alpha^s_{n,p}}\Big]\\&\frac{\sum_{i=1}^{n}Y_i\mathrm{sgn}(X'_{ij}X'_{ik})}{\|\mathbf{Y}\|_1}\in\Big[-\frac{m_1(1-r_s)+\alpha^s_{n,p}}{m_1-\alpha^s_{n,p}},\frac{m_1(1-r_s)+\alpha^s_{n,p}}{m_1-\alpha^s_{n,p}}\Big]\ \forall(j,k)\neq(j^*,k^*)\end{aligned}$$

with probability at least $1-\exp(-t)$.

We now extend this result to the case of unbounded variables, that is we now assume that with high probability the variables X_{ij} and ε_i are bounded:

$$\mathbb{P}(\,X_{ij}=X'_{ij},\ \forall\,i,j)=1-\exp(-t)\ \text{ and }\ \mathbb{P}(\varepsilon_i=\varepsilon'_{ij},\ \forall\,i)=1-\exp(-t).$$

Here we used the sub-exponential tail behaviour of the X_{ij} and ε_i. There exists constants C_1^X, C_2^X such that $\mathbb{P}(|X_{ij}|\ge t)\le C_1^X\exp(-C_2^Xt)$ and similarly for ε. Hence we set

$$t=C_2^XM-\log(pnC_1^X)\ \Rightarrow M=\frac{t+\log(pnC_1^X)}{C_2^X}$$

$$t=C_2^\varepsilon\sigma-\log(nC_1^\varepsilon)\ \Rightarrow\sigma=\frac{t+\log(nC_1^\varepsilon)}{C_2^\varepsilon}$$

Thus we have

$$\alpha^s_{n,p}=\frac{\sqrt{2(t+\log(4p))\Big(\Big(\frac{t+\log(pnC_1^X)}{C_2^X}\Big)^4+\Big(\frac{t+\log(nC_1^\varepsilon)}{C_2^\varepsilon}\Big)^2\Big)}}{\sqrt{n}}$$

■

Next we prove **Theorem** 7:

Proof Given $\delta,\epsilon>0$, choose t such that $3\exp(-t)<\epsilon$. From (B3) we have that $\alpha^u_{n,p}(t)$ defined in Lemma 17 satisfies $\alpha^u_{n,p}(t)\to0$ as $n\to\infty$. Thus from Lemma 17 we know that there exists N such that for all $n\ge N$, with probability $1-\epsilon$ we have

$$\frac{\log(\gamma^g_{j^*k^*})}{\log(\gamma^g_{jk})}<\frac{\log\{(1+\frac{m_2}{m_1+m_\varepsilon})/2\}}{\log\{(1+\frac{m_1}{m_2(1-r_s)})/2\}}+\delta/2.$$

Thus for $n\ge N$, applying Corollary 5 we have that with probability $1-\epsilon$,

$$C(M,L)\le cnp^{1+\delta/2+\frac{\log(1/2+m_2/2((m_1+m_\varepsilon)))}{\log(1/2+m_2(1-r_u)/(2m_1))}},$$

for some constant c. ■

The proof of **Theorem** 8 is very similar and is thus omitted.

References

D. Achlioptas. Database-friendly random projections: Johnson–lindenstrauss with binary coins. *Journal of computer and System Sciences*, 2003.

P. Agarwal, H. Edelsbrunner, O. Schwarzkopf, and E. Welzl. Euclidean minimum spanning trees and bichromatic closest pairs. *Discrete & Computational Geometry*, 1991.

Y. Arkin, E. Rahmani, M. Kleber, R. Laaksonen, W. März, and E. Halperin. Epiq: efficient detection of snp–snp epistatic interactions for quantitative traits. *Bioinformatics*, 2014.

P. Bickel, Y. Ritov, and A. Tsybakov. Hierarchical selection of variables in sparse high-dimensional regression. *IMS Collections*, 2010.

J. Bien, J. Taylor, and R. Tibshirani. A lasso for hierarchical interactions. *The Annals of Statistics*, 2013.

L. Breiman. Random Forests. *Machine Learning*, 2001.

L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Wadsworth, Belmont, 1984.

P. Bühlmann, M. Kalisch, and L. Meier. High-dimensional statistics with a view towards applications in biology. *Annual Review of Statistics and Its Application*, 2014.

A. Davie and A. Stothers. Improved bound for complexity of matrix multiplication. *Proceedings of the Royal Society of Edinburgh: Section A Mathematics*, 2013.

B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least Angle Regression. *Annals of Statistics*, 2004.

J. Friedman. Multivariate adaptive regression splines. *Annals of Statistics*, 1991.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 2010.

N. Hao and H. Zhang. Interaction screening for ultrahigh-dimensional data. *Journal of the American Statistical Association*, 2014.

P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

P. Kemmeren and et al. Large-scale genetic perturbations reveal regulatory networks and an abundance of gene-specific repressors. *Cell*, 2014.

R. Knutti, D. Masson, and A. Gettelman. Climate model genealogy: Generation cmip5 and how we got there. *Geophysical Research Letters*, 2013.

Y. Kong, D. Li, Y. Fan, and J. Lv. Interaction Pursuit with Feature Screening and Selection. *arXiv preprint arXiv:1605.08933*, 2016.

F. Le Gall. Faster algorithms for rectangular matrix multiplication. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*. IEEE, 2012.

J. Leskovec, A. Rajaraman, and J. Ullman. *Mining of massive datasets*. Cambridge University Press, 2014.

R. Paturi, S. Rajasekaran, and J. Reif. The light bulb problem. *Proceedings of the second annual workshop on Computational learning theory*, 1989.

V. Petrov. On local limit theorems for sums of independent random variables. *Theory of Probability & Its Applications*, 1964.

R. Sedgewick. *Algorithms in C*. Addison-Wesley, 1998.

R.D. Shah. Modelling interactions in high-dimensional data with backtracking. *Journal of Machine Learning Research*, 2016.

R.D. Shah and N. Meinshausen. Random intersection trees. *The Journal of Machine Learning Research*, 2014.

M. Shamos and D. Hoey. Closest-point problems. In *Foundations of Computer Science, 1975., 16th Annual Symposium on*. IEEE, 1975.

V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 1969.

G. Thanei. xyz r package, 2016. URL `https://github.com/gathanei/xyz`.

R. Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society, Series B*, 1996.

V. Williams. Multiplying matrices faster than coppersmith-winograd. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*. ACM, 2012.

B. Winkelmann, W. März, B. Boehm, R. Zotz, J. Hager, P. Hellstern, and J. Senges. Rationale and design of the luric study-a resource for functional genomics, pharmacogenomics and long-term prognosis of cardiovascular disease. *Pharmacogenomics*, 2001.

J. Wu, B. Devlin, S. Ringquist, M. Trucco, and K. Roeder. Screen and clean: a tool for identifying interactions in genome-wide association studies. *Genetic Epidemiology*, 2010.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 2005.

Journal of Machine Learning Research 19 (2018) 1-47 Submitted 3/17; Revised 4/18; Published 8/18

Local Rademacher Complexity-based Learning Guarantees for Multi-Task Learning

Niloofar Yousefi NILOOFAR.YOUSEFI@UCF.EDU
Department of Electrical Engineering and Computer Science
University of Central Florida
Orlando, FL 32816, USA

Yunwen Lei LEIYW@SUSTC.EDU.CN
Department of Computer Science and Engineering
Southern University of Science and Technology
Shenzhen, 518055, China

Marius Kloft KLOFT@CS.UNI-KL.DE
Department of Computer Science
Technische Universität Kaiserslautern
67653 Kaiserslautern, Germany

Mansooreh Mollaghasemi MANSOOREH.MOLLAGHASEMI@UCF.EDU
Department of Industrial Engineering & Management Systems
University of Central Florida
Orlando, FL 32816, USA

Georgios C. Anagnostopoulos GEORGIO@FIT.EDU
Department of Electrical and Computer Engineering
Florida Institute of Technology
Melbourne, FL 32901, USA

Editor: Massimiliano Pontil

Abstract

We show a Talagrand-type concentration inequality for Multi-Task Learning (MTL), with which we establish sharp excess risk bounds for MTL in terms of the Local Rademacher Complexity (LRC). We also give a new bound on the LRC for any norm regularized hypothesis classes, which applies not only to MTL, but also to the standard Single-Task Learning (STL) setting. By combining both results, one can easily derive fast-rate bounds on the excess risk for many prominent MTL methods, including—as we demonstrate—Schatten norm, group norm, and graph regularized MTL. The derived bounds reflect a relationship akin to a conservation law of asymptotic convergence rates. When compared to the rates obtained via a traditional, global Rademacher analysis, this very relationship allows for trading off slower rates with respect to the number of tasks for faster rates with respect to the number of available samples per task.

Keywords: Excess Risk Bounds, Local Rademacher Complexity, Multi-task Learning

1. Introduction

A commonly occurring problem, when applying machine learning in the sciences, is the lack of a sufficient amount of training data to attain acceptable performance results; either obtaining such data may be very costly or they may be unavailable due to technological limitations. For example, in cancer genomics, tumor bioptic samples may be relatively scarce due to the limited number of cancer patients, when compared to samples of healthy individuals. Also, in neuroscience, electroencephalogram experiments are carried out on human subjects to record training data and typically involve only a few dozen subjects.

In such settings, when considering any type of prediction task per individual subject (for example, whether the subject is indeed suffering from a specific medical affliction or not), relying solely on the scarce data per individual most often leads to inadequate predictive performance. Such a direct approach completely ignores the advantages that might be gained, when considering intrinsic, strong similarities between subjects and, hence, tasks. For instance, in the area of genomics, different living organisms can be related to each other in terms of their evolutionary relationships as given by the tree of life. Taking into account such relationships may be instrumental in detecting genes of recently developed organisms, for which only a limited number of training data is available. While our discussion here has focused on the realm of biomedicine, similar limitations and opportunities to overcome them exist in other fields as well.

Transfer learning (Pan and Yang, 2010) and, in particular, Multi-Task Learning (MTL) (Caruana, 1997) leverage such underlying common links among a group of tasks, while respecting the tasks' individual idiosyncrasies to the extent warranted. This is achieved by phrasing the learning process as a joint, mutually dependent learning problem. An early example of such a learning paradigm is the neural network-based approach introduced by Caruana (1997), while more recent works consider convex MTL problems (Ando and Zhang, 2005; Evgeniou and Pontil, 2004; Argyriou et al., 2008a). At the core of each such MTL formulation lies a mechanism that encodes task relatedness into the learning problem (Evgeniou et al., 2005). Such relatedness mechanism can always be thought of as jointly constraining the tasks' hypothesis spaces, so that their geometry is mutually coupled, *e.g.*, via a block norm constraint (Yousefi et al., 2015). Thus, from a regularization perspective, the tasks mutually regularize their learning based on their inter-task relatedness. This process of information exchange during co-learning is often referred to as *information sharing*. With respect to learning theory results, the analysis of MTL goes back to the seminal work of Baxter (2000), which was followed up by the works of Ando and Zhang (2005); Maurer (2006a). Nowadays, MTL frameworks are routinely employed in a variety of settings. Some recent applications include computational genetics (Widmer et al., 2013), image segmentation (An et al., 2008), HIV therapy screening (Bickel et al., 2008), collaborative filtering (Cao et al., 2010), age estimation from facial images (Zhang and Yeung, 2010), and sub-cellular location prediction (Xu et al., 2011), just to name a few prominent ones.

MTL learning guarantees are centered around notions of (global) Rademacher complexities, which were introduced to machine learning by Bartlett et al. (2002); Bartlett and Mendelson (2002); Koltchinskii and Panchenko (2000); Koltchinskii (2001); Koltchinskii and Panchenko (2002), and employed in the context of MTL by Maurer (2006a,b); Kakade et al. (2012); Maurer and Pontil (2013); Maurer (2016); Maurer et al. (2016). All these works are briefly surveyed in Sect. 1.3. It is worth noting that, if T denotes the number of tasks being co-learned and n denotes the number of

available observations per task, then the fastest-converging error or excess risk bounds derived in these works are of the order $O(1/\sqrt{nT})$.

More recently, Koltchinskii (2006) and Bartlett et al. (2005) introduced a more nuanced variant of these complexities, termed Local Rademacher Complexity (LRC), as opposed to the original Global Rademacher Complexity (GRC). This new, modified function class complexity measure is attention-worthy, since, as shown by Bartlett et al. (2005), an LRC-based analysis is capable of producing more rapidly-converging excess risk bounds ("fast rates"), when compared to the ones obtained via a GRC analysis. This can be attributed to the fact that, unlike LRCs, GRCs ignore the fact that learning algorithms typically choose well-performing hypotheses that belong only to a subset of the entire hypothesis space under consideration. The end result of this distinction empowers a local analysis to provide less conservative and, hence, sharper bounds than the standard global analysis. To date, there have been only a few additional works attempting to reap the benefits of such local analysis in various contexts: active learning for binary classification tasks (Koltchinskii, 2010), multiple kernel learning (Kloft and Blanchard, 2011; Cortes et al., 2013), transductive learning (Tolstikhin et al., 2014), semi-supervised learning (Oneto et al., 2015) and bounds on the LRCs via covering numbers (Lei et al., 2015).

1.1 Our Contributions

Through a Talagrand-type concentration inequality adapted to the MTL case, this paper's main contribution is the derivation of sharp bounds on the MTL excess risk in terms of the distribution- and data-dependent LRC. For a given number of tasks T, these bounds admit faster (asymptotic) convergence characteristics in the number of observations per task n, when compared to corresponding bounds hinging on the GRC. Hence, these faster rates ensure us that the MTL hypothesis selected by a learning algorithm approaches the best-in-class solution as n increases beyond a certain threshold. We also prove a new bound on the LRC, which generally holds for hypothesis classes with any norm regularizers. This bound readily facilitates the bounding of the LRC for a range of such regularizers not only for MTL, but also for the standard Single-Task Learning (STL) setting. As a matter of fact, we demonstrate such results, in Sect. 4, for classes induced by graph-based, Schatten and group norm regularizers. Moreover, we prove matching lower bounds and, thus, show that, aside from constants, the LRC-based bounds are tight for the considered applications.

Our derived bounds reflect that one can trade off a slow convergence speed w.r.t. T for an improved convergence rate w.r.t. n. The latter one ranges from the typical GRC-based $O(1/\sqrt{n})$ bounds, all the way up to the fastest rate of order $O(1/n)$ by allowing the bound to depend less on T. Nevertheless, the premium in question becomes less relevant to MTL, since T is typically considered fixed is such setting.

Fixing all other parameters when the number of samples per task n approaches infinity, our local bounds yield faster rates compared to their global counterparts. Also, it is observed that, if the number of tasks T and the radius R of the ball-norms can grow with n, there are cases wherein local analysis always improves over the global one. When our local bounds are compared to the ones in(Maurer and Pontil, 2013; Maurer, 2006b), which stem from a global analysis, one observes that our bounds yield faster, $O(1/T)$ and $O(1/n)$ convergence rates.

1.2 Organization

The paper is organized as follows: Sect. 2 lays the foundations for our analysis by considering a Talagrand-type concentration inequality suitable for deriving our bounds. Next, in Sect. 3, after suitably defining LRCs for MTL hypothesis spaces, we provide our LRC-based MTL excess risk bounds. Based on these bounds, we follow up this section with a local analysis of linear MTL frameworks, in which task-relatedness is presumed and enforced by imposing a norm constraint. More specifically, leveraging off Hölder's inequality, Sect. 4 presents generic upper bounds for the relevant LRC of any norm regularized hypothesis class. These results are subsequently specialized to the case of group norm, Schatten norm and graph regularized linear MTL. Sect. 5 supplies the corresponding excess risk bounds based on the LRC of the aforementioned hypothesis classes. The paper concludes with Sect. 6, which investigates the convergence rate of our LRC-based excess risk bounds for the previously mentioned hypothesis spaces. We also compare our local bounds with those obtained from a GRC-based analysis provided in Maurer and Pontil (2013); Maurer (2006b).

1.3 Previous Related Works

An earlier work by Maurer (2006a), which considers linear MTL frameworks for binary classification, investigates the generalization guarantees based on Rademacher averages. In this framework, all tasks are pre-processed by a common bounded linear operator and operator norm constraints are used to control the complexity of the associated hypothesis spaces. The GRC-based error bounds derived are of order $O(1/\sqrt{nT})$. Another contemporary study (Maurer, 2006b) provides bounds for the empirical and expected Rademacher complexities of linear transformation classes. Based on Hölder's inequality, GRC-based risk bounds of order $O(1/\sqrt{nT})$ are established for MTL hypothesis spaces with graph-based and L_{S_q}-Schatten norm regularizers, where $q \in \{2\} \cup [4, \infty]$.

The subject of MTL generalization guarantees benefited from renewed attention in recent years. Kakade et al. (2012) take advantage of the strongly-convex nature of certain matrix-norm regularizers to easily obtain generalization bounds for a variety of machine learning problems. Part of their work is devoted to the realm of online and off-line MTL. In the latter case, which pertains to the focus of our work, the paper provides a GRC-based excess risk bound of order $O(1/\sqrt{nT})$. Moreover, Maurer and Pontil (2013) present a global Rademacher complexity analysis leading to excess risk bounds of order $O(\sqrt{\log(nT)/nT}$ for a trace norm regularized MTL model. Also, Maurer (2016) examines the bounding of (global) Gaussian complexities of function classes that result from considering composite maps, as is typical in several settings, including MTL. An application of the paper's results yields MTL risk bounds of order $O(1/\sqrt{nT})$. More recently, Maurer et al. (2016) presents excess risk bounds of order $O(1/\sqrt{nT})$ for both MTL and Learning-to-Learn (LTL) and reveals conditions, under which MTL is more beneficial over learning tasks independently.

Finally, although loosely related to our focus, we mention in passing a few works that pertain to generalization guarantees in the realm of life-long learning and domain adaptation. Generalization performance analysis in life-long learning has been investigated by Thrun and Pratt (2012); Ben-David and Schuller (2003); Ben-David and Borbely (2008); Pentina and Lampert (2015) and Pentina and Ben-David (2015). Also, in the context of domain adaptation, similar considerations are examined by Mansour et al. (2009a,b,c); Cortes and Mohri (2011); Zhang et al. (2012); Mansour and Schain (2013) and Cortes and Mohri (2014).

1.4 Basic Assumptions & Notations

Consider T supervised learning tasks sampled from the same input-output space $\mathcal{X} \times \mathcal{Y}$. Each task t is represented by an independent random variable (X_t, Y_t) governed by a probability distribution μ_t. Also, the *i.i.d.* samples related to each task t are described by the sequence $(X_t^i, Y_t^i)_{i=1}^n$, drawn from μ_t.

In what follows, we use the following notational conventions: vectors and matrices are depicted in bold face. The superscript T, when applied to a vector/matrix, denotes the transpose of that quantity. We define $\mathbb{N}_T := \{1, \ldots, T\}$. For any random variables X, Y and function f we use $\mathbb{E}f(X,Y)$ and $\mathbb{E}_X f(X,Y)$ to denote the expectation with w.r.t. all the involved random variables and the conditional expectation w.r.t. the random variable X respectively. For any vector-valued function $\boldsymbol{f} = (f_1, \ldots, f_T)$, we introduce the following two notations:

$$P\boldsymbol{f} := \frac{1}{T}\sum_{t=1}^{T} Pf_t, \qquad P_n\boldsymbol{f} := \frac{1}{T}\sum_{t=1}^{T} P_n f_t.$$

where $Pf_t := \mathbb{E}[f_t(X_t)]$ and $P_n f_t := \frac{1}{n}\sum_{i=1}^n f_t(X_t^i)$. When well-defined, we denote the component-wise exponentiation of a vector $\boldsymbol{f}$ as $\boldsymbol{f}^\alpha = (f_1^\alpha, \ldots, f_T^\alpha), \forall \alpha \in \mathbb{R}$. For any loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}_+$ and any $\boldsymbol{f} = (f_1, \ldots, f_T)$ we define $\ell_{\boldsymbol{f}} = (\ell_{f_1}, \ldots, \ell_{f_T})$ where ℓ_{f_t} is the function defined by $\ell_{f_t}((X_t, Y_t)) = \ell(f_t(X_t), Y_t)$.

Finally, in the subsequent material, we always assume the measurability of functions and suprema whenever necessary. Furthermore, operators on separable Hilbert spaces are assumed to be of trace class.

2. Talagrand-Type Inequality for Multi-Task Learning

Our derivation of LRC-based error bounds for MTL is founded on a Talagrand-type concentration inequality, which was adapted to the context of MTL and is presented next. It shows that the uniform deviation between the true and empirical means for a vector-valued function class $\mathcal{F}$ can be dominated by the associated *multi-task Rademacher complexity* plus a term involving the variance of functions in $\mathcal{F}$. A notable property of Theorem 1 is that the correlation among different components of $\boldsymbol{f}$, encoded by either the constraint on variances or the constraint imposed in the hypothesis space, is preserved. This last observation is congruent with the spirit of MTL. The proof of Theorem 1, which is deferred to Appendix A, is based on a so-called *Logarithmic Sobolev inequality* on log-moment generating functions.

Theorem 1 (TALAGRAND-TYPE INEQUALITY FOR MTL) *Let $\mathcal{F} = \{\boldsymbol{f} := (f_1, \ldots, f_T)\}$ be a class of vector-valued functions satisfying $\max_{t\in N_T} \sup_{x\in\mathcal{X}} |f_t(x)| \leq b$. Also, assume that $X := (X_t^i)_{(t,i)=(1,1)}^{(T,N_t)}$ is a vector of $\sum_{t=1}^T N_t$ independent random variables where $X_t^1, \ldots, X_t^n, \forall t$ are identically distributed. Let $\{\sigma_t^i\}_{t,i}$ be a sequence of independent Rademacher variables. If $\frac{1}{T}\sup_{\boldsymbol{f}\in\mathcal{F}} \sum_{t=1}^T \mathbb{E}\left[f_t(X_t^1)\right]^2 \leq r$, then, for every $x > 0$, with probability at least $1 - e^{-x}$,*

$$\sup_{\boldsymbol{f}\in\mathcal{F}}(P\boldsymbol{f} - P_n\boldsymbol{f}) \leq 4\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{8xr}{nT}} + \frac{12bx}{nT}, \tag{1}$$

where $n := \min_{t\in\mathbb{N}_T} N_t$, and the multi-task Rademacher complexity *of function class $\mathcal{F}$ is defined as*

$$\mathfrak{R}(\mathcal{F}) := \mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}=(f_1,\ldots,f_T)\in\mathcal{F}} \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\sigma_t^i f_t(X_t^i)\right\}.$$

Note that the same bound also holds for $\sup_{\boldsymbol{f}\in\mathcal{F}}(P_n\boldsymbol{f} - P\boldsymbol{f})$.

In Theorem 1, the data from different tasks are assumed to be mutually independent, which is typically presumed in MTL (Maurer, 2006a). To present the results in a clear way we always assume in the following that the available data for each task is the same, namely n.

Remark 2 *Note that Theorem 1 pertains to classes of uniformly bounded functions and is used in the sequel to bound the excess risk of multi-task learning function classes. However, using a new argument by Mendelson (2014), Theorem 1 can be extended beyond the case of classes of uniformly bounded loss functions. In particular, rather than adopting a concentration-based inequality, which is crucial to our approach here to bound the suprema of the resulting empirical processes, the approach in Mendelson (2014) relies on a "small ball" assumption. Such an assumption holds for functions with "well-behaved high-order moments" (e.g. heavy-tailed functions).*

Remark 3 *At this point, we present the result of the previous theorem for the special case of single task learning. It is very straightforward to verify that, for $T = 1$, the bound in* (1) *can be written as*

$$\sup_{f\in\mathcal{F}}(Pf - P_n f) \leq 4\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{8xr}{n}} + \frac{12bx}{n}, \tag{2}$$

where the function f is chosen from a scalar-valued function class $\mathcal{F}$. This bound can be compared to the result of Theorem 2.1 of Bartlett et al. (2005), which for $\alpha = 1$ reads as

$$\sup_{f\in\mathcal{F}}(Pf - P_n f) \leq 4\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{2xr}{n}} + \frac{8bx}{3n}. \tag{3}$$

Note that the difference between the constants in (2) *and* (3) *is due to the fact that we were unable to directly apply Bousquet's version of Talagrand's inequality (like it was done in Bartlett et al. (2005) for scalar-valued functions) to the class of vector-valued functions. To be more clear, let Z be defined as in* (A.2) *with the jackknife replication $Z_{s,j}$. We find a lower bound $Z''_{s,j}$ such that $Z''_{s,j} \leq Z - Z_{s,j}$. Then, in order to apply Theorem 2.5 of Bousquet (2002), one needs to show that the quantity $\frac{1}{nT}\sum_{s=1}^T\sum_{j=1}^n \mathbb{E}_{s,j}[(Z''_{s,j})^2]$ is bounded. This goal, ideally, can be achieved by including a constraint similar to $\frac{1}{T}\sup_{\boldsymbol{f}\in\mathcal{F}}\sum_{t=1}^T \mathbb{E}\left[f_t(X_t^1)\right]^2 \leq r$ in Theorem 1. However, we found that—when dealing with MTL class of functions—it is not very straightforward to define such a constraint that satisfies the boundedness condition $\frac{1}{nT}\sum_{s=1}^T\sum_{j=1}^n \mathbb{E}_{s,j}[(Z''_{s,j})^2]$ in terms of r. With that being said, the key ingredient to Theorem 1's proof is the so-called* Logarithmic Sobolev *inequality—Theorem A.1—which can be considered as the exponential version of* Efron-Stein*'s inequality.*

3. MTL Excess Risk Bounds based on Local Rademacher Complexities

At the heart of Theorem 1 lies a variance bound, which motivates us to consider Rademacher averages associated with a function sub-class enjoying small variances. As pointed out in Bartlett et al. (2005), these (local) averages are always smaller than the corresponding global Rademacher averages and allow for eventually deriving sharper generalization bounds. Herein, we exploit this very fact for MTL generalization guarantees.

Definition 4 (MULTI-TASK LOCAL RADEMACHER COMPLEXITY) *For a vector-valued function class $\mathcal{F} = \{\boldsymbol{f} = (f_1, \dots, f_T)\}$, the* Multi-Task Local Rademacher Complexity (MT-LRC) *$\mathfrak{R}(\mathcal{F}, r)$ is defined as*

$$\mathfrak{R}(\mathcal{F}, r) := \mathbb{E}_{X,\sigma}\Big[\sup_{\substack{\boldsymbol{f}=(f_1,\dots,f_T)\in\mathcal{F}\\ V(\boldsymbol{f})\le r}} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i f_t(X_t^i)\Big], \tag{4}$$

where $V(\boldsymbol{f})$ is an upper bound on the variance of the functions in $\mathcal{F}$.

For the case $T = 1$, it is clear that the MT-LRC reduces to the standard LRC for scalar-valued function classes. Analogous to single task learning, a challenge in using the MT-LRC to refine existing learning rates is to find an optimal radius trading-off the variance and the associated complexity, which, as we show later, reduces to the calculation of the fixed-point of a sub-root function.

Definition 5 (SUB-ROOT FUNCTION) *A function $\psi : [0, \infty] \to [0, \infty]$ is sub-root if and only if it is non-decreasing and the function $r \mapsto \psi(r)/\sqrt{r}$ is non-increasing for $r > 0$.*

Lemma 6 (Lemma 3.2 Bartlett et al. (2005)) *If ψ is a sub-root function, then it is continuous on $[0, \infty]$, and the equation $\psi(r) = r$ has a unique (non-zero) solution r^*, which is known as the fixed point of ψ. Moreover, for any $r > 0$, it holds that $r > \psi(r)$ if and only if $r^* \le r$.*

Intuitively, the model sought for by learning algorithms would hopefully attain a small generalization error and enjoy a small variance, when there is a relationship between risks and variances. The concept of local Rademacher complexity allows us to focus on identifying such models.

Definition 7 (VECTOR-VALUED BERNSTEIN CLASS) *Let $0 < \beta \le 1$ and $B > 0$. A vector-valued function class $\mathcal{F}$ is said to be a (β, B)-Bernstein class with respect to the probability measure P if there exists a function $V : \mathcal{F} \to \mathbb{R}^+$ such that*

$$P\boldsymbol{f}^2 \le V(\boldsymbol{f}) \le B(P\boldsymbol{f})^\beta, \quad \forall \boldsymbol{f} \in \mathcal{F}. \tag{5}$$

It can be shown that the Bernstein condition (5) is not too restrictive and it holds, for example, for non-negative bounded functions with respect to any probability distribution as shown in (Bartlett et al., 2004). Other examples include the class of excess risk functions $\mathcal{L}_\mathcal{F} := \{\ell_f - \ell_{f^*} : f \in \mathcal{F}\}$—with $f^* \in \mathcal{F}$ being the minimizer of $P\ell_f$— when the function class $\mathcal{F}$ is convex and the loss function ℓ is strictly convex.

In this section, we show that under some mild assumptions on a vector-valued Bernstein class, LRC-based excess risk bounds can be established for MTL. We will assume that the loss function ℓ and the vector-valued hypothesis space $\mathcal{F}$ satisfy the following conditions:

Assumption 8

1. *There is a function $\boldsymbol{f}^* = (f_1^*, \dots, f_T^*) \in \mathcal{F}$ satisfying $P\ell_{\boldsymbol{f}^*} = \inf_{\boldsymbol{f}\in\mathcal{F}} P\ell_{\boldsymbol{f}}$.*
2. *There is a constant $B' \ge 1$ and $0 < \beta \le 1$, such that for every $\boldsymbol{f} \in \mathcal{F}$ we have $P(\boldsymbol{f} - \boldsymbol{f}^*)^2 \le B'\big(P(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*})\big)^\beta$.*
3. *There is a constant L, such that the loss function ℓ is L-Lipschitz in its first argument.*

As pointed out in Bartlett et al. (2005), many regularized algorithms satisfy these conditions. More specifically, a uniform convexity condition on the loss function ℓ is usually sufficient to satisfy Assumption 8.2. A typical example is the quadratic loss function $\ell(f(X), Y) = (f(X) - Y)^2$. More specifically, if $|f(X) - Y| \le 1$ for any $f \in \mathcal{F}, x \in \mathcal{X}$ and $Y \in \mathcal{Y}$, then it can be shown that the conditions of Assumption 8 are met with $L = 1$ and $B = 1$.

We now present the main result of this section showing that the excess risk of MTL can be bounded by the fixed-point of a sub-root function dominating the MT-LRC. The proof of the results is provided in Appendix B.

Theorem 9 (Excess risk bound for MTL) *Let $\mathcal{F} := \{\boldsymbol{f} := (f_1, \dots, f_T)\}$ be a class of vector-valued functions $\boldsymbol{f}$ satisfying $\max_{t\in\mathbb{N}_T}\sup_{x\in\mathcal{X}}|f_t(x)| \le b$. Assume that $X := (X_t^i, Y_t^i)_{(t,i)=(1,1)}^{(T,n)}$ is a vector of nT independent random variables, where for each task t, the samples $(X_t^1, Y_t^1)\dots,(X_t^n, Y_t^n)$ are identically distributed. Suppose that Assumption 8 holds. Define $\mathcal{F}^* := \{\boldsymbol{f} - \boldsymbol{f}^*\}$, where $\boldsymbol{f}^*$ is the function satisfying $P\ell_{\boldsymbol{f}^*} = \inf_{\boldsymbol{f}\in\mathcal{F}} P\ell_{\boldsymbol{f}}$. Let $B := \max(B'L^2, 1)$ and ψ be a sub-root function with fixed point r^* such that $BL\mathfrak{R}(\mathcal{F}^*, r) \le \psi(r), \forall r \ge r^*$, where $\mathfrak{R}(\mathcal{F}^*, r)$ is the LRC of the functions class $\mathcal{F}^*$:*

$$\mathfrak{R}(\mathcal{F}^*, r) := \mathbb{E}_{X,\sigma}\Big[\sup_{\substack{\boldsymbol{f}\in\mathcal{F},\\ L^2P(\boldsymbol{f}-\boldsymbol{f}^*)^2\le r}} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i f_t(X_t^i)\Big]. \tag{6}$$

Then, for any $\boldsymbol{f} \in \mathcal{F}$, $K > 1$ and $x > 0$, with probability at least $1 - e^{-x}$,

$$P(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*}) \le \frac{K}{K-\beta}P_n(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*}) + (2K)^{\frac{\beta}{2-\beta}}20^{\frac{2}{2-\beta}}\max\Big((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\Big) + \Big(\frac{2^{\beta+3}B^2K^\beta x}{nT}\Big)^{\frac{1}{2-\beta}} + \frac{48LBbx}{(2-\beta)nT}. \tag{7}$$

The following corollary is direct by noting that $P_n(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \le 0$.

Corollary 10 *Let $\hat{\boldsymbol{f}}$ be any element of function class $\mathcal{F}$ satisfying $P_n\ell_{\hat{\boldsymbol{f}}} = \inf_{\boldsymbol{f}\in\mathcal{F}} P_n\ell_{\boldsymbol{f}}$. Assume that the conditions of Theorem 9 hold. Then for any $x > 0$ and $r > \psi(r)$, with probability at least $1 - e^{-x}$,*

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \le (2K)^{\frac{\beta}{2-\beta}}20^{\frac{2}{2-\beta}}\max\Big((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\Big) + \Big(\frac{2^{\beta+3}B^2K^\beta x}{nT}\Big)^{\frac{1}{2-\beta}} + \frac{48LBbx}{(2-\beta)nT}. \tag{8}$$

An immediate consequence of this section's results is that one can derive excess risk bounds for given regularized MTL hypothesis spaces. In the sequel, we will derive excess risk bounds for several commonly used norm regularized MTL hypothesis spaces by further bounding the fixed point r^* appearing in Corollary 10.

4. Local Rademacher Complexity Bounds for Norm Regularized MTL Models

This section presents very general MT-LRC bounds for hypothesis spaces defined by norm regularizers, which allows us to immediately derive, as specific application cases, LRC bounds for group norm, Schatten norm, and graph regularized MTL models.

4.1 Preliminaries

We consider linear MTL models, where we associate to each task a functional $f_t(X) := \langle \boldsymbol{w}_t, \phi(X)\rangle$. Here, $\boldsymbol{w}_t$ belongs to a *Reproducing Kernel Hilbert Space (RKHS)* $\mathcal{H}$, equipped with an inner product $\langle .,.\rangle$ and an induced norm $\|.\| := \sqrt{\langle .,.\rangle}$. Also, $\phi : \mathcal{X} \to \mathcal{H}$ is a feature map associated to $\mathcal{H}$'s reproducing kernel k satisfying $k(X, \tilde{X}) = \langle \phi(X), \phi(\tilde{X})\rangle, \forall X, \tilde{X} \in \mathcal{X}$. We assume that the multi-task model $\boldsymbol{W} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_T) \in \mathcal{H} \times \ldots \times \mathcal{H}$ is learned using the regularized cost function:

$$\min_{\boldsymbol{W}} \Omega\left(\boldsymbol{D}^{1/2}\boldsymbol{W}\right) + C\sum_{t=1}^{T}\sum_{i=1}^{n} \ell(\langle \boldsymbol{w}_t, \phi(X_t^i)\rangle, Y_t^i), \tag{9}$$

where the regularizer $\Omega(\cdot)$ may be used to reflect a priori information. This regularization scheme amounts to performing *Empirical Risk Minimization (ERM)* using the hypothesis space

$$\mathcal{F} := \left\{X \mapsto [\langle \boldsymbol{w}_1, \phi(X_1)\rangle, \ldots, \langle \boldsymbol{w}_T, \phi(X_T)\rangle]^T : \Omega(\boldsymbol{D}^{1/2}\boldsymbol{W}) \leq R^2\right\}, \tag{10}$$

where $\boldsymbol{D}$ is a given positive operator defined on $\mathcal{H}$. Note that the hypothesis spaces corresponding to the group and Schatten norms can be recovered by setting $\boldsymbol{D} = \boldsymbol{I}$ and by using their corresponding norms. More specifically, by choosing $\Omega(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|_{2,q}^2$, one obtains an $L_{2,q}$-group norm hypothesis space in (10). Similarly, the choice $\Omega(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|_{S_q}^2$ gives an L_{S_q}-Schatten norm hypothesis space in (10). Furthermore, the graph regularized MTL (Micchelli and Pontil, 2004; Evgeniou et al., 2005; Maurer, 2006b) can be obtained by taking $\Omega(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{D}^{1/2}\boldsymbol{W}\|_F^2$, where $\|.\|_F$ is a Frobenius norm, $\boldsymbol{D} := \boldsymbol{L} + \eta\boldsymbol{I}$, $\boldsymbol{L}$ is the relevant graph Laplacian, and $\eta > 0$ is a regularization constant. On balance, all these MTL models can be considered as norm regularized models. Note that in the sequel, we let q^* be the Hölder conjugate exponent of q, *i.e.* $1/q + 1/q^* = 1$.

4.2 General Bound on the LRC

Now, we can provide the main results on general LRC bounds for any MTL hypothesis space of the form $\Omega(\boldsymbol{W}) = \frac{1}{2}\|\boldsymbol{W}\|^2$ for a norm $\|.\|$. In what follows, the Hilbert-Schmidt operator $\phi(X) \otimes \phi(X) : \mathcal{H} \to \mathcal{H}$ is defined as $\phi(X) \otimes \phi(X)(\boldsymbol{u}) = \langle \phi(X), \boldsymbol{u}\rangle\phi(X)$.

Theorem 11 (LRC bounds for MTL models with norm regularizers) *Let the regularizer $\Omega(\boldsymbol{W})$ in* (9) *be given as an appropriate norm $\|.\|$, whose dual is denoted by $\|.\|_*$. Let the kernels be uniformly bounded, that is, $\|k\|_\infty \leq \mathcal{K} < \infty$, and $X_t^1, \ldots, X_t^n$ be an i.i.d. sample drawn from P_t. Also, assume that for each task t, the eigen-decomposition of the Hilbert-Schmidt covariance operator J_t is given by $J_t := \mathbb{E}(\phi(X_t) \otimes \phi(X_t)) = \sum_{j=1}^{\infty} \lambda_t^j \boldsymbol{u}_t^j \otimes \boldsymbol{u}_t^j$, where $(\boldsymbol{u}_t^j)_{j=1}^{\infty}$ forms an orthonormal basis of $\mathcal{H}$ and $(\lambda_t^j)_{j=1}^{\infty}$ are the corresponding eigenvalues in non-increasing order. Then, for any given positive operator $\boldsymbol{D}$ on $\mathbb{R}^T$, any $r > 0$ and any non-negative integers $h_1, \ldots, h_T$:*

$$\mathfrak{R}(\mathcal{F}, r) \leq \min_{0 \leq h_t \leq \infty} \left\{ \sqrt{\frac{r\sum_{t=1}^{T} h_t}{nT}} + \frac{\sqrt{2}R}{T}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_* \right\}, \tag{11}$$

where $\boldsymbol{V} := \left(\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}$

Proof Using the LRC's definition, we have

$$\begin{aligned}
\mathfrak{R}(\mathcal{F}, r) &= \frac{1}{nT}\mathbb{E}_{X,\sigma}\left\{\sup_{\substack{\boldsymbol{f}=(f_1,\ldots,f_T)\in\mathcal{F},\\ P\boldsymbol{f}^2\leq r}} \sum_{i=1}^{n}\left\langle (\boldsymbol{w}_t)_{t=1}^{T}, \left(\sigma_t^i\phi(X_t^i)\right)_{t=1}^{T}\right\rangle\right\} \\
&= \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\substack{\boldsymbol{f}\in\mathcal{F},\\ P\boldsymbol{f}^2\leq r}} \left\langle (\boldsymbol{w}_t)_{t=1}^{T}, \left(\sum_{j=1}^{\infty}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}\right\rangle\right\} \\
&\leq \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{P\boldsymbol{f}^2\leq r}\left\langle \left(\sum_{j=1}^{h_t}\sqrt{\lambda_t^j}\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}, \right.\right. \\
&\qquad\qquad \left.\left.\left(\sum_{j=1}^{h_t}\sqrt{\lambda_t^j}^{-1}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}\right\rangle\right\} \qquad (12) \\
&\quad + \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}}\left\langle (\boldsymbol{w}_t)_{t=1}^{T}, \left(\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}\right\rangle\right\} \qquad (13) \\
&= A_1 + A_2,
\end{aligned}$$

where A_1 and A_2 stand respectively for the first (12) and second (13) term of the previous bound.

Step 1. Controlling A_1: By applying the Cauchy-Schwartz (C.S.) inequality on A_1, one gets

$$\begin{aligned}
A_1 &\leq \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{P\boldsymbol{f}^2\leq r}\left[\left(\sum_{t=1}^{T}\left\|\sum_{j=1}^{h_t}\sqrt{\lambda_t^j}\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right\|^2\right)^{\frac{1}{2}}\right.\right. \\
&\qquad\qquad \left.\left.\left(\sum_{t=1}^{T}\left\|\sum_{j=1}^{h_t}\left(\sqrt{\lambda_t^j}\right)^{-1}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right\|^2\right)^{\frac{1}{2}}\right]\right\} \\
&= \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{P\boldsymbol{f}^2\leq r}\left[\left(\sum_{t=1}^{T}\sum_{j=1}^{h_t}\lambda_t^j\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{1}{2}}\right.\right.
\end{aligned}$$

$$\left(\sum_{t=1}^{T}\sum_{j=1}^{h_t}\left(\lambda_t^j\right)^{-1}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i),\boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{1}{2}}\Bigg]\Bigg\}.$$

With the help of Jensen's inequality and taking advantage of the fact that $\mathbb{E}_{X,\sigma}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i),\boldsymbol{u}_t^j\right\rangle^2=\frac{\lambda_t^j}{n}$ and $P\boldsymbol{f}^2\leq r$ together imply that $\frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{\infty}\lambda_t^j\left\langle\boldsymbol{w}_t,\boldsymbol{u}_t^j\right\rangle^2\leq r$ (see Lemma C.1 in the Appendix for the proof), we can further bound A_1 as

$$A_1\leq\sqrt{\frac{r\sum_{t=1}^{T}h_t}{nT}}. \tag{14}$$

Step 2. Controlling A_2: We now use Hölder's inequality to bound the second term A_2 as follows:

$$\begin{aligned}A_2&=\frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}}\left\langle(\boldsymbol{w}_t)_{t=1}^{T},\left(\sum_{j>h_t}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i),\boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right)_{t=1}^{T}\right\rangle\right\}\\&=\frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}}\left\langle\boldsymbol{D}^{1/2}\boldsymbol{W},\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\rangle\right\}\\&\overset{\text{Hölder}}{\leq}\frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}}\left\|\boldsymbol{D}^{1/2}\boldsymbol{W}\right\|.\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_*\right\}\\&\leq\frac{\sqrt{2}R}{T}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_*.\end{aligned} \tag{15}$$

Combining (15) and (14) completes the proof. ■

In what follows, we demonstrate the power of Theorem 11 by applying it to derive the LRC bounds for some popular MTL models, including group norm, Schatten norm and graph regularized models, which have been extensively studied in the MTL literature; for example, see (Maurer, 2006b; Argyriou et al., 2007a,b, 2008a; Li et al., 2015; Argyriou et al., 2014).

4.3 Group Norm Regularized MTL

We first consider the MTL scheme, which captures the inter-task relationships by the group norm regularizer $\frac{1}{2}\|\boldsymbol{W}\|_{2,q}^2:=\frac{1}{2}\left(\sum_{t=1}^{T}\|\boldsymbol{w}_t\|_2^q\right)^{2/q}$ (Argyriou et al., 2007a, 2008a; Lounici et al., 2009; Romera-Paredes et al., 2012). Its associated hypothesis space takes the form

$$\mathcal{F}_q:=\left\{X\mapsto[\langle\boldsymbol{w}_1,\phi(X_1)\rangle,\ldots,\langle\boldsymbol{w}_T,\phi(X_T)\rangle]^T:\frac{1}{2}\|\boldsymbol{W}\|_{2,q}^2\leq R_{max}^2\right\}. \tag{16}$$

Before presenting the result for this case, we point out that A_1 does not depend on the hypothesis space's $\boldsymbol{W}$-constraint. Therefore, the bound for A_1 is independent of the the choice of regularizers we consider in this study. However, A_2 can be further bounded in a manner that depends on the regularization function.

We start off with a useful lemma which helps with bounding A_2 for the group norm hypothesis space (16). The proof of this lemma, which is based on the application of the Khintchine (C.1) and Rosenthal (C.2) inequalities, is presented in Appendix C.

Lemma 12 *Assume that the kernels in* (9) *are uniformly bounded, that is,* $\|k\|_\infty\leq\mathcal{K}<\infty$. *Then, for the group norm regularizer* $\frac{1}{2}\|\boldsymbol{W}\|_{2,q}^2$ *in* (16) *and for any* $1\leq q\leq 2$, *the expectation* $\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_{2,q^*}$ *(for* $\boldsymbol{D}=\boldsymbol{I}$*) can be upper-bounded as*

$$\mathbb{E}_{X,\sigma}\left\|\left(\sum_{j>h_t}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i),\boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right)_{t=1}^{T}\right\|_{2,q^*}\leq\frac{\sqrt{\mathcal{K}}eq^*T^{\frac{1}{q^*}}}{n}+\sqrt{\frac{eq^{*2}}{n}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}.$$

Corollary 13 *Using Theorem 11, for any* $1<q\leq 2$, *the LRC of function class* $\mathcal{F}_q$ *in* (16) *can be bounded as*

$$\mathfrak{R}(\mathcal{F}_q,r)\leq\sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{q^*}},\frac{2eq^{*2}R_{max}^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}+\frac{\sqrt{2\mathcal{K}e}R_{max}q^*T^{\frac{1}{q^*}}}{nT}. \tag{17}$$

Proof Sketch: We use Lemma 12 to upper-bound A_2 for the group norm hypothesis space (16) as

$$A_2(\mathcal{F}_q)\leq\sqrt{\frac{2eq^{*2}R_{max}^2}{nT^2}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}+\frac{\sqrt{2\mathcal{K}e}R_{max}q^*T^{\frac{1}{q^*}}}{nT}. \tag{18}$$

Now, combining (14) and (18) provides the following bound on $\mathfrak{R}(\mathcal{F}_q,r)$

$$\mathfrak{R}(\mathcal{F}_q,r)\leq\sqrt{\frac{r\sum_{t=1}^{T}h_t}{nT}}+\sqrt{\frac{2eq^{*2}R_{max}^2}{nT^2}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}+\frac{\sqrt{2\mathcal{K}e}R_{max}q^*T^{\frac{1}{q^*}}}{nT}. \tag{19}$$

Then, using the inequalities shown below, which hold for any $\alpha_1,\alpha_2>0$, any vectors $\boldsymbol{a}_1,\boldsymbol{a}_2\in\mathbb{R}^T$ with non-negative elements, any $0\leq q\leq p\leq\infty$ and any $s\geq 1$,

$$(\star)\ \sqrt{\alpha_1}+\sqrt{\alpha_2}\leq\sqrt{2(\alpha_1+\alpha_2)} \tag{20}$$

$$(\star\star)\quad l_q-to-l_p:\quad\|\boldsymbol{a}_1\|_q=\langle\boldsymbol{1},\boldsymbol{a}_1^q\rangle^{\frac{1}{q}}\overset{\text{Hölder's}}{\leq}\left(\|\boldsymbol{1}\|_{(p/q)^*}\|\boldsymbol{a}_1^q\|_{(p/q)}\right)^{\frac{1}{q}}=T^{\frac{1}{q}-\frac{1}{p}}\|\boldsymbol{a}_1\|_p \tag{21}$$

$$(\star\star\star)\quad\|\boldsymbol{a}_1\|_s+\|\boldsymbol{a}_2\|_s\leq 2^{1-\frac{1}{s}}\|\boldsymbol{a}_1+\boldsymbol{a}_2\|_s\leq 2\|\boldsymbol{a}_1+\boldsymbol{a}_2\|_s, \tag{22}$$

one obtains the desired result. See Appendix C for the detailed proof.

Remark 14 *Since the LRC bound above is non-monotonic in q, it is more practical to state the above bound in terms of* $\kappa\geq q$; *note that choosing* $\kappa=q$ *is not always the optimal choice. Trivially,*

for the group norm regularizer with any $\kappa \geq q$, it holds that $\|\boldsymbol{W}\|_{2,\kappa} \leq \|\boldsymbol{W}\|_{2,q}$ and, therefore, $\mathfrak{R}(\mathcal{F}_q, r) \leq \mathfrak{R}(\mathcal{F}_\kappa, r)$. Thus, we have the following bound on $\mathfrak{R}(\mathcal{F}_q, r)$ for any $\kappa \in [q, 2]$,

$$\mathfrak{R}(\mathcal{F}_q, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{\kappa^*}}, \frac{2e\kappa^{*2}R_{max}^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\frac{\kappa^*}{2}}} + \frac{\sqrt{2Ke}R_{max}\kappa^* T^{\frac{1}{\kappa^*}}}{nT}. \tag{23}$$

Remark 15 (Sparsity-inducing group norm) *The use of the group norm regularizer $\frac{1}{2}\|\boldsymbol{W}\|_{2,1}^2$ encourages a sparse representation that is shared across multiple tasks (Argyriou et al., 2007b, 2008a). Notice that for any $\kappa \geq 1$, it holds that $\mathfrak{R}(\mathcal{F}_1, r) \leq \mathfrak{R}(\mathcal{F}_\kappa, r)$. Also, assuming an identical tail sum $\sum_{j\geq h}\lambda^j$ for all tasks, the bound gets minimized for $\kappa^* = \log T$. For this particular choice of κ^*, it is easy to show that*

$$\begin{aligned}\mathfrak{R}(\mathcal{F}_1, r) &\leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{\kappa^*}}, \frac{2e\kappa^{*2}R_{max}^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\frac{\kappa^*}{2}}} + \frac{\sqrt{2Ke}R_{max}\kappa^* T^{\frac{1}{\kappa^*}}}{nT} \\ &\overset{(l_{\frac{\kappa^*}{2}}-to-l_\infty)}{\leq} \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT, \frac{2e^3(\log T)^2R_{max}^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\infty}} + \frac{\sqrt{2K}R_{max}e^{\frac{3}{2}}\log T}{nT}.\end{aligned}$$

Remark 16 ($L_{2,q}$ Group norm regularizer with $q \geq 2$) *For any $q \geq 2$, Theorem 11 provides an LRC bound for the function class $\mathcal{F}_q$ in* (16) *given as*

$$\mathfrak{R}(\mathcal{F}_q, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{q^*}}, \frac{2R_{max}^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}, \tag{24}$$

where $q^* := \frac{q}{q-1}$.

Proof Using $\boldsymbol{D} = \boldsymbol{I}$, and $\|.\| = \|.\|_{2,q}$ in (15) gives

$$\begin{aligned}A_2(\mathcal{F}_q) &\overset{\text{Hölder's}}{\leq} \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\boldsymbol{f}\in\mathcal{F}_q}\|\boldsymbol{W}\|_{2,q}\|\boldsymbol{V}\|_{2,q^*}\right\} \\ &\leq \frac{\sqrt{2}R_{max}}{T}\mathbb{E}_{X,\sigma}\left(\sum_{t=1}^{T}\left\|\sum_{j>h_t}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right\|^{q^*}\right)^{\frac{1}{q^*}} \\ &\overset{\text{Jensen's}}{\leq} \frac{\sqrt{2}R_{max}}{T}\left(\sum_{t=1}^{T}\left(\mathbb{E}_{X,\sigma}\left\|\sum_{j>h_t}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right\|^2\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}} \\ &= \frac{\sqrt{2}R_{max}}{T}\left(\sum_{t=1}^{T}\left(\sum_{j>h_t}\mathbb{E}_{X,\sigma}\left\langle\frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}}\end{aligned}$$

$$= \frac{\sqrt{2}R_{max}}{T}\left(\sum_{t=1}^{T}\left(\sum_{j>h_t}\frac{\lambda_t^j}{n}\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}} = \sqrt{\frac{2R_{max}^2}{nT^2}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}.$$

By applying (20), (21) and (22), this last result together with the bound for A_1 in (14) yields the result. ■

To investigate the tightness of the bound in (17), we derive the corresponding lower bound, which holds for the LRC of $\mathcal{F}_q$ with $q \geq 1$. The proof of this result can be found in Appendix C.

Theorem 17 (Lower bound) *Consider the hypothesis space shown in* (16)*. The following lower bound holds for the local Rademacher complexity of $\mathcal{F}_q$ for any $q \geq 1$. There is an absolute constant c such that, if $\lambda_t^1 \geq 1/(nR_{max}^2)$ $\forall t$, then, for all $r \geq \frac{1}{n}$ and $q \geq 1$,*

$$\mathfrak{R}(\mathcal{F}_{q,R_{max},T}, r) \geq \sqrt{\frac{c}{nT^{1-\frac{2}{q^*}}}\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{q^*}}, \frac{R_{max}^2}{T}\lambda_1^j\right)}. \tag{25}$$

To make a clear comparison between the lower bound in (25) and the upper bound in (17), we assume identical eigenvalue tail sums $\sum_{j\geq\infty}\lambda_t^j$ for all tasks. In this case, the upper bound translates to

$$\mathfrak{R}(\mathcal{F}_{q,R_{max},T}, r) \leq \sqrt{\frac{4}{nT^{1-\frac{2}{q^*}}}\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{q^*}}, \frac{2eq^{*2}R_{max}^2}{T}\lambda_t^j\right)} + \frac{\sqrt{2Ke}R_{max}q^*T^{\frac{1}{q^*}}}{nT}.$$

By comparing this to (25), we see that the lower bound matches the upper bound up to constants. The same analysis for MTL models with Schatten norm and graph regularizers yields similar results and confirms that the LRC upper bounds that we have obtained are reasonably tight.

Remark 18 *It is worth pointing out that a matching lower bound on the local Rademacher complexity does not necessarily imply a tight bound on the expectation of an empirical minimizer $\hat{f}$. As shown in Section 4 of Bartlett et al. (2004), a direct analysis of the empirical minimizer can lead to sharper bounds compared to the LRC-based bounds. Consequently, based on Theorem 8 in Bartlett et al. (2004), there might be cases in which the local Rademacher complexity bounds are constants, while $P\hat{f}$ is a decreasing function of the number of samples n. Moreover, it is shown in the same paper that, under some mild conditions on the loss function ℓ, a similar argument also holds for the class of loss functions $\{\ell_f - \ell_{f^*} : f \in \mathcal{F}\}$.*

4.4 Schatten Norm Regularized MTL

Argyriou et al. (2007b) developed a spectral regularization framework for MTL, wherein the L_{S_q}-Schatten norm $\frac{1}{2}\|\boldsymbol{W}\|_{S_q}^2 := \frac{1}{2}\left[\mathrm{tr}\left(\boldsymbol{W}^T\boldsymbol{W}\right)^{\frac{q}{2}}\right]^{\frac{2}{q}}$ is studied as a concrete example that corresponds to performing ERM in the following hypothesis space:

$$\mathcal{F}_{S_q} := \left\{X \mapsto \left[\langle\boldsymbol{w}_1, \phi(X_1)\rangle, \ldots, \langle\boldsymbol{w}_T, \phi(X_T)\rangle\right]^T : \frac{1}{2}\|\boldsymbol{W}\|_{S_q}^2 \leq R'^2_{max}\right\}. \tag{26}$$

Corollary 19 *For any $1 < q \leq 2$ in (26), the LRC of function class $\mathcal{F}_{S_q}$ is bounded as*

$$\mathfrak{R}(\mathcal{F}_{S_q}, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(r, \frac{2q^* R_{max}'^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_1}.$$

The proof is provided in Appendix C.

Remark 20 (Sparsity-inducing Schatten norm (trace norm)) *Trace norm regularized MTL, corresponding to Schatten norm regularization with $q = 1$ (Maurer and Pontil, 2013; Pong et al., 2010), imposes a low-rank structure on the spectrum of $\boldsymbol{W}$. It can also be interpreted as low dimensional subspace learning (Argyriou et al., 2008b; Kumar and Daume III, 2012; Kang et al., 2011). Note that for any $q \geq 1$, it holds that $\mathfrak{R}(\mathcal{F}_{S_1}, r) \leq \mathfrak{R}(\mathcal{F}_{S_q}, r)$. Therefore, choosing the optimal $q^* = 2$, we get*

$$\mathfrak{R}(\mathcal{F}_{S_1}, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(r, \frac{4R_{max}'^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_1}.$$

Remark 21 (L_{S_q} Schatten norm regularizer with $q \geq 2$) *For any $q \geq 2$, Theorem 11 provides an LRC bound for the function class $\mathcal{F}_{S_q}$ in (26) as*

$$\mathfrak{R}(\mathcal{F}_{S_q}, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(rT^{1-\frac{2}{q^*}}, \frac{2R_{max}'^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}, \tag{27}$$

where $q^* := \frac{q}{q-1}$.

Proof We first bound the expectation $\mathbb{E}_{X,\sigma}\|\boldsymbol{V}\|_{S_{q^*}}$. Take $\boldsymbol{U}_t^i$ as a matrix with T columns where the only non-zero column t of $\boldsymbol{U}_t^i$ is defined as $\sum_{j>h_t}\left\langle \frac{1}{n}\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j$. Based on the definition of $\boldsymbol{V} = \left(\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^n \sigma_t^i \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^T$, we can then provide a bound for this expectation as follows

$$\begin{aligned}
\mathbb{E}_{X,\sigma}\|\boldsymbol{V}\|_{S_{q^*}} &= \mathbb{E}_{X,\sigma}\left\|\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i \boldsymbol{U}_t^i\right\|_{S_{q^*}} \\
&\overset{\text{Jensen}}{\leq} \left[\text{tr}\left(\left(\sum_{t,s=1}^{T}\sum_{i,j=1}^{n}\mathbb{E}_{X,\sigma}\left(\sigma_t^i\sigma_s^j \boldsymbol{U}_t^{i\,T}\boldsymbol{U}_s^j\right)\right)^{\frac{q^*}{2}}\right)\right]^{\frac{1}{q^*}} \\
&= \left[\text{tr}\left(\left(\sum_{t=1}^{T}\sum_{i=1}^{n}\mathbb{E}_X\left(\boldsymbol{U}_t^{i\,T}\boldsymbol{U}_t^i\right)\right)^{\frac{q^*}{2}}\right)\right]^{\frac{1}{q^*}} \\
&= \left[\text{tr}\left(\left(\text{diag}\left(\mathbb{E}_X\sum_{i=1}^{n}\sum_{j>h_1}\langle\frac{1}{n}\phi(X_1^i), \boldsymbol{u}_1^j\rangle^2, \ldots, \mathbb{E}_X\sum_{i=1}^{n}\sum_{j>h_T}\langle\frac{1}{n}\phi(X_T^i), \boldsymbol{u}_T^j\rangle^2\right)\right)^{\frac{q^*}{2}}\right)\right]^{\frac{1}{q^*}} \\
&= \left[\text{tr}\left(\left(\frac{1}{n}\text{diag}\left(\sum_{j>h_1}\lambda_1^j, \ldots, \sum_{j>h_T}\lambda_T^j\right)\right)^{\frac{q^*}{2}}\right)\right]^{\frac{1}{q^*}} \\
&= \sqrt{\frac{1}{n}}\left(\sum_{t=1}^{T}\left(\sum_{j>h_t}\lambda_t^j\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}} = \sqrt{\frac{1}{n}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_{\frac{q^*}{2}}}.
\end{aligned}$$

One can derive the final result by replacing this last expression into (11) and by utilizing (20), (21) and (22). ■

4.5 Graph Regularized MTL

The idea underlying graph regularized MTL is to force the models of related tasks to be close to each other, by penalizing the squared distance $\|\boldsymbol{w}_t - \boldsymbol{w}_s\|^2$ with different weights ω_{ts}. Here we consider the following MTL graph regularizer (Maurer, 2006b)

$$\Omega(\boldsymbol{W}) = \frac{1}{2}\sum_{t=1}^{T}\sum_{s=1}^{T}\omega_{ts}\|\boldsymbol{w}_t - \boldsymbol{w}_s\|^2 + \eta\sum_{t=1}^{T}\|\boldsymbol{w}_t\|^2 = \sum_{t=1}^{T}\sum_{s=1}^{T}(\boldsymbol{L} + \eta\boldsymbol{I})_{ts}\langle\boldsymbol{w}_t, \boldsymbol{w}_s\rangle,$$

where $\boldsymbol{L}$ is the graph-Laplacian associated to a matrix of edge weights ω_{ts}, $\boldsymbol{I}$ is the identity operator, and $\eta > 0$ is a regularization parameter. According to the identity $\sum_{t=1}^T\sum_{s=1}^T(\boldsymbol{L} + \eta\boldsymbol{I})_{ts}\langle\boldsymbol{w}_t, \boldsymbol{w}_s\rangle = \|(\boldsymbol{L} + \eta\boldsymbol{I})^{1/2}\boldsymbol{W}\|_F^2$, the corresponding hypothesis space is:

$$\mathcal{F}_G := \left\{X \mapsto [\left\langle \boldsymbol{w}_1, \phi(X_1)\right\rangle, \ldots, \left\langle \boldsymbol{w}_T, \phi(X_T)\right\rangle]^T : \frac{1}{2}\|\boldsymbol{D}^{1/2}\boldsymbol{W}\|_F^2 \leq R_{max}''^2\right\}. \tag{28}$$

where we define $\boldsymbol{D} := \boldsymbol{L} + \eta\boldsymbol{I}$.

Corollary 22 *For any given positive operator $\boldsymbol{D}$ in (28), the LRC of $\mathcal{F}_G$ is bounded by*

$$\mathfrak{R}(\mathcal{F}_G, r) \leq \sqrt{\frac{4}{nT}\left\|\left(\sum_{j=1}^{\infty}\min\left(r, \frac{2\boldsymbol{D}_{tt}^{-1}R_{max}''^2}{T}\lambda_t^j\right)\right)_{t=1}^{T}\right\|_1}. \tag{29}$$

where $\left(\boldsymbol{D}_{tt}^{-1}\right)_{t=1}^T$ are the diagonal elements of $\boldsymbol{D}^{-1}$.

See Appendix C for the proof.

Remark 23 *Note that if one considers a strongly convex norm of $\boldsymbol{W}$, an alternative proof strategy can be used to bound the A_2 term in* (13)*. This strategy is based on the duality of strong convexity and strong smoothness (Theorem 3 in Kakade et al. (2012)) along with the application of the Fenchel-Young inequality. This approach results in $A_2 \leq \mathcal{A}_{ub} := \sqrt{\frac{2}{\mu}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_*^2}$, where μ is the strong convexity parameter. For the strongly convex cases considered in our study (e.g. $\frac{1}{2}\|\boldsymbol{W}\|_{2,q}^2$ or $\frac{1}{2}\|\boldsymbol{W}\|_{S_q}^2$ for any $q \in (1,2]$), it holds that $\mu \leq 1$ (see Theorem 16 and Corollary 19 in Kakade et al. (2009)). Now, comparing $\sqrt{2}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_*$ in* (15) *with $\mathcal{A}_{ub}$, one can easily verify that $\sqrt{2}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_* \overset{Jensen's}{\leq} \sqrt{\frac{2}{\mu}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_*^2}$ for any $\mu \leq 1$. Therefore, for the strongly convex norms we considered here, Hölder's inequality yields slightly tighter bounds for the MT-LRC.*

5. Excess Risk Bounds for Norm Regularized MTL Models

In this section we will provide excess risk bounds for the hypothesis spaces considered earlier. Note that, due to space limitations, the proofs are provided only for the hypothesis space $\mathcal{F}_q$ with $q \in (1,2]$. However, for the cases involving the $L_{2,q}$-group norm with $q = 1$ or $q \geq 2$, as well as the L_{S_q}-Schatten and graph norms, the proofs can be obtained in a very similar fashion. More specifically, by using the LRC bounds of Remark 16, Corollary 19, Remark 21 and Corollary 22, one can follow the same proof steps shown in this section to arrive at the results pertaining to these cases.

Theorem 24 (Excess risk bound for an $L_{2,q}$ group norm regularized MTL) *Assume that $\mathcal{F}_q$ in* (16) *is a convex class of functions with ranges in $[-b,b]$ and let the loss function ℓ of Problem* (9) *satisfy Assumption 8. Let $\hat{\boldsymbol{f}}$ be any element of $\mathcal{F}_q$ for $1 < q \leq 2$ satisfying $P_n\ell_{\hat{\boldsymbol{f}}} = \inf_{\boldsymbol{f}\in\mathcal{F}_q} P_n\ell_{\boldsymbol{f}}$. Assume, moreover, that k is a positive definite kernel on $\mathcal{X}$ such that $\|k\|_\infty \leq \mathcal{K} < \infty$. Denote by r^* the fixed point of $2BL\mathfrak{R}(\mathcal{F}_q, \frac{r}{4L^2})$. Then, for any $K > 1$ and $x > 0$, with probability at least $1 - e^{-x}$, the excess loss of function class $\mathcal{F}_q$ is bounded as*

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq (2K)^{\frac{\beta}{2-\beta}} 20^{\frac{2}{2-\beta}} \max\left((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\right) + \left(\frac{2^{\beta+3}B^2K^\beta x}{nT}\right)^{\frac{1}{2-\beta}} + \frac{48LBbx}{(2-\beta)nT},$$

where the fixed point r^* of the local Rademacher complexity $2BL\mathfrak{R}(\mathcal{F}_q, \frac{r}{4L^2})$ satisfies

$$r^* \leq \min_{0\leq h_t\leq\infty} \frac{B^2\sum_{t=1}^T h_t}{nT} + 4BL\sqrt{\frac{2eq^{*2}R_{max}^2}{nT^2}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^T\right\|_{\frac{q^*}{2}}} + \frac{4\sqrt{2\mathcal{K}e}R_{max}BLq^*T^{\frac{1}{q^*}}}{nT}, \tag{30}$$

and where $h_1, \ldots, h_T$ are arbitrary non-negative integers.

Proof First, notice that $\mathcal{F}_q$ is convex and, therefore, it is star-shaped around any of its elements. Hence, according to Lemma 3.4 in Bartlett et al. (2005)—which indicates that the local Rademacher complexity of the star-hull of any function class $\mathcal{F}$ is a sub-root function—$\mathfrak{R}(\mathcal{F}_q, r)$ is a sub-root function. Moreover, because of the symmetry of the σ_t^i's distribution and because $\mathcal{F}_q$ is convex and symmetric, it can be shown that $\mathfrak{R}(\mathcal{F}_q^*, r) \leq 2\mathfrak{R}(\mathcal{F}_q, \frac{r}{4L^2})$, where $\mathfrak{R}(\mathcal{F}_q^*, r)$ is defined in (6) for the

class of functions $\mathcal{F}_q$. Therefore, it suffices to find the fixed point of $2BL\mathfrak{R}(\mathcal{F}_q, \frac{r}{4L^2})$ by solving $\phi(r) = r$. For this purpose, we will use (19) as a bound for $\mathfrak{R}(\mathcal{F}_q, r)$, and solve $\sqrt{\alpha r} + \gamma = r$ (or equivalently $r^2 - (\alpha + 2\gamma)r + \gamma^2 = 0$) for r, where we define

$$\alpha := \frac{B^2\sum_{t=1}^T h_t}{nT}, \text{ and } \gamma := 2BL\sqrt{\frac{2eq^{*2}R_{max}^2}{nT^2}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^T\right\|_{\frac{q^*}{2}}} + \frac{2\sqrt{2\mathcal{K}e}R_{max}BLq^*T^{\frac{1}{q^*}}}{nT}. \tag{31}$$

It is not hard to verify that $r^* \leq \alpha + 2\gamma$. Substituting the definition of α and γ in $r^* \leq \alpha + 2\gamma$ gives the result. ■

Now, regarding the fact that the λ_t^js are non-increasing with respect to j, we can assume $\exists d_t : \lambda_t^j \leq d_t j^{-\alpha_t}$ for some $\alpha_t > 1$. For example, this assumption holds for finite rank kernels, as well as for convolution kernels. Thus, it can be shown that

$$\sum_{j>h_t}\lambda_t^j \leq d_t\sum_{j>h_t} j^{-\alpha_t} \leq d_t\int_{h_t}^\infty x^{-\alpha_t}dx = d_t\left[\frac{1}{1-\alpha_t}x^{1-\alpha_t}\right]_{h_t}^\infty = -\frac{d_t}{1-\alpha_t}h_t^{1-\alpha_t}. \tag{32}$$

Note that via the $l_q - to - l_p$ conversion inequality (21), for $p = 1$ and $q = \frac{q^*}{2}$, we have

$$\frac{B^2\sum_{t=1}^T h_t}{Tn} \leq B\sqrt{\frac{B^2T\sum_{t=1}^T h_t^2}{n^2T^2}} \overset{(\star\star)}{\leq} B\sqrt{\frac{B^2T^{2-\frac{2}{q^*}}\left\|\left(h_t^2\right)_{t=1}^T\right\|_{\frac{q^*}{2}}}{n^2T^2}}.$$

which with the help of $\sqrt{\alpha_1} + \sqrt{\alpha_2} \leq \sqrt{2(\alpha_1+\alpha_2)}$ for any $\alpha_1, \alpha_2 > 0$, and $\|\boldsymbol{a}_1\|_s + \|\boldsymbol{a}_2\|_s \leq 2\|\boldsymbol{a}_1 + \boldsymbol{a}_2\|_s$ for any non-negative vectors $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^T$ and $s = \frac{q^*}{2}$ gives

$$r^* \leq \min_{0\leq h_t\leq\infty} 2B\sqrt{\left\|\left(\frac{B^2T^{2-\frac{2}{q^*}}h_t^2}{n^2T^2} - \frac{32d_teq^{*2}R_{max}^2L^2}{nT^2(1-\alpha_t)}h_t^{1-\alpha_t}\right)_{t=1}^T\right\|_{\frac{q^*}{2}}} + \frac{4\sqrt{2\mathcal{K}e}R_{max}BLq^*T^{\frac{1}{q^*}}}{nT}. \tag{33}$$

Taking the partial derivative of the above bound with respect to h_t and setting it to zero yields the optimal h_t as

$$h_t = \left(16d_teq^{*2}R_{max}^2B^{-2}L^2T^{\frac{2}{q^*}-2}n\right)^{\frac{1}{1+\alpha_t}}.$$

Note that substituting the previous expression for $\alpha := \min_{t\in\mathbb{N}_T}\alpha_t$ and $d = \max_{t\in\mathbb{N}_T} d_t$ into (33), we can upper-bound the fixed point of r^* as

$$r^* \leq \frac{14B^2}{n}\sqrt{\frac{\alpha+1}{\alpha-1}}\left(dq^{*2}R_{max}^2B^{-2}L^2T^{\frac{2}{q^*}-2}n\right)^{\frac{1}{1+\alpha}} + \frac{10\sqrt{\mathcal{K}}R_{max}BLq^*T^{\frac{1}{q^*}}}{nT},$$

which implies that

$$r^* = O\left(d^{\frac{1}{1+\alpha}}\left(\frac{T^{1-\frac{1}{q^*}}}{q^*}\right)^{\frac{-2}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).$$

It can be seen that the convergence rate can be as slow as $O\left(\frac{q^* T^{1/q^*}\sqrt{d}}{T\sqrt{n}}\right)$ (for small α, where at least one $\alpha_t \approx 1$), and as fast as $O(\frac{1}{n})$ (when $\alpha_t \to \infty$, for all t). The bound obtained for the fixed point together with Theorem 24 provides a bound for the excess risk, which leads to the following remark. Note that in the sequel, we assume that the data distribution of each task is concentrated and uniform on the same M-dimensional unit sphere. This implies that (by symmetry) the eigenvalues must all be equal and they sum up to 1. Thus, for each task t, $\lambda_t^j = \frac{1}{M}$. On the other hand, we assumed earlier that $\lambda_t^j \le d_t j^{-\alpha}$ for all $1 \le j \le M$. Therefore, choosing $j = M$, we are forced to set $d = M^{\alpha-1}$.

Remark 25 (Excess risk bounds for selected norm regularized MTL problems) *Assume that $\mathcal{F}$ is a class of functions with ranges in $[-b, b]$. Let the loss function ℓ of Problem* (9) *satisfy Assumption 8. Additionally, assume that k is a positive definite kernel on $\mathcal{X}$, such that $\|k\|_\infty \le \mathcal{K} < \infty$. Also, denote $\alpha := \min_{t\in\mathbb{N}_T} \alpha_t$ and $d := max_{t\in\mathbb{N}_T} d_t$. Then, for any $\boldsymbol{f} \in \mathcal{F}$, $K > 1$ and $x > 0$, with probability at least $1 - e^{-x}$,*

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \le (2K)^{\frac{\beta}{2-\beta}} 20^{\frac{2}{2-\beta}} \max\left((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\right) + \left(\frac{2^{\beta+3} B^2 K^\beta x}{nT}\right)^{\frac{1}{2-\beta}} + \frac{48LBbx}{(2-\beta)nT}, \tag{34}$$

where, for $\mathcal{F} \in \{\mathcal{F}_q, \mathcal{F}_{S_q}, \mathcal{F}_G\}$, $\hat{\boldsymbol{f}}$ is such that $P_n \ell_{\hat{\boldsymbol{f}}} = \inf_{\boldsymbol{f}\in\mathcal{F}} P_n \ell_{\boldsymbol{f}}$ and r^* is the fixed point of the local Rademacher complexity $2BL\mathfrak{R}(\mathcal{F}, \frac{r}{4L^2})$. Furthermore, r^* can be bounded for each of the three hypothesis spaces as follows:

- *Group norm: For any $1 < q \le 2$,*

$$r^* \le \min_{\kappa\in[q,2]} 14\sqrt{\frac{\alpha+1}{\alpha-1}}\left(\kappa^{*2} R_{max}^2 L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} \left(T^{\frac{2}{\kappa}}\right)^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}} + \frac{10\sqrt{\mathcal{K}} R_{max} BL\kappa^* T^{\frac{1}{\kappa^*}}}{nT}. \tag{35}$$

  Also, for any $q \ge 2$, we have

$$r^* \le 8\sqrt{\frac{\alpha+1}{\alpha-1}}\left(R_{max}^2 L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} \left(T^{\frac{2}{q}}\right)^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}. \tag{36}$$

- *Schatten norm: For any $1 < q \le 2$,*

$$r^* \le 8\sqrt{\frac{\alpha+1}{\alpha-1}}\left(q^* R_{max}'^2 L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} T^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}. \tag{37}$$

  Note that for the trace norm, we would have $q^* = 2$ in the previous bound (see Remark 20). Additionally, for any $q \ge 2$, it holds

$$r^* \le 8\sqrt{\frac{\alpha+1}{\alpha-1}}\left(R_{max}'^2 L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} \left(T^{\frac{2}{q}}\right)^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}. \tag{38}$$

- *Graph regularizer: For any positive operator $\boldsymbol{D}$,*

$$r^* \le 8\sqrt{\frac{\alpha+1}{\alpha-1}}\left(R_{max}''^2 L^2 \boldsymbol{D}_{max}^{-1}\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} T^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}. \tag{39}$$

  where $\boldsymbol{D}_{max}^{-1} := \max_{t\in\mathbb{N}_T} \boldsymbol{D}_{tt}^{-1}$.

6. Discussion

In this section, we investigate the convergence rate of our LRC-based excess risk bounds, which were established in the previous section. We also discuss related works and provide a new excess risk bound by employing a rather different approach, which exhibits the benefit of a MTL regularizer at the expense of a slower convergence rate in terms of the number of examples per task n. Note that, for the purpose of this section, we will assume that $\beta = 1$, which hold for many loss function classes, see Bartlett et al. (2004) for a discussion.

6.1 Convergence Rates

In order to facilitate a more concrete comparison of convergence rates, we will assume the same spherical M-dimensional data distribution for each task t; this assumption leads to $\lambda_t^j = \frac{1}{M}$, or equivalently $d = M^{\alpha-1}$. Furthermore, we will concentrate only on the parameters R, n, T, q^*, M and α and we will assume that all the other parameters are fixed and, hence, hidden in the big-O notation. Thus, for our LRC-based bounds we have

$$\begin{aligned}
\text{Group norm:}\quad & \text{(a) } \forall \kappa \in [q,2], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}^2 \kappa^{*2}\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} \left(T^{\frac{2}{\kappa}}\right)^{-\frac{1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).\\
& \text{(b) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} \left(T^{\frac{2}{q}}\right)^{-\frac{1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).\\
\text{Schatten norm:}\quad & \text{(c) } \forall q \in (1,2], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}'^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} T^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).\\
& \text{(d) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}'^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} \left(T^{\frac{2}{q}}\right)^{-\frac{1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).\\
\text{Graph:}\quad & \text{(e)} \qquad\qquad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}''^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} T^{\frac{-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}\right).
\end{aligned} \tag{40}$$

A close appraisal of the results in (40) points to a conservation of asymptotic rates between n and T, when all other remaining quantities are held fixed. This phenomenon is more apparent for the Schatten norm and graph-based regularization cases, where the rates (exponents) of n and T sum up to -1. Note that the trade-off is determined by the value of α, which can facilitate faster n-rates and, simultaneously, compromise with slower T-rates. A similar trade-off is witnessed in the case of group norm regularization, but this time between n and $T^{2/\kappa}$, instead of T, due to the specific characteristics of the group norm. Now, consider the following two cases:

- M is large (high-dimensional data distribution): Note that in the case of very large M, $\alpha > 1$; Also, large M implies small α, that is, $\alpha \to 1$. In this case we get dimension-independent bounds, which should be considered as an advantage for the case of high-dimensional data distribution.

$$\text{Group norm:}\quad \text{(a) } \forall \kappa \in [q,2], \qquad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(\left(R_{max}^2 \kappa^{*2}\right)^{\frac{1}{2}} \left(T^{\frac{2}{\kappa}}\right)^{-\frac{1}{2}} n^{-\frac{1}{2}}\right).$$

$$\text{(b) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left((R_{max}^2)^{\frac{1}{2}} \left(T^{\frac{2}{q}}\right)^{-\frac{1}{2}} n^{-\frac{1}{2}}\right).$$

Schatten-norm:

$$\text{(c) } \forall q \in (1,2], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left((R_{max}'^2 q^*)^{\frac{1}{2}} T^{\frac{-1}{2}} n^{-\frac{1}{2}}\right).$$

$$\text{(d) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left((R_{max}'^2)^{\frac{1}{2}} \left(T^{\frac{2}{q}}\right)^{-\frac{1}{2}} n^{-\frac{1}{2}}\right).$$

Graph:

$$\text{(e)} \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left((R_{max}''^2)^{\frac{1}{2}} T^{\frac{-1}{2}} n^{-\frac{1}{2}}\right).$$

- M is small (low-dimensional data distribution): This case happens when the decay rate α is fast ($\alpha \to \infty$), which gives the following rates

Group norm:

$$\text{(a) } \forall \kappa \in [q,2], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(Mn^{-1}\right).$$

$$\text{(b) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(Mn^{-1}\right).$$

Schatten-norm:

$$\text{(c) } \forall q \in (1,2], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(Mn^{-1}\right).$$

$$\text{(d) } \forall q \in [2,\infty], \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(Mn^{-1}\right).$$

Graph:

$$\text{(e)} \quad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) = O\left(Mn^{-1}\right).$$

Note that, most likely, a more realistic case lies somewhere in between these two extreme cases, which can be interpreted as follows: when the data is relatively low-dimensional (small M and fast decay of eigenvalues), we will have bounds with fast rates in n. However, MTL may offer little advantage in this case due to the corresponding slow rates in T. This analysis confirms the general belief that MTL proffers a potential advantage if there are many tasks with little data per task and are sampled from high-dimensional data distributions.

6.2 Comparisons to Related Works

It is interesting to compare our local bound for the trace norm regularized MTL with the GRC-based excess risk bound provided in Maurer and Pontil (2013), wherein they apply a trace norm regularizer to capture the tasks' relatedness. It is worth mentioning that they consider a slightly different hypothesis space for $\boldsymbol{W}$ than the one we mentioned earlier; in our notation, this space reads as

$$\mathcal{F}'_{S_1} := \left\{ \boldsymbol{W} : \frac{1}{2} \|\boldsymbol{W}\|_{S_1}^2 \leq T R_{max}'^2 \right\}. \tag{41}$$

The form of this space is based on the premise that, assuming a common vector $\boldsymbol{w}$ for all tasks, the regularizer should not be a function of the number of tasks (Maurer and Pontil, 2013). Given the task-averaged covariance operator $C := 1/T \sum_{t=1}^T J_t = 1/T \sum_{t=1}^T \mathbb{E}\left(\phi(X_t) \otimes \phi(X_t)\right)$, the excess risk bound in Maurer and Pontil (2013) reads as

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq 2\sqrt{2} L R'_{max} \left(\sqrt{\frac{\|C\|_\infty}{n}} + 5 \sqrt{\frac{\ln(nT)+1}{nT}} \right) + \sqrt{\frac{bLx}{nT}}.$$

Under the aforementioned M-dimensional data distributions and by using the hypothesis space of (41), our local bound for the trace norm for any $\alpha > 1$ is given as

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq 6400K \sqrt{\frac{\alpha+1}{\alpha-1}} \left(R_{max}'^2 L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} n^{\frac{-\alpha}{1+\alpha}} + \frac{(48Lb + 16BK)Bx}{nT}. \tag{42}$$

Now, let λ_t^{max} be the maximum eigenvalue of the trace operator J_t. Also, let $\lambda_{max} := \max_{t \in \mathbb{N}_T} \{\lambda_t^{max}\}$. It is easy to verify that $\mathbf{tr}(J_t) \leq M \lambda_t^{max}$ and $\|C\|_\infty \leq \lambda_{max} = 1/M$, which renders the GRC-based bound in Maurer and Pontil (2013) into the form

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq 2\sqrt{2} L R'_{max} \left(\sqrt{\frac{\lambda_{max}}{n}} + 5 \sqrt{\frac{\ln(nT)+1}{nT}} \right) + \sqrt{\frac{bLx}{nT}}. \tag{43}$$

One observes that, in both cases, the bound vanishes as $n \to \infty$. However, it does so at a rate of $n^{-\alpha/1+\alpha}$ for our local bound in (42) and at a slower rate of $\sqrt{\ln n / n}$ for the one in (43). Also, we remark that, as $T \to \infty$, both bounds converge to a non-zero limit: our local bound in (42) at a fast rate of $1/T$ and at a the slower rate of $\sqrt{\ln T / T}$ for the bound in (43). More specifically, making the benevolent choices $B = 1$ and $R'_{max} L = 1$ and ignoring the factor of $6400K\sqrt{\frac{\alpha+1}{\alpha-1}}$, the limit of our local bound in (42) as $T \to \infty$ becomes $g(\alpha) := M^{\frac{\alpha-1}{1+\alpha}} n^{\frac{-\alpha}{1+\alpha}}$. One can very easily verify that $g(\alpha)$ is increasing in α (*i.e.* $g'(\alpha) > 0$), if and only if $\ln(Mn^{-\frac{1}{2}}) > 0$, or, equivalently, $M > \sqrt{n}$. In this case the optimal choice of $\alpha \in (1,\infty)$ (*i.e.* $\alpha \approx 1$) makes our local bound of the order $O(\frac{1}{\sqrt{n}})$. In other words, when the data distribution is sufficiently high-dimensional relative to n, the LRC bound fails in competing with the $O(\frac{1}{\sqrt{Mn}})$ GRC bound in Maurer and Pontil (2013). On the other hand, for lower dimensional distributions or sufficiently large n, we obtain a rate of $1/n$ for the LRC bound at the expense of explicit dependence on the dimension. In particular, the local bound remains larger than the GRC bound in (43) until $n = M^3$ and improves only for larger sample sizes per task.

Another interesting comparison can be performed between our bounds and the one introduced in Maurer (2006b) for a graph regularized MTL. Similar to Maurer (2006b), we consider the following hypothesis space

$$\mathcal{F}'_G = \left\{ \boldsymbol{W} : \frac{1}{2} \left\| \boldsymbol{D}^{1/2} \boldsymbol{W} \right\|_F^2 \leq T R_{max}''^2 \right\}. \tag{44}$$

Maurer (2006b) provides a bound on the empirical GRC of the aforementioned hypothesis space that can be easily converted to a distribution dependent GRC bound of the form

$$\mathfrak{R}\left(\mathcal{F}'_G\right) \leq \sqrt{\frac{2R_{max}''^2}{nT} \left\| \left(\boldsymbol{D}_{tt}^{-1} \mathbf{tr}(J_t)\right)_{t=1}^T \right\|_1}.$$

Now, with $\boldsymbol{D} := \boldsymbol{L} + \eta \boldsymbol{I}$ (where $\boldsymbol{L}$ is the graph-Laplacian, $\boldsymbol{I}$ is the identity operator, and $\eta > 0$ is a regularization parameter) and the same M-dimensional distributional assumptions, it can be shown that

$$\left\| \left(\boldsymbol{D}_{tt}^{-1} \mathbf{tr}(J_t)\right)_{t=1}^T \right\|_1 = \sum_{t=1}^T \boldsymbol{D}_{tt}^{-1} \mathbf{tr}(J_t) \leq M \lambda_{max} \sum_{t=1}^T \boldsymbol{D}_{tt}^{-1} = M \lambda_{max} \mathbf{tr}\left(\boldsymbol{D}^{-1}\right) =$$

$$= M \lambda_{max} \mathbf{tr}\left(\boldsymbol{L} + \eta \boldsymbol{I}\right)^{-1} = M \lambda_{max} \left(\sum_{t=1}^T \frac{1}{\delta_t + \eta} + \frac{1}{\eta} \right) \leq M \lambda_{max} \left(\frac{T}{\delta_{min} + \eta} + \frac{1}{\eta} \right).$$

where $\lambda_{max} = \frac{1}{M}$ as argued earlier. Furthermore, let $\{\delta_2, \ldots, \delta_T\}$ be the nonzero eigenvalues of $\boldsymbol{L}$ with $\delta_{min} := \min\{\delta_2, \ldots, \delta_T\}$. Then, the GRC-based excess risk bound is obtained as

$$\text{Maurer (2006b) :} \qquad P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq \frac{2LR''_{max}}{\sqrt{n}} \sqrt{2M\lambda_{max} \left(\frac{1}{\delta_{min}} + \frac{1}{T\eta} \right)} + \sqrt{\frac{bLx}{nT}}$$

(45)

Also, based on Remark 25, the LRC-based bound is given as

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq 6400K\sqrt{\frac{\alpha+1}{\alpha-1}}\left(R''^2_{max}L^2\boldsymbol{D}^{-1}_{max}\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} n^{\frac{-\alpha}{1+\alpha}} + \frac{(48Lb+16BK)Bx}{nT}. \tag{46}$$

The above results show that, when $n \to \infty$, both GRC and LRC bounds approach zero, albeit at different rates: the global bound at a rate of $\sqrt{1/n}$ and the local one at a faster rate of $n^{-\alpha/\alpha+1}$, since $\alpha > 1$. Additionally, both bounds approach non-zero limits as $T \to \infty$. Nevertheless, the global bound does so at a rate of $\sqrt{1/T}$ and the local one at a faster rate of $1/T$. Furthermore, similar to the previous case, it can be shown that at the limit $T \to \infty$, for high-dimensional data distribution (large M, small $\alpha \approx 1$), both local and global bounds yield the same convergence rate of $O(\frac{1}{\sqrt{n}})$. However, for low number of dimensions relative to n (in specific, for $M < n^{\frac{1}{3}}$), our bound improves over the GRC bound.

6.3 A Different Technique for The Trace Norm Regularized Space $\mathcal{F}'_{S_1}$

In what follows, we show that, by applying a rather different proof technique (departing from Theorem 11), we can obtain an excess risk bound for the MTL space $\mathcal{F}'_{S_1}$ in (41), which aims at slower rates in n and T, but exhibits the benefits of a multi-task regularizer. Recall that $\mathcal{F}'_{S_1}$ is given as

$$\mathcal{F}'_{S_1} := \left\{ X \mapsto [\langle \boldsymbol{w}_1, \phi(X_1)\rangle, \ldots, \langle \boldsymbol{w}_T, \phi(X_T)\rangle]^T : \frac{1}{2}\|\boldsymbol{W}\|^2_{S_1} \leq TR'^2_{max} \right\}. \tag{47}$$

Also recall that, from Theorem 11, it can be shown that the LRC of $\mathcal{F}'_{S_1}$ can be bounded as

$$\mathfrak{R}(\mathcal{F}'_{S_1}, r) \leq \min_{0\leq h_t \leq \infty} \left\{ \sqrt{\frac{r\sum_{t=1}^T h_t}{nT}} + \sqrt{\frac{2R'^2_{max}}{n^2T}}\mathbb{E}_{X,\sigma}\left\|\boldsymbol{V}'\right\|_{S_\infty} \right\}, \tag{48}$$

where

$$\boldsymbol{V}' := \left(\sum_{j>h_t} \left\langle \sum_{i=1}^n \sigma_t^i \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle \boldsymbol{u}_t^j \right)_{t=1}^T. \tag{49}$$

Now, following an approach similar to the one applied in Maurer and Pontil (2013), we will bound $\mathbb{E}_{X,\sigma}\|\boldsymbol{V}'\|_{S_\infty}$ to yield the next theorem. Note that $\|.\|_{S_\infty}$ stands for the operator norm on the separable Hilbert space $\mathcal{H}$.

Theorem 26 *Assume that the conditions of Theorem 24 hold for the hypothesis space $\mathcal{F}'_{S_1}$ in* (47)*. Also, denote by r^* the fixed point of $2BL\mathfrak{R}(\mathcal{F}'_{S_1}, \frac{r}{4L^2})$. Then, for any $K > 1$ and $x > 0$, with probability at least $1 - e^{-x}$, the excess loss of function class $\mathcal{F}'_{S_1}$ is bounded as*

$$P(\ell_{\hat{\boldsymbol{f}}} - \ell_{\boldsymbol{f}^*}) \leq (2K)^{\frac{\beta}{2-\beta}} 20^{\frac{2}{2-\beta}} \max\left((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\right) + \left(\frac{2^{\beta+3}B^2K^\beta x}{nT}\right)^{\frac{1}{2-\beta}} + \frac{48LBbx}{(2-\beta)nT},$$

for

$$r^* \leq \min_{0\leq h_t\leq\infty} \left\{ \frac{B^2\sum_{t=1}^T h_t}{nT} + 4BL\sqrt{\frac{2R'^2_{max}\lambda_h}{n}} + 24BL\sqrt{\frac{2R'^2_{max}\mathcal{K}\left(\ln(nT)+1\right)}{nT}} \right\}, \tag{50}$$

where $\lambda_h := \max_{t\in\mathbb{N}_T}\{\lambda_t^{h_t}\}$ and where $h_1, \ldots, h_T$ are arbitrary non-negative integers.

The proof of the results is provided in Appendix D.

By considering the same M-dimensional data distribution, the bound in (50) becomes

$$r^* \leq 6BLR'_{max}\left(\sqrt{\frac{1}{Mn}} + 6\sqrt{\frac{\mathcal{K}\left(\ln(nT)+1\right)}{nT}}\right). \tag{51}$$

It can be seen that, when the number of tasks T approaches ∞, the above bound simplifies to

$$r^* \leq \frac{6BLR'_{max}}{\sqrt{Mn}}.$$

In the sequel, we compare the two bounds (37) and (51) for the trace norm regularized MTL models in terms of their convergence rates.

Remark 27 *Using two different techniques, we proved the two following bounds on the fixed point r^* of the local Rademacher complexity $2BL\mathfrak{R}(\mathcal{F}'_{S_1}, \frac{r}{4L^2})$:*

- *Our approach*

$$r^* \leq 12\sqrt{\frac{\alpha+1}{\alpha-1}}\left(R'^2_{max}L^2\right)^{\frac{1}{1+\alpha}} M^{\frac{\alpha-1}{1+\alpha}} B^{\frac{2\alpha}{\alpha+1}} n^{\frac{-\alpha}{1+\alpha}}. \tag{52}$$

- *MP approach (Maurer and Pontil, 2013)*

$$r^* \leq 6BLR'_{max}\left(\sqrt{\frac{1}{Mn}} + 6\sqrt{\frac{\mathcal{K}\left(\ln(nT)+1\right)}{nT}}\right). \tag{53}$$

As a reminder, the proof of Theorem 11 refers to two terms: A_1, which embodies a variance constraint, and A_2, which constitutes a MTL regularization constraint. The aforementioned bounds were derived by using two different approaches to bound the A_2 term, namely the LRC-based approach for (52) *and the MP technique for* (53)*. In the case of* (52)*, due to the LRC-based approach, the variance constraint (A_1 term) plays a dominant role in the overall bound and, thus, yields faster rates in n for any $\alpha > 1$. However, it offers no improvements in the limit $T \to \infty$, since this bound does not decrease with increasing T. In contrast, using the MP technique, the MTL regularization constraint (A_2 term) is dominant in* (53)*. While this prevents obtaining faster rates in terms of the number of samples, it potentially offers the advantages of MTL for large T and high-dimensional data distributions.*

Acknowledgments

NY acknowledges financial support from National Science Foundation (NSF) grant No. 1161228, and No. 1200566. MK acknowledges support from the German Research Foundation (DFG) award KL 2698/2-1 and from the Federal Ministry of Science and Education (BMBF) award 031B0187B. YL acknowledges support from the Science and Technology Innovation Committee Foundation of Shenzhen (Grant No. ZDSYS201703031748284). Finally, GCA acknowledges partial support from the US National Science Foundation (NSF) under Grant No. 1560345. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF.

Appendices

Appendix A. Proof of Theorem 1

This section presents the proof of Theorem 1. We first provide some useful foundations used in the derivation of our result in Theorem 1.

Theorem A.1 (Theorem 2 in Boucheron et al. (2003)) *Let $X_1, \dots, X_n$ be n independent random variables taking values in a measurable space $\mathcal{X}$. Assume that $g : \mathcal{X}^n \to \mathbb{R}$ is a measurable function and $Z := g(X_1, \dots, X_n)$. Let $X'_1, \dots, X'_n$ denote an independent copy of $X_1, \dots, X_n$, and $Z'_i := g(X_1, \dots, X_{i-1}, X'_i, X_{i+1}, \dots, X_n)$, which is obtained by replacing the variable X_i with X'_i. Define the random variable $V^+ := \sum_{i=1}^n \mathbb{E}'\big[(Z - Z'_i)_+^2\big]$, where $(u)_+ := \max\{u, 0\}$, and $\mathbb{E}'[\cdot] := \mathbb{E}[\cdot|X]$ denotes the expectation only w.r.t. the variables $X'_1, \dots, X'_n$. Let $\theta > 0$ and $\lambda \in (0, 1/\theta)$. Then,*

$$\log \mathbb{E}\big[e^{\lambda(Z - \mathbb{E}Z)}\big] \leq \frac{\lambda\theta}{1 - \lambda\theta} \log \mathbb{E}\big[\exp\big(\frac{\lambda V^+}{\theta}\big)\big].$$

Definition A.2 (Section 3.3 in Boucheron et al. (2013)) *A function $g : \mathcal{X}^n \to [0, \infty)$ is said to be b-self bounding ($b > 0$), if there exist functions $g_i : \mathcal{X}^{n-1} \to \mathbb{R}$, such that for all $X_1, \dots, X_n \in \mathcal{X}$ and all $i \in \mathbb{N}_n$,*

$$0 \leq g(X_1, \dots, X_n) - g_i(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n) \leq b,$$

and

$$\sum_{i=1}^n \big[g(X_1, \dots, X_n) - g_i(X_1, \dots, X_{i-1}, X_{i+1}, \dots, X_n)\big] \leq g(X_1, \dots, X_n).$$

Theorem A.3 (Theorem 6.12 in Boucheron et al. (2013)) *Assume that $Z = g(X_1, \dots, X_n)$ is a b-self bounding function ($b > 0$). Then, for any $\lambda \in \mathbb{R}$ we have*

$$\log \mathbb{E}e^{\lambda Z} \leq \frac{\big(e^{\lambda b} - 1\big)}{b} \mathbb{E}Z.$$

Lemma A.4 (Lemma 2.11 in Bousquet (2002)) *Let Z be a random variable, $A, B > 0$ be some constants. If for any $\lambda \in (0, 1/B)$ it holds*

$$\log \mathbb{E}\big(e^{\lambda(Z - \mathbb{E}Z)}\big) \leq \frac{A\lambda^2}{2\big(1 - B\lambda\big)},$$

then, for all $x \geq 0$,

$$\Pr\big[Z \geq \mathbb{E}Z + \sqrt{2Ax} + Bx\big] \leq e^{-x}.$$

Lemma A.5 (Contraction property in Bartlett et al. (2005)) *Let ϕ be a Lipschitz function with Lipschitz constant $L \geq 0$, that is, $|\phi(a) - \phi(b)| \leq L|a - b|$, $\forall a, b \in \mathbb{R}$. Let $X_1, \dots, X_n$ be n independent random variables. Then, for every real-valued function class $\mathcal{F}$, it holds*

$$\mathbb{E}_\sigma \sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i \phi(f(X_i)) \leq L \mathbb{E}_\sigma \sup_{f \in \mathcal{F}} \sum_{i=1}^n \sigma_i f(X_i). \tag{A.1}$$

Note that, in Theorem 17 of Maurer (2006a), it has been shown that the result of this lemma also holds for classes of vector-valued functions.

Proof of Theorem 1

Before laying out the details, we first provide a sketch of the proof. By defining

$$Z := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big[\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]\Big], \tag{A.2}$$

we first apply Theorem A.1 to control the log-moment generating function $\log\mathbb{E}\big(e^{\lambda(Z-\mathbb{E}Z)}\big)$. From Theorem A.1, we know that the main component to control $\log\mathbb{E}\big(e^{\lambda(Z-\mathbb{E}Z)}\big)$ is the variance-type quantity $V^+ = \sum_{s=1}^{T}\sum_{j=1}^{N_s}\mathbb{E}'\big[\big(Z - Z'_{s,j}\big)_+^2\big]$. In the next step, we show that V^+ can also be bounded in terms of two other quantities denoted by W and Υ. Applying Theorem A.1 for a specific value of θ, then gives a bound for $\log\mathbb{E}\big(e^{\lambda(Z-\mathbb{E}Z)}\big)$ in terms of $\log\mathbb{E}[e^{\frac{\lambda}{b'}(W+\Upsilon)}]$. We then turn to controlling W and Υ respectively. Our approach to tackle W is to show that it is a self-bounding function and then apply Theorem A.3 to control $\log\mathbb{E}[e^{\frac{\lambda W}{b'}}]$. The Υ term is closely related to the constraint imposed on the variance of functions in $\mathcal{F}$ and can be easily upper-bounded in terms of r. We finally apply Lemma A.4 to transfer the upper bound on the log-moment generating function $\log\mathbb{E}\big(e^{\lambda(Z-\mathbb{E}Z)}\big)$ to the tail probability on Z. For clarify, we divide the proof into four main steps.

Step 1. Controlling the log-moment generating function of Z with the random variable W and variance Υ. Let $X' := (X_t'^i)_{(t,i)=(1,1)}^{(T,N_t)}$ be an independent copy of $X := (X_t^i)_{(t,i)=(1,1)}^{(T,N_t)}$. Define the quantity $Z'_{s,j}$ by replacing the variable X_s^j in Z with $X_s'^j$. Then,

$$Z'_{s,j} := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big[\frac{1}{TN_s}\big[\mathbb{E}'f_s(X_s'^j) - f_s(X_s'^j)\big] - \frac{1}{TN_s}\big[\mathbb{E}f_s(X_s^j) - f_s(X_s^j)\big] + \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]\Big]. \tag{A.3}$$

Let $\hat{\boldsymbol{f}} := (\hat{f}_1, \dots \hat{f}_T)$ be such that $Z = \frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\big[\mathbb{E}\hat{f}_t(X_t^i) - \hat{f}_t(X_t^i)\big]$ and introduce

$$W := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big[\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]^2\Big],$$

$$\Upsilon := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big[\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}\mathbb{E}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]^2\Big].$$

It can be shown that, for any $j \in \mathbb{N}_n$ and any $s \in \mathbb{N}_T$,

$$Z - Z'_{s,j} \le \frac{1}{TN_s}\big[\mathbb{E}\hat{f}_s(X_s^j) - \hat{f}_s(X_s^j)\big] - \frac{1}{TN_s}\big[\mathbb{E}'\hat{f}_s(X_s'^j) - \hat{f}_s(X_s'^j)\big]$$

and, therefore,

$$(Z - Z'_{s,j})_+^2 \le \frac{1}{T^2N_s^2}\big([\mathbb{E}\hat{f}_s(X_s^j) - \hat{f}_s(X_s^j)] - [\mathbb{E}'\hat{f}_s(X_s'^j) - \hat{f}_s(X_s'^j)]\big)^2.$$

Then, from the identity $\mathbb{E}'[\mathbb{E}'\hat{f}_s(X_s'^j) - \hat{f}_s(X_s'^j)] = 0$, it follows that

$$\sum_{s=1}^{T}\sum_{j=1}^{N_s}\mathbb{E}'\big[\big(Z - Z'_{s,j}\big)_+^2\big] \le \sum_{s=1}^{T}\sum_{j=1}^{N_s}\frac{1}{T^2N_s^2}\mathbb{E}'\Big[\Big([\mathbb{E}\hat{f}_s(X_s^j) - \hat{f}_s(X_s^j)] - [\mathbb{E}'\hat{f}_s(X_s'^j) - \hat{f}_s(X_s'^j)]\Big)^2\Big]$$

$$\begin{aligned}
&= \sum_{s=1}^{T}\sum_{j=1}^{N_s}\frac{1}{T^2N_s^2}[\mathbb{E}\hat{f}_s(X_s^j) - \hat{f}_s(X_s^j)]^2 + \sum_{s=1}^{T}\sum_{j=1}^{N_s}\frac{1}{T^2N_s^2}\mathbb{E}'[\mathbb{E}'\hat{f}_s(X_s'^j) - \hat{f}_s(X_s'^j)]^2\\
&\le \sup_{\boldsymbol{f}\in\mathcal{F}}\sum_{s=1}^{T}\sum_{j=1}^{N_s}\frac{1}{T^2N_s^2}[\mathbb{E}f_s(X_s^j) - f_s(X_s^j)]^2 + \sup_{\boldsymbol{f}\in\mathcal{F}}\sum_{s=1}^{T}\sum_{j=1}^{N_s}\frac{1}{T^2N_s^2}\mathbb{E}[\mathbb{E}f_s(X_s^j) - f_s(X_s^j)]^2\\
&= W + \Upsilon.
\end{aligned}$$

Introduce $b' := \frac{2b}{nT}$. Applying Theorem A.1 and the above bound to $\sum_{s=1}^{T}\sum_{j=1}^{N_s}\mathbb{E}'\big[\big(Z - Z'_{s,j}\big)_+^2\big]$ yields the following bound on the log-moment generating function of Z

$$\log\mathbb{E}\big[e^{\lambda(Z-\mathbb{E}Z)}\big] \le \frac{\lambda b'}{1-\lambda b'}\log\mathbb{E}\big[e^{\frac{\lambda}{b'}(W+\Upsilon)}\big], \quad \forall\lambda \in (0, 1/b'). \tag{A.4}$$

Step 2. Controlling the log-moment generating function of W. We now upper-bound the log-moment generating function of W by showing that it is a self-bounding function. For any $s \in \mathbb{N}_T, j \in \mathbb{N}_{N_s}$, introduce

$$W_{s,j} := \sup_{\boldsymbol{f}\in\mathcal{F}} \Big[\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]^2 - \frac{1}{T^2N_s^2}[\mathbb{E}f_s(X_s^j) - f_s(X_s^j)]^2\Big].$$

Note that $W_{s,j}$ is a function of $\{X_t^i, t \in \mathbb{N}_T, i \in \mathbb{N}_{N_t}\}\backslash\{X_s^j\}$. Letting $\tilde{\boldsymbol{f}} := (\tilde{f}_1, \dots, \tilde{f}_T)$ be the function achieving the supremum in the definition of W, one can verify that (note that $b' = \frac{2b}{nT}$)

$$T^2[W - W_{s,j}] \le \frac{1}{N_s^2}[\mathbb{E}\tilde{f}_s(X_s^j) - \tilde{f}_s(X_s^j)]^2 \le \frac{4b^2}{n^2} = T^2b'^2. \tag{A.5}$$

Similarly, if $\tilde{\boldsymbol{f}}^{s,j} := (\tilde{f}_1^{s,j} \dots, \tilde{f}_T^{s,j})$ is the function achieving the supremum in the definition of $W_{s,j}$, then one can derive the following inequality

$$T^2[W - W_{s,j}] \ge \frac{1}{N_s^2}[\mathbb{E}\tilde{f}_s^{s,j}(X_s^j) - \tilde{f}_s^{s,j}(X_s^j)]^2 \ge 0.$$

Also, it can be shown that

$$\begin{aligned}
\sum_{s=1}^{T}\sum_{i=1}^{N_s}[W - W_{s,j}] &\le \frac{1}{T^2}\sum_{s=1}^{T}\frac{1}{N_s^2}\sum_{i=1}^{N_s}[\mathbb{E}\tilde{f}_s(X_s^j) - \tilde{f}_s(X_s^j)]^2\\
&= \sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)]^2\Big] = W.
\end{aligned} \tag{A.6}$$

Therefore, according to Definition A.2, W/b' is a b'-self bounding function. Applying Theorem A.3 then gives the following inequality for any $\lambda \in (0, 1/b')$:

$$\log\mathbb{E}e^{\lambda(W/b')} \le \frac{(e^{\lambda b'} - 1)}{b'^2}\mathbb{E}W = \frac{(e^{\lambda b'} - 1)}{b'^2}\Sigma^2 \le \frac{\lambda\Sigma^2}{b'(1-\lambda b')}, \tag{A.7}$$

where we introduced $\Sigma^2 := \mathbb{E}W$ and where the last step uses the inequality $(e^x - 1)(1 - x) \leq x, \forall x \in [0, 1]$. By further noting that (σ_t^i) is a sequence of independent Rademacher variables independent of X_t^i, the Σ^2 term can be controlled as follows

$$\begin{aligned}
\Sigma^2 &\leq \frac{1}{T^2}\mathbb{E}_X \sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}\big[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)\big]^2 - \sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}\mathbb{E}\big[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)\big]^2\Big] + \Upsilon \\
&\leq 2\mathbb{E}_{X,\sigma}\Big[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}\sigma_t^i\big[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)\big]^2\Big] + \Upsilon \\
&\leq 8b\mathbb{E}_{X,\sigma}\Big[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{T^2}\sum_{t=1}^{T}\frac{1}{N_t^2}\sum_{i=1}^{N_t}\sigma_t^i\big[\mathbb{E}f_t(X_t^i) - f_t(X_t^i)\big]\Big] + \Upsilon \\
&\leq \frac{16b\mathfrak{R}(\mathcal{F})}{nT} + \Upsilon,
\end{aligned}$$

where the first inequality follows from the definition of W and Υ and the second inequality follows from the standard symmetrization technique used to relate the Rademacher complexity to the uniform deviation of empirical averages from their expectation; see Bartlett et al. (2005). The third inequality comes from a direct application of Lemma A.5 with $\phi(x) = x^2$ (with Lipschitz constant $4b$ on $[-2b, 2b]$), and the last inequality uses Jensen's inequality together with the definition of $\mathfrak{R}(\mathcal{F})$ and the fact that $\frac{1}{N_t^2} \leq \frac{1}{nN_t}$. Substituting the previous inequality on Σ^2 back into (A.7) gives

$$\log \mathbb{E}e^{\lambda(W/b')} \leq \frac{\lambda}{b'(1-\lambda b')}\Big[\frac{16b\mathfrak{R}(\mathcal{F})}{nT} + \Upsilon\Big], \quad \forall \lambda \in (0, 1/b'). \tag{A.8}$$

Step 3. Controlling the term Υ. Note that Υ can be upper-bounded as

$$\begin{aligned}
\Upsilon :&= \sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\frac{1}{T^2}\sum_{s=1}^{T}\frac{1}{N_s^2}\sum_{j=1}^{N_s}\mathbb{E}[\mathbb{E}f_s(X_s^j) - f_s(X_s^j)]^2\Big] \\
&\leq \frac{1}{nT^2}\sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\sum_{s=1}^{T}\mathbb{E}[\mathbb{E}f_s(X_s^1) - f_s(X_s^1)]^2\Big] \\
&\leq \frac{1}{nT^2}\sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\sum_{s=1}^{T}\mathbb{E}[f_s(X_s^1)]^2\Big] \\
&\leq \frac{r}{nT},
\end{aligned} \tag{A.9}$$

where the last inequality follows from the assumption $\frac{1}{T}\sup_{\boldsymbol{f}\in\mathcal{F}}\Big[\sum_{s=1}^{T}\mathbb{E}[f_s(X_s^1)]^2\Big] \leq r$ of the theorem.

Step 4. Transferring the bound on log-moment generating function of Z into tail probabilities. Substituting the bound on $\log \mathbb{E}e^{\lambda W/b'}$ in (A.8) and the bound on Υ in (A.9) back into (A.4) immediately yields the following inequality on the log-moment generating function of Z for any

$\lambda \in (0, 1/2b')$

$$\begin{aligned}
\log \mathbb{E}[e^{\lambda(Z-\mathbb{E}Z)}] &\leq \frac{\lambda b'}{1-\lambda b'}\Big[\frac{\lambda}{b'(1-\lambda b')}\big[16(nT)^{-1}b\mathfrak{R}(\mathcal{F}) + \Upsilon\big] + \frac{\lambda\Upsilon}{b'}\Big] \\
&\leq \frac{\lambda b'}{1-\lambda b'}\frac{\lambda}{b'(1-\lambda b')}\Big[\frac{16b\mathfrak{R}(\mathcal{F})}{nT} + 2\Upsilon\Big] \\
&\leq \frac{2\lambda^2}{2(1-2\lambda b')}\Big[\frac{16b\mathfrak{R}(\mathcal{F})}{nT} + \frac{2r}{nT}\Big],
\end{aligned} \tag{A.10}$$

where the last inequality uses $(1 - \lambda b')^2 \geq 1 - 2\lambda b' > 0$ since $\lambda \in (0, 1/2b')$. That is, the conditions of Lemma A.4 hold and we can apply it (with $A = 2\big[\frac{16b\mathfrak{R}(\mathcal{F})}{nT} + \frac{2r}{nT}\big]$ and $B = 2b'$) to get the following inequality with probability at least $1 - e^{-x}$ (note that $b' = \frac{2b}{nT}$)

$$\begin{aligned}
Z &\leq \mathbb{E}[Z] + \sqrt{4x\Big[\frac{16b\mathfrak{R}(\mathcal{F})}{nT} + \frac{2r}{nT}\Big]} + 2b'x \\
&\leq \mathbb{E}[Z] + 8\sqrt{\frac{bx\mathfrak{R}(\mathcal{F})}{nT}} + \sqrt{\frac{8xr}{nT}} + \frac{4bx}{nT} \\
&\leq \mathbb{E}[Z] + 2\mathfrak{R}(\mathcal{F}) + \frac{8bx}{nT} + \sqrt{\frac{8xr}{nT}} + \frac{4bx}{nT} \\
&\leq 4\mathfrak{R}(\mathcal{F}) + \sqrt{\frac{8xr}{nT}} + \frac{12bx}{nT},
\end{aligned}$$

where the third inequality follows from $2\sqrt{uv} \leq u + v$, and the last step uses the following inequality due to the symmetrization technique (here, the "ghost" sample X' is an *i.i.d.* copy of the initial sample X)

$$\begin{aligned}
\mathbb{E}Z &= \mathbb{E}_X\Big[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{T}\mathbb{E}_{X'}\Big[\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\big(f_t(X_t'^i) - f_t(X_t^i)\big)\Big]\Big] \\
&\leq \mathbb{E}_{X,X'}\Big[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\big(f_t(X_t'^i) - f_t(X_t^i)\big)\Big] \\
&= \mathbb{E}_{X,X',\sigma}\Big[\sup_{\boldsymbol{f}\in\mathcal{F}}\frac{1}{T}\sum_{t=1}^{T}\frac{1}{N_t}\sum_{i=1}^{N_t}\sigma_t^i\big(f_t(X_t'^i) - f_t(X_t^i)\big)\Big] \\
&\leq 2\mathfrak{R}(\mathcal{F}).
\end{aligned}$$

Note that the second identity holds since for any σ_t^i, the random variable $f_t(X_t'^i) - f_t(X_t^i)$ has the same distribution as $\sigma_t^i(f_t(X_t'^i) - f_t(X_t^i))$.

Appendix B. Proofs of the results in Sect. 3

Theorem B.3 is at the core of proving Theorem 9 in Sect. 3. We first present some useful lemmata.

Lemma B.1 *Let $c_1, c_2 > 0$ and $s > q > 0$. Then the equation $x^s - c_1x^q - c_2 = 0$ has a unique positive solution x_0 satisfying*

$$x_0 \leq \Big[c_1^{\frac{s}{s-q}} + \frac{sc_2}{s-q}\Big]^{\frac{1}{s}}.$$

Furthermore, for any $x \geq x_0$, we have $x^s \geq c_1 x^q + c_2$.

Proof Denote $p(x) := x^s - c_1 x^q - c_2$. The uniqueness of a positive solution for the equation $p(x) = 0$ is shown in Lemma 7.2 in Cucker and Zhou (2007). Let x_0 be this unique positive solution. Then, it follows from Young's inequality

$$xy \leq p^{-1}x^p + q^{-1}y^q, \quad \forall x, y \geq 0, p, q > 0, p^{-1} + q^{-1} = 1, \tag{B.1}$$

that

$$x_0^s = c_1 x_0^q + c_2 \leq \frac{x_0^{q \cdot \frac{s}{q}}}{\frac{s}{q}} + \frac{c_1^{\frac{s}{s-q}}}{\frac{s}{s-q}} + c_2 = \frac{q}{s} x_0^s + \frac{s-q}{s} c_1^{\frac{s}{s-q}} + c_2,$$

from which we have $x_0^s \leq c_1^{\frac{s}{s-q}} + \frac{sc_2}{s-q}$. The inequality $p(x) \geq 0$ for any $x \geq x_0$ then follows immediately from the facts that $p(x_0) = 0$, $\lim_{x\to\infty} p(x) = \infty$ and the uniqueness of roots for the equation $p(x) = 0$. ■

Also, we will need the following lemma for the second step of the proof of Theorem B.3.

Lemma B.2 *Let $K > 1, r > 0, 0 < \beta \leq 1$ and $B \geq 1$. Assume that $\mathcal{F} = \{\boldsymbol{f} := (f_1, \ldots, f_T)\}$ is a vector-valued (β, B)-Bernstein class of functions. Define the re-scaled version of $\mathcal{F}$ as*

$$\mathcal{F}_r := \left\{ \boldsymbol{f}' = (f_1', \ldots, f_T') : f_t' := \frac{r f_t}{\max(r, V(\boldsymbol{f}))}, \boldsymbol{f} = (f_t, \ldots, f_T) \in \mathcal{F} \right\}. \tag{B.2}$$

If $V_r^+ := \sup_{\boldsymbol{f}' \in \mathcal{F}_r} [P\boldsymbol{f}' - P_n \boldsymbol{f}'] \leq \frac{r^{\frac{1}{\beta}}}{BK}$, then

$$\forall \boldsymbol{f} \in \mathcal{F} \qquad P\boldsymbol{f} \leq \frac{K}{K-\beta} P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}}}{K}. \tag{B.3}$$

Proof We prove (B.3) by considering two cases. Let $\boldsymbol{f}$ be any element in $\mathcal{F}$. If $V(\boldsymbol{f}) \leq r$, then $\boldsymbol{f}' = \boldsymbol{f}$ and the inequality $V_r^+ \leq \frac{r^{\frac{1}{\beta}}}{BK}$ leads to

$$P\boldsymbol{f} \leq P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}}}{BK} \leq \frac{K}{K-\beta} P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}}}{K}. \tag{B.4}$$

If $V(\boldsymbol{f}) \geq r$, then $\boldsymbol{f}' = r\boldsymbol{f}/V(\boldsymbol{f})$ and the inequality $V_r^+ \leq \frac{r^{\frac{1}{\beta}}}{BK}$ yields

$$\begin{aligned} P\boldsymbol{f} &\leq P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}-1} V(\boldsymbol{f})}{BK} \leq P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}-1} (P\boldsymbol{f})^\beta}{K} \\ &\overset{\text{(B.1)}}{\leq} P_n \boldsymbol{f} + \frac{1}{K} \frac{[(P\boldsymbol{f})^\beta]^{\frac{1}{\beta}}}{\frac{1}{\beta}} + \frac{1}{K} \frac{(r^{\frac{1}{\beta}-1})^{\frac{1}{1-\beta}}}{\frac{1}{1-\beta}} \\ &= P_n \boldsymbol{f} + \frac{\beta}{K} P\boldsymbol{f} + \frac{(1-\beta) r^{\frac{1}{\beta}}}{K}, \end{aligned}$$

where we have used Bernstein's condition $V(\boldsymbol{f}) \leq BP(\boldsymbol{f})^\beta$. The previous inequality can be equivalently written as

$$P\boldsymbol{f} \leq \frac{K}{K-\beta} P_n \boldsymbol{f} + \frac{1-\beta}{K-\beta} r^{\frac{1}{\beta}} \leq \frac{K}{K-\beta} P_n \boldsymbol{f} + \frac{r^{\frac{1}{\beta}}}{K}. \tag{B.5}$$

Eq. (B.3) follows by combining (B.4) and (B.5). ■

Theorem B.3 (LRC-based bounds for MTL) *Let $\mathcal{F} = \{\boldsymbol{f} := (f_1, \ldots, f_T) : \forall t, f_t \in \mathbb{R}^{\mathcal{X}}\}$ be a class of vector-valued functions satisfying $\max_{t\in\mathbb{N}_T} \sup_{x\in\mathcal{X}} |f_t(x)| \leq b$. Let $X := (X_t^i, Y_t^i)_{(t,i)=(1,1)}^{(T,n)}$ be a vector of nT independent random variables where $(X_t^1, Y_t^1), \ldots, (X_t^n, Y_t^n), \forall t \in \mathbb{N}_T$ are identically distributed. Assume that $\mathcal{F}$ is a (β, B)-Bernstein class of vector-valued functions with $0 < \beta \leq 1$ and $B \geq 1$. Let ψ be a sub-root function with fixed point r^*. If $B\mathfrak{R}(\mathcal{F}, r) \leq \psi(r), \forall r \geq r^*$, then, for any $K > 1$, and $x > 0$, with probability at least $1 - e^{-x}$, every $\boldsymbol{f} \in \mathcal{F}$ satisfies*

$$P\boldsymbol{f} \leq \frac{K}{K-\beta} P_n \boldsymbol{f} + (2K)^{\frac{\beta}{2-\beta}} 20^{\frac{2}{2-\beta}} \max\left((r^*)^{\frac{1}{2-\beta}}, (r^*)^{\frac{1}{\beta}}\right) + \left(\frac{2^{\beta+3} B^2 K^\beta x}{nT}\right)^{\frac{1}{2-\beta}} + \frac{24Bbx}{(2-\beta)nT}. \tag{B.6}$$

Proof Let $r \geq r^*$ be a fixed real number. Here, we use the vector-valued function class $\mathcal{F}_r$ as defined in (B.2). The proof is broken down into two major steps. The first step applies Theorem 1 and the "peeling" technique (Van De Geer, 1987; Van Der Vaart and Wellner, 1996) to establish an inequality on the uniform deviation over the function class $\mathcal{F}_r$. The second step then uses the Bernstein assumption $V(\boldsymbol{f}) \leq B(P\boldsymbol{f})^\beta$ to convert this inequality stated for $\mathcal{F}_r$ to a uniform deviation inequality for $\mathcal{F}$.

Step 1. Controlling uniform deviations for $\mathcal{F}_r$. To apply Theorem 1 to $\mathcal{F}_r$, we need to control the variances and uniform bounds for elements in $\mathcal{F}_r$. We first show that $P\boldsymbol{f}'^2 \leq r, \forall \boldsymbol{f}' \in \mathcal{F}_r$. Indeed, for any $\boldsymbol{f} \in \mathcal{F}$ with $V(\boldsymbol{f}) \leq r$, the definition of $\mathcal{F}_r$ implies $f_t' = f_t$ and, hence, $P\boldsymbol{f}'^2 = P\boldsymbol{f}^2 \leq V(\boldsymbol{f}) \leq r$. Otherwise, if $V(\boldsymbol{f}) \geq r$, then $f_t' = r f_t / V(\boldsymbol{f})$ and we get

$$P\boldsymbol{f}'^2 = \frac{1}{T} \sum_{t=1}^T P f_t'^2 = \frac{r^2}{\left[V(\boldsymbol{f})\right]^2} \left(\frac{1}{T} \sum_{t=1}^T P f_t^2\right) \leq \frac{r^2}{\left[V(\boldsymbol{f})\right]^2} V(\boldsymbol{f}) \leq r.$$

Therefore, $\frac{1}{T} \sup_{\boldsymbol{f}' \in \mathcal{F}_r} \sum_{t=1}^T \mathbb{E}[f_t'(X_t)]^2 \leq r$. Also, since functions in $\mathcal{F}$ admit a range of $[-b, b]$ and since $0 \leq r / \max(r, V(\boldsymbol{f})) \leq 1$, it holds that $\max_{t\in\mathbb{N}_T} \sup_{x\in\mathcal{X}} |f_t'(x)| \leq b$ for any $\boldsymbol{f}' \in \mathcal{F}_r$. Applying Theorem 1 to the function class $\mathcal{F}_r$ then yields the following inequality with probability at least $1 - e^{-x}, \forall x > 0$

$$\sup_{\boldsymbol{f}' \in \mathcal{F}_r} [P\boldsymbol{f}' - P_n \boldsymbol{f}'] \leq 4\mathfrak{R}(\mathcal{F}_r) + \sqrt{\frac{8xr}{nT}} + \frac{12bx}{nT}. \tag{B.7}$$

It remains to control the Rademacher complexity of $\mathcal{F}_r$. Denote $\mathcal{F}(u, v) := \left\{\boldsymbol{f} \in \mathcal{F} : u \leq V(\boldsymbol{f}) \leq v\right\}, \forall 0 \leq u \leq v$, and introduce

$$\mathfrak{R}_n \boldsymbol{f}' := \frac{1}{nT} \sum_{t=1}^T \sum_{i=1}^n \sigma_t^i f_t'(X_t^i), \qquad \mathfrak{R}_n(\mathcal{F}_r) := \sup_{\boldsymbol{f}' \in \mathcal{F}_r} \left[\mathfrak{R}_n \boldsymbol{f}'\right].$$

Note that $\mathfrak{R}(\mathcal{F}_r) = \mathbb{E}\mathfrak{R}_n(\mathcal{F}_r)$. Our assumption implies that $V(\boldsymbol{f}) \leq B(P\boldsymbol{f})^\beta \leq Bb^\beta, \forall \boldsymbol{f} \in \mathcal{F}$. Fix $\lambda > 1$ and define k as the smallest integer such that $r\lambda^{k+1} \geq Bb^\beta$. Then, according to the union bound inequality

$$\mathfrak{R}(\mathcal{G}_1 \cup \mathcal{G}_2) \leq \mathfrak{R}(\mathcal{G}_1) + \mathfrak{R}(\mathcal{G}_2), \tag{B.8}$$

we obtain

$$\begin{aligned}
\mathfrak{R}(\mathcal{F}_r) &= \mathbb{E}\Big[\sup_{\boldsymbol{f}' \in \mathcal{F}_r} \mathfrak{R}_n \boldsymbol{f}'\Big] = \mathbb{E}\Big[\sup_{\boldsymbol{f} \in \mathcal{F}} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \frac{r}{\max(r, V(\boldsymbol{f}))}\sigma_t^i f_t(X_t^i)\Big] \\
&\overset{\text{(B.8)}}{\leq} \mathbb{E}\Big[\sup_{\boldsymbol{f} \in \mathcal{F}(0,r)} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \sigma_t^i f_t(X_t^i)\Big] + \mathbb{E}\Big[\sup_{\boldsymbol{f} \in \mathcal{F}(r, Bb^\beta)} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \frac{r}{V(\boldsymbol{f})}\sigma_t^i f_t(X_t^i)\Big] \\
&\overset{\text{(B.8)}}{\leq} \mathbb{E}\Big[\sup_{\boldsymbol{f} \in \mathcal{F}(0,r)} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n} \sigma_t^i f_t(X_t^i)\Big] + \sum_{j=0}^{k} \lambda^{-j}\mathbb{E}\Big[\sup_{\boldsymbol{f} \in \mathcal{F}(r\lambda^j, r\lambda^{j+1})} \mathfrak{R}_n \boldsymbol{f}\Big] \\
&\leq \mathfrak{R}(\mathcal{F}, r) + \sum_{j=0}^{k} \lambda^{-j}\mathfrak{R}\big(\mathcal{F}, r\lambda^{j+1}\big) \\
&\leq \frac{\psi(r)}{B} + \frac{1}{B}\sum_{j=0}^{k} \lambda^{-j}\psi(r\lambda^{j+1}).
\end{aligned}$$

The sub-root property of ψ implies that $\psi(\xi r) \leq \xi^{\frac{1}{2}}\psi(r)$ for any $\xi \geq 1$ and, hence,

$$\mathfrak{R}(\mathcal{F}_r) \leq \frac{\psi(r)}{B}\Big(1 + \sqrt{\lambda}\sum_{j=0}^{k} \lambda^{-\frac{j}{2}}\Big) \leq \frac{\psi(r)}{B}\Big(1 + \frac{\lambda}{\sqrt{\lambda} - 1}\Big).$$

Choosing $\lambda = 4$ in the above inequality implies that $\mathfrak{R}(\mathcal{F}_r) \leq 5\psi(r)/B$, which, together with the inequality $\psi(r) \leq \sqrt{r/r^*}\psi(r^*) = \sqrt{rr^*}, \forall r \geq r^*$, gives

$$\mathfrak{R}(\mathcal{F}_r) \leq \frac{5}{B}\sqrt{rr^*}, \quad \forall r \geq r^*.$$

Combining (B.7) and the above inequality, for any $r \geq r^*$ and $x > 0$, we derive the following inequality with probability at least $1 - e^{-x}$,

$$\sup_{\boldsymbol{f}' \in \mathcal{F}_r} [P\boldsymbol{f}' - P_n\boldsymbol{f}'] \leq \frac{20}{B}\sqrt{rr^*} + \sqrt{\frac{8xr}{nT}} + \frac{12bx}{nT}. \tag{B.9}$$

Step 2. Transferring uniform deviations for $\mathcal{F}_r$ to uniform deviations for $\mathcal{F}$. letting $A := 20\sqrt{r^*}/B + \sqrt{8x/nT}$ and $C := 12bx/nT$, the upper bound of (B.9) can be written as $A\sqrt{r} + C$, that is, $\sup_{\boldsymbol{f}' \in \mathcal{F}_r}[P\boldsymbol{f}' - P_n\boldsymbol{f}'] \leq A\sqrt{r} + C$. Now, according to Lemma B.2, if $\sup_{\boldsymbol{f}' \in \mathcal{F}_r}[P\boldsymbol{f}' - P_n\boldsymbol{f}'] \leq \frac{r^{\frac{1}{\beta}}}{BK}$, then for any $\boldsymbol{f} \in \mathcal{F}$,

$$P\boldsymbol{f} \leq \frac{K}{K - \beta}P_n\boldsymbol{f} + \frac{r^{\frac{1}{\beta}}}{K}.$$

To apply Lemma B.2, we let $A\sqrt{r} + C = r^{\frac{1}{\beta}}/(BK)$. Assume r_0 is the unique positive solution of the equation $A\sqrt{r} + C = r^{\frac{1}{\beta}}/(BK)$, which can be written as

$$r^{\frac{1}{\beta}} - ABKr^{\frac{1}{2}} - BKC = 0.$$

Lemma B.1 then implies

$$\begin{aligned}
r_0^{\frac{1}{\beta}} &\leq (ABK)^{\frac{2}{2-\beta}} + \frac{2BKC}{2 - \beta} \\
&\leq (BK)^{\frac{2}{2-\beta}} 2^{\frac{\beta}{2-\beta}}\Big[(20B^{-1})^{\frac{2}{2-\beta}}(r^*)^{\frac{1}{2-\beta}} + \Big(\frac{8x}{nT}\Big)^{\frac{1}{2-\beta}}\Big] + \frac{24BKbx}{(2-\beta)nT},
\end{aligned} \tag{B.10}$$

where we have used the inequality $(x+y)^p \leq 2^{p-1}(x^p + y^p)$ for any $x, y \geq 0, p \geq 1$. If $r^* \leq r_0$, we can take $r = r_0$ in (B.9) to show that $V_{r_0}^+ \leq A\sqrt{r_0} + C = r_0^{\frac{1}{\beta}}/(BK)$, which, coupled with (B.10) and Lemma B.2, gives

$$P\boldsymbol{f} \leq \frac{K}{K - \beta}P_n\boldsymbol{f} + (2K)^{\frac{\beta}{2-\beta}}20^{\frac{2}{2-\beta}}(r^*)^{\frac{1}{2-\beta}} + \Big(\frac{2^{\beta+3}B^2K^\beta x}{nT}\Big)^{\frac{1}{2-\beta}} + \frac{24Bbx}{(2-\beta)nT}. \tag{B.11}$$

If $r^* > r_0$, Lemma B.1 implies that $A\sqrt{r^*} + C \leq (r^*)^{\frac{1}{\beta}}/(BK)$. We now take $r = r^*$ in (B.9) to get $V_{r^*}^+ \leq A\sqrt{r^*} + C \leq (r^*)^{\frac{1}{\beta}}/(BK)$, from which—via Lemma B.2—we obtain that

$$P\boldsymbol{f} \leq \frac{K}{K - \beta}P_n\boldsymbol{f} + \frac{r_*^{\frac{1}{\beta}}}{K}. \tag{B.12}$$

Note that inequality (B.6) follows immediately by combining (B.11) and (B.12). ■

Proof of Theorem 9

Note that the proof of this theorem relies on the results of Theorem B.3. Introduce the following class of excess loss functions

$$\mathcal{H}_{\mathcal{F}}^* := \{h_{\boldsymbol{f}} = (h_{f_1}, \ldots, h_{f_T}), h_{f_t} : (X_t, Y_t) \mapsto \ell(f_t(X_t), Y_t) - \ell(f_t^*(X_t), Y_t), \boldsymbol{f} \in \mathcal{F}\}. \tag{B.13}$$

It can be shown that

$$\max_{t \in \mathbb{N}_T}\sup_{x \in \mathcal{X}}|h_{f_t}(x, y)| = \max_{t \in \mathbb{N}_T}\sup_{x \in \mathcal{X}}|\ell(f_t(x), y) - \ell(f_t^*(x), y)| \leq L\max_{t \in \mathbb{N}_T}\sup_{x \in \mathcal{X}}|f_t(x) - f_t^*(x)| \leq 2Lb.$$

Also, Assumption 8 implies that

$$P(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*})^2 \leq L^2P(\boldsymbol{f} - \boldsymbol{f}^*)^2 \leq B'L^2\big(P(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*})\big)^\beta, \quad \forall h_{\boldsymbol{f}} \in \mathcal{H}_{\mathcal{F}}^*,$$

By letting $B := \max(B'L^2, 1)$, we have for all $h_{\boldsymbol{f}} \in \mathcal{H}_{\mathcal{F}}^*$,

$$Ph_{\boldsymbol{f}}^2 \leq V(h_{\boldsymbol{f}}) := L^2P(\boldsymbol{f} - \boldsymbol{f}^*)^2 \leq B\big(P(\ell_{\boldsymbol{f}} - \ell_{\boldsymbol{f}^*})\big)^\beta = B(Ph_{\boldsymbol{f}})^\beta.$$

which implies that $\mathcal{H}^*_{\mathcal{F}}$ is a (β, B)-Bernstein class of vector-valued functions. In addition, for any $r \geq r^*$, one can verify that

$$\begin{aligned} B\mathfrak{R}(\mathcal{H}^*_{\mathcal{F}}, r) &= B\mathbb{E}_{X,\sigma}\left[\sup_{V(h_{\boldsymbol{f}})\leq r, \boldsymbol{f}\in\mathcal{F}} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i h_{f_t}(X_t^i, Y_t^i)\right] \\ &= B\mathbb{E}_{X,\sigma}\left[\sup_{V(h_{\boldsymbol{f}})\leq r, \boldsymbol{f}\in\mathcal{F}} \frac{1}{nT}\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i \ell_{f_t}(X_t^i, Y_t^i)\right] \\ &\leq BL\mathfrak{R}(\mathcal{F}^*, r) \leq \psi(r), \end{aligned}$$

where the second to last inequality is due to Lemma A.5. Applying Theorem B.3 to the function class $\mathcal{H}^*_{\mathcal{F}}$ completes the proof.

Appendix C. Proofs of the results in Sect. 4: "Local Rademacher Complexity Bounds for Norm Regularized MTL Models"

Lemma C.1 *Assume that the conditions of Theorem 11 hold. Then, for ever $\boldsymbol{f} \in \mathcal{F}$,*

*(**a**) $P\boldsymbol{f}^2 \leq r$ implies $1/T\sum_{t=1}^{T}\sum_{j=1}^{\infty}\lambda_t^j\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle^2 \leq r$.*
*(**b**) $\mathbb{E}_{X,\sigma}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2 = \frac{\lambda_t^j}{n}$.*

Proof We first prove part (**a**). Given the eigen-decomposition $\mathbb{E}(\phi(X_t)\otimes\phi(X_t)) = \sum_{j=1}^{\infty}\lambda_t^j \boldsymbol{u}_t\otimes\boldsymbol{u}_t^j$ for each task $t \in \mathbb{N}_T$, we obtain

$$\begin{aligned} P\boldsymbol{f}^2 =& \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\langle \boldsymbol{w}_t, \phi(X_t)\rangle\right)^2 = \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left(\langle \boldsymbol{w}_t\otimes\boldsymbol{w}_t, \phi(X_t)\otimes\phi(X_t)\rangle\right) \\ =& \frac{1}{T}\sum_{t=1}^{T}\langle \boldsymbol{w}_t\otimes\boldsymbol{w}_t, \mathbb{E}_X\left(\phi(X_t)\otimes\phi(X_t)\right)\rangle = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{\infty}\lambda_t^j\left\langle \boldsymbol{w}_t\otimes\boldsymbol{w}_t, \boldsymbol{u}_t^j\otimes\boldsymbol{u}_t^j\right\rangle \\ =& \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{\infty}\lambda_t^j\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=1}^{\infty}\lambda_t^j\left\langle \boldsymbol{w}_t, \boldsymbol{u}_t^j\right\rangle^2 \leq r. \end{aligned}$$

Now, we turn to part (**b**). From the independence among the elements of the sequence $\left\{\sigma_t^i\right\}_{\substack{i\in\mathbb{N}_n \\ t\in\mathbb{N}_T}}$, it follows that

$$\begin{aligned} &\mathbb{E}_{X,\sigma}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2 = \frac{1}{n^2}\mathbb{E}_{X,\sigma}\sum_{i,k=1}^{n}\sigma_t^i\sigma_t^k\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\left\langle \phi(X_t^k), \boldsymbol{u}_t^j\right\rangle \\ &\overset{\sigma_t^{i.i.d.}}{=} \frac{1}{n^2}\mathbb{E}_X\left(\sum_{i=1}^{n}\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right) = \frac{1}{n}\left\langle \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_X\left(\phi(X_t^i)\otimes\phi(X_t^i)\right), \boldsymbol{u}_t^j\otimes\boldsymbol{u}_t^j\right\rangle \\ &= \frac{1}{n}\sum_{l=1}^{\infty}\lambda_t^l\left\langle \boldsymbol{u}_t^l\otimes\boldsymbol{u}_t^l, \boldsymbol{u}_t^j\otimes\boldsymbol{u}_t^j\right\rangle = \frac{\lambda_t^j}{n}. \end{aligned}$$

■

The next lemmata are used in the proof of the LRC bound for the $L_{2,q}$-group norm regularized MTL in Corollary 13.

Lemma C.2 (Khintchine-Kahane Inequality in (Peshkir and Shiryaev, 1995)) *Let $\mathcal{H}$ be an inner-product space with induced norm $\|\cdot\|_{\mathcal{H}}$, $v_1, \ldots, v_M \in \mathcal{H}$ and $\sigma_1, \ldots, \sigma_n$ i.i.d. Rademacher random variables. Then, for any $p \geq 1$, we have that*

$$\mathbb{E}_{\boldsymbol{\sigma}}\left\|\sum_{i=1}^{n}\sigma_i v_i\right\|_{\mathcal{H}}^p \leq \left(c\sum_{i=1}^{n}\|v_i\|_{\mathcal{H}}^2\right)^{\frac{p}{2}}. \tag{C.1}$$

where $c := \max\{1, p-1\}$. The inequality also holds for p in place of c.

Lemma C.3 (Rosenthal-Young Inequality; Lemma 3 of (Kloft and Blanchard, 2012)) *Let the independent non-negative random variables $X_1, \ldots, X_n$ satisfy $X_i \leq B < +\infty$ almost surely for all $i = 1, \ldots, n$. If $q \geq \frac{1}{2}$, $c_q := (2qe)^q$, then it holds*

$$\mathbb{E}\left(\frac{1}{n}\sum_{i=1}^{n}X_i\right)^q \leq c_q\left[\left(\frac{B}{n}\right)^q + \left(\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}X_i\right)^q\right]. \tag{C.2}$$

Proof of Lemma 12

For the group norm regularizer $\|\boldsymbol{W}\|_{2,q}$, we can further bound the expectation term in (15) for $\boldsymbol{D} = \boldsymbol{I}$ as follows

$$\begin{aligned} \mathbb{E} :=& \mathbb{E}_{X,\sigma}\left\|\left(\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right)_{t=1}^{T}\right\|_{2,q^*} \\ =& \mathbb{E}_{X,\sigma}\left(\sum_{t=1}^{T}\left\|\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right\|^{q^*}\right)^{\frac{1}{q^*}} \\ \overset{\text{Jensen}}{\leq}& \mathbb{E}_X\left(\sum_{t=1}^{T}\mathbb{E}_\sigma\left\|\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right\|^{q^*}\right)^{\frac{1}{q^*}} \\ \overset{\text{(C.1)}}{\leq}& \mathbb{E}_X\left(\sum_{t=1}^{T}\left(q^*\sum_{i=1}^{n}\left\|\sum_{j>h_t}\left\langle \frac{1}{n}\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\boldsymbol{u}_t^j\right\|^2\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}} \\ =& \sqrt{\frac{q^*}{n}}\mathbb{E}_X\left(\sum_{t=1}^{T}\left(\sum_{j>h_t}\frac{1}{n}\sum_{i=1}^{n}\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{q^*}{2}}\right)^{\frac{1}{q^*}} \end{aligned}$$

$$\overset{\text{Jensen}}{\leq} \sqrt{\frac{q^*}{n}} \left(\sum_{t=1}^{T} \mathbb{E}_X \left(\sum_{j>h_t} \frac{1}{n} \sum_{i=1}^{n} \left\langle \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle^2 \right)^{\frac{q^*}{2}} \right)^{\frac{1}{q^*}}. \tag{C.3}$$

Note that, for $q \leq 2$, it holds that $q^*/2 \geq 1$. Therefore, we cannot employ Jensen's inequality to move the expectation operator inside the inner term and, instead, we need to apply the Rosenthal-Young (R+Y) inequality (see Lemma C.3), which yields

$$\begin{aligned}
\mathbb{E} &\overset{\text{R+Y}}{\leq} \sqrt{\frac{q^*}{n}} \left(\sum_{t=1}^{T} (eq^*)^{\frac{q^*}{2}} \left(\left(\frac{\mathcal{K}}{n} \right)^{\frac{q^*}{2}} + \left(\sum_{j>h_t} \frac{1}{n} \sum_{i=1}^{n} \mathbb{E}_X \left\langle \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle^2 \right)^{\frac{q^*}{2}} \right) \right)^{\frac{1}{q^*}} \\
&= \sqrt{\frac{q^*}{n}} \left(\sum_{t=1}^{T} (eq^*)^{\frac{q^*}{2}} \left(\left(\frac{\mathcal{K}}{n} \right)^{\frac{q^*}{2}} + \left(\sum_{j>h_t} \lambda_t^j \right)^{\frac{q^*}{2}} \right) \right)^{\frac{1}{q^*}}.
\end{aligned} \tag{C.4}$$

The last quantity can be further bounded using the sub-additivity of $\sqrt[q^*]{\cdot}$ as shown next

$$\begin{aligned}
\mathbb{E} &\leq q^* \sqrt{\frac{e}{n}} \left[\left(T \left(\frac{\mathcal{K}}{n} \right)^{\frac{q^*}{2}} \right)^{\frac{1}{q^*}} + \left(\sum_{t=1}^{T} \left(\sum_{j>h_t} \lambda_t^j \right)^{\frac{q^*}{2}} \right)^{\frac{1}{q^*}} \right] \\
&= q^* \sqrt{\frac{e}{n}} \left[T^{\frac{1}{q^*}} \sqrt{\frac{\mathcal{K}}{n}} + \left\| \left(\sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}^{\frac{1}{2}} \right] \\
&= \frac{\sqrt{\mathcal{K}e} q^* T^{\frac{1}{q^*}}}{n} + \sqrt{\frac{eq^{*2}}{n} \left\| \left(\sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}}.
\end{aligned} \tag{C.5}$$

Proof of Corollary 13

Combining (14), (15) and Lemma 12 provides the next bound on $\mathfrak{R}(\mathcal{F}_q, r)$

$$\begin{aligned}
\mathfrak{R}(\mathcal{F}_q, r) &\leq \sqrt{\frac{r \sum_{t=1}^{T} h_t}{nT}} + \sqrt{\frac{2eq^{*2} R_{max}^2}{nT^2} \left\| \left(\sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT} \\
&\overset{(\star)}{\leq} \sqrt{\frac{2}{nT} \left(r \sum_{t=1}^{T} h_t + \frac{2eq^{*2} R_{max}^2}{T} \left\| \left(\sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}} \right)} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT}
\end{aligned} \tag{C.6}$$

$$\begin{aligned}
&\overset{(\star\star)}{\leq} \sqrt{\frac{2}{nT} \left(rT^{1-\frac{2}{q^*}} \left\| (h_t)_{t=1}^{T} \right\|_{\frac{q^*}{2}} + \frac{2eq^{*2} R_{max}^2}{T} \left\| \left(\sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}} \right)} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT} \\
&\overset{(\star\star\star)}{\leq} \sqrt{\frac{4}{nT} \left\| \left(rT^{1-\frac{2}{q^*}} h_t + \frac{2eq^{*2} R_{max}^2}{T} \sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT},
\end{aligned}$$

where in steps $(\star)$, $(\star\star)$ and $(\star\star\star)$ we applied the corresponding inequalities shown next, which hold for all non-negative numbers α_1 and α_2, any non-negative vectors $\boldsymbol{a}_1, \boldsymbol{a}_2 \in \mathbb{R}^T$, any p, q such that $0 \leq q \leq p \leq \infty$ and any $s \geq 1$.

$(\star) \sqrt{\alpha_1} + \sqrt{\alpha_2} \leq \sqrt{2(\alpha_1 + \alpha_2)}$

$(\star\star) \quad l_p - to - l_q: \quad \|\boldsymbol{a}_1\|_q = \langle \mathbf{1}, \boldsymbol{a}_1^q \rangle^{\frac{1}{q}} \overset{\text{Hölder}}{\leq} \left(\|\mathbf{1}\|_{(p/q)^*} \|\boldsymbol{a}_1^q\|_{(p/q)} \right)^{\frac{1}{q}} = T^{\frac{1}{q}-\frac{1}{p}} \|\boldsymbol{a}_1\|_p$

$(\star\star\star) \quad \|\boldsymbol{a}_1\|_s + \|\boldsymbol{a}_2\|_s \leq 2^{1-\frac{1}{s}} \|\boldsymbol{a}_1 + \boldsymbol{a}_2\|_s \leq 2 \|\boldsymbol{a}_1 + \boldsymbol{a}_2\|_s.$

Since inequality $(\star\star\star)$ holds for any non-negative h_t, it follows that

$$\begin{aligned}
\mathfrak{R}(\mathcal{F}_q, r) &\leq \sqrt{\frac{4}{nT} \left\| \left(\min_{h_t \geq 0} rT^{1-\frac{2}{q^*}} h_t + \frac{2eq^{*2} R_{max}^2}{T} \sum_{j>h_t} \lambda_t^j \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT} \\
&\leq \sqrt{\frac{4}{nT} \left\| \left(\sum_{j=1}^{\infty} \min \left(rT^{1-\frac{2}{q^*}}, \frac{2eq^{*2} R_{max}^2}{T} \lambda_t^j \right) \right)_{t=1}^{T} \right\|_{\frac{q^*}{2}}} + \frac{\sqrt{2\mathcal{K}e} R_{max} q^* T^{\frac{1}{q^*}}}{nT}.
\end{aligned}$$

Proof of Theorem 17

By considering the hypothesis space in (16) and the MT-LRC's definition, we have

$$\begin{aligned}
\mathfrak{R}(\mathcal{F}_{q,R_{max},T}, r) &= \frac{1}{T} \mathbb{E}_{X,\sigma} \left\{ \sup_{\substack{Pf^2 \leq r, \\ \|\boldsymbol{W}\|_{2,q}^2 \leq 2R_{max}^2}} \sum_{t=1}^{T} \left\langle \boldsymbol{w}_t, \frac{1}{n} \sum_{i=1}^{n} \sigma_t^i \phi(X_t^i) \right\rangle \right\} \\
&= \frac{1}{T} \mathbb{E}_{X,\sigma} \left\{ \sup_{\substack{1/T \sum_{t=1}^{T} \mathbb{E}\langle \boldsymbol{w}_t, \phi(X_t) \rangle^2 \leq r, \\ \|\boldsymbol{W}\|_{2,q}^2 \leq 2R_{max}^2}} \sum_{t=1}^{T} \left\langle \boldsymbol{w}_t, \frac{1}{n} \sum_{i=1}^{n} \sigma_t^i \phi(X_t^i) \right\rangle \right\} \\
&\geq \frac{1}{T} \mathbb{E}_{X,\sigma} \left\{ \sup_{\substack{\forall t\, \mathbb{E}_X \langle \boldsymbol{w}_t, \phi(X_t) \rangle^2 \leq r, \\ \|\boldsymbol{W}\|_{2,q}^2 \leq 2R_{max}^2, \\ \|\boldsymbol{w}_1\|_2 = \ldots = \|\boldsymbol{w}_t\|_2}} \sum_{t=1}^{T} \left\langle \boldsymbol{w}_t, \frac{1}{n} \sum_{i=1}^{n} \sigma_t^i \phi(X_t^i) \right\rangle \right\}
\end{aligned}$$

$$= \frac{1}{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\substack{\forall t\, \mathbb{E}_X\langle \boldsymbol{w}_t,\phi(X_t)\rangle^2\le r,\\ \forall t\, \|\boldsymbol{w}_t\|_2^2\le 2R_{max}^2 T^{-\frac{2}{q}}}} \sum_{t=1}^{T}\left\langle \boldsymbol{w}_t, \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i)\right\rangle\right\}$$

$$= \frac{1}{T}\sum_{t=1}^{T}\mathbb{E}_{X,\sigma}\left\{\sup_{\substack{\forall t\, \mathbb{E}_X\langle \boldsymbol{w}_t,\phi(X_t)\rangle^2\le r,\\ \forall t\, \|\boldsymbol{w}_t\|_2^2\le 2R_{max}^2 T^{-\frac{2}{q}}}} \left\langle \boldsymbol{w}_t, \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i)\right\rangle\right\}$$

$$= \mathbb{E}_{X,\sigma}\left\{\sup_{\substack{\mathbb{E}_X\langle \boldsymbol{w}_1,\phi(X_1)\rangle^2\le r,\\ \|\boldsymbol{w}_1\|_2^2\le 2R_{max}^2 T^{-\frac{2}{q}}}} \left\langle \boldsymbol{w}_1, \frac{1}{n}\sum_{i=1}^{n}\sigma_1^i\phi(X_1^i)\right\rangle\right\}$$

$$= \mathfrak{R}(\mathcal{F}_{1,R_{max}T^{-\frac{1}{q}},1}, r).$$

According to Mendelson (2003), it can be shown that there is a constant c such that if $\lambda_t^1 \ge \frac{1}{nR_{max}^2}$, then, for all $r \ge \frac{1}{n}$, it holds that $\mathfrak{R}(\mathcal{F}_{1,R_{max}T^{-\frac{1}{q}},1}, r) \ge \sqrt{\frac{c}{n}\sum_{j=1}^{\infty}\min\left(r, R_{max}^2 T^{-\frac{2}{q}}\lambda_1^j\right)}$, which provides the desired result after some algebraic manipulations. The following lemma is used in the proof of the LRC bounds for the L_{S_q}-Schatten norm regularized MTL in Corollary 19.

Lemma C.4 (Khintchine's inequality for arbitrary matrices in Tomczak-Jaegermann (1974)) *Let $\boldsymbol{Q}_1, \ldots, \boldsymbol{Q}_n$ be a set of arbitrary $m \times n$ matrices and let $\sigma_1, \ldots, \sigma_n$ be a sequence of independent Bernoulli random variables. Then for all $p \ge 2$,*

$$\mathbb{E}_\sigma\left[\left\|\sum_{i=1}^{n}\sigma_i\boldsymbol{Q}_i\right\|_{S_p}^{p}\right] \le p^{p/2}\left(\sum_{i=1}^{n}\|\boldsymbol{Q}_i\|_{S_p}^2\right)^{p/2}. \tag{C.7}$$

Proof of Corollary 19

In order to find an LRC bound for an L_{S_q}-Schatten norm regularized hypothesis space of (26), one just needs to bound the expectation term in (11). Define $\boldsymbol{U}_t^i$ as the matrix with T columns, whose only non-zero t^{th} column equals $\sum_{j>h_t}\left\langle \frac{1}{n}\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j$. Recall that, for the Schatten norm regularized hypothesis space of (26), it holds that $\boldsymbol{D} = \boldsymbol{I}$. Therefore, we will have that

$$\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_* = \mathbb{E}_{X,\sigma}\left\|\left(\sum_{j>h_t}\left\langle \frac{1}{n}\sum_{i=1}^{n}\sigma_t^i\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right)_{t=1}^{T}\right\|_{S_{q^*}}$$

$$= \mathbb{E}_{X,\sigma}\left\|\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i\boldsymbol{U}_t^i\right\|_{S_{q^*}} \overset{\text{Jensen}}{\le} \mathbb{E}_X\left\{\mathbb{E}_\sigma\left\|\sum_{t=1}^{T}\sum_{i=1}^{n}\sigma_t^i\boldsymbol{U}_t^i\right\|_{S_{q^*}}^{q^*}\right\}^{\frac{1}{q^*}}$$

$$\overset{\text{(C.7)}}{\le} \mathbb{E}_X\left\{(q^*)^{q^*/2}\left(\sum_{t=1}^{T}\sum_{i=1}^{n}\|\boldsymbol{U}_t^i\|_{S_{q^*}}^2\right)^{q^*/2}\right\}^{\frac{1}{q^*}}$$

$$= \sqrt{q^*}\mathbb{E}_X\left(\sum_{t=1}^{T}\sum_{i=1}^{n}\left\|\sum_{j>h_t}\left\langle \frac{1}{n}\phi(X_t^i), \boldsymbol{u}_t^j\right\rangle \boldsymbol{u}_t^j\right\|^2\right)^{1/2}$$

$$= \sqrt{q^*}\mathbb{E}_X\left(\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j>h_t}\frac{1}{n^2}\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right)^{1/2}$$

$$\overset{\text{Jensen}}{\le} \frac{\sqrt{q^*}}{n}\left(\sum_{t=1}^{T}\sum_{i=1}^{n}\sum_{j>h_t}\lambda_t^j\right)^{\frac{1}{2}} = \sqrt{\frac{q^*}{n}\left\|\left(\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_1}. \tag{C.8}$$

Proof of Corollary 22

Similar to the proof of Corollary 19, for the graph regularized hypothesis space depicted in (28), one can bound the expectation term in (11) in this manner

$$\mathbb{E}_{X,\sigma}\left\|\boldsymbol{D}^{-1/2}\boldsymbol{V}\right\|_* = \mathbb{E}_{X,\sigma}\left[\mathrm{tr}\left(\boldsymbol{V}^T\boldsymbol{D}^{-1}\boldsymbol{V}\right)\right]^{\frac{1}{2}}$$

$$\overset{\text{Jensen}}{\le} \mathbb{E}_X\left(\frac{1}{n^2}\sum_{t,s=1}^{T,T}\sum_{i,l=1}^{n,n}\sum_{j>h_t}\sum_{k>h_s}\boldsymbol{D}_{st}^{-1}\mathbb{E}_\sigma\left(\sigma_t^i\sigma_s^l\right)\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle\left\langle \phi(X_s^l), \boldsymbol{u}_s^k\right\rangle\left\langle \boldsymbol{u}_t^j, \boldsymbol{u}_s^k\right\rangle\right)^{\frac{1}{2}}$$

$$= \mathbb{E}_X\left(\frac{1}{n}\sum_{t=1}^{T}\boldsymbol{D}_{tt}^{-1}\sum_{j>h_t}\frac{1}{n}\sum_{i=1}^{n}\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{1}{2}}$$

$$\overset{\text{Jensen}}{\le} \left(\frac{1}{n}\sum_{t=1}^{T}\boldsymbol{D}_{tt}^{-1}\sum_{j>h_t}\frac{1}{n}\sum_{i=1}^{n}\mathbb{E}_X\left\langle \phi(X_t^i), \boldsymbol{u}_t^j\right\rangle^2\right)^{\frac{1}{2}}$$

$$= \frac{1}{\sqrt{n}}\left(\sum_{t=1}^{T}\sum_{j>h_t}\boldsymbol{D}_{tt}^{-1}\lambda_t^j\right)^{\frac{1}{2}} = \sqrt{\frac{1}{n}\left\|\left(\boldsymbol{D}_{tt}^{-1}\sum_{j>h_t}\lambda_t^j\right)_{t=1}^{T}\right\|_1}. \tag{C.9}$$

The remainder of the derivation is similar to that of Corollary 13 and is omitted for brevity.

Appendix D. Proof of the results in Sect. 6: "Discussion"

In what follows, we provide some general results that imply Theorem 26. More specifically, we restate two concentration results for sums of non-negative operators with finite-dimensional ranges. Towards this end, we will say that two operators A and B are related as $A \preceq B$, if $B - A$ is a positive semi-definite operator.

Theorem D.1 (Theorem A.3 in Maurer and Pontil (2016)) *Consider the separable Hilbert space $\mathcal{H}$. Let $\mathcal{M} \subseteq \mathcal{H}$ be a subspace of finite dimension d. Also, consider the finite sequence A_k of random, independent, self-adjoint operators on $\mathcal{H}$. Assume that, for all $m \in \mathbb{N}$, $k \in \mathbb{N}_N$ and some $R \geq 0$, it holds that $A_k \succeq 0$, Ran$(A_k) \subseteq M$ almost surely and*

$$\mathbb{E}A_k^m \preceq m! R^{m-1} \mathbb{E}A_k.$$

Then,

$$\sqrt{\mathbb{E}\|\sum_k A_k\|_{S_\infty}} \leq \sqrt{\|\mathbb{E}\sum_k A_k\|_{S_\infty}} + \sqrt{R\left(\ln \dim(M) + 1\right)}. \tag{D.1}$$

Lemma D.2 (Lemma A.4 in Maurer and Pontil (2016)) *Let $a_1, \ldots, a_n \in \mathbb{R}^d$. Let*

$$\alpha := \sum_{i=1}^n \|a_i\|^2,$$

and define a rank-one operator Q_x on H, such that $Q_x v = \langle v, x\rangle x$. Also, let $D := \sum_{i=1}^n \sigma_i a_i$. Then, for any $p \geq 1$, it holds that

$$\mathbb{E}[(Q_D)^p] \preceq (2p-1)!! \alpha^{p-1} \mathbb{E}[Q_D],$$

where $(2p-1)!! := \prod_{i=1}^p (2i-1) = (2p-1)(2(p-1)-1) \times \ldots \times 5 \times 3 \times 1$.

Theorem D.3 (Theorem 7 in Maurer and Pontil (2013)) *Consider the independent random operators $A_1, \ldots, A_N$, which satisfy $0 \preceq A_k \preceq I$, $\forall k$. Also, assume that, for some $d \in \mathbb{N}$, it holds that*

$$\text{dimSpan}(\text{Ran}(A_1), \ldots, \text{Ran}(A_N)) \leq d,$$

almost surely. Then

$$\sqrt{\mathbb{E}\|\sum_{k=1}^N A_k\|_{S_\infty}} \leq \sqrt{\|\mathbb{E}\sum_{k=1}^N A_k\|_{S_\infty}} + \sqrt{6\left(\ln\left(4d^2\right) + 1\right)}. \tag{D.2}$$

Proof of Theorem 26

We first proceed to bound $\mathbb{E}_\sigma \|\boldsymbol{V}'\|_{S_\infty}$. Let $\boldsymbol{D}_t$ be the random vector $\sum_{j>h_t} \left\langle \sum_{i=1}^n \sigma_t^i \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle \boldsymbol{u}_t^j$, and recall that the rank-one operator $Q_{\boldsymbol{D}_t}$ is such that $Q_{\boldsymbol{D}_t} v := \langle v, \boldsymbol{D}_t\rangle \boldsymbol{D}_t$. Then, it is clear that $\boldsymbol{V}'^* \boldsymbol{V}' = \sum_{t=1}^T Q_{\boldsymbol{D}_t}$ and, by using Jensen's inequality, we have

$$\mathbb{E}_\sigma \|\boldsymbol{V}'\|_{S_\infty} \leq \sqrt{\mathbb{E}_\sigma \|\sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_\infty}.$$

Note that $\boldsymbol{D}_t$ is the projection of $\sum_{i=1}^n \sigma_t^i \phi(X_t^i)$ onto the space spanned by $\left(\boldsymbol{u}_t^j\right)_{j>h_t}$. Since $\sum_{i=1}^n \sigma_t^i \phi(X_t^i)$ belongs to the space spanned by $\left(\phi(X_t^i)\right)_{i=1}^n$, we know that $\boldsymbol{D}_t$ belongs to a subspace of dimension at most n. It then follows that Ran$(Q_{\boldsymbol{D}_1}), \ldots,$ Ran$(Q_{\boldsymbol{D}_T})$ lie in a subspace of dimension at most nT. Thus, Lemma D.2 for $\alpha_t := \sum_{i=1}^n \|\phi(X_t^i)\|^2$ yields

$$\mathbb{E}_\sigma[(Q_{\boldsymbol{D}_t})^m] \preceq (2m-1)!! \alpha_t^{m-1} \mathbb{E}_\sigma[Q_{\boldsymbol{D}_t}] \preceq m! \left(2 \max_t \alpha_t\right)^{m-1} \mathbb{E}_\sigma[Q_{\boldsymbol{D}_t}],$$

Therefore, applying Theorem D.1 with $R = 2\max_t \alpha_t$ and dimension less than nT gives

$$\sqrt{\mathbb{E}_\sigma \|\sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} \leq \sqrt{\|\mathbb{E}_\sigma \sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} + \sqrt{2\max_t \alpha_t \left(\ln(nT) + 1\right)}.$$

Since $\alpha_t = \sum_{i=1}^n \|\phi(X_t^i)\|^2 \leq n\mathcal{K}$, we get

$$\sqrt{\mathbb{E}_\sigma \|\sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} \leq \sqrt{\|\mathbb{E}_\sigma \sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} + \sqrt{2n\mathcal{K}\left(\ln(nT) + 1\right)}.$$

Now, we define

$$\begin{aligned} B_t := \mathbb{E}_\sigma Q_{\boldsymbol{D}_t} &= \sum_{i=1}^n \sum_{j,j'>h_t} \left\langle \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle \left\langle \phi(X_t^i), \boldsymbol{u}_t^{j'} \right\rangle \boldsymbol{u}_t^i \otimes \boldsymbol{u}_t^{j'} \\ &= \sum_{i=1}^n \left\langle \sum_{j>h_t} \left\langle \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle \boldsymbol{u}_t^j \right\rangle \otimes \left\langle \sum_{j'>h_t} \left\langle \phi(X_t^i), \boldsymbol{u}_t^{j'} \right\rangle \boldsymbol{u}_t^{j'} \right\rangle = \sum_{i=1}^n Q_{\boldsymbol{D}_{t,i}}, \end{aligned}$$

where we introduce $\boldsymbol{D}_{t,i} := \sum_{j>h_t} \left\langle \phi(X_t^i), \boldsymbol{u}_t^j \right\rangle \boldsymbol{u}_t^j$. Note that, in taking the expectation with respect to the random variables X_t^i's, Theorem D.1 cannot be utilized, since the covariance may have infinite rank, that is, we might not be able to find a finite-dimensional subspace, which contains the range of all the $Q_{\boldsymbol{D}_{t,i}}$'s. However, since for all $t \in \mathbb{N}_T$ and $i \in \mathbb{N}_n$, it holds that $\|\boldsymbol{D}_{t,i}\| \leq \sqrt{\mathcal{K}}$, all the $Q_{\boldsymbol{D}_{t,i}}$'s satisfy $0 \preceq Q_{\boldsymbol{D}_{t,i}} \preceq \mathcal{K} I$ and they all are rank-one operators. It then follows that

$$\text{dimSpan}\Big((\text{Ran}(\boldsymbol{B}_1), \ldots, \text{Ran}(\boldsymbol{B}_T))\Big) \leq nT.$$

Therefore, we can apply Theorem D.3 with $d = nT$, which, in conjunction with the Jensen's inequality, yields

$$\begin{aligned} \mathbb{E}_X \mathbb{E}_\sigma \|\boldsymbol{V}'\|_{S_\infty} &\leq \mathbb{E}_X \sqrt{\mathbb{E}_\sigma \|\sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} \\ &\leq \mathbb{E}_X \sqrt{\|\mathbb{E}_\sigma \sum_{t=1}^T Q_{\boldsymbol{D}_t}\|_{S_\infty}} + \sqrt{2n\mathcal{K}\left(\ln(nT) + 1\right)} \\ &\leq \sqrt{\mathbb{E}_X \|\sum_{t=1}^T \boldsymbol{B}_t\|_{S_\infty}} + \sqrt{2n\mathcal{K}\left(\ln(nT) + 1\right)} \\ &\leq \sqrt{\|\mathbb{E}_X \sum_{t=1}^T \boldsymbol{B}_t\|_{S_\infty}} + \sqrt{6\mathcal{K}\left(\ln(4n^2T^2) + 1\right)} + \sqrt{2n\mathcal{K}\left(\ln(nT) + 1\right)}. \end{aligned} \tag{D.3}$$

After some simplifications, we arrive at

$$\mathbb{E}_{X,\sigma} \|\boldsymbol{V}'\|_{S_\infty} \leq \sqrt{\|\mathbb{E}_X \sum_{t=1}^T \boldsymbol{B}_t\|_{S_\infty}} + 6\sqrt{n\mathcal{K}\left(\ln(nT) + 1\right)}. \tag{D.4}$$

Furthermore, it can be shown that

$$\mathbb{E}_X \boldsymbol{B}_t = n \sum_{j>h_t} \lambda_t^j \boldsymbol{u}_t^j \otimes \boldsymbol{u}_t^j.$$

By considering the task-averaged operator $C = 1/T \sum_{t=1}^T J_t = 1/T \sum_{t=1}^T \sum_{j=1}^\infty \lambda_t^j \boldsymbol{u}_t^j \otimes \boldsymbol{u}_t^j$ and choosing $\lambda_h := \max_{t \in \mathbb{N}_T} \{\lambda_t^{h_t}\}$, we get

$$\begin{aligned} \mathbb{E}_{X,\sigma} \|\boldsymbol{V}'\|_{S_\infty} &\leq \sqrt{n \| \sum_{t=1}^T \sum_{j>h_t} \lambda_t^j \boldsymbol{u}_t^j \otimes \boldsymbol{u}_t^j \|_{S_\infty}} + 6\sqrt{n\mathcal{K}\left(\ln(nT)+1\right)} \\ &= \sqrt{nT\lambda_h} + 6\sqrt{n\mathcal{K}\left(\ln(nT)+1\right)}. \end{aligned} \tag{D.5}$$

This last result, combined with (48), provides the LRC bound for the trace norm regularized class $\mathcal{F}'_{S_1}$

$$\mathfrak{R}(\mathcal{F}'_{S_1}, r) \leq \sqrt{\frac{r\sum_{t=1}^T h_t}{nT}} + \sqrt{\frac{2R'^2_{max}\lambda_h}{n}} + 6\sqrt{\frac{2R'^2_{max}\mathcal{K}\left(\ln(nT)+1\right)}{nT}}. \tag{D.6}$$

Finally, using a similar argument as the one in Theorem 24, we get

$$r^* \leq \min_{0 \leq h_t \leq \infty} \left\{ \frac{B^2 \sum_{t=1}^T h_t}{nT} + 4BL\sqrt{\frac{2R'^2_{max}\lambda_h}{n}} + 24BL\sqrt{\frac{2R'^2_{max}\mathcal{K}\left(\ln(nT)+1\right)}{nT}} \right\}. \tag{D.7}$$

References

Qi An, Chunping Wang, Ivo Shterev, Eric Wang, Lawrence Carin, and David B Dunson. Hierarchical kernel stick-breaking process for multi-task image analysis. In *International Conference on Machine learning*, pages 17–24. ACM, 2008.

Rie Kubota Ando and Tong Zhang. A framework for learning predictive structures from multiple tasks and unlabeled data. *The Journal of Machine Learning Research*, 6:1817–1853, 2005.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Multi-task feature learning. In *Advances in Neural Information Processing Systems*, pages 41–48, 2007a.

Andreas Argyriou, Massimiliano Pontil, Yiming Ying, and Charles A Micchelli. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems*, pages 25–32, 2007b.

Andreas Argyriou, Theodoros Evgeniou, and Massimiliano Pontil. Convex multi-task feature learning. *Machine Learning*, 73(3):243–272, 2008a.

Andreas Argyriou, Andreas Maurer, and Massimiliano Pontil. An algorithm for transfer learning in a heterogeneous environment. In *Machine Learning and Knowledge Discovery in Databases*, pages 71–85. Springer, 2008b.

Andreas Argyriou, Stéphan Clémençon, and Ruocong Zhang. Learning the graph of relations among multiple tasks. *ICML workshop on New Learning Frameworks and Models for Big Data*, 2014.

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

Peter L Bartlett, Stéphane Boucheron, and Gábor Lugosi. Model selection and error estimation. *Machine Learning*, 48(1-3):85–113, 2002.

Peter L Bartlett, Shahar Mendelson, and Petra Philips. Local complexities for empirical risk minimization. In *International Conference on Computational Learning Theory*, pages 270–284. Springer, 2004.

Peter L Bartlett, Olivier Bousquet, and Shahar Mendelson. Local rademacher complexities. *Annals of Statistics*, pages 1497–1537, 2005.

Jonathan Baxter. A model of inductive bias learning. *Journal of Artificial Intelligence Research*, 12 (149-198):3, 2000.

Shai Ben-David and Reba Schuller Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine learning*, 73(3):273–287, 2008.

Shai Ben-David and Reba Schuller. Exploiting task relatedness for multiple task learning. In *Learning Theory and Kernel Machines*, pages 567–580. Springer, 2003.

Steffen Bickel, Jasmina Bogojeska, Thomas Lengauer, and Tobias Scheffer. Multi-task learning for hiv therapy screening. In *International Conference on Machine learning*, pages 56–63. ACM, 2008.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. Concentration inequalities using the entropy method. *Annals of Probability*, pages 1583–1614, 2003.

Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

Olivier Bousquet. *Concentration inequalities and empirical processes theory applied to the analysis of learning algorithms*. PhD thesis, Ecole Polytechnique, Paris, 2002.

Bin Cao, Nathan N Liu, and Qiang Yang. Transfer learning for collective link prediction in multiple heterogenous domains. In *International Conference on Machine Learning (ICML-10)*, pages 159–166, 2010.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Corinna Cortes and Mehryar Mohri. Domain adaptation in regression. In *International Conference on Algorithmic Learning Theory*, pages 308–323. Springer, 2011.

Corinna Cortes and Mehryar Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.

Corinna Cortes, Marius Kloft, and Mehryar Mohri. Learning kernels using local rademacher complexity. In *Advances in Neural Information Processing Systems*, pages 2760–2768, 2013.

Felipe Cucker and Ding Xuan Zhou. *Learning theory: an approximation theory viewpoint*. Cambridge University Press, 2007.

Theodoros Evgeniou and Massimiliano Pontil. Regularized multi–task learning. In *ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 109–117, 2004.

Theodoros Evgeniou, Charles A Micchelli, and Massimiliano Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(Apr):615–637, 2005.

Sham Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. On the duality of strong convexity and strong smoothness: Learning applications and matrix regularization. Unpublished Manuscript, 2009.

Sham M Kakade, Shai Shalev-Shwartz, and Ambuj Tewari. Regularization techniques for learning with matrices. *The Journal of Machine Learning Research*, 13(1):1865–1890, 2012.

Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *International Conference on Machine Learning*, pages 521–528, 2011.

Marius Kloft and Gilles Blanchard. The local rademacher complexity of lp-norm multiple kernel learning. In *Advances in Neural Information Processing Systems*, pages 2438–2446, 2011.

Marius Kloft and Gilles Blanchard. On the convergence rate of lp-norm multiple kernel learning. *The Journal of Machine Learning Research*, 13(1):2465–2502, 2012.

V. Koltchinskii and D. Panchenko. Empirical margin distributions and bounding the generalization error of combined classifiers. *The Annals of Statistics*, 30(1):1–50, 02 2002.

Vladimir Koltchinskii. Rademacher penalties and structural risk minimization. *IEEE Transactions on Information Theory*, 47(5):1902–1914, 2001.

Vladimir Koltchinskii. Local rademacher complexities and oracle inequalities in risk minimization. *The Annals of Statistics*, 34(6):2593–2656, 12 2006. doi: 10.1214/009053606000001019.

Vladimir Koltchinskii. Rademacher complexities and bounding the excess risk in active learning. *Journal of Machine Learning Research*, 11:2457–2485, December 2010. ISSN 1532-4435.

Vladimir Koltchinskii and Dmitriy Panchenko. Rademacher processes and bounding the risk of function learning. In *High dimensional probability II*, pages 443–457. Springer, 2000.

Abhishek Kumar and Hal Daume III. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*, 2012.

Yunwen Lei, Lixin Ding, and Yingzhou Bi. Local rademacher complexity bounds based on covering numbers. *arXiv:1510.01463 [cs.AI]*, 2015.

Cong Li, Michael Georgiopoulos, and Georgios C Anagnostopoulos. Multitask classification hypothesis space with improved generalization bounds. *IEEE Transactions on Neural Networks and Learning Systems*, 26(7):1468–1479, 2015.

K Lounici, M Pontil, AB Tsybakov, and SA Van De Geer. Taking advantage of sparsity in multi-task learning. In *Conference on Learning Theory*, 2009.

Yishay Mansour and Mariano Schain. Robust domain adaptation. *Annals of Mathematics and Artificial Intelligence*, 71(4):365–380, 2013.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory*, June 2009a.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *Conference on Uncertainty in Artificial Intelligence*, pages 367–374, 2009b.

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In *Advances in Neural Information Processing Systems*, pages 1041–1048. 2009c.

Andreas Maurer. Bounds for linear multi-task learning. *The Journal of Machine Learning Research*, 7:117–139, January 2006a.

Andreas Maurer. The rademacher complexity of linear transformation classes. In *International Conference on Computational Learning Theory*, pages 65–78. Springer, 2006b.

Andreas Maurer. A chain rule for the expected suprema of gaussian processes. *Theoretical Computer Science*, 650:109–122, 2016.

Andreas Maurer and Massimiliano Pontil. Excess risk bounds for multitask learning with trace norm regularization. In *Conference on Learning Theory*, volume 30, pages 55–76, 2013.

Andreas Maurer and Massimiliano Pontil. Bounds for vector-valued function estimation. *arXiv preprint arXiv:1606.01487*, 2016.

Andreas Maurer, Massimiliano Pontil, and Bernardino Romera-Paredes. The benefit of multitask representation learning. *Journal of Machine Learning Research*, 17(81):1–32, 2016.

Shahar Mendelson. On the performance of kernel classes. *The Journal of Machine Learning Research*, 4:759–771, 2003.

Shahar Mendelson. Learning without concentration. In *Conference on Learning Theory*, pages 25–39, 2014.

Charles A Micchelli and Massimiliano Pontil. Kernels for multi–task learning. In *Advances in Neural Information Processing Systems*, pages 921–928, 2004.

Luca Oneto, Alessandro Ghio, Sandro Ridella, and Davide Anguita. Local rademacher complexity: Sharper risk bounds with and without unlabeled samples. *Neural Networks*, 65:115 – 125, 2015.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

Anastasia Pentina and Shai Ben-David. Multi-task and lifelong learning of kernels. In *Algorithmic Learning Theory*, pages 194–208. Springer, 2015.

Anastasia Pentina and Christoph H Lampert. Lifelong learning with non-iid tasks. In *Advances in Neural Information Processing Systems*, pages 1540–1548, 2015.

G Peshkir and Albert Nikolaevich Shiryaev. The khintchine inequalities and martingale expanding sphere of their action. *Russian Mathematical Surveys*, 50(5):849–904, 1995.

Ting Kei Pong, Paul Tseng, Shuiwang Ji, and Jieping Ye. Trace norm regularization: Reformulations, algorithms, and multi-task learning. *SIAM Journal on Optimization*, 20(6):3465–3489, 2010.

Bernardino Romera-Paredes, Andreas Argyriou, Nadia Berthouze, and Massimiliano Pontil. Exploiting unrelated tasks in multi-task learning. In *International Conference on Artificial Intelligence and Statistics*, pages 951–959, 2012.

Sebastian Thrun and Lorien Pratt. *Learning To Learn*. Springer Science & Business Media, 2012.

I. Tolstikhin, G. Blanchard, and M. Kloft. Localized complexities for transductive learning. In *Conference on Learning Theory*, volume 35, pages 857–884, 2014.

Nicole Tomczak-Jaegermann. The moduli of smoothness and convexity and the rademacher averages of the trace classes s_p $(1 \leq p < \infty)$. *Studia Mathematica*, 2(50):163–182, 1974.

Sara Van De Geer. A new approach to least-squares estimation, with applications. *The Annals of Statistics*, pages 587–602, 1987.

Aad W Van Der Vaart and Jon A Wellner. Weak convergence. In *Weak Convergence and Empirical Processes*, pages 16–28. Springer, 1996.

Christian Widmer, Marius Kloft, and Gunnar Rätsch. Multi-task learning for computational biology: Overview and outlook. In *Empirical Inference*, pages 117–127. Springer, 2013.

Qian Xu, Sinno Jialin Pan, Hannah Hong Xue, and Qiang Yang. Multitask learning for protein subcellular location prediction. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(3):748–759, 2011.

Niloofar Yousefi, Michael Georgiopoulos, and Georgios C Anagnostopoulos. Multi-task learning with group-specific feature space sharing. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 120–136. Springer, 2015.

Chao Zhang, Lei Zhang, and Jieping Ye. Generalization bounds for domain adaptation. In *Advances in Neural Information Processing Systems 25*, pages 3320–3328. 2012.

Yu Zhang and Dit-Yan Yeung. Multi-task warped gaussian process for personalized age estimation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2622–2629, 2010.

Journal of Machine Learning Research 19 (2018) 1-46 Submitted 6/17; Revised 7/18; Published 9/18

State-by-state Minimax Adaptive Estimation for Nonparametric Hidden Markov Models

Luc Lehéricy LUC.LEHERICY@MATH.U-PSUD.FR
Laboratoire de Mathématiques d'Orsay
Univ. Paris-Sud, CNRS, Université Paris-Saclay
91405 Orsay, France

Editor: Animashree Anandkumar

Abstract

In this paper, we introduce a new estimator for the emission densities of a nonparametric hidden Markov model. It is adaptive and minimax with respect to each state's regularity–as opposed to globally minimax estimators, which adapt to the worst regularity among the emission densities. Our method is based on Goldenshluger and Lepski's methodology. It is computationally efficient and only requires a family of preliminary estimators, without any restriction on the type of estimators considered. We present two such estimators that allow to reach minimax rates up to a logarithmic term: a spectral estimator and a least squares estimator. We show how to calibrate it in practice and assess its performance on simulations and on real data.

Keywords: hidden Markov model, model selection, nonparametric density estimation, oracle inequality, adaptive minimax estimation, spectral method, least squares method

1. Introduction

Finite state space hidden Markov models, or HMMs in short, are powerful tools for studying discrete time series and have been used in a variety of applications such as economics, signal processing and image analysis, genomics, ecology, speech recognition and ecology among others. The core idea is that the behaviour of the observations depends on a hidden variable that evolves like a Markov chain.

Formally, a hidden Markov model is a process $(X_j, Y_j)_{j\geq 1}$ in which $(X_j)_j$ is a Markov chain on $\mathcal{X}$, the Y_i's are independent conditionally on $(X_j)_j$ and the conditional distribution of Y_i given $(X_j)_j$ depends only on X_i. The parameters of the HMM are the parameters of the Markov chain, that is its initial distribution and transition matrix, and the parameters of the observations, that is the *emission distributions* $(\nu_k^*)_{k\in\mathcal{X}}$ where ν_k^* is the distribution of Y_j conditionally to $X_j = k$. Only the observations $(Y_j)_j$ are available.

In this article, we focus on estimating the emission distributions in a nonparametric setting. More specifically, assume that the emission distributions have a density with respect to some known dominating measure μ, and write f_k^* their densities–which we call the *emission densities*. The goal of this paper is to estimate all f_k^*'s with their minimax rate of convergence when the emission densities are not restricted to belong to a set of densities described by finitely many parameters.

1.1 Nonparametric State-by-state Adaptivity

Theoretical results in the nonparametric setting have only been developed recently. De Castro et al. (2017) and Bonhomme et al. (2016b) introduce spectral methods, and the latter is proved to be minimax but not adaptive–which means one needs to know the regularity of the densities beforehand to reach the minimax rate of convergence. De Castro et al. (2016) introduce a least squares estimator which is shown to be minimax adaptive up to a logarithmic term. However, all these papers have a common drawback: they study the emission densities as a whole and can not handle them separately. This comes from their error criterion, which is the supremum of the errors on all densities: what they actually prove is that $\max_{k\in\mathcal{X}} \|\hat{f}_k - f_k^*\|_2$ converges with minimax rate when $(\hat{f}_k)_k$ are their density estimators. In general, the regularity of each emission density could be different, leading to different rates of convergence. This means that having just one emission density that is very hard to estimate is enough to deteriorate the rate of convergence of all emission densities.

In this paper, we construct an estimator that is adaptive and estimates each emission density with its own minimax rate of convergence. We call this property state-by-state adaptivity. Our method does so by handling each emission density individually in a way that is theoretically justified–reaching minimax and adaptive rates of convergence with respect to the regularity of the emission densities–and computationally efficient thanks to its low computational and sample complexity.

Our approach for estimating the densities nonparametrically is model selection. The core idea is to approximate the target density using a family of parametric models that is dense within the nonparametric class of densities. For a square integrable density f^*, we consider its projection f_M^* on a finite-dimensional space $\mathfrak{P}_M$ (the parametric model), where M is a model index. This projection introduces an error, the *bias*, which is the distance $\|f^* - f_M^*\|_2$ between the target quantity and the model. The larger the model, the smaller the bias. On the other hand, larger models will make the estimation harder, resulting in a larger *variance* $\|\hat{f}_M - f_M^*\|_2^2$. The key step of model selection is to select a model with a small total error–or alternatively, a good *bias-variance tradeoff*.

In many situations, it is possible to reach the minimax rate of convergence with a good bias-variance tradeoff. Previous estimators of the emission densities of a HMM perform such a tradeoff based on an error that takes the transition matrix and all emission densities into account. Such an error leads to a rate of convergence that corresponds to the slowest minimax rate among the different parameters. In contrast, our method performs a bias-variance tradeoff for each emission density using an error term that depends only on the density in question, which makes it possible to reach the minimax rates for each density.

1.2 Plug-in Procedure

The method we propose is based on the method developed in the seminal papers of Goldenshluger and Lepski (2011, 2014) for density estimation, extended by Goldenshluger and Lepski (2013) to the white noise and regression models. It takes a family of estimators as input and chooses the estimator that performs a good bias-variance tradeoff separately for each hidden state. We recommend the article of Lacour et al. (2017) for an insightful presentation of this method in the case of conditional density estimation.

Our method and assumptions are detailed in Section 2. Let us give a quick overview of the method. Assume the densities belong to a Hilbert space $\mathcal{H}$. Given a family of subsets of finite-dimensional subspaces of $\mathcal{H}$ (the models) indexed by M and estimators $\hat{f}_k^{(M)}$ of the emission densities for each hidden state k and each model M, one computes a substitute for the bias of the estimators by

$$A_k(M) = \sup_{M'} \left\{ \left\| \hat{f}_k^{(M')} - \hat{f}_k^{(M \wedge M')} \right\|_2 - \sigma(M') \right\}.$$

for some penalty σ. Then, for each state k, one selects the estimator $\hat{M}_k$ from the model M minimizing the quantity $A_k(M) + 2\sigma(M)$. The penalty σ can also be interpreted as a variance bound, so that this penalization procedure can be seen as performing a bias-variance tradeoff. The novelty of this method is that it selects a different $\hat{M}_k$, that is a different model, for each hidden state: this is where the state-by-state adaptivity comes from. Also note that contrary to Goldenshluger and Lepski (2013), we do not make any assumption on how the estimators are computed, provided a variance bound holds.

The main theoretical result is an oracle inequality on the selected estimators $\hat{f}_k^{(\hat{M}_k)}$, see Theorem 2. As a consequence, we are able to get a rate of convergence that is different for each state. These rates of convergence will even be adaptive minimax up to a logarithmic factor when the method is applied to our two families of estimators: spectral estimators and least squares estimators. To the best of our knowledge, this is the first state-by-state adaptive algorithm for hidden Markov models.

Note that finding the right penalty term σ is essential in order to obtain minimax rates of convergence. This requires a fine theoretical control of the variance of the auxiliary estimators, in the form of assumption $[\mathbf{H}(\epsilon)]$ (see Section 2.1). To the best of our knowledge, there is no suitable result in the literature. This is the second theoretical contribution of this paper: we control two families of estimators in a way that makes it possible to reach adaptive minimax rate with our state-by-state selection method, up to a logarithmic term.

On the practical side, we run this method and several variants on data simulated from a HMM with three hidden states and one irregular density, as illustrated in Section 4. The simulations confirm that it converges with a different rate for each emission density, and that the irregular density does not alter the rate of convergence of the other ones, which is exactly what we wanted to achieve.

Better still, the added computation time is negligible compared to the computation time of the estimators: even for the spectral estimators of Section 3.2 (which can be computed much faster than the least squares estimators and the maximum likelihood estimators using EM), computing the estimators on 200 models for 50,000 observations (the lower bound of our sample sizes) of a 3-states HMM requires a few minutes, compared to a couple of seconds for the state-by-state selection step. The difference becomes even larger for more observations, since the complexity of the state-by-state selection step is independent of the sample size: for instance, computing the spectral estimators on 300 models for 2,200,000 observations requires a bit less than two hours, and a bit more than ten hours for 10,000,000 observations, compared to less than ten seconds for the selection step in both cases. We refer to Section 4.6 for a more detailled discussion about the algorithmic complexity of the algorithms.

1.3 Families of Estimators

We use two methods to construct families of estimators and apply the selection algorithm. The motivation and key result of this part of the paper is to control the variances of the estimators by the right penalty σ. This part is crucial if one wants to get adaptive minimax rates, and has not been adressed in previous papers. For both methods, we develop new theoretical results that allow to obtain a penalty σ that leads to adaptive minimax rates of convergence up to a logarithmic term. We present the algorithms and theoretical guarantees in Section 3.

The first method is a spectral method and is detailed in Section 3.2. Several spectral algorithms were developed, see for instance Anandkumar et al. (2012) and Hsu et al. (2012) in the parametric setting, and Bonhomme et al. (2016b) and De Castro et al. (2017) in a nonparametric framework. The main advantages of spectral methods are their computational efficiency and the fact that they do not resort to optimization procedure such as the EM and more generally nonconvex optimization algorithm, thus avoiding the well-documented issue of getting stuck into local sub-optimal minima.

Our spectral algorithm is based on the one studied in De Castro et al. (2017). However, their estimator cannot reach the minimax rate of convergence: the variance bound $\sigma(M)$ deduced from their results is proportional to M^3, while reaching the minimax rate requires $\sigma(M)$ to be proportional to M. To solve this issue, we introduce a modified version of their algorithm and show that it has the right variance bound, so that it is able to reach the adaptive minimax rate after our state-by-state selection procedure, up to a logarithmic term. Our algorithm also has an improved complexity: it is at most quasi-linear in the number of observations and in the model dimension, instead of cubic in the model dimension for the original algorithm.

The second method is a least squares method and is detailed in Section 3.3. Nonparametric least squares methods were first introduced by De Castro et al. (2016) to estimate the emission densities and extended by Lehéricy (to appear) to estimate all parameters at once. They rely on estimating the density of three consecutive observations of the HMM using a least squares criterion. Since the model is identifiable from the distribution of three consecutive observations when the emission distributions are linearly independent, it is possible to recover the parameters from this density. In practice, these methods are more accurate than the spectral methods and are more stable when the models are close to not satisfying the identifiability condition, see for instance De Castro et al. (2016) for the accuracy and Lehéricy (to appear) for the stability. However, since they rely on the minimization of a nonconvex criterion, the computation times of the corresponding algorithms are often longer than the ones from spectral methods.

A key step in proving theoretical guarantees for least squares methods is to relate the error on the density of three consecutive observations to the error on the HMM parameters in order to obtain an oracle inequality on the parameters from the oracle inequality on the density of three observations. More precisely, the difficult part is to lower bound the error on the density by the error on the parameters. Let us write g and g' the densities of the first three observations of a HMM with parameters θ and θ' respectively (these parameters actually correspond to the transition matrix and the emission densities of the HMM). Then

one would like to get

$$\|g - g'\|_2 \geq \mathcal{C}(\theta)\, d(\theta, \theta')$$

where d is the natural $\mathbf{L}^2$ distance on the parameters and $\mathcal{C}(\theta)$ is a positive constant which does not depend on θ'. Such inequalities are then used to lower bound the variance of the estimator of the density of three observations g^* by the variance of the parameter estimators: let g be the projection of g^* and g' be the estimator of g^* on the current approximation space (with index M). Denote θ^*_M and $\hat{\theta}_M$ the corresponding parameters and assume that the error $\|g - g'\|_2$ is bounded by some constant $\sigma'(M)$, then the result will be that

$$d(\hat{\theta}_M, \theta^*_M) \leq \frac{\sigma'(M)}{\mathcal{C}(\theta^*_M)}.$$

Such a result is crucial to control the variance of the estimators by a penalty term σ, which is the result we need for the state-by-state selection method. In the case where only the emission densities vary, De Castro et al. (2016) proved that such an inequality always holds for HMMs with 2 hidden states using brute-force computations, but it is still unknown whether it is always true for larger number of states. When the number of states is larger than 2, they show that this inequality holds under a generic assumption. Lehéricy (to appear) extended this result to the case where all parameters may vary. However, the constants deduced from both articles are not explicit, and their regularity (when seen as a function of θ) is unknown, which makes it impossible to use in our setting: one needs the constants $\mathcal{C}(\theta^*_M)$ to be lower bounded by the same positive constant, which requires some sort of regularity on the function $\theta \longmapsto \mathcal{C}(\theta)$ in the neighborhood of the true parameters.

To solve this problem, we develop a finer control of the behaviour of the difference $\|g - g'\|_2$, which is summarized in Theorem 10. We show that it is possible to assume $\mathcal{C}$ to be lower semicontinuous and positive without any additional assumption. In addition, we give an explicit formula for the constant when θ' and θ are close, which gives an explicit bound for the asymptotical rate of convergence. This result allows us to control the variance of the least squares estimators by a penalty σ which ensures that the state-by-state method reaches the adaptive minimax rate up to a logarithmic term.

1.4 Numerical Validation and Application to Real Data Sets

Section 4 shows how to apply the state-by-state selection method in practice and shows its performance on simulated data and a comparison with a method based on cross validation that does note estimate state by state.

Note that the theoretical results give a penalty term σ known only up to a multiplicative constant which is unknown in practice. This problem, the *penalty calibration* issue, is usual in model selection methods. It can be solved using algorithms such as the dimension jump heuristics, see for instance Birgé and Massart (2007), who introduce this heuristics and prove that it leads to an optimal penalization in the special case of Gaussian model selection framework. This method has been shown to behave well in practice in a variety of domains, see for instance Baudry et al. (2012). We describe the method and show how to use this heuristics to calibrate the penalties in Section 4.2.

We propose and compare several variants of our algorithm. Section 4.2 shows some variants in the calibration of the penalties and Section 4.3 shows other ways to select the final estimator. We discuss the result of the simulations and the convergence of the selected estimators in Section 4.4.

In Section 4.5, we compare our method with a non state-by-state adaptive method based on cross validation. Finally, we discuss the complexities of the auxiliary estimation methods and of our selection procedures in Section 4.6.

In Section 5, we apply our algorithm to two sets of GPS tracks. The first data set contains trajectories of artisanal fishers from Madagascar, recorded using a regular sampling with 30 seconds time steps. The second data set contains GPS positions of Peruvian seabird, recorded with 1 second time steps. We convert these tracks into the average velocity during each time step and apply our method using spectral estimators as input. The observed behaviour confirms the ability of our method to adapt to the different regularities by selecting different dimensions for each emission density.

Section 6 contains a conclusion and perspectives for this work.

Finally, Appendix A contains the details of our spectral algorithm and Appendix B is dedicated to the proofs.

1.5 Notations

We will use the following notations throughout the paper.

- $[K] = \{1, \dots, K\}$ is the set of integers between 1 and K.
- $\mathfrak{S}(K)$ is the set of permutations of $[K]$.
- $\|\cdot\|_F$ is the Frobenius norm. We implicitely extend the definition of the Frobenius norm to tensors with more than 2 dimensions.
- $\mathrm{Span}(A)$ is the linear space spanned by the family A.
- $\sigma_1(A) \geq \cdots \geq \sigma_{p\wedge n}(A)$ are the singular values of the matrix $A \in \mathbb{R}^{n\times p}$.
- $\mathbf{L}^2(\mathcal{Y}, \mu)$ is the set of real square integrable measurable functions on $\mathcal{Y}$ with respect to the measure μ.
- For $\mathbf{f} = (f_1, \dots, f_K) \in \mathbf{L}^2(\mathcal{Y}, \mu)^K$, $G(\mathbf{f})$ is the Gram matrix of $\mathbf{f}$, defined by $G(\mathbf{f})_{i,j} = \langle f_i, f_j \rangle$ for all $i, j \in [K]$.

2. The State-by-state Selection Procedure

In this section, we introduce the framework and our state-by-state selection method.

In Section 2.1, we introduce the notations and assumptions. In Section 2.2, we present our selection method and prove that it satisfies an oracle inequality.

2.1 Framework and Assumptions

Let $(X_j)_{j\geq 1}$ be a Markov chain with finite state space $\mathcal{X}$ of size K. Let $\mathbf{Q}^*$ be its transition matrix and π^* be its initial distribution. Let $(Y_j)_{j\geq 1}$ be random variables on a measured space $(\mathcal{Y}, \mu)$ with μ σ-finite such that conditionally on $(X_j)_{j\geq 1}$ the Y_j's are independent with a distribution depending only on X_j. Let ν^*_k be the distribution of Y_j conditionally to $\{X_j = k\}$. Assume that ν^*_k has density f^*_k with respect to μ. We call $(\nu^*_k)_{k\in\mathcal{X}}$ the *emission distributions* and $\mathbf{f}^* = (f^*_k)_{k\in\mathcal{X}}$ the *emission densities*. Then $(X_j, Y_j)_{j\geq 1}$ is a hidden Markov

model with parameters $(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$. The hidden chain $(X_j)_{j\geq 1}$ is assumed to be unobserved, so that the estimators are based only on the observations $(Y_j)_{j\geq 1}$.

Let $(\mathfrak{P}_M)_{M\in\mathbb{N}}$ be a nested family of finite-dimensional subspaces such that their union is dense in $\mathbf{L}^2(\mathcal{Y}, \mu)$. The spaces $(\mathfrak{P}_M)_{M\in\mathbb{N}}$ are our models; in the following we abusively call M the model instead of $\mathfrak{P}_M$. For each index $M \in \mathbb{N}$, we write $\mathbf{f}^{*,(M)} = (f_k^{*,(M)})_{k\in\mathcal{X}}$ the projection of $\mathbf{f}^*$ on $(\mathfrak{P}_M)^K$. It is the best approximation of the true densities within the model M.

In order to estimate the emission densities, we do not need to use every models. Typically there is no point in taking models with more dimensions than the sample size, since they will likely be overfitting. Let $\mathcal{M}_n \subset \mathbb{N}$ be the set of indices which will be used for the estimation from n observations. For each $M \in \mathcal{M}_n$, we assume we are given an estimator $\hat{\mathbf{f}}_n^{(M)} = (\hat{f}_{n,k}^{(M)})_{k\in\mathcal{X}} \in (\mathfrak{P}_M)^K$. We will need to assume that for all models, the variance–that is the distance between $\hat{\mathbf{f}}_n^{(M)}$ and $\mathbf{f}^{*,(M)}$–is small with high probability. In the following, we drop the dependency in n and simply write $\mathcal{M}$ and $\hat{\mathbf{f}}^{(M)}$.

The following result is what one usually obtains in model selection. It bounds the distance between the estimators $\hat{\mathbf{f}}^{(M)}$ and the projections $\mathbf{f}^{*,(M)}$ by some penalty function σ. Thus, $\sigma/2$ can be seen as a bound of the variance term.

[H(ϵ)] With probability $1-\epsilon$,

$$\forall M \in \mathcal{M}, \qquad \inf_{\tau_{n,M}\in\mathfrak{S}(K)} \max_{k\in\mathcal{X}} \left\| \hat{f}_k^{(M)} - f^{*,(M)}_{\tau_{n,M}(k)} \right\|_2 \leq \frac{\sigma(M,\epsilon,n)}{2}$$

where the upper bound $\sigma : (M, \epsilon, n) \in \mathcal{M} \times [0,1] \times \mathbb{N}^* \longmapsto \sigma(M,\epsilon,n) \in \mathbb{R}_+$ is nondecreasing in M. We show in Sections 3.2 and 3.3 how to obtain such a result for a spectral method and for a least squares method (using an algorithm from Lehéricy (to appear)). In the following, we omit the parameters ϵ and n in the notations and only write $\sigma(M)$.

What is important for the selection step is that the permutation $\tau_{n,M}$ does not depend on the model M: one needs all estimators $(\hat{f}_k^{(M)})_{M\in\mathcal{M}}$ to correspond to the same emission density, namely $f^*_{\tau_n(k)}$ when $\tau_{n,M} = \tau_n$ is the same for all models M. This can be done in the following way: let $M_0 \in \mathcal{M}$ and let

$$\hat{\tau}^{(M)} \in \arg\min_{\tau\in\mathfrak{S}(K)} \left\{ \max_{k\in\mathcal{X}} \left\| \hat{f}^{(M)}_{\tau(k)} - \hat{f}^{(M_0)}_k \right\|_2 \right\}$$

for all $M \in \mathcal{M}$. Then, consider the estimators obtained by swapping the hidden states by these permutations. In other words, for all $k \in \mathcal{X}$, consider

$$\hat{f}^{(M)}_{k,\text{new}} = \hat{f}^{(M)}_{\hat{\tau}^{(M)}(k)}.$$

Now, assume that the error on the estimators is small enough. More precisely, write $B_{M,M_0} = \max_{k\in\mathcal{X}} \left\| f_k^{*,(M)} - f_k^{*,(M_0)} \right\|_2$ the distance between the projections of $\mathbf{f}^*$ on the models M and M_0 and assume that $2\left[\sigma(M)/2 + \sigma(M_0)/2 + B_{M,M_0}\right]$ (that is twice the upper bound of the distance between two estimated emission densities corresponding to the same hidden states in models M and M_0) is smaller than $m(\mathbf{f}^*, M_0) := \min_{k'\neq k} \left\| f_k^{*,(M_0)} - f_{k'}^{*,(M_0)} \right\|_2$, which is the smallest distance between two different densities of the vector $\mathbf{f}^{*,(M_0)}$.

Then **[H(ϵ)]** ensures that with probability as least $1-\epsilon$, for all k, there exists a single component of $\hat{\mathbf{f}}^{(M)}$ that is closer than $\sigma(M)/2 + \sigma(M_0)/2$ of $f_k^{*,(M_0)}$, and this component will be $\hat{f}^{(M)}_{\hat{\tau}^{(M)}(k)}$ by definition. This is summarized in the following lemma.

Lemma 1 *Assume [**H**(ϵ)] holds. Then with probability $1-\epsilon$, there exists a permutation $\tau_n \in \mathfrak{S}(K)$ such that for all $k \in \mathcal{X}$ and for all $M \in \mathcal{M}$ such that*

$$\sigma(M) + \sigma(M_0) + 2B_{M,M_0} < m(\mathbf{f}^*, M_0),$$

one has

$$\max_{k\in\mathcal{X}} \left\| \hat{f}^{(M)}_{k,new} - f^{*,(M)}_{\tau_n(k)} \right\|_2 \leq \frac{\sigma(M)}{2}. \tag{1}$$

Proof Proof in Section B.1 ■

Thus, this property holds asymptotically as soon as $\inf \mathcal{M}$ tends to infinity and $\sup_{M\in\mathcal{M}} \sigma(M)$ tends to zero.

2.2 Estimator and Oracle Inequality

Let us now introduce our selection procedure. This method and the following theorem are based on the approach of Goldenshluger and Lepski (2011), but do not require any assumption on the structure of the estimators, provided a variance bound such as Equation (1) holds.

For each $k \in \mathcal{X}$ and $M \in \mathcal{M}$, let

$$A_k(M) = \sup_{M'\in\mathcal{M}} \left\{ \left\| \hat{f}_k^{(M')} - \hat{f}_k^{(M\wedge M')} \right\|_2 - \sigma(M') \right\}.$$

$A_k(M)$ serves as a replacement for the bias of the estimator $\hat{f}_k^{(M)}$, as can be seen in Equation (2). This comes from the fact that for large M', the quantity $\|\hat{f}_k^{(M')} - \hat{f}_k^{(M)}\|_2$ is upper bounded by the variances $\|\hat{f}_k^{(M')} - f_k^{*,(M')}\|_2$ and $\|\hat{f}_k^{(M)} - f_k^{*,(M)}\|_2$ (which are bounded by $\sigma(M')/2$) plus the bias $\|f_k^{*,(M)} - f_k^*\|_2$. Thus, only the bias term remains after substracting the variance bound $\sigma(M')$.

Then, for all $k \in \mathcal{X}$, select a model through the bias-variance tradeoff

$$\hat{M}_k \in \arg\min_{M\in\mathcal{M}} \{A_k(M) + 2\sigma(M)\}$$

and finally take

$$\hat{f}_k = \hat{f}_k^{(\hat{M}_k)}.$$

The following theorem shows an oracle inequality on this estimator.

Theorem 2 *Let $\epsilon \geq 0$ and assume equation (1) holds for all $k \in \mathcal{X}$ with probability $1-\epsilon$. Then with probability $1-\epsilon$,*

$$\forall k \in \mathcal{X}, \quad \|\hat{f}_k - f^*_{\tau_n(k)}\|_2 \leq 4 \inf_{M\in\mathcal{M}} \left\{ \|f^{*,(M)}_{\tau_n(k)} - f^*_{\tau_n(k)}\|_2 + \sigma(M,\epsilon) \right\}.$$

Proof We restrict ourselves to the event of probability at least $1-\epsilon$ where equation (1) holds for all $k \in \mathcal{X}$.

The first step consists in decomposing the total error: for all $M \in \mathcal{M}$ and $k \in \mathcal{X}$,

$$\left\|\hat{f}_k^{(\hat{M}_k)} - f^*_{\tau_n(k)}\right\|_2 \leq \left\|\hat{f}_k^{(\hat{M}_k)} - \hat{f}_k^{(\hat{M}_k \wedge M)}\right\|_2 + \left\|\hat{f}_k^{(\hat{M}_k \wedge M)} - \hat{f}_k^{(M)}\right\|_2 + \left\|\hat{f}_k^{(M)} - f^{*,(M)}_{\tau_n(k)}\right\|_2 + \left\|f^{*,(M)}_{\tau_n(k)} - f^*_{\tau_n(k)}\right\|_2.$$

From now on, we will omit the subscripts k and $\tau_n(k)$. Using equation (1) and the definition of $A(M)$ and $\hat{M}$, one gets

$$\begin{aligned}\left\|\hat{f}^{(\hat{M})} - f^*\right\|_2 \leq & (A(M) + \sigma(\hat{M})) + (A(\hat{M}) + \sigma(M)) \\ & + \sigma(M) + \left\|f^{*,(M)} - f^*\right\|_2 \\ \leq & 2A(M) + 4\sigma(M) + \left\|f^{*,(M)} - f^*\right\|_2.\end{aligned}$$

Then, notice that $A(M)$ can be bounded by

$$\begin{aligned}A(M) \leq & \sup_{M'} \left\{\left\|\hat{f}^{(M')} - f^{*,(M')}\right\|_2 + \left\|\hat{f}^{(M\wedge M')} - f^{*,(M\wedge M')}\right\|_2 - \sigma(M')\right\} \\ & + \sup_{M'} \left\|f^{*,(M')} - f^{*,(M\wedge M')}\right\|_2.\end{aligned}$$

Since σ is nondecreasing, $\sigma(M \wedge M') \leq \sigma(M')$, so that the first term is upper bounded by zero thanks to equation (1). The second term can be controlled since the orthogonal projection is a contraction. This leads to

$$A(M) \leq \left\|f^* - f^{*,(M)}\right\|_2, \tag{2}$$

which is enough to conclude. ■

Remark 3 *The oracle inequality also holds when taking*

$$A_k(M) = \sup_{M' \geq M} \left\{\left\|\hat{f}_k^{(M')} - \hat{f}_k^{(M)}\right\|_2 - \sigma(M')\right\}_+.$$

Remark 4 *Note that the selected $\hat{M}_k$ implicitely depends on the probability of error ϵ through the penalty σ.*

In the asymptotic setting, we take ϵ as a function of n, so that $\hat{M}_k$ is a function of n only. This will be used to get rid of ϵ when proving that the estimators reach the minimax rates of convergence.

3. Plug-in Estimators and Theoretical Guarantees

In this section, we introduce two methods to construct families of estimators of the emission densities. We show that they satisfy assumption [$\mathbf{H}(\epsilon)$] for a given variance bound σ.

In Section 3.1, we introduce the assumptions we will need for both methods. Section 3.2 is dedicated to the spectral estimator and Section 3.3 to the least squares estimator.

3.1 Framework and Assumptions

Recall that we approximate $\mathbf{L}^2(\mathcal{Y}, \mu)$ by a nested family of finite-dimensional subspaces $(\mathfrak{P}_M)_{M\in\mathcal{M}}$ such that their union is dense in $\mathbf{L}^2(\mathcal{Y}, \mu)$ and write $f_k^{*,(M)}$ the orthogonal projection of f_k^* on $\mathfrak{P}_M$ for all $k \in \mathcal{X}$ and $M \in \mathcal{M}$. We assume that $\mathcal{M} \subset \mathbb{N}$ and that the space $\mathfrak{P}_M$ has dimension M. A typical way to construct such spaces is to take $\mathfrak{P}_M$ spanned by the first M vectors of an orthonormal basis.

Both methods will construct an estimator of the emission densities for each model of this family. These estimators will then be plugged in the state-by-state selection method of Section 2.2, which will select one model for each state of the HMM.

We will need the following assumptions.

[**HX**] $(X_j)_{j\geq 1}$ is a stationary ergodic Markov chain with parameters $(\pi^*, \mathbf{Q}^*)$;

[**Hid**] $\mathbf{Q}^*$ is invertible and the family $\mathbf{f}^*$ is linearly independent.

The ergodicity assumption in [**HX**] is standard in order to obtain convergence results. In this case, the initial distribution is forgotten exponentially fast, so that the HMM will essentially behave like a stationary process after a short period of time. For the sake of simplicity, we assume the Markov chain to be stationary.

[**Hid**] appears in identifiability results, see for instance Gassiat et al. (2015) and Theorem 8. It is sufficient to ensure identifiability of the HMM from the law of three consecutive observations. Note that it is in general not possible to recover the law of a HMM from two observations (see for instance Appendix G of Anandkumar et al. (2012)), so that three is actually the minimum to obtain general identifiability.

3.2 The Spectral Method

Algorithm 1 is a variant of the spectral algorithm introduced in De Castro et al. (2017). Unlike the original one, it is able to reach the minimax rate of convergence thanks to two improvements. The first one consists in decomposing the joint density on different models, hence the use of two dimensions m and M. The second one consists in trying several randomized joint diagonalizations instead of just one, and selecting the best one, hence the parameter r. These additional parameters do not actually add much to the complexity of the algorithm: in theory, the choice $m, r \approx \log(n)$ is fine (see Corollary 6), and in practice, any large enough constant works, see Section 4 for more details.

For all $M \in \mathcal{M}$, let $(\varphi_1^M, \dots, \varphi_M^M)$ be an orthonormal basis of $\mathfrak{P}_M$. Let

$$\eta_3(m, M)^2 := \sup_{y,y' \in \mathcal{Y}^3} \sum_{a,c=1}^{m} \sum_{b=1}^{M} (\varphi_a^m(y_1)\varphi_b^M(y_2)\varphi_c^m(y_3) - \varphi_a^m(y_1')\varphi_b^M(y_2')\varphi_c^m(y_3'))^2.$$

The following theorem follows the proof of Theorem 3.1 of De Castro et al. (2017), with modifications that allow to control the error of the spectral estimators in expectation and are essential to obtain the right rates of convergence in Corollary 6.

Theorem 5 *Assume **[HX]** and **[Hid]** hold. Then there exists a constant M_0 depending on $\mathbf{f}^*$ and constants C_σ and n_1 depending on $\mathbf{f}^*$ and $\mathbf{Q}^*$ such that for all $\epsilon \in (0,1)$, for all*

Algorithm 1: Spectral estimation of the emission densities of a HMM (short version)

Data: A sequence of observations $(Y_1, \dots, Y_{n+2})$, two dimensions $m \leq M$, an orthonormal basis $(\varphi_1, \dots, \varphi_M)$ and number of retries r.

Result: Spectral estimators $(\hat{f}_k^{(M,r)})_{k\in\mathcal{X}}$.

[Step 1] Consider the following empirical estimators: for any $a, c \in [m]$ and $b \in [M]$,

- $\hat{\mathbf{M}}_{m,M,m}(a,b,c) := \frac{1}{n}\sum_{s=1}^{n} \varphi_a(Y_s)\varphi_b(Y_{s+1})\varphi_c(Y_{s+2})$
- $\hat{\mathbf{P}}_{m,m}(a,c) := \frac{1}{n}\sum_{s=1}^{n} \varphi_a(Y_s)\varphi_c(Y_{s+2})$.

[Step 2] Let $\hat{\mathbf{U}}_m$ be the $m \times K$ matrix of orthonormal left singular vectors of $\hat{\mathbf{P}}_{m,m}$ corresponding to its top K singular values. $\hat{\mathbf{U}}_m$ can be seen as a projection. Denote by $\mathbf{P}'$ and $\mathbf{M}'(\cdot,b,\cdot)$ the projected tensors, defined by $\mathbf{P}' = \hat{\mathbf{U}}_m^\top \hat{\mathbf{P}}_{m,m}\hat{\mathbf{U}}_m$ and likewise for $\mathbf{M}'$.

[Step 3] Form the matrices $\mathbf{B}(b) := (\mathbf{P}')^{-1}\mathbf{M}'$ for all $b \in [M]$.

[Step 4] Construct a matrix $\hat{\mathbf{O}}$ by taking the best approximate simultaneous diagonalization of all $\mathbf{B}(b)$ among r attempts: for all $b \in [M]$, $\mathbf{B}(b) \approx \mathbf{R}\mathrm{Diag}[\hat{\mathbf{O}}(b,\cdot)]\mathbf{R}^{-1}$ for some matrix $\mathbf{R}$ (see details in Algorithm 3, in Appendix A).

[Step 5] Define the emission densities estimators $\hat{\mathbf{f}}^{(M,r)} := (\hat{f}_k^{(M,r)})_{k\in\mathcal{X}}$ by: for all $k \in \mathcal{X}$, $\hat{f}_k^{(M,r)} := \sum_{b=1}^{M} \hat{\mathbf{O}}(b,k)\varphi_b$.

$m, M \in \mathcal{M}$ such that $M \geq m \geq M_0$ and for all $n \geq n_1\eta_3^2(m,M)(-\log\epsilon)^2$, with probability greater than $1-6\epsilon$,

$$\inf_{\tau\in\mathfrak{S}(K)} \max_{k\in\mathcal{X}} \|\hat{f}_k^{(M,\lceil t\rceil)} - f_{\tau(k)}^{*,(M)}\|_2^2 \leq C_\sigma \eta_3^2(m,M)\frac{(-\log\epsilon)^2}{n}$$

Proof Proof in Section B.2. ■

Note that the constants n_1 and C_σ depend on $\mathbf{Q}^*$ and $\mathbf{f}^*$. This dependency will not affect the rates of convergence of the estimators (with respect to the sample size n), but it can change the constants of the bounds and the minimum sample size needed to reach the asymptotic regime.

Let us now apply the state-by-state selection method to these estimators. The following corollary shows that it is possible to reach the minimax rate of convergence up to a logarithmic term separately for each state under standard assumptions. Note that we need to bound the resulting estimators by some power of n, but this assumption is not very restrictive since α can be arbitrarily large.

Corollary 6 *Assume **[HX]** and **[Hid]** hold. Also assume that $\eta_3^2(m,M) \leq C_\eta m^2 M$ for a constant $C_\eta > 0$ and that for all $k \in \mathcal{X}$, there exists s_k such that $\|f_k^{*,(M)} - f_k^*\|_2 = O(M^{-s_k})$. Then there exists a constant C_σ depending on $\mathbf{f}^*$ and $\mathbf{Q}^*$ such that the following holds.*

Let $\alpha > 0$ and $C \geq 2(1+2\alpha)\sqrt{C_\eta C_\sigma}$. Let $\hat{\mathbf{f}}^{sbs}$ be the estimators selected from the family $(\hat{\mathbf{f}}^{(M,\lceil(1+2\alpha)\log(n)\rceil)})_{M\leq M_{\max}(n)}$ with $M_{\max}(n) = n/\log(n)^5$, $m_M = \log(n)$ and $\sigma(M) = C\sqrt{\frac{M\log(n)^4}{n}}$ for all M. Then there exists a sequence of random permutations $(\tau_n)_{n\geq 1}$ such that

$$\forall k \in \mathcal{X}, \qquad \mathbb{E}\left[\left\|(-n^\alpha)\vee(\hat{f}^{sbs}_{\tau_n(k)}\wedge n^\alpha) - f_k^*\right\|_2^2\right] = O\left(\left(\frac{n}{\log(n)^4}\right)^{\frac{-2s_k}{2s_k+1}}\right).$$

The novelty of this result is that each emission density is estimated with its own rate of convergence: the rate $\frac{-s_k}{2s_k+1}$ is different for each emission density, even though the original spectral estimators did not handle them separately. This is due to our state-by-state selection method.

Moreover, it is able to reach the minimax rate for each density in an adaptive way. For instance, in the case of a β-Hölder density on $\mathcal{Y} = [0,1]^D$ (equipped with a trigonometric basis), one can easily check the control of η_3, and the control $\|f_k^{*,(M)} - f_k^*\|_2 = O(M^{-\beta/D})$ follows from standard approximation results, see for instance DeVore and Lorentz (1993). Thus, our estimators converge with the rate $(n/\log(n)^4)^{-2\beta/(2\beta+D)}$ to this density: this is the minimax rate up to a logarithmic factor.

Remark 7 *By aligning the estimators like in Section 2.1, one can replace the sequence of permutations in Corollary 6 by a single permutation, in other words there exists a random permutation τ which does not depend on n such that*

$$\forall k \in \mathcal{X}, \qquad \mathbb{E}\left[\left\|(-n^\alpha)\vee(\hat{f}^{sbs}_{\tau(k)}\wedge n^\alpha) - f_k^*\right\|_2^2\right] = O\left(\left(\frac{n}{\log(n)^4}\right)^{\frac{-2s_k}{2s_k+1}}\right).$$

This means that the sequence $(\hat{f}_k^{sbs})_{n\geq 1}$ is an adaptive rate-minimax estimator of f_k^*–or more precisely of one of the emission densities $(f_{k'}^*)_{k'\in\mathcal{X}}$, but since the distribution of the HMM is invariant under relabelling of the hidden states, one can assume the limit to be f_k^* without loss of generality–up to a logarithmic term.

At this point, it is important to note that the choice of the constant $C \geq 2(1+2\alpha)\sqrt{C_\eta C_\sigma}$ depends on the hidden parameters of the HMM and as such is unknown. This penalty calibration problem is very common in the model selection framework and can be solved in practice using methods such as the slope heuristics or the dimension jump method which have been proved to be theoretically valid in several cases, see for instance Baudry et al. (2012) and references therein. We use the dimension jump method and explain its principle and implementation in Section 4.2.

Proof Using Theorem 5, one gets that for all n and for all $M \in \mathcal{M}$ such that $n \geq n_1\eta_3^2(m_M,M)(1+2\alpha)^2\log(n)^2$, with probability $1-6n^{-1-2\alpha}$,

$$\begin{aligned}\inf_{\tau\in\mathfrak{S}(K)} \max_{k\in\mathcal{X}} \|\hat{f}_k^{(M,\lceil t\rceil)} - f_{\tau(k)}^{*,(M)}\|_2^2 &\leq C_\sigma\eta_3^2(m_M,M)\frac{(1+2\alpha)^2\log(n)^2}{n}\\ &\leq (1+\alpha)^2 C_\sigma C_\eta M\frac{\log(n)^4}{n}\\ &\leq \frac{\sigma(M)^2}{4}\end{aligned}$$

where $\sigma(M) = C\sqrt{\frac{M\log(n)^4}{n}}$ with C such that $C^2 \geq 4(1+2\alpha)^2 C_\sigma C_\eta$.

The condition on M becomes

$$n \geq n_1 \log(n)^4 M(1+2\alpha)^2$$

and is asymptotically true for all $M \leq M_{\max}(n)$ as soon as $M_{\max}(n) = o(n/\log(n)^4)$.

Thus, $[\mathbf{H}(6n^{-(1+2\alpha)})]$ is true for the family $(\hat{\mathbf{f}}^{(M,\lceil(1+2\alpha)\log(n)\rceil)})_{M\leq M_{\max}(n)}$. Note that the assumption $M_{\max}(n) = o(n/\log(n)^4)$ also implies that there exists M_1 such that for n large enough, Lemma 1 holds for all $M \geq M_1$, so that Theorem 2 implies that for n large enough, there exists a permutation τ_n such that with probability $1 - 6n^{-(1+2\alpha)}$, for all $k \in \mathcal{X}$,

$$\begin{aligned}
\|\hat{f}^{\mathrm{sbs}}_{\tau_n(k)} - f^*_k\|_2 &\leq 4 \inf_{M_1\leq M\leq M_{\max}} \{\|f^{*,(M)}_k - f^*_k\|_2 + \sigma(M)\} \\
&= O\left(\inf_{M_1\leq M\leq M_{\max}} \left\{M^{-s_k} + \sqrt{\frac{M\log(n)^4}{n}}\right\}\right) \\
&= O\left(\left(\frac{n}{\log(n)^4}\right)^{-s_k/(1+2s_k)}\right),
\end{aligned}$$

where the tradeoff is reached for $M = (\frac{n}{\log(n)^4})^{1/(1+2s_k)}$, which is in $[M_1, M_{\max}(n)]$ for n large enough.

Finally, write A the event of probability smaller than $6n^{-(1+2\alpha)}$ where $[\mathbf{H}(6n^{-(1+\alpha)})]$ doesn't hold, then for n large enough and for all $k \in \mathcal{X}$,

$$\begin{aligned}
\mathbb{E}\left[\left\|(-n^\alpha)\vee(\hat{f}^{\mathrm{sbs}}_{\tau_n(k)}\wedge n^\alpha) - f^*_k\right\|_2^2\right] &\leq \mathbb{E}\left[\mathbf{1}_A\left\|\hat{f}^{\mathrm{sbs}}_{\tau_n(k)} - f^*_k\right\|_2^2\right] + \mathbb{E}\left[\mathbf{1}_{A^c}(n^{2\alpha} + \|f^*_k\|_2^2)\right] \\
&= O\left(\left(\frac{n}{\log(n)^4}\right)^{-2s_k/(1+2s_k)}\right) + O\left(\frac{n^{2\alpha} + \|f^*_k\|_2^2}{n^{1+2\alpha}}\right) \\
&= O\left(\left(\frac{n}{\log(n)^4}\right)^{-2s_k/(1+2s_k)}\right).
\end{aligned}$$

■

3.3 The Penalized Least Squares Method

Let $\mathcal{F}$ be a subset of $\mathbf{L}^2(\mathcal{Y},\mu)$. We will need the following assumption on $\mathcal{F}$ in order to control the deviations of the estimators:

[HF] $\mathbf{f}^* \in \mathcal{F}^{K^*}$, $\mathcal{F}$ is closed under projection on $\mathfrak{P}_M$ for all $M \in \mathcal{M}$ and

$$\forall f \in \mathcal{F}, \quad \begin{cases} \|f\|_\infty \leq C_{\mathcal{F},\infty} \\ \|f\|_2 \leq C_{\mathcal{F},2} \end{cases}$$

with $C_{\mathcal{F},\infty}$ and $C_{\mathcal{F},2}$ larger than 1.

A simple way to construct such a set $\mathcal{F}$ when μ is a finite measure is to take the sets $(\mathfrak{P}_M)_M$ spanned by the first M vectors of an orthonormal basis $(\varphi_i)_{i\geq 0}$ whose first vector φ_0 is proportional to $\mathbf{1}$. Then any set $\mathcal{F}$ of densities such that $\int f d\mu = 1$, $\sum_i \langle f,\varphi_i\rangle^2 \leq C_{\mathcal{F},2}$ and $\sum_i |\langle f,\varphi_i\rangle| \|\varphi_i\|_\infty \leq C_{\mathcal{F},\infty}$ for given constants $C_{\mathcal{F},2}$ and $C_{\mathcal{F},\infty}$ and for all $f \in \mathcal{F}$ satisfies **[HF]**.

When $\mathbf{Q} \in \mathbb{R}^{K\times K}$, $\pi \in \mathbb{R}^K$ and $\mathbf{f} \in (\mathbf{L}^2(\mathcal{Y},\mu))^K$, let

$$g^{\pi,\mathbf{Q},\mathbf{f}}(y_1,y_2,y_3) = \sum_{k_1,k_2,k_3=1}^K \pi(k_1)\mathbf{Q}(k_1,k_2)\mathbf{Q}(k_2,k_3) f_{k_1}(y_1) f_{k_2}(y_2) f_{k_3}(y_3).$$

When π is a probability distribution, $\mathbf{Q}$ a transition matrix and $\mathbf{f}$ a K-tuple of probability densities, then $g^{\pi,\mathbf{Q},\mathbf{f}}$ is the density of the first three observations of a HMM with parameters $(\pi,\mathbf{Q},\mathbf{f})$. The motivation behind estimating $g^{\pi,\mathbf{Q},\mathbf{f}}$ is that it allows to recover the true parameters under the identifiability assumption **[Hid]**, as shown in the following theorem.

Let $\mathcal{Q}$ be the set of transition matrices on $\mathcal{X}$ and Δ the set of probability distributions on $\mathcal{X}$. For a permutation $\tau \in \mathfrak{S}(K)$, write $\mathbb{P}_\tau$ its matrix (that is the matrix defined by $\mathbb{P}_\tau(i,j) = \mathbf{1}_{\{j=\tau(i)\}}$). Finally, define the distance on the HMM parameters

$$\begin{aligned}
&d_{\mathrm{perm}}((\pi_1,\mathbf{Q}_1,\mathbf{f}_1),(\pi_2,\mathbf{Q}_2,\mathbf{f}_2))^2 \\
&\qquad = \inf_{\tau\in\mathfrak{S}(K)} \left\{\|\pi_1 - \mathbb{P}_\tau\pi_2\|_2^2 + \|\mathbf{Q}_1 - \mathbb{P}_\tau\mathbf{Q}_2\mathbb{P}_\tau^\top\|_F^2 + \sum_{k\in\mathcal{X}} \|f_{1,k} - f_{2,\tau(k)}\|_2^2\right\}.
\end{aligned}$$

This distance is invariant under permutation of the hidden states. This corresponds to the fact that a HMM is only identifiable up to relabelling of its hidden states.

Theorem 8 (Identifiability) *Let $(\pi^*,\mathbf{Q}^*,\mathbf{f}^*) \in \Delta\times\mathcal{Q}\times(\mathbf{L}^2(\mathcal{Y},\mu))^K$ such that $\pi^*_x > 0$ for all $x \in \mathcal{X}$ and **[Hid]** holds. Then for all $(\pi,\mathbf{Q},\mathbf{f}) \in \Delta\times\mathcal{Q}\times(\mathbf{L}^2(\mathcal{Y},\mu))^K$,*

$$(g^{\pi,\mathbf{Q},\mathbf{f}} = g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}) \;\Rightarrow\; d_{perm}((\pi,\mathbf{Q},\mathbf{f}),(\pi^*,\mathbf{Q}^*,\mathbf{f}^*)) = 0.$$

Proof The spectral algorithm of De Castro et al. (2017) applied on the finite dimensional space spanned by the components of $\mathbf{f}$ and $\mathbf{f}^*$ allows to recover all the parameters even when the emission densities are not probability densities and when the Markov chain is not stationary. ■

Define the empirical contrast

$$\gamma_n(t) = \|t\|_2^2 - \frac{2}{n}\sum_{j=1}^n t(Z_j)$$

where $Z_j := (Y_j, Y_{j+1}, Y_{j+2})$ and $(Y_j)_{1\leq j\leq n+2}$ are the observations. It is a biased estimator of the $\mathbf{L}^2$ loss: for all $t \in (\mathbf{L}^2(\mathcal{Y},\mu))^3$,

$$\mathbb{E}[\gamma_n(t)] = \|t - g^*\|_2^2 - \|g^*\|_2^2$$

Algorithm 2: Least squares estimation of the emission densities of a HMM

Data: A sequence of observations $(Y_1, \dots, Y_{n+2})$, a dimension M and an orthonormal basis $\Phi = (\varphi_1, \dots, \varphi_M)$.

Result: Least squares estimators $\hat{\pi}^{(M)}$, $\hat{\mathbf{Q}}^{(M)}$ and $(\hat{f}_k^{(M)})_{k\in\mathcal{X}}$.

[Step 1] Compute the tensor $\hat{\mathbf{M}}_M$ defined by $\hat{\mathbf{M}}_M(a,b,c) := \frac{1}{n}\sum_{s=1}^{n} \varphi_a(Y_s)\varphi_b(Y_{s+1})\varphi_c(Y_{s+2})$ for all $a, b, c \in [M]$.

[Step 2] Find a minimizer $(\hat{\pi}^{(M)}, \hat{\mathbf{Q}}^{(M)}, \hat{\mathbf{O}})$ of $(\pi, \mathbf{Q}, \mathbf{O}) \longmapsto \|\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{O})} - \hat{\mathbf{M}}_M\|_F^2$ where

- $\pi \in \mathbb{R}^K$ is a probability distribution on $\mathcal{X}$, i.e. $\sum_{k\in\mathcal{X}} \pi_k = 1$;
- $\mathbf{Q} \in \mathbb{R}^{K\times K}$ is a transition matrix on $\mathcal{X}$, i.e. $\sum_{k'\in\mathcal{X}} Q(k,k') = 1$ for all $k \in \mathcal{X}$;
- $\mathbf{O}$ is a $M \times K$ matrix such that for all $k \in \mathcal{X}$, $\sum_{b=1}^{M} \mathbf{O}(b,k)\varphi_b \in \mathcal{F}$;
- $\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{O})} \in \mathbb{R}^{M\times M\times M}$ is defined by $\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{O})}(\cdot,b,\cdot) = \mathbf{O}\mathrm{Diag}[\pi]\mathbf{Q}\mathrm{Diag}[\mathbf{O}(b,\cdot)]\mathbf{Q}\mathbf{O}^\top$ for all $b \in [M]$.

[Step 3] Consider the emission densities estimators $\hat{\mathbf{f}}^{(M)} := (\hat{f}_k^{(M)})_{k\in\mathcal{X}}$ defined by for all $k \in \mathcal{X}$, $\hat{f}_k^{(M)} := \sum_{b=1}^{M} \hat{\mathbf{O}}(b,k)\varphi_b$.

where $g^* = g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}$. Since the bias does not depend on the function t, one can hope that the minimizers of γ_n are close to minimizers of $\|t - g^*\|_2$. We will show that this is indeed the case.

The least squares estimators of all HMM parameters are defined for each model $\mathfrak{P}_M$ by

$$(\hat{\pi}^{(M)}, \hat{\mathbf{Q}}^{(M)}, \hat{\mathbf{f}}^{(M)}) \in \underset{\pi\in\Delta,\, \mathbf{Q}\in\mathcal{Q},\, \mathbf{f}\in(\mathfrak{P}_M\cap\mathcal{F})^K}{\arg\min} \gamma_n(g^{\pi,\mathbf{Q},\mathbf{f}}).$$

The procedure is summarized in Algorithm 2. Note that with the notations of the algorithm,

$$\gamma_n(g^{\pi,\mathbf{Q},\mathbf{O}^\top\Phi}) = \|\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{O})} - \hat{\mathbf{M}}_M\|_F^2 - \|\hat{\mathbf{M}}_M\|_F^2.$$

Then, the proof of the oracle inequality of Lehéricy (to appear) allows to get the following result.

Theorem 9 *Assume **[HF]**, **[HX]** and **[Hid]** hold.*

Then there exists constants C and n_0 depending on $C_{\mathcal{F},2}$, $C_{\mathcal{F},\infty}$ and $\mathbf{Q}^*$ such that for all $n \geq n_0$, for all $t > 0$, with probability greater than $1 - e^{-t}$, one has for all $M \in \mathcal{M}$ such that $M \leq n$:

$$\|\hat{g}^{\hat{\pi}^{(M)},\hat{\mathbf{Q}}^{(M)},\hat{\mathbf{f}}^{(M)}} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^{*,(M)}}\|_2^2 \leq C\Big(\frac{t}{n} + M\frac{\log(n)}{n}\Big).$$

In order to deduce a control of the error on the parameters–and in particular on the emission densities–from the previous result, we will need to assume that the quadratic form derived from the second-order expansion of $(\pi, \mathbf{Q}, \mathbf{f}) \in \Delta \times \mathcal{Q} \times \mathcal{F}^K \longmapsto \|g^{\pi,\mathbf{Q},\mathbf{f}} - g^*\|_2^2$ around $(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$ is nondegenerate.

It is still unknown whether this nondegeneracy property is true for all parameters $(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$ such that **[Hid]** and **[HX]** hold. De Castro et al. (2016) prove it for $K = 2$ hidden states when only the emission densities are allowed to vary by using brute-force computations. To do so, they introduce an (explicit) polynomial in the coefficients of π^*, $\mathbf{Q}^*$ and of the Gram matrix of $\mathbf{f}^*$ and prove that its value is nonzero if and only if the quadratic form is nondegenerate for the corresponding parameters. The difficult part of the proof is to show that this polynomial is always nonzero.

For the expression of this polynomial–which we will write H–in our setting, we refer to Section B.3. Note that Lehéricy (to appear) proves that this polynomial H is non identically zero: it is shown that there exists parameters $(\pi, \mathbf{Q}, \mathbf{f})$ satisfying **[HX]** and **[Hid]** such that $H(\pi, \mathbf{Q}, \mathbf{f}) \neq 0$, which means that the following assumption is generically satisfied:

[Hdet] $H(\pi^*, \mathbf{Q}^*, \mathbf{f}^*) \neq 0$.

The following result allows to lower bound the $\mathbf{L}^2$ error on the density of three consecutive observations by the error on the parameters of the HMM using this condition. It is an improvement of Theorem 6 of De Castro et al. (2016) and Theorem 9 of Lehéricy (to appear). The main difference is that the constant $c^*(\pi^*, \mathbf{Q}^*, \mathbf{f}^*, \mathcal{F})$ does not depend on the $\mathbf{f}$ around which the parameters are taken. This is crucial to obtain Corollary 11, from which we will deduce **[H0]**. Note that we do not need $\mathbf{f}$ to be in a compact neighborhood of $\mathbf{f}^*$. Another improvement is that the constant in the minoration only depends on the true parameters and on the set $\mathcal{F}$.

Theorem 10 *1. Assume that **[HF]** holds and that for all $f \in \mathcal{F}$, $\int f d\mu = 1$.*

*Then there exist a lower semicontinuous function $(\pi^*, \mathbf{Q}^*, \mathbf{f}^*) \longmapsto c^*(\pi^*, \mathbf{Q}^*, \mathbf{f}^*, \mathcal{F})$ that is positive when **[Hid]** and **[Hdet]** hold and a neighborhood $\mathcal{V}$ of $\mathbf{f}^*$ in $\mathcal{F}^K$ depending only on π^*, $\mathbf{Q}^*$, $\mathbf{f}^*$ and $\mathcal{F}$ such that for all $\mathbf{f} \in \mathcal{V}$ and for all $\pi \in \Delta$, $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{h} \in \mathcal{F}^K$,*

$$\|g^{\pi,\mathbf{Q},\mathbf{h}} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}}\|_2^2 \geq c^*(\pi^*, \mathbf{Q}^*, \mathbf{f}^*, \mathcal{F}) d_{perm}((\pi, \mathbf{Q}, \mathbf{h}), (\pi^*, \mathbf{Q}^*, \mathbf{f}))^2.$$

*2. There exists a continuous function $\epsilon : (\pi^*, \mathbf{Q}^*, \mathbf{f}^*) \mapsto \epsilon(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$ that is positive when **[Hid]** and **[Hdet]** hold and such that for all $\pi \in \Delta$, $\mathbf{Q} \in \mathcal{Q}$ and $\mathbf{h} \in (\mathbf{L}^2(\mathcal{Y}, \mu))^K$ a K-tuple of probability densities such that $d_{perm}((\pi, \mathbf{Q}, \mathbf{h}), (\pi^*, \mathbf{Q}^*, \mathbf{f}^*)) \leq \epsilon(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$, one has*

$$\|g^{\pi,\mathbf{Q},\mathbf{h}} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2 \geq c_0(\pi^*, \mathbf{Q}^*, \mathbf{f}^*) d_{perm}((\pi, \mathbf{Q}, \mathbf{h}), (\pi^*, \mathbf{Q}^*, \mathbf{f}^*))^2.$$

where

$$c_0(\pi^*, \mathbf{Q}^*, \mathbf{f}^*) = \frac{(\inf_{k\in\mathcal{X}} \pi^*(k))\, \sigma_K(\mathbf{Q}^*)^4 \sigma_K(G(\mathbf{f}^*))^2}{4} \wedge \frac{H(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)}{2(1 \wedge K\|G(\mathbf{f}^*)\|_\infty)(3K^3(1 \vee \|G(\mathbf{f}^*)\|_\infty^4))^{K^2-K/2}}.$$

Proof Proof in Section B.4. ∎

Corollary 11 *Assume **[HX]**, **[HF]**, **[Hid]** and **[Hdet]** hold. Also assume that for all $f \in \mathcal{F}$, $\int f d\mu = 1$.*

Then there exists a constant n_0 depending on $C_{\mathcal{F},2}$, $C_{\mathcal{F},\infty}$ and $\mathbf{Q}^*$ and constants M_0 and C' depending on $\mathcal{F}$, $\mathbf{Q}^*$ and $\mathbf{f}^*$ such that for all $n \geq n_0$ and $t > 0$, with probability greater than $1 - e^{-t}$, one has for all $M \in \mathcal{M}$ such that $M_0 \leq M \leq n$:

$$\inf_{\tau \in \mathfrak{S}(K)} \max_{k \in \mathcal{X}} \|\hat{f}_k^{(M)} - f_{\tau(k)}^{*,(M)}\|_2^2 \leq C' \left(M \frac{\log(n)}{n} + \frac{t}{n} \right).$$

Remark 12 *Using the second point of Theorem 10, one can alternatively take n_0 and M_0 depending on $\mathcal{F}$, $\mathbf{Q}^*$ and $\mathbf{f}^*$, and C' depending on $C_{\mathcal{F},2}$, $C_{\mathcal{F},\infty}$, $\mathbf{Q}^*$ and $\mathbf{f}^*$ only. For instance, one can take $C' = C/c_0(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$ with the notations of Theorems 9 and 10.*

In particular, this means that the asymptotic variance bound of the least squares estimators (and therefore the rate of convergence of the estimators selected by our state-by-state selection method) does not depend on the set $\mathcal{F}$, but only on the HMM parameters and on the bounds $C_{\mathcal{F},2}$ and $C_{\mathcal{F},\infty}$ on the square and supremum norms of the emission densities. Note that this universality result is essentially an asymptotic one since it requires n_0 to depend on $\mathcal{F}$ in a non-explicit way.

Proof Let $\mathcal{V}$ be the neighborhood given by Theorem 10, then there exists M_0 such that for all $M \geq M_0$, $\mathbf{f}^{*,(M)} \in \mathcal{V}$. Then Theorem 9 and Theorem 10 applied to $\pi = \hat{\pi}^{(M)}$, $\mathbf{Q} = \hat{\mathbf{Q}}^{(M)}$, $\mathbf{h} = \hat{\mathbf{f}}^{(M)}$ and $\mathbf{f} = \mathbf{f}^{*,(M)}$ for all M allow to conclude. ■

We may now state the following result which shows that the state-by-state selection method applied to these estimators reaches the minimax rate of convergence (up to a logarithmic factor) in an adaptive manner under generic assumptions. Its proof is the same as the one of Corollary 6.

Corollary 13 *Assume **[HX]**, **[HF]**, **[Hid]** and **[Hdet]** hold. Also assume that for all $f \in \mathcal{F}$, $\int f d\mu = 1$ and that for all k, there exists s_k such that $\|f_k^{*,(M)} - f_k^*\|_2 = O(M^{-s_k})$. Then there exists a constant C_σ depending on $C_{\mathcal{F},2}$, $C_{\mathcal{F},\infty}$, $\mathbf{Q}^*$ and $\mathbf{f}^*$ such that the following holds.*

Let $C \geq C_\sigma$ and let $\hat{\mathbf{f}}^{sbs}$ be the estimators selected from the family $(\hat{\mathbf{f}}^{(M)})_{M \leq n}$ with $\sigma(M) = C\sqrt{\frac{M \log(n)}{n}}$ for all M, aligned like in Remark 7. Then there exists a random permutation τ which does not depend on n such that

$$\forall k \in \mathcal{X}, \qquad \mathbb{E}\left[\left\| \hat{f}_{\tau(k)}^{sbs} - f_k^* \right\|_2 \right] = O\left(\left(\frac{n}{\log(n)} \right)^{\frac{-s_k}{2s_k+1}} \right).$$

4. Numerical Experiments

This section is dedicated to the discussion of the practical implementation of our method. We run the spectral estimators on simulated data for different number of observations and study the rate of convergence of the selected estimators for several variants of our method.

Finally, we discuss the algorithmic complexity of the different estimators and selection methods.

In Section 4.1, we introduce the parameters with which we generate the observations. In Section 4.2, we discuss how to calibrate the constant of the penalty in practice. In Section 4.3, we introduce two other ways to select the final estimators, the POS and MAX variants. Section 4.4 contains the results of the simulations for each variant and calibration method. In Section 4.5, we present a cross validation procedure and compare its results with the one obtained using our method. Finally, we discuss the algorithmic complexity of the different algorithms and estimators in Section 4.6.

4.1 Setting and Parameters

We take $\mathcal{Y} = [0, 1]$ equipped with the Lebesgue measure. We choose the approximation spaces spanned by a trigonometric basis: $\mathfrak{P}_M := \mathrm{Span}(\varphi_1, \dots, \varphi_M)$ with

$$\begin{cases} \varphi_1(x) & = 1 \\ \varphi_{2m}(x) & = \sqrt{2}\cos(2\pi m x) \\ \varphi_{2m+1}(x) & = \sqrt{2}\sin(2\pi m x) \end{cases}$$

for all $x \in [0, 1]$ and $m \in \mathbb{N}^*$. We will consider a hidden Markov model with $K = 3$ hidden states and the following parameters:

- Transition matrix

$$\mathbf{Q}^* = \begin{pmatrix} 0.7 & 0.1 & 0.2 \\ 0.08 & 0.8 & 0.12 \\ 0.15 & 0.15 & 0.7 \end{pmatrix};$$

- Emission densities (see Figure 1)
  - Uniform distribution on $[0; 1]$;
  - Symmetrized Beta distribution, that is a mixture with the same weight of $\frac{2}{3}X$ and $1 - \frac{1}{3}X'$ with X, X' iid following a Beta distribution with parameters $(3, 1.6)$;
  - Beta distribution with parameters $(3, 7)$.

We generate n observations and run the spectral algorithm in order to obtain estimators for the models $\mathfrak{P}_M$ with $\texttt{M\_min} \leq M \leq \texttt{M\_max}$, $m = 20$ and $r = \lceil 2\log(n) + 2\log(M) \rceil$, where $\texttt{M\_min} = 3$ and $\texttt{M\_max} = 300$. Finally, we use the state-by-state selection method to choose the final estimator for each emission density. The main reason for using spectral estimators instead of maximum likelihood estimation or least squares estimation is its computational speed: it is much faster for large n than the least squares algorithm or the EM algorithm, which makes studying asymptotic behaviours possible.

We made 300 simulations, 20 per value of n, with n taking values in $\{5 \times 10^4, 7 \times 10^4, 1 \times 10^5, 1.5 \times 10^5, 2.2 \times 10^5, 3.5 \times 10^5, 5 \times 10^5, 7 \times 10^5, 1 \times 10^6, 1.5 \times 10^6, 2.2 \times 10^6, 3.5 \times 10^6, 5 \times 10^6, 7 \times 10^6, 1 \times 10^7\}$.

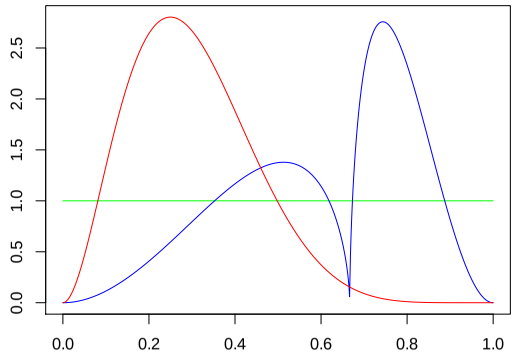


Figure 1: Emission densities. In all following figures, the uniform distribution corresponds to the green lines, the Beta distribution to the red lines and the symmetrized Beta distribution to the blue lines.

4.2 Penalty Calibration

It is important to note that when considering spectral and least squares methods, the penalty σ in the state-by-state selection procedure depends on the hidden parameters of the HMM and as such is unknown in practice. This penalty calibration problem is well known and several procedures exist that allow to solve it, for instance the slope heuristics and the dimension jump method (see for instance Baudry et al. (2012) and references therein). In the following, we will use the dimension jump method to calibrate the penalty in the state-by-state selection procedure.

Consider a penalty shape $\mathrm{pen}_{\mathrm{shape}}$ and define $\hat{M}_k(\rho)$ the model selected for the hidden state k by the state-by-state selection estimator using the penalty $\rho\,\mathrm{pen}_{\mathrm{shape}}$:

$$\hat{M}_k(\rho) \in \operatorname*{arg\,min}_{M\in\mathcal{M}}\{A_k(M) + 2\rho\,\mathrm{pen}_{\mathrm{shape}}(M)\}.$$

where

$$A_k(M) = \sup_{M'\in\mathcal{M}} \left\{ \left\| \hat{f}_k^{(M')} - \hat{f}_k^{(M\wedge M')} \right\|_2 - \rho\,\mathrm{pen}_{\mathrm{shape}}(M') \right\}.$$

The dimension jump method relies on the heuristics that there exists a constant C such that $C\,\mathrm{pen}_{\mathrm{shape}}$ is a *minimal penalty*. This means that for all $\rho < C$, the selected models $\hat{M}_k(\rho)$ will be very large, while for $\rho > C$, the models will remain small. This translates into a sharp jump located around a value $\rho_{\mathrm{jump},k} = C$ in the plot of $\rho \longmapsto \hat{M}_k(\rho)$. The final step consists in taking twice this value to calibrate the constant of the penalty, thus

selecting the model $\hat{M}(2\rho_{\mathrm{jump},k})$. In practice, we take $\rho_{\mathrm{jump},k}$ as the position of the largest jump of the function $\rho \longmapsto \hat{M}_k(\rho)$.

Figure 2 shows the resulting dimension jumps for $n = 220,000$ observations. Each curve corresponds to one of the $\hat{M}_k(\rho)$ and has a clear dimension jump, which confirms the relevance of the heuristics. Several methods may be used to calibrate the constant of the penalty:

eachjump. Calibrate the constant independently for each state. This method has the advantage of being easy to calibrate since there is usually a single sharp jump in each state's complexity. However, our theoretical results do not suggest that the penalty constant is different for each state;

jumpmax. Calibrate the constant for all states together using only the latest jump. This consists in taking the maximum of the $\rho_{\mathrm{jump},k}$ to select the final models. Since the penalty is known up to a multiplicative constant and taking a constant larger than needed does not affect the rates of convergence–contrary to smaller constants–this is the "safe" option;

jumpmean. Calibrate the constant for all states together using the mean of the positions of the different jumps.

We try and compare these calibration methods in Section 4.4.

4.3 Alternative Selection Procedures

4.3.1 Variant POS.

As mentionned in Section 2.2, it is also possible to select the estimators using the criterion

$$A_k(M) = \sup_{M'\geq M} \left\{ \left\| \hat{f}_k^{(M')} - \hat{f}_k^{(M)} \right\|_2 - \sigma(M') \right\}_+$$

followed by

$$\hat{M}_k \in \operatorname*{arg\,min}_{M\in\mathcal{M}}\{A_k(M) + 2\sigma(M)\}.$$

This positivity condition was in the original Goldenshluger-Lepski method. The theoretical guarantees remain the same as the previous method and both behave almost identically in practice, as shown in Section 4.4.

4.3.2 Variant MAX.

In the context of kernel density estimation, Lacour et al. (2017) show that the Goldenshluger-Lepski method still works when the biais estimate $A_k(M)$ of the model M is replaced by the distance between the estimator of the model M and the estimator with the smallest bandwidth (the analog of the largest model in our setting). They also prove an oracle inequality for this method after adding a corrective term to the penalty.

The following variant is based on the same idea. It consists in selecting the model

$$\hat{M}_k \in \operatorname*{arg\,min}_{M\in\mathcal{M}}\{\| \hat{f}_k^{(M_{\max})} - \hat{f}_k^{(M)} \|_2 + \sigma(M)\}$$

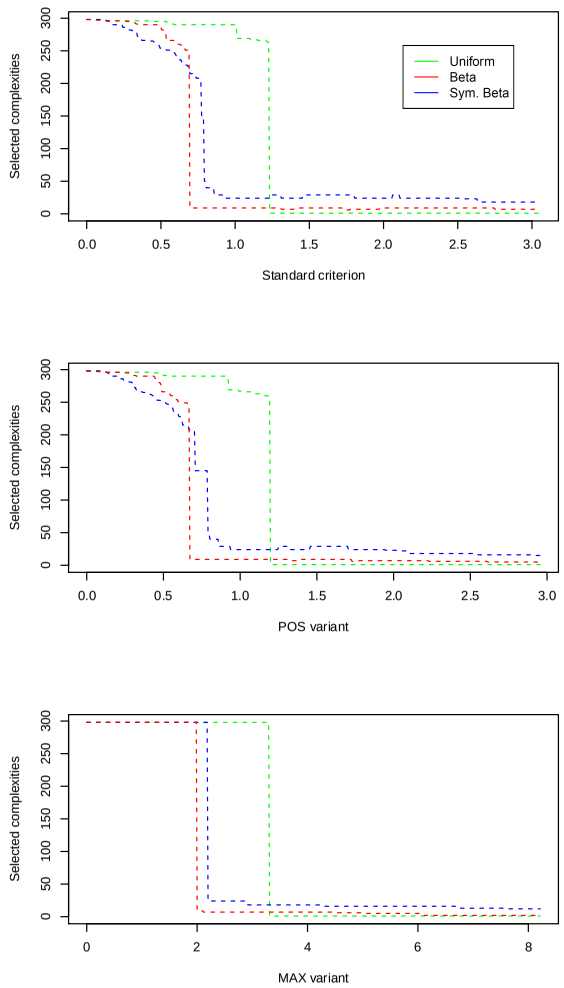


Figure 2: Selected complexities with respect to the penalty constant ρ for the same simulation of $n = 500,000$ observations. The colored dashed lines correspond to the single-state complexities $M_k(\rho)$.

for each $k \in \mathcal{X}$ and takes

$$\hat{f}_k = \hat{f}_k^{(\hat{M}_k)},$$

where σ is the same penalty as the one in the usual state-by-state selection method.

An advantage of this algorithm is its lower complexity, since it requires $O(M_{\max})$ computations of $\mathbf{L}^2$ norms instead of $O(M_{\max}^2)$. We do not study this method theoretically in our setting. However, the simulations (and in particular Figure 4) show that it behaves similarly to the standard state-by-state selection method in the asymptotic regime and even has a smaller error for small number of observations. In addition, the dimension jumps are much sharper for this method than for the usual state-by-state selection method (see Figure 2), which makes the calibration heuristics easier to use.

4.4 Results

Figure 3 shows the evolution of the error $\|\hat{f}_k - f_k^*\|_2$ for each state k with respect to the number of observations n, for all penalty calibration methods and all variants of the model selection procedure. Figure 4 compares the evolution of the median error for the different calibration methods and for the different selection variants, and Figure 5 compares two estimators with the oracle estimators.

When the number of observations n is large enough, the logarithm of the error decreases linearly with respect to $\log(n)$. This corresponds to the asymptotic convergence regime: the error is expected to decrease as a power of the number of observations n when n tends to infinity. The corresponding slopes are listed in Table 1.

For each state, the confidence intervals of the rates of all estimators–including the oracle estimators–have a common intersection (except for the symmetrized Beta distribution in the jumpmax MAX variant, whose estimators seem to converge faster than the others). This tends to confirm that the calibration and selection variants are asymptotically equivalent. This phenomenon is also visible in Figures 3 and 4: in the asymptotic regime, the errors decrease in a similar way for all methods.

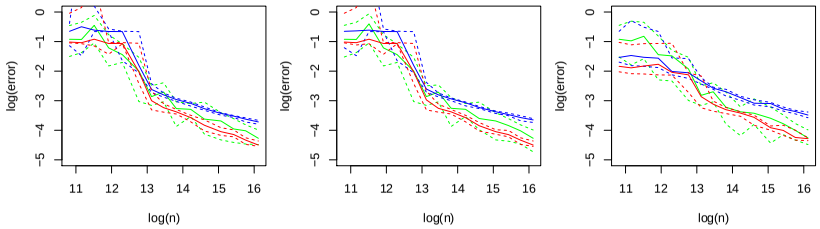
Furthermore, the rates of convergence are clearly distinct. The uniform distribution is estimated with a rate of convergence of approximately $n^{-1/2}$, which is also the best possible rate (it corresponds to a parametric estimation rate). In comparison, the rate of convergence for the symmetrized Beta distribution is much slower (around $n^{-0.36}$). This shows that the algorithm effectively adapts to the regularity of each state and that one irregular emission density does not deteriorate the rates of convergence of the other densities.

Note that the above rates are in accordance with the minimax rates as far as the Hölder regularity is concerned. The minimax Hölder rate for the symmetrized Beta (which is 0.6-Hölder) is $n^{-3/11}$, or approximately $n^{-0.27}$, which means our estimator converges faster than the minimax rate would suggest. The minimax Hölder rate for the Beta distribution (which is 3-Hölder) is $n^{-3/7}$, or approximately $n^{-0.43}$, which is around the observed value.

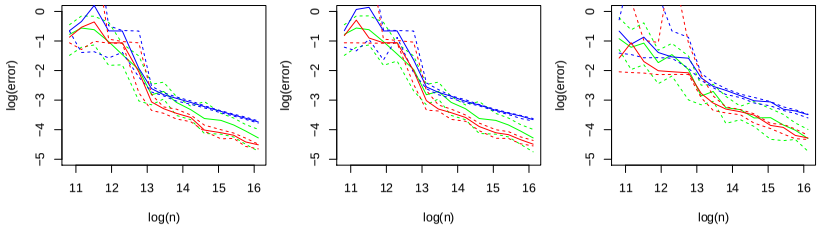
4.5 Comparison with Cross Validation

In this section, we use a cross validation procedure based on our spectral estimators to check whether our method actually improves estimation accuracy.œ

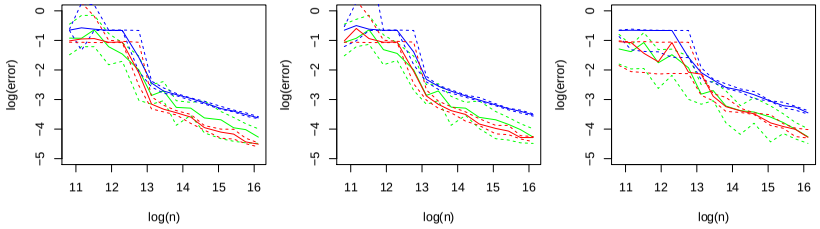
When estimating a density by taking an estimator within some class (the model), two sources of error appear: the bias, that is the (deterministic) distance between the true



(a) eachjump (b) eachjump POS (c) eachjump MAX

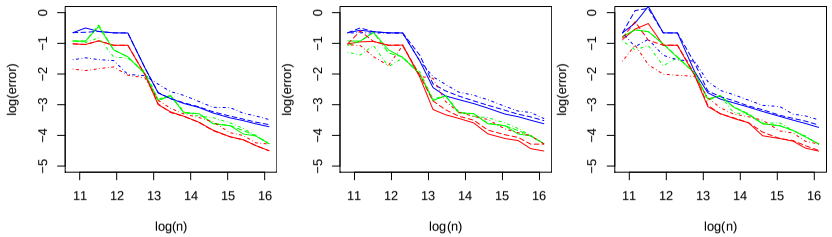


(d) jumpmean (e) jumpmean POS (f) jumpmean MAX

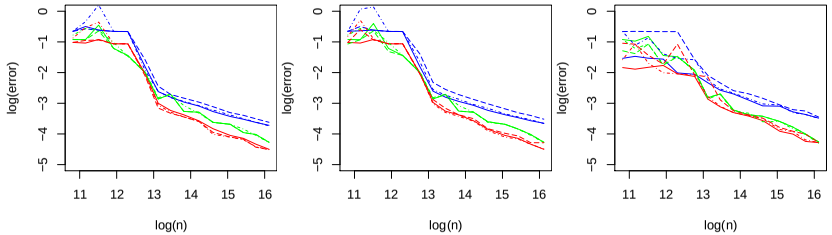


(g) jumpmax (h) jumpmax POS (i) jumpmax MAX

Figure 3: Logarithm of the $\mathbf{L}^2$ error on each emission densities depending on the logarithm of the number of observations for each of the selection and calibration methods. Each color corresponds to one emission density. The full lines are the medians of the 20 observations and the dashed ones are the 25 and 75 percentiles.



(a) eachjump summary (b) jumpmax summary (c) jumpmean summary



(d) none summary (e) POS summary (f) MAX summary

Figure 4: Superposition of the median lines of Figure 3 by selection method and by calibration variant. Each color corresponds to one emission density. In Subfigures (a)-(c), the full lines correspond to the basic selection method, the dashed ones to the POS method and the dotted ones to the MAX method. In Subfigures (d)-(f), the full lines correspond to the eachjump method, the dashed ones to the jumpmax method and the dotted ones to the jumpmean method.

density and the model, and the variance, that is the (random) error of the estimation within the model. Small models will have a large bias but a small variance, while large models will have a small bias and a large variance. The core issue of model selection is to select a model that minimizes the total error, that is large enough to accurately describe the true densities and small enough to prevent overfitting: in other words, perform a bias-variance tradeoff.

Cross validation seeks to achieve such a tradeoff by computing an estimate of the total error. This is done by splitting the sample into two sets, the training sample being used for the calibration of the estimator and the validation sample for measuring the error.

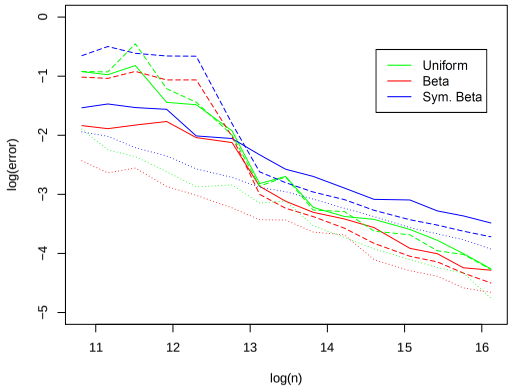


Figure 5: Comparison of the errors for the eachjump MAX method (full lines), for the eachjump method (dashed lines) and for the oracle estimators (dotted lines). For each k, the oracle estimator is defined as $\hat{f}_k^{(M_k^{\text{oracle}})}$ where M_k^{oracle} minimizes $M \longmapsto \|\hat{f}_k^{(M)} - f_k^*\|_2$. The oracle corresponds to the best estimator one could possibly select among the preliminary estimator if the true densities were known.

Taking the mean of these errors for different splits between training and validation samples provides an estimator of the total error. This method has become popular for its simplicity of use. We refer to the survey of Arlot et al. (2010) for an overview on this method and its guarantees.

4.5.1 Risk

We use the least squares criterion of Algorithm 2 to quantify the error of the estimators. Since the guarantees on spectral estimators rely on the $\mathbf{L}^2$ norm, a least squares criterion is more natural than the likelihood. In addition, the spectral estimators might take negative values depending on the orthonormal basis, which is not a problem as far as $\mathbf{L}^2$ error is concerned but can be an issue for the likelihood.

Let us first recall this criterion. Given an orthonormal basis $(\varphi_i)_{i\in\mathbb{N}}$ of $\mathbf{L}^2(\mathcal{Y},\mu)$, define the coordinate tensor of the empirical distribution of the triplet (Y_1,Y_2,Y_3) on this basis by

$$\hat{\mathbf{M}}(a,b,c) := \frac{1}{n}\sum_{s=1}^{n} \varphi_a(Y_s)\varphi_b(Y_{s+1})\varphi_c(Y_{s+2}) \quad \text{for all } a,b,c \in \mathbb{N}.$$

Given a transition matrix $\mathbf{Q}$ of size K, a stationary distribution π of $\mathbf{Q}$ and a vector of densities $\mathbf{f} = (f_1,\dots,f_K)$, define the coordinate matrix $\mathbf{O}$ of $\mathbf{f}$ by $\mathbf{O}(b,k) = \langle \varphi_b, f_k\rangle$. Let $\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{f})}$ be the coordinate tensor of the distribution of (Y_1,Y_2,Y_3) under the parameters $(\pi,\mathbf{Q},\mathbf{f})$, that is

$$\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{f})}(\cdot,b,\cdot) = \mathbf{O}\text{Diag}[\pi]\mathbf{Q}\text{Diag}[\mathbf{O}(b,\cdot)]\mathbf{Q}\mathbf{O}^\top \quad \text{for all } b \in \mathbb{N}.$$

Estimator	Convergence rate exponents		
	Uniform	Sym. Beta	Beta
Eachjump	-0.500 ± 0.046	-0.347 ± 0.007	-0.470 ± 0.015
Eachjump POS	-0.503 ± 0.047	-0.327 ± 0.008	-0.469 ± 0.015
Eachjump MAX	-0.480 ± 0.052	-0.335 ± 0.009	-0.449 ± 0.015
Jumpmean	-0.532 ± 0.048	-0.349 ± 0.006	-0.471 ± 0.017
Jumpmean POS	-0.540 ± 0.048	-0.350 ± 0.006	-0.456 ± 0.016
Jumpmean MAX	-0.493 ± 0.049	-0.374 ± 0.009	-0.437 ± 0.015
Jumpmax	-0.500 ± 0.046	-0.349 ± 0.006	-0.464 ± 0.016
Jumpmax POS	-0.492 ± 0.046	-0.358 ± 0.006	-0.442 ± 0.015
Jumpmax MAX	-0.480 ± 0.052	-0.404 ± 0.009	-0.466 ± 0.015
Cross Validation	-0.434 ± 0.007	-0.263 ± 0.011	-0.377 ± 0.008
Oracle	-0.517 ± 0.048	-0.360 ± 0.006	-0.459 ± 0.017
Hidden states known	-0.526 ± 0.031	-0.293 ± 0.005	-0.428 ± 0.007
Minimax (Hölder)	-0.5	$-3/11 \approx -0.273$	$-3/7 \approx -0.429$

Table 1: Exponents of the rates of convergence for the different algorithms. The rates are obtained from a linear regression with the relation $\log(\|\hat{f}_k - f_k^*\|_2) \sim \log(n)$ for the estimators $\hat{f}_k$ computed with $n \geq 700,000$ observations ($n \geq 1,000,000$ for the cross validation estimators from Section 4.5). The smaller the exponent, the faster the estimators converge. The line "hidden states known" is obtained by density estimation when the hidden states are observed.

The empirical least squares criterion is $\|\mathbf{M}_{(\pi,\mathbf{Q},\mathbf{f})} - \hat{\mathbf{M}}\|_F^2$. It corresponds to the $\mathbf{L}^2$ error between the empirical distribution of three consecutive observations and the theoretical distribution under the parameters $(\pi,\mathbf{Q},\mathbf{f})$.

4.5.2 Implementation

We use 10-fold cross validation, that is we split the sequence into 10 segments of same size $I_1,\dots,I_{10}$. In order to avoid interferences between samples, we prune the ends of each segment, so that the observations in each segment can be considered independent. In practice, we take a gap of 30 observations between two segments.

We ran 150 simulations, 10 per value of n, with the same parameters as in Section 4.1. Each simulation is as follows.

For each segment I_j, we run the spectral algorithm on all models $\mathfrak{P}_M$ for `M_min` $\leq M \leq$ `M_max` using only the observations from the other segments. The transition matrix is estimated using an additional step of the spectral method which is adapted from Steps 8 and 9 of Algorithm 1 of De Castro et al. (2017). Then, we compute the least squares

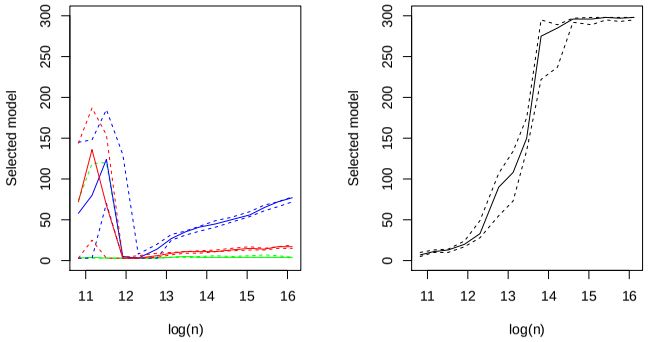


Figure 6: Selected model dimensions for each n using our state-by-state selection method (left) and 10-fold cross validation (right). The full lines are the median model dimensions and the dashed lines are the 25 and 75 percentiles.

criterion for the estimated parameters using the segment I_j as observed sample. Finally, for each M, we average this error on all segments I_j, which gives the least squares cross validation error $E_{\mathrm{VC}}(M)$.

This cross validation criterion is used to select one model $\hat{M}_{\mathrm{VC}} \in \arg\min_M E_{\mathrm{VC}}(M)$, from which we construct the final estimators of the emission densities $\hat{f}_k = \hat{f}_k^{(\hat{M}_{\mathrm{VC}})}$ for all k. Note that the selected model is the same for all emission densities.

4.5.3 Results

Figure 6 compares the selected model dimensions for each n using our state-by-state selection method and using the cross validation method. When the number of observations n becomes larger than 10^6, the cross validation tends to always pick the largest model, which means that it does not prevent overfitting as well as our method.

The $\mathbf{L}^2$ errors on the emission densities are shown in Figure 7. It appears that the cross validation has a lower error for small n ($n \leq 350,000$) than our method. However, for larger values of n, the errors becomes larger than the ones of our method (see Figure 5) by up to one order of magnitude, and only start decreasing once the selected model is set to the maximum dimension.

Finally, the estimated rates of convergence are shown in Table 1. Our state-by-state method outperforms the cross validation method for all emission densities. The cross validation estimators only reach the minimax rate of convergence for the less regular density: the symmetrized Beta, and even then they converge slower than the state-by-state estimator. All other emission densities are estimated slower than their minimax rate.

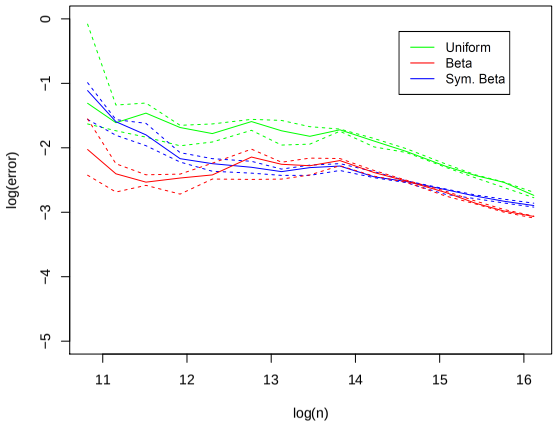


Figure 7: Error of the cross validation estimators for each n using 10-fold cross validation. The full lines are the median errors for each density and the dashed lines are the 25 and 75 percentiles.

4.6 Algorithmic Complexity

In the following, we treat K as a constant as far as the algorithmic complexity is concerned. The different complexities are summarized in Table 2.

4.6.1 Spectral Algorithm (see Section 3.2)

We consider the algorithmic complexity of estimating the emission densities for all models M such that $M_{\min} \leq M \leq M_{\max}$ with n observations and auxiliary parameters r and m depending on n and M (upper bounded by $m_{\max}$ and $r_{\max}$).

Step 1 can be computed for all models with $O(nM_{\max}m_{\max}^2)$ operations. It is the only step whose complexity depends on n. Steps 2 and 3 require $O(m^3M)$ operations for each model and Steps 4 to 7 require $O(Mr)$ operations for each model, for a total of $O(nM_{\max}m_{\max}^2 + M_{\max}^2 m_{\max}^3 + M_{\max}^2 r_{\max})$ operations.

In practice, one takes $m \propto \log(n)$, $r \propto \log(n) + \log(M)$ and $M_{\max} \leq n$, so that the total complexity of the spectral algorithm is $O(n \log(n)^2 M_{\max})$.

In comparison, the complexity of the spectral algorithm of De Castro et al. (2017) is $O(nM_{\max}^3)$ because of Step 1. This becomes much larger than our complexity when $M_{\max}$ grows as a power of n (which is necessary in order to reach minimax rates).

4.6.2 Least Squares Algorithm (see Section 3.3)

We consider the algorithmic complexity of estimating the emission densities for all models M such that $M_{\min} \leq M \leq M_{\max}$ with n observations.

Step 1 is similar to the one of the spectral algorithm, but with $O(nM_{\max}^3)$ operations. The complexity of Step 2 is more difficult to evaluate. Since the criterion is nonconvex, finding the minimizer requires to run an approximate minimization algorithm whose complexity C_n will depend on the desired precision–which will in turn depend on the number of observations n–and on the initial points. As discussed in Lehéricy (to appear), this is usually the longest step when computing least squares estimators. Thus, the total complexity of the least squares algorithm is $O(nM_{\max}^3 + C_n)$.

Note that despite the worse sample complexity, the least squares algorithm is tractable and can greatly improve the estimation for small sample size. As shown in Section 4.4, the spectral algorithm is unstable for small samples, which makes the state-by-state selection procedure return abnormal results. This can be explained by the matrix inversions of the spectral method, which sometimes lead to nearly singular matrices when the noise is too large. On the other hand, the least squares method does not involve any matrix inversion, and often gives better results than the spectral estimators, as shown in De Castro et al. (2016), thus making it a relevant choice for small to medium data sets.

4.6.3 Selection Method and POS Variant (see Sections 2.2 and 4.3)

We consider the algorithmic complexity of selecting estimators from a family of estimators $(\hat{\mathbf{f}}^{(M)})_{M_{\min} \leq M \leq M_{\max}}$. The selection algorithms can be decomposed in two parts.

- Compute the distances $\|\hat{f}_k^{(M)} - \hat{f}_k^{(M')}\|_2$ for all M, M' and k. This has complexity $O(M_{\max}^3)$: it requires to compute the $\mathbf{L}^2$ distance of at most $M_{\max}^2$ couples of functions in a Hilbert space of dimension $M_{\max}$.

- Compute $\hat{\rho}_k$ defined as the abscissa of the largest jump of the function $\rho \longmapsto \hat{M}_k(\rho)$ for all k, where $\hat{M}_k$ is defined as in Section 4.2. Note that computing $\hat{M}_k(\rho)$ requires $O(M_{\max}^2)$ operations. An approximate value of $\hat{\rho}_k$ can be computed in $O(\log(\hat{\rho}_k)M_{\max}^2)$ operations, which is usually $O(M_{\max}^2)$.

Once the $\hat{\rho}_k$ are known, it is possible to calibrate the penalty in constant time for the three calibrations methods (eachjump, jumpmax and jumpmean) and to select the final models in $O(M_{\max}^2)$ operations.

Thus, the total complexity of the selection algorithm and of its POS variant is $O(M_{\max}^3)$.

4.6.4 Selection Method, MAX Variant (see Section 4.3)

In the MAX variant, the first step of the standard selection procedure is replaced by computing the distances $\|\hat{f}_k^{(M_{\max})} - \hat{f}_k^{(M)}\|_2$ for all M. This has complexity $O(M_{\max}^2)$. The complexity of the other steps remains unchanged.

Thus, the total complexity of the MAX variant of the selection algorithm is $O(M_{\max}^2)$.

	Algorithm	Complexity
Preliminary estimators	Spectral method	$O(n\log(n)^2 M_{\max})$
	Spectral method (De Castro et al. (2017))	$O(nM_{\max}^3)$
	Least squares method	$O(nM_{\max}^3 + C_n)$
Selection step	Standard and POS variant	$O(M_{\max}^3)$
	MAX variant	$O(M_{\max}^2)$

Table 2: Complexities of the different algorithms. n is the number of observations, $M_{\max}$ is the largest model dimension considered.

5. Application to Real Data

In this section, we present the results of our method on two sets of trajectories. Trajectories are a typical example of dependent data that shows several behaviours depending on the activity of the entity being tracked, which makes hidden Markov models a popular modelling choice. For instance, the movement of a fisher is not the same depending on whether he's travelling to the next fishing zone or actually fishing.

The first data set follows artisanal fishers in Madagascar. The second one contains seabird movements. Studying the movements of fishers and seabirds has many applications, for instance understanding the fishing habits of the tracked entity, controlling the fishing pressure on local ecosystems and monitoring the dynamics of coastal ecosystems, see for instance Boyd et al. (2014); Vermard et al. (2010) and references therein.

5.1 Artisanal Fishery

We use GPS tracks of artisanal fishers with a regular sampling period of 30 seconds. These tracks were produced by Faustinato Behivoke (Institut Halieutiques et des Sciences Marines, Université de Toliara, Madagascar) and Marc Léopold (IRD), who recorded artisanal fishers from Ankilibe, in Madagascar. Their fishing method is a seine netting.

From this data, we compute the velocity of the fisher during each time step. In order to estimate densities on $[0,1]$, we divide this velocity by an upper bound of the maximum observed velocity. We consider the observation space $\mathcal{Y} = [0,1]$ endowed with the dominating measure $\delta_0 + \text{Leb}$, where δ_0 is the dirac measure in zero and Leb is the Lebesgue measure on $[0,1]$. As a proof of concept, we use the orthonormal basis consisting of the trigonometric basis on $[0,1]$ and the indicator function of $\{0\}$, that is the family $(\varphi_m)_{m\in\mathbb{N}}$ defined on $[0,1]$

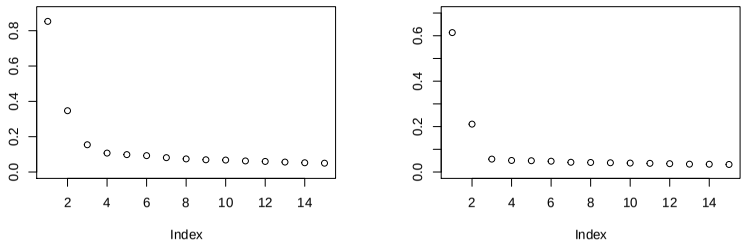


Figure 8: First 15 eigenvalues of the spectrum of the empirical tensor $\hat{\mathbf{N}}_{50,50}$ (see Algorithm 3 in Appendix A). Left: fisher 1, right: fisher 2.

by

$$\text{if } x = 0, \begin{cases} \varphi_0(x) = 1 \\ \varphi_m(x) = 0 \quad \text{for all } m \in \mathbb{N}^* \end{cases}$$

$$\text{if } x \neq 0, \begin{cases} \varphi_0(x) = 0 \\ \varphi_1(x) = 1 \\ \varphi_{2m}(x) = \sqrt{2}\cos(2\pi m x) \quad \text{for all } m \in \mathbb{N}^* \\ \varphi_{2m+1}(x) = \sqrt{2}\sin(2\pi m x) \quad \text{for all } m \in \mathbb{N}^* \end{cases}$$

The number of hidden states is chosen using the spectral thresholding method of Lehéricy (to appear). This methods consists is based on the fact that the rank of the spectral tensor $\mathbb{E}\hat{\mathbf{N}}_{m,m}$ (with the notations of Algorithm 3 in Appendix A) is the number of hidden states. This is visible in the spectrum of $\hat{\mathbf{N}}_{m,m}$ by an elbow, as shown in Figure 8. Based on these spectra, we use two hidden states.

The results using `M_max` $= 1000$ are shown in Figures 9 and 10. We took the normalizing velocity large enough that all observed normalized velocities belong to $[0, 0.8]$, hence the plot betweeen 0 and 0.8 for the densities.

In both cases, the selected model complexities differ greatly depending on the state. This comes from the fact that in both cases, one of the density is spiked, thus requiring more vectors of the orthonormal basis to be approximated. This illustrates that our method is able to estimate the smoother densities with fewer vectors of the basis, thus preventing overfitting.

As a side note, we needed considerably less observations than in the simulations: around 10,000, compared to 500,000 in the simulations. This can be explained by the fact that each state is very stable, with an estimated probability of leaving the states below 0.02–compared to 0.3 in the simulations. This is encouraging, as hidden states in real data are expected to be rather stable, especially when the sampling frequency is high, as long as the conditional independance of the observations can be assumed to hold.

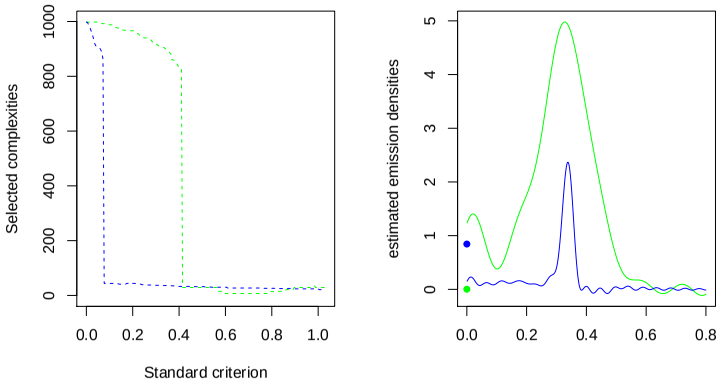


Figure 9: Selected complexities and estimated densities on artisanal fishery data (fisher 1, $n = 17,300$). Green = state 1, blue = state 2. The dirac component is shown as a dot at $y = 0$. The selected dimensions are $(14, 41)$.

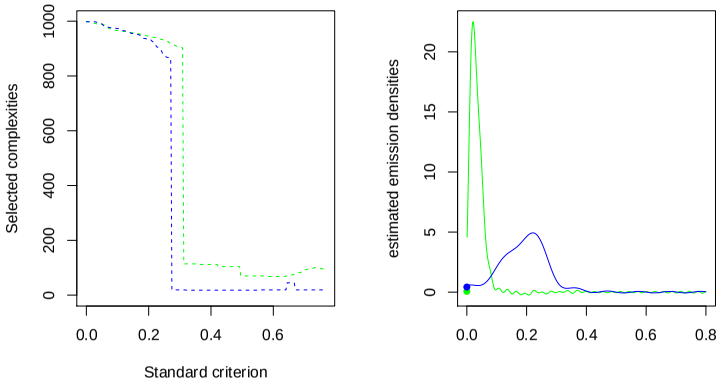


Figure 10: Selected complexities and estimated densities on artisanal fishery data (fisher 2, $n = 11,600$). Green = state 1, blue = state 2. The dirac component is shown as a dot at $y = 0$. The selected dimensions are $(68, 18)$.

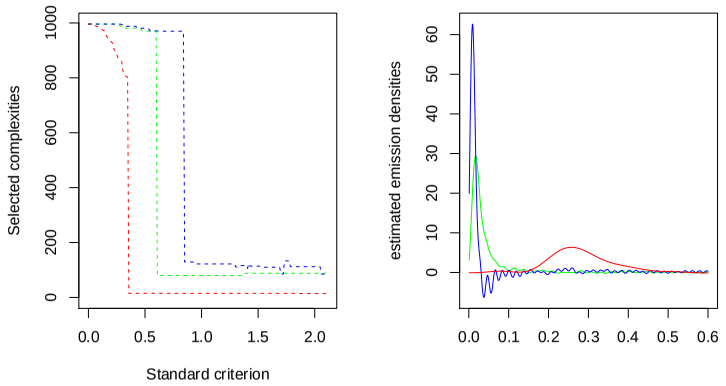


Figure 11: Selected complexities and estimated densities for Cormorant d's trajectory ($n = 2,891$). Green = state 1, blue = state 2, red = state 3. The selected dimensions are $(80, 110, 15)$.

5.2 Seabird Foraging

In this Section, we consider the seabird data from Bertrand et al. (2015) and we focus on the tracks named cormorant d in this paper.

We apply the same transformation as in the previous section to obtain normalized velocities in $[0, 0.8]$ (after removal of anomalous velocities exceeding 150 m/s) and run the spectral algorithm with the trigonometric basis on $[0, 1]$ plus the indicator of $\{0\}$. The spectral thresholding gives a number of hidden states equal to two; we set it to three to account for more complex behaviours of the seabirds. The results are shown in Figure 11.

Note that the use of the trigonometric basis allows the estimated densities to take negative values. This is not a problem as far as minimax rates of convergence (in $\mathbf{L}^2$ norm) are concerned, however this can become an issue if one wants to use these densities in a forward-backward algorithm in order to get an estimator of the hidden states. One way to circumvent this problem is to use simplex projection to compute an approximation of the projection of these estimated density on the simplex of all probability densities. Note that since this is an $\mathbf{L}^2$ projection on a convex set which contains the true densities, the projected densities have an even smaller error, thus keeping the minimax rate of convergence of the original estimators. The resulting densities are shown in Figure 12

The number of observations in this setting is even smaller than for the fishery's data set: our algorithm was able to recover three emission densities from less than 3,000 observations, despite the states being less stable than in the fishery data set: the diagonal terms of the estimated transition matrix using the EM algorithm are $(0.83, 0.93, 0.98)$. In addition, the result of our method is consistent with other estimation methods, as shown in Figure 12:

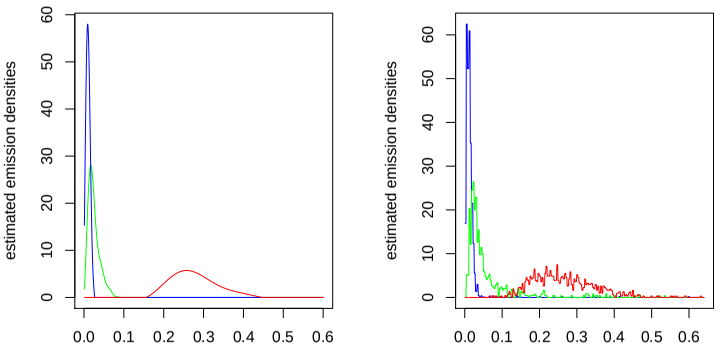


Figure 12: Projection of the estimated densities of Figure 11 for Cormorant d on the set of probability densities (left) and comparison with an estimation with histogram densities on a regular partition of size 300 using the EM algorithm (right).

estimating the parameters with the EM algorithm using piecewise constant densities leads to a very similar result.

6. Conclusion and Perspectives

We propose a state-by-state selection method to infer the emission densities of a HMM. Using a family of estimators, our method selects one estimator for each hidden state in a way that is adaptive with respect to this state's regularity. This method does not depend on the type of preliminary estimator, as long as a suitable variance bound is available. As such, it may be seen as a plug-in that takes a family of estimators and the corresponding variance bound and outputs the selected estimator. Note that its complexity does not depend on the number of observations used to compute the estimators, which makes it applicable to arbitrarily large data sets.

To apply this method, we present two families of estimators: a least squares estimator and a spectral estimator. For both, we prove a bound on their variance and show that this bound allows to recover the minimax rate of convergence separately on each hidden state, up to a logarithmic factor. The variance bounds are similar to a BIC penalty, with an additional logarithmic factor for the spectral estimators.

We carry out a numerical study of the method and some variants on simulated data. We use the spectral estimators, which are both fast and don't suffer from initialization issues, unlike the least squares and maximum likelihood estimators. The simulations show that

our selection method is very fast compared to the computation of the estimators and that indeed, the final estimators reach the minimax rate of convergence on each state.

Then, we compare our method with a cross validation estimator based on a least square risk. This estimator only reaches the minimax rate corresponding to the worst regularity among the emission densities and fails to select models with small dimensions. It is still noteworthy that the cross validation returns relevant results for small sample sizes, whereas our method requires the sample size to be large enough to work properly. An interesting problem would be to investigate whether cross validation or other methods can be combined with our state-by-state selection method to give an algorithm that is both fast, stable for small sample sizes and optimal in the asymptotic setting.

Finally, we apply our algorithm to real trajectory data sets. On this data, our method proves that it is able to match the regularity of the underlying emission densities. In addition, it is able to produce sensible results with far fewer observations than in our simulation study.

Our state-by-state selection method can be easily applied to multiview mixture models (also named mixture models with repeated measurement, see for instance Bonhomme et al. (2016a) and Gassiat et al. (2018)). Let us first describe the model. A multiview mixture model consists of two random variables, a hidden state U and an observation vector $\mathbf{Y} := (Y_i)_{i\in[m]}$ such that conditionally to U, the components Y_i of $\mathbf{Y}$ are independent with a distribution depending only on U and i. Let us assume that U takes its values in a finite set $\mathcal{X}$ of size K and that the Y_i have some density $f^*_{u,i}$ conditionally to $U = u$ with respect to a dominating measure. A question of interest is to estimate the densities $f^*_{u,i}$ from a sequence of observed $(\mathbf{Y}_n)_{n\geq 1}$.

Our state-by-state selection method can be applied directly to such a model as long as estimators with a proper variance bound are available (see assumption $[\mathbf{H}(\epsilon)]$ in Section 2.1). Indeed, we never use the dependency structure of the model. Regarding the development of preliminary estimators, multiview mixture models appear closely related to hidden Markov models: Anandkumar et al. (2012) and Bonhomme et al. (2016b) develop spectral methods that work for both multiview mixtures and HMMs at the same time using the same theoretical arguments. Thus, it seems clear that variance bounds such as the ones we developed can also be written for multiview mixture models.

Acknowledgments

I am grateful to Elisabeth Gassiat and Claire Lacour for their precious advice. I thank Augustin Touron for providing me with a R implementation of the spectral algorithm. I would also like to thank Marie-Pierre Etienne and of course Faustinato Behivoke (Institut Halieutiques et des Sciences Marines, Université de Toliara, Madagascar), Marc Léopold (IRD) and Sophie Bertrand (IRD) for letting me work on their data sets.

Appendix A. Spectral Algorithm, Full Version

Algorithm 3: Spectral estimation of the emission densities of a HMM (full version)

Data: A sequence of observations $(Y_1, \dots, Y_{n+2})$, two dimensions $m \leq M$, an orthonormal basis $(\varphi_1, \dots, \varphi_M)$ and number of retries r.

Result: Spectral estimators $(\hat{f}_k^{(M,r)})_{k\in\mathcal{X}}$, $\hat{\mathbf{Q}}$ and $\hat{\pi}$.

[Step 1] Consider the following empirical estimators: for any $a, c \in [m]$ and $b \in [M]$,

- $\hat{\mathbf{L}}_m(a) := \frac{1}{n}\sum_{s=1}^n \varphi_a(Y_s)$
- $\hat{\mathbf{M}}_{m,M,m}(a,b,c) := \frac{1}{n}\sum_{s=1}^n \varphi_a(Y_s)\varphi_b(Y_{s+1})\varphi_c(Y_{s+2})$
- $\hat{\mathbf{N}}_{m,M}(a,b) := \frac{1}{n}\sum_{s=1}^n \varphi_a(Y_s)\varphi_b(Y_{s+1})$
- $\hat{\mathbf{P}}_{m,m}(a,c) := \frac{1}{n}\sum_{s=1}^n \varphi_a(Y_s)\varphi_c(Y_{s+2})$.

[Step 2] Let $\hat{\mathbf{U}}_m$ be the $m \times K$ matrix of orthonormal left singular vectors and $\hat{\mathbf{V}}_M$ be the $M \times K$ matrix of orthonormal right singular vectors of $\hat{\mathbf{N}}_{m,M}$ corresponding to its top K singular values.

[Step 3] Form the matrices for all $b \in [M]$, $\hat{\mathbf{B}}(b) := (\hat{\mathbf{U}}_m^\top \hat{\mathbf{P}}_{m,m}\hat{\mathbf{U}}_m)^{-1}\hat{\mathbf{U}}_m^\top \hat{\mathbf{M}}_{m,M,m}(.,b,.)\hat{\mathbf{U}}_m$.

[Step 4] Set $(\Theta_i)_{1\leq i\leq r}$ r iid $(K \times K)$ unitary matrix uniformly drawn. Form the matrices for all $k \in \mathcal{X}$ and $i \in [r]$, $\hat{\mathbf{C}}_i(k) := \sum_{b=1}^M (\hat{\mathbf{V}}_M\Theta_i)(b,k)\hat{\mathbf{B}}(b)$.

[Step 5] Compute $\hat{\mathbf{R}}_i$ a $(K \times K)$ unit Euclidean norm columns matrix that diagonalizes the matrix $\hat{\mathbf{C}}_i(1)$: $\hat{\mathbf{R}}_i^{-1}\hat{\mathbf{C}}_i(1)\hat{\mathbf{R}}_i = \mathrm{Diag}(\hat{\Lambda}_i(1,1), \dots, \hat{\Lambda}_i(1,K))$.

[Step 6] Set for all $k, k' \in \mathcal{X}$, $\hat{\Lambda}_i(k,k') := (\hat{\mathbf{R}}_i^{-1}\hat{\mathbf{C}}_i(k)\hat{\mathbf{R}}_i)(k',k')$. Choose i_0 maximizing $\min_k \min_{k_1\neq k_2} |\hat{\Lambda}_i(k,k_1) - \hat{\Lambda}_i(k,k_2)|$ and set $\hat{\mathbf{O}} := \hat{\mathbf{V}}_M\Theta_{i_0}\hat{\Lambda}_{i_0}$.

[Step 7] Consider the emission densities estimators $\hat{\mathbf{f}}^{(M,r)} := (\hat{f}_k^{(M,r)})_{k\in\mathcal{X}}$ defined by for all $k \in \mathcal{X}$, $\hat{f}_k^{(M,r)} := \sum_{b=1}^M \hat{\mathbf{O}}(b,k)\varphi_b$.

[Step 8] Let $\hat{\mathbf{O}}_m$ be the $m \times K$ matrix containing the first m rows of $\hat{\mathbf{O}}$. Set $\hat{\pi} = \Pi_\Delta\left((\hat{\mathbf{U}}_m^\top\hat{\mathbf{O}}_m)^{-1}\hat{\mathbf{U}}_m^\top\hat{\mathbf{L}}_m\right)$ where Π_Δ is the $\mathbf{L}^2$ projection onto the probability simplex.

[Step 9] Let $\hat{\mathbf{Q}}$ be the transition matrix defined by $\hat{\mathbf{Q}} = \Pi_{\mathrm{TM}}\left((\hat{\mathbf{U}}_m^\top\hat{\mathbf{O}}_m\mathrm{Diag}[\hat{\pi}])^{-1}\hat{\mathbf{U}}_m^\top\hat{\mathbf{N}}_{m,M}\hat{\mathbf{V}}_M(\hat{\mathbf{O}}^\top\hat{\mathbf{V}}_M)^{-1}\right)$ where Π_{TM} is the projection onto the set of transition matrices. This projection is obtained by projecting each line of the matrix onto the probability simplex.

Appendix B. Proofs

B.1 Proof of Lemma 1

Let $\tau_{n,M}$ be the permutation that minimizes $\tau \longmapsto \max_{k\in\mathcal{X}} \left\|\hat{f}_k^{(M)} - f^{*,(M)}_{\tau(k)}\right\|_2$. $[\mathbf{H}(\epsilon)]$ means that with probability $1-\epsilon$, one has $\max_{k\in\mathcal{X}} \left\|\hat{f}_k^{(M)} - f^{*,(M)}_{\tau(k)}\right\|_2 \leq \frac{\sigma(M)}{2}$.

Let $M \in \mathcal{M}$. Let us show that $\left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k')} - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2 > \left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k)} - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2$ for all $k, k' \in \mathcal{X}$ such that $k' \neq k$. If this holds, then the definition of $\hat{\tau}^{(M)}$ implies that $\hat{\tau}^{(M)} = \tau^{-1}_{n,M} \circ \tau_{n,M_0}$. Thus, one has $\max_{k\in\mathcal{X}} \left\|\hat{f}^{(M)}_{k,\text{new}} - f^{*,(M)}_{\tau_{n,M_0}(k)}\right\|_2 \leq \frac{\sigma(M)}{2}$, which is exactly Equation (1) with $\tau_n = \tau_{n,M_0}$.

Applying the triangular inequality leads to

$$\begin{aligned}\left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k)} - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2 \leq& \left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k)} - f^{*,(M)}_k\right\|_2 + \left\|f^{*,(M)}_k - f^{*,(M_0)}_k\right\|_2 + \left\|f^{*,(M_0)}_k - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2\\ \leq& \frac{\sigma(M)}{2} + B_{M,M_0} + \frac{\sigma(M_0)}{2}\end{aligned}$$

and

$$\begin{aligned}\left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k')} - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2 \geq& \left\|f^{*,(M_0)}_{k'} - f^{*,(M_0)}_k\right\|_2 - \left\|\hat{f}^{(M)}_{\tau^{-1}_{n,M}(k')} - f^{*,(M)}_{k'}\right\|_2\\ &- \left\|f^{*,(M)}_{k'} - f^{*,(M_0)}_{k'}\right\|_2 - \left\|f^{*,(M_0)}_k - \hat{f}^{(M_0)}_{\tau^{-1}_{n,M_0}(k)}\right\|_2\\ \geq& m(\mathbf{f}^*, M_0) - \frac{\sigma(M)}{2} - B_{M,M_0} - \frac{\sigma(M_0)}{2}.\end{aligned}$$

Thus, the result holds as soon as $m(\mathbf{f}^*, M_0) - \frac{\sigma(M)}{2} - B_{M,M_0} - \frac{\sigma(M_0)}{2} > \frac{\sigma(M)}{2} + B_{M,M_0} + \frac{\sigma(M_0)}{2}$, which is the condition of Lemma 1.

B.2 Proof of Theorem 5

The structure of the proof is the same as the one of Theorem 3.1 of De Castro et al. (2017).

The first difference lies in the fact that we consider different models for each component of the tensors $\hat{\mathbf{N}}_{m,M}$ and $\hat{\mathbf{M}}_{m,M,m}$ in Step 1. As a consequence, we use the left *and* right singular vectors of $\hat{\mathbf{N}}_{m,M}$ instead of just the right singular vectors of $\hat{\mathbf{P}}_{m,m}$. A careful reading shows that their proof can be adapted straightforwardly to this situation.

The second difference consists in generating several independant random unitary matrices in Step 4 and keeping the one that separates the eigenvalues of all $\hat{\mathbf{C}}_i(k)$ best. This allows to replace Lemma F.6 of De Castro et al. (2017) by the following one, based on the independence of the unitary matrices:

Lemma 14 *For all $x > 0$ and $r \in \mathbb{N}^*$,*

$$\mathbb{P}\left[\forall k, k_1 \neq k_2, |\hat{\Lambda}_{i_0}(k, k_1) - \hat{\Lambda}_{i_0}(k, k_2)| \geq \frac{2e^{-x/r}(1 - \epsilon^2_{\mathbf{N}_{m,M}})^{1/2}}{\sqrt{e}K^{5/2}(K-1)}\gamma(\mathbf{O}_M)\right] \geq 1 - e^{-x}$$

and

$$\mathbb{P}\left[\|\hat{\Lambda}_{i_0}\|_\infty \geq \frac{1 + \sqrt{2}\sqrt{x + \log(K^2 r)}}{\sqrt{K}}\|\mathbf{O}_M\|_{2,\infty}\right] \leq e^{-x},$$

The notations $\epsilon_{\mathbf{N}_{m,M}}$ (or $\epsilon_{\mathbf{P}_M}$ in the original proof), $\gamma(\mathbf{O}_M)$ et $\|\mathbf{O}_M\|_{2,\infty}$ are introduced in De Castro et al. (2017).

Using this lemma, their proof leads to our result by taking $r = x = t$.

B.3 Definition of the Polynomial H

B.3.1 Definition

We parameterize the application

$$(\pi, \mathbf{Q}, \mathbf{f}) \in \Delta \times \mathcal{Q} \times \text{Span}(\mathbf{f}^*)^K \longmapsto \|g^{\pi,\mathbf{Q},\mathbf{f}} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2 \tag{3}$$

in the following way. For $p \in \mathbb{R}^{K-1}$, $q \in \mathbb{R}^{K\times(K-1)}$ and $A \in \mathbb{R}^{K\times(K-1)}$, define the extensions

- $\bar{p} \in \mathbb{R}^K$ defined by $\bar{p}(k) = p(k)$ for all $k \in [K-1]$ and $\bar{p}(K) = -\sum_{k\in[K-1]} p(k)$;
- $\bar{q} \in \mathbb{R}^{K\times K}$ by $\bar{q}(k, K) = -\sum_{k'\in[K-1]} q(k, k')$;
- $\bar{A} \in \mathbb{R}^{K\times K}$ by $\bar{A}(k, K) = -\sum_{k'\in[K-1]} A(k, k')$.

$\bar{p}$ corresponds to $\pi - \pi^*$, $\bar{q}$ to $\mathbf{Q} - \mathbf{Q}^*$ and A to the components of $\mathbf{f} - \mathbf{f}^*$ on $\mathbf{f}^*$ (which is a basis as soon as **[Hid]** holds). The condition on the last component of $\bar{p}$ and of each line of $\bar{q}$ and $\bar{A}$ follows from the fact that $\bar{p}$ corresponds to the difference of two probability vectors, $\bar{q}$ corresponds to the difference of two transition matrices and $\bar{A}$ correspond to the difference of two vectors of probability densities on a basis of probability densities.

Then, consider the quadratic form derived from the Taylor expansion of

$$(p, q, A) \in \mathbb{R}^{K-1} \times \mathbb{R}^{K\times(K-1)} \times \mathbb{R}^{(K-1)\times K} \longmapsto \|g^{\pi^*+\bar{p},\mathbf{Q}^*+\bar{q},\mathbf{f}+\bar{A}\mathbf{f}^*} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2.$$

Let M be the matrix associated to this quadratic form. We define H as the determinant of M. Direct computations show that H is a polynomial in the coefficients of π^*, $\mathbf{Q}^*$ and $G(\mathbf{f}^*)$.

B.3.2 Link between H and the Quadratic Form from Equation (3)

The goal of this section is to show how H can be used to lower bound the quadratic form from Equation (3) by a positive constant times the distance between $(\pi, \mathbf{Q}, \mathbf{f})$ and $(\pi^*, \mathbf{Q}^*, \mathbf{f}^*)$. We will not need the assumptions **[Hid]**, **[HF]** or **[Hdet]** unless specified otherwise.

Let us start by the relation between the norms of (p, q, A) and $(\bar{p}, \bar{q}, \bar{A})$.

Lemma 15 *For all $(p, q, A) \in \mathbb{R}^{K-1} \times \mathbb{R}^{K\times(K-1)} \times \mathbb{R}^{(K-1)\times K}$,*

$$\begin{aligned}\|p\|_2^2 \leq \|\bar{p}\|_2^2 \leq K\|p\|_2^2,\\ \|q\|_F^2 \leq \|\bar{q}\|_F^2 \leq K\|q\|_F^2,\\ \|A\|_F^2 \leq \|\bar{A}\|_F^2 \leq K\|A\|_F^2.\end{aligned}$$

Proof $\|p\|_2^2 \leq \|\bar{p}\|_2^2$ is immediate. Then,

$$\begin{aligned}\|\bar{p}\|_2^2 =& \|p\|_2^2 + \left(\sum_{k\in[K-1]} p(k)\right)^2\\ \leq& \|p\|_2^2 + (K-1)\sum_{k\in[K-1]} p(k)^2\\ =& K\|p\|_2^2.\end{aligned}$$

The proof is the same for q and A. ■

The next lemma will be used to link the norms of A and $A\mathbf{f}$.

Lemma 16 *For all $\bar{A} \in \mathbb{R}^{K\times K}$ and $\mathbf{f}^* \in (\mathbf{L}^2(\mathcal{Y},\mu))^K$,*

$$\sigma_K(G(\mathbf{f}^*))\|\bar{A}\|_F^2 \leq \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_2^2 \leq K\|G(\mathbf{f}^*)\|_\infty \|\bar{A}\|_F^2$$

Proof For the first inequality, we use that for all $k \in \mathcal{X}$,

$$\begin{aligned}\|(\bar{A}\mathbf{f}^*)_k\|_2^2 =& \bar{A}(k,\cdot)G(\mathbf{f}^*)\bar{A}(k,\cdot)^\top \\ \geq& \sigma_K(G(\mathbf{f}^*))\|\bar{A}(k,\cdot)\|_2^2\end{aligned}$$

and the inequality follows by summing over k.

For the second inequality,

$$\begin{aligned}\sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_2^2 &= \sum_{k\in[K]} \int (\bar{A}\mathbf{f}^*)_k(x)^2\mu(dx) \\ &= \sum_{k\in[K]} \int \left(\sum_{j\in[K]} \bar{A}(k,j) f_j^*(x)\right)^2 \mu(dx) \\ &\leq \sum_{k\in[K]} \int K \sum_{j\in[K]} \bar{A}(k,j)^2 (f_j^*)^2(x)\mu(dx) \\ &\leq K\left(\sum_{k,j\in[K]} \bar{A}(k,j)^2\right) \sup_{j\in\mathcal{X}} \int (f_j^*)^2(x)\mu(dx) \\ &= K\|\bar{A}\|_F^2 \|G(\mathbf{f}^*)\|_\infty.\end{aligned}$$

■

Finally, we will use the following result of Lehéricy (to appear) (Section B.2) in order to upper bound the spectrum of the matrix M.

Lemma 17 *For all $\pi_1,\pi_2 \in \Delta$, for all $\mathbf{Q}_1,\mathbf{Q}_2 \in \mathcal{Q}$ and for all $\mathbf{f}_1,\mathbf{f}_2 \in (\mathbf{L}^2(\mathcal{Y},\mu))^K$,*

$$\|g^{\pi_1,\mathbf{Q}_1,\mathbf{f}_1} - g^{\pi_2,\mathbf{Q}_2,\mathbf{f}_2}\|_2 \leq \sqrt{3K(\|G(\mathbf{f}_1)\|_\infty^3 \vee \|G(\mathbf{f}_2)\|_\infty^3)}\, d_{perm}((\pi_1,\mathbf{Q}_1,\mathbf{f}_1),(\pi_2,\mathbf{Q}_2,\mathbf{f}_2))$$

Together, these results imply that for all (p,q,A),

$$\begin{aligned}\|g^{\pi^*+\bar{p},\mathbf{Q}^*+\bar{q},\mathbf{f}^*+\bar{A}\mathbf{f}^*} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2 & \\ \leq& 3K(\|G(\mathbf{f}^*+\bar{A}\mathbf{f}^*)\|_\infty^3 \vee \|G(\mathbf{f}^*)\|_\infty^3)(\|\bar{p}\|_2^2 + \|\bar{q}\|_F^2 + \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f})_k\|_F^2) \\ \leq& 3K\|G(\mathbf{f}^*)\|_\infty^3 (1+K^2\|A\|_F^2)^3 (K\|p\|_2^2 + K\|q\|_F^2 + K^2\|G(\mathbf{f}^*)\|_\infty \|A\|_F^2)\end{aligned}$$

so that $\sigma_1(M) \leq \sqrt{3K^3}(1 \vee \|G(\mathbf{f})\|_\infty^2)$. Since $H = \prod_{i=1}^{(K-1)(2K+1)} \sigma_i(M)$, one has

$$\sigma_{(K-1)(2K+1)}(M) \geq \frac{H}{(3K^3(1\vee\|G(\mathbf{f})\|_\infty^4))^{K^2-K/2}}.$$

Now, assume that **[Hid]** holds, so that $\sigma_K(G(\mathbf{f}^*)) > 0$, then

$$\begin{aligned}\|g^{\pi^*+\bar{p},\mathbf{Q}^*+\bar{q},\mathbf{f}^*+\bar{A}\mathbf{f}^*} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2 \geq& \sigma_{(K-1)(2K+1)}(M)(\|p\|_2^2 + \|q\|_F^2 + \|A\|_F^2) \\ &+ o(\|p\|_2^2 + \|q\|_F^2 + \|A\|_F^2) \\ \geq& \frac{\sigma_{(K-1)(2K+1)}(M)}{1\wedge K\|G(\mathbf{f}^*)\|_\infty}\left(\|\bar{p}\|_2^2 + \|\bar{q}\|_F^2 + \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_F^2\right) \\ &+ o\left(\frac{1}{1\wedge\sigma_K(G(\mathbf{f}^*))}\left(\|\bar{p}\|_2^2 + \|\bar{q}\|_F^2 + \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_F^2\right)\right)\end{aligned}$$

and finally

$$\begin{aligned}\|g^{\pi^*+\bar{p},\mathbf{Q}^*+\bar{q},\mathbf{f}^*+\bar{A}\mathbf{f}^*} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}^*}\|_2^2 & \\ \geq& c_2(\pi^*,\mathbf{Q}^*,\mathbf{f}^*)\left(\|\bar{p}\|_2^2 + \|\bar{q}\|_F^2 + \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_F^2\right) \\ &+ o\left(\|\bar{p}\|_2^2 + \|\bar{q}\|_F^2 + \sum_{k\in\mathcal{X}} \|(\bar{A}\mathbf{f}^*)_k\|_F^2\right) \quad (4)\end{aligned}$$

where

$$c_2(\pi^*,\mathbf{Q}^*,\mathbf{f}^*) = \frac{H}{(1\wedge K\|G(\mathbf{f}^*)\|_\infty)(3K^3(1\vee\|G(\mathbf{f}^*)\|_\infty^4))^{K^2-K/2}}$$

is positive as soon as **[Hid]** and **[Hdet]** hold.

B.4 Proof of Theorem 10

Let

$$N_{\mathbf{f}}(p,q,\mathbf{h}) = \|g^{\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{h}} - g^{\pi^*,\mathbf{Q}^*,\mathbf{f}}\|_2^2$$

and

$$\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2 = d_{\text{perm}}((\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{h}),(\pi^*,\mathbf{Q}^*,\mathbf{f}))^2.$$

We want to show that there exists a constant $c^* > 0$ such that there exists a neighborhood $\mathcal{V}$ of $\mathbf{f}^*$ such that if one writes

$$c_{\mathbf{f}} := \inf_{p\in(\Delta-\Delta),\, q\in(\mathcal{Q}-\mathcal{Q}),\, \mathbf{h}\in(\mathcal{F}-\mathcal{F})^K} \frac{N_{\mathbf{f}}(p,q,\mathbf{h})}{\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2}$$

then $\inf_{\mathbf{f}\in\mathcal{V}} c_{\mathbf{f}} \geq c^*$.

The proof follows the structure of the proof of Theorem 6 of De Castro et al. (2016). It consists of three steps: the first one controls the component of $\mathbf{h}$ that is orthogonal to $\mathbf{f}$. This makes it possible to restrict $\mathbf{h}$ to the finite-dimensional space spanned by $\mathbf{f}$ in the two other parts. The second step controls the case when $\mathbf{h}$ is small, so that the behaviour of $N_{\mathbf{f}}$ is given by its quadratic form, and the last step controls the case where $\mathbf{h}$ is far from zero.

B.4.1 The Orthogonal Part

Let $\mathbf{u}$ be the orthogonal projection of $\mathbf{h}$ on Span($\mathbf{f}$). Then

$$N_{\mathbf{f}}(p,q,\mathbf{h}) = N_{\mathbf{f}}(p,q,\mathbf{u}) + M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{h}-\mathbf{u})$$

where

$$\begin{aligned}
M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a}) = \sum_{i_1,j_1,k_1}\sum_{i_2,j_2,k_2} &(\pi^*+p)(i_1)(\mathbf{Q}^*+q)(i_1,j_1)(\mathbf{Q}^*+q)(j_1,k_1)\\
&(\pi^*+p)(i_2)(\mathbf{Q}^*+q)(i_2,j_2)(\mathbf{Q}^*+q)(j_2,k_2)\\
&\Big(\langle a_{i_1},a_{i_2}\rangle\langle (f+u)_{j_1},(f+u)_{j_2}\rangle\langle (f+u)_{k_1},(f+u)_{k_2}\rangle\\
&+\langle (f+u)_{i_1},(f+u)_{i_2}\rangle\langle a_{j_1},a_{j_2}\rangle\langle (f+u)_{k_1},(f+u)_{k_2}\rangle\\
&+\langle (f+u)_{i_1},(f+u)_{i_2}\rangle\langle (f+u)_{j_1},(f+u)_{j_2}\rangle\langle a_{k_1},a_{k_2}\rangle\\
&\quad+\langle a_{i_1},a_{i_2}\rangle\langle a_{j_1},a_{j_2}\rangle\langle (f+u)_{k_1},(f+u)_{k_2}\rangle\\
&\quad+\langle a_{i_1},a_{i_2}\rangle\langle (f+u)_{j_1},(f+u)_{j_2}\rangle\langle a_{k_1},a_{k_2}\rangle\\
&\quad+\langle (f+u)_{i_1},(f+u)_{i_2}\rangle\langle a_{j_1},a_{j_2}\rangle\langle a_{k_1},a_{k_2}\rangle\\
&\qquad+\langle a_{i_1},a_{i_2}\rangle\langle a_{j_1},a_{j_2}\rangle\langle a_{k_1},a_{k_2}\rangle\Big).
\end{aligned}$$

Let us write Π' the matrix whose diagonal terms are the elements of π^*+p and $\mathbf{Q}'$ the matrix $\mathbf{Q}^*+q$, then $M_{\mathbf{f}}$ can be written as

$$\begin{aligned}
M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a}) = \sum_{i,j}\Big(&((\Pi'\mathbf{Q}')^\top G(\mathbf{a})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{f}+\mathbf{u})_{i,j}(\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{a})_{i,j}(\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{f}+\mathbf{u})_{i,j}(\mathbf{Q}'^\top G(\mathbf{a})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{a})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{a})_{i,j}(\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{a})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{f}+\mathbf{u})_{i,j}(\mathbf{Q}'^\top G(\mathbf{a})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{a})_{i,j}(\mathbf{Q}'^\top G(\mathbf{a})\mathbf{Q}')_{i,j}\\
&+((\Pi'\mathbf{Q}')^\top G(\mathbf{a})\Pi'\mathbf{Q}')_{i,j}G(\mathbf{a})_{i,j}(\mathbf{Q}'^\top G(\mathbf{a})\mathbf{Q}')_{i,j}\Big).
\end{aligned}$$

By the Schur product theorem, these terms are nonnegative since they correspond to Hadamard products of three Gram matrices which are nonnegative. Thus, one can lower bound $M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a})$ by the second term of the sum, which leads to

$$M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a}) \geq \sum_{i,j=1}^{K}((\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}')_{i,j}(\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}')_{i,j}\langle a_i,a_j\rangle$$

Assume [**Hid**] holds for the parameters $(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u})$, then the matrices $(\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}'$ and $\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}'$ are positive symmetric with respective lowest eigenvalue lower

bounded by $(\inf_k(\pi_k^*+p_k)\sigma_K(\mathbf{Q}^*+q))^2\sigma_K(G(\mathbf{f}+\mathbf{u}))$ and $\sigma_K(\mathbf{Q}^*+q)^2\sigma_K(G(\mathbf{f}+\mathbf{u}))$. Therefore, their Hadamard product is positive, and one has

$$(((\Pi'\mathbf{Q}')^\top G(\mathbf{f}+\mathbf{u})\Pi'\mathbf{Q}')_{i,j}(\mathbf{Q}'^\top G(\mathbf{f}+\mathbf{u})\mathbf{Q}')_{i,j})_{i,j} = (D\mathbf{U})^\top(D\mathbf{U})$$

with $\mathbf{U}$ an orthogonal matrix and D a diagonal matrix with positive diagonal coefficients. Moreover, the Schur product theorem implies that $\sigma_K(D)^2 \geq (\inf_k(\pi_k^*+p_k))^2\sigma_K(\mathbf{Q}^*+q)^4\sigma_K(G(\mathbf{f}+\mathbf{u}))^2$. Then

$$\begin{aligned}
M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a}) &\geq \sum_{i,j=1}^{K}((D\mathbf{U})^\top(D\mathbf{U}))_{i,j}\langle a_i,a_j\rangle\\
&= \sum_{j=1}^{K}\|D\mathbf{U}\mathbf{a}\|_2^2\\
&\geq \sigma_K(D)^2\|\mathbf{U}\mathbf{a}\|_2^2\\
&\geq (\inf_k(\pi_k^*+p_k))^2\sigma_K(\mathbf{Q}^*+q)^4\sigma_K(G(\mathbf{f}+\mathbf{u}))^2\|\mathbf{a}\|_2^2.
\end{aligned}$$

Finally, let $c_1(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u}) = (\inf_k(\pi_k^*+p_k))^2\sigma_K(\mathbf{Q}^*+q)^4\sigma_K(G(\mathbf{f}+\mathbf{u}))^2$. The application $(p,\pi^*,q,\mathbf{Q}^*,\mathbf{u},\mathbf{f}) \mapsto c_1(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u})$ is continuous and nonnegative, it is positive when [**Hid**] holds for the parameters $(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u})$, and one has

$$M_{\mathbf{f}}(p,q,\mathbf{u},\mathbf{a}) \geq c_1(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u})\|\mathbf{a}\|_2^2.$$

We will now control the term $N_{\mathbf{f}}(p,q,\mathbf{u})$. Two cases appear: when $(\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u})$ is close to $(\pi^*,\mathbf{Q}^*,\mathbf{f}^*)$ in some sense and when it is not. The first case will be solved using the nondegeneracy of the quadratic form ensured by [**Hdet**]. The second case will be solved using the identifiability of the HMM.

B.4.2 In the Neighborhood of $\mathbf{f}^*$.

The Taylor expansion of

$$(p,q,\mathbf{u}) \in (\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times((\mathcal{F}-\mathcal{F})\cap\text{Span}(\mathbf{f}))^K \mapsto N_{\mathbf{f}}(p,q,\mathbf{u})$$

around $(0,0,0)$ leads to a nonnegative quadratic form and no linear part. [**Hdet**], [**Hid**] and equation (4) ensure that this form is positive for $\mathbf{f}=\mathbf{f}^*$. Let $c_2(\mathbf{Q}^*,\pi^*,\mathbf{f})$ be as defined in Section B.3.2, then $\mathbf{f} \mapsto c_2(\mathbf{Q}^*,\pi^*,\mathbf{f})$ is continuous and it is positive in the neighborhood of $\mathbf{f}^*$. Moreover, there exists a positive constant η depending on $\|G(\mathbf{f})\|_\infty$ such that for all $(p,q,\mathbf{u})$ such that $\|(p,q,\mathbf{u})\|_{\mathbf{f}} \leq 1$, one has

$$N_{\mathbf{f}}(p,q,\mathbf{u}) \geq c_2(\mathbf{Q}^*,\pi^*,\mathbf{f})\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2 - \eta\|(p,q,\mathbf{u})\|_{\mathbf{f}}^3.$$

For instance, $\eta = 4000K^6\|G(\mathbf{f})\|_\infty^3$ works: the terms of order 2 or more in the Taylor expansion of $N_{\mathbf{f}}$ are the scalar product of sums of terms of the form $\sum_{i,j,k\in\mathcal{X}}\pi^*(i)\mathbf{Q}^*(i,j)\mathbf{Q}^*(j,k)f_i\otimes f_j\otimes f_k$ where zero to three of the f may be replaced by u, zero to two of the $\mathbf{Q}^*$ by q and π^* may be replaced by p and at least one of them is replaced. There are 63 possibilities, which leads to a sum of $(63K^3)^2$ terms, each of which can be bounded by

$\|G(\mathbf{f})\|_\infty^3(\max\{p(i),q(i,j),\|u_i\|_2 \mid i,j\in\mathcal{X}\})^r$ where r is the number of replaced terms. By taking the right permutation of states, the max can be bounded by $\|(p,q,\mathbf{u})\|_{\mathbf{f}}$, hence the result.

Then, using $\|(p,q,\mathbf{h})\|_{\tilde{\mathbf{f}}}^2 = \|(p,q,\mathbf{u})\|_{\mathbf{f}}^2 + \|\mathbf{a}\|_2^2$ leads to

$$\begin{aligned}\frac{N_{\mathbf{f}}(p,q,\mathbf{h})}{\|(p,q,\mathbf{h})\|_{\tilde{\mathbf{f}}}^2} \geq& c_1(\mathbf{Q}^*+q,\pi^*+p,\mathbf{f}+\mathbf{u})\frac{\|\mathbf{a}\|_2^2}{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2+\|\mathbf{a}\|_2^2}\\ &+c_2(\mathbf{Q}^*,\pi^*,\mathbf{f})\frac{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2}{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2+\|\mathbf{a}\|_2^2}-\eta\frac{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2}{(\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2+\|\mathbf{a}\|_2^2)^{1/2}}\\ \geq& c_1(\mathbf{Q}^*+q,\pi^*+p,\mathbf{f}+\mathbf{u})\frac{\|\mathbf{a}\|_2^2}{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2+\|\mathbf{a}\|_2^2}\\ &+c_2(\mathbf{Q}^*,\pi^*,\mathbf{f})\frac{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2}{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2+\|\mathbf{a}\|_2^2}-\eta\sqrt{\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2}\end{aligned}$$

Let $c_0 = \min(c_1/2,c_2)/2$, then c_0 is continuous and there exists a continuous function $(\pi^*,\mathbf{Q}^*,\mathbf{f})\mapsto\epsilon(\pi^*,\mathbf{Q}^*,\mathbf{f})$ which is positive as soon as **[Hid]** and **[Hdet]** hold for $(\pi^*,\mathbf{Q}^*,\mathbf{f})$ and such that

$$\|(p,q,\mathbf{u})\|_{\mathbf{f}}\leq\epsilon(\pi^*,\mathbf{Q}^*,\mathbf{f}) \;\Rightarrow\; \frac{N_{\mathbf{f}}(p,q,\mathbf{h})}{\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2}\geq c_0(\mathbf{Q}^*,\pi^*,\mathbf{f}).$$

Thus, there exists positive constants ϵ_0 and c_{near} depending on $\mathbf{Q}^*$, π^* and $\mathbf{f}^*$ such that

$$\begin{aligned}&\forall(p,q,\mathbf{h},\mathbf{f})\in(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times(\mathcal{F}-\mathcal{F})^K\times\mathcal{F}^K\\ &\qquad\text{s.t. }\|(p,q,\mathbf{u})\|_{\mathbf{f}}\leq\epsilon_0 \text{ and } \sum_{k\in\mathcal{X}}\|f_k-f_k^*\|_2^2\leq\epsilon_0^2, \qquad \frac{N_{\mathbf{f}}(p,q,\mathbf{h})}{\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2}\geq c_{\text{near}}.\end{aligned}$$

B.4.3 Far from $\mathbf{f}^*$.

Lemma 18 *The application*

$$(p,q,\mathbf{u},\mathbf{f})\in(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times(\mathcal{F}-\mathcal{F})^K\times\mathcal{F}^K\longmapsto N_{\mathbf{f}}(p,q,\mathbf{u})$$

restricted to the set of $(p,q,\mathbf{u},\mathbf{f})$ such that $\mathbf{u}\in Span(\mathbf{f})^K$ is uniformly continuous for the norm $\|\cdot\|_{tot}$ defined by

$$\|(p,q,\mathbf{u},\mathbf{f})\|_{tot}^2 := \|p\|_2^2+\|q\|_F^2+\sum_{k\in\mathcal{X}}\left(\|u_k\|_2^2+\|f_k\|_2^2\right).$$

Thus, by compactness of $(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times((\mathcal{F}-\mathcal{F})\cap\text{Span}(\mathbf{f}))^K$, the application

$$c_{\text{far}}:\mathbf{f}\longmapsto\inf_{(p,q,\mathbf{u})\in(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times((\mathcal{F}-\mathcal{F})\cap\text{Span}(\mathbf{f}))^K\text{ s.t. }\|(p,q,\mathbf{u})\|_{\mathbf{f}}>\epsilon_0}N_{\mathbf{f}}(p,q,\mathbf{u})$$

is continuous. Let us now prove that $c_{\text{far}}(\mathbf{f}^*)>0$.

Let $(p_n,q_n,\mathbf{u}_n)_n\in((\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times((\mathcal{F}-\mathcal{F})\cap\text{Span}(\mathbf{f}^*))^K)^{\mathbb{N}}$ be a sequence such that $\|(p_n,q_n,\mathbf{u}_n)\|_{\mathbf{f}^*}>\epsilon_0$ for all n and

$$c_{\text{far}}(\mathbf{f}^*)=\lim_n N_{\mathbf{f}^*}(p_n,q_n,\mathbf{u}_n).$$

By compactness, this sequences converges towards a limit $(p,q,\mathbf{u})$ up to taking a subsequence. Necessarily $\|(p,q,\mathbf{u})\|_{\mathbf{f}^*}\geq\epsilon_0$. Since **[Hid]** holds, Theorem 8 shows that $N_{\mathbf{f}^*}(p,q,\mathbf{u})>0$, which implies $c_{\text{far}}(\mathbf{f}^*)>0$ by continuity of $N_{\mathbf{f}^*}$. Note that $c_{\text{far}}(\mathbf{f}^*)$ may depend on $\mathcal{F}$ in addition to the parameters π^*, $\mathbf{Q}^*$ and $\mathbf{f}^*$.

Thus, by continuity, there exists $\epsilon_1>0$ such that for all $\mathbf{f}\in\mathcal{F}^K$ such that $\sum_{k\in\mathcal{X}}\|f_k-f_k^*\|_2^2\leq\epsilon_1^2$, $c_{\text{far}}(\mathbf{f})\geq c_{\text{far}}(\mathbf{f}^*)/2$.

Finally, **[HF]** implies that there exists a constant $\mathcal{C}$ depending only on $C_{\mathcal{F},2}$ such that $\|(p,q,\mathbf{u})\|_{\mathbf{f}}^2\leq\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2\leq\mathcal{C}$ for all $(p,q,\mathbf{h})\in(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times(\mathcal{F}-\mathcal{F})^K$. Therefore,

$$\begin{aligned}&\forall(p,q,\mathbf{h},\mathbf{f})\in(\Delta-\Delta)\times(\mathcal{Q}-\mathcal{Q})\times(\mathcal{F}-\mathcal{F})^K\times\mathcal{F}^K\\ &\text{s.t. }\|(p,q,\mathbf{u})\|_{\mathbf{f}}\geq\epsilon_0 \text{ and } \sum_{k\in\mathcal{X}}\|f_k-f_k^*\|_2^2\leq\epsilon_1^2, \qquad \frac{N_{\mathbf{f}}(p,q,\mathbf{h})}{\|(p,q,\mathbf{h})\|_{\mathbf{f}}^2}\geq\frac{N_{\mathbf{f}}(p,q,\mathbf{u})}{\mathcal{C}}\\ &\qquad\qquad\geq\frac{c_{\text{far}}(\mathbf{f}^*)}{2\mathcal{C}}.\end{aligned}$$

The theorem follows by taking $c^*(\pi^*,\mathbf{Q}^*,\mathbf{f}^*,\mathcal{F})=\min\left(\frac{c_{\text{far}}(\mathbf{f}^*)}{2\mathcal{C}},c_{\text{near}}\right)$ and the neighborhood containing all $\mathbf{f}\in\mathcal{F}^K$ such that $\sum_{k\in\mathcal{X}}\|f_k-f_k^*\|_2^2\leq\min(\epsilon_0,\epsilon_1)^2$. Moreover, $(\pi,\mathbf{Q},\mathbf{f})\longmapsto c^*(\pi,\mathbf{Q},\mathbf{f},\mathcal{F})$ is lower bounded by this value in a neighborhood of $(\pi^*,\mathbf{Q}^*,\mathbf{f}^*)$, so that it can be assumed to be lower semicontinuous.

Note that the dependency of c^* on $\mathcal{F}$ appears during this last step and is made non explicit because of the compactness assumption.

B.4.4 Proof of Lemma 18

$$\begin{aligned}&\left|N_{\mathbf{f}}(p,q,\mathbf{u})-N_{\mathbf{f}'}(p',q',\mathbf{u}')\right|\\ &\quad=\left|\|g^{\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u}}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}}\|_2^2-\|g^{\pi^*+p',\mathbf{Q}^*+q',\mathbf{f}'+\mathbf{u}'}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}'}\|_2^2\right|\\ &\quad\leq 2\|g^{\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u}}-g^{\pi^*+p',\mathbf{Q}^*+q',\mathbf{f}'+\mathbf{u}'}\|_2^2+2\|g^{\pi^*,\mathbf{Q}^*,\mathbf{f}}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}'}\|_2^2\\ &\qquad+2\left|\left\langle g^{\pi^*+p',\mathbf{Q}^*+q',\mathbf{f}'+\mathbf{u}'}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}'},g^{\pi^*+p,\mathbf{Q}^*+q,\mathbf{f}+\mathbf{u}}-g^{\pi^*+p',\mathbf{Q}^*+q',\mathbf{f}'+\mathbf{u}'}\right\rangle\right|\\ &\qquad+2\left|\left\langle g^{\pi^*+p',\mathbf{Q}^*+q',\mathbf{f}'+\mathbf{u}'}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}'},g^{\pi^*,\mathbf{Q}^*,\mathbf{f}}-g^{\pi^*,\mathbf{Q}^*,\mathbf{f}'}\right\rangle\right|\end{aligned}$$

Then, using the fact that $\|g^{\pi,\mathbf{Q},\mathbf{f}}-g^{\pi',\mathbf{Q}',\mathbf{f}'}\|_2\leq\sqrt{3K}C_{\mathcal{F},2}^3\|(\pi-\pi',\mathbf{Q}-\mathbf{Q}',\mathbf{f}-\mathbf{f}',0)\|_{\text{tot}}$ (see Lemma 17), that $\|g^{\pi,\mathbf{Q},\mathbf{f}}\|_2\leq C_{\mathcal{F},2}^3$ (see for instance Lemma 29 of Lehéricy (to appear)) and the Cauchy-Schwarz inequality,

$$\begin{aligned}\left|N_{\mathbf{f}}(p,q,\mathbf{u})-N_{\mathbf{f}'}(p',q',\mathbf{u}')\right|\leq&6KC_{\mathcal{F},2}^6\|(p-p',q-q',\mathbf{f}+\mathbf{u}-\mathbf{f}'-\mathbf{u}',0)\|_{\text{tot}}^2\\ &+6KC_{\mathcal{F},2}^6\|(0,0,0,\mathbf{f}-\mathbf{f}')\|_{\text{tot}}^2\\ &+4\sqrt{3K}C_{\mathcal{F},2}^6\|(p-p',q-q',\mathbf{f}+\mathbf{u}-\mathbf{f}'-\mathbf{u}',0)\|_{\text{tot}}\\ &+4\sqrt{3K}C_{\mathcal{F},2}^6\|(0,0,0,\mathbf{f}-\mathbf{f}')\|_{\text{tot}}^2\\ \leq&24KC_{\mathcal{F},2}^6\Big(\|(p-p',q-q',\mathbf{u}-\mathbf{u}',\mathbf{f}-\mathbf{f}')\|_{\text{tot}}^2\\ &+\|(p-p',q-q',\mathbf{u}-\mathbf{u}',\mathbf{f}-\mathbf{f}')\|_{\text{tot}}\Big),\end{aligned}$$

which proves the uniform continuity of the application.

References

Animashree Anandkumar, Daniel J Hsu, and Sham M Kakade. A method of moments for mixture models and hidden Markov models. In *COLT*, volume 1, page 4, 2012.

Sylvain Arlot, Alain Celisse, et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, 4:40–79, 2010.

Jean-Patrick Baudry, Cathy Maugis, and Bertrand Michel. Slope heuristics: overview and implementation. *Statistics and Computing*, 22(2):455–470, 2012.

Sophie Bertrand, Rocío Joo, and Ronan Fablet. Generalized Pareto for pattern-oriented random walk modelling of organisms' movements. *PloS one*, 10(7):e0132231, 2015.

Lucien Birgé and Pascal Massart. Minimal penalties for Gaussian model selection. *Probability theory and related fields*, 138(1-2):33–73, 2007.

Stéphane Bonhomme, Koen Jochmans, and Jean-Marc Robin. Non-parametric estimation of finite mixtures from repeated measurements. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 78(1):211–229, 2016a.

Stéphane Bonhomme, Koen Jochmans, and Jean-Marc Robin. Estimating multivariate latent-structure models. *The Annals of Statistics*, 44(2):540–563, 2016b.

Charlotte Boyd, André E Punt, Henri Weimerskirch, and Sophie Bertrand. Movement models provide insights into variation in the foraging effort of central place foragers. *Ecological modelling*, 286:13–25, 2014.

Yohann De Castro, Élisabeth Gassiat, and Claire Lacour. Minimax adaptive estimation of nonparametric hidden Markov models. *Journal of Machine Learning Research*, 17(111): 1–43, 2016.

Yohann De Castro, Elisabeth Gassiat, and Sylvain Le Corff. Consistent estimation of the filtering and marginal smoothing distributions in nonparametric hidden Markov models. *IEEE Transactions on Information Theory*, 2017.

Ronald A DeVore and George G Lorentz. *Constructive approximation*, volume 303. Springer Science & Business Media, 1993.

Elisabeth Gassiat, Alice Cleynen, and Stéphane Robin. Finite state space non parametric hidden Markov models are in general identifiable. *Stat. Comp.*, pages 1–11, 2015.

Elisabeth Gassiat, Judith Rousseau, and Elodie Vernet. Efficient semiparametric estimation and model selection for multidimensional mixtures. *Electronic Journal of Statistics*, 12 (1):703–740, 2018.

Alexander Goldenshluger and Oleg Lepski. Bandwidth selection in kernel density estimation: oracle inequalities and adaptive minimax optimality. *The Annals of Statistics*, pages 1608–1632, 2011.

Alexander Goldenshluger and Oleg Lepski. General selection rule from a family of linear estimators. *Theory of Probability & Its Applications*, 57(2):209–226, 2013.

Alexander Goldenshluger and Oleg Lepski. On adaptive minimax density estimation on $\mathbb{R}^d$. *Probability Theory and Related Fields*, 159(3-4):479–543, 2014.

Daniel Hsu, Sham M Kakade, and Tong Zhang. A spectral algorithm for learning hidden Markov models. *Journal of Computer and System Sciences*, 78(5):1460–1480, 2012.

Claire Lacour, Pascal Massart, and Vincent Rivoirard. Estimator selection: a new method with applications to kernel density estimation. *Sankhya A*, 79(2):298–335, 2017.

Luc Lehéricy. Consistent order estimation for nonparametric hidden Markov models. *Bernouilli*, to appear.

Youen Vermard, Etienne Rivot, Stéphanie Mahévas, Paul Marchal, and Didier Gascuel. Identifying fishing trip behaviour and estimating fishing effort from VMS data using Bayesian hidden Markov models. *Ecological Modelling*, 221(15):1757–1769, 2010.

Journal of Machine Learning Research 19 (2018) 1-95 Submitted 10/17; Revised 7/18; Published 8/18

Learning from Comparisons and Choices

Sahand Negahban SAHAND.NEGAHBAN@YALE.EDU
Statistics Department
Yale University

Sewoong Oh SWOH@ILLINOIS.EDU
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Kiran K. Thekumparampil THEKUMP2@ILLINOIS.EDU
Department of Electrical and Computer Engineering
University of Illinois at Urbana-Champaign

Jiaming Xu XU972@PURDUE.EDU
Krannert School of Management
Purdue University

Editor: Qiang Liu

Abstract

When tracking user-specific online activities, each user's preference is revealed in the form of choices and comparisons. For example, a user's purchase history is a record of her choices, i.e. which item was chosen among a subset of offerings. A user's preferences can be observed either explicitly as in movie ratings or implicitly as in viewing times of news articles. Given such individualized ordinal data in the form of comparisons and choices, we address the problem of collaboratively learning representations of the users and the items. The learned features can be used to predict a user's preference of an unseen item to be used in recommendation systems. This also allows one to compute similarities among users and items to be used for categorization and search. Motivated by the empirical successes of the MultiNomial Logit (MNL) model in marketing and transportation, and also more recent successes in word embedding and crowdsourced image embedding, we pose this problem as learning the MNL model parameters that best explain the data. We propose a convex relaxation for learning the MNL model, and show that it is minimax optimal up to a logarithmic factor by comparing its performance to a fundamental lower bound. This characterizes the minimax sample complexity of the problem, and proves that the proposed estimator cannot be improved upon other than by a logarithmic factor. Further, the analysis identifies how the accuracy depends on the topology of sampling via the spectrum of the sampling graph. This provides a guideline for designing surveys when one can choose which items are to be compared. This is accompanied by numerical simulations on synthetic and real data sets, confirming our theoretical predictions.

Keywords: Collaborative Ranking, Nuclear Norm Minimization, Multi-Nomial Logit Model

1. Introduction

Given data on how users compared subsets of items, we address the fundamental problem of learning a representation of users and items. Such data can be observed in the form of choices (e.g. which item was bought) or in the form of comparisons (e.g. which items are rated higher). From such ordinal data on the items, we want to find low dimensional representations, which we call (latent) features, that explain crucial aspects of the users' choices. Once learned, these features can be used to predict each user's preference over items that the user has not seen yet, which can be used in recommendation systems and revenue management. These learned features also provide an embedding of the users and items on the same Euclidean space that allows us to directly quantify similarities via distances, that can be used to categorize and cluster. These embeddings can reveal the underlying structure of data such as images. Such an embedding of a discrete set of objects based on ordinal data has recently gained tremendous attraction mainly due to word embeddings based on co-occurrence data and their successes in numerous downstream natural language processing tasks Mikolov et al. (2013b).

The fundamental question in such a representation learning is: what makes one representation better than the others? Our guiding principle is that a good representation is the one that defines a generative model that best explains the given data in the maximum likelihood sense. To this end, we focus on a parametric generative model known as Multi-Nomial Logit (MNL) model, widely used and studied in revenue management. The MNL model has a natural interpretation of human choices as an outcome of maximizing a utility by agents with noisy perception of the utility, also known as *random utility model* in Walker and Ben-Akiva (2002); Azari Soufiani et al. (2012), defined as follows. Each user and item has a latent low-dimensional feature $u_i \in \mathbb{R}^r$ and $v_j \in \mathbb{R}^r$ respectively. The true utility of an item is the inner product of these two features $\Theta_{ij} \triangleq \langle\!\langle u_i, v_j \rangle\!\rangle = \sum_k u_{ik} v_{jk}$. The inherent low-rank structure of $\Theta = [\Theta_{ij}]$ captures the collaborative nature of the problem, where users with similar preferences in the past are likely to prefer similar items in the future.

When presented with a set of items, a user reveals a noisy ordering of the items sorted according to her perceived utilities of the items, each of which is perturbed by an i.i.d. noise added to the true utility Θ_{ij}. The MNL model is a special case where the noise follows the standard Gumbel distribution, and is one of the most popular models in choice theory for its simplicity and empirical success McFadden (1973); McFadden and Train (2000). The MNL model has several important properties, making this model realistic in various domains, including marketing Guadagni and Little (1983), transportation McFadden (1980); Ben-Akiva and Lerman (1985), biology Sham and Curtis (1995), sports games Tsokos et al. (2018) and natural language processing Mikolov et al. (2013a). The MNL model (*i*) satisfies the 'independence of irrelevant alternatives' in social choice theory Ray (1973); (*ii*) has a maximum likelihood estimator (MLE) which is a convex program in Θ; and (*iii*) has a simple characterization of sequential (random) choices as follows. Let $\mathbb{P}\{a > \{b, c, d\}\}$ denote the probability a was chosen as the best alternative among the set $\{a, b, c, d\}$. Then, the probability that user i reveals a linear order $(a > b > c > d)$ is $\mathbb{P}\{a > \{b, c, d\}\}\mathbb{P}\{b > \{c, d\}\}\mathbb{P}\{c > d\}$, where $\mathbb{P}\{a > \{b, c, d\}\} = e^{\Theta_{ia}}/(e^{\Theta_{ia}} + e^{\Theta_{ib}} + e^{\Theta_{ic}} + e^{\Theta_{id}})$. Essentially the user is modeled as making a sequence of choices, choosing the best alternative first and then making choices on the remaining ones. We give the precise definition of the MNL model in Section 2 for pairwise comparisons and in Section 4 for higher order comparisons and choices. Beyond its success in classical applications such as transportation and marketing, the MNL model and its variants are being rediscovered and successfully applied in more modern applications

such as embedding images using crowdsourcing Tamuz et al. (2011) and word embedding Mikolov et al. (2013b), whose connections we make precise in Section 6.

Motivated by recent advances in learning low-rank models, e.g. Negahban et al. (2009); Davenport et al. (2014), we ask the fundamental question of learning the MNL model from data on comparisons and choices. We provide a general framework using convex relaxations for learning the model. As data is collected in various forms on modern social computing systems, we consider the following four canonical scenarios:

- *Pairwise comparisons.* The most simple and canonical piece of ordinal data one can collect from a user at a time is a pairwise comparison; given two options, we ask the user which one is better. Such data is prevalent in the real world and is the most popular scenario studied in ranking literature, e.g. Shah et al. (2014). However, one significant aspect of the real data that has not been addressed in the literature is irregularities in the sampling. Consider an online seller with various products, say cars and watches. It does not make sense to ask a user to compare a car and a watch; one cannot sample an outcome of a comparison between a watch and a car. However, knowing a user's preference on cars can help in learning her preference on watches. We want to propose a model and design an inference algorithm that can take into account such restrictions in sampling. We further want to quantify the gain in using all such data together in inference, as opposed to running inference in each category separately. To this end, we propose a new model for sampling that we call *graph sampling*. This model explains such irregularities in the real world data. We propose a novel inference algorithm tailored for the given sampling pattern. Our analysis captures precisely how the accuracy depends on the different topologies of the sampling.

- *Higher order comparisons.* Consider an online market that collects each of its user's preference as a ranking over a subset of items that is 'seen' by the user. Such data can be obtained by directly asking to compare some items, or by indirectly tracking online activities on which items are viewed, how much time is spent on the page or how the user rated the items. However, collecting such comparisons over multiple items might come at a cost. We, therefore, want to quantify the gain in the accuracy of the inference when higher order comparison outcomes are collected. We characterize the optimal trade-off between accuracy and the number of items compared, and show that our proposed algorithm seamlessly generalizes to this setting and also achieves the optimal trade-off.

- *Customer choices.* One of the most widely applicable data collection scenarios is customer purchase history. Online and offline service providers can track each customer on which subset of items is offered and which item is chosen. Given historical data on such choices on best-out-of-a-subset, we extract features on the users and items that best explains the collected data.

- *Bundled choices.* Another data collection scenario that is gaining interest recently is bundled choices Chu et al. (2011); Benson et al. (2018). Typical choice models assume that the willingness to buy an item is independent of what else the user bought. In many cases, however, we make 'bundled' purchases: we buy particular ingredients together for one recipe or we buy two connecting flights. One choice (the first flight) has a significant impact on the other (the connecting flight). In order to optimize the assortment (which flight schedules to offer) for maximum expected revenue, it is crucial to accurately predict the willingness of the consumers to purchase bundled items, based on past history. We propose a model that can capture such interacting preferences for bundled items (e.g. jeans and shirts), and use this model to extract the features of the items in each category from historical bundled purchase data. Both our inference algorithm and the analyses extend to this setting, achieving the optimal trade-off between sample size and accuracy.

Contribution. We first study the canonical scenario of pairwise comparisons from the MNL model in Section 3. Our contribution in the modeling is a new sampling scenario we call *graph sampling* that captures how different pairs of items have varying likelihood of being compared together. Our algorithmic contribution is a convex relaxation with a new regularizer using a variation of the standard nuclear norm tailored for the graph sampling topology. Our theoretical contribution is in the analysis of the proposed estimator and a matching fundamental lower bound (up to a poly-logarithmic factor). This (a) characterizes the minimax sample complexity of the problem; (b) proves that the proposed estimator cannot be improved upon; and (c) identifies how the accuracy depends on the topology of sampling. This in turn provides a guideline for designing surveys when one has a choice on which pairs are to be compared. This is accompanied by experiments on synthetic and real data sets confirming our theoretical predictions.

This framework is extended to higher order comparisons in Section 4. We establish minimax optimality (up to a poly-logarithmic factor) of our estimator and identify the fundamental trade-off between accuracy and sample size. When each user provides a total linear ordering among k items, we show that the required sample size effectively is reduced by a factor of k. When the user provides her best choice (as in purchase history) instead of the total linear ordering, we extend our framework and establish minimax optimality in Section 5.2. We also consider a bundled purchase scenario in Section 5, where customers buy pairs of items from each of the two categories. We extend our framework and establish minimax optimality under the bundled purchase setting. We present experimental results on both synthetic and real-world data sets confirming our theoretical predictions and showing the improvement of the proposed approach in predicting users' choices[1].

Technically, we borrow analysis tools from 1-bit matrix completion Davenport et al. (2014), matrix completion Negahban and Wainwright (2012), and restricted strong convexity Negahban et al. (2009), and crucially utilize the Random Utility Model (RUM) Thurstone (1927); Marschak (1960); Luce (1959) interpretation (outlined in Section 2.1) of the MNL model to prove both the upper bound and the fundamental limit. This could be of interest to analyzing more general class of RUMs.

Notations. We use $|||A|||_{\rm F}$ and $|||A|||_{\infty}$to denote the Frobenius norm and the ℓ_∞ norm, $|||A|||_{\rm nuc} = \sum_i \sigma_i(A)$ to denote the nuclear norm where $\sigma_i(A)$ denotes the i-th singular value, and $|||A|||_2 = \sigma_1(A)$ for the spectral norm. We use $\langle\!\langle u, v \rangle\!\rangle = \sum_i u_i v_i$ and $\|u\|$ to

1. Code for our experiments are available at https://github.com/POLane16/Nucnorm-Ranking.

denote the inner product and the Euclidean norm. All ones vector is denoted by $\mathbb{1}$, $\mathbf{I}$ denotes the identity matrix and $\mathbb{I}(A)$ is the indicator function of the event A. The set of the first N integers are denoted by $[N] = \{1, \ldots, N\}$.

1.1 Related Work

Bradley-Terry and Plackett-Luce models. The simplest form of the MNL model is when all users are sharing the same feature vector such that each item is parametrized by a scalar value. This is known as Bradley-Terry (BT) model when pairwise comparisons are concerned and Plackett-Luce (PL) model when higher order comparisons are concerned. This has been proposed and rediscovered several times in the last century Zermelo (1929); Thurstone (1927); Bradley and Terry (1955); Luce (1959); Plackett (1975); McFadden (1973, 1980) in the context of ranking teams in sports games, ranking items based on surveys, and ranking routes in transportation systems. Unlike the general MNL model, maximum likelihood estimator for the BT and PL models are naturally convex programs. However, learning the BT model has first been addressed in Jr. (1957) where the convergence of the iterative algorithm is analyzed, without explicitly relying on the convexity of the problem. A new algorithm based on Majorize-Minimize framework was proposed in Hunter (2004). First sample complexity of learning BT model was provided in Negahban et al. (2012) where a novel estimator, called Rank Centrality, of the BT parameters was proposed. The authors construct a random walk over a graph where the nodes are the items and the transition probability is constructed from the comparisons outcomes. This spectral approach is proven to achieve a minimax optimal sample complexity. This has been a building block for several ranking algorithms, which further process the Rank Centrality to get better accuracy on top of it Chen and Suh (2015); Jang et al. (2016, 2017); Chen et al. (2017). For higher order comparisons, the sample complexity of learning PL model was provided in Hajek et al. (2014); Shah et al. (2014), where the Maximum Likelihood (ML) estimator is shown to achieve the minimax optimality. Later, Maystre and Grossglauser (2015) made the connection between the spectral approach of Rank Centrality and the ML estimator precise by providing a unifying random walk view to the problem. This led to a novel Accelerated Spectral Ranking algorithm introduced in Agarwal et al. (2018), which not only finds the parameters of the PL model more efficiently in computation, but also achieves optimal sample complexity under general sampling graphs. Recently, Borkar et al. (2016) treat the learning problem as solving a noisy linear system, and propose an algorithm that is amenable to on-line, distributed and asynchronous variants. Vojnovic and Yun (2016) analyzes a more general class of random utility models known as Thurstone models, and provide the minimax sample complexity by analyzing the ML estimator. Note that the ML estimators for Thurstone models in general are computationally intractable.

Generalized BT and PL models. As studied in Rajkumar and Agarwal (2014), the BT model covers a subset of probabilistic models over comparisons. There is a hierarchy of models with increasing complexity and descriptive power. One popular extension is the mixture of BT or PL models. It is known that any choice model can be approximated arbitrarily close with a mixed PL model with sufficient number of mixture components McFadden and Train (2000). The sample complexity of learning a mixed PL model was analyzed in Oh and Shah (2014) where a tensor decomposition for learning a mixture model

was proposed and analyzed under some separation conditions between the weights of the mixtures. For a mixture of two PL model, Chierichetti et al. (2018) shows identifiability and uniqueness of the mixture weights, when all marginal probability over all possible rankings among two items and three items are known. In a crowdsourced setting, Chen et al. (2013) models pairwise comparisons using a mixture of PL models consisting of hammer distribution, which reports the true output of a comparison, and spammer distribution, which reports the exact opposite of a comparison. A different approach that tackles the problem by learning to cluster the users based on the pairwise comparisons is proposed in Rui et al. (2015). The MNL model we study in this paper can be thought of as a generalization of the mixed PL models, where each user has her own preference. To make learning feasible, we inherently impose similarities among users via a low-rank condition. Note that a mixed PL model with r mixture is a special case of the MNL model with rank r, where each user's membership is encoded as a r-dimensional feature in standard basis. In the context of collaborative ranking, algorithms for learning the MNL model from pairwise comparisons have been proposed in Park et al. (2015). Instead of nuclear norm regularization as we propose in this paper, Park et al. (2015) proposes solving a convex relaxation of maximizing the likelihood over matrices with bounded nuclear norm. Under the standard assumption of uniformly chosen pairs, it is shown that this approach achieves statistically optimal generalization error rate, instead of Frobenius norm error that we analyze.

Beyond BT and PL models. Modeling choice is an important problem where the ultimate goal is to find the right parametric model to capture human choices. Ragain and Ugander (2016); Blanchet et al. (2013) use Markov chains to model choices with the parameters in the transition matrix defining the probability model. Ideal point model Massimino and Davenport (2018); Kazemi et al. (2018) assumes that the pairwise comparisons of two items by a user depends on their distance from an ideal item (ideal point) for the user in some metric embedding space of the items. Novel nonparametric models have also been proposed to model human choices, for example Shah et al. (2016b); Pananjady et al. (2017); Falahatgar et al. (2018) uses strong stochastic transitivity to model pairwise choices and Farias et al. (2009) uses distribution over all permutations with sparse support to model higher order choices. We also note that in the context of (non-collaborative) ranking, Gleich and Lim (2011) has proposed nuclear norm minimization based algorithm when comparisons between all pairs items are modeled as a low-rank skew-symmetric matrix. Other nonparametric approaches to solving ranking include empirical risk minimization. Clémençon et al. (2005) analyses risk minimization of U-statistics and a more feasible surrogate convex loss minimization to estimate ranking. Katz-Samuels and Scott (2017) assumes that rating of an item by a user is a Lipschitz function of the user-item pair and analyses a nonparametric collaborative ranking algorithm from partial observation of such ratings.

While we are interested in the (parameters of) full ranking over all items, there have been several recent works which aim to only approximately rank the items, such as retrieving only the top-m items Chen and Suh (2015); Jang et al. (2016, 2017) or partitioning the items into ordered buckets of fixed size Katariya et al. (2018); Heckel et al. (2018).

2. Model and Approach for Pairwise Comparisons

The MultiNomial Logit (MNL) model is one of the most popular models that explains how people make choices when given multiple options and is widely used in behavioral psychology and revenue management. For brevity, we focus our discussion on data collected in the form of pairwise comparisons in Sections 2 and 3, and defer the discussion of the MNL model in its full generality to Sections 4 and 5 . We give a precise definition of the model for paired comparisons and provide a novel algorithmic solution to learn this model from samples.

2.1 MultiNomial Logit (MNL) Model for Pairwise Comparisons

Let Θ^* be a $d_1 \times d_2$ dimensional matrix capturing preferences of d_1 users on d_2 items. The probability with which a user, $i \subseteq [d_1]$, when presented with two items $j_1, j_2 \subseteq [d_2]$, prefers item j_1 over item j_2 is,

$$\mathbb{P}\{j_1 > j_2\} = \frac{e^{\Theta^*_{ij_1}}}{e^{\Theta^*_{ij_1}} + e^{\Theta^*_{ij_2}}}\,. \tag{1}$$

This implies that, more preferred items (as per the ordering of Θ^*_{ij}) are more likely to be ranked higher, with the randomness in choices captured by the probabilistic model.

If we do not impose any further constraints on Θ^*, one entry of Θ^* is not related in any way to any other entries. This implies that one user's preference is completely independent of others' and no efficient learning is possible. Each user's preference has to be learned separately. On the other hand, in real applications, it is reasonable to say that preferences of users depend only on a handful of factors for example, quality, price, and aesthetics. We do not know which features affect users' choices, but we assume that there are r-dimensional latent features for each of the users and items that govern such choices, and that $r \ll d_1, d_2$. This assumption mathematically captures the conventional belief that when two people have similar preferences over a subset of items, they tend to have similar tastes on other items as well. Formally, MNL model assumes that Θ^* is a rank r matrix with $r \ll d_1, d_2$. In this paper, we do not impose a hard constraint on the rank and provide general results for matrices of any rank. In this case, we identify how the accuracy depends on the rate of decay of the singular values.

This MNL model has many roots. In revenue management, this has been proposed as a special case of Random Utility Model (RUM). RUM explains choices that a person makes as the result of maximizing perceived random utilities associated with the set of alternatives presented. In the case of MNL, each decision maker and each alternative are associated with an r-dimensional vector, u_i and v_j, resulting in a low-rank Θ^* if $\Theta^*_{ij} = \langle\!\langle u_i, v_j \rangle\!\rangle$. The perceived utility of the item j for decision maker i is,

$$U_{ij} = \langle\!\langle u_i, v_j \rangle\!\rangle + \xi_{ij}\,, \tag{2}$$

where ξ_{ij}'s are i.i.d. random variables following the standard Gumbel distribution. Different choices of distributions give different variants of RUMs. In our analyses, we utilize this RUM interpretation of the MNL model to prove a particular concentration in Section C.4, for example. The model in Equation. (1) has also been re-discoverd several times in the literature Zermelo (1929); Thurstone (1927); Luce (1959); Bradley and Terry (1955) in several domains.

2.2 Low-rank Regularization using Nuclear Norm Minimization

Given the low-rank structure of the model, a natural but inefficient approach is to minimize the negative of the log likelihood, $\mathcal{L}(\cdot)$, regularized by the rank:

$$\widehat{\Theta} \;\in\; \arg\min_{\Theta \in \mathbb{R}^{d_1 \times d_2}} -\mathcal{L}(\Theta) \,+\, \lambda\, \mathrm{rank}(\Theta), \tag{3}$$

for some parameter $\lambda > 0$. As this rank minimization is a notoriously challenging problem, we instead solve a convex relaxation of it. Note that the nuclear norm ball is the convex hull of rank-1 matrices Recht et al. (2010). Analogous to l_1-norm in the case of sparse vectors, nuclear norm is a tight convex surrogate for low-rank solutions. We propose the following nuclear norm regularized optimization problem,

$$\widehat{\Theta} \;\in\; \arg\min_{\Theta \in \Omega} -\mathcal{L}(\Theta) \,+\, \lambda\, |||\Theta|||_{\mathrm{nuc}}, \tag{4}$$

where Ω is a convex constraint which takes care of identifiability and Lipschitz smoothness conditions. Nuclear norm regularization has been widely used Recht et al. (2010) for rank minimization; however, provable guarantees exist only for quadratic loss functions $\mathcal{L}(\Theta)$ Candès and Recht (2009); Negahban and Wainwright (2012). Our analyses extend such results to a convex loss, by first proving that $-\mathcal{L}(\cdot)$ satisfies restricted strong convexity property with high probability. Similar to how (non-collaborative) rank aggregation has been generalized to any strongly log-concave distribution in Shah et al. (2014), our analysis can naturally be extended to a general class of strongly log-concave distributions. We give the expression for the log likelihood in Equation. (8) for pairwise comparisons.

3. Learning MNL Model from Pairwise Comparisons under Graph Sampling

Probabilistic model for sampling. In order to provide performance guarantees on the proposed approach, we need to specify how we sample the pairs that are to be compared. We provide a novel sampling model, which we call *graph sampling* with respect to a weighted graph $\mathcal{G}$. This naturally generalizes Bernoulli sampling typically studied under matrix completion literature Candès and Recht (2009); Keshavan et al. (2010a); Negahban and Wainwright (2012); Jain et al. (2013), and the resulting analysis captures how the performance depends on the topology of the samples. Note that the proposed graph sampling is different from deterministic sampling graphs studied in Hajek et al. (2014); Shah et al. (2016a). This is analytically tractable only in the simpler case of estimating the weight vector of the PL model where there is only one user and the ML estimator is a convex program. However, such deterministic sampling is notoriously hard to handle for matrix estimation, even in the simpler case of matrix completion Bhojanapalli and Jain (2014). Hence, we introduce a probabilistic model that allows enough flexibility to capture the interesting aspects of sampling biases, i.e. grouping.

Precisely, we have a weighted undirected graph $\mathcal{G} = ([d_2], E, \{P_{j_1,j_2}\}_{(j_1,j_2)\in E})$ with d_2 nodes, which represent items, a set of edges E and the edge weight P_{j_1,j_2} between nodes j_1 and j_2. The weights can be written in a symmetric matrix $P \in \mathbb{R}^{d_2 \times d_2}$, and $P_{j_1,j_2} + P_{j_2,j_1} = 2P_{j_1,j_2}$ represent the probability with which the pair (j_1, j_2) is chosen for comparison. Note

that $P_{j,j} = 0$,$\forall j \in [d_2]$, $P_{j_1,j_2} = P_{j_2,j_1}$ and $\sum_{j_1,j_2\in[d_2]} P_{j_1,j_2} = 1$. We assume we get i.i.d. samples from first choosing a random user among $[d_1]$ users, and then choosing a pair (j_1, j_2) of items at random from P, and finally getting a random comparison from the MNL model, i.e. the probability with which user i prefers item j_1 over item j_2 is $\exp\Theta^*_{ij_1}/\left(\exp\Theta^*_{ij_1} + \exp\Theta^*_{ij_2}\right)$.

One of the most important aspects of real-world data that is captured by this graph sampling model is grouping. Consider two groups of items, say, cars and phones. It does not make sense to ask an individual to compare a phone with a brand of a car (i.e. direct comparison is not feasible), but knowing an individual's preference on cars can help in learning her preference on phones. In graph sampling terms, we are sampling from a graph $\mathcal{G}$ consisting of two disjoint cliques: one for cars and another for phones. By analyzing such a sampling scenario, we want to characterize the gain in using the data from both groups of items together, although there are no inter-group comparisons.

In the preference matrix Θ^*, the values in the set of columns corresponding to each connected component in the sampling graph can be arbitrarily shifted together, without changing the pairwise comparisons outcome distributions. This is because adding the same constant to those items that are compared does not change the probability (for those items within the same group), i.e.

$$\mathbb{P}\{j_1 > j_2\} \;=\; \frac{e^{\Theta^*_{ij_1}}}{e^{\Theta^*_{ij_1}} + e^{\Theta^*_{ij_2}}} \;=\; \frac{e^{\Theta^*_{ij_1}+c}}{e^{\Theta^*_{ij_1}+c} + e^{\Theta^*_{ij_2}+c}} \;,$$

and adding different constants to those items that are not in the same group does not change the probability of the outcome as those items are never compared. Hence, to handle this unidentifiability, we let a centered version of Θ^* represent all those shifted versions defining the same probability distribution. Formally, let a zero-one vector $g_k \in \{0,1\}^{d_2}$ denote the group membership such that $g_{i,k} = 1$ if item j is in group k, else $g_{i,k} = 0$. Note that, by definition, no item can be present in more than one group, that is, $\sum_{k=1}^{G} g_k = \mathbb{1}$, where G is the number of groups. We define an equivalence class of Θ^* which represent the same probabilistic model as

$$[\Theta^*] \;=\; \left\{\Theta^* + \sum_{k=1}^{G} u_k g_k^T \;\text{ for all } u_k \in \mathbb{R}^{d_1}\right\}. \tag{5}$$

To overcome the identifiability issue, we represent each equivalence class with the centered matrix satisfying

$$\Theta^* g_k = 0, \quad \forall\, k \in \{1, 2, \ldots, G\} \tag{6}$$

As matrices with large "spikiness" are known to be hard to estimate Negahban and Wainwright (2012), we capture the dependence of the sample complexity on the spikiness as measured by $\alpha := |||\Theta^*|||_\infty$. This captures the dynamic range of the underlying preference matrix. For a related problem of matrix completion, where the loss $\mathcal{L}(\theta)$ is quadratic, either a similar condition on ℓ_∞ norm is required or another condition on incoherence is required.

Graph Laplacian. The performance of our approach depends on the sampling graph P via its graph Laplacian defined as

$$L \;=\; \mathrm{diag}(P\mathbb{1}) - P \tag{7}$$

where $\mathrm{diag}(P\mathbb{1})$ is a diagonal matrix with $\sum_v P_{u,v}$ in the diagonals. Notice that, L is singular and the nullspace is spanned by vectors $\{g_k\}_{k=1}^G$. Let $\sigma_{\max}(L) = \|L\|_2$ and $\sigma_{\min}(L)$ be the smallest eigenvalue of L discounting the G zero-valued eigenvalues. Since the graph has G disconnected maximal components and L is real symmetric, by spectral theorem, $L = U\Sigma U^T$, where U is a matrix of size $d_2 \times (d_2 - G)$ and its $d_2 - G$ columns form an orthonormal set, and Σ is a diagonal matrix such that its diagonal elements are the singular values of L. Let $L^\dagger := U\Sigma^{-1}U^T$ and $L^x := U\Sigma^x U^T$ for all $x \in \mathbb{R}$. We also define the Laplacian induced norms of matrices as,

$$|||\Theta|||_{\mathrm{L}} := \left|\left|\left|\Theta L^{1/2}\right|\right|\right|_{\mathrm{F}}, \text{ and, } |||\Theta|||_{\text{L-nuc}} := \left|\left|\left|\Theta L^{1/2}\right|\right|\right|_{\mathrm{nuc}}.$$

These Laplacian induced norms are more appropriate to analyze and quantify the distance between the estimated matrix $\widehat{\Theta}$ and Θ^*.

When items $k(i)$, $l(i)$ are chosen for comparison by user $j(i)$ as the i-th pair of items, we capture this choice with the matrix $X^{(i)} = e_{j(i)}(e_{k(i)} - e_{l(i)})^T$. The outcome of the comparison is represented by y_i, with $y_i = 1$ when item $k(i)$ wins over item $l(i)$ and $y_i = 0$ if otherwise. The log-likelihood of the comparison outcomes with respect to a parameter matrix Θ is,

$$\mathcal{L}(\Theta) \;=\; \frac{1}{n}\sum_{i=1}^{n}\left[y_i\langle\!\langle\Theta, X^{(i)}\rangle\!\rangle - \log\left(1 + \exp\left(\langle\!\langle\Theta, X^{(i)}\rangle\!\rangle\right)\right)\right]. \tag{8}$$

We propose and analyze the following convex optimization problem,

$$\widehat{\Theta} \;\in\; \underset{\Theta\in\Omega_\alpha}{\mathrm{argmin}} - \mathcal{L}(\Theta) + \lambda|||\Theta|||_{\text{L-nuc}}, \tag{9}$$

where,

$$\Omega_\alpha \;=\; \left\{\Theta \in \mathbb{R}^{d_1\times d_2} \mid |||\Theta|||_\infty \le \alpha, \Theta g_k = 0,\ \forall\, k \in [G]\right\}, \tag{10}$$

with an appropriately chosen $\lambda = 8\sqrt{2}\max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$ with $\sigma = \max\{(d_2 - G)/d_1, 1\}$, where $d = (d_1 + d_2)/2$. In practice, the sampling probability distribution P and the corresponding Laplacian L might not be known. In those cases, we propose using the empirical sampling probability distribution $\hat{P}$ and corresponding empirical Laplacian $\hat{L}$ instead. We describe this version of the algorithm formally in Section 3.4.4, where we empirically demonstrate the robustness of this approach. Further, in experiments with real data sets, we use the empirical Laplacian Section 3.4.5.

3.1 Performance Guarantee

We consider the graph sampling scenario where each sample is i.i.d., the ℓ-th sample consists of user i_ℓ chosen uniformly at random, pair of items $(j_{1,\ell}, j_{2,\ell})$ chosen according to the sampling graph $\mathcal{G} = ([d_2], E, P)$, and the resulting outcome y_ℓ distributed as the MNL model with parameter Θ^*.

Theorem 1 *Under the graph sampling with respect to $\mathcal{G} = ([d_2], E, P)$ with a graph Laplacian L, and under the MNL preference model with preference matrix Θ^*, solving the optimization problem in (9) with n i.i.d. samples achieves, with probability greater than $1 - 1/4d^3$,*

$$\frac{1}{d_1}\left|\left|\left|\left(\Theta^* - \widehat{\Theta}\right) L^{1/2}\right|\right|\right|_{\mathrm{F}}^2 \leq 36\lambda\left(\alpha + \frac{1}{\psi(2\alpha)}\right)\left(\sqrt{2r}\left|\left|\left|\left(\Theta^* - \widehat{\Theta}\right) L^{1/2}\right|\right|\right|_{\mathrm{F}} + \sum_{j=r+1}^{\min\{d_1, d_2 - G\}} \sigma_j(\Theta^* L^{1/2})\right), \tag{11}$$

for any $r \in \{1, 2, \ldots, \min\{d_1, d_2 - G\}\}$, any $\lambda \geq 8\sqrt{2}\max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$ where $\sigma = \max\{(d_2 - G)/d_1, 1\}$ and $d = (d_1 + d_2)/2$, $\psi(x) \triangleq e^x/(1 + e^x)^2$, and for $n \leq \min\{2^6 d_1^2\sigma^2, 2^2\left(d_1\sigma_{\min}(L)^{-1}\right)^{2/3}\}\log(2d)$.

We provide a proof in Appendix A. The above bound holds for any r, where r allows us to trade off the two types of errors: the estimation error and the approximation error. Concretely, the above bound shows a natural splitting of the error into two terms; the first term corresponding to the *estimation error* for the top rank-r component of Θ^* and the second term corresponding to the *approximation error* for how well one can approximate Θ^* with a rank-r matrix. If we know the singular values of Θ^*, we can optimize over r to get the tightest bound. If Θ^* is exactly low-rank then applying a matching rank in the bound gives the following guarantee.

Corollary 2 (Exact rank-r matrix) *Under the same hypothesis as in Theorem 1 with a choice of $\lambda = c_0\max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$ for some $c_0 > 0$, if Θ^* is exactly rank r, there exists a positive constant c_1 such that the proposed estimator achieves,*

$$\frac{1}{\sqrt{d_1}}\left|\left|\left|\left(\Theta^* - \widehat{\Theta}\right) L^{1/2}\right|\right|\right|_{\mathrm{F}} \leq c_1\left(\alpha + \frac{1}{\psi(2\alpha)}\right)\sqrt{r}\max\left\{\sqrt{\frac{\sigma\, d_1\log(2d)}{n}}, \frac{\sqrt{(\sigma_{\min}(L)^{-1}d_1)}\,\log(2d)}{n}\right\}, \tag{12}$$

with probability at least $1 - 2/(d_1 + d_2)^3$ and $\sigma = \max\{(d_2 - G)/d_1, 1\}$.

The second term in the maximization is an artifact of the weakness of current analysis technique and does not reflect the actual error. This is confirmed in our simulation results on graphs with very small spectral gap in Figures 1.(b), (d), and (f), where the error in Laplacian-induced norm error does not decrease with spectral gap of L as the line graph has a much smaller spectral gap compared to a complete graph, for example. In fact, for a special Θ^* in Figure 1.(d) it is the other way, for which we do not have a theoretical explanation.

The number of entries in Θ^* is $d_1 d_2$ and we want to rescale the Frobenius norm error appropriately by $1/\sqrt{d_1 d_2}$. As a typical scaling of $L^{1/2}$ is $1/\sqrt{d_2}$ in spectral norm, we

only need to rescale the Laplacian-induced norm error by $1/\sqrt{d_1}$ in the left-hand side of the above bound. For a rank-r Θ^*, the number of degrees of freedom in describing it is $r(d_1 + d_2) - r^2 = O(r(d_1 + d_2))$. The above theorem shows that the total number of samples n needs to scale as $O(r(d_1 + d_2)\log d)$ in order to achieve an arbitrarily small error. This is only a poly-logarithmic factor larger than the degrees of freedom. In Section 3.2 we make this comparison precise by providing a lower bound that matches the upper bound up to a logarithmic factor.

The upper-bound constraint in Theorem 1 on the number of samples n can be met for large enough d_1 and d_2. For simplicity, assume that $d_1 = d_2 = d$ and r is a constant. Since $\sigma_{\min}(L) = O(1/d_2)$, the upper-bound on n becomes $O(\max\{d^2, d^{4/3}\})$. For large enough d, the upper bound on the RHS of Eq. (12) can be made arbitrarily small with n only scaling as $O(r\,d)$. This is significantly smaller than the upper-bound of $O(d^2)$ on n. Further, in the experiments in Section 3.4, we show that n has no practical upper-bound constraint since the error decreases at the same rate as predicted, for arbitrarily large values of n. This constraint may not be necessary and might be a by-product of the proof techniques.

The dependence on the dynamic range α, however, is sub-optimal. It is expected that the error increases with α, since the Θ^* scales as α, but the exponential dependence in the bound seems to be a weakness of the analysis (for example as seen from numerical experiments in the right panel of Figure 6). Although the error increase with α, numerical experiments suggest that it only increases at most linearly. However, tightening the scaling with respect to α is a challenging problem, and such sub-optimal dependence is also present in existing literature for learning even simpler models, such as the Bradley-Terry model Negahban et al. (2012) or the Plackett-Luce model Hajek et al. (2014), which are special cases of the MNL model studied in this paper.

Another issue is that the underlying matrix might not be exactly low rank. It is more realistic to assume that it is approximately low rank. Following Negahban and Wainwright (2012) we formalize this notion with "ℓ_q-ball" of matrices defined as

$$\mathbb{B}_q(\rho_q) \equiv \{\Theta \in \mathbb{R}^{d_1 \times d_2} \mid \sum_{j \in [\min\{d_1, d_2\}]} |\sigma_j(\Theta^*)|^q \leq \rho_q\}\,. \tag{13}$$

When $q = 0$, this is a set of rank-ρ_0 matrices. For $q \in (0, 1]$, this is set of matrices whose singular values decay relatively fast. By optimizing the choice of r in Theorem 1, we get the following result.

Corollary 3 (Approximately low-rank matrices) *Suppose $\Theta^* \in \mathbb{B}_q(\rho_q)$ for some $q \in (0, 1]$ and $\rho_q > 0$. Under the hypotheses of Theorem 1, with a choice of $\lambda = c_0\max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$ for some constant $c_0 > 0$ there exists a constant $c_1 > 0$ such that solving the optimization (9) achieves with probability at least $1 - 2/(d_1 + d_2)^3$,*

$$\frac{1}{\sqrt{d_1}}\left|\left|\left|\left(\Theta^* - \widehat{\Theta}\right) L^{1/2}\right|\right|\right|_{\mathrm{F}} \leq \frac{c_1\sqrt{\rho_q}}{\sqrt{d_1}}\left(\left(\alpha + \frac{1}{\psi(2\alpha)}\right)\sqrt{\frac{d_1^2\,\sigma\,\log(2d))}{n}}\right)^{\frac{2-q}{2}}, \tag{14}$$

provided $n \geq \sigma\,\log(2d)/\sigma_{\min}(L)$.

This is a strict generalization of Corollary 2. For $q = 0$ and $\rho_0 = r$, this recovers the exact low-rank estimation bound up to a factor of two. For approximate low-rank matrices in an ℓ_q-ball, we lose in the error exponent, which reduces from one to $(2-q)/2$.

3.2 Information-theoretic Lower Bound

For a polynomial-time algorithm of convex relaxation, we gave in the previous section a bound on the achievable error. We next compare this to the fundamental limit of this problem, by giving a lower bound on the achievable error by any algorithm (efficient or not). A simple parameter counting argument indicates that it requires the number of samples to scale as the number of degrees of freedom i.e., $n \propto r(d_1+d_2)$, to estimate a $d_1 \times d_2$ dimensional matrix of rank r. We construct an appropriate packing over the set of low-rank matrices with bounded entries in Ω_α defined as (10), and show that no algorithm can accurately estimate the true matrix with high probability using the generalized Fano's inequality. This provides a constructive argument to lower bound the minimax error rate, which in turn establishes that the bounds in Theorem 1 is sharp up to a logarithmic factor, and proves no other algorithm can significantly improve over the nuclear norm minimization.

Theorem 4 *Suppose Θ^* has a rank r. Under the previously described graph based sampling model, there exists a constant $c > 0$ such that*

$$\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\frac{1}{\sqrt{d_1}}\Big|\Big|\Big|\Big(\Theta^* - \widehat{\Theta}\Big) L^{1/2}\Big|\Big|\Big|_{\mathrm{F}}\Big] \geq c \min\left\{e^{-\alpha}\sqrt{\frac{r\,d_1}{n}}\,, \alpha \max\left\{\sqrt{\frac{r}{\mathrm{tr}\left(L_r^\dagger\right)}}\,, \frac{d_2}{\sqrt{d_1 \log d}}\right\}\right\}, \quad (15)$$

where the infimum is taken over all measurable functions over the observed comparison results and $L_r^\dagger$ is the pseudo inverse of the rank r approximation of the graph Laplacian.

A proof of this theorem is provided in Appendix B. The term of primary interest in this bound is the first one, which shows the scaling of the (rescaled) minimax rate as $\sqrt{rd_1/n}$ and matches the upper bound in (12) up to a logarithmic factor. It is the dominant term in the bound whenever the number of samples is larger than $n \geq d_1 \max\{\mathrm{tr}\left(L_r^\dagger\right), d_1 \log d/d_2^2\}$. As suggested in numerical simulations on graphs with very small spectral gap in Figures 1.(b), (d), and (f), the dependence in $\mathrm{tr}\left(L_r^\dagger\right)$ is an artifact of the weakness of the current analysis technique. Here we note that, while the lower bound in Theorem 4 is in expectation, the upper bound in Theorem 1 is a high-probability result. The upper bound can immediately be translated into a bound in expected error with an additional term scaling as $\alpha\sigma_{\max}(L)^{1/2}\sqrt{d_2}d^{-3}$, which is smaller than other terms in the bound.

3.3 Performance Guarantee and Lower Bound for Complete Graph

It follows from a simple relation $|||(\Theta^* - \widehat{\Theta})L^{1/2}|||_F \geq \sigma_{min}^{1/2}|||\Theta^* - \widehat{\Theta}|||_F$, which is true since Θ^*, $\widehat{\Theta}$ are in the range space of L, that the above upper bounds automatically give the error bound in the Frobenius norm. When the sampling graph is uniform, i.e. a complete graph

with equal weights $P_{j_1,j_2} = 1/d_2(d_2-1)$, $\forall j_1 \neq j_2$, Frobenius norm is the right metric and we show matching upper and lower bounds.

Corollary 5 (Complete graph upper-bound) *Under the same hypothesis as in Corollary 2, if $\mathcal{G}$ is a complete graph, with a choice of $\lambda = c_0 \max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sqrt{(d_2-1)}\log(2d)}{n}\right\}$ for some $c_0 > 0$, if Θ^* is exactly rank r, there exists a positive constant c_1 such that the proposed estimator achieves,*

$$\frac{\left|\left|\left|\Theta^* - \widehat{\Theta}\right|\right|\right|_{\mathrm{F}}}{\sqrt{d_1(d_2-1)}} \leq c_1\left(\alpha + \frac{1}{\psi(2\alpha)}\right)\sqrt{r}\max\left\{\sqrt{\frac{\sigma\ d_1\log(2d)}{n}}, \frac{\sqrt{(d_2-1)d_1}\ \log(2d)}{n}\right\}, \quad (16)$$

with probability at least $1 - 2/(d_1+d_2)^3$ and $\sigma = \max\{(d_2-1)/d_1, 1\}$.

Corollary 6 (Complete graph lower-bound) *Suppose Θ^* has rank r. Under the previously described graph based sampling model with graph being a complete graph, there is a universal numerical constant $c > 0$ such that*

$$\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\frac{1}{\sqrt{d_1\,(d_2-1)}}\Big|\Big|\Big|\widehat{\Theta} - \Theta^*\Big|\Big|\Big|_{\mathrm{F}}\Big] \geq c \min\left\{e^{-\alpha}\sqrt{\frac{r\,d_1}{n}}\,, \alpha\max\left\{\frac{1}{\sqrt{(d_2-1)}}\,, \frac{d_2}{\sqrt{d_1\log d}}\right\}\right\}, \quad (17)$$

where the infimum is taken over all measurable functions over the observed comparison results.

3.4 Experiments

We provide a first-order method to solve the proposed convex optimization, and provide numerical experiments using this algorithm. We present two simulation results followed by an experiment on real data.

For the synthetic experiments, we generate random rank-r matrices of dimension $d \times d$, of the form $\Theta^* = UV^T$ with $U \in \mathbb{R}^{d\times r}$ and $V \in \mathbb{R}^{d\times r}$ entries generated i.i.d from uniform distribution over $[0,1]$. Then the connected-component-mean is subtracted form each connected component, and then the whole matrix is scaled such that the largest entry is $\alpha = 5$. Note that this operation does not increase the rank of the matrix Θ. This is because this de-meaning can be written as $\Theta - \sum_k \Theta g g^T/(g^T\mathbb{1})$ and both terms in the operation are of the same column space as Θ which is of rank r.

3.4.1 Algorithm

Let $\Theta' \triangleq \Theta L^{1/2}$. As the nuclear norm regularizer in (9) is non differentiable, we use the proximal gradient descent Agarwal et al. (2010); Cai et al. (2010). At each iteration, we apply the following two operations on the current estimate, Θ'_t, of $\Theta^* L^{1/2}$,

$$\widetilde{\Theta'}_{t+1} = \Theta'_t - \eta_t \nabla_\Theta \mathcal{L}(\Theta'_t L^{-1/2})L^{-1/2} \quad \text{(gradient descent)} \quad (18)$$

$$\Theta'_{t+1} = M_t(\Gamma_t - \eta_t\lambda\mathbf{I})^+ N_t^T \quad \text{(singular value shrinkage and thresholding)} \quad (19)$$

where $M_t \Gamma_t N_t^T := \widetilde{\Theta'}_t$ is the singular value decomposition of $\widetilde{\Theta'}_t$, such that Γ_t is a diagonal matrix with positive entries, $(\cdot)^+$ is the entry-wise thresholding operation $\max(0, x)$, and η_t is an appropriate step-size. Constraint of zero row sum, is taken care of by initializing the descent algorithm with $\Theta'_0 = 0$, since rows of gradients sum to zero. In practice we do not know the value of α, and hence in experiments we do not enforce the $\|\Theta\|_\infty \leq \alpha$ constraint.

Another issue in the implementation is that the convergence rate can be significantly slower for some graph topologies. We accelerate the proximal gradient descent with the following (modified) Barzilai-Borwein (BB) rule Barzilai and Borwein (1988) for choosing the step-size η_t,

$$\eta_t = \begin{cases} \frac{\|\Theta'_t - \Theta'_{t-1}\|_2^2}{\left\langle\!\left\langle \Theta'_t - \Theta'_{t-1}, \nabla_{\Theta'}\mathcal{L}'(\Theta'_t) - \nabla_{\Theta'}\mathcal{L}'(\Theta'_{t-1}) \right\rangle\!\right\rangle}, & \text{when } t \text{ is odd} \\ \frac{\left\langle\!\left\langle \Theta'_t - \Theta'_{t-1}, \nabla_{\Theta'}\mathcal{L}'(\Theta'_t) - \nabla_{\Theta'}\mathcal{L}'(\Theta'_{t-1}) \right\rangle\!\right\rangle}{\|\nabla_{\Theta'}\mathcal{L}'(\Theta'_t) - \nabla_{\Theta'}\mathcal{L}'(\Theta'_{t-1})\|_2^2}, & \text{when } t \text{ is even} \end{cases}, \tag{20}$$

where $\nabla_{\Theta'}\mathcal{L}'(\Theta') := \nabla_\Theta \mathcal{L}(\Theta' L^{-1/2}) L^{-1/2}$. We stop the descent algorithm whenever an upper bound of the KKT error is smaller than 10^{-5}.

3.4.2 The Role of the Topology of the Sampling Pattern

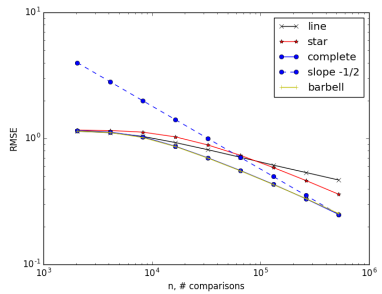
In figure 1, we plot the error of our nuclear norm minimization based algorithm versus number of samples (in log-scale), n for $d_1 = d_2 = 300$, $r = 4$, $\alpha = 5.0$, $G = 1$. We consider two errors here; root mean squared error (RMSE) $= \left|\!\left|\!\left| \Theta - \widehat{\Theta} \right|\!\right|\!\right|_{\mathrm{F}} / \sqrt{d_1 d_2}$ and Laplacian induced RMSE (L-RMSE)$= \left|\!\left|\!\left| (\Theta - \widehat{\Theta}) L^{1/2} \right|\!\right|\!\right|_{\mathrm{F}} / \sqrt{d_1}$. We plot these errors for four topologies of varying spectral gaps. As discussed Section 3.1, we do not expect the L-RMSE error to change much as we change the topology of sampling. However, as seen from the simple relation $|||(\Theta^* - \widehat{\Theta}) L^{1/2}|||_F \geq \sigma_{min}^{1/2} |||\Theta^* - \widehat{\Theta}|||_F$ Frobenius norm error is more sensitive to the topology of the sampling pattern, captured via the spectral gap, i.e. $\sigma_{\min}(L)$. Specifically we use the following graph topologies.

- **Complete graph.** We first consider a uniform sampling over a complete graph where $P_{j_1,j_2} = 1/d_2(d_2-1)$ for all $j_1, j_2 \in [d_2]$. The resulting spectral gap is $1/(d_2-1)$, which is the maximum possible value. Hence, complete graphs are optimal for learning MNL models, compared in the error metric of the Frobenius norm for fairness.

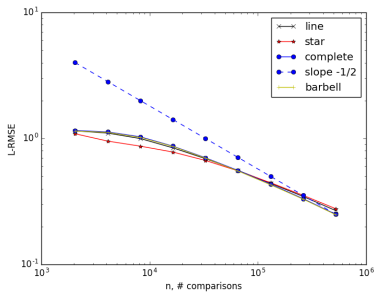
- **Star graph.** Here we choose one item to be the center, and every other items can only be compared to this center item uniformly at random. Let item 1 be the center one, then $P_{j_1,1} = P_{1,j_2} = 1/2(d_2 - 1)$. Standard spectral analysis shows that the spectral gap is $\Theta(1/d_2)$, and thus the graph is near-optimal for learning MNL models.

- **Line graph.** Next, we consider a line graph with $d_2 - 1$ edges where $P_{j,j+1} = P_{j+1,j} = 1/2(d_2 - 1)$. It has a spectral gap of $\Theta(1/d_2^2)$, and is strictly sub-optimal for learning MNL models.

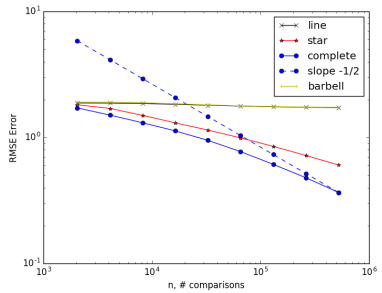
- **Barbell graph.** Consider two equal sized groups of items. Within each group the sub-graph is complete, and between the groups there is a single edge connecting one of the node from group one and one of the node from group two. Each edge is chosen



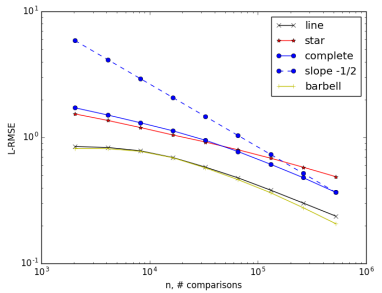
(a) RMSE for i.i.d. Θ^*_{ij}



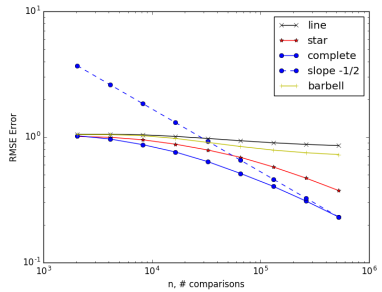
(b) L-RMSE for i.i.d. Θ^*_{ij}



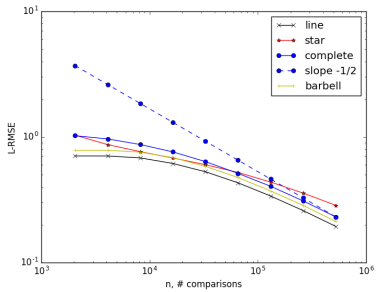
(c) RMSE for barbell bias Θ^*_{ij}



(d) L-RMSE for barbell bias Θ^*_{ij}



(e) RMSE for line bias Θ^*_{ij}



(f) L-RMSE for line bias Θ^*_{ij}

Figure 1: Graphs with small spectral gap achieve significantly larger Frobenius norm error (RMSE) $\left|\!\left|\!\left| \Theta - \widehat{\Theta} \right|\!\right|\!\right|_{\mathrm{F}} / \sqrt{d_1 d_2}$, whereas the Laplacian-induced norm error (L-RMSE) $\left|\!\left|\!\left| (\Theta - \widehat{\Theta}) L^{1/2} \right|\!\right|\!\right|_{\mathrm{F}} / \sqrt{d_1}$ is not sensitive to the spectral gap.

uniformly at random for comparisons. The resulting spectral gap is $\Theta(1/d_2^2)$, and this graph too is strictly sub-optimal for learning MNL models.

First in sub-figures 1a, 1b, we plot RMSE and L-RMSE errors for different graphs using randomly generated Θ^*_{ij}. We see that L-RMSE curves for different graphs are the same (and slopes in log-scale are as expected approaches $-1/2$ with more samples). Further, we do not see any significant difference w.r.t the graph topology even when error is measured in Frobenius norm. The reason is that since Θ^*_{ij}'s are generated i.i.d., the empirical distributions of any large sub-group of items would be similar. Thus, the means of the two cliques of the barbell graph or the means of the items on the two far ends of the line graph are similar. Thus although barbell and line graphs have small spectral gap (high mixing-time), its effect is minimized because these sub-groups can individually be solved without having them to mix since the empirical distributions of the Θ^*_{ij} in the two sub-groups are similar.

To illustrate the role of the topology of the graph, we choose specific Θ^* which depends on the topology of the graph as guided by our analysis on the lower bound (Theorem 4) in sub-figures 1c, 1d. The items are divided into two sets (corresponding to each side of barbell graph), such that corresponding Θ^*_{ij} are i.i.d. inside a set but have similar but shifted means across the sets. We call this type of preference data as *barbell biased*. As expected from theoretical analyses, L-RMSE behave similar to the i.i.d. case. However, we see the Frobenius norm error significantly worse in the case of line and barbell shaped graphs, as expected from the Frobenius error bound. In sub-figures 1e, 1f, we simulate *line biased* preference data Θ^*. Items are ordered (in the order of the line graph), such that Θ^*_{ij}'s have similar distributions but their means get shifted in an arithmetic progression as you go down the ordering. Again, Frobenius norm error is significantly larger for line and barbell graphs as spectral gaps are small.

3.4.3 The Gain in Inference over Multiple Groups of Items

Consider G groups of items such that, within each group, every pair of items is uniformly likely to get compared, but items from different group are never compared with each each other. As a baseline, one can run inference on each group separately. On the other hand, we propose running inference on all the G groups jointly. Let $\widehat{\Theta}$ be the estimate of Θ^* when solving the groups together, and let $\bar{\Theta}$ be the estimate when groups are estimated separately. Let L, $L^{(k)}$ be the graph Laplacians of the whole graph and k-th connected component (group) respectively. suppose, for simplicity, that $d_1 = d_2$ and the groups are equally sized complete sub-graph components,

$$L = \frac{1}{(d_2 - G)} \left(\mathbf{I}_{d_2 \times d_2} - \frac{G}{d_2} \sum_{k=1}^{G} g_k g_k^T \right), \text{ and,} \tag{21}$$

$$L^{(k)} = \frac{1}{(d_2/G - 1)} \left(\mathbf{I}_{\frac{d_2}{G} \times \frac{d_2}{G}} - \frac{G}{d_2} \mathbb{1}\mathbb{1}^T \right). \tag{22}$$

According to Theorems 1 and 4, the L-RMSE error of $\widehat{\Theta}$ satisfies,

$$\frac{1}{\sqrt{d_1}} \left\| \left(\Theta^* - \widehat{\Theta} \right) L^{1/2} \right\|_{\mathrm{F}} = \frac{1}{\sqrt{d_1 (d_2 - G)}} \left\| \Theta^* - \widehat{\Theta} \right\|_{\mathrm{F}} = \tilde{O} \left(\sqrt{\frac{r d_1}{n}} \right). \tag{23}$$

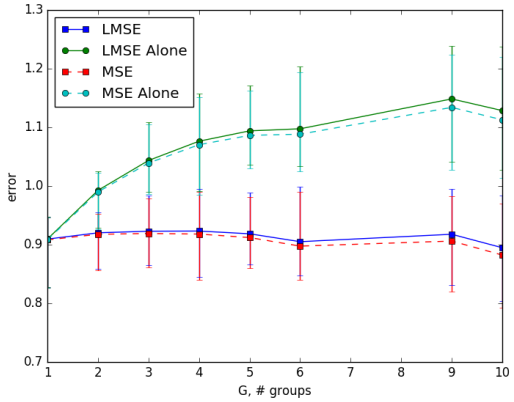


Figure 2: As the number of groups increase, the gain in joint inference increases.

Similarly L-RMSE error (with respect to the full Laplacian L) of $\bar{\Theta}$ satisfies,

$$\frac{1}{d_1} \left\| \left(\Theta^* - \bar{\Theta} \right) L^{1/2} \right\|_{\mathrm{F}}^2 = \frac{1}{d_1 (d_2 - G)} \left\| \Theta^* - \bar{\Theta} \right\|_{\mathrm{F}}^2 \tag{24}$$

$$\overset{(a)}{=} \frac{(d_2/G - 1)}{(d_2 - G)} \sum_{k=1}^{G} \frac{\left\| \left(\Theta^*_k - \bar{\Theta}_k \right) (L^{(k)})^{1/2} \right\|_{\mathrm{F}}^2}{d_2} \tag{25}$$

$$= \frac{1}{G} \sum_{k=1}^{G} \tilde{O} \left(\frac{r d_1}{n/G} \right) \tag{26}$$

$$= \tilde{O} \left(\frac{G r d_1}{n} \right), \tag{27}$$

where (a) follows from Eq. (22) and assuming $\Theta^*_k, \bar{\Theta}_k$ are sub-matrices restricted to the columns in group k. Thus the estimation errors when running joint inference and separate inference for each group are of the order of $O_G(1)$ and $O_G(\sqrt{G})$ respectively. That is, a user's preference in one group of items will be useful in inferring the same user's preference in another group of items. We illustrate this gain of joint inference in Figure 2. Concretely, the sampling graph $\mathcal{G}$ has G groups where each component is a complete graph and $d_1 = d_2 = 360$, $r = 4$, $\alpha = 5.0$, $n = 2^{14}$. Figure 2 plots the L-RMSE (RMSE) $= \left\| (\Theta - \widehat{\Theta}) L^{1/2} \right\|_{\mathrm{F}} / \sqrt{d_1}$ ($\left\| (\Theta - \widehat{\Theta}) \right\|_{\mathrm{F}} / \sqrt{d_1 \, d_2}$) errors vs. G, when all the groups are solved together (labelled as LMSE and MSE) or when the groups are solved separately (labelled as LMSE Alone and MSE Alone) using our algorithm. We see that solving the components together keeps the error relatively similar as the number of groups increase, but if we solve

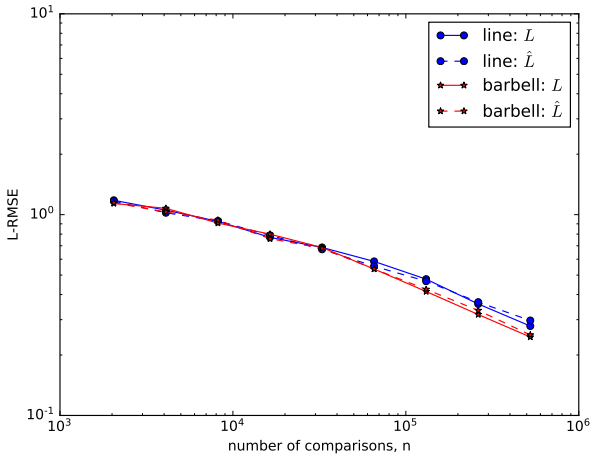


Figure 3: L-RMSE for various sampling graphs when true Laplacian L is known (solid) and when empirical Laplacian $\hat{L}$ is used (dashed)

.

the groups separately the error increases with number of groups, although it is at a lower rate than predicted by the upper bound.

3.4.4 Robustness to the Mismatched L-nuc Norm Regularizer

In practical scenarios, one might not have access to the sampling graph Laplacian L. We propose using empirical Laplacian $\hat{L}$ defined as

$$\hat{L} \triangleq \operatorname{diag}\big(\hat{P}\mathbb{1}\big) - \hat{P}\,, \tag{28}$$

where $\hat{P} \in \mathbb{R}^{d_2 \times d_2}$ is the empirical distribution of sampled pairs in the given data. Under the experimental setting from Figure 1b, we run additional experiments with this empirical Laplacian $\hat{L}$ in the optimization: minimize $-\mathcal{L}(\Theta) + \lambda \|\!|\Theta|\!\|_{\hat{L}\text{-nuc}}$. Figure 3 illustrates that the effect on the performance of not knowing the true L is marginal. Both approaches achieve the same error.

3.4.5 Real data: Food100

To showcase the practicality of our nuclear norm based algorithm (9) we apply our algorithm to the Food100 Data set[2] Wilber et al. (2014). In the data set, $n = 250320$ triplets, denoted by $\{(a_i, b_i, c_i)\}_i$, of 3 distinct food dishes from a selection of $d = 100$ were sampled. Then in

2. Data set is from `https://vision.cornell.edu/se3/projects/cost-effective-hits`.

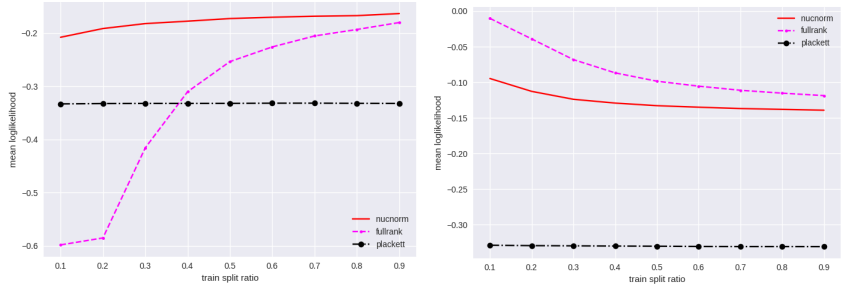
a crowdsourcing setting, users were asked if, a_i is more similar to b_i than to c_i. The goal is to learn an low-dimensional embedding of the 100 food items where the above similarities are captured. We model the problem as learning an MNL model, parameterized by Θ^*, which gives the following probability distribution for i-th user's answer,

$$\mathbb{P}\{a_i \text{ is more similar to } b_i \text{ than to } c_i\} \;=\; \frac{e^{\Theta^*_{a_i,b_i}}}{e^{\Theta^*_{a_i,b_i}} + e^{\Theta^*_{a_i,c_i}}}\,.$$

This is the same model as the pairwise comparisons from Section 2, except for the fact that instead of a user (row) comparing two items (columns), here we compare a food item (row) to two other food items (columns). We implement three different algorithms: our nuclear norm based algorithm ('nucnorm'), unregualrized ($\lambda = 0$) likelihood maximization ('fullrank') and maximum likelihood based algorithm to learn rank-1 Plackett-Luce model Luce (1959); Plackett (1975) ('plackett').

In Figure 4, we plot the mean log-likelihood of the learned model versus fraction x of the data used for training for the various algorithms for testing (a) and training (b) data. If x fraction of the data is used for training, we use the rest $(1-x)$ of the data for testing. For the nuclear norm minimization, we estimate the Laplacian L using the empirical distribution of the triplets and λ is chosen to be $0.1\sqrt{\log(d)/2\,d\,x n}$.

In the Fig. 4(b) (to the left) on the testing data set, we see that our MNL model based nuclear norm regularized algorithm clearly outperforms both unregularized algorithm and the Placket-Luce model estimator, especially when there is less training data. In fact, the mean likelihood ($\log(P_{\text{model}}(\text{test data}))$) on the testing data remains relatively the same when we decrease the size of the training data, which supports our claim that real data has low-rank structure. In the Fig. 4(b) (right) on the training data sets, the non-regularized approach of 'fullrank' achieves higher likelihood on the training data, indicating that it overfits to training data.



(a) Mean log-likelihood on testing data set (b) Mean likelihood on training data set

Figure 4: Mean log-likelihood vs fraction of the total data used for training. Our nuclear norm regularized algorithm ('nucnorm') fits the test data better than both unregularized algorithm ('fullrank') and Plackett-Luce model based estimation, especially when training data is small in size.

Figure 5: Food100: t-SNE embedding of the columns of the learned MNL parameter $\widehat{\Theta}$. Desserts (bottom left) are seperated from other dishes. Meat dishes are also separated from vegetable dishes.

In Fig. 5 we plot the t-SNE embedding Maaten and Hinton (2008) of the columns of the estimated MNL parameter matrix $\widehat{\Theta}$ when all the data is used for training. The desserts (left bottom) are separated from other dishes, and meat dishes and salad dishes form two clusters (top right).

4. Learning the MNL Model under Higher Order Comparisons

Higher order comparisons, where a subset of k items are offered to a user who then provides a complete ranking (total linear ordering) of those item, is a natural generalization of pairwise comparisons, that captures some aspect of heterogeneous and complex modern data sets. We refer to such scenarios as k-wise comparisons or k-wise rankings. The MNL model generalizes to such comparisons. Let Θ^* be the $d_1 \times d_2$ dimensional matrix capturing the preference of d_1 users on d_2 items, where the rows and columns correspond to users and items, respectively. In this k-wise ranking setting, when a user i is presented with a set,

$S_i \subseteq [d_2]$, of k alternatives she reveals her preferences as a ranked list over those items. To simplify the notations, we assume that all the users compare the same number k of items, but the analysis naturally generalizes to the case when the size might differ from a user to a user and when each user provides more than one k-wise ranking. Let $v_{i,\ell} \in S_i$ denote the (random) ℓ-th best choice of user i. Each user gives a ranking, independent of other users' rankings, from

$$\mathbb{P}\left\{v_{i,1}, \ldots, v_{i,k} | S_i \text{ is presented to user } i\right\} = \prod_{\ell=1}^{k} \frac{e^{\Theta^*_{i,v_{i,\ell}}}}{\sum_{j \in S_{i,\ell}} e^{\Theta^*_{i,j}}}\,, \tag{29}$$

where with $S_{i,\ell} \equiv S_i \setminus \{v_{i,1}, \ldots, v_{i,\ell-1}\}$ and $S_{i,1} \equiv S_i$. For a user i, the i-th row of Θ^* represents the underlying preference vector of the user, and the more preferred items are more likely to be ranked higher.

Similar to the pairwise comparisons, the distribution (29) is independent of shifting each row of Θ^* by a constant. Since we can only estimate Θ^* up to this equivalent class, we search for the one whose rows sum to zero, i.e. $\sum_{j \in [d_2]} \Theta^*_{i,j} = 0$ for all $i \in [d_1]$. For capturing the "spikiness" Negahban and Wainwright (2012) of Θ^*, we define $\alpha \equiv \max_{i,j_1,j_2} |\Theta^*_{ij_1} - \Theta^*_{ij_2}|$ to denote the dynamic range of the underlying Θ^*, such that when k items are compared, we always have

$$\frac{1}{k} e^{-\alpha} \;\le\; \frac{1}{1+(k-1)e^{\alpha}} \;\le\; \mathbb{P}\left\{v_{i,1} = j\right\} \;\le\; \frac{1}{1+(k-1)e^{-\alpha}} \;\le\; \frac{1}{k} e^{\alpha}\,, \tag{30}$$

for all $j \in S_i$, all $S_i \subseteq [d_2]$ satisfying $|S_i| = k$ and all $i \in [d_1]$. We do not make any assumptions on α other than that $\alpha = O(1)$ with respect to d_1 and d_2. Given this definition, we solve the following optimization

$$\widehat{\Theta} \in \arg\min_{\Theta \in \Omega_\alpha} -\mathcal{L}(\Theta) + \lambda |||\Theta|||_{\text{nuc}}, \tag{31}$$

where,

$$\mathcal{L}(\Theta) = \frac{1}{k\, d_1} \sum_{i=1}^{d_1} \sum_{\ell=1}^{k} \left(\langle\!\langle \Theta, e_i e_{v_{i,\ell}}^T \rangle\!\rangle - \log \left(\sum_{j \in S_{i,\ell}} \exp\left(\langle\!\langle \Theta, e_i e_j^T \rangle\!\rangle \right) \right) \right)\,, \tag{32}$$

over

$$\Omega_\alpha = \left\{ A \in \mathbb{R}^{d_1 \times d_2} \,\middle|\, |||A|||_\infty \le \alpha, \text{ and } \forall i \in [d_1] \text{ we have } \sum_{j \in [d_2]} A_{ij} = 0 \right\}. \tag{33}$$

Note that unlike graph sampling for pairwise comparisons, we assume that each user is presented a subset of k items and provides a complete ranking over those k items. This choice of sampling scenario, together with independent choices of the items in subset S_i's, is crucial for getting a bound that is tight in its scaling with respect to not only d_1, d_2, and r, but also k, as a certain independence is required to apply the symmetrization technique (in Lemma 29) which gives us the desired tight bound on the error. It trivially follows from our analysis that one can relax the assumptions in the sampling scenario significantly (e.g. sampling without replacement, heterogeneous sampling probabilities for each item-user pair, etc.), and the only change in the upper bound of Eq. (34) will be a weaker dependence k.

4.1 Performance Guarantee

We provide an upper bound on the resulting error of our convex relaxation, when a *multi-set* of items S_i presented to user i is drawn uniformly at random with replacement. Precisely, for a given k, $S_i = \{j_{i,1}, \ldots, j_{i,k}\}$ where $j_{i,\ell}$'s are independently drawn uniformly at random over the d_2 items. Further, if an item is sampled more than once, i.e. if there exists $j_{i,\ell_1} = j_{i,\ell_2}$ for some i and $\ell_1 \neq \ell_2$, then we assume that the user treats these two items as if they are two distinct items with the same MNL weights $\Theta^*_{i,j_{i,\ell_1}} = \Theta^*_{i,j_{i,\ell_2}}$. The resulting preference is therefore always over k items (with possibly multiple copies of the same item), and distributed according to (29). For example, if $k = 3$, it is possible to have $S_i = \{j_{i,1} = 1, j_{i,2} = 1, j_{i,3} = 2\}$, in which case the resulting ranking can be $(v_{i,1} = j_{i,1}, v_{i,2} = j_{i,3}, v_{i,3} = j_{i,2})$ with probability $(e^{\Theta^*_{i,1}})/(2\,e^{\Theta^*_{i,1}} + e^{\Theta^*_{i,2}}) \times (e^{\Theta^*_{i,2}})/(e^{\Theta^*_{i,1}} + e^{\Theta^*_{i,2}})$. Such a sampling with replacement is necessary for the analysis, where we require independence in the choice of the items in S_i in order to apply the symmetrization technique (e.g. Boucheron et al. (2013)) to bound the expectation of the deviation (cf. Appendix C.4). Similar sampling assumptions have been made in existing analyses on learning low-rank models from noisy observations, e.g. Negahban and Wainwright (2012). Let $d \equiv (d_1 + d_2)/2$, and let $\sigma_j(\Theta^*)$ denote the j-th singular value of the matrix Θ^*. Define

$$\lambda_0 \;\equiv\; e^{2\alpha}\sqrt{\frac{d_1 \log d + d_2\,(\log d)^2(\log 2d)^4}{k\,d_1^2\,d_2}}\,. \tag{34}$$

Theorem 7 *Under the described sampling model, assume* $24 \leq k \leq \min\{d_1^2 \log d, (d_1^2 + d_2^2)/(2d_1)\log d,\ (1/e)\,d_2(4\log d_2 + 2\log d_1)\}$*, and* $\lambda \in [480\lambda_0, c_0\lambda_0]$ *with any constant* $c_0 = O(1)$ *larger than 480. Then, solving the optimization* (31) *achieves*

$$\frac{1}{d_1 d_2}\left|\!\left|\!\left|\widehat{\Theta} - \Theta^*\right|\!\right|\!\right|_{\rm F}^2 \;\leq\; 288\sqrt{2}\,e^{4\alpha}c_0\lambda_0\sqrt{r}\left|\!\left|\!\left|\widehat{\Theta} - \Theta^*\right|\!\right|\!\right|_{\rm F} + 288e^{4\alpha}c_0\lambda_0\sum_{j=r+1}^{\min\{d_1,d_2\}}\sigma_j(\Theta^*)\,, \tag{35}$$

for any $r \in \{1, \ldots, \min\{d_1, d_2\}\}$ *with probability at least* $1 - 2d^{-3} - d_2^{-3}$ *where* $d = (d_1 + d_2)/2$.

A proof is provided in Appendix C. This bound holds for all values of r and one could potentially optimize over r. We show such results in the following corollaries.

Corollary 8 (Exact low-rank matrices) *Suppose* Θ^* *has rank at most* r*. Under the hypotheses of Theorem 7, solving the optimization* (31) *with the choice of the regularization parameter* $\lambda \in [480\lambda_0, c_0\lambda_0]$ *achieves with probability at least* $1 - 2d^{-3} - d_2^{-3}$*,*

$$\frac{1}{\sqrt{d_1 d_2}}\left|\!\left|\!\left|\widehat{\Theta} - \Theta^*\right|\!\right|\!\right|_{\rm F} \;\leq\; 288\sqrt{2}e^{6\alpha}c_0\sqrt{\frac{r(d_1 \log d + d_2\,(\log d)^2(\log 2d)^4)}{k\,d_1}}\,. \tag{36}$$

The number of entries is $d_1 d_2$ and we rescale the Frobenius norm error appropriately by $1/\sqrt{d_1 d_2}$. For a rank-r matrix Θ^* with $r(d_1 + d_2) - r^2 = O(r(d_1 + d_2))$ degrees of freedom, the above theorem shows that the total number of samples, which is $(k\,d_1)$, needs to scale

as $O(rd_1(\log d) + rd_2\,(\log d)^2(\log 2d)^4)$ in order to achieve an arbitrarily small error. This is only poly-logarithmic factor larger than the degrees of freedom. In Section 4.2, we provide a lower bound on the error directly, that matches the upper bound up to a logarithmic factor. The dependence on the dynamic range α is sub-optimal. The exponential dependence in the bound seems to be a weakness of the analysis, as seen from numerical experiments in the right panel of Figure 6. Although the error increase with α, numerical experiments suggests that it only increases at most linearly. A practical issue in achieving the above rate is the choice of λ, since the dynamic range α is not known in advance. Figure 6 illustrates that the error is not sensitive to the choice of λ for a wide range.

For approximately low-rank matrices in ℓ_q-ball defined in (13), optimizing the choice of r in Theorem 7, we get the following result. This is a strict generalization of Corollary 8 and a proof of this Corollary is provided in Appendix D.

Corollary 9 (Approximately low-rank matrices) *Suppose* $\Theta^* \in \mathbb{B}_q(\rho_q)$ *for some* $q \in (0, 1]$ *and* $\rho_q > 0$*. Under the hypotheses of Theorem 7, solving the optimization* (31) *with the choice of the regularization parameter* $\lambda \in [480\lambda_0, c_0\lambda_0]$ *achieves with probability at least* $1 - 2d^{-3}$*,*

$$\frac{1}{\sqrt{d_1 d_2}}\left|\!\left|\!\left|\widehat{\Theta} - \Theta^*\right|\!\right|\!\right|_{\rm F} \;\leq\; \frac{2\sqrt{\rho_q}}{\sqrt{d_1 d_2}}\left(288\sqrt{2}c_0 e^{6\alpha}\sqrt{\frac{d_1 d_2(d_1 \log d + d_2\,(\log d)^2(\log 2d)^4)}{k\,d_1}}\right)^{\frac{2-q}{2}}. \tag{37}$$

4.2 Information-theoretic Lower Bound for Low-rank Matrices

A simple parameter counting argument indicates that it requires the number of samples to scale as the degrees of freedom i.e., $kd_1 \propto r(d_1 + d_2)$, to estimate a $d_1 \times d_2$ dimensional matrix of rank r. By applying Fano's inequality with appropriately chosen hypotheses, the following lower bound establishes that the bound in Theorem 7 is sharp up to a logarithmic factor.

Theorem 10 *Suppose* Θ^* *has rank* r*. Under the described sampling model, for large enough* d_1 *and* $d_2 \geq d_1$*, there is a universal numerical constant* $c > 0$ *such that*

$$\inf_{\widehat{\Theta}}\sup_{\Theta^* \in \Omega_\alpha}\mathbb{E}\Big[\frac{1}{\sqrt{d_1 d_2}}\left|\!\left|\!\left|\widehat{\Theta} - \Theta^*\right|\!\right|\!\right|_{\rm F}\Big] \;\geq\; c\,\min\left\{\alpha e^{-\alpha}\sqrt{\frac{r\,d_2}{k\,d_1}}\,,\ \frac{\alpha d_2}{\sqrt{d_1 d_2 \log d}}\right\}, \tag{38}$$

where the infimum is taken over all measurable functions over the observed ranked lists $\{(v_{i,1}, \ldots, v_{i,k})\}_{i \in [d_1]}$.

A proof of this theorem is provided in Appendix E. The term of primary interest in this bound is the first one, which shows the scaling of the (rescaled) minimax rate as $\sqrt{r(d_1 + d_2)/(kd_1)}$ (when $d_2 \geq d_1$), and matches the upper bound in (35). It is the dominant term in the bound whenever the number of samples is larger than the degrees of freedom by a logarithmic factor, i.e., $kd_1 > r(d_1 + d_2)\log d$, ignoring the dependence on α. This is a typical regime of interest, where the sample size is comparable to the latent dimension of the problem. In this regime, Theorem 10 establishes that the upper bound in Theorem 7 is minimax-optimal up to a logarithmic factor in the dimension d.

4.3 Rank Breaking for Higher Order Comparisons

A common approach in practice to handle higher order comparisons is *rank breaking*, which refers to the practice of breaking the higher order comparisons into a set of pairwise comparisons and applying an estimator tailored for pairwise comparisons treating each pair as independent Azari Soufiani et al. (2013, 2014). When the higher order comparison is given as partial rankings (as opposed to total linear ordering as we assume) then rank breaking can be inconsistent, and special algorithms are needed for weighted rank breaking Khetan and Oh (2016a,b). However, when k-wise rankings (also called total linear orderings) are observed as we assume, simple and standard rank breaking achieves a similar performance as the higher order estimator in (31). Assume that $u_{i,m},\ i \in [d_1],\ m \in [k]$, denotes the m-th element observed by the i-th user. Concretely, in rank breaking, we convert the k-wise ranking data into pairwise ranking data and then we solve the following optimization problem:

$$\mathcal{L}(\Theta) = \frac{1}{d_1 \binom{k}{2}} \sum_{i \in [d_1]} \sum_{(m_1,m_2) \in \mathcal{P}_0} \left(\Theta_{i,\, h_i(m_1,m_2)} - \log\left(\exp\left(\Theta_{i,\, u_{i,m_1}} \right) + \exp\left(\Theta_{i,\, u_{i,m_2}} \right) \right) \right) \,, \tag{39}$$

where $\mathcal{P}_0 = \{(i,j) :\ 1 \le i < j \le k\}$, and $h_i\,(m_1,m_2)$ and $l_i\,(m_1,m_2)$ is defined as the higher and lower ranked index among u_{i,m_1} and u_{i,m_2} respectively. Then modified optimization problem becomes,

$$\widehat{\Theta} \ \in \ \arg\min_{\Theta \in \Omega_\alpha} -\mathcal{L}(\Theta) + \lambda |||\Theta|||_{\text{nuc}} \tag{40}$$

Let $d \equiv (d_1+d_2)/2$, and let $\sigma_j(\Theta^*)$ denote the j-th singular value of the matrix Θ^*. Define

$$\lambda_0 \ \equiv \ \sqrt{\frac{d \log d}{k\, d_1^2\, d_2}} \,. \tag{41}$$

With this choice of regularization coefficient, we get the following upper bounds on the rank breaking estimator (40) that are comparable to the upper bounds of k-wise ranking estimator in Theorem 7 and Corollary 8.

Theorem 11 *Under the described sampling model, assume* $2(c+4)\log d \le k \le \max\{d_1, d_2^2/d_1\}\log d$, $d_1 \ge 4$, *and* $\lambda \in [2\sqrt{32(c+4)}\lambda_0, c_p\lambda_0]$ *with any constant* $c = O(1)$ *larger than* $2\sqrt{32(c+4)}$. *Then, solving the optimization* (40) *achieves*

$$\frac{1}{d_1 d_2} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\text{F}}^2 \ \le \ 144\sqrt{2}\, e^{2\alpha} c\lambda\sqrt{r} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\text{F}} + 144 e^{2\alpha} c\lambda \sum_{j=r+1}^{\min\{d_1,d_2\}} \sigma_j(\Theta^*) \,, \tag{42}$$

for any $r \in \{1, \ldots, \min\{d_1,d_2\}\}$ *with probability at least* $1 - 2d^{-c} - 2d^{-2^{13}}$ *where* $d = (d_1+d_2)/2$.

A proof of this theorem is provided in Appendix F, and the following corollary follows for rank-r matrices.

Corollary 12 (Exact low-rank matrices) *Suppose* Θ^* *has rank at most* r. *Under the hypotheses of Theorem 11, there exists a constant* $c_1 > 0$ *such that solving the optimization* (40) *with the choice of the regularization parameter* $\lambda \in [2\sqrt{32(c+4)}\lambda_0, c\lambda_0]$ *achieves with probability at least* $1 - 2d^{-c} - 2d^{-2^{13}}$,

$$\frac{1}{\sqrt{d_1 d_2}} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\text{F}} \ \le \ 144\sqrt{2} e^{2\alpha} c_1 \sqrt{\frac{r d \log d}{k\, d_1}} \,. \tag{43}$$

4.4 Experiments

We provide results from numerical experiments on both synthetic and real data sets.

4.4.1 Algorithm

Similar to the case of pairwise comparisons in Section 3.4.1, we use proximal gradient descent Agarwal et al. (2010); Cai et al. (2010) along with modified Barzilai-Borwein (BB) step-size selection rule Barzilai and Borwein (1988) with the initial point $\Theta_0 = 0$. Each iteration of the algorithm applies the following two operations on the current estimate, Θ_t, of Θ^*,

$$\widetilde{\Theta}_{t+1} = \Theta_t - \eta_t \nabla_\Theta \mathcal{L}(\Theta_t) \qquad \text{(gradient descent)} \tag{44}$$

$$\Theta_{t+1} = M_t(\Gamma_t - \eta_t\lambda \mathbf{I})^+ N_t^T \qquad \text{(singular value shrinkage and thresholding)} \tag{45}$$

where $M_t\Gamma_t N_t^T := \widetilde{\Theta}_t$ is the singular value decomposition of $\widetilde{\Theta}_t$, such that Γ_t is a diagonal matrix with positive entries, $(\cdot)^+$ is the entry-wise thresholding operation $\max(0,x)$, and η_t is an BB step-size calculated as,

$$\eta_t = \begin{cases} |||\Theta_t - \Theta_{t-1}|||_2^2 / \langle\!\langle \Theta_t - \Theta_{t-1}, \nabla_\Theta \mathcal{L}(\Theta_t) - \nabla_\Theta \mathcal{L}(\Theta_{t-1}) \rangle\!\rangle, & \text{when } t \text{ is odd} \\ \langle\!\langle \Theta_t - \Theta_{t-1}, \nabla_\Theta \mathcal{L}(\Theta_t) - \nabla_\Theta \mathcal{L}(\Theta_{t-1}) \rangle\!\rangle / |||\nabla_\Theta \mathcal{L}(\Theta_t) - \nabla_\Theta \mathcal{L}(\Theta_{t-1})|||_2^2, & \text{when } t \text{ is even} \end{cases} . \tag{46}$$

4.4.2 Simulation: Higher Order Comparisons

The left panel of Figure 6 confirms the scaling of the error rate as predicted by Corollary 8. The lines merge to a single line when the sample size is rescaled appropriately (inset). We make a choice of $\lambda = \sqrt{(\log d)/(kd^2)}$. This choice is independent of α and is smaller than proposed in Theorem 7. We generate the random rank-r true MNL parameters matrices of dimension $d \times d$ using the process mentioned in Section 3.4.1. The root mean squared error (RMSE) is plotted where RMSE $= (1/\sqrt{d_1\, d_2})|||\Theta^* - \widehat{\Theta}|||_{\text{F}}$. We implement and solve the convex optimization (31) using proximal gradient descent method as analyzed in Agarwal et al. (2010). The right panel in Figure 6 illustrates that the actual error is insensitive to the choice of λ for a broad range of $\lambda \in [\sqrt{(\log d)/(kd^2)}, 2^8\sqrt{(\log d)/(kd^2)}]$, after which it increases with λ.

4.4.3 Simulation: Rank Breaking

In this section we compare the higher order k-wise comparison algorithm (31) ('kwise') with the pairwise rank breaking algorithm (40) ('kbreak'). We use the same setting as

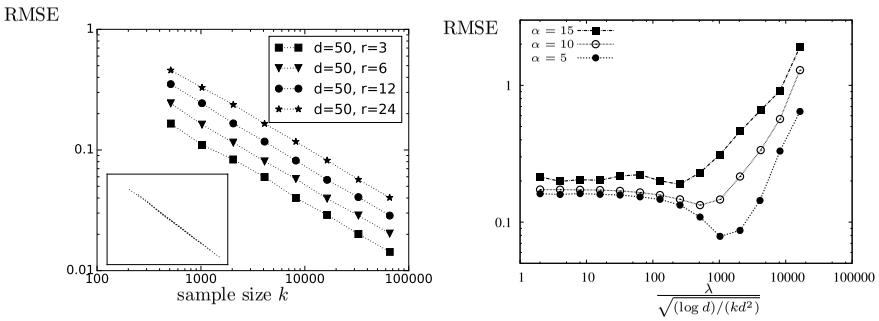


Figure 6: The (rescaled) RMSE scales as $\sqrt{r(\log d)/k}$ as expected from Corollary 8 for fixed $d = 50$ (left). In the inset, the same data is plotted versus rescaled sample size $k/(r \log d)$. The (rescaled) RMSE is stable for a broad range of λ and α for fixed $d = 50$ and $r = 3$ (right).

in Section 4.4.2, where we observe samples from k-wise ranking from an underlying true MNL model and the aim is to recover the true parameter Θ^* of the model. We use $\lambda = 0.45\sqrt{(\log d)/(kd^2)}$ and $\lambda = 0.1\sqrt{(\log d)/(kd^2)}$ for k-wise and pairwise rank breaking algorithms respectively. In Fig. 7 we plot the RMSE for both the algorithm for $d = 50$ and $r = 3, 12$. We note that the even though the RMSE decreases in the rate as predicted by the theorem, we see that pairwise rank breaking is worser than the higher order k-wise algorithm which directly uses the k-wise rankings. This is consistent with the experimental observation made previously in Hajek et al. (2014). Further we note that rank breaking is much slower than the other algorithm, since gradient computation of the former takes $O(k^2)$ time whereas for the latter it can be computed in $O(k)$ time.

4.4.4 Real data: Jester

Jester data set[3] Goldberg et al. (2001) has $24,982$ users, each rating a subset of 100 jokes on continuous scale of $[-10, 10]$. As the scale is continuous, we derive ordinal data from the scores (ties broken uniformly at random). We use only the 7200 users who rated all the jokes for our experiments. For each user, $k = 100x$ jokes were randomly selected uniformly at random for training, rest of the $100 - k = 100(1 - x)$ jokes where used for testing, where x is the fraction of jokes selected for training. We implment four algorithms: nuclear norm minimization ('nucnorm') (31), unregularized ($\lambda = 0$) log-likelihood maximization ('fullrank'), rank-1 Plackett-Luce model estimation ('plackett'), and rank breaking algorithm ('rankbreak') (40). We use $\lambda = 0.7\sqrt{(0.5\log(d_1 d_2))/(kd_1\sqrt{d_1 d_2})}$ and $\lambda = 0.16\sqrt{(0.5\log(d_1 d_2))/(kd_1\sqrt{d_1 d_2})}$ for k-wise and pairwise rank breaking algorithms respectively. In Fig. 8 (a) we plot the multiplicative bias in the mean log-likelihood on the testing data versus the fraction x of training data used. For each model in {'nucnorm',

3. Data set is from `http://eigentaste.berkeley.edu/dataset/`.

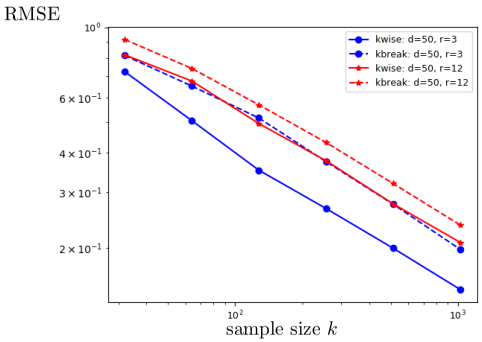


Figure 7: Rank Breaking: RMSE error versus number of samples per user k. k-wise ('kwise') algorithm performs better than the rank breaking ('kbreak') approach.

'fullrank', 'plackett', 'rankbreak'}, we plot in the y-axis

$$\frac{\log(P_{\text{model}}(\text{test data})) - \log(P_{\text{fullrank}}(\text{test data}))}{|\log(P_{\text{fullrank}}(\text{test data}))|},$$

using fullrank model as a baseline as it has the least test likelihood. Plackett-Luce model achieves the best performance when sample size is small, as this simplest model avoids overfitting. However, for most regimes of sample size, both the nuclear norm minimization and rank breaking achieve similar performance improving upon the others.

The same trend holds when we measure the perfomrance in the normalized Spearman's footule distance Diaconis and Graham (1977) $F(\pi_1, \pi_2) \in [0, 1]$ between two rank-lists π_1, π_2 of length k:

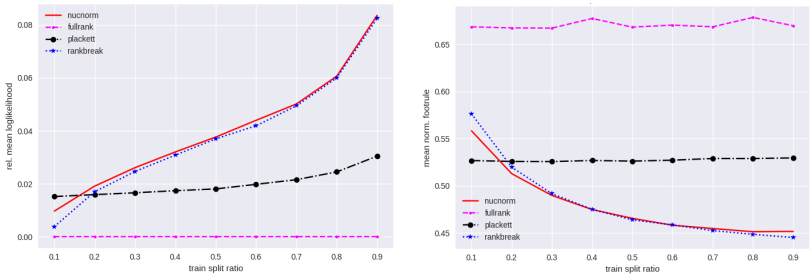
$$F(\pi_1, \pi_2) = \frac{2}{k^2}\sum_{i=1}^{k} |\pi_1(i) - \pi_2(i)|$$

In Fig. 8 (b) we plot the average normalized Spearman's footrule distance between the ground truths and the most likely ranking on the testing data under the estimated model parameters. We see that k-wise nuclear norm minimization and rank breaking algorithms perform the best in recovering the true ranking, except when the fraction of training data used is very small so that the rank-1 Plackett-Luce recovers better ranking.

4.4.5 Real data: Irish Election

The Irish Election data set[4] is an opinion poll conducted among 1083 participants during the 1997 Irish presidential election campaign Gormley and Murphy (2009). Each participant responded with a ranking the of their top 1, 2, 3, 4, or 5 choices from the 5 candidates:

4. Data set is from `https://projecteuclid.org/euclid.aoas/1231424218#supplemental`.



(a) Multiplicative bias in the mean log-likelihood (b) Spearman's footrule distance

Figure 8: Jester data set: Performance on test data vs. fraction of the total data used for training. The proposed nuclear norm regularized algorithm ('nucnorm') and rank breaking ('rankbreak') improves upon both the unregularized algorithm ('fullrank') and the Plackett-Luce model estimation ('plackett') for most regimes of the sample size.

Banotti, McAleese, Nally, Roche, and Scallon. For our experiments we use only the 807 participants who gave their top-5 choices, i.e. full-rankings of all the candidates. Next we divide these participants into 60 (2x3x5x2) group according to a Cartesian product of four categorizations: sex (male/female), marital status (single/married/widowed+divorced), social class (F/AB/C1/C2/DE)[5], location (rural/city+town). We assume that within each group the responses of its member follow the same distriubtion and these distributions of all all the groups are captured by an MNL model with parameter $\Theta^* \in \mathbb{R}^{60\times 5}$. We implement three algorithms: nuclear norm minimization ('nucnorm') (31), unregularized ($\lambda = 0$) log-likelihood maximization ('fullrank'), and rank-1 Plackett-Luce model estimation ('plackett'). We use $\lambda = 0.8\sqrt{(0.5\log(d_1 d_2))/(kd_1\sqrt{d_1 d_2})}$. If x randomly sampled fraction of the data is used for training, then rest of the data is used for testing. In Fig. 9 we plot the mean log-likelihood ($\log(P_{\text{model}}(\text{test data}))$) on the testing data versus the fraction of training data used. We see that nuclear norm minimization and Plackett-Luce model estimation tie for the first place and both improves significantly upon the un-regularized full-rank MNL model estimation. In Fig. 11 we plot the t-SNE Maaten and Hinton (2008) embedding of the rows of the estimated parameter matrix $\widehat{\Theta}$ when all the data is used for training. In Fig. 11a the markers represent the marital status of the group: single/married/divorced+widowed. In Fig. 11b the markers represent the social class of the groups. We see that married (left) and divorced+widowed (right) groups are clearly separated in the embedding, indicating that marital status influences the preference of candidates. However, we see that the social classes are less influential.

In Fig. 10 we represent the voting characteristics of the top 4 right singular vectors $\{\hat{v}_j\}_{j=1}^4$ of $\widehat{\Theta}$, which has a rank of 4 when all the data is used for training. The each stacked bar corresponds to the singular value σ_j marked on the x-axis. Partition of a bar represents the choice model distribution of the corresponding singular vector: $\exp(v_j)/(\mathbb{1}^T\exp(v_j))$,

5. Social classes are F: farmer, AB: middle class. C1: lower middle class, C2: skilled working class, and DE: other working class.

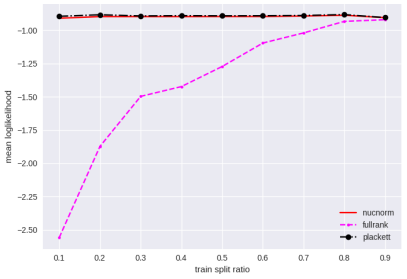


Figure 9: Irish Election: Mean log-likelihood on the test data versus fraction of the data used for training. Nuclear norm minimization and Plackett-Luce model estimator tie for the best performance.

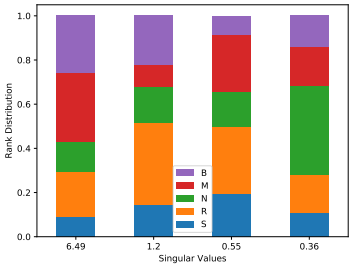
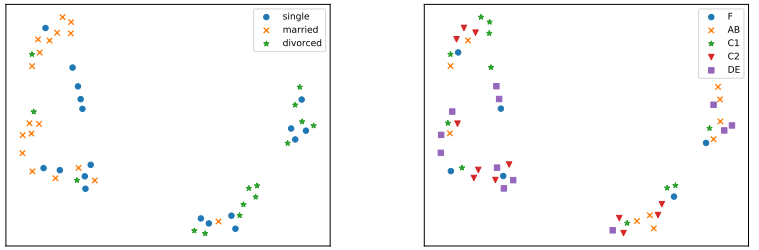


Figure 10: Irish Election: Each bar corresponds to rank distribution of one of the singular 4 values (x-axis) of the $\widehat{\Theta}$. Heights of the partitions represent the probability with which the distribution ranks the corresponding candidate as first (Section 4.4.5).

where $\exp(v_j)$ is the element-wise exponentiation operator. We see that there are 2 majors voting "basis" distributions; one favoring McAleese and another favoring Roche. Similar voting blocs have been observed earlier Gormley and Murphy (2009). Even though Placket-Luce model estimator achieves the similar likelihood as our nuclear norm regularized algorithm, the latter helps us in identifying voter "basis", solely from rank data without using the side-information on the voters as in Gormley and Murphy (2009).



(a) Martial status (b) Social class

Figure 11: Irish Election: t-SNE embedding of rows of the estimated parameter matrix $\widehat{\Theta}$. The markers correspond to the marital or social status (F: farmer, AB: middle class. C1: lower middle class, C2: skilled working class, DE: other working class) of the rows.

5. Learning the MNL Model from Choices

Choice modeling has had widespread success in numerous application domains such as transportation and marketing Train (1986); Guadagni and Little (1983). Choice models stem from revenue management to tackle the fundamental problem of maximizing expected revenue where the expectation is taken over a probabilistic choice model that is learned from historical purchase data. Revenue management has focused on designing efficient solvers for the optimization problem with exact or approximation guarantees, and has less to do with *learning* the parameters of probabilistic choice model of interest.

In this section, we tackle this unexplored domain of learning choice models from samples with provable guarantees on the sample complexity. In particular, we study learning the MNL model from choices. We study two types of choices under the MNL model that together include all practical scenarios of interest: *bundled choice* and *consumer choice*.

Bundled choice. We consider a novel scenario of significant practical interest: choice modeling from bundled purchase history. In this setting, we assume that we have bundled purchase history data from n users. Precisely, there are two categories of interest with d_1 and d_2 alternatives in each category respectively. For example, there are d_1 tooth pastes to choose from and d_2 tooth brushes to choose from. For the i-th user, a subset $S_i \subseteq [d_1]$ of alternatives from the first category is presented along with a subset $T_i \subseteq [d_2]$ of alternatives from the second category. We use k_1 and k_2 to denote the number of alternatives presented to a single user, i.e. $k_1 = |S_i|$ and $k_2 = |T_i|$, and we assume that the number of alternatives presented to each user is fixed, to simplify notations. However, the analysis naturally generalizes if the number differs from a user to another user. Given these sets of alternatives, each user makes a 'bundled' purchase, of an item from S_i and another item from T_i together, and we use (u_i, v_i) to denote these bundled pair of alternatives (e.g. a tooth brush and a tooth paste) purchased by the i-th user. Each user makes a choice of the best alternative, independent of other users's choices, according to the MNL model as

$$\mathbb{P}\left\{(u_i, v_i) = (j_1, j_2)\right\} \;=\; \frac{e^{\Theta^*_{j_1,j_2}}}{\sum_{j_1' \in S_i, j_2' \in T_i} e^{\Theta^*_{j_1',j_2'}}}\,, \tag{47}$$

for all $j_1 \in S_i$ and $j_2 \in T_i$. We emphasize here that the preference matrix is indexed by items of type one (in the rows) and items of type two (in the columns). We are taking the existing standard MNL model over user-item pairs to propose a novel choice model for bundled purchases over two types of items. One could go beyond paired bundled choices and include the user identity as another dimension, or add other types of items and consider higher order bundled purchases. This would require MNL model over higher order tensors, which is outside the scope of this paper, but are interesting generalizations. The main challenge in learning such tensor MNL models is that nuclear norm of a higher order tensor is not a computable quantity and hence minimizing the nuclear norm is not algorithmically feasible Yuan and Zhang (2014). Efficient methods exist based on alternating minimizations, but existing analysis tools can handle only quadratic losses Jain and Oh (2014).

The distribution (47) is independent of shifting all the values of Θ^* by a constant. Hence, there is an equivalent class of Θ^* that gives the same distribution for the choices: $[\Theta^*] \equiv \{A \in \mathbb{R}^{d_1 \times d_2} \,|\, A = \Theta^* + c\mathbb{1}\mathbb{1}^T \text{ for some } c \in \mathbb{R}\}$. Since we can only estimate Θ^* up to

this equivalent class, we search for the one that sums to zero, i.e. $\sum_{j1 \in [d_1], j_2 \in [d_2]} \Theta^*_{j_1,j_2} = 0$. Let $\alpha = \max_{j_1,j_1' \in [d_1], j_2, j_2' \in [d_2]} |\Theta^*_{j_1,j_2} - \Theta^*_{j_1',j_2'}|$, denote the dynamic range of the underlying Θ^*, such that when $k_1 \times k_2$ alternatives are presented, we always have

$$\frac{1}{k_1 k_2} e^{-\alpha} \;\le\; \mathbb{P}\left\{(u_i, v_i) = (j_1, j_2)\right\} \;\le\; \frac{1}{k_1 k_2} e^{\alpha}\,, \tag{48}$$

for all $(j_1, j_2) \in S_i \times T_i$ and for all $S_i \subseteq [d_1]$ and $T_i \subseteq [d_2]$ such that $|S_i| = k_1$ and $|T_i| = k_2$. We do not make any assumptions on α other than that $\alpha = O(1)$ with respect to d_1 and d_2. Assuming Θ^* is well approximated by a low-rank matrix, we solve the following convex relaxation, given the observed bundled purchase history $\{(u_i, v_i, S_i, T_i)\}_{i \in [n]}$:

$$\widehat{\Theta} \;\in\; \arg\min_{\Theta \in \Omega_\alpha} \mathcal{L}(\Theta) + \lambda |||\Theta|||_{\text{nuc}}\,, \tag{49}$$

where the negative log likelihood function according to (47) is

$$\mathcal{L}(\Theta) \;=\; -\frac{1}{n} \sum_{i=1}^{n} \left(\langle\!\langle \Theta, e_{u_i} e_{v_i}^T \rangle\!\rangle - \log \left(\sum_{j_1 \in S_i, j_2 \in T_i} \exp\left(\langle\!\langle \Theta, e_{j_1} e_{j_2}^T \rangle\!\rangle \right) \right) \right), \text{ and} \tag{50}$$

$$\Omega_\alpha \;\equiv\; \left\{ A \in \mathbb{R}^{d_1 \times d_2} \,\middle|\, |||A|||_\infty \le \alpha, \text{ and } \sum_{j_1 \in [d_1], j_2 \in [d_2]} A_{j_1,j_2} = 0 \right\}. \tag{51}$$

Compared to collaborative ranking, (a) rows and columns of Θ^* correspond to an alternative from the first and second category, respectively; (b) each sample corresponds to the purchase choice of a user which follow the MNL model with Θ^*; (c) each person is presented subsets S_i and T_i of items from each category; (d) each sampled data represents the most preferred bundled pair of alternatives.

Customer choice. The standard customer choice can be thought of as either a special case of *bundled choice* or as a special case of *higher order comparisons*. We consider the standard customer choice data from purchase history. In this setting, we assume that we have purchase history data from d_1 users over d_2 alternatives. The i-th sample is i.i.d. with user u_i chosen uniformly at random and a subset $S_i \subseteq [d_2]$ of alternatives of size k. We fix k in order to be efficient in the notations and any variable size offerings can be handled seamlessly. We assume S_i is chosen uniformly at random with replacement, in a similar way as bundled choice and higher order comparisons.

Given these sets of alternatives, the user u_i makes a 'choice' and we use v_i to denote the purchased alternative by the i-th (sampled) user. Each user makes a choice of the best alternative, independent of other users's choices, according to the MNL model as

$$\mathbb{P}\left\{v_i = j_2 | u_i = j_1\right\} \;=\; \frac{e^{\Theta^*_{j_1,j_2}}}{\sum_{j_2' \in S_i} e^{\Theta^*_{j_1,j_2'}}}\,, \tag{52}$$

for all $j_2 \in S_i$. Up to the fact that we index rows by users and not items of one category, this is a special case of the *bundled choice* model where we fix $k_1 = 1$. Mathematically, all of our results under consumer choices are derived as corollaries from our results under bundled choices, but given the prevalent interest in customer choice models, we emphasize the implications of our framework under customer choice models in a separate section (see Section 5.2).

5.1 Learning the MNL Model from Bundled Choices

We provide an upper bound on the error achieved by our convex relaxation, when the *multi-set* of alternatives S_i from the first category and T_i from the second category are drawn uniformly at random with replacement from $[d_1]$ and $[d_2]$ respectively. Precisely, for given k_1 and k_2, we let $S_i = \{j_{1,1}^{(i)}, \ldots, j_{1,k_1}^{(i)}\}$ and $T_i = \{j_{2,1}^{(i)}, \ldots, j_{2,k_2}^{(i)}\}$, where $j_{1,\ell}^{(i)}$'s and $j_{2,\ell}^{(i)}$'s are independently drawn uniformly at random over the d_1 and d_2 alternatives, respectively. Similar to the previous section, this sampling with replacement is necessary for the analysis. Define

$$\lambda_0 = \sqrt{\frac{e^{2\alpha} \max\{d_1, d_2\} \log d}{n\, d_1\, d_2}}\,. \tag{53}$$

Theorem 13 *Under the described sampling model, assume* $16e^{2\alpha} \min\{d_1, d_2\} \log d \le n$ *and* $n \le \min\{d^5, k_1 k_2 \max\{d_1^2, d_2^2\}\} \log d$*, and* $\lambda \in [8\lambda_0, c_1\lambda_0]$ *with any constant* $c_1 = O(1)$ *larger than* $\max\{8, 128/\sqrt{\min\{k_1, k_2\}}\}$*. Then, solving the optimization* (49) *achieves*

$$\frac{1}{d_1 d_2} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}}^2 \;\le\; 48\sqrt{2}\, e^{2\alpha} c_1 \lambda \sqrt{r} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} + 48 e^{2\alpha} c_1 \lambda \sum_{j=r+1}^{\min\{d_1,d_2\}} \sigma_j(\Theta^*)\,, \tag{54}$$

for any $r \in \{1, \ldots, \min\{d_1, d_2\}\}$ *with probability at least* $1 - 2d^{-3}$ *where* $d = (d_1 + d_2)/2$.

A proof is provided in Appendix G. Optimizing over r gives the following corollaries.

Corollary 14 (Exact low-rank matrices) *Suppose* Θ^* *has rank at most* r*. Under the hypotheses of Theorem 13, solving the optimization* (49) *with the choice of the regularization parameter* $\lambda \in [8\lambda_0, c_1\lambda_0]$ *achieves with probability at least* $1 - 2d^{-3}$*,*

$$\frac{1}{\sqrt{d_1 d_2}} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} \;\le\; 48\sqrt{2} e^{3\alpha} c_1 \sqrt{\frac{r(d_1 + d_2)\log d}{n}}\,. \tag{55}$$

This corollary shows that the number of samples n needs to scale as $O(r(d_1 + d_2)\log d)$ in order to achieve an arbitrarily small error. This is only a logarithmic factor larger than the number of degrees of freedom. For approximately low-rank matrices in an ℓ_1-ball as defined in (13), we show an upper bound on the error, whose error exponent reduces from one to $(2 - q)/2$.

Corollary 15 (Approximately low-rank matrices) *Suppose* $\Theta^* \in \mathbb{B}_q(\rho_q)$ *for some* $q \in (0, 1]$ *and* $\rho_q > 0$*. Under the hypotheses of Theorem 13, solving the optimization* (49) *with the choice of the regularization parameter* $\lambda \in [8\lambda_0, c_1\lambda_0]$ *achieves with probability at least* $1 - 2d^{-3}$*,*

$$\frac{1}{\sqrt{d_1 d_2}} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} \;\le\; \frac{2\sqrt{\rho_q}}{\sqrt{d_1 d_2}} \left(48\sqrt{2} c_1 e^{3\alpha} \sqrt{\frac{d_1 d_2 (d_1 + d_2)\log d}{n}} \right)^{\frac{2-q}{2}}\,. \tag{56}$$

This follows from the same line of proof as in the proof of Corollary 9 in Appendix D. We next, provide a fundamental lower bound on the error, that matches the upper bound up to a logarithmic factor.

Theorem 16 *Suppose* Θ^* *has rank* r*. Under the described sampling model, there is a universal constant* $c > 0$ *such that that the minimax rate where the infimum is taken over all measurable functions over the observed purchase history* $\{(u_i, v_i, S_i, T_i)\}_{i\in[n]}$ *is lower bounded by*

$$\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\frac{1}{\sqrt{d_1 d_2}} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} \Big] \;\ge\; c \min\left\{ \sqrt{\frac{e^{-5\alpha}\, r\,(d_1 + d_2)}{n}}\,,\; \frac{\alpha(d_1 + d_2)}{\sqrt{d_1 d_2 \log d}} \right\}\,. \tag{57}$$

We provide a proof in Appendix H. The first term is dominant, and when the sample size is comparable to the latent dimension of the problem, Theorem 13 is minimax optimal up to a logarithmic factor. We emphasize here that the bound in (55) and the matching lower bound in (57) do not depend on the size of the offerings k_1 and k_2. It is independent of how large k_1 and k_2 are because, we only observe one choice, and intuitively the information we get scales at best by a factor of $\log(k_1 k_2)$. The theorems prove that there is no essential gain in learning from large offerings. One might be tempted to stop at proving an upper bound that scales as $O(\sqrt{k_1 k_2 r(d_1 + d_2)\log d/n})$, which is larger than (55) by a factor of $\sqrt{k_1 k_2}$. Such a loose bound follows if one ignores the tight concentration analysis that we do using the symmetrization technique (e.g. in Lemma 29). Getting the tight dependency in k_1 and k_2 is one of the crucial technical challenges we overcome in this paper.

5.2 Learning the MNL Model from Customer Choices

The results for the *customer choice* model follow immediately from the results in *bundled choice* model by simply setting $k_1 = 1$, and we explicitly write those corollaries in this section for completeness. The proposed estimator is minimax optimal up to a logarithmic factor under the standard customer choice model of sampling.

Corollary 17 *Under the described sampling model, assume* $16e^{2\alpha} \min\{d_1, d_2\} \log d \le n \le \min\{d^5, k \max\{d_1^2, d_2^2\}\} \log d$*, and* $\lambda \in [8\lambda_0, c_1\lambda_0]$ *with any constant* $c_1 = O(1)$ *larger than* 128*. Then, solving the optimization* (49) *achieves*

$$\frac{1}{d_1 d_2} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}}^2 \;\le\; 48\sqrt{2}\, e^{2\alpha} c_1 \lambda \sqrt{r} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} + 48 e^{2\alpha} c_1 \lambda \sum_{j=r+1}^{\min\{d_1,d_2\}} \sigma_j(\Theta^*)\,, \tag{58}$$

for any $r \in \{1, \ldots, \min\{d_1, d_2\}\}$ *with probability at least* $1 - 2d^{-3}$ *where* $d = (d_1 + d_2)/2$.

Corollary 18 (Exact low-rank matrices) *Suppose* Θ^* *has rank at most* r*. Under the hypotheses of Theorem 17, solving the optimization* (49) *with the choice of the regularization parameter* $\lambda \in [8\lambda_0, c_1\lambda_0]$ *achieves with probability at least* $1 - 2d^{-3}$*,*

$$\frac{1}{\sqrt{d_1 d_2}} \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{F}} \;\le\; 48\sqrt{2} e^{3\alpha} c_1 \sqrt{\frac{r(d_1 + d_2)\log d}{n}}\,. \tag{59}$$

Corollary 19 (Approximately low-rank matrices) *Suppose $\Theta^* \in \mathbb{B}_q(\rho_q)$ for some $q \in (0,1]$ and $\rho_q > 0$. Under the hypotheses of Theorem 17, solving the optimization (49) with the choice of the regularization parameter $\lambda \in [8\lambda, c_1\lambda]$ achieves with probability at least $1 - 2d^{-3}$,*

$$\frac{1}{\sqrt{d_1 d_2}} \left\| \widehat{\Theta} - \Theta^* \right\|_{\mathrm{F}} \leq \frac{2\sqrt{\rho_q}}{\sqrt{d_1 d_2}} \left(48\sqrt{2} c_1 e^{3\alpha} \sqrt{\frac{d_1 d_2 (d_1 + d_2) \log d}{n}} \right)^{\frac{2-q}{2}} . \tag{60}$$

We emphasize again that the bound in (59) does not depend on the size of the offerings k. It is significantly easier to stop at proving an upper bounds that scale as $O(\sqrt{kr(d_1+d_2)\log d/n})$, which are larger than (59) by a factor of $\sqrt{k}$. Such a loose bound follows if one ignores the tight concentration analysis that we do using the symmetrization technique (e.g. in Lemma 29). Getting the tight dependency in k is one of the crucial technical challenges we overcome in this paper.

5.3 Experiments

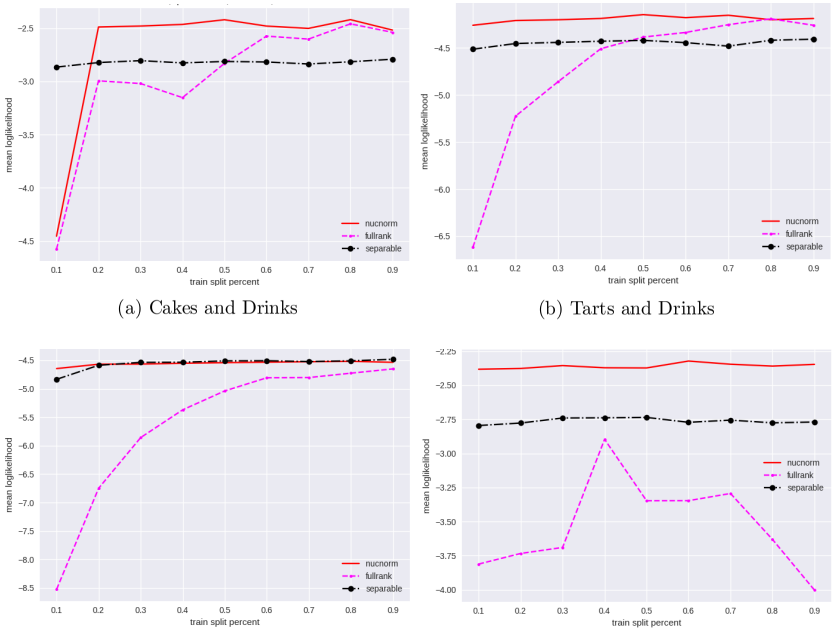
We applied our algorithm to a real world choice data set. The implementation is similar to that of higher order comparisons (see Section 4.4.1).

5.3.1 Real data: Extended Bakery

Extended Bakery data set [6] Benson et al. (2018) consists of details of 75,000 purchases at a bakery from a selection of 50 items, specifically each purchase is recorded by the set of items bought together. We use only the 13,579 purchases where a pair of items where bought. We divide the items into five categories: cakes (1-10), tarts (11-21), cookies (22-30), pastries (31-40) and drinks (41-50). We study four cases of bundled pairs, cakes and drinks, tarts and drinks, cookies and drinks, and pastries and drinks. These cases have 1503, 910, 500 and 1791 purchases in them respectively. We model the data with an MNL model parameterized by a matrix Θ^*, such that rows and columns corresponds to first and second categories respectively. For every purchase we assume that the subset of alternatives presented is the universal choice set, that is $k_1 = d_1$ and $k_2 = d_2$, so that the purchase of item j_1 from category 1 and j_1 from category 2 has a probability of $\exp(\Theta^*_{j_1,j_2}) / \sum_{j_1',j_2'=1}^{d_1,d_2} \exp(\Theta^*_{j_1',j_2'})$, where d_1, d_2 are number of items in category 1 and 2 respectively. We also fit the separable model proposed in Benson et al. (2018), which is a simpler model with $\Theta^*_{j,j_2} = a^*_{j_1} + b^*_{j_2}$, where $a \in \mathbb{R}^{d_1}, b \in \mathbb{R}^{d_2}$.

In Fig. 12 we plot the mean log-likelihood on the test data versus the fraction of the data used for our nuclear norm minimization based algorithm ('nucnorm'), un-regularized log-likelihood maximization algorithm ('fullrank'), and maximum likelihood estimator for the separable model ('separable') over 10 trials. We see that nuclear norm minimization outperforms the 'fullrank' and 'separable' algorithms. This is consistent with the marketing practice, of providing different prices for bundled combinations of products, which uses the rationale that, worth of a bundle of products might be different from the sum of the worths

6. Data set is from `https://github.com/arbenson/discrete-subset-choice/tree/master/data`.



(a) Cakes and Drinks (b) Tarts and Drinks

(c) Cookies and Drinks (d) Pastries and Drinks

Figure 12: Bakery: Mean log-likelihood on test data versus fraction of data used for training. Nuclear norm minimization improves upon the un-regularized likelihood maximization and separable model, for most of the regimes we consider.

of the individual products constituting the bundle. We also note that the un-regularized algorithm ('fullrank') has the worst performance and high variance and separable model fits the samples almost as good as the nuclear norm minimization in the case of cookies and drinks in the large training data regime.

6. Conclusion

The sample complexity of learning one of the most popular choice models known as Multi-Nomial Logit model has not been addressed in the literature. The main challenge is in the inherent low-rank structure of the parameter to be learned, which leads to a non-convex likelihood maximized problem. Thanks to recent advances in learning low-rank matrices, in particular in 1-bit matrix completion Davenport et al. (2014), matrix completion Negahban and Wainwright (2012), and restricted strong convexity Negahban et al. (2009), we have a polynomial time algorithm and the technical tools to characterize the fundamental sample complexity of learning MNL from samples. This provides a novel algorithm to learn a

low-dimensional representation of users and items from users' historical comparisons and choices. We study three types of data, pairwise comparison, higher order comparison, and choices, and take the first principle approach of identifying the fundamental limits and also developing efficient algorithms matching those fundamental trade offs. We provide a unifying framework to learn the latent preferences by solving a convex program. For each of the data types, accompanied by natural sampling scenarios, we show that our framework achieves a minimax optimal performance, and hence cannot be improved upon other than a small logarithmic factor. This opens a new door to learn representations from comparisons and choices, and we propose new research directions and challenges below. Beyond the low-rank model studied in this paper, recent advances in modeling data in a matrix form such as low algebraic dimension by Ongie et al. (2018) and non-parametric approximation by Borgs et al. (2017) can provide new research directions for modeling choice.

Efficient implementations via non-convex optimization. Nuclear norm minimization, while polynomial-time, is still slow. We want first-order methods that are efficient with provable guarantees. Two main challenges are providing a good initialization to start such non-convex approaches and analyzing gradient descent on the likelihood maximization which is non-convex.

Recent advances in non-convex optimization with rank-constraints have developed via a sequence of innovations that can be summarized as follows, in a number of example problems including matrix completion, robust PCA, matrix sensing, phase retrieval. First, a convex relaxation of nuclear norm minimization is analyzed, e.g. Candès and Recht (2009). Then, a more efficient two-step non-convex optimization approach is proposed with provable guarantees where a global initialization step is followed by a first-order method e.g. Keshavan et al. (2010a,b). Next, first-order methods starting at any initialization point is analyzed via understanding the geometry and checking the stationary points of the objective function e.g. Ge et al. (2016). This recipe, spurred by the advances in the matrix completion problem, has been repeated for several interesting problems involving low rank matrices, over the last decade and over numerous publications by collective effort of the machine learning community.

For the problem of learning MNL, we are at the first stage of this progression where we propose a convex relaxation and provide minimax optimal guarantees. We currently do not have the analysis tools to follow up in analyzing an efficient non-convex optimization problem, although writing the algorithm and implementing is straight forward, and also has been proposed in Park et al. (2015). It is a promising research direction to overcome the challenges in analyzing non-convex optimization methods for the MNL likelihood objective function.

Assumption on sampling with replacement. As mentioned earlier, we assume sampling with replacement, where we can ask a user to compare the same pair more than once, and also we can ask a user to compare two copies of the identical item. Although such sampling with replacement does not happen in practice, the number of such collisions is also very low with high probability under the proposed model. Further, such assumption is critical for getting an upper bound that is tight not only in r, d_1 and d_2, but also in k for higher order comparisons and choices. If, instead, one is interested in sampling without

replacement, then one either can resort to proving a loose bound that is weaker in its dependence in k (and follows trivially as a corollary of the proof of our results) or needs to invent new innovative concentration bounds that do not rely on the powerful symmetrization. The first option is trivial, so we do not provide such corollaries in this paper, and the second option provides an interesting but technically challenging question of resolving between sampling with replacement and sampling without replacement. This we believe is outside the scope of this paper.

Modern data analysis applications. As learning representation from ordinal data is of fundamental interest, there are numerous exciting applications that both the algorithmic framework and also the analysis techniques we develop could be naturally extended to. We present two such examples. First is a recent application of embedding objects with crowdsourced similarity measures, first proposed in Tamuz et al. (2011). Consider a crowdsourcing setting where you have d images and want to learn similarities among those images such that one can embed those images in a lower dimensional Euclidean space. One can show to a person a triplet of images (i_1, i_2, i_3) and ask whether the image in the middle is more similar to the one in the left or the right. A natural model proposed in Tamuz et al. (2011) is to assume that there exists a similarity parameter matrix $\Theta \in \mathbb{R}^{d\times d}$ such that

$$\mathbb{P}\left\{i_1 \text{ is more similar to } i_2 \text{ than } i_3 \text{ is to } i_2\right\} \;=\; \frac{e^{\Theta_{i_2,i_1}}}{e^{\Theta_{i_2,i_1}} + e^{\Theta_{i_2,i_3}}}\,.$$

A heuristic algorithm is proposed to learn a low-rank Θ without guarantees. Given the similarity of this model to MNL in (1), both our algorithm and also the analysis will go through to provide a tight characterization of the sample complexity of this problem.

The second application is in word embedding Mikolov et al. (2013b), where the goal is to find embeddings for English words in a lower dimensional Euclidean space. The most successful word embedding has been based on fitting a low-rank matrix $\Theta \in \mathbb{R}^{d\times d}$ where d is the size of the vocabulary, over an MNL-type model:

$$\mathbb{P}\{\text{word } i \text{ and word } j \text{ appear within distance ten}|\text{word } j \text{ appear in a sentence}\} \;=\; \frac{e^{\Theta_{ij}}}{\sum_{i'} e^{\Theta_{i'j}}}\,.$$

As the denominator involves summation over millions of words in the vocabulary, efficient heuristics are proposed to learn such a model from skip-grams; a skip-gram is the count matrix counting how many times words co-appear in the same sentence within a predefined distance. There are several challenges in applying our framework directly to such a setting mainly due to the size of the problem, but nevertheless our analysis can be applied directly to identify the fundamental minimax sample complexity of learning a word embedding from skip-grams.

Acknowledgments

SN acknowledges support from NSF Grant DMS 1723128. SO and KT acknowledge support from NSF grants CCF 1553452, CNS 1527754, CCF 1705007, and RI 1815535. The authors acknowledge Yu Lu and Ruoyu Sun for helpful and fruitful discussions.

Appendix A. Proof of the Upper Bound for Graph Sampling Theorem 1

The proof of the theorem relies on the following two lemmas. First lemma shows that the negative of the log-likelihood satisfies Restricted Strong Convexity with high probability.

Lemma 20 ***(Restricted Strong Convexity)*** *Let* $R = \max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$ *and the set* $\mathcal{A}(\alpha) = \left\{\Theta \in \mathbf{R}^{d_1\times d_2}, |||\Theta|||_\infty \le \alpha, |||\Theta|||_{\text{L-nuc}} \le \frac{|||\Theta L^{1/2}|||_{\text{F}}^2}{16\alpha d_1 R}\right\}$. *Then we have,*

$$\frac{1}{n}\sum_{i=1}^{n}(\langle\!\langle\Theta, X_i\rangle\!\rangle)^2 \ge \frac{1}{3d_1}|||\Theta|||_{\text{L}}^2, \quad \forall\ \Theta \in \mathcal{A}(\alpha), \tag{61}$$

with probability at least $1 - 2(2d)^{-4}$, *provided that* $n \le \log(2d)\min\{2^2(d_1\sigma_{\min}(L)^{-1})^{2/3}, 2^6 d_1^2\sigma^2\}$, .

Here the upper bound on n may not be necessary, but it is present due to a technical difficulty in using the peeling argument. The intuition behind the above lemma is that the empirical average uniformly concentrates around its expectation. Proof is in Section A.1. The next lemma says that the gradient of the log-likelihood at the actual parameter matrix, Θ^* is controllably small.

Lemma 21 ***(Bounded Gradient)*** *Let* $R = \max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$. *The spectral norm of gradient of the log-likelihood at the actual parameter matrix,* $\nabla\mathcal{L}(\Theta^*)$, *can be upper-bounded with high probability as follows,*

$$\mathbb{P}\left\{\left\|\nabla\mathcal{L}(\Theta^*)L^{-1/2}\right\|_2 \ge \sqrt{32}R\right\} \le \frac{1}{(d_1+d_2)^3} \tag{62}$$

Proof the above lemma is in Section A.4. Let $\Delta = \hat{\Theta} - \Theta^*$.
Case 1: $\Delta \notin \mathcal{A}(2\alpha)$ Then,

$$|||\Delta|||_{\text{L}}^2 \le 32\alpha d_1 R|||\Delta|||_{\text{L-nuc}}$$

Case 2: $\Delta \in \mathcal{A}(2\alpha)$ We first write down the second order Taylor series expansion of $\mathcal{L}(\hat{\Theta})$ at around $\Theta = \Theta^*$.

$$-\mathcal{L}(\hat{\Theta}) = -\mathcal{L}(\Theta^*) + \langle\!\langle -\nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle + \frac{1}{2n}\sum_{i=1}^{n}\psi\left(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle + s\langle\!\langle\Delta, X^{(i)}\rangle\!\rangle\right)\langle\!\langle\Delta, X^{(i)}\rangle\!\rangle^2, \tag{63}$$

where $\psi(x) = e^x/(1+e^x)^2$, $x \in [-2\alpha, 2\alpha]$ and $s \in [0,1]$. Next using Lemma 20 and the fact that $\psi(x)$ attains minimum at $x = 2\alpha$ we get,

$$-\mathcal{L}(\hat{\Theta}) + \mathcal{L}(\Theta^*) + \langle\!\langle\nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle \ge \frac{1}{2n}\sum_{i=1}^{n}\psi(2\alpha)\langle\!\langle\Delta, X^{(i)}\rangle\!\rangle^2 \ge \frac{\psi(2\alpha)}{6d_1}|||\Delta|||_{\text{L}}^2, \tag{64}$$

with probability at least $1 - 1/(d_1+d_2)^3$. Since $\hat{\Theta}$ is the minimizer for the objective function 9, we have,

$$-\mathcal{L}(\hat{\Theta}) + \lambda\left|||\hat{\Theta}|||_{\text{L-nuc}} \le -\mathcal{L}(\Theta^*) + \lambda|||\Theta^*|||_{\text{L-nuc}},$$

which in turn gives us,

$$\begin{aligned}\frac{\psi(2\alpha)}{6d_1}|||\Delta|||_{\text{L}}^2 &\le -\mathcal{L}(\hat{\Theta}) + \mathcal{L}(\Theta^*) + \langle\!\langle\nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle \qquad (65)\\ &\le \lambda\left(|||\Theta^*|||_{\text{L-nuc}} - \left|||\hat{\Theta}|||_{\text{L-nuc}}\right) + \langle\!\langle\nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle\\ &\le \lambda\left(|||\Delta|||_{\text{L-nuc}}\right) + \langle\!\langle\nabla\mathcal{L}(\Theta^*)L^{-1/2}, \Delta L^{1/2}\rangle\!\rangle\\ &\le \lambda\left(|||\Delta|||_{\text{L-nuc}}\right) + \left\|\nabla\mathcal{L}(\Theta^*)L^{-1/2}\right\|_2|||\Delta|||_{\text{L-nuc}}, \qquad (66)\end{aligned}$$

where last two inequalities follow from the triangle inequality for nuclear norm and generalized Hölder's inequality. Now we put $\lambda = 2\sqrt{32}R$ and use Lemma 21 to get,

$$|||\Delta|||_{\text{L}}^2 \le \frac{6d_1}{\psi(2\alpha)}\left(\lambda + \frac{\lambda}{2}\right)|||\Delta|||_{\text{L-nuc}} \le \frac{9d_1\lambda}{\psi(2\alpha)}|||\Delta|||_{\text{L-nuc}}, \tag{67}$$

with probability at least $1 - 1/(d_1+d_2)^3$. Combining Case 1 and 2 we get,

$$|||\Delta|||_{\text{L}}^2 \le 9\left(\alpha + \frac{1}{\psi(2\alpha)}\right)d_1\lambda|||\Delta|||_{\text{L-nuc}}$$

Lemma 22 *If* $\lambda \ge 2|||\nabla\mathcal{L}(\Theta^*)|||_2$, *then we have*

$$|||\Delta|||_{\text{L-nuc}} \le 4\sqrt{2r}|||\Delta|||_{\text{L}} + 4\sum_{j=r+1}^{\min\{d_1, d_2-G\}}\sigma_j(\Theta^* L^{1/2}), \tag{68}$$

for all $r \in [\min\{d_1, d_2 - G\}]$. *(Proof in Section A.5)*

Finally, utilizing the above lemma, we get,

$$\frac{1}{d_1}|||\Delta|||_{\text{L}}^2 \le 36\lambda\left(\alpha + \frac{1}{\psi(2\alpha)}\right)\left(\sqrt{2r}|||\Delta|||_{\text{L}} + \sum_{j=r+1}^{\min\{d_1, d_2-G\}}\sigma_j(\Theta^* L^{1/2})\right)$$

A.1 Proof of Lemma 20

$$\begin{aligned}&\mathbb{P}\left\{\frac{1}{n}\sum_{i=1}^{n}\left(\langle\!\langle\Theta, X^{(i)}\rangle\!\rangle\right)^2 \ge \frac{1}{3d_1}|||\Theta|||_{\text{L}}^2,\ \forall\ \Theta \in \mathcal{A}\right\}\\ &= 1 - \mathbb{P}\left\{\exists\ \Theta \in \mathcal{A}, \text{ such that } \frac{1}{n}\sum_{i=1}^{n}\left(\langle\!\langle\Theta, X^{(i)}\rangle\!\rangle\right)^2 < \frac{1}{3d_1}|||\Theta|||_{\text{L}}^2\right\} \qquad (69)\end{aligned}$$

When $\Theta \in \mathcal{A}$,

$$\|\|\Theta\|\|_{\mathrm{L}}^2 \geq 16\alpha d_1 R \|\|\Theta\|\|_{\text{L-nuc}} \geq 16\alpha d_1 R \|\|\Theta\|\|_{\mathrm{L}} \implies \|\|\Theta\|\|_{\mathrm{L}} \geq 16\alpha d_1 R := \mu\,, \tag{70}$$

where the second inequality follows from $\|\|\Theta\|\|_{\text{L-nuc}} = \left\|\left|\Theta\, L^{1/2}\right|\right\|_{\text{nuc}} \geq \left\|\left|\Theta\, L^{1/2}\right|\right\|_{\mathrm{F}} = \|\|\Theta\|\|_{\mathrm{L}}$.

Lemma 23 *Let* $\mathcal{B}(D) := \left\{\Theta \in \mathbb{R}^{d_1 \times d_2} | \|\|\Theta\|\|_{\infty} \leq \alpha,\ \|\|\Theta\|\|_{\mathrm{L}} \leq D,\ \|\|\Theta\|\|_{\text{L-nuc}} \leq \frac{D^2}{16\alpha d_1 R}\right\}$, *and,* $Z_D := \sup_{\Theta \in \mathcal{B}(D)} \left(-\frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 + \frac{2}{d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right)$, *then,*

$$\mathbb{P}\left\{Z_D \geq \frac{3}{2d_1}D^2\right\} \leq \exp\left(-\frac{nD^4}{32\alpha^4 d_1^2}\right). \tag{71}$$

Above lemma is proved in Section A.2. Let $\beta = \sqrt{\frac{10}{9}}$, then the sets,

$$\mathcal{S}_\ell = \left\{\Theta \in \mathbb{R}^{d_1 \times d_2} | \|\|\Theta\|\|_{\infty} \leq \alpha, \beta^{\ell-1}\mu \leq \|\|\Theta\|\|_{\mathrm{L}} \leq \beta^\ell \mu,\ \|\|\Theta\|\|_{\text{L-nuc}} \leq \frac{(\beta^\ell \mu)^2}{16\alpha d_1 R}\right\},\ \ell = 1, 2, 3, \ldots, \tag{72}$$

cover the set $\mathcal{A}$, that is $\mathcal{A} \subset \cup_{\ell=1}^{\infty} \mathcal{S}_\ell$ and $\mathcal{S}_\ell \subseteq \mathcal{B}(\beta^\ell \mu)$. This gives us,

$$\begin{aligned}
&\mathbb{P}\left\{\exists\, \Theta \in \mathcal{A} \text{ s.t.} \frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 < \frac{1}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right\} \\
&\leq \sum_{\ell=1}^{\infty} \mathbb{P}\left\{\exists\, \Theta \in \mathcal{S}_\ell \text{ s.t. } \frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 < \frac{1}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right\} \\
&\leq \sum_{\ell=1}^{\infty} \mathbb{P}\left\{\exists\, \Theta \in \mathcal{B}(\beta^\ell \mu) \text{ s.t. } \frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 < \frac{1}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right\}
\end{aligned} \tag{73}$$

If there exists a $\Theta \in \mathcal{B}(\beta^\ell \mu)$ such that $\frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 < \frac{1}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2$ then,

$$Z_{\beta^\ell \mu} \geq -\frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 + \frac{2}{d_1}\|\|\Theta\|\|_{\mathrm{L}}^2 > \frac{5}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2 \geq \frac{5}{3d_1}\beta^{2\ell-2}\mu^2 = \frac{3}{2d_1}(\beta^\ell \mu)^2,$$

which gives us,

$$\begin{aligned}
\mathbb{P}\left\{\exists\, \Theta \in \mathcal{A} \ni \frac{1}{n}\sum_{i=1}^n \left(\langle\!\langle \Theta, X^{(i)} \rangle\!\rangle\right)^2 < \frac{1}{3d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right\} &\leq \sum_{\ell=1}^{\infty} \mathbb{P}\left\{Z_{\beta^\ell \mu} > \frac{3}{2d_1}(\beta^\ell \mu)^2\right\} \\
&\overset{(a)}{\leq} \sum_{\ell=1}^{\infty} \exp\left(-\frac{n(\beta^\ell \mu)^4}{32\alpha^4 d_1^2}\right) \\
&\overset{(b)}{\leq} \sum_{\ell=1}^{\infty} \exp\left(-\frac{4\ell(\beta-1)n\mu^4}{32\alpha^4 d_1^2}\right) \\
&\overset{(c)}{\leq} 2\exp\left(-\frac{4(\beta-1)n\mu^4}{32\alpha^4 d_1^2}\right)
\end{aligned} \tag{74}$$

where (a) is from Lemma 23, (b) is true since $\beta^{4\ell} \geq 4\ell(\beta-1)$ when $\beta \geq 1$ and (c) is obtained by summing the geometric series, with common ratio less than $1/2$, in previous inequality. Finally if we assume that $n \leq 2^6 d_1^2 \sigma^2 \log(2d)$ and $n \leq 2^2 (d_1 \sigma_{\min}(L)^{-1})^{2/3} \log(2d)$, then we have $2^2 \log(2d) \leq 4(\beta-1)n\mu^4/32\alpha^4 d_1^2$ as follows

$$\begin{aligned}
\frac{4(\beta-1)n\mu^4}{32\alpha^4 d_1^2} = \frac{4(\beta-1)n(16\alpha d_1 R)^4}{32\alpha^4 d_1^2} &= 2^{13}(\beta-1)d_1^2 \max\left\{\frac{\sigma^2 \log^2(2d)}{n}, \frac{\sigma_{\min}(L)^{-2}\log^4(2d)}{n^3}\right\} \\
&\geq 2^2 \log(2d)
\end{aligned} \tag{75}$$

A.2 Proof of Lemma 23

Notice that the $\frac{2}{d_1}\|\|\Theta\|\|_{\mathrm{L}}^2$ is the mean of $\frac{1}{n}\sum_{i=1}^n \langle\!\langle \Theta, X^{(i)} \rangle\!\rangle^2$,

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n \langle\!\langle \Theta, X^{(i)} \rangle\!\rangle^2\right] &= \frac{1}{d_1}\sum_{j \in [d1]}\sum_{k,l \in [d2]} (\Theta_{j,k} - \Theta_{j,l})^2 P_{k,l} \\
&= \frac{2}{d_1}\sum_j \sum_k \Theta_{j,k}^2 \sum_l P_{k,l} - 2\sum_{k,l} \Theta_{j,k}\Theta_{j,l} P_{k,l} \\
&\overset{(a)}{=} \frac{2}{d_1}\sum_j \langle\!\langle \Theta_j \Theta_j^T, \mathrm{diag}(P_k) \rangle\!\rangle - 2\langle\!\langle \Theta_j \Theta_j^T, P \rangle\!\rangle \\
&= \frac{2}{d_1}\sum_j \langle\!\langle \Theta_j \Theta_j^T, L \rangle\!\rangle = \frac{2}{d_1}\left\|\left|\Theta L^{1/2}\right|\right\|_{\mathrm{F}}^2
\end{aligned}$$

where, in (a) $P_k = \sum_{l \in [d2]} P_{k,l}$ and Θ_j is the j-th row of Θ. Therefore we use the following standard technique to get a handle on supremum of deviation from mean.

First, we use bounded differences property to prove that Z_D concentrates around its mean. We write $Z_D(X^{(1)}, \ldots, X^{(n)})$ to represent Z_D as a function of n independent random variables. Now, let $X^{(i)}$ and $\tilde{X}^{(i)}$ be two realization of the i-th ($1 \leq i \leq n$) random parameter of Z_D, then,

$$\begin{aligned}
&\left|Z_D(X^{(1)}, \ldots, X^{(i)}, \ldots, X^{(n)}) - Z_D(X^{(1)}, \ldots, \tilde{X}^{(i)}, \ldots, X^{(n)})\right| \\
&= \left|\sup_{\Theta \in \mathcal{B}(D)} \left(-\frac{1}{n}\sum_{i=1}^n \langle\!\langle \Theta, X^{(i)} \rangle\!\rangle^2 + \frac{2}{d_1}\|\|\Theta\|\|_{\mathrm{L}}^2\right) - \right. \\
&\quad \left. \sup_{\Theta' \in \mathcal{B}(D)} \left(-\frac{1}{n}\left(\sum_{i=1, i \neq i'}^n \langle\!\langle \Theta', X^{(i)} \rangle\!\rangle^2 + \langle\!\langle \Theta', \tilde{X}^{(i')} \rangle\!\rangle^2\right) + \frac{2}{d_1}\|\|\Theta'\|\|_{\mathrm{L}}^2\right)\right|
\end{aligned} \tag{76}$$

Now WLOG assume that $Z_D(X^{(1)},\dots,X^{(i)},\dots,X^{(n)}) \geq Z_D(X^{(1)},\dots,\tilde{X}^{(i)},\dots,X^{(n)})$ and the first supremum is achieved at $\tilde{\Theta}$, which gives us.

$$
\begin{aligned}
&= \sup_{\Theta\in\mathcal{B}(D)} \left(-\frac{1}{n}\sum_{i=1}^{n}\langle\!\langle \Theta, X^{(i)}\rangle\!\rangle^2 + \frac{2}{d_1}|||\Theta|||_{\mathrm{L}}^2\right) - \\
&\sup_{\Theta'\in\mathcal{B}(D)} \left(-\frac{1}{n}\left(\sum_{\substack{i=1\\ i\neq i'}}^{n}\langle\!\langle \Theta', X^{(i)}\rangle\!\rangle^2 + \langle\!\langle \Theta', \tilde{X}^{(i')}\rangle\!\rangle^2\right) + \frac{2}{d_1}|||\Theta'|||_{\mathrm{L}}^2\right) \\
&\leq \left(-\frac{1}{n}\sum_{i=1}^{n}\langle\!\langle \bar{\Theta}, X^{(i)}\rangle\!\rangle^2 + \frac{2}{d_1}|||\bar{\Theta}|||_{\mathrm{L}}^2\right) - \left(-\frac{1}{n}\left(\sum_{\substack{i=1\\ i\neq i'}}^{n}\langle\!\langle \bar{\Theta}, X^{(i)}\rangle\!\rangle^2 + \langle\!\langle \bar{\Theta}, \tilde{X}^{(i')}\rangle\!\rangle^2\right) + \frac{2}{d_1}|||\bar{\Theta}|||_{\mathrm{L}}^2\right) \\
&\leq \sup_{\Theta\in\mathcal{B}(D)} \frac{1}{n}\left|\langle\!\langle \Theta, X^{(i)}\rangle\!\rangle^2 - \langle\!\langle \Theta, \tilde{X}^{(i)}\rangle\!\rangle^2\right| \\
&\leq \frac{4\alpha^2}{n}, \qquad (77)
\end{aligned}
$$

where the last inequality is true since, $|||\Theta|||_\infty \leq \alpha$ for any $\Theta \in \mathcal{B}(D) \subseteq \Omega_\alpha$. Now we upper bound $\mathbb{E}[Z_D]$ as follows,

$$
\begin{aligned}
\mathbb{E}[Z_D] &\overset{(a)}{\leq} 2\mathbb{E}\left[\sup_{\Theta\in\mathcal{B}(D)} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\langle\!\langle \Theta, X^{(i)}\rangle\!\rangle^2\right] \\
&\overset{(b)}{\leq} 4\alpha\mathbb{E}\left[\sup_{\Theta\in\mathcal{B}(D)} \frac{1}{n}\sum_{i=1}^{n}\varepsilon_i\langle\!\langle \Theta L^{1/2}, X^{(i)}L^{-1/2}\rangle\!\rangle\right] \\
&\leq 4\alpha\mathbb{E}\left[\sup_{\Theta\in\mathcal{B}(D)} |||\Theta|||_{\text{L-nuc}}\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-1/2}\right\|_2\right] \\
&\leq 4\alpha \sup_{\Theta\in\mathcal{B}(D)} |||\Theta|||_{\text{L-nuc}}\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-\text{`}1/2}\right\|_2\right],
\end{aligned}
$$

where (a) is standard symmetrization argument using i.i.d. Rademacher variables $\{\varepsilon_i\}_{i=1}^n$ and since $|\langle\!\langle \Theta, X^{(i)}\rangle\!\rangle| \leq 2\alpha$ we use Ledoux-Talagrand contraction inequality Ledoux and Talagrand (2013) to obtain (b).

Lemma 24 *Let* $R = \max\left\{\sqrt{\frac{\sigma\log(2d)}{n}}, \frac{\sigma_{\min}(L)^{-1/2}\log(2d)}{n}\right\}$. *For* $\{X^{(i)}\}_{i=1}^n$ *as defined in the graph sampling and for a binary random variable* ε_i *such that* $\mathbb{E}\left[\varepsilon_i|X^{(i)}\right] = 0$ *and* $|\varepsilon_i| \leq 1$, *we have,*

$$
\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-1/2}\right\|_2 \geq \sqrt{32}R\right\} \leq \frac{1}{(d_1+d_2)^3}, \quad \textit{and,} \quad \mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-1/2}\right\|_2\right] \leq 4R. \qquad (78)
$$

Proof of the lemma is in Section A.3. Now using Lemma 24 we have $\mathbb{E}[Z_D]$ $\leq 16R\alpha\sup_{\Theta\in\mathcal{B}(D)}|||\Theta|||_{\text{L-nuc}} \leq \frac{D^2}{d_1}$. Then using the bounded differences property and the upper bound on the mean, we get the McDiarmid's concentration,

$$
\begin{aligned}
\mathbb{P}\left\{Z_D - D^2/d_1 \geq t\right\} &\leq \mathbb{P}\left\{Z_D - \mathbb{E}[Z_D] \geq t\right\} \\
&\leq \exp\left(-\frac{nt^2}{8\alpha^4}\right) \qquad (79)
\end{aligned}
$$

and putting $t = D^2/2d_1$ gives us the theorem.

A.3 Proof of Lemma 24

Let $W_i := \frac{1}{n}\varepsilon_i X^{(i)}L^{-1/2} = \frac{1}{n}\varepsilon_i e_{j(i)}\left(e_{k(i)} - e_{l(i)}\right)^T L^{-1/2}$ and pseudo-inverse of L be $L^\dagger$, then, $\|W_i\|_2 \leq \sigma_{\min}(L)^{-1/2}\sqrt{2}/n$,

$$
\begin{aligned}
\mathbb{E}\left[W_i W_i^T\right] &= \mathbb{E}\left[\frac{1}{n^2}e_{j(i)}\left(e_{k(i)} - e_{l(i)}\right)^T L^{-1/2}L^{-1/2}\left(e_{k(i)} - e_{l(i)}\right)e_{j(i)}^T\right] \\
&= \mathbb{E}\left[\frac{1}{n^2}e_{j(i)}e_{j(i)}^T\right]\mathbb{E}\left[\left(e_{k(i)} - e_{l(i)}\right)^T L^\dagger\left(e_{k(i)} - e_{l(i)}\right)\right] \\
&= \frac{1}{n^2 d_1}\mathbf{I}_{d_1\times d_1} \times 2\left(\mathbb{E}\left[e_{k(i)}^T L^\dagger e_{k(i)}\right] - \mathbb{E}\left[e_{k(i)}^T L^\dagger e_{l(i)}\right]\right) \\
&= \frac{2}{n^2 d_1}\left(\sum_{u\in[d_1]} P_u L^\dagger_{u,u} - \sum_{u,v\in[d_1]} P_{u,v}L^\dagger_{u,v}\right)\mathbf{I}_{d_1\times d_1} \\
&= \frac{2}{n^2 d_1}\langle\!\langle L, L^\dagger\rangle\!\rangle\mathbf{I}_{d_1\times d_1} \\
&= \frac{2(d_2 - G)}{n^2 d_1}\mathbf{I}_{d_1\times d_1}, \qquad (80)
\end{aligned}
$$

$$
\begin{aligned}
\mathbb{E}\left[W_i^T W_i\right] &= L^{-1/2}\mathbb{E}\left[\frac{1}{n^2}\left(e_{k(i)} - e_{l(i)}\right)\left(e_{k(i)} - e_{l(i)}\right)^T\right]L^{-1/2} \\
&= \frac{1}{n^2}L^{-1/2}\left(\sum_{u,v=1}^{d_2}\left(e_u - e_v\right)\left(e_u - e_v\right)^T P_{u,v}\right)L^{-1/2} \\
&= \frac{1}{n^2}L^{-1/2}\left(2L\right)L^{-1/2} \\
&= \frac{2}{n^2}(\mathbf{I}_{d_2\times d_2} - \sum_{k\in[G]} g_k g_k^T/\|g_k\|_2^2), \text{ and,} \qquad (81)
\end{aligned}
$$

$$
\max\left\{\left\|\mathbb{E}\left[\sum_{i=1}^{n}W_i W_i^T\right]\right\|_2, \left\|\mathbb{E}\left[\sum_{i=1}^{n}W_i^T W_i\right]\right\|_2\right\} \leq \sum_{i=1}^{n}\max\left\{\left\|\mathbb{E}\left[W_i W_i^T\right]\right\|_2, \left\|\mathbb{E}\left[W_i^T W_i\right]\right\|_2\right\} \qquad (82)
$$

$$
\leq \frac{2}{n}\sigma\,. \qquad (83)
$$

where $\sigma = \max\left\{\frac{d_2-G}{d_1}, 1\right\}$. Now by Matrix Bernstein concentration theorem Tropp (2015) we have,

$$\mathbb{P}\left\{\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-1/2}\right\|_2 \geq t\right\} \leq \exp\left(\frac{-nt^2/2}{2\sigma + \sqrt{2}\sigma_{\min}(L)^{-1/2}t/3}\right), \text{ and,} \tag{84}$$

$$`\mathbb{E}\left[\left\|\frac{1}{n}\sum_{i=1}^{n}\varepsilon_i X^{(i)}L^{-1/2}\right\|_2\right] \leq \sqrt{\frac{4\sigma\log(d_1+d_2)}{n}} + \frac{\sqrt{2\sigma_{\min}(L)^{-1}}}{3n}\log(d_1+d_2)\,. \tag{85}$$

Choosing $t = \max\left\{\sqrt{\frac{24\sigma\log(d_1+d_2)}{n}}, \frac{16\sqrt{2\sigma_{\min}(L)^{-1}}\log(d_1+d_2)}{n}\right\}$ produces the desired result.

A.4 Proof of Lemma 21

The gradient can be written down as,

$$\nabla\mathcal{L}(\Theta^*) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \frac{\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}{1+\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}\right)X^{(i)}. \tag{86}$$

Then Lemma 24 directly gives the result because,

$$\mathbb{E}\left[y_i - \frac{\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}{1+\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}\middle| X^{(i)}\right] = 0 \quad \text{and} \quad \left|y_i - \frac{\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}{1+\exp(\langle\!\langle\Theta^*, X^{(i)}\rangle\!\rangle)}\right| \leq 1.$$

A.5 Proof of Lemma 22

Denote the singular value decomposition of $\Theta^*L^{1/2}$ by $\Theta^*L^{1/2} = U\Sigma V^T$, where $U \in \mathbb{R}^{d_1\times d_1}$ and $V \in \mathbb{R}^{d_2\times d_2}$ are orthogonal matrices. For a given $r \in [\min\{d_1, d_2 - G\}]$, let $U_r = [u_1, \ldots, u_r]$ and $V_r = [v_1, \ldots, v_r]$, where $u_i \in \mathbb{R}^{d_1\times 1}$ and $v_i \in \mathbb{R}^{d_2\times 1}$ are the left and right singular vectors corresponding to the i-th largest singular value, respectively. Define T to be the subspace spanned by all matrices in $\mathbb{R}^{d_1\times d_2}$ of the form U_rA^T or BV_r^T for any $A \in \mathbb{R}^{d_2\times r}$ or $B \in \mathbb{R}^{d_1\times r}$, respectively. The orthogonal projection of any matrix $M \in \mathbb{R}^{d_1\times d_2}$ onto the space T is given by $\mathcal{P}_T(M) = U_rU_r^TM + MV_rV_r^T - U_rU_r^TMV_rV_r^T$. The projection of M onto the complement space $T^\perp$ is $\mathcal{P}_{T^\perp}(M) = (I - U_rU_r^T)M(I - V_rV_r^T)$. The subspace T and the respective projections onto T and $T^\perp$ play crucial a role in the analysis of nuclear norm minimization, since they define the sub-gradient of the nuclear norm at Θ^*. We refer to Candès and Recht (2009) for more detailed treatment of this topic.

Let $\Delta' = \mathcal{P}_T(\Delta L^{1/2})$ and $\Delta'' = \mathcal{P}_{T^\perp}(\Delta L^{1/2})$. Notice that $\mathcal{P}_T(\Theta^*L^{1/2}) = U_r\Sigma_rV_r^T$, where $\Sigma_r \in \mathbb{R}^{r\times r}$ is the diagonal matrix formed by the top r singular values. Since $\mathcal{P}_T(\Theta^*L^{1/2})$ and Δ'' have row and column spaces that are orthogonal, it follows from Lemma 2.3 in Recht et al. (2010) that

$$\left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2}) - \Delta''\right|\right|\right|_{\text{nuc}} = \left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} + \left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}}.$$

Hence, in view of the triangle inequality,

$$\begin{aligned}
\left|\left|\left|\widehat{\Theta}L^{1/2}\right|\right|\right|_{\text{nuc}} &= \left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2}) + \mathcal{P}_{T^\perp}(\Theta^*L^{1/2}) - \Delta' - \Delta''\right|\right|\right|_{\text{nuc}} \\
&\geq \left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2}) - \Delta''\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2}) - \Delta'\right|\right|\right|_{\text{nuc}} \\
&= \left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} + \left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2}) - \Delta'\right|\right|\right|_{\text{nuc}} \\
&\geq \left|\left|\left|\mathcal{P}_T(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} + \left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}} \\
&= \left|\left|\left|\Theta^*L^{1/2}\right|\right|\right|_{\text{nuc}} + \left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}} - 2\left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}}.
\end{aligned} \tag{87}$$

Because $\widehat{\Theta}$ is an optimal solution, we have

$$\begin{aligned}
\lambda\left(\left|\left|\left|\widehat{\Theta}L^{1/2}\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\Theta^*L^{1/2}\right|\right|\right|_{\text{nuc}}\right) &\leq -\mathcal{L}(\Theta^*) + \mathcal{L}(\widehat{\Theta}) \\
&\overset{(a)}{\leq} \langle\!\langle\Delta L^{1/2}, \nabla\mathcal{L}(\Theta^*)L^{-1/2}\rangle\!\rangle \\
&\overset{(b)}{\leq} \left|\left|\left|\Delta\right|\right|\right|_{\text{L-nuc}}\left\|\nabla\mathcal{L}(\Theta^*)L^{-1/2}\right\|_2 \leq \frac{\lambda}{2}\left|\left|\left|\Delta\right|\right|\right|_{\text{L-nuc}},
\end{aligned} \tag{88}$$

where (a) holds due to the convexity of $-\mathcal{L}$; (b) follows from the Cauchy-Schwarz inequality; the last inequality holds due to the assumption that $\lambda \geq 2\|\nabla\mathcal{L}(\Theta^*)\|_2$. Combining (87) and (88) yields

$$2\left(\left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}} - 2\left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} - \left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}}\right) \leq \left|\left|\left|\Delta\right|\right|\right|_{\text{L-nuc}} \leq \left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}} + \left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}}.$$

Thus $\left|\left|\left|\Delta''\right|\right|\right|_{\text{nuc}} \leq 3\left|\left|\left|\Delta'\right|\right|\right|_{\text{L-nuc}} + 4\left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}}$. By triangle inequality,

$$\left|\left|\left|\Delta\right|\right|\right|_{\text{L-nuc}} \leq 4\left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}} + 4\left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}}.$$

Notice that $\Delta' = U_rU_r^T\Delta L^{1/2} + (I - U_rU_r^T)\Delta L^{1/2}V_rV_r^T$. Both $U_rU_r^T\Delta L^{1/2}$ and $(I - U_rU_r^T)\Delta L^{1/2}V_rV_r^T$ have rank at most r. Thus Δ' has rank at most $2r$. Hence, $\left|\left|\left|\Delta'\right|\right|\right|_{\text{nuc}} \leq \sqrt{2r}\left|\left|\left|\Delta'\right|\right|\right|_{\text{F}} \leq \sqrt{2r}\left|\left|\left|\Delta L^{1/2}\right|\right|\right|_{\text{F}} \leq \sqrt{2r}\left|\left|\left|\Delta\right|\right|\right|_{\text{L}}$. Then the theorem follows because $\left|\left|\left|\mathcal{P}_{T^\perp}(\Theta^*L^{1/2})\right|\right|\right|_{\text{nuc}} = \sum_{j=r+1}^{\min\{d_1,d_2\}}\sigma_j(\Theta^*L^{1/2})$.

Appendix B. Proof of the Information-theoretic Graph Sampling Lower Bound, Theorem 4

The proof uses Fano Inequality based packing set argument to get an lower bound on the error of any (measurable) estimator. We will construct a packing set in Ω_α with a minimum distance of δ between any pair of elements in the packing.

Let $\{\Theta^{(1)}, \Theta^{(2)}, \ldots, \Theta^{(M)}\}$ be a set of M matrices within the set Ω_α, satisfying $\left|\left|\left|\Theta^{(\ell_1)} - \Theta^{(\ell_1)}\right|\right|\right|_{\text{L}} \geq \delta$ for all $\ell_1, \ell_2 \in [M]$. Now, $\Theta^{(N)}$ is uniformly drawn from this set and then the comparison results (according to MNL model) of n randomly chosen pairs of items,

each drawn according to the probability matrix P and each compared by uniformly chosen user. Let $\widehat{N}$ be the best estimator of N from the observations. Then we can show that,

$$\sup_{\Theta^* \in \Omega_\alpha} \mathbb{P}\left\{ \left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{L}}^2 \geq \frac{\delta^2}{2} \right\} \quad \geq \quad \mathbb{P}\left\{ \widehat{N} \neq N \right\}, \tag{89}$$

Now we have converted the problem of finding the minimum estimation error, into finding the minimum probability error of a M-ary hypothesis testing problem. If we can prove that the above RHS is lower bounded by $1/2$, we are done.

The generalized Fanos inequality along with data processing inequality gives us,

$$\begin{aligned} \mathbb{P}\left\{ \widehat{N} \neq N \right\} &\geq 1 - \frac{\mathbb{E}[\, I(\widehat{N}; N)\,] + \log 2}{\log M} && (90) \\ &\geq 1 - \frac{\binom{M}{2}^{-1} \sum_{\ell_1, \ell_2 \in [M]} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) + \log 2}{\log M}, && (91) \end{aligned}$$

where $D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)})$ denotes the *expected* Kullback-Leibler divergence between the probability distributions of the comparison results of the observed nd_1 pairs, for $N = \ell_1$ and $N = \ell_2$. The expectation is taken over different choices for the selected pairs for comparison.

$$\begin{aligned} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) = \frac{n}{d_1} \sum_{\substack{i \in [d_1] \\ \{j,j'\} \subset [d_2]}} 2P_{j,j'} \Bigg[& \frac{e^{\Theta_{ij}^{(\ell_1)}}}{e^{\Theta_{ij}^{(\ell_1)}} + e^{\Theta_{ij'}^{(\ell_1)}}} \log \left(\frac{e^{\Theta_{ij}^{(\ell_1)}} \Big/ \left(e^{\Theta_{ij}^{(\ell_1)}} + e^{\Theta_{ij'}^{(\ell_1)}} \right)}{e^{\Theta_{ij}^{(\ell_2)}} \Big/ \left(e^{\Theta_{ij}^{(\ell_2)}} + e^{\Theta_{ij'}^{(\ell_2)}} \right)} \right) && (92) \\ & + \frac{e^{\Theta_{ij'}^{(\ell_1)}}}{e^{\Theta_{ij}^{(\ell_1)}} + e^{\Theta_{ij'}^{(\ell_1)}}} \log \left(\frac{e^{\Theta_{ij'}^{(\ell_1)}} \Big/ \left(e^{\Theta_{ij}^{(\ell_1)}} + e^{\Theta_{ij'}^{(\ell_1)}} \right)}{e^{\Theta_{ij'}^{(\ell_2)}} \Big/ \left(e^{\Theta_{ij}^{(\ell_2)}} + e^{\Theta_{ij'}^{(\ell_2)}} \right)} \right) \Bigg] && (93) \end{aligned}$$

where n is the number of pairs of items selected and compared by one random user each, $P_{j,j'}$ is half the probability with which item pair $\{j, j'\}$ is selected and the observation probabilities come from the standard MNL model. Let $x_{ijj'} \equiv e^{\Theta_{ij'}^{(\ell_1)}} / (e^{\Theta_{ij}^{(\ell_1)}} + e^{\Theta_{ij'}^{(\ell_1)}})$ and $y_{ijj'} \equiv e^{\Theta_{ij'}^{(\ell_2)}} / (e^{\Theta_{ij}^{(\ell_2)}} + e^{\Theta_{ij'}^{(\ell_2)}})$.

$$\begin{aligned} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) &\overset{(a)}{=} n \sum_{i \in [d_1]} \frac{1}{d_1} \sum_{\{j,j'\} \subset [d_2]} 2P_{j,j'} \left[x_{ijj'} \log \frac{x_{ijj'}}{y_{ijj'}} + (1 - x_{ijj'}) \log \frac{1 - x_{ijj'}}{1 - y_{ijj'}} \right] && (94) \\ &\overset{(a)}{\leq} n \sum_{i \in [d_1]} \frac{1}{d_1} \sum_{\{j,j'\} \subset [d_2]} 2P_{j,j'} \left[x_{ijj'} \frac{x_{ijj'} - y_{ijj'}}{y_{ijj'}} + (1 - x_{ijj'}) \frac{y_{ijj'} - x_{ijj'}}{1 - y_{ijj'}} \right] && (95) \\ &= 2n \sum_{i \in [d_1]} \frac{1}{d_1} \sum_{\{j,j'\} \subset [d_2]} \frac{(x_{ijj'} - y_{ijj'}) P_{j,j'} (x_{ijj'} - y_{ijj'})}{y_{ijj'}(1 - y_{ijj'})} && (96) \\ &\overset{(b)}{\leq} 8ne^{2\alpha} \sum_{i \in [d_1]} \frac{1}{d_1} \sum_{\{j,j'\} \subset [d_2]} (x_{ijj'} - y_{ijj'}) P_{j,j'} (x_{ijj'} - y_{ijj'}), && (97) \end{aligned}$$

where (a) is due to the fact that $\log(x/y) \leq (x - y)/y$ for $x/y \geq 0$ and (b) is true because $|\Theta_{ij}^{(\ell_2)}| \leq \alpha$ implies, $y_{ijj'} = e^{\Theta_{ij}^{(\ell_2)}} / (e^{\Theta_{ij}^{(\ell_2)}} + e^{\Theta_{ij'}^{(\ell_2)}}) \geq e^{-2\alpha}/2$ which in turn implies, $y_{ijj'}(1 - y_{ijj'}) \geq e^{-2\alpha}(2 - e^{-2\alpha})/4 \geq e^{-2\alpha}/4$. Let $f(z) = 1/(1 + e^{-z})$, a 1-Lipschitz function, it can be seen that $(x_{ijj'} - y_{ijj'})^2 = (f(\Theta_{ij}^{(\ell_1)} - \Theta_{ij'}^{(\ell_1)}) - f(\Theta_{ij}^{(\ell_2)} - \Theta_{ij'}^{(\ell_2)}))^2 \leq ((\Theta_{ij}^{(\ell_1)} - \Theta_{ij'}^{(\ell_1)}) - (\Theta_{ij}^{(\ell_2)} - \Theta_{ij'}^{(\ell_2)}))^2$. This gives us,

$$\begin{aligned} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) &\leq \frac{8ne^{2\alpha}}{d_1} \sum_{i \in [d_1]} \sum_{\{j,j'\} \subset [d_2]} P_{u,v} ((\Theta_{ij}^{(\ell_1)} - \Theta_{ij}^{(\ell_2)}) - (\Theta_{ij'}^{(\ell_1)} - \Theta_{ij'}^{(\ell_2)}))^2, && (98) \\ &\overset{(a)}{\leq} \frac{8ne^{2\alpha}}{d_1} \sum_{i \in [d_1]} (\Theta^{(\ell_1)} - \Theta^{(\ell_2)})_i L (\Theta^{(\ell_1)} - \Theta^{(\ell_2)})_i, && (99) \\ &= \frac{8ne^{2\alpha}}{d_1} \sum_{i \in [d_1]} (\Theta^{(\ell_1)} - \Theta^{(\ell_2)})_i L (\Theta^{(\ell_1)} - \Theta^{(\ell_2)})_i, && (100) \\ &= \frac{8ne^{2\alpha}}{d_1} \left|\!\left|\!\left| (\Theta^{(\ell_1)} - \Theta^{(\ell_2)}) L^{1/2} \right|\!\right|\!\right|_{\mathrm{F}}^2 && (101) \\ &= \frac{8ne^{2\alpha}}{d_1} \left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{L}}^2 && (102) \\ & && (103) \end{aligned}$$

where (a) is due to the fact that $L = \mathrm{diag}(P_u) - P$ is the Laplacian of the probability matrix P, and Θ_i denotes the i-th row of matrix Θ. Combining the above with (91), we get,

$$\mathbb{P}\left\{ \widehat{N} \neq N \right\} \geq 1 - \frac{\binom{M}{2}^{-1} \sum_{\ell_1, \ell_2 \in [M]} (8ne^{2\alpha}/d_1) \left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{L}}^2 + \log 2}{\log M}. \tag{104}$$

The remainder of the proof relies on the following probabilistic packing.

Lemma 25 *For each $r \in \{1, \dots, d_1\}$, and for any positive $\delta > 0$ there exists a family of $d_1 \times d_2$ dimensional matrices $\{\Theta^{(1)}, \dots, \Theta^{(M(\delta))}\}$ with cardinality $M(\delta) = \lfloor \exp(rd_1/256) \rfloor$ such that each matrix is rank r and the following bounds hold:*

$$\begin{aligned} \left|\!\left|\!\left| \Theta^{(\ell)} \right|\!\right|\!\right|_{\mathrm{L}} &\leq \delta \ , \ \textit{for all } \ell \in [M] && (105) \\ \left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{L}} &\geq \delta \ , \ \textit{for all } \ell_1, \ell_2 \in [M] && (106) \\ \Theta^{(\ell)} &\in \Omega_{\tilde{\alpha}} \ , \ \textit{for all } \ell \in [M] \ , && (107) \end{aligned}$$

with $\tilde{\alpha} = (8\delta/d_2)\sqrt{2 \log d}$ for $d = (d_1 + d_2)/2$.

Now if we assume $\delta \leq \alpha d_2 / 8\sqrt{2 \log d}$, we get $\Theta^{(\ell)} \in \Omega_\alpha$ for $\ell \in [M]$. The above lemma also implies that $\left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{F}}^2 \leq 4\delta^2$ which implies,

$$\mathbb{P}\left\{ \widehat{N} \neq N \right\} \geq 1 - \frac{32ne^{2\alpha}\delta^2/d_1 + \log 2}{rd_1/256} \geq \frac{1}{2} \ , \tag{108}$$

where the last inequality holds when $\delta \leq (e^{-\alpha}/128)\sqrt{rd_1^2/n}$. Along with (89), this proves that,

$$\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{L}} \Big] \geq \frac{\delta}{2} \ , \tag{109}$$

for all $\delta \leq \min\{\alpha d_2 / 8\sqrt{2 \log d}, (e^{-\alpha}/128)\sqrt{rd_1^2/n}\}$.

Similarly using the following lemma, Lemma 26, we can prove that,

$$\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\left|\!\left|\!\left| \widehat{\Theta} - \Theta^* \right|\!\right|\!\right|_{\mathrm{L}} \Big] \geq \frac{\delta}{2} \ , \tag{110}$$

for all $\delta \leq \min\{\alpha\sqrt{rd_1}/\mathrm{tr}\left(\sqrt{(L_r)^\dagger}\right), (e^{-\alpha}/128)\sqrt{rd_1^2/n}\}$.

Lemma 26 *For each $r \in \{1, \dots, d_1\}$, and for any positive $\delta > 0$ there exists a family of $d_1 \times d_2$ dimensional matrices $\{\Theta^{(1)}, \dots, \Theta^{(M(\delta))}\}$ with cardinality $M(\delta) = \lfloor \exp(rd_1/256) \rfloor$ such that each matrix is rank r and the following bounds hold:*

$$\begin{aligned} \left|\!\left|\!\left| \Theta^{(\ell)} \right|\!\right|\!\right|_{\mathrm{L}} &\leq \delta \ , \ \textit{for all } \ell \in [M] && (111) \\ \left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{L}} &\geq \delta \ , \ \textit{for all } \ell_1, \ell_2 \in [M] && (112) \\ \Theta^{(\ell)} &\in \Omega_{\tilde{\alpha}} \ , \ \textit{for all } \ell \in [M] \ , && (113) \end{aligned}$$

with $\tilde{\alpha} = \delta\sqrt{\mathrm{tr}\left((L_r)^\dagger\right)}/\sqrt{rd_1}$, where L_r is the (best) rank r approximation of L.

Now combining (109) and (110), and maximizing the RHS proves the theorem.

B.1 Proof of Lemma 25

Following the construction in Negahban and Wainwright (2012), we use probabilistic method to prove the existence of the desired family. We will show that the following procedure succeeds in producing the desired family with probability at least half, which proves its existence. Let $d = (d_1 + d_2)/2$, and suppose $d_2 \geq d_1$ without loss of generality. For the choice of $M' = e^{rd_2/576}$, and for each $\ell \in [M']$, generate a rank-r matrix $\Theta^{(\ell)} \in \mathbb{R}^{d_1 \times d_2}$ as follows:

$$\Theta^{(\ell)} \ = \ \frac{\delta}{\sqrt{rd_2}} U (V^{(\ell)})^T \Big(\mathbf{I}_{d_2 \times d_2} - \sum_{i \in [G]} \frac{g_i g_i^T}{g_i^T g_i} \Big) \ , \tag{114}$$

where $U \in \mathbb{R}^{d_1 \times r}$ is a random orthogonal basis such that $U^T U = \mathbf{I}_{r \times r}$ and $V^{(\ell)} \in \mathbb{R}^{d_2 \times r}$ is a random matrix with each entry $V_{ij}^{(\ell)} \in \{-1, +1\}$ chosen independently and uniformly at random. By construction, notice that,

$$\left|\!\left|\!\left| \Theta^{(\ell)} \right|\!\right|\!\right|_{\mathrm{L}}^2 = \frac{\delta^2}{r\, d_2} \left|\!\left|\!\left| (V^{(\ell)})^T L^{1/2} \right|\!\right|\!\right|_{\mathrm{F}}^2 = \frac{\delta^2}{r\, d_2} \sum_{i \in [d_2]} (V_i^{(\ell)})^T L V_i^{(\ell)} \leq \frac{\delta^2}{r\, d_2} |\!|\!| L |\!|\!|_2 \sum_{i \in [d_2]} \left|\!\left|\!\left| V_i^{(\ell)} \right|\!\right|\!\right|_{\mathrm{F}}^2 = \delta^2 \ , \tag{115}$$

where $V_i^{(\ell)}$ is the i-th column of $V^{(\ell)}$, since $g_{i i=1}^G$ span the null space of the Laplacian L, $|\!|\!| L |\!|\!|_2 \leq 1$, and $\left|\!\left|\!\left| V^{(\ell)} \right|\!\right|\!\right|_{\mathrm{F}} = \sqrt{rd_2}$. Now, consider $\left|\!\left|\!\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\!\right|\!\right|_{\mathrm{F}}^2 = (\delta^2/(rd_2)) \left|\!\left|\!\left| (V^{(\ell_1)} - V^{(\ell_2)})^T \right|\!\right|\!\right|_{\mathrm{L}}^2 \equiv f(V^{(\ell_1)}, V^{(\ell_2)})$ which is a function over $2rd_2$ i.i.d. random Rademacher variables $V^{(\ell_1)}$ and $V^{(\ell_2)}$ which define $\Theta^{(\ell_1)}$ and $\Theta^{(\ell_2)}$ respectively. Since f is Lipschitz in the following sense, we can apply McDiarmid's concentration inequality. For all $(V^{(\ell_1)}, V^{(\ell_2)})$ and $(\widetilde{V}^{(\ell_1)}, \widetilde{V}^{(\ell_2)})$ that differ in only one variable, say $\widetilde{V}^{(\ell_1)} = V^{(\ell_1)} + 2e_{ij}$, for some standard basis matrix e_{ij}, we have

$$\begin{aligned} & \left| f(V^{(\ell_1)}, V^{(\ell_2)}) - f(\widetilde{V}^{(\ell_1)}, \widetilde{V}^{(\ell_2)}) \right| \ = \\ & \left| \ \frac{\delta^2}{r\, d_2} \left|\!\left|\!\left| (V^{(\ell_1)} - V^{(\ell_2)})^T \right|\!\right|\!\right|_{\mathrm{L}}^2 - \frac{\delta^2}{r\, d_2} \left|\!\left|\!\left| (V^{(\ell_1)} - V^{(\ell_2)} + 2e_{ij})^T \right|\!\right|\!\right|_{\mathrm{L}}^2 \ \right| && (116) \\ & = \ \frac{\delta^2}{r\, d_2} \left| \ 2e_i^T L (V_j^{(\ell_1)} - V_j^{(\ell_2)}) + 4 e_i^T L e_i \ \right| && (117) \\ & = \ \frac{\delta^2}{r\, d_2} (2 |\!|\!| L_i |\!|\!|_1 \left|\!\left|\!\left| V_j^{(\ell_1)} - V_j^{(\ell_2)} \right|\!\right|\!\right|_\infty + 4P_i) && (118) \\ & \leq \ \frac{12\, \delta^2}{r\, d_2} \ , && (119) \end{aligned}$$

where we used the fact that $|\!|\!| L_i |\!|\!|_1 = 2P_i \leq 2$ and $V^{(\ell_1)} - V^{(\ell_2)}$ is entry-wise bounded by 2. The expectation $\mathbb{E}[f(V^{(\ell_1)}, V^{(\ell_2)})]$ is

$$\frac{\delta^2}{r\, d_2} \mathbb{E}\left[\left|\!\left|\!\left| (V^{(\ell_1)} - V^{(\ell_2)})^T \right|\!\right|\!\right|_{\mathrm{L}}^2 \right] = \frac{2\delta^2}{r\, d_2} \mathbb{E}\left[\left|\!\left|\!\left| (V^{(\ell_1)})^T \right|\!\right|\!\right|_{\mathrm{L}}^2 \right] \leq 2\delta^2 \ , \tag{120}$$

where we use equation (115). Applying McDiarmid's inequality with bounded difference $12\delta^2/(rd_2)$, we get that

$$\mathbb{P}\Big\{ \ f(V^{(\ell_1)}, V^{(\ell_2)}) \leq 2\delta^2 - t \ \Big\} \ \leq \ \exp\Big\{ -\frac{t^2\, r\, d_2}{144\, \delta^4} \Big\} \ , \tag{121}$$

Since there are less than $(M')^2$ pairs of (ℓ_1, ℓ_2), setting $t = \delta^2$ and applying the union bound gives

$$\mathbb{P}\left\{ \min_{\ell_1,\ell_2 \in [M']} \left|\left|\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\right|\right|_{\mathrm{F}}^2 \geq \delta^2 \right\} \;\geq\; 1 - \exp\left\{ -\frac{r\, d_2}{144} + 2\log M' \right\} \geq \frac{7}{8}\,, \quad (122)$$

where we used $M' = \exp\{r d_2/576\}$ and $d_2 \geq 607$.

We are left to prove that $\Theta^{(\ell)}$'s are in $\Omega_{(8\delta/d_2)\sqrt{2\log d_2}}$ as defined in (10). Since we removed the mean for each connected component, such that $\Theta^{(\ell)} g_i = 0$, $\forall i \in [G]$ by construction, we only need to show that the maximum entry is bounded by $(8\delta/d_2)\sqrt{2\log d_2}$. We first prove an upper bound in (124) for a fixed $\ell \in [M']$, and use this to show that there exists a large enough subset of matrices satisfying this bound. From (114), consider $(UV^T)_{ij} = \langle\!\langle u_i, v_j \rangle\!\rangle$, where $u_i \in \mathbb{R}^r$ is the first r entries of a random vector drawn uniformly from the d_2-dimensional sphere, and $v_j \in \mathbb{R}^r$ is drawn uniformly at random from $\{-1,+1\}^r$ with $\|v_j\| = \sqrt{r}$. Using Levy's theorem for concentration on the sphere Ledoux (2005), we have

$$\mathbb{P}\left\{ |\langle\!\langle u_i, v_j \rangle\!\rangle| \geq t \right\} \;\leq\; 2\exp\left\{ -\frac{d_2\, t^2}{8\, r} \right\}. \quad (123)$$

Notice that by the definition (114), $\max_{i,j} |\Theta^{(\ell)}_{ij}| \leq (2\delta/\sqrt{r d_2}) \max_{i,j} |\langle\!\langle u_i, v_j \rangle\!\rangle|$. Setting $t = \sqrt{(32r/d_2)\log d_2}$ and taking the union bound over all $d_1 d_2$ indices, we get

$$\mathbb{P}\left\{ \max_{i,j} |\Theta^{(\ell)}_{ij}| \leq \frac{2\delta\sqrt{32\log d_2}}{d_2} \right\} \;\geq\; 1 - 2 d_1 d_2 \exp\left\{ -4\log d_2 \right\} \;\geq\; \frac{1}{2}\,, \quad (124)$$

for a fixed $\ell \in [M']$. Consider the event that there exists a subset $S \subset [M']$ of cardinality $M = (1/4)M'$ with the same bound on maximum entry, then from (124) we get

$$\mathbb{P}\left\{ \exists S \subset [M'] \text{ such that } \left|\left|\left| \Theta^{(\ell)} \right|\right|\right|_\infty \leq \frac{2\delta\sqrt{32\log d_2}}{d_2} \text{ for all } \ell \in S \right\} \geq \sum_{m=M}^{M'} \binom{M'}{m} \left(\frac{1}{2}\right)^m, \quad (125)$$

which is larger than half for our choice of $M < M'/2$.

B.2 Proof of Lemma 26

Inspired from the construction in Negahban and Wainwright (2012), we furnish the following probabilistic argument for the existence of the desired family. For the choice of $M = \lfloor e^{r d_1/256} \rfloor$, and for each $\ell \in [M]$, generate a rank-r matrix $\Theta^{(\ell)} \in \mathbb{R}^{d_1 \times d_2}$ as follows:

$$\Theta^{(\ell)} \;=\; \frac{\delta}{\sqrt{r d_1}} V^{(\ell)} \sqrt{\Lambda_r^\dagger}\, U_r^T\,, \quad (126)$$

where the columns of $U_r \in \mathbb{R}^{d_2 \times r}$ are the top r singular vectors of $L = U\Lambda U^T$, Λ_r is a diagonal matrix in $\mathbb{R}^{r\times r}$ and its diagonal elements are the top r singular values of L corresponding to columns of U_r, $\dagger$ represents the Moore-Penrose pseudo inverse, and $V^{(\ell)}$ is a random matrix with each entry $V^{(\ell)}_{ij} \in \{-1,+1\}$ chosen independently and uniformly at random. First by definition, $\left|\left|\left| \Theta^{(\ell)} \right|\right|\right|_{\mathrm{L}} = (\delta/\sqrt{r d_1}) \left|\left|\left| V^{(\ell)} \right|\right|\right|_{\mathrm{F}} \leq \delta$, since $\left|\left|\left| V^{(\ell)} \right|\right|\right|_{\mathrm{F}} = \sqrt{r d_1}$.

Define f as $f(V^{(\ell_1)}, V^{(\ell_2)}) \equiv \left|\left|\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\right|\right|_{\mathrm{L}}^2 = (\delta^2/(r d_1)) \left|\left|\left| V^{(\ell_1)} - V^{(\ell_2)} \right|\right|\right|_{\mathrm{F}}^2$ which is a function of $2rd_1$ i.i.d. random Rademacher variables. Now we can apply McDiarmid's concentration inequality since f is Lipschitz as folows. For all $(V^{(\ell_1)}, V^{(\ell_2)})$ and $(\widetilde{V}^{(\ell_1)}, \widetilde{V}^{(\ell_2)})$ that differ in only one variable, say $\widetilde{V}^{(\ell_1)} = V^{(\ell_1)} + 2e_{ij}$, for some standard basis matrix e_{ij}, we have

$$\begin{aligned}
&\left| f(V^{(\ell_1)}, V^{(\ell_2)}) - f(\widetilde{V}^{(\ell_1)}, \widetilde{V}^{(\ell_2)}) \right| \\
&= \left| \frac{\delta^2}{r\, d_2} \left|\left|\left| V^{(\ell_1)} - V^{(\ell_2)} \right|\right|\right|_{\mathrm{F}}^2 - \frac{\delta^2}{r\, d_2} \left|\left|\left| V^{(\ell_1)} - V^{(\ell_2)} + 2e_{ij} \right|\right|\right|_{\mathrm{F}}^2 \right| \\
&= \left| \frac{\delta^2}{r\, d_2} \|2e_{ij}\|_{\mathrm{F}}^2 + \frac{\delta^2}{r\, d_2} \langle\!\langle (V^{(\ell_1)} - V^{(\ell_2)}), 2e_{ij} \rangle\!\rangle \right| \\
&\leq \frac{4\,\delta^2}{r\, d_1} + \frac{\delta^2}{r\, d_1} \left|\left|\left| V^{(\ell_1)} - V^{(\ell_2)} \right|\right|\right|_\infty \|2e_{ij}\|_1 \\
&\leq \frac{8\,\delta^2}{r\, d_1}\,, \qquad (127)
\end{aligned}$$

where the penultimate step is true since $(V^{(\ell_1)} - V^{(\ell_2)})$ is entry-wise bounded by 2. The expectation $\mathbb{E}[f(V^{(\ell_1)}, V^{(\ell_2)})]$ is

$$\begin{aligned}
\frac{\delta^2}{r\, d_1} \mathbb{E}\left[\left|\left|\left| (V^{(\ell_1)} - V^{(\ell_2)}) \right|\right|\right|_{\mathrm{F}}^2 \right] &= \frac{2\delta^2}{r\, d_1} \mathbb{E}\left[\left|\left|\left| V^{(\ell_1)} \right|\right|\right|_{\mathrm{F}}^2 \right] \\
&= 2\,\delta^2\,. \qquad (128)
\end{aligned}$$

Now applying McDiarmid's inequality on the function f, we get that

$$\mathbb{P}\left\{ f(V^{(\ell_1)}, V^{(\ell_2)}) \leq 2\delta^2 - t \right\} \;\leq\; \exp\left\{ -\frac{t^2\, r\, d_1}{64\, \delta^4} \right\}, \quad (129)$$

Setting $t = \delta^2$ and applying the union bound gives us,

$$\mathbb{P}\left\{ \min_{\ell_1,\ell_2 \in [M]} \left|\left|\left| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right|\right|\right|_{\mathrm{F}}^2 \geq \delta^2 \right\} \;\geq\; 1 - \exp\left\{ -\frac{r\, d_1}{64} + 2\log M \right\} > 0\,. \quad (130)$$

In the last step, we used $M = \lfloor \exp\{r d_1/\ 256\} \rfloor$. At last we prove that $\Theta^{(\ell)}$'s are in $\Omega_{\delta\sqrt{\mathrm{tr}\left((L_r)^\dagger\right)/r d_1}}$ as defined in (10). Since we know that g_i belongs to the kernel of L for all $i \in [G]$, $\Theta^{(\ell)} g = 0$ by construction (6). From (126), consider $(V\sqrt{\Lambda_r^\dagger} U_r^T)_{ij} = \langle\!\langle v_i, \sqrt{\Lambda_r^\dagger}(u_r)_j \rangle\!\rangle$, where $(u_r)_j \in \mathbb{R}^r$ is the vector of i-th entries of the top r singular vectors of L, and $v_i \in \mathbb{R}^r$ is drawn uniformly at random from $\{-1,+1\}^r$.

$$\left| \langle\!\langle v_i, \sqrt{\Lambda_r^\dagger}(u_r)_j \rangle\!\rangle \right| \leq \|v_i\|_\infty \left\| \sqrt{\Lambda_r^\dagger}(u_r)_j \right\|_1 \leq \sqrt{\mathrm{tr}\left(\Lambda_r^\dagger\right)} = \sqrt{\mathrm{tr}\left((L_r)^\dagger\right)}\,. \quad (131)$$

The above inequality proves that $\left|\left|\left| \Theta^{(\ell)} \right|\right|\right|_\infty$ is upper bounded as desired.

Appendix C. Proof of Theorem 7

We first introduce some additional notations used in the proof. Recall that $\mathcal{L}(\Theta)$ is the log likelihood function. Let $\nabla\mathcal{L}(\Theta) \in \mathbb{R}^{d_1\times d_2}$ denote its gradient such that $\nabla_{ij}\mathcal{L}(\Theta) = \frac{\partial\mathcal{L}(\Theta)}{\partial\Theta_{ij}}$. Let $\nabla^2\mathcal{L}(\Theta) \in \mathbb{R}^{d_1d_2\times d_1d_2}$ denote its Hessian matrix such that $\nabla^2_{ij,i'j'}\mathcal{L}(\Theta) = \frac{\partial^2\mathcal{L}(\Theta)}{\partial\Theta_{ij}\partial\Theta_{i'j'}}$. By the definition of $\mathcal{L}(\Theta)$ in (32), we have

$$\nabla\mathcal{L}(\Theta^*) \;=\; -\frac{1}{k\,d_1}\sum_{i=1}^{d_1}\sum_{\ell=1}^{k} e_i(e_{v_{i,\ell}} - p_{i,\ell})^T\,, \tag{132}$$

where $p_{i,\ell}$ denotes the conditional choice probability at ℓ-th position. Precisely, $p_{i,\ell} = \sum_{j\in S_{i,\ell}} p_{j|(i,\ell)}e_j$ where $p_{j|(i,\ell)}$ is the probability that item j is chosen at ℓ-th position from the top by the user i conditioned on the top $\ell-1$ choices such that $p_{j|(i,\ell)} \equiv \mathbb{P}\{v_{i,\ell} = j|v_{i,1},\ldots,v_{i,\ell-1},S_i\} = e^{\Theta^*_{ij}}/(\sum_{j'\in S_{i,\ell}} e^{\Theta_{ij'}})$ and $S_{i,\ell} \equiv S_i\setminus\{v_{i,1},\ldots,v_{i,\ell-1}\}$, where S_i is the set of alternatives presented to the i-th user and $v_{i,\ell}$ is the item ranked at the ℓ-th position by the user i. Notice that for $i\neq i'$, $\frac{\partial^2\mathcal{L}(\Theta)}{\partial\Theta_{ij}\partial\Theta_{i'j'}} = 0$ and the Hessian is

$$\begin{aligned}\frac{\partial^2\mathcal{L}(\Theta)}{\partial\Theta_{ij}\partial\Theta_{ij'}} &= \frac{1}{k\,d_1}\sum_{\ell=1}^{k}\mathbb{I}\big(j\in S_{i,\ell}\big)\frac{\partial p_{j|(i,\ell)}}{\partial\Theta_{ij'}}\\ &= \frac{1}{k\,d_1}\sum_{\ell=1}^{k}\mathbb{I}\big(j,j'\in S_{i,\ell}\big)\left(p_{j|(i,\ell)}\mathbb{I}(j=j') - p_{j|(i,\ell)}p_{j'|(i,\ell)}\right).\end{aligned} \tag{133}$$

This Hessian matrix is a block-diagonal matrix $\nabla^2\mathcal{L}(\Theta) = \mathrm{diag}(H^{(1)}(\Theta),\ldots,H^{(d_1)}(\Theta))$ with

$$H^{(i)}(\Theta) = \frac{1}{k\,d_1}\sum_{\ell=1}^{k}\left(\mathrm{diag}(p_{i,\ell}) - p_{i,\ell}p_{i,\ell}^T\right). \tag{134}$$

Let $\Delta = \Theta^* - \widehat{\Theta}$ where $\widehat{\Theta}$ is the optimal solution of the convex program in (31). We first introduce three key technical lemmas. The first lemma follows from Lemma 1 of Negahban and Wainwright (2012), and shows that Δ is approximately low-rank.

Lemma 27 *If $\lambda \geq 2|||\nabla\mathcal{L}(\Theta^*)|||_2$, then we have*

$$|||\Delta|||_{\mathrm{nuc}} \;\leq\; 4\sqrt{2r}|||\Delta|||_{\mathrm{F}} + 4\sum_{j=r+1}^{\min\{d_1,d_2\}}\sigma_j(\Theta^*)\,, \tag{135}$$

for all $r\in[\min\{d_1,d_2\}]$.

Proof of the above lemma is omitted because of its similarity to that of Lemma 22. The following lemma provides a bound on the gradient using the concentration in measure of sum of independent random matrices Tropp (2011).

Lemma 28 *For any positive constant $c\geq 1$ and $k\leq(1/e)\;d_2(4\log d_2+\log d_1)$, with probability at least $1-2d^{-c}-d_2^{-3}$,*

$$\begin{aligned}&|||\nabla\mathcal{L}(\Theta^*)|||_2\\ &\leq\sqrt{\frac{4(1+c)\,\log d}{k\,d_1^2}}\max\Big\{\sqrt{d_1/d_2},\;e^{2\alpha}\sqrt{4(1+c)\log(d)}(8\log d_2+2\log d_1)\log k\Big\}\,.\end{aligned} \tag{136}$$

Since we are typically interested in the regime where the number of samples is much smaller than the dimension $d_1\times d_2$ of the problem, the Hessian is typically not positive definite. However, when we restrict our attention to the vectorized Δ with relatively small nuclear norm, then we can prove restricted strong convexity, which gives the following bound.

Lemma 29 (Restricted Strong Convexity for collaborative ranking) *Fix any $\Theta\in\Omega_\alpha$ and assume $24\leq k\leq\min\{d_1^2,(d_1^2+d_2^2)/(2d_1)\}\log d$. Under the random sampling model of the alternatives $\{j_{i\ell}\}_{i\in[d_1],\ell\in[k]}$ and the random outcome of the comparisons described in section 1, with probability larger than $1-2d^{-2^{18}}$,*

$$\mathrm{Vec}(\Delta)^T\,\nabla^2\mathcal{L}(\Theta)\,\mathrm{Vec}(\Delta) \;\geq\; \frac{e^{-4\alpha}}{24\,d_1d_2}|||\Delta|||_{\mathrm{F}}^2\,, \tag{137}$$

for all Δ in $\mathcal{A}$ where

$$\mathcal{A} = \Big\{\Delta\in\mathbb{R}^{d_1\times d_2}\;\Big|\;|||\Delta|||_\infty\leq 2\alpha\,,\;\sum_{j\in[d_2]}\Delta_{ij} = 0\;\textit{for all}\;i\in[d_1]\;\textit{and}\;|||\Delta|||_{\mathrm{F}}^2\geq\mu|||\Delta|||_{\mathrm{nuc}}\Big\}\,. \tag{138}$$

with

$$\mu \;\equiv\; 2^{10}\,e^{2\alpha}\,\alpha\,d_2\sqrt{\frac{d_1\,\log d}{k\,\min\{d_1,d_2\}}}\,. \tag{139}$$

Building on these lemmas, the proof of Theorem 7 is divided into the following two cases. In both cases, we will show that

$$|||\Delta|||_{\mathrm{F}}^2 \;\leq\; 72\,e^{4\alpha}c_0\lambda_0\,d_1d_2\,|||\Delta|||_{\mathrm{nuc}}\,, \tag{140}$$

with high probability. Applying Lemma 27 proves the desired theorem. We are left to show Eq. (140) holds.

Case 1: Suppose $|||\Delta|||_{\mathrm{F}}^2\geq\mu|||\Delta|||_{\mathrm{nuc}}$. With $\Delta = \Theta^* - \widehat{\Theta}$, the Taylor expansion yields

$$\mathcal{L}(\widehat{\Theta}) = \mathcal{L}(\Theta^*) - \langle\!\langle\nabla\mathcal{L}(\Theta^*),\Delta\rangle\!\rangle + \frac{1}{2}\mathrm{Vec}(\Delta)\nabla^2\mathcal{L}(\Theta)\mathrm{Vec}^T(\Delta), \tag{141}$$

where $\Theta = a\widehat{\Theta}+(1-a)\Theta^*$ for some $a\in[0,1]$. It follows from Lemma 29 that with probability at least $1-2d^{-2^{18}}$,

$$\begin{aligned}\mathcal{L}(\widehat{\Theta})-\mathcal{L}(\Theta^*) &\geq -\langle\!\langle\nabla\mathcal{L}(\Theta^*),\Delta\rangle\!\rangle + \frac{e^{-4\alpha}}{48\,d_1\,d_2}|||\Delta|||_{\mathrm{F}}^2\\ &\geq -|||\nabla\mathcal{L}(\Theta^*)|||_2|||\Delta|||_{\mathrm{nuc}} + \frac{e^{-4\alpha}}{48\,d_1\,d_2}|||\Delta|||_{\mathrm{F}}^2\,.\end{aligned}$$

From the definition of $\widehat{\Theta}$ as an optimal solution of the minimization, we have

$$\mathcal{L}(\widehat{\Theta}) - \mathcal{L}(\Theta^*) \;\le\; \lambda\left(|||\Theta^*|||_{\rm nuc} - \left|\left|\left|\widehat{\Theta}\right|\right|\right|_{\rm nuc}\right) \;\le\; \lambda|||\Delta|||_{\rm nuc}\,.$$

By the assumption, we choose $\lambda \ge 480\lambda_0$. In view of Lemma 28, this implies that $\lambda \ge 2|||\nabla\mathcal{L}(\Theta^*)|||_2$ with probability at least $1-2d^{-3}$. It follows that with probability at least $1-2d^{-3}-2d^{-2^{18}}$,

$$\frac{e^{-4\alpha}}{48d_1d_2}|||\Delta|||_{\rm F}^2 \;\le\; \left(\lambda + |||\nabla\mathcal{L}(\Theta^*)|||_2\right)|||\Delta|||_{\rm nuc} \;\le\; \frac{3\lambda}{2}|||\Delta|||_{\rm nuc}\,.$$

By our assumption on $\lambda \le c_0\lambda_0$, this proves the desired bound in Eq. (140)
Case 2: Suppose $|||\Delta|||_{\rm F}^2 \le \mu|||\Delta|||_{\rm nuc}$. By the definition of μ and the fact that $c_0 \ge 480$, it follows that $\mu \le 72\,e^{4\alpha}c_0\lambda_0\,d_1d_2$, and we get the same bound as in Eq. (140).

C.1 Proof of Lemma 28

Define $X_i = -e_i\sum_{\ell=1}^k(e_{v_{i,\ell}} - p_{i,\ell})^T$ such that $\nabla\mathcal{L}(\Theta^*) = \frac{1}{k\,d_1}\sum_{i=1}^{d_1}X_i$, which is a sum of d_1 independent random matrices. Although $|||X_i|||_2$ can be as large as $O(k)$, this occurs with very low probability. We make this precise in the following lemma and focus on the case where $|||X_i|||_2 = O(\sqrt{k})$ for all $i \in [d_1]$.

Lemma 30 *For a fixed $i \in [d_1]$ and $j \in [d_2]$, if $k \le (1/e)\,d_2\,(4\log d_2 + \log d_1)$, then the number of times the item j is observed by the user i is at most $8(\log d_2) + 2(\log d_1)$ with probability larger than $1 - 1/(d_2^4d_1)$.*

Proof is given in the end of this Section. Applying union bound over the d_1 items and d_2 users, we have the multiplicity in sampling for any item for all users is bounded by $8(\log d_2) + 2(\log d_1)$ with probability at least $1 - d_2^{-3}$. We denote this event by $\mathcal{A}$ and let $\mathbb{I}(\mathcal{A})$ be the indicator function that all the multiplicities in sampling are bounded. We first upper bound $|||(\sum_i X_i)\,\mathbb{I}(\mathcal{A})|||_2$ using the Matrix Bernstein inequality Tropp (2011).

$$\begin{aligned}
|||X_i\mathbb{I}(\mathcal{A})|||_2 &= \left\|\mathbb{I}(\mathcal{A})\sum_{\ell=1}^k\left(e_{v_{i,\ell}} - p_{i,\ell}\right)\right\| \\
&\overset{(a)}{\le} \left\|\mathbb{I}(\mathcal{A})\sum_{\ell=1}^k e_{v_{i,l}}\right\| + \left\|\mathbb{I}(\mathcal{A})\sum_{\ell=1}^k p_{i,\ell}\right\| \\
&\overset{(b)}{\le} (8(\log d_2) + 2(\log d_1))\sqrt{\min\{k,d_2\}}\left(1 + \left(\sum_{\ell=1}^k\frac{e^{2\alpha}}{\ell}\right)\right) \\
&\overset{(c)}{\le} \sqrt{k}(8(\log d_2) + 2(\log d_1))\left(1 + 2e^{2\alpha}\;\log k\right) \\
&\le 3\sqrt{k}(8(\log d_2) + 2(\log d_1))e^{2\alpha}\;\log k\,, \qquad (142)
\end{aligned}$$

where (a) is by triangle inequality, (b) is because under the given event $\mathcal{A}$ each term in $\sum_\ell e_{v_{i,\ell}}$ and $\sum_l p_{i,\ell}$ are upper bounded by $\log d_2$ and $\left(\sum_{\ell=1}^k\frac{e^{2\alpha}}{\ell}\right)\log d_2$ respectively and because there can be at most $\min\{d_2,k\}$ non-zero entries in the two vectors $\sum_\ell e_{v_{i,\ell}}$ and

$\sum_\ell p_{i,\ell}$ and, (c) is due to the fact that k-th harmonic number $\sum_{\ell=1}^k\frac{1}{\ell}$ is upper bounded by $\log k$. We also have,

$$\begin{aligned}
\left\|\sum_i\mathbb{E}\left[X_iX_i^T\mathbb{I}(\mathcal{A})\right]\right\|_2 &\le \left\|\sum_i\mathbb{E}\left[X_iX_i^T\right]\right\|_2 \\
&\le \left\|\sum_{i=1}^{d_1}e_ie_i^T\mathbb{E}\left[\sum_{\ell,\ell'=1}^k\left(e_{v_{i,\ell}} - p_{i,\ell}\right)^T\left(e_{v_{i,\ell'}} - p_{i,\ell'}\right)\right]\right\|_2 \\
&= \left\|\sum_{i=1}^{d_1}e_ie_i^T\mathbb{E}\left[\sum_{\ell=1}^k\left(e_{v_{i,\ell}} - p_{i,\ell}\right)^T\left(e_{v_{i,\ell}} - p_{i,\ell}\right)\right]\right\|_2 \\
&= \left\|\sum_{i=1}^{d_1}e_ie_i^T\mathbb{E}\left[\sum_{\ell=1}^k e_{v_{i,\ell}}^Te_{v_{i,\ell}} - p_{i,\ell}^Tp_{i,\ell}\right]\right\|_2 \\
&\le \left\|\sum_{i=1}^{d_1}e_ie_i^T\mathbb{E}\left[\sum_{\ell=1}^k e_{v_{i,\ell}}^Te_{v_{i,\ell}}\right]\right\|_2 \\
&= k|||\mathbf{I}_{d_1\times d_1}|||_2 = k, \qquad (143)
\end{aligned}$$

and

$$\begin{aligned}
\left\|\sum_{i=1}^{d_1}\mathbb{E}\left[X_i^TX_i\mathbb{I}(\mathcal{A})\right]\right\|_2 &\le \left\|\sum_{i=1}^{d_1}\mathbb{E}\left[X_i^TX_i\right]\right\|_2 \\
&\le \left\|\sum_{i=1}^{d_1}\mathbb{E}\left[\sum_{\ell,\ell'=1}^k(e_{v_{i,\ell}} - p_{i,\ell})(e_{v_{i,\ell'}} - p_{i,\ell'})^T\right]\right\|_2 \\
&= \left\|\sum_{i=1}^{d_1}\mathbb{E}\left[\sum_{\ell=1}^k(e_{v_{i,\ell}} - p_{i,\ell})(e_{v_{i,\ell}} - p_{i,\ell})^T\right]\right\|_2 \qquad (144) \\
&= \left\|\sum_{i=1}^{d_1}\mathbb{E}\left[\sum_{\ell=1}^k e_{v_{i,\ell}}e_{v_{i,\ell}}^T - p_{i,\ell}p_{i,\ell}^T\right]\right\|_2 \\
&\le \left\|\sum_{i=1}^{d_1}\mathbb{E}\left[\sum_{\ell=1}^k e_{v_{i,\ell}}e_{v_{i,\ell}}^T\right]\right\|_2 \\
&= \left\|\sum_{i=1}^{d_1}\frac{k}{d_2}\mathbf{I}_{d_2\times d_2}\right\|_2 = \frac{kd_1}{d_2}\,. \qquad (145)
\end{aligned}$$

By matrix Bernstein inequality Tropp (2011),

$$\begin{aligned}
&\mathbb{P}\Big(|||\nabla\mathcal{L}(\Theta^*)\mathbb{I}(\mathcal{A})|||_2 > t\Big) \\
&\le (d_1+d_2)\exp\Big(\frac{-k^2\,d_1^2t^2/2}{(d_1k/\min\{d_2,d_1\}) + (3e^{2\alpha}k^{3/2}d_1(8(\log d_2) + 2(\log d_1))\log k\;t/3)}\Big)\,,
\end{aligned}$$

which gives the tail probability of $2d^{-c}$ for the choice of

$$t = \max\left\{\sqrt{\frac{4(1+c)\,\log d}{k\,d_1\,\min\{d_2,d_1\}}},\ \frac{4(1+c)e^{2\alpha}\log(d)\,(8(\log d_2)+2(\log d_1))\log k}{k^{1/2}\,d_1}\right\}$$
$$= \frac{\sqrt{4(1+c)\,\log d}}{k^{1/2}\,d_1}\max\left\{\sqrt{d_1/d_2}\,,\ e^{2\alpha}\sqrt{4(1+c)\log(d)}\,(8(\log d_2)+2(\log d_1))\log k\right\}.$$

Now with a high probability of $1-\frac{2}{d^c}-\frac{1}{d_2^3}$ the desired bound is true.

C.2 Proof of Lemma 30

In a classical balls-in-bins setting, we consider k as the number of balls and d_2 as the number of bins. We can consider the number of balls in a particular bin as the number of times the user i observes item j. Let the event that this number is at least δ be denoted by the event A_δ^j. Then, $\mathbb{P}\left\{A_\delta^j\right\} \le \binom{k}{\delta}\frac{1}{d_2^\delta} \le \left(\frac{ke}{d_2\delta}\right)^\delta$. Using the fact that $(1/x)^x \le a$ for any $x \ge (2\log(1/a))/(\log\log(1/a))$, we let $x = d_2\delta/(ke)$ to get

$$\left(\frac{ke}{d_2\delta}\right)^\delta \le a^{\frac{ke}{d_2}}\,,$$

for $\delta \ge (ke/d_2)(2\log(1/a))/(\log\log(1/a))$. Choosing $a = (1/d_2^4 d_1)^{d_2/ke}$, we have $\mathbb{P}\left\{A_\delta^j\right\} \le 1/(d_1 d_2^4)$, for a choice of $\delta = 2\ \log(d_2^4 d_1) \ge 2\log(d_2^4 d_1)/(\log((d_2/ke)\log(d_2^4 d_1)))$.

C.3 Proof of Lemma 29

Recall that the Hessian matrix is a block-diagonal matrix with the i-th block $H^{(i)}(\Theta)$ given by (134). We use the following remark from Hajek et al. (2014) to bound the Hessian.

Remark 31 *(Hajek et al., 2014, Claim 1) Given $\theta \in \mathbb{R}^r$, let p be the column probability vector with $p_i = e^{\theta_i}/(e^{\theta_1}+\cdots+e^{\theta_\rho})$ for each $i \in [\rho]$ and for any positive integer ρ. If $|\theta_i| \le \alpha$, for all $i \in [\rho]$, then*

$$e^{2\alpha}\Big(\mathrm{diag}(p) - pp^T\Big) \ \succeq\ \frac{1}{\rho}\mathrm{diag}(\mathbb{1}) - \frac{1}{\rho^2}\mathbb{1}\mathbb{1}^T\,.$$

By letting $\mathbb{1}_{S_{i,\ell}} = \sum_{j\in S_{i,\ell}} e_j$ and applying the above claim, we have

$$e^{2\alpha}H^{(i)}(\Theta) \succeq \frac{1}{k\,d_1}\sum_{\ell=1}^{k}\left(\frac{1}{k-\ell+1}\mathrm{diag}(\mathbb{1}_{S_{i,\ell}}) - \frac{1}{(k-\ell+1)^2}\mathbb{1}_{S_{i,\ell}}\mathbb{1}_{S_{i,\ell}}^T\right)$$
$$= \frac{1}{2\,k\,d_1}\sum_{\ell=1}^{k}\frac{1}{(k-\ell+1)^2}\sum_{j,j'\in S_{i,\ell}}(e_j-e_{j'})(e_j-e_{j'})^T$$
$$\succeq \frac{1}{2\,k^3\,d_1}\sum_{\ell=1}^{k}\sum_{j,j'\in S_{i,\ell}}(e_j-e_{j'})(e_j-e_{j'})^T.$$

Hence,

$$\mathrm{Vec}(\Delta)\nabla^2\mathcal{L}(\Theta)\mathrm{Vec}^T(\Delta) = \sum_{i=1}^{d_1}(\Delta^T e_i)^T H^{(i)}(\Theta)(\Delta^T e_i)$$
$$\ge \frac{e^{-2\alpha}}{2\,k^3\,d_1}\sum_{i=1}^{d_1}\sum_{\ell=1}^{k}\sum_{j,j'\in S_{i,\ell}}\left|\left|\left|e_i^T\Delta(e_j-e_{j'})\right|\right|\right|_2^2.$$

By changing the order of the summation, we get that

$$\sum_{\ell=1}^{k}\sum_{j,j'\in S_{i,\ell}}\left|\left|\left|e_i^T\Delta(e_j-e_{j'})\right|\right|\right|_2^2 = \sum_{\ell,\ell'=1}^{k}\langle\!\langle\Delta, e_{i,j_{i,\ell}}-e_{i,j_{i,\ell'}}\rangle\!\rangle^2\sum_{\ell''=1}^{k}\mathbb{I}\big(\,\sigma_i(j_{i,\ell''})$$
$$\le \min\{\sigma_i(j_{i,\ell}),\sigma_i(j_{i,\ell'})\}\big).$$

Define

$$\chi_{i,\ell,\ell',\ell''} \ \equiv\ \mathbb{I}\big(\,\sigma_i(j_{i,\ell''}) \le \min\{\sigma_i(j_{i,\ell}),\sigma_i(j_{i,\ell'})\}\,\big)\,, \tag{146}$$

and let

$$H(\Delta)\ \equiv\ \frac{e^{-2\alpha}}{2\,k^3\,d_1}\sum_{i=1}^{d_1}\sum_{\ell,\ell'=1}^{k}\langle\!\langle\Delta, e_{i,j_{i,\ell}}-e_{i,j_{i,\ell'}}\rangle\!\rangle^2\sum_{\ell''=1}^{k}\chi_{i,\ell,\ell',\ell''}.$$

Then we have $\mathrm{Vec}^T(\Delta)\nabla^2\mathcal{L}(\Theta)\mathrm{Vec}(\Delta) \ge H(\Delta)$. To prove the theorem, it suffices to bound $H(\Delta)$ from the below. First, we prove a lower bound on the expectation $\mathbb{E}[H(\Delta)]$. Notice that for $\ell \ne \ell'$, the conditional expectation of $\chi_{i,\ell,\ell',\ell''}$'s, given the set of alternatives presented to user i is

$$\mathbb{E}\Big[\sum_{\ell''=1}^{k}\chi_{i,\ell,\ell',\ell''}\,\Big|\,j_{i,1},\ldots,j_{i,k}\Big] = 1+\sum_{\ell''\ne\ell,\ell'}\frac{\exp(\theta_{i,j_{i,\ell''}})}{\exp(\theta_{i,j_{i,\ell''}})+\exp(\theta_{i,j_{i,\ell'}})+\exp(\theta_{i,j_{i,\ell}})}$$
$$\ge 1+\frac{k-2}{1+2e^{2\alpha}} \ge \frac{k}{3e^{2\alpha}}.$$

Then,

$$\mathbb{E}[H(\Delta)] = \frac{e^{-2\alpha}}{2\,k^3\,d_1}\sum_{i,\ell,\ell'}\mathbb{E}\Big[\langle\!\langle\Delta, e_{i,j_{i,\ell}}-e_{i,j_{i,\ell'}}\rangle\!\rangle^2\mathbb{E}\big[\sum_{\ell''=1}^{k}\chi_{i,\ell,\ell',\ell''}\,\big|\,j_{i,1},\ldots,j_{i,k}\big]\Big]$$
$$\ge \frac{e^{-4\alpha}}{6\,k^2\,d_1}\sum_{i=1}^{d_1}\sum_{\ell,\ell'\in[k]}\mathbb{E}\left[\langle\!\langle\Delta, e_{i,j_{i,\ell}}-e_{i,j_{i,\ell'}}\rangle\!\rangle^2\right]$$
$$= \frac{e^{-4\alpha}}{6\,k^2\,d_1}\sum_{i=1}^{d_1}\sum_{\ell\ne\ell'\in[k]}\left(\frac{2}{d_2}\sum_{j=1}^{d_2}\Delta_{ij}^2 - \frac{2}{d_2^2}\sum_{j,j'=1}^{d_2}\Delta_{ij}\Delta_{ij'}\right)$$
$$= \frac{e^{-4\alpha}(k-1)}{3\,k\,d_1\,d_2}|||\Delta|||_{\mathrm{F}}^2\,, \tag{147}$$

where the last equality holds because $\sum_{j\in[d_2]}\Delta_{ij}=0$ for $\Delta\in\Omega_{2\alpha}$ and for all $i\in[d_1]$.

We are left to prove that $H(\Delta)$ cannot deviate from its mean too much. Suppose there exists a $\Delta\in\mathcal{A}$ such that Eq. (137) is violated, i.e. $H(\Delta)<(e^{-4\alpha}/(24\,d_1d_2))|||\Delta|||_{\rm F}^2$. We will show this happens with a small probability. From Eq. (147), we get that for $k\geq 24$,

$$\begin{aligned}\mathbb{E}[H(\Delta)]-H(\Delta) &\geq \frac{(7k-8)}{24k}\frac{e^{-4\alpha}}{d_1\,d_2}|||\Delta|||_{\rm F}^2\\ &\geq \frac{(20/3)\,e^{-4\alpha}}{24\,d_1d_2}|||\Delta|||_{\rm F}^2\,. \end{aligned}\tag{148}$$

We use a peeling argument as in (Negahban and Wainwright, 2012, Lemma 3), Van De Geer (2000) to upper bound the probability that Eq. (148) is true. We first construct the following family of subsets to cover $\mathcal{A}$ such that $\mathcal{A}\subseteq\bigcup_{\ell=1}^{\infty}\mathcal{S}_\ell$. Recall $\mu=2^{10}e^{2\alpha}\alpha d_2\sqrt{(d_1\log d)/(k\min\{d_1,d_2\})}$, define in (139). Notice that since for any $\Delta\in\mathcal{A}$, $|||\Delta|||_{\rm F}^2\geq\mu|||\Delta|||_{\rm nuc}\geq\mu|||\Delta|||_{\rm F}$, it follows that $|||\Delta|||_{\rm F}\geq\mu$. Then, we can cover $\mathcal{A}$ with the family of sets

$$\mathcal{S}_\ell=\Big\{\Delta\in\mathbb{R}^{d_1\times d_2}\,\Big|\,|||\Delta|||_\infty\leq 2\alpha\,,\,\beta^{\ell-1}\mu\leq|||\Delta|||_{\rm F}\leq\beta^\ell\mu\,,\\ \sum_{j\in[d_2]}\Delta_{ij}=0\text{ for all }i\in[d_1],\text{ and }|||\Delta|||_{\rm nuc}\leq\beta^{2\ell}\mu\Big\}\,,$$

where $\beta=\sqrt{10/9}$ and for $\ell\in\{1,2,3,\ldots\}$. This implies that when there exists a $\Delta\in\mathcal{A}$ such that (148) holds, then there exists an $\ell\in\mathbb{Z}_+$ such that $\Delta\in\mathcal{S}_\ell$ and

$$\begin{aligned}\mathbb{E}[H(\Delta)]-H(\Delta) &\geq \frac{(20/3)\,e^{-4\alpha}}{24\,d_1d_2}\beta^{2(\ell-1)}\mu^2\\ &\geq \frac{e^{-4\alpha}}{4\,d_1d_2}\beta^{2\ell}\mu^2\,.\end{aligned}\tag{149}$$

Applying the union bound over $\ell\in\mathbb{Z}_+$, we get from (148) and (149) that

$$\begin{aligned}&\mathbb{P}\Big\{\exists\Delta\in\mathcal{A}\,,\;H(\Delta)<\frac{e^{-4\alpha}}{24\,d_1d_2}|||\Delta|||_{\rm F}^2\Big\}\\ &\leq\sum_{\ell=1}^{\infty}\mathbb{P}\Big\{\sup_{\Delta\in\mathcal{S}_\ell}\big(\,\mathbb{E}[H(\Delta)]-H(\Delta)\,\big)>\frac{e^{-4\alpha}}{4\,d_1d_2}(\beta^\ell\mu)^2\Big\}\\ &\leq\sum_{\ell=1}^{\infty}\mathbb{P}\Big\{\sup_{\Delta\in\mathcal{B}(\beta^\ell\mu)}\big(\,\mathbb{E}[H(\Delta)]-H(\Delta)\,\big)>\frac{e^{-4\alpha}}{4\,d_1d_2}(\beta^\ell\mu)^2\Big\}\,,\end{aligned}\tag{150}$$

where we define a new set $\mathcal{B}(D)$ such that $\mathcal{S}_\ell\subseteq\mathcal{B}(\beta^\ell\mu)$:

$$\mathcal{B}(D)=\big\{\,\Delta\in\mathbb{R}^{d_1\times d_2}\,\big|\,|||\Delta|||_\infty\leq 2\alpha,|||\Delta|||_{\rm F}\leq D,\\ \sum_{j\in[d_2]}\Delta_{ij}=0\text{ for all }i\in[d_1],\mu|||\Delta|||_{\rm nuc}\leq D^2\,\big\}\,.\tag{151}$$

The following key lemma provides the upper bound on this probability.

Lemma 32 *For* $(16\min\{d_1,d_2\}\log d)/(3d_1)\leq k\leq d_1^2\log d$,

$$\mathbb{P}\Big\{\sup_{\Delta\in\mathcal{B}(D)}\big(\,\mathbb{E}[H(\Delta)]-H(\Delta)\,\big)\geq\frac{e^{-4\alpha}}{4d_1d_2}D^2\Big\}\;\leq\;\exp\Big\{-\frac{e^{-4\alpha}\,k\,D^4}{2^{19}\alpha^4d_1d_2^2}\Big\}\,.\tag{152}$$

Let $\eta=\exp\Big(-\frac{e^{-4\alpha}4k(\beta-1.002)\mu^4}{2^{19}\alpha^4d_1d_2^2}\Big)$. Applying the tail bound to (150), we get

$$\begin{aligned}\mathbb{P}\Big\{\exists\Delta\in\mathcal{A}\,,\;H(\Delta)<\frac{e^{-4\alpha}}{24\,d_1d_2}|||\Delta|||_{\rm F}^2\Big\} &\leq \sum_{\ell=1}^{\infty}\exp\Big\{-\frac{e^{-4\alpha}k(\beta^\ell\mu)^4}{2^{19}\alpha^4d_1d_2^2}\Big\}\\ &\overset{(a)}{\leq}\sum_{\ell=1}^{\infty}\exp\Big\{-\frac{e^{-4\alpha}4k\ell(\beta-1.002)\mu^4}{2^{19}\alpha^4d_1d_2^2}\Big\}\\ &\leq\frac{\eta}{1-\eta},\end{aligned}$$

where (a) holds because $\beta^x\geq x\log\beta\geq x(\beta-1.002)$ for the choice of $\beta=\sqrt{10/9}$. By the definition of μ,

$$\eta\;=\;\exp\Big\{-\frac{2^{23}\,e^{4\alpha}d_2^2d_1(\log d)^2(\beta-1.002)}{k(\min\{d_1,d_2\})^2}\Big\}\;\leq\;\exp\{-2^{18}\,\log d\}\,,$$

where the last inequality follows from the assumption that $k\leq\max\{d_1,d_2^2/d_1\}\log d=(d_2^2d_1\log d)/(\min\{d_1,d_2\})^2$, and $\beta-1.002\geq 2^{-5}$. Since for $d\geq 2$, $\exp\{-2^{18}\log d\}\leq 1/2$ and thus $\eta\leq 1/2$, the lemma follows by assembling the last two displayed inequalities.

C.4 Proof of Lemma 32

Recall that

$$H(\Delta)=\frac{e^{-2\alpha}}{2\,k^3\,d_1}\sum_{i=1}^{d_1}\sum_{\ell,\ell'=1}^{k}\langle\!\langle\Delta,e_{i,j_{i,\ell}}-e_{i,j_{i,\ell'}}\rangle\!\rangle^2\sum_{\ell''=1}^{k}\chi_{i,\ell,\ell',\ell''}\,,$$

with $\chi_{i,\ell,\ell',\ell''}=\mathbb{I}\Big(\sigma_i(j_{i,\ell''})\leq\min\{\sigma_i(j_{i,\ell}),\sigma_i(j_{i,\ell'})\}\Big)$. Let $Z=\sup_{\Delta\in\mathcal{B}(D)}\mathbb{E}[H(\Delta)]-H(\Delta)$ be the worst-case random deviation of $H(\Delta)$ form its mean. We prove an upper bound on Z by showing that $Z-\mathbb{E}[Z]\leq e^{-4\alpha}D^2/(64d_1d_2)$ with high probability, and $\mathbb{E}[Z]\leq 9e^{-4\alpha}D^2/(40d_1d_2)$. This proves the desired claim in Lemma 32.

To prove the concentration of Z, we utilize the random utility model (RUM) theoretic interpretation of the MNL model. The random variable Z depends on the random choice of alternatives $\{j_{i,\ell}\}_{i\in[d_1],\ell\in[k]}$ and the random k-wise ranking outcomes $\{\sigma_i\}_{i\in[d_1]}$. The random utility theory, pioneered by Thurstone (1927); Marschak (1960); Luce (1959), tells us that the k-wise ranking from the MNL model has the same distribution as first drawing independent (unobserved) utilities $u_{i,\ell}$'s of the item $j_{i,\ell}$ for user i according to the standard Gumbel Cumulative Distribution Function (CDF) $F(c-\Theta_{i,j_{i,\ell}})$ with $F(c)=e^{-e^{-c}}$, and then ranking the k items for user i according to their respective utilities. Given this definition of the MNL model, we have $\chi_{i,\ell,\ell',\ell''}=\mathbb{I}\Big(u_{i,\ell''}\geq\max\{u_{i,\ell},u_{i,\ell'}\}\Big)$. Thus Z is a function of independent

choices of the items and their (unobserved) utilities, i.e. $Z = f(\{(j_{i,\ell}, u_{i,\ell})\}_{i\in[d_1],\ell\in[k]})$. Let $x_{i,\ell} = (j_{i,\ell}, u_{i,\ell})$ and write $H(\Delta)$ as $H(\Delta, \{x_{i,\ell}\}_{i\in[d_1],\ell\in[k]})$. This allows us to bound the difference and apply McDiarmid's tail bound. Note that for any $i \in [d_1]$, $\ell \in [k]$, $x_{1,1}, \ldots, x_{d_1,k}$, and $x'_{i,\ell}$,

$$
\begin{aligned}
&\left| f\big(x_{1,1}, \ldots, x_{i,\ell}, \ldots, x_{d_1,k}\big) - f\big(x_{1,1}, \ldots, x'_{i,\ell}, \ldots, x_{d_1,k}\big) \right| \\
&= \Big| \sup_{\Delta\in\mathcal{B}(D)} \big(\mathbb{E}\left[H(\Delta)\right] - H(\Delta, x_{1,1}, \ldots, x_{i,\ell}, \ldots, x_{d_1,k})\big) - \\
&\sup_{\Delta\in\mathcal{B}(D)} \big(\mathbb{E}\left[H(\Delta)\right] - H(\Delta, x_{1,1}, \ldots, x'_{i,\ell}, \ldots, x_{d_1,k})\big) \Big| \\
&\le \sup_{\Delta\in\mathcal{B}(D)} \left|H(\Delta, x_{1,1}, \ldots, x_{i,\ell}, \ldots, x_{d_1,k}) - H(\Delta, x_{1,1}, \ldots, x'_{i,\ell}, \ldots, x_{d_1,k})\right| \\
&\overset{(a)}{\le} \frac{e^{-2\alpha}}{2\,k^3\,d_1} \sup_{\Delta\in\mathcal{B}(D)} \Big\{ 2 \sum_{\ell'\in[k]} \langle\!\langle \Delta, e_{i,j_{i,\ell}} - e_{i,j_{i,\ell'}} \rangle\!\rangle^2 \sum_{\ell''=1}^{k} \chi_{i,\ell,\ell',\ell''} + \\
&\sum_{\ell',\ell''\in[k]} \langle\!\langle \Delta, e_{i,j_{i,\ell'}} - e_{i,j_{i,\ell''}} \rangle\!\rangle^2 \chi_{i,\ell',\ell'',\ell} \Big\} \\
&\overset{(b)}{\le} \frac{8\alpha^2 e^{-2\alpha}}{k^3\,d_1} \Big\{ 2 \sum_{\ell'\in[k]\setminus\{\ell\}} \sum_{\ell''=1}^{k} \chi_{i,\ell,\ell',\ell''} + \sum_{\ell',\ell''\in[k],\ell'\neq\ell''} \chi_{i,\ell',\ell'',\ell} \Big\} \\
&\le \frac{16\alpha^2 e^{-2\alpha}}{k\,d_1} \,,
\end{aligned}
$$

where (a) follows because for a fixed i and ℓ, the random variable $x_{i,\ell} = (j_{i,\ell}, u_{i,\ell})$ can appear in three terms, i.e. $\sum_{\ell',\ell''} \langle\!\langle \Delta, e_{i,j_{i,\ell}} - e_{i,j_{i,\ell'}} \rangle\!\rangle^2 \chi_{i,\ell,\ell',\ell''} + \sum_{\ell',\ell''} \langle\!\langle \Delta, e_{i,j_{i,\ell'}} - e_{i,j_{i,\ell}} \rangle\!\rangle^2 \chi_{i,\ell',\ell,\ell''} + \sum_{\ell',\ell''} \langle\!\langle \Delta, e_{i,j_{i,\ell'}} - e_{i,j_{i,\ell''}} \rangle\!\rangle^2 \chi_{i,\ell',\ell'',\ell}$, and (b) follows because $|\Delta_{ij}| \le 2\alpha$ for all i, j since $\Delta \in \mathcal{B}(D)$. The last inequality follows because in the worst case, $\sum_{\ell'\in[k]\setminus\{\ell\}} \sum_{\ell''=1}^{k} \chi_{i,\ell,\ell',\ell''} \le k(k-1)/2$ and $\sum_{\ell',\ell''\in[k],\ell'\neq\ell''} \chi_{i,\ell',\ell'',\ell} \le k(k-1)$. This holds with equality if $\sigma_i(j_{i,\ell}) = k$ and $\sigma_i(j_{i,\ell}) = 1$, respectively. By bounded differences inequality, we have

$$
\mathbb{P}\left\{Z - \mathbb{E}\left[Z\right] \ge t\right\} \le \exp\left(-\frac{k^2\,d_1^2\,t^2}{2^7\,\alpha^4 e^{-4\alpha} d_1 k}\right),
$$

It follows that for the choice of $t = e^{-4\alpha} D^2/(64 d_1 d_2)$,

$$
\mathbb{P}\left\{Z - \mathbb{E}\left[Z\right] \ge \frac{e^{-4\alpha} D^2}{64 d_1 d_2}\right\} \le \exp\Big(-\frac{e^{-4\alpha} k D^4}{2^{19}\alpha^4 d_1 d_2^2}\Big)\,.
$$

We are left to prove the upper bound on $\mathbb{E}[Z]$ using symmetrization and contraction. Define random variables

$$
Y_{i,\ell,\ell',\ell''}(\Delta) \;\equiv\; (\Delta_{i,j_{i,\ell}} - \Delta_{i,j_{i,\ell'}})^2 \chi_{i,\ell,\ell',\ell''}\,, \tag{153}
$$

where the randomness is in the choice of alternatives $j_{i,\ell}, j_{i,\ell'}$, and $j_{i,\ell''}$, and the outcome of the comparisons of those three alternatives.

The main challenge in applying the symmetrization to $\sum_{\ell,\ell',\ell''\in[k]} Y_{i,\ell,\ell',\ell''}(\Delta)$ is that we need to partition the summation over the set $[k] \times [k] \times [k]$ into subsets of independent random variables, such that we can apply the standard symmetrization argument. To this end, we prove in the following lemma a generalization of the well-known problem of scheduling a round robin tournament to a tournament of matches involving three teams each. No teams are present in more than one triple in a single round, and we want to minimize the number of rounds to cover all combination of triples are matched. For example, when there are $k = 6$ teams, there is a simple construction of such a tournament: $T_1 = \{(1,2,3),(4,5,6)\}$, $T_2 = \{1,2,4),(3,5,6)\}$, $T_3 = \{(1,2,5),(3,4,6)\}$, $T_4 = \{(1,2,6),(3,4,5)\}$, $T_5 = \{(1,3,4),(2,5,6)\}$, $T_6 = \{(1,3,5),(2,4,6)\}$, $T_7 = \{(1,3,6),(2,4,5)\}$, $T_8 = \{(1,4,5),(2,3,6)\}$, $T_9 = \{(1,4,6),(2,3,5)\}$, $T_{10} = \{(1,5,6),(2,3,4)\}$. This is a perfect scheduling of a tournament with three teams in each match. For a general k, the following lemma provides a construction with $O(k^2)$ rounds.

Lemma 33 *There exists a partition $(T_1, \ldots, T_N)$ of $[k] \times [k] \times [k]$ for some $N \le 24k^2$ such that T_a's are disjoint subsets of $[k] \times [k] \times [k]$, $\bigcup_{a\in[N]} T_a = [k] \times [k] \times [k]$, $|T_a| \le \lfloor k/3 \rfloor$ and for any $a \in [N]$ the set of random variables in T_a satisfy*

$$
\{Y_{i,\ell,\ell',\ell''}\}_{i\in[d_1],(\ell,\ell',\ell'')\in T_a} \text{ are mutually independent } .
$$

Now, we are ready to partition the summation.

$$
\begin{aligned}
\mathbb{E}\big[Z\big] &= \frac{e^{-2\alpha}}{2\,k^3\,d_1} \mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)} \sum_{i\in[d_1]} \sum_{\ell,\ell',\ell''\in[k]} \big\{\mathbb{E}[Y_{i,\ell,\ell',\ell''}(\Delta)] - Y_{i,\ell,\ell',\ell''}(\Delta)\big\}\Big] \\
&= \frac{e^{-2\alpha}}{2\,k^3\,d_1} \mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)} \sum_{i\in[d_1]} \sum_{a\in[N]} \sum_{(\ell,\ell',\ell'')\in T_a} \big\{\mathbb{E}[Y_{i,\ell,\ell',\ell''}(\Delta)] - Y_{i,\ell,\ell',\ell''}(\Delta)\big\}\Big] \\
&\le \frac{e^{-2\alpha}}{2\,k^3\,d_1} \sum_{a\in[N]} \mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)} \sum_{i\in[d_1]} \sum_{(\ell,\ell',\ell'')\in T_a} \big\{\mathbb{E}[Y_{i,\ell,\ell',\ell''}(\Delta)] - Y_{i,\ell,\ell',\ell''}(\Delta)\big\}\Big] \\
&\le \frac{e^{-2\alpha}}{k^3\,d_1} \sum_{a\in[N]} \mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)} \sum_{i\in[d_1]} \sum_{(\ell,\ell',\ell'')\in T_a} \xi_{i,\ell,\ell',\ell''} Y_{i,\ell,\ell',\ell''}(\Delta)\Big] \\
&= \frac{e^{-2\alpha}}{k^3\,d_1} \sum_{a\in[N]} \mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)} \sum_{i\in[d_1]} \sum_{(\ell,\ell',\ell'')\in T_a} \xi_{i,\ell,\ell',\ell''} (\Delta_{i,j_{i,\ell}} - \Delta_{i,j_{i,\ell'}})^2 \chi_{i,\ell,\ell',\ell''}\Big]\,, \qquad (154)
\end{aligned}
$$

where the first inequality follows from the fact that sum of the supremum is no less than the supremum of the sum, and the second inequality follows from standard symmetrization argument applied to independent random variables $\{Y_{i,\ell,\ell',\ell''}(\Delta)\}_{i\in[d_1],(\ell,\ell',\ell'')\in T_a}$ with i.i.d. Rademacher random variables $\xi_{i,\ell,\ell',\ell''}$'s. Since $(\Delta_{i,j_{i,\ell}} - \Delta_{i,j_{i,\ell'}})^2 \chi_{i,\ell,\ell',\ell''} \le 4\alpha |\Delta_{i,j_{i,\ell}} - \Delta_{i,j_{i,\ell'}}| \chi_{i,\ell,\ell',\ell''}$, we have by the Ledoux-Talagrand contraction inequality Ledoux and Tala-

grand (2013) that

$$\mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)}\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}\xi_{i,\ell,\ell',\ell''}(\Delta_{i,j_{i,\ell}}-\Delta_{i,j_{i,\ell'}})^2\chi_{i,\ell,\ell',\ell''}\Big]$$
$$\le\ 8\alpha\mathbb{E}\Big[\sup_{\Delta\in\mathcal{B}(D)}\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}\xi_{i,\ell,\ell',\ell''}\,\chi_{i,\ell,\ell',\ell''}\,\langle\!\langle\Delta,e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\rangle\!\rangle\Big] \tag{155}$$

Applying Hölder's inequality, we get that

$$\Big|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}\xi_{i,\ell,\ell',\ell''}\,\chi_{i,\ell,\ell',\ell''}\,\langle\!\langle\Delta,e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\rangle\!\rangle\Big|$$
$$\le\ |||\Delta|||_{\rm nuc}\left|\left|\left|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}\xi_{i,\ell,\ell',\ell''}\,\chi_{i,\ell,\ell',\ell''}\left(e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\right)\right|\right|\right|_2. \tag{156}$$

We are left to prove that the expected value of the right-hand side of the above inequality is bounded by $C|||\Delta|||_{\rm nuc}\sqrt{kd_1\log d/\min\{d_1,d_2\}}$ for some numerical constant C. For $i\in[d_1]$ and $(\ell,\ell',\ell'')\in T_a$, let $W_{i,\ell,\ell',\ell''}=\xi_{i,\ell,\ell',\ell''}\,\chi_{i,\ell,\ell',\ell''}\left(e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\right)$ be independent zero-mean random matrices, such that

$$|||W_{i,\ell,\ell',\ell''}|||_2=\left|\left|\left|\xi_{i,\ell,\ell',\ell''}\,\chi_{i,\ell,\ell',\ell''}\left(e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\right)\right|\right|\right|_2\le\sqrt{2}\,,$$

almost surely, and

$$\begin{aligned}\mathbb{E}[W_{i,\ell,\ell',\ell''}W_{i,\ell,\ell',\ell''}^T]&=\mathbb{E}[\big(e_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T(e_{j_{i,\ell}}-e_{j_{i,\ell'}})e_i^T\big)\chi_{i,\ell,\ell',\ell''}]\\&=2\mathbb{E}\left[\chi_{i,\ell,\ell',\ell''}\right]e_ie_i^T\\&\preceq 2e_ie_i^T\,,\end{aligned}$$

and

$$\begin{aligned}\mathbb{E}[W_{i,\ell,\ell',\ell''}^TW_{i,\ell,\ell',\ell''}]&=\mathbb{E}[\big((e_{j_{i,\ell}}-e_{j_{i,\ell'}})e_i^Te_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T\big)\chi_{i,\ell,\ell',\ell''}]\\&\preceq\mathbb{E}[(e_{j_{i,\ell}}-e_{j_{i,\ell'}})e_i^Te_i(e_{j_{i,\ell}}-e_{j_{i,\ell'}})^T]\\&=\frac{2}{d_2}\mathbf{I}_{d_2\times d_2}-\frac{2}{d_2^2}\mathbb{1}\mathbb{1}^T\,.\end{aligned}$$

This gives

$$\begin{aligned}\sigma^2&=\max\left\{\left|\left|\left|\sum_{\substack{i\in[d_1]\\(\ell,\ell',\ell'')\in T_a}}\mathbb{E}[W_{i,\ell,\ell',\ell''}W_{i,\ell,\ell',\ell''}^T]\right|\right|\right|_2,\left|\left|\left|\sum_{\substack{i\in[d_1]\\(\ell,\ell',\ell'')\in T_a}}\mathbb{E}[W_{i,\ell,\ell',\ell''}^TW_{i,\ell,\ell',\ell''}]\right|\right|\right|_2\right\}\\&\le\max\left\{2|T_a|\,,\frac{2d_1|T_a|}{d_2}\right\}=\frac{2d_1|T_a|}{\min\{d_1,d_2\}}\le\frac{2d_1k}{3\min\{d_1,d_2\}}\,,\end{aligned}$$

since we have designed T_a's such that $|T_a|\le k/3$. Applying matrix Bernstein inequality Tropp (2011) yields the tail bound

$$\mathbb{P}\left\{\left|\left|\left|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}W_{i,\ell,\ell',\ell''}\right|\right|\right|_2\ge t\right\}\ \le\ (d_1+d_2)\exp\Big(\frac{-t^2/2}{\sigma^2+\sqrt{2}t/3}\Big)\,.$$

Choosing $t=\max\big\{\sqrt{32kd_1\log d/(3\min\{d_1,d_2\})},(16\sqrt{2}/3)\log d\big\}$, we obtain with probability at least $1-2d^{-3}$,

$$\left|\left|\left|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}W_{i,\ell,\ell',\ell''}\right|\right|\right|_2\ \le\ \max\left\{\sqrt{\frac{32kd_1\log d}{3\min\{d_1,d_2\}}},\frac{16\sqrt{2}\log d}{3}\right\}\,.$$

It follows from the fact $\left|\left|\left|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}W_{i,\ell,\ell',\ell''}\right|\right|\right|_2\le\sum_{i,(\ell,\ell',\ell'')}|||W_{i,\ell,\ell',\ell''}|||_2\le\frac{\sqrt{2}d_1k}{3}$ that

$$\begin{aligned}\mathbb{E}\left[\left|\left|\left|\sum_{i\in[d_1]}\sum_{(\ell,\ell',\ell'')\in T_a}W_{i,\ell,\ell',\ell''}\right|\right|\right|_2\right]&\le\max\left\{\sqrt{\frac{32kd_1\log d}{3\min\{d_1,d_2\}}},\frac{16\sqrt{2}\log d}{3}\right\}+\frac{2\sqrt{2}d_1k}{3d^3}\\&\le 2\sqrt{\frac{32kd_1\log d}{3\min\{d_1,d_2\}}}\,,\end{aligned}$$

where the last inequality follows from the assumption that $(16\min\{d_1,d_2\}\log d)/(3d_1)\le k\le d_1^2\log d$. Substituting this in the RHS of Eq. (156), and then together with Eqs. (155) and (154), this gives the following desired bound:

$$\begin{aligned}\mathbb{E}[Z]&\le\sum_{a\in[N]}\sup_{\Delta\in\mathcal{B}(D)}\frac{16\alpha e^{-2\alpha}}{k^3\,d_1}\sqrt{\frac{32kd_1\log d}{3\min\{d_1,d_2\}}}|||\Delta|||_{\rm nuc}\\&\le\sum_{a\in[N]}\frac{e^{-4\alpha}\sqrt{2}}{16\sqrt{3}k^2\,d_1\,d_2}\underbrace{\left(2^{10}e^{2\alpha}\alpha d_2\sqrt{\frac{d_1\log d}{k\min\{d_1,d_2\}}}\right)}_{=\mu}|||\Delta|||_{\rm nuc}\\&\le\frac{9e^{-4\alpha}D^2}{40d_1d_2}\,,\end{aligned}$$

where the last inequality holds because $N\le 4k^2$ and $\mu|||\Delta|||_{\rm nuc}\le D^2$.

C.5 Proof of Lemma 33

Recall that $Y_{i,\ell,\ell',\ell''}(\Delta)=(\Delta_{i,j_{i,\ell}}-\Delta_{i,j_{i,\ell'}})^2\chi_{i,\ell,\ell',\ell''}$, as defined in (153). From the random utility model (RUM) interpretation of the MNL model presented in Section 1, it is not difficult to show that $Y_{i,\ell,\ell',\ell''}$ and $Y_{i,\tilde{\ell},\tilde{\ell}',\tilde{\ell}''}$ are mutually independent if the two triples (ℓ,ℓ',ℓ'') and $(\tilde{\ell},\tilde{\ell}',\tilde{\ell}'')$ do not overlap, i.e., no index is present in both triples.

Now, borrowing the terminologies from round robin tournaments, we construct a schedule for a tournament with k teams where each match involve three teams. Let $T_{a,b}$ denote

a set of triples playing at the same round, indexed by two integers $a \in \{3, \ldots, 2k-3\}$ and $b \in \{5, \ldots, 2k-1\}$. Hence, there are total $N = (2k-5)^2$ rounds.

Each round (a, b) consists of disjoint triples and is defined as

$$T_{a,b} \equiv \left\{(\ell, \ell', \ell'') \in [k] \times [k] \times [k] \,|\, \ell < \ell' < \ell'', \ell + \ell' = a, \text{ and } \ell' + \ell'' = b\right\}.$$

We need to prove that (a) there is no missing triple; and (b) no team plays twice in a single round. First, for any ordered triple (ℓ, ℓ', ℓ''), there exists $a \in \{3, \ldots, 2k-3\}$ and $b \in \{5, \ldots, 2k-1\}$ such that $\ell + \ell' = a$ and $\ell' + \ell'' = b$. This proves that all ordered triples are covered by the above construction. Next, given a pair (a, b), no two triples in $T_{a,b}$ can share the same team. Suppose there exists two distinct ordered triples (ℓ, ℓ', ℓ'') and $(\tilde{\ell}, \tilde{\ell}', \tilde{\ell}'')$ both in $T_{a,b}$, and one of the triples are shared. Then, from the two equations $\ell + \ell' = \tilde{\ell} + \tilde{\ell}' = a$ and $\ell' + \ell'' = \tilde{\ell}' + \tilde{\ell}'' = b$, it follows that all three indices must be the same, which is a contradiction. This proves the desired claim for ordered triples.

One caveat is that we wanted to cover the whole $[k] \times [k] \times [k]$, and not just the ordered triples. In the above construction, for example, a triple $(3, 2, 1)$ does not appear. This can be resolved by simply taking all $T_{a,b}$'s from the above construction, and make 6 copies of each round, and permuting all the triples in each copy according to the same permutation over $\{1, 2, 3\}$. This increases the total rounds to $N = 6(2k-5)^2 \le 24k^2$. Note that $|T_{a,b}| \le \lfloor k/3 \rfloor$ since no item can be in more than one triple.

Appendix D. Proof of Estimating Approximate Low-rank Matrices in Corollary 9

We follow closely the proof of a similar corollary in Negahban and Wainwright (2012). First fix a threshold $\tau > 0$, and set $r = \max\{j | \sigma_j(\Theta^*) > \tau\}$. With this choice of r, we have

$$\sum_{j=r+1}^{\min\{d_1,d_2\}} \sigma_j(\Theta^*) \;=\; \tau \sum_{j=r+1}^{\min\{d_1,d_2\}} \frac{\sigma_j(\Theta^*)}{\tau} \;\le\; \tau \sum_{j=r+1}^{\min\{d_1,d_2\}} \Big(\frac{\sigma_j(\Theta^*)}{\tau}\Big)^q \;\le\; \tau^{1-q}\rho_q \,.$$

Also, since $r\tau^q \le \sum_{j=1}^r \sigma_j(\Theta^*)^q \le \rho_q$, it follows that $\sqrt{r} \le \sqrt{\rho_q}\tau^{-q/2}$. Using these bounds, Eq. (35) is now

$$\Big|\Big|\Big|\widehat{\Theta} - \Theta\Big|\Big|\Big|_{\mathrm{F}}^2 \;\le\; \underbrace{288\sqrt{2}c_0 e^{4\alpha} d_1 d_2 \lambda_0}_{=A} \left(\sqrt{\rho_q}\tau^{-q/2}\Big|\Big|\Big|\widehat{\Theta} - \Theta\Big|\Big|\Big|_{\mathrm{F}} + \tau^{1-q}\rho_q\right).$$

With the choice of $\tau = A$ and due to the fact that $x^2 \le bx + c$ implies $x \le (b + \sqrt{b^2 + 4c})/2$ we have,

$$\Big|\Big|\Big|\widehat{\Theta} - \Theta\Big|\Big|\Big|_{\mathrm{F}} \;\le\; 2\sqrt{\rho_q} A^{(2-q)/2} \,.$$

Appendix E. Proof of the Information-theoretic Lower Bound in Theorem 10

The proof uses information-theoretic methods which reduces the estimation problem to a multiway hypothesis testing problem. To prove a lower bound on the expected error, it

suffices to prove that,

$$\sup_{\Theta^* \in \Omega_\alpha} \mathbb{P}\left\{\Big|\Big|\Big|\widehat{\Theta} - \Theta^*\Big|\Big|\Big|_{\mathrm{F}}^2 \ge \frac{\delta^2}{4}\right\} \;\ge\; \frac{1}{2}\,. \tag{157}$$

To prove the above claim, we follow the standard recipe of constructing a packing in Ω_α. Consider a family $\{\Theta^{(1)}, \ldots, \Theta^{(M(\delta))}\}$ of $d_1 \times d_2$ dimensional matrices contained in Ω_α satisfying $\big|\big|\big|\Theta^{(\ell_1)} - \Theta^{(\ell_2)}\big|\big|\big|_{\mathrm{F}} \ge \delta$ for all $\ell_1, \ell_2, \in [M(\delta)]$. We will use M to refer to $M(\delta)$ for simplify the notation. Suppose we draw an index $L \in [M(\delta)]$ uniformly at random, and we are given direct observations σ_i as per MNL model with $\Theta^* = \Theta^{(L)}$ on a randomly chosen set of k items S_i for each user $i \in [d_1]$. It follows from triangular inequality that

$$\sup_{\Theta^* \in \Omega_\alpha} \mathbb{P}\left\{\Big|\Big|\Big|\widehat{\Theta} - \Theta^*\Big|\Big|\Big|_{\mathrm{F}}^2 \ge \frac{\delta^2}{4}\right\} \;\ge\; \mathbb{P}\left\{\widehat{L} \ne L\right\}, \tag{158}$$

where $\widehat{L}$ is the resulting best estimate of the multiway hypothesis testing on L. The generalized Fano's inequality gives

$$\begin{aligned}
\mathbb{P}\left\{\widehat{L} \ne L | S(1), \ldots, S(d_1)\right\} &\ge 1 - \frac{I(\widehat{L}; L) + \log 2}{\log M} &(159)\\
&\ge 1 - \frac{\binom{M}{2}^{-1} \sum_{\ell_1, \ell_2 \in [M]} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) + \log 2}{\log M}\,, &(160)
\end{aligned}$$

where $D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)})$ denotes the Kullback-Leibler divergence between the distributions of the partial rankings $\mathbb{P}\left\{\sigma_1, \ldots, \sigma_{d_1} | \Theta^{(\ell_1)}, S(1), \ldots, S(d_1)\right\}$ and $\mathbb{P}\left\{\sigma_1, \ldots, \sigma_{d_1} | \Theta^{(\ell_2)}, S(1), \ldots, S(d_1)\right\}$. The second inequality follows from a standard technique, which we repeat here for completeness. Let $\Sigma = \{\sigma_1, \ldots, \sigma_{d_1}\}$ denote the observed outcome of comparisons. Since L–$\Theta^{(L)}$–Σ–$\widehat{L}$ form a Markov chain, the data processing inequality gives $I(\widehat{L}; L) \le I(\Sigma; L)$. For simplicity, we drop the conditioning on the set of alternatives $\{S(1), \ldots, S(d_1)\}$, and and let $p(\cdot)$ denotes joint, marginal, and conditional distribution of respective random variables. It follows that

$$\begin{aligned}
I(\Sigma; L) &= \sum_{\ell \in [M], \Sigma} p(\Sigma|\ell) \frac{1}{M} \log \frac{p(\ell, \Sigma)}{p(\ell)p(\Sigma)}\\
&= \frac{1}{M} \sum_{\ell \in [M]} \sum_{\Sigma} p(\Sigma|\ell) \log \frac{p(\Sigma|\ell)}{\frac{1}{M}\sum_{\ell'} p(\Sigma|\ell')}\\
&\le \frac{1}{M^2} \sum_{\ell, \ell' \in [M]} \sum_{\Sigma} p(\Sigma|\ell) \log \frac{p(\Sigma|\ell)}{p(\Sigma|\ell')}\\
&= \frac{1}{M^2} \sum_{\ell, \ell' \in [M]} D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)})\,, &(161)
\end{aligned}$$

where the first inequality follows from Jensen's inequality. To compute the KL-divergence, recall that from the RUM interpretation of the MNL model (see Section 1), one can generate sample rankings Σ by drawing random variables with exponential distributions with mean

$e^{\Theta^*_{ij}}$'s. Precisely, let $X^{(\ell)} = [X^{(\ell)}_{ij}]_{i\in[d_1], j\in S_i}$ denote the set of random variables, where $X^{(\ell)}_{ij}$ is drawn from the exponential distribution with mean $e^{-\Theta^{(\ell)}_{ij}}$. The MNL ranking follows by ordering the alternatives in each S_i according to this $\{X^{(\ell)}_{ij}\}_{j\in S_i}$ by ranking the smaller ones on the top. This forms a Markov chain L–$X^{(L)}$–Σ, and the standard data processing inequality gives

$$\begin{aligned} D_{\mathrm{KL}}(\Theta^{(\ell_1)}\|\Theta^{(\ell_2)}) &\leq D_{\mathrm{KL}}(X^{(\ell_1)}\|X^{(\ell_2)}) && (162)\\ &= \sum_{i\in[d_1]}\sum_{j\in S_i}\Big\{e^{\Theta^{(\ell_1)}_{ij}-\Theta^{(\ell_2)}_{ij}} - (\Theta^{(\ell_1)}_{ij}-\Theta^{(\ell_2)}_{ij}) - 1\Big\} && (163)\\ &\leq \frac{e^{2\alpha}}{4\alpha^2}\sum_{i\in[d_1]}\sum_{j\in S_i}(\Theta^{(\ell_1)}_{ij}-\Theta^{(\ell_2)}_{ij})^2\,, && (164)\end{aligned}$$

where the last inequality follows from the fact that $e^x - x - 1 \leq (e^{2\alpha}/(4\alpha^2))x^2$ for any $x \in [-2\alpha, 2\alpha]$. Taking expectation over the randomly chosen set of alternatives,

$$\mathbb{E}_{S(1),\ldots,S(d_1)}[D_{\mathrm{KL}}(\Theta^{(\ell_1)}\|\Theta^{(\ell_2)})] \;\leq\; \frac{e^{2\alpha}\,k}{4\,\alpha^2\,d_2}\Big|\Big|\Big|\Theta^{(\ell_1)}-\Theta^{(\ell_2)}\Big|\Big|\Big|^2_{\mathrm{F}}\,. \tag{165}$$

Combined with (160), we get that

$$\begin{aligned}\mathbb{P}\Big\{\widehat{L}\neq L\Big\} &= \mathbb{E}_{S(1),\ldots,S(d_1)}[\mathbb{P}\Big\{\widehat{L}\neq L|S(1),\ldots,S(d_1)\Big\}] && (166)\\ &\geq 1 - \frac{\binom{M}{2}^{-1}\sum_{\ell_1,\ell_2\in[M]}(e^{2\alpha}k/(4\alpha^2 d_2))\Big|\Big|\Big|\Theta^{(\ell_1)}-\Theta^{(\ell_2)}\Big|\Big|\Big|^2_{\mathrm{F}} + \log 2}{\log M}\,, && (167)\end{aligned}$$

The remainder of the proof relies on the following probabilistic packing.

Lemma 34 *Let $d_2 \geq d_1 \geq 607$ be positive integers. Then for each $r \in \{1,\ldots,d_1\}$, and for any positive $\delta > 0$ there exists a family of $d_1 \times d_2$ dimensional matrices $\{\Theta^{(1)},\ldots,\Theta^{(M(\delta))}\}$ with cardinality $M(\delta) = \lfloor(1/4)\exp(rd_2/576)\rfloor$ such that each matrix is rank r and the following bounds hold:*

$$\begin{aligned}\Big|\Big|\Big|\Theta^{(\ell)}\Big|\Big|\Big|_{\mathrm{F}} &\leq \delta\,,\ \textit{for all } \ell\in[M] && (168)\\ \Big|\Big|\Big|\Theta^{(\ell_1)}-\Theta^{(\ell_2)}\Big|\Big|\Big|_{\mathrm{F}} &\geq \delta\,,\ \textit{for all } \ell_1,\ell_2\in[M] && (169)\\ \Theta^{(\ell)} &\in \Omega_{\tilde{\alpha}}\,,\ \textit{for all } \ell\in[M]\,, && (170)\end{aligned}$$

with $\tilde{\alpha} = (8\delta/d_2)\sqrt{2\log d}$ for $d = (d_1+d_2)/2$.

We omit the proof of the above lemma since it is similar to that of Lemma 25. Suppose $\delta \leq \alpha d_2/(8\sqrt{2\log d})$ such that the matrices in the packing set are entry-wise bounded by α, then the above lemma implies that $\big|\big|\big|\Theta^{(\ell_1)}-\Theta^{(\ell_2)}\big|\big|\big|^2_{\mathrm{F}} \leq 4\delta^2$, which gives

$$\mathbb{P}\Big\{\widehat{L}\neq L\Big\} \;\geq\; 1 - \frac{\frac{e^{2\alpha}k\delta^2}{\alpha^2 d_2} + \log 2}{\frac{rd}{576} - 2\log 2} \;\geq\; \frac{1}{2}\,,$$

where the last inequality holds for $\delta^2 \leq (\alpha^2 d_2/(e^{2\alpha}k))((rd/1152) - 2\log 2)$. If we assume $rd \geq 3195$ for simplicity, this bound on δ can be simplified to $\delta \leq \alpha e^{-\alpha}\sqrt{r\,d_2\,d/(2304\,k)}$. Together with (157) and (158), this proves that for all $\delta \leq \min\{\alpha d_2/(8\sqrt{2\log d}),$ $\alpha e^{-\alpha}\sqrt{r\,d_2\,d/(2304\,k)}\}$,

$$\inf_{\widehat{\Theta}}\sup_{\Theta^*\in\Omega_\alpha}\mathbb{E}\Big[\Big|\Big|\Big|\widehat{\Theta}-\Theta^*\Big|\Big|\Big|_{\mathrm{F}}\Big] \;\geq\; \frac{\delta}{4}\,.$$

Choosing δ appropriately to maximize the right-hand side finishes the proof of the desired claim.

Appendix F. Proof of Pairwise Rank Breaking in Theorem 11

Analogous to Section C, we define the gradient $\nabla\mathcal{L}(\Theta)$ as $\nabla_{ij}\mathcal{L} = \frac{\partial\mathcal{L}(\Theta)}{\partial\Theta_{ij}}$ and $\Delta \equiv \hat{\Theta} - \Theta^*$, and provide two main technical lemmas.

Lemma 35 *If $\lambda \geq 2|||\nabla\mathcal{L}(\Theta^*)|||_2$, then we have,*

$$|||\Delta|||_{\mathrm{nuc}} \leq 4\sqrt{2r}|||\Delta|||_{\mathrm{F}} + 4\sum_{j=\rho+1}^{\min\{d_1,d_2\}}\sigma_j(\Theta^*)\,, \tag{171}$$

for all $\rho \in [\min\{d_1,d_2\}]$.

Proof This follows from the proof of Lemma 27, which only depends on the convexity of $\mathcal{L}(\Theta)$. ∎

Lemma 36 *For any positive constant $c \geq 1$, if $k \leq \max\{d_1, d_2^2/d_1\}\log d$ and $d_1 \geq 4$ then with probability at least $1 - 2d^{-c}$,*

$$|||\nabla\mathcal{L}(\Theta^*)|||_2 \;\leq\; \sqrt{\frac{16(c+4)\log d}{k\,d_1^2}}\max\left\{\sqrt{\max\left\{\frac{1}{4},\frac{d_1}{d_2}\right\}}\,,\ \frac{2}{3}\sqrt{\frac{2(c+4)\log d}{k}}\right\}\,. \tag{172}$$

The proof of this lemma is provided in Section F.1. We will simplify the above lemma by assuming, $2(c+4)\log d \leq k$ which implies the last term in RHS is less than equal to first term,

$$\frac{2}{3}\sqrt{\frac{2(4+c)\log d}{k}} \leq \sqrt{\frac{1}{4}}\,. \tag{173}$$

(173) simplifies (172) as,

$$\begin{aligned}|||\nabla\mathcal{L}(\Theta^*)|||_2 &\leq \sqrt{\frac{16(c+4)\log d}{k\,d_1^2}\max\left\{\frac{1}{4},\frac{d_1}{d_2}\right\}}\\ &\leq \sqrt{\frac{32d\,(c+4)\log d}{k\,d_1^2\,d_2}}\\ &\overset{(a)}{\leq} \sqrt{32(c+4)}\lambda\,, && (174)\end{aligned}$$

where (a) is due to (41) .

For Lemma 35 and further proof of Theorem11 we want $\lambda \geq 2|||\nabla \mathcal{L}(\Theta)|||_2$, therefore we assume that,

$$\lambda \in [2\sqrt{32(c+4)}\lambda, c_p\lambda], \text{ for some } c_p \geq 2\sqrt{32(c+4)} \tag{175}$$

Similar to the k-wise ranking,we will divide the proof into two cases and each part we will prove that $\|\Delta\|_{\mathrm{F}}^2 \leq 36e^{2\alpha}\, c\, \lambda\, d_1 d_2\, |||\Delta|||_{\text{nuc}}$ with probability at least $1 - 2/d^c - 2/d^{2^{13}}$. We define a new constant μ as,

$$\mu = 16\alpha\sqrt{\frac{48\, d_1 d_2^2 \log d}{k\, \min\{d_1, d_2\}}}\,. \tag{176}$$

Case 1: Assume $\mu|||\Delta|||_{\text{nuc}} \leq \|\Delta\|_{\mathrm{F}}^2$.
Since $\mathcal{L}$ is a sum of a linear function of Θ and log-sum-exponential functions, which are convex, we know that $\mathcal{L}$ is a convex function of Θ. Therefore, by convexity and Taylor expansion we get,

$$\begin{aligned}\mathcal{L}(\hat{\Theta}) = \mathcal{L}(\Theta^*) - \langle\!\langle \nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle \; + \\ \frac{1}{2!\; d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e^{\Theta_{i,u_{i,m_1}}} e^{\Theta_{i,u_{i,m_2}}}}{\left(e^{\Theta_{i,u_{i,m_1}}} + e^{\Theta_{i,u_{i,m_2}}}\right)^2} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2 ,\end{aligned} \tag{177}$$

where $\Theta = a\Theta^* + (1-a)\hat{\Theta}$ for some $a \in [0,1]$ and $\mathcal{P}_0 = \{(i,j)|\;\; 1 \leq i < j \leq k\}$. We lower bound the final term in (177) as,

$$\begin{aligned}&\frac{1}{2!\; d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e^{\Theta_{i,u_{i,m_1}}} e^{\Theta_{i,u_{i,m_2}}}}{\left(e^{\Theta_{i,u_{i,m_1}}} + e^{\Theta_{i,u_{i,m_2}}}\right)^2} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2 \\ &\overset{(a)}{\geq} \frac{1}{2\; d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e^{-\alpha}e^{\alpha}}{(e^{-\alpha}+e^{\alpha})^2} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2 \\ &\geq \frac{1}{2\; d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e^{-2\alpha}}{4} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2 ,\end{aligned} \tag{178}$$

where (a) is due to the fact that Δ_{ij}'s are upper and lower bounded by α and $-\alpha$ respectively. We can bound this term further according to the following Lemma.

Lemma 37 *For $(4\log d)/9 \leq k \leq \max\{d_1, d_2^2/d_1\}\log d$, with probability at least $1 - 2d^{-2^{13}}$,*

$$\frac{1}{d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2 \geq \frac{1}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2 \,, \tag{179}$$

for all $\Delta \in \mathcal{A}_p$ where,

$$\mathcal{A} = \left\{ \Delta \in \mathbb{R}^{d_1\times d_2} \;\middle|\; |||\Delta|||_\infty \leq 2\alpha,\; \sum_{j\in[d_2]} \Delta_{ij} = 0 \text{ , for all } i \in [d_2],\; \text{and } \mu|||\Delta|||_{\text{nuc}} \leq \|\Delta\|_{\mathrm{F}}^2 \right\}. \tag{180}$$

The proof is given in Section F.2. Now using Lemma 37 and (178) with high probability we get,

$$\frac{1}{2!\; d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e^{\Theta_{i,u_{i,m_1}}} e^{\Theta_{i,u_{i,m_2}}}}{\left(e^{\Theta_{i,u_{i,m_1}}} + e^{\Theta_{i,u_{i,m_2}}}\right)^2} \left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_1}}\right)^2 \geq \frac{e^{-2\alpha}}{24\; d_1\; d_2}|||\Delta|||_{\mathrm{F}}^2 \,. \tag{181}$$

Incorporating the above inequality in (177) we obtain,

$$\frac{e^{-2\alpha}}{24\; d_1\; d_2}|||\Delta|||_{\mathrm{F}}^2 \leq \mathcal{L}(\hat{\Theta}) - \mathcal{L}(\Theta^*) + \langle\!\langle \nabla\mathcal{L}(\Theta^*), \Delta\rangle\!\rangle \,. \tag{182}$$

From the definition of $\hat{\Theta}$ we have $\mathcal{L}(\hat{\Theta}) - \mathcal{L}(\Theta^*) \leq \lambda\left(|||\Theta^*|||_{\text{nuc}} - \left|||\hat{\Theta}\right|||_{\text{nuc}}\right) \leq \lambda|||\Delta|||_{\text{nuc}}$, and we assume that $\lambda \geq 2\sqrt{32(c+1)}\;\lambda$, so that $\lambda \geq 2|||\nabla\mathcal{L}\,(\Theta^*)|||_2$ is true with a probability of at least $1 - 2d^{-c}$ from Lemma 36 . These give us the following with at least probability $1 - 2d^{-c} - 2d^{-2^{13}}$.

$$\begin{aligned}\frac{e^{-2\alpha}}{24\; d_1\; d_2}|||\Delta|||_{\mathrm{F}}^2 &\leq \lambda|||\Delta|||_{\text{nuc}} + |||\nabla\mathcal{L}(\Theta^*)|||_2|||\Delta|||_{\text{nuc}} \\ &\leq \frac{3\lambda}{2}|||\Delta|||_{\text{nuc}}\end{aligned} \tag{183}$$

which gives us,

$$\begin{aligned}|||\Delta|||_{\mathrm{F}}^2 &\leq 36e^{2\alpha}\;\lambda\; d_1d_2\;|||\Delta|||_{\text{nuc}} \\ &\overset{(a)}{\leq} 36e^{2\alpha}\; c_p\;\lambda\; d_1d_2\;|||\Delta|||_{\text{nuc}} \,,\end{aligned} \tag{184}$$

where (a) is due to the fact that $\lambda \leq c_p\lambda$.

Case 2: Assume $|||\Delta|||_{\mathrm{F}}^2 \leq \mu|||\Delta|||_{\text{nuc}}$.
Here we prove that $\mu \leq 36\; e^{2\alpha}\; c_p\lambda\; d_1d_2$.

$$\begin{aligned}\frac{\mu}{36\; e^{2\alpha}\; c_p\lambda\; d_1d_2} &\overset{(a)}{\leq} \frac{\alpha}{e^{2\alpha}} \times \frac{16\sqrt{48}}{72\sqrt{32(c+4)}} \times \sqrt{\frac{d_1d_2}{min\{d_1,d_2\}d}} \\ &\overset{(b)}{\leq} 1 \times \frac{16\sqrt{48}}{72\sqrt{32\times 4}} \times \sqrt{\frac{\max\{d_1,d_2\}}{d}} \\ &\overset{(c)}{\leq} \sqrt{\frac{\max\{d_1,d_2\}}{2d}} \\ &\overset{(d)}{\leq} 1 \,,\end{aligned} \tag{185}$$

where (a) is by substituting μ, λ and c_p from (176), (41) and (175) respectively, (b) is because $x \leq e^x$ (c) is because $d = (\max\{d_1,d_2\} + \min\{d_1,d_2\})/2$.

Now combining the above result with (171) we get with probability at least $1 - 2d^{-c} - 2d^{-2^{13}}$,

$$\frac{1}{d_1d_2}|||\Delta|||_{\mathrm{F}}^2 \leq 144\sqrt{2}e^{2\alpha}c_p\lambda\sqrt{r}|||\Delta|||_{\mathrm{F}} + 144e^{2\alpha}c_p\lambda \sum_{j=\rho+1}^{\min\{d_1,d_2\}} \sigma_j(\Theta^*) \,. \tag{186}$$

F.1 Proof of Lemma 36

From definition of $\mathcal{L}(\Theta)$ in (39) we get,

$$\nabla \mathcal{L}_p(\Theta^*) = \frac{1}{d_1\binom{k}{2}} \sum_{i=1}^{d_1} \sum_{(m_1,m_2)\in\mathcal{P}_0} \frac{e_i \left(e_{l_i(m_1,m_2)} - e_{h_i(m_1,m_2)}\right)^T}{\exp\left(\Theta^*_{i,h_i(m_1,m_2)} - \Theta^*_{i,l_i(m_1,m_2)}\right) + 1} , \tag{187}$$

where $\mathcal{P}_0 = \{(i,j)|\ \ 1 \le i < j \le k\}$. We use the matrix Bernstein inequality Tropp (2011) for the sum of independent matrices. Similar to Lemma 42, we can partition the set of all pairs $\mathcal{P}_0$ into $(k-1)$ sets $\{\mathcal{P}_a\}_{a\in[k-1]}$ of $k/2$ disjoint pairs each. Define $Y_a \equiv \sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a} \tilde{X}_{i,m_1,m_2}$, and

$$\tilde{X}_{i,m_1,m_2} \ \equiv \ \frac{\exp\left(\Theta^*_{i,l_i(m_1,m_2)}\right)}{\exp\left(\Theta^*_{i,h_i(m_1,m_2)}\right) + \exp\left(\Theta^*_{i,l_i(m_1,m_2)}\right)} e_i \left(e_{l_i(m_1,m_2)} - e_{h_i(m_1,m_2)}\right)^T ,$$

such that

$$\nabla \mathcal{L}_p(\Theta^*) = \frac{1}{d_1\binom{k}{2}} \sum_{a=1}^{k-1} \tilde{Y}_a . \tag{188}$$

For a fixed value of a, it is easy to see that $\tilde{X}_{i,m_1,m_2}$'s are independent. Further, we can easily show that $\mathbb{E}\left[\tilde{X}_{i,m_1,m_2}\right] = 0$, and $\|\tilde{X}_{i,m_1,m_2}\|_2 \le \sqrt{2}$. We also have,

$$\begin{aligned}
&\mathbb{E}\left[\tilde{X}_{i,m_1,m_2}\tilde{X}^T_{i,m_1,m_2}\right] \\
&\preceq 2\, e_i e_i^T\, \mathbb{E}\left[\mathbb{E}\left[\frac{\exp\left(\Theta^*_{i,l_i(m_1,m_2)}\right)^2}{\left(\exp\left(\Theta^*_{i,u_{i,m_1}}\right) + \exp\left(\Theta^*_{i,u_{i,m_2}}\right)\right)^2}\,\middle|\, u_{i,m_1}, u_{i,m_1}\right]\right] \\
&\overset{(a)}{=} 2\, e_i e_i^T\, \mathbb{E}\left[\frac{\exp\left(\Theta^*_{iu_{i,m_1}}\right)\exp\left(\Theta^*_{iu_{i,m_2}}\right)}{\left(\exp\left(\Theta^*_{i,u_{i,m_1}}\right) + \exp\left(\Theta^*_{i,u_{i,m_2}}\right)\right)^2}\right] \\
&\overset{(b)}{\preceq} \frac{1}{2} e_i e_i^T ,
\end{aligned} \tag{189}$$

where we get (a) from the MNL model for the random choice of $l_i(m_1,m_2)$, (b) is due to the fact that $xy/(x+y)^2 \le 1/4$ for all $x, y > 0$.

Let $p_{i,m_1,m_2} \equiv \frac{\left(\exp(\Theta^*_{i,u_{i,m_1}})e_{u_{i,m_1}} + \exp(\Theta^*_{i,u_{i,m_2}})e_{u_{i,m_2}}\right)}{\left(\exp(\Theta^*_{i,u_{i,m_1}}) + \exp(\Theta^*_{i,u_{i,m_2}})\right)}$, then we have,

$$\begin{aligned}
\mathbb{E}\left[\tilde{X}^T_{i,m_1,m_2}\tilde{X}_{i,m_1,m_2}\right] &= \mathbb{E}\left[(e_{h_i(m_1,m_2)} - p_{i,m_1,m_2})(e_{h_i(m_1,m_2)} - p_{i,m_1,m_2})^T\right] \\
&= \mathbb{E}\left[e_{h_i(m_1,m_2)}e^T_{h_i(m_1,m_2)}\right] - \mathbb{E}\left[p_{i,m_1,m_2}p^T_{i,m_1,m_2}\right] \\
&\overset{(a)}{\preceq} \mathbb{E}\left[e_{u_{i,m_1}}e^T_{u_{i,m_1}} + e_{u_{i,m_2}}e^T_{u_{i,m_2}}\right] \\
&= \frac{2}{d_2}\mathbf{I}_{d_2\times d_2} ,
\end{aligned} \tag{190}$$

where (a) comes from the fact that $p_{i,m_1,m_2}p^T_{i,m_1,m_2}$ is a positive semi-definite matrix. Therefore using (189) and (190), we get

$$\begin{aligned}
\sigma^2 &\equiv \left\{ \left\| \sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}} \mathbb{E}\left[\tilde{X}_{i,m_1,m_2}\tilde{X}^T_{i,m_1,m_2}\right] \right\|_2 , \left\| \sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}} \mathbb{E}\left[\tilde{X}^T_{i,m_1,m_2}\tilde{X}_{i,m_1,m_2}\right] \right\|_2 \right\} \\
&\le k \max\left\{\frac{1}{4}, \frac{d_1}{d_2}\right\} .
\end{aligned} \tag{191}$$

Define $\rho \equiv \max\{1/4, d_1/d_2\}$, then by the matrix Bernstein inequality Tropp (2011), $\forall\ a \in [k-1]$,

$$\mathbb{P}\Big(\left\|\tilde{Y}_a\right\|_2 > t\Big) \le (d_1+d_2)\exp\left(\frac{-t^2/2}{k\rho + \sqrt{2}t/3}\right) ,$$

which gives a tail probability of $2d^{-c}/(k-1)$ for the choice of

$$t = \max\left\{\sqrt{4k\rho\left((1+c)\log d + \log(k-1)\right)}\,,\ \frac{4\sqrt{2}((1+c)\log d + \log(k-1))}{3}\right\} . \tag{192}$$

For this choice of t, using union bound we can get the probabilistic bound on the derivative of log likelihood as,

$$\begin{aligned}
\mathbb{P}\left(\|\nabla\mathcal{L}_p(\Theta^*)\|_2 \ge \frac{k-1}{d_1\binom{k}{2}}t\right) &\le \mathbb{P}\left(\sum_{a=1}^{k-1}\left\|\tilde{Y}_a\right\|_2 \ge (k-1)t\right) \\
&\overset{(a)}{\le} \mathbb{P}\left(\max_{a\in[k-1]}\left\|\tilde{Y}_a\right\|_2 \ge t\right) \\
&\overset{(b)}{\le} \sum_{a=1}^{k-1}\mathbb{P}\Big(\left\|\tilde{Y}_a\right\|_2 \ge t\Big) \\
&= 2\,d^{-c} ,
\end{aligned} \tag{193}$$

where we obtain (a) by pigeon-hole principle which implies that among a set of numbers, there should be, at the very least one number greater or equal to the average of the set of numbers and (b) by union-bound. Assuming $k \le \max\{d_1, d_2^2/d_1\}\log d$ and $d_1 \ge 4$, we have,

$$(c+1)\log d + \log(k-1) \le (c+4)\log d\,, \tag{194}$$

from $\log(k-1) \le \log\left(\max\{d_1, d_2^2/d_1\}\log d\right) \le \log(\left((d_1^2+d_2^2)\log d\right)/d_1) \le \log\left((4\,d^2\log d)/d_1\right)$ $\le 3\log d$. This proves the desired lemma.

F.2 Proof of Lemma 37

With a slight abuse of notation, we define $\tilde{H}$ as

$$\tilde{H}(\Delta) \;\equiv\; \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_0}\left(\Delta_{i,u_{i,m_1}} - \Delta_{i,u_{i,m_2}}\right)^2\,, \tag{195}$$

and provide a lower bound. The mean is easily computed as

$$\begin{aligned}\mathbb{E}\left[\tilde{H}(\Delta)\right] &= \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_0}\left[\frac{2}{d_2}\sum_{j\in[d_2]}\Delta_{ij}^2 - \frac{2}{d_2^2}\sum_{j\in[d_2]}\Delta_{ij}\sum_{j'\in[d_2]}\Delta_{ij'}\right]\\ &= \frac{2}{d_1 d_2}|||\Delta|||_{\mathrm{F}}^2\,,\end{aligned} \tag{196}$$

where we used the fact that $\sum_j \Delta_{ij} = 0$. We want to upper bound the probability that $\tilde{H}(\Delta) \le \frac{1}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2$ for some $\Delta \in \mathcal{A}$. As in the case of k-wise ranking we using the following peeling argument used in (Negahban and Wainwright, 2012, Lemma 3), Van De Geer (2000). The strategy is to split this above event as union of many event events as follows. We construct the following family of subsets $\{\tilde{\mathcal{S}}_\ell\}$ such that $\mathcal{A} \subseteq \cup_{\ell=1}^{\infty}\tilde{\mathcal{S}}_\ell$ and,

$$\begin{aligned}\tilde{\mathcal{S}}_\ell = \Bigg\{\Delta \in \mathbb{R}^{d_1\times d_2} \;\Big|\; |||\Delta|||_\infty \le 2\alpha,\; \beta^{\ell-1}\mu \le |||\Delta|||_{\mathrm{F}} \le \beta^\ell\mu,\\ \sum_{j\in[d_2]}\Delta_{ij} = 0 \text{ for all } i\in[d_2],\text{ and } |||\Delta|||_{\mathrm{nuc}} \le \beta^{2\ell}\mu\Bigg\}\,,\end{aligned} \tag{197}$$

where $\beta = \sqrt{10/9}$ and $\ell \in \{1,2,3,\ldots\}$. This is true since, for any $\Delta \in \mathcal{A}$, $|||\Delta|||_{\mathrm{F}}^2 \ge \mu|||\Delta|||_{\mathrm{nuc}}$ and this implies $|||\Delta|||_{\mathrm{F}}^2 \ge \mu|||\Delta|||_{\mathrm{F}}$ (or, $|||\Delta|||_{\mathrm{F}} \ge \mu$). Also note that,

$$\begin{aligned}\tilde{H}(\Delta) \le \frac{1}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2 &\implies \frac{2}{d_1d_2}|||\Delta|||_{\mathrm{F}}^2 - \tilde{H}(\Delta) \ge \frac{5}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2\\ &\implies \left(\mathbb{E}\left[\tilde{H}(\Delta)\right] - \tilde{H}(\Delta)\right) \ge \frac{5}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2\,.\end{aligned} \tag{198}$$

Therefore using union bound we get,

$$\begin{aligned}&\mathbb{P}\Big(\exists\,\Delta\in\mathcal{A}\ s.t.\ \tilde{H}(\Delta) \le \frac{1}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2\Big)\\ &\le \sum_{\ell=1}^{\infty}\mathbb{P}\left(\sup_{\Delta\in\tilde{\mathcal{S}}_\ell}\left(\mathbb{E}\left[\tilde{H}(\Delta)\right] - \tilde{H}(\Delta)\right) \ge \frac{5}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2\right)\\ &\overset{(a)}{\le} \sum_{\ell=1}^{\infty}\mathbb{P}\left(\sup_{\Delta\in\tilde{\mathcal{S}}_\ell}\left(\mathbb{E}\left[\tilde{H}(\Delta)\right] - \tilde{H}(\Delta)\right) \ge \frac{3}{2d_1d_2}(\beta^\ell\mu)^2\right)\\ &\overset{(b)}{\le} \sum_{\ell=1}^{\infty}\mathbb{P}\left(\sup_{\Delta\in\tilde{\mathcal{B}}(\beta^\ell\mu)}\left(\mathbb{E}\left[\tilde{H}(\Delta)\right] - \tilde{H}(\Delta)\right) \ge \frac{3}{2d_1d_2}(\beta^\ell\mu)^2\right)\,,\end{aligned} \tag{199}$$

where $\tilde{\mathcal{B}}(\mathcal{D})$ is defined as,

$$\begin{aligned}\tilde{\mathcal{B}}(D) = \Bigg\{\Delta\in\mathbb{R}^{d_1\times d_2} \;\Big|\; |||\Delta|||_\infty \le 2\alpha,\ |||\Delta|||_{\mathrm{F}} \le D,\\ \sum_{j\in[d_2]}\Delta_{ij} = 0 \text{ for all } i\in[d_2],\text{ and } \mu|||\Delta|||_{\mathrm{nuc}} \le D^2\Bigg\}\,,\end{aligned} \tag{200}$$

and $(a)$is true because for $\Delta \in \tilde{\mathcal{S}}_l$,

$$\frac{5}{3d_1d_2}|||\Delta|||_{\mathrm{F}}^2 \ge \frac{5}{3d_1d_2}(\beta^{\ell-1}\mu)^2 = \frac{3}{2d_1d_2}(\beta^\ell\mu)^2\,, \tag{201}$$

and (b) is true because $\tilde{\mathcal{S}}_\ell \subset \tilde{\mathcal{B}}(\beta^\ell\mu)$. Now we use following lemma to upper bound (199).

Lemma 38 *For* $4(\log d)/3 \le k \le d^2\log d$,

$$\mathbb{P}\left(\sup_{\Delta\in\tilde{\mathcal{B}}(D)}\left(\mathbb{E}\left[\tilde{H}(\Delta)\right] - \tilde{H}(\Delta)\right) \ge \frac{3}{2d_1d_2}D^2\right) \le \exp\left(\frac{-kD^4}{2048\ \alpha^4\ d_1d_2^2}\right) \tag{202}$$

Proof has been relegated to Section F.3. Now by (199) and Lemma 38 we get,

$$\begin{aligned}
\mathbb{P}\Big(\exists\, \Delta \in \mathcal{A} \text{ s.t. } \tilde{H}(\Delta) \le \frac{1}{3d_1d_2}\|\Delta\|_{\mathrm{F}}^2\Big) &\le \sum_{\ell=1}^{\infty} \exp\left(\frac{-k\left(\beta^\ell\,\mu\right)^4}{2048\;\alpha^4\;d_1d_2^2}\right)\\
&\overset{(a)}{\le} \sum_{\ell=1}^{\infty} \exp\left(\frac{-2^{13}\;9\;\beta^{4\ell}\,d_1d_2^2\log^2 d}{k\;\min^2\{d_1,d_2\}}\right)\\
&\overset{(b)}{\le} \sum_{\ell=1}^{\infty} \exp\left(\frac{-2^{13}\;9\;4\ell\times\frac{1}{36}\,d_1d_2^2\log^2 d}{k\;\min^2\{d_1,d_2\}}\right)\\
&\overset{(c)}{\le} \sum_{\ell=1}^{\infty} \exp\left(-2^{13}\;\ell\;\log d\right)\\
&= \sum_{\ell=1}^{\infty}\left(\frac{1}{d^{2^{13}}}\right)^{\ell}\\
&\overset{(d)}{=} \frac{1/d^{2^{13}}}{1-1/d^{2^{13}}}\\
&\overset{(e)}{\le} \frac{2}{d^{2^{13}}}\,,
\end{aligned} \tag{203}$$

where we get (a) by substituting μ from (176), (b) by the fact that for $\beta=\sqrt{10/9}$ and $x\ge 1$, $\beta^x \ge x\log\beta \ge x(\beta-1)\ge x/32$, (c) is obtained by assuming $k\le \max\{d_1,d_2^2/d_1\}\log d$, we get (d) because we are summing an infinite geometric sequence with common ratio of $1/d^{2^{13}}$ and (e) is because for $d\ge 2$, $1/d^{2^{13}}$ is less than $1/2$.

F.3 Proof of Lemma 38

With a slight abuse of notations, let $\tilde{Z} \equiv \sup_{\Delta\in\tilde{\mathcal{B}}(D)}\Big(\mathbb{E}\left[\tilde{H}(\Delta)\right]-\tilde{H}(\Delta)\Big)$. Notice that $\tilde{Z}$ is a function of d_1k random variables, $\{u_{i,\ell}\}_{i\in[d_1],\ell\in[k]}$. We apply the McDiarmid's bounded differences inequality. Let $\tilde{Z}_1$ and $\tilde{Z}_2$ be two realizations of $\tilde{Z}$ where value of only one random variable $u_{i',\ell'}$ is changed to $u'_{i',\ell'}$. Also with a little more abuse of notation the two realizations of $\tilde{H}(\Delta)$ are written as $\tilde{H}(\Delta', u_{1,1},\ldots,u_{i',\ell'},\ldots,u_{d_1,k})$ and $\tilde{H}(\Delta', u_{1,1},\ldots,u'_{i',\ell'},\ldots,u_{d_1,k})$. We let Δ^* be the maximizer of $\max\{\tilde{Z}_1,\tilde{Z}_2\}$. Maximum

absolute difference between them is upper bounded as follows,

$$\begin{aligned}
&|\tilde{Z}_1-\tilde{Z}_2|\\
&=\Bigg|\max_{\Delta\in\tilde{\mathcal{B}}(D)}\Big(\mathbb{E}\left[\tilde{H}(\Delta)\right]-\tilde{H}(\Delta,u_{1,1},\ldots,u_{i',\ell'},\ldots,u_{d_1,k})\Big)-\\
&\qquad\max_{\Delta'\in\tilde{\mathcal{B}}(D)}\Big(\mathbb{E}\left[\tilde{H}(\Delta')\right]-\tilde{H}(\Delta',u_{1,1},\ldots,u'_{i',\ell'},\ldots,u_{d_1,k})\Big)\Bigg|\\
&\overset{(a)}{\le}\Bigg|\Big(\mathbb{E}\left[\tilde{H}(\Delta^*)\right]-\tilde{H}(\Delta^*,u_{1,1},\ldots,u_{i',\ell'},\ldots,u_{d_1,k})\Big)-\\
&\qquad\Big(\mathbb{E}\left[\tilde{H}(\Delta^*)\right]-\tilde{H}(\Delta^*,u_{1,1},\ldots,u'_{i',\ell'},\ldots,u_{d_1,k})\Big)\Bigg|\\
&\le \sup_{\Delta\in\tilde{\mathcal{B}}(D)}\Big|\tilde{H}(\Delta,u_{1,1},\ldots,u_{i',\ell'},\ldots,u_{d_1,k})-\tilde{H}(\Delta,u_{1,1},\ldots,u'_{i',\ell'},\ldots,u_{d_1,k})\Big|\\
&\overset{(b)}{\le}\sup_{\Delta\in\tilde{\mathcal{B}}(D)}\Bigg|\frac{1}{d_1\binom{k}{2}}\sum_{\ell\neq\ell'}\Big(\Delta_{i',u_{i',\ell}}-\Delta_{i',u_{i',\ell'}}\Big)^2-\Big(\Delta_{i',u_{i',\ell}}-\Delta_{i',u'_{i',\ell'}}\Big)^2\Bigg|\\
&\overset{(c)}{\le}\frac{1}{d_1\binom{k}{2}}(k-1)\,(4\alpha)^2=\frac{32\alpha^2}{d_1k}.
\end{aligned} \tag{204}$$

where (a) follows from the fact that Δ^* is maximizer of $\max\{\tilde{Z}_1,\tilde{Z}_2\}$, (b) is due to the fact that the terms which change because of $u'_{i',\ell'}$ are the $k-1$ difference square terms between $\Delta_{iu_{i',\ell\neq\ell'}}$ and $\Delta_{i,\,u_{i',\ell'}}$ and (c) is because maximum and minimum value of difference square terms are $(4\alpha)^2$ and 0 respectively. Using McDiarmid's bounded differences inequality we get,

$$\mathbb{P}\{\tilde{Z}-\mathbb{E}\left[\tilde{Z}\right]\ge\epsilon\}\le\exp\left(-\frac{2\epsilon^2}{d_1k\left(\frac{32\alpha^2}{d_1k}\right)^2}\right)\,, \tag{205}$$

because of (204) and the fact that there are $d_1 k$ random variables. We upper bound $\mathbb{E}\left[\tilde{Z}\right]$ as follows.

$$\begin{aligned}
\mathbb{E}\left[\tilde{Z}\right] &= \mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_0}\mathbb{E}\left[\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2\right]-\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2\\
&\overset{(a)}{\le} \mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_0} 2\tilde{\xi}_{i,m_1,m_2}\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2\\
&\overset{(b)}{\le} \mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{a=1}^{k-1}\sum_{(m_1,m_2)\in\mathcal{P}_a} 2\tilde{\xi}_{i,m_1,m_2}\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2\\
&\overset{(c)}{\le} \sum_{a=1}^{k-1}\mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a} 2\tilde{\xi}_{i,m_1,m_2}\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2 ,
\end{aligned} \tag{206}$$

where (a) is by standard symmetrization technique as used in k-wise ranking and $\{\xi_{i,m_1,m_2}\}_{i\in[d_1],\, m_1,m_2\in[k]}$ are i.i.d. Rademacher variables, (b) is due to the fact that we can partition set of all pairs into $k-1$ independent sets as in (188) and (c) is because of fact that supremum of sum is less than or equal to sum of supremum and the linearity of expectation. Since $|\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}|\le 4\alpha$, we can use Ledoux-Talagrand contraction inequality Ledoux and Talagrand (2013) on (206) to get,

$$\begin{aligned}
E[\tilde{Z}] &\le \sum_{a=1}^{k-1}\mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a} 2\tilde{\xi}_{i,m_1,m_2}\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)^2\\
&\le \sum_{a=1}^{k-1}\mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)} \frac{1}{d_1\binom{k}{2}}\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a} 4\alpha\, 2\tilde{\xi}_{i,m_1,m_2}\left(\Delta_{i,\,u_{i,m_1}}-\Delta_{i,\,u_{i,m_2}}\right)\\
&\overset{(a)}{\le} \sum_{a=1}^{k-1}\frac{8\alpha}{d_1\binom{k}{2}}\mathbb{E}\sup_{\Delta\in\tilde{\mathcal{B}}(D)}\langle\!\langle\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a}\tilde{W}_{i,m_1,m_2},\Delta\rangle\!\rangle\\
&\overset{(b)}{\le} \sum_{a=1}^{k-1}\frac{8\alpha}{d_1\binom{k}{2}}\mathbb{E}\left[\left|\!\left|\!\left|\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a}\tilde{W}_{i,m_1,m_2}\right|\!\right|\!\right|_2\right]\sup_{\Delta\in\tilde{\mathcal{B}}(D)}|\!|\!|\Delta|\!|\!|_{\text{nuc}} ,
\end{aligned} \tag{207}$$

where we get (a) by putting $\tilde{W}_{i,m_1,m_2}=\tilde{\xi}_{i,m_1,m_2}e_i(e_{u_{i,m_1}}-e_{u_{i,m_2}})^T$ and (b) is due to Hölder's inequality ($\langle\!\langle x,y\rangle\!\rangle\le|\!|\!|x|\!|\!|_2|\!|\!|y|\!|\!|_{\text{nuc}}$). Now we use Bernstein's inequality Tropp (2011) to upperbound the above expectation terms. First fix a to value in $[k-1]$. We can easily show that $\tilde{W}_{i,m_1,m_2}$ is zero mean and,

$$\left|\!\left|\!\left|\tilde{W}_{i,m_1,m_2}\right|\!\right|\!\right|_2\le\sqrt{2}\,. \tag{208}$$

We also get,

$$\begin{aligned}
\mathbb{E}\left[\tilde{W}_{i,m_1,m_2}\tilde{W}_{i,m_1,m_2}^T\right] &= 2e_ie_i^T\mathbb{E}\left[1-e_{u_{i,m_1}}^Te_{u_{i,m_2}}\right]\\
&\preceq e_ie_i^T\left(2-\frac{2}{d_2}\right)\\
&\preceq 2e_ie_i^T ,
\end{aligned} \tag{209}$$

and,

$$\begin{aligned}
\mathbb{E}\left[\tilde{W}_{i,m_1,m_2}^T\tilde{W}_{i,m_1,m_2}\right] &= \mathbb{E}\left[2e_{u_{i,m_1}}e_{u_{i,m_1}}^T-2e_{u_{i,m_1}}e_{u_{i,m_2}}^T\right]\\
&\preceq \frac{2}{d_2}\mathbf{I}_{d_2\times d_2}-\frac{2}{d_2^2}\mathbf{1}\mathbf{1}_{d_2\times d_2}\\
&\preceq \frac{2}{d_2}\mathbf{I}_{d_2\times d_2}\,.
\end{aligned} \tag{210}$$

Therefore, using (209) and (210), the standard deviation of $\sum_{(i,m_1,m_2)}Z_{i,m_2,m_2}$ is,

$$\begin{aligned}
\sigma^2 &= \max\left\{\left|\!\left|\!\left|\sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}}\mathbb{E}\left[\tilde{W}_{i,m_2,m_2}\tilde{W}_{i,m_2,m_2}^T\right]\right|\!\right|\!\right|_2,\left|\!\left|\!\left|\sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}}\mathbb{E}\left[\tilde{W}_{i,m_2,m_2}^T\tilde{W}_{i,m_2,m_2}\right]\right|\!\right|\!\right|_2\right\}\\
&\le \max\left\{\frac{d_1k}{2}\frac{2}{d_1}|\!|\!|\mathbf{I}|\!|\!|_2,\frac{d_1k}{2}\frac{2}{d_2}|\!|\!|\mathbf{I}|\!|\!|_2\right\}\\
&= \frac{kd_1}{\min\{d_1,d_2\}}\,.
\end{aligned} \tag{211}$$

By matrix Bernstein inequality Tropp (2011), $\forall\ a\in[k-1]$,

$$\mathbb{P}\left(\left|\!\left|\!\left|\sum_{i\in[d_1]}\sum_{(m_1,m_2)\in\mathcal{P}_a}\tilde{W}_{i,m_2,m_2}\right|\!\right|\!\right|_2>t\right)\le(d_1+d_2)\exp\left(\frac{-t^2/2}{2kd_1/\min\{d_1,d_2\}+\sqrt{2}t/3}\right),$$

which gives a tail probability of $2d^{-c_1}$ for the choice of

$$\begin{aligned}
t &= \max\left\{\sqrt{\frac{8kd_1\left((1+c_1)\log d\right)}{\min\{d_1,d_2\}}},\frac{4\sqrt{2}\left((1+c_1)\log d\right)}{3}\right\}\\
&= \sqrt{\frac{8kd_1\left((1+c_1)\log d\right)}{\min\{d_1,d_2\}}},\text{when } k\ge 4(c_1+1)\log d/9\,.
\end{aligned} \tag{212}$$

Therefore $\forall\ a\in[k-1]$,

$$\mathbb{E}\left[\left|\!\left|\!\left|\sum_{i=1}^{d_1}\sum_{(m_1,m_2)\in\mathcal{P}_a}\tilde{W}_{i,m_2,m_2}\right|\!\right|\!\right|_2\right]\le\sqrt{\frac{8kd_1\left((1+c_1)\log d\right)}{\min\{d_1,d_2\}}}+\frac{2}{d^{c_1}}\frac{\sqrt{2}d_1k}{2}\,, \tag{213}$$

because from (208) we get $\left\|\left\|\sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}}\tilde{W}_{i,m_2,m_2}\right\|\right\|_2 \le \sum_{\substack{i\in[d_1]\\(m_1,m_2)\in\mathcal{P}_a}}\left\|\left\|\tilde{W}_{i,m_2,m_2}\right\|\right\|_2 \le \frac{d_1 k}{2(\sqrt{2})}$. From (207) and (213), putting $c_1 = 2$, we get,

$$\begin{aligned}\mathbb{E}\left[\tilde{Z}\right] &\le \sum_{a=1}^{k-1}\frac{8\alpha}{d_1\binom{k}{2}}\left(\sqrt{\frac{24\,k d_1\log d}{\min\{d_1,d_2\}}} \;+\; \frac{\sqrt{2}d_1 k}{d^2}\right)\sup_{\Delta\in\tilde{\mathcal{B}}(D)}|||\Delta|||_{\rm nuc}\\ &\overset{(a)}{\le} 8\alpha\left(2\sqrt{\frac{24\,\log d}{k\,d_1\min\{d_1,d_2\}}} \;+\; \frac{2\sqrt{2}}{d^2}\right)\frac{D^2}{\mu}\\ &\overset{(b)}{\le} 16\alpha\sqrt{\frac{48\log d}{k\,d_1\min\{d_1,d_2\}}}D^2\frac{1}{16\alpha}\sqrt{\frac{k\,\min\{d_1,d_2\}}{48 d_1 d_2^2\log d}}\\ &= \frac{D^2}{d_1 d_2}\,,\end{aligned} \tag{214}$$

where (a) is obtained because of (200) which gives $\sup_{D\in\mathcal{B}(D)}|||\Delta|||_{\rm nuc} \le D^2/\mu$ and (b) can be got by assuming $k \le d^2\log d$. Using the above bound in (205) we get,

$$\mathbb{P}\{\tilde{Z} - D^2/(d_1 d_2) \ge \epsilon\} \le \mathbb{P}\{\tilde{Z} - \mathbb{E}\left[\tilde{Z}\right] \ge \epsilon\} \le \exp\left(-\frac{2\epsilon^2}{d_1 k\left(\frac{32\alpha^2}{d_1 k}\right)^2}\right)\,, \tag{215}$$

and using $\epsilon = D^2/(2d_1 d_2)$ will get us the required bound.

Appendix G. Proof of Bundled Choices Theorem 13

We use similar notations and techniques as the proof of Theorem 7 in Appendix C. From the definition of $\mathcal{L}(\Theta)$ in Eq. (51), we have for the true parameter Θ^*, the gradient evaluated at the true parameter is

$$\nabla\mathcal{L}(\Theta^*) \;=\; -\frac{1}{n}\sum_{i=1}^{n}(e_{u_i}e_{v_i}^T - p_i)\,, \tag{216}$$

where p_i denotes the conditional probability of the MNL choice for the i-th sample. Precisely, $p_i = \sum_{j_1\in S_i}\sum_{j_2\in T_i} p_{j_1,j_2|S_i,T_i}e_{j_1}e_{j_2}^T$ where $p_{j_1,j_2|S_i,T_i}$ is the probability that the pair of items (j_1,j_2) is chosen at the i-th sample such that $p_{j_1,j_2|S_i,T_i} \equiv \mathbb{P}\{(u_i,v_i) = (j_1,j_2)|S_i,T_i\} = e^{\Theta^*_{j_1,j_2}}/(\sum_{j_1'\in S_i,j_2'\in T_i}e^{\Theta^*_{j_1',j_2'}})$, where (u_i,v_i) is the pair of items selected by the i-th user among the set of pairs of alternatives $S_i\times T_i$. The Hessian can be computed as

$$\frac{\partial^2\mathcal{L}(\Theta)}{\partial\Theta_{j_1,j_2}\,\partial\Theta_{j_1'j_2'}} \;=\; \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\big((j_1,j_2)\in S_i\times T_i\big)\frac{\partial p_{j_1,j_2|S_i,T_i}}{\partial\Theta_{j_1',j_2'}} \tag{217}$$

$$= \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}\big((j_1,j_2),(j_1',j_2')\in S_i\times T_i\big)\left(p_{j_1,j_2|S_i,T_i}\mathbb{I}((j_1,j_2)=(j_1',j_2')) - p_{j_1,j_2|S_i,T_i}p_{j_1',j_2'|S_i,T_i}\right)\,, \tag{218}$$

We use $\nabla^2\mathcal{L}(\Theta)\in\mathbb{R}^{d_1d_2\times d_1d_2}$ to denote this Hessian. Let $\Delta = \Theta^* - \widehat{\Theta}$ where $\widehat{\Theta}$ is an optimal solution to the convex optimization in (49). We introduce the following key technical lemmas. The following lemma provides a bound on the gradient using the concentration of measure for sum of independent random matrices Tropp (2011).

Lemma 39 *For any positive constant $c \ge 1$ and $n \ge (4(1+c)e^{2\alpha}d_1d_2\log d)/\max\{d_1,d_2\}$, with probability at least $1-2d^{-c}$,*

$$|||\nabla\mathcal{L}(\Theta^*)|||_2 \;\le\; \sqrt{\frac{4(1+c)e^{2\alpha}\max\{d_1,d_2\}\,\log d}{d_1\,d_2\,n}}\,. \tag{219}$$

Since we are typically interested in the regime where the number of samples is much smaller than the dimension $d_1\times d_2$ of the problem, the Hessian is typically not positive definite. However, when we restrict our attention to the vectorized Δ with relatively small nuclear norm, then we can prove restricted strong convexity, which gives the following bound.

Lemma 40 (Restricted Strong Convexity for bundled choice modeling) *Fix any $\Theta\in\Omega_\alpha$ and assume $(\min\{d_1,d_2\}/\min\{k_1,k_2\})\log d \le n \le \min\{d^5\log d, k_1k_2\max\{d_1^2,d_2^2\}\log d\}$. Under the random sampling model of the alternatives $\{j_{ia}\}_{i\in[n],a\in[k_1]}$ from the first set of items $[d_1]$, $\{j_{ib}\}_{i\in[n],b\in[k_1]}$ from the second set of items $[d_2]$ and the random outcome of the comparisons described in section 1, with probability larger than $1-2d^{-2^{25}}$,*

$$\mathrm{Vec}(\Delta)^T\,\nabla^2\mathcal{L}(\Theta)\,\mathrm{Vec}(\Delta) \;\ge\; \frac{e^{-2\alpha}}{8\,d_1\,d_2}|||\Delta|||_{\rm F}^2\,, \tag{220}$$

for all Δ in $\mathcal{A}$ where

$$\mathcal{A} = \left\{\Delta\in\mathbb{R}^{d_1\times d_2}\;\middle|\;|||\Delta|||_\infty \le 2\alpha\,,\ \sum_{j_1\in[d_1],j_2\in[d_2]}\Delta_{j_1j_2} = 0\ \ \textit{and}\ |||\Delta|||_{\rm F}^2 \ge \mu'|||\Delta|||_{\rm nuc}\right\}\,. \tag{221}$$

with

$$\mu' \;\equiv\; 2^{10}\,\alpha\,d_1d_2\sqrt{\frac{\log d}{n\,\min\{d_1,d_2\}\,\min\{k_1,k_2\}}}\,. \tag{222}$$

Building on these lemmas, the proof of Theorem 13 is divided into the following two cases. In both cases, we will show that

$$|||\Delta|||_{\rm F}^2 \;\le\; 12\,e^{2\alpha}c_1\lambda\,d_1d_2\,|||\Delta|||_{\rm nuc}\,, \tag{223}$$

with high probability. Finally, applying an omitted result similar to Lemma 27 proves the desired theorem. We are left to show Eq. (223) holds.

Case 1: Suppose $|||\Delta|||_{\rm F}^2 \ge \mu'|||\Delta|||_{\rm nuc}$. With $\Delta = \Theta^* - \widehat{\Theta}$, the Taylor expansion yields

$$\mathcal{L}(\widehat{\Theta}) = \mathcal{L}(\Theta^*) - \langle\!\langle\nabla\mathcal{L}(\Theta^*),\Delta\rangle\!\rangle + \frac{1}{2}\mathrm{Vec}(\Delta)\nabla^2\mathcal{L}(\Theta)\mathrm{Vec}^T(\Delta), \tag{224}$$

where $\Theta = a\widehat{\Theta}+(1-a)\Theta^*$ for some $a \in [0,1]$. It follows from Lemma 40 that with probability at least $1-2d^{-2^{25}}$,

$$\mathcal{L}(\widehat{\Theta}) - \mathcal{L}(\Theta^*) \;\geq\; -|||\nabla\mathcal{L}(\Theta^*)|||_2 |||\Delta|||_{\rm nuc} + \frac{e^{-2\alpha}}{8\,d_1\,d_2}|||\Delta|||_{\rm F}^2 \,.$$

From the definition of $\widehat{\Theta}$ as an optimal solution of the minimization, we have

$$\mathcal{L}(\widehat{\Theta}) - \mathcal{L}(\Theta^*) \;\leq\; \lambda\left(|||\Theta^*|||_{\rm nuc} - \left|||\widehat{\Theta}\right|||_{\rm nuc}\right) \;\leq\; \lambda|||\Delta|||_{\rm nuc}\,.$$

By the assumption, we choose $\lambda \geq 8\lambda_0$. In view of Lemma 39, this implies that $\lambda \geq 2|||\nabla\mathcal{L}(\Theta^*)|||_2$ with probability at least $1-2d^{-3}$. It follows that with probability at least $1-2d^{-3}-2d^{-2^{25}}$,

$$\frac{e^{-2\alpha}}{8d_1d_2}|||\Delta|||_{\rm F}^2 \;\leq\; \left(\lambda + |||\nabla\mathcal{L}(\Theta^*)|||_2\right)|||\Delta|||_{\rm nuc} \;\leq\; \frac{3\lambda}{2}|||\Delta|||_{\rm nuc}\,.$$

By our assumption on $\lambda \leq c_1\lambda_0$, this proves the desired bound in Eq. (223)

Case 2: Suppose $|||\Delta|||_{\rm F}^2 \leq \mu'|||\Delta|||_{\rm nuc}$**.** By the definition of μ and the fact that $c_1 \geq 128/\sqrt{\min\{k_1,k_2\}}$, it follows that $\mu' \leq 12\,e^{2\alpha}c_1\lambda\,d_1d_2$, and we get the same bound as in Eq. (223).

G.1 Proof of Lemma 39

Define $X_i = -(e_{u_i}e_{v_i}^T - p_i)$ such that $\nabla\mathcal{L}(\Theta^*) = (1/n)\sum_{i=1}^n X_i$, which is a sum of n independent random matrices. Note that since p_i is entry-wise bounded by $e^{2\alpha}/(k_1k_2)$,

$$|||X_i|||_2 \;\leq\; 1 + \frac{e^{2\alpha}}{\sqrt{k_1k_2}}\,,$$

and

$$\sum_{i=1}^n \mathbb{E}[X_iX_i^T] \;=\; \sum_{i=1}^n(\mathbb{E}[e_{u_i}e_{u_i}^T] - p_ip_i^T) \tag{225}$$

$$\preceq\; \sum_{i=1}^n \mathbb{E}[e_{u_i}e_{u_i}^T] \tag{226}$$

$$\preceq\; \frac{e^{2\alpha}n}{d_1}\mathbf{I}_{d_1\times d_1}\,, \tag{227}$$

where the last inequality follows from the fact that for any given S_i, u_i will be chosen with probability at most $e^{2\alpha}/k_1$, if it is in the set S_i which happens with probability k_1/d_1. Therefore,

$$\left|\left|\left|\sum_{i=1}^n \mathbb{E}[X_iX_i^T]\right|\right|\right|_2 \;\leq\; \frac{e^{2\alpha}n}{d_1}\,. \tag{228}$$

Similarly,

$$\left|\left|\left|\sum_{i=1}^n \mathbb{E}[X_i^TX_i]\right|\right|\right|_2 \;\leq\; \frac{e^{2\alpha}n}{d_2}\,. \tag{229}$$

Applying matrix Bernstein inequality Tropp (2011), we get

$$\begin{aligned}&\mathbb{P}\{|||\nabla\mathcal{L}(\Theta^*)|||_2 > t\}\\ &\leq (d_1+d_2)\exp\left\{\frac{-n^2t^2/2}{(e^{2\alpha}n\max\{d_1,d_2\}/(d_1d_2)) \;+\; ((1+(e^{2\alpha}/\sqrt{k_1k_2}))nt/3)}\right\},\end{aligned} \tag{230}$$

which gives the desired tail probability of $2d^{-c}$ for the choice of

$$\begin{aligned}t \;&=\; \max\left\{\sqrt{\frac{4(1+c)e^{2\alpha}\max\{d_1,d_2\}\log d}{d_1d_2n}}\,,\; \frac{4(1+c)(1+\frac{e^{2\alpha}}{\sqrt{k_1k_2}})\log d}{3n}\right\}\\ &=\; \sqrt{\frac{4(1+c)e^{2\alpha}\max\{d_1,d_2\}\log d}{d_1d_2n}}\,,\end{aligned}$$

where the last equality follows from the assumption, $n \geq (4(1+c)e^{2\alpha}d_1d_2\log d)/\max\{d_1,d_2\}$.

G.2 Proof of Lemma 40

Thee quadratic form of the Hessian defined in (218) can be lower bounded by

$$\mathrm{Vec}(\Delta)^T\,\nabla^2\mathcal{L}(\Theta)\,\mathrm{Vec}(\Delta) \;\geq\; \underbrace{\frac{e^{-2\alpha}}{2\,k_1^2\,k_2^2\,n}\sum_{i=1}^n\sum_{j_1,j_1'\in S_i}\sum_{j_2,j_2'\in T_i}\left(\Delta_{j_1,j_2}-\Delta_{j_1',j_2'}\right)^2}_{\equiv H(\Delta)}\,, \tag{231}$$

which follows from Remark 31. To lower bound $H(\Delta)$, we first compute the mean:

$$\mathbb{E}[H(\Delta)] \;=\; \frac{e^{-2\alpha}}{2\,k_1^2\,k_2^2\,n}\sum_{i=1}^n\mathbb{E}\Big[\sum_{j_1,j_1'\in S_i}\sum_{j_2,j_2'\in T_i}\left(\Delta_{j_1,j_2}-\Delta_{j_1',j_2'}\right)^2\Big] \tag{232}$$

$$=\; \frac{e^{-2\alpha}}{d_1\,d_2}|||\Delta|||_{\rm F}^2\,, \tag{233}$$

where we used the fact that $\mathbb{E}[\sum_{j_1\in S_i,j_2\in T_i}\Delta_{j_1,j_2}] = (k_1k_2/(d_1d_2))\sum_{j_1'\in[d_1],j_2'\in[d_2]}\Delta_{j_1',j_2'} = 0$ for $\Delta\in\Omega_{2\alpha}$ in (51).

We now prove that $H(\Delta)$ does not deviate from its mean too much. Suppose there exists a $\Delta\in\mathcal{A}$ defined in (221) such that Eq. (220) is violated, i.e. $H(\Delta) < (e^{-2\alpha}/(8d_1d_2))|||\Delta|||_{\rm F}^2$. In this case,

$$\mathbb{E}[H(\Delta)] - H(\Delta) \;\geq\; \frac{7\,e^{-2\alpha}}{8d_1d_2}|||\Delta|||_{\rm F}^2\,. \tag{234}$$

We will show that this happens with a small probability. We use the same peeling argument as in Appendix C with

$$\begin{aligned}\mathcal{S}_\ell = \Big\{\Delta\in\mathbb{R}^{d_1\times d_2}\,|\,&|||\Delta|||_\infty \leq 2\alpha, \beta^{\ell-1}\mu' \leq |||\Delta|||_{\rm F} \leq \beta^\ell\mu',\\ &\sum_{j_1\in[d_1],j_2\in[d_2]}\Delta_{j_1,j_2} = 0,\ \text{and}\ |||\Delta|||_{\rm nuc} \leq \beta^{2\ell}\mu'\Big\}\,,\end{aligned}$$

where $\beta = \sqrt{10/9}$ and for $\ell \in \{1, 2, 3, \ldots\}$, and μ' is defined in (222). By the peeling argument, there exists an $\ell \in \mathbb{Z}_+$ such that $\Delta \in \mathcal{S}_\ell$ and

$$\mathbb{E}[H(\Delta)] - H(\Delta) \;\; \geq \;\; \frac{7\, e^{-2\alpha}}{8 d_1 d_2} \beta^{2\ell-2} (\mu')^2 \;\; \geq \;\; \frac{7\, e^{-2\alpha}}{9\, d_1 d_2} \beta^{2\ell} (\mu')^2 \, . \tag{235}$$

Applying the union bound over $\ell \in \mathbb{Z}_+$,

$$\begin{aligned}
&\mathbb{P}\left\{ \exists \Delta \in \mathcal{A} \, , \; H(\Delta) < \frac{e^{-2\alpha}}{8\, d_1\, d_2} |\!|\!| \Delta |\!|\!|_{\mathrm{F}}^2 \right\} \\
&\leq \sum_{\ell=1}^{\infty} \mathbb{P}\left\{ \sup_{\Delta \in \mathcal{S}_\ell} \big(\, \mathbb{E}[H(\Delta)] - H(\Delta) \, \big) > \frac{7\, e^{-2\alpha}}{9 d_1 d_2} (\beta^\ell \mu')^2 \right\} \\
&\qquad\qquad \leq \sum_{\ell=1}^{\infty} \mathbb{P}\left\{ \sup_{\Delta \in \mathcal{B}'(\beta^\ell \mu')} \big(\, \mathbb{E}[H(\Delta)] - H(\Delta) \, \big) > \frac{7 e^{-2\alpha}}{9 d_1 d_2} (\beta^\ell \mu')^2 \right\} ,
\end{aligned} \tag{236}$$

where we define the set $\mathcal{B}'(D)$ such that $\mathcal{S}_\ell \subseteq \mathcal{B}'(\beta^\ell \mu')$:

$$\mathcal{B}'(D) = \Big\{ \, \Delta \in \mathbb{R}^{d_1 \times d_2} \, \Big| \, \|\Delta\|_\infty \leq 2\alpha, |\!|\!| \Delta |\!|\!|_{\mathrm{F}} \leq D, \sum_{j_1 \in [d_1], j_2 \in [d_2]} \Delta_{j_1 j_2} = 0, \mu' |\!|\!| \Delta |\!|\!|_{\mathrm{nuc}} \leq D^2 \, \Big\} \, . \tag{237}$$

The following key lemma provides the upper bound on this probability.

Lemma 41 *For* $(\min\{d_1, d_2\}/\min\{k_1, k_2\}) \log d \leq n \leq d^5 \log d$,

$$\mathbb{P}\left\{ \sup_{\Delta \in \mathcal{B}'(D)} \Big(\, \mathbb{E}[H(\Delta)] - H(\Delta) \, \Big) \geq \frac{e^{-2\alpha} D^2}{2 d_1 d_2} \right\} \leq \quad \exp\Big\{ - \frac{n \, \min\{k_1^2, k_2^2\} \, k_1 k_2 \, D^4}{2^{10} \alpha^4 d_1^2 d_2^2} \Big\} \, . \tag{238}$$

Let $\eta = \exp\Big(- \frac{n k_1 k_2 \min\{k_1^2, k_2^2\} (\beta - 1.002) (\mu')^4}{2^{10} \alpha^4 d_1^2 d_2^2} \Big)$. Applying the tail bound to (236), we get

$$\begin{aligned}
\mathbb{P}\left\{ \exists \Delta \in \mathcal{A} \, , \; H(\Delta) < \frac{e^{-2\alpha}}{8\, d_1 d_2} |\!|\!| \Delta |\!|\!|_{\mathrm{F}}^2 \right\} \;\; &\leq \;\; \sum_{\ell=1}^{\infty} \exp\Big\{ - \frac{n \, k_1 k_2 \, \min\{k_1^2, k_2^2\} \, (\beta^\ell \mu')^4}{2^{10} \alpha^4 d_1^2 d_2^2} \Big\} \\
&\overset{(a)}{\leq} \;\; \sum_{\ell=1}^{\infty} \exp\Big\{ - \frac{n k_1 k_2 \min\{k_1^2, k_2^2\} \ell (\beta - 1.002) (\mu')^4}{2^{10} \alpha^4 d_1^2 d_2^2} \Big\} \\
&\leq \;\; \frac{\eta}{1 - \eta},
\end{aligned}$$

where (a) holds because $\beta^x \geq x \log \beta \geq x(\beta - 1.002)$ for the choice of $\beta = \sqrt{10/9}$. By the definition of μ',

$$\eta \;\; = \;\; \exp\Big\{ - \frac{2^{30} \, k_1 k_2 \max\{d_2^2, d_1^2\} (\log d)^2 (\beta - 1.002)}{n} \Big\} \;\; \leq \;\; \exp\{ - 2^{25} \, \log d \} \, ,$$

where the last inequality follows from the assumption that $n \leq k_1 k_2 \max\{d_1^2, d_2^2\} \log d$, and $\beta - 1.002 \geq 2^{-5}$. Since for $d \geq 2$, $\exp\{-2^{25} \log d\} \leq 1/2$ and thus $\eta \leq 1/2$, the lemma follows by assembling the last two displayed inequalities.

G.3 Proof of Lemma 41

Let $Z \equiv \sup_{\Delta \in \mathcal{B}'(D)} \mathbb{E}[H(\Delta)] - H(\Delta)$ and consider the tail bound using McDiarmid's inequality. Note that Z has a bounded difference of $(8\alpha^2 e^{-2\alpha} \max\{k_1, k_2\})/(k_1^2 k_2^2 n)$ when one of the $k_1 k_2 n$ independent random variables are changed, which gives

$$\mathbb{P}\left\{ Z - \mathbb{E}[Z] \geq t \right\} \;\; \leq \;\; \exp\Big(- \frac{k_1^4 k_2^4 n^2 t^2}{64 \alpha^4 e^{-4\alpha} \max\{k_1^2, k_2^2\} k_1 k_2 n} \Big) \, . \tag{239}$$

With the choice of $t = D^2/(4 e^{2\alpha} d_1 d_2)$, this gives

$$\mathbb{P}\left\{ Z - \mathbb{E}[Z] \geq \frac{e^{-2\alpha}}{4 d_1 d_2} D^2 \right\} \;\; \leq \;\; \exp\Big(- \frac{k_1^3 k_2^3 n D^4}{2^{10} \alpha^4 d_1^2 d_2^2 \max\{k_1^2, k_2^2\}} \Big) \, . \tag{240}$$

We first construct a partition of the space similar to Lemma 33. Let

$$\tilde{k} \;\; \equiv \;\; \min\{k_1, k_2\} \, . \tag{241}$$

Lemma 42 *There exists a partition* $(\mathcal{T}_1, \ldots, \mathcal{T}_N)$ *of* $\{[k_1] \times [k_2]\} \times \{[k_1] \times [k_2]\}$ *for some* $N \leq 2 k_1^2 k_2^2 / \tilde{k}$ *such that* $\mathcal{T}_\ell$*'s are disjoint subsets,* $\bigcup_{\ell \in [N]} \mathcal{T}_\ell = \{[k_1] \times [k_2]\} \times \{[k_1] \times [k_2]\}$, $|\mathcal{T}_\ell| \leq \tilde{k}$ *and for any* $\ell \in [N]$ *the set of random variables in* $\mathcal{T}_\ell$ *satisfy*

$$\{ (\Delta_{j_{i,a}, j_{i,b}} - \Delta_{j_{i,a'}, j_{i,b'}})^2 \}_{i \in [n], ((a,b),(a',b')) \in \mathcal{T}_\ell} \text{ are mutually independent } .$$

where $j_{i,a}$ *for* $i \in [n]$ *and* $a \in [k_1]$ *denote the* a*-th chosen item to be included in the set* S_i*.*

Now we prove an upper bound on $\mathbb{E}[Z]$ using the symmetrization technique. Recall that $j_{i,a}$ is independently and uniformly chosen from $[d_1]$ for $i \in [n]$ and $a \in [k_1]$. Similarly, $j_{i,b}$ is independently and uniformly chosen from $[d_1]$ for $i \in [n]$ and $b \in [k_2]$.

$$\begin{aligned}
&\mathbb{E}[Z] \\
&= \frac{e^{-2\alpha}}{2\, k_1^2 k_2^2 \, n} \mathbb{E}\left[\sup_{\Delta \in \mathcal{B}'(D)} \sum_{i=1}^{n} \sum_{\substack{a, a' \in [k_1] \\ b, b' \in [k_2]}} \mathbb{E}\big[\big(\Delta_{j_{i,a}, j_{i,b}} - \Delta_{j_{i,a'}, j_{i,b'}} \big)^2 \big] - \big(\Delta_{j_{i,a}, j_{i,b}} - \Delta_{j_{i,a'}, j_{i,b'}} \big)^2 \right] \\
&\leq \frac{e^{-2\alpha}}{2\, k_1^2 k_2^2 \, n} \sum_{\ell \in [N]} \mathbb{E}\left[\sup_{\Delta \in \mathcal{B}'(D)} \sum_{i=1}^{n} \sum_{(j_1, j_2, j_1', j_2') \in \mathcal{T}_\ell} \mathbb{E}\big[\big(\Delta_{j_1, j_2} - \Delta_{j_1', j_2'} \big)^2 \big] - \big(\Delta_{j_1, j_2} - \Delta_{j_1', j_2'} \big)^2 \right] \\
&\leq \frac{e^{-2\alpha}}{k_1^2 k_2^2 \, n} \sum_{\ell \in [N]} \mathbb{E}\left[\sup_{\Delta \in \mathcal{B}'(D)} \sum_{i=1}^{n} \sum_{(j_1, j_2, j_1', j_2') \in \mathcal{T}_\ell} \xi_{i, j_1, j_2, j_1', j_2'} \big(\Delta_{j_1, j_2} - \Delta_{j_1', j_2'} \big)^2 \right] ,
\end{aligned} \tag{242}$$

where the first inequality follows for the fact that the supremum of the sum is smaller than the sum of supremum, and the second inequality follows from standard symmetrization

with i.i.d. Rademacher random variables $\xi_{i,j_1,j_2,j_1',j_2'}$'s. It follows from Ledoux-Talagrand contraction inequality Ledoux and Talagrand (2013) that

$$\mathbb{E}\left[\sup_{\Delta\in\mathcal{B}'(D)}\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(\Delta_{j_1,j_2}-\Delta_{j_1',j_2'}\big)^2\right] \tag{243}$$

$$\leq\ 8\alpha\,\mathbb{E}\left[\sup_{\Delta\in\mathcal{B}'(D)}\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(\Delta_{j_1,j_2}-\Delta_{j_1',j_2'}\big)\right] \tag{244}$$

$$\leq\ 8\alpha\,\mathbb{E}\left[\sup_{\Delta\in\mathcal{B}'(D)}|||\Delta|||_{\mathrm{nuc}}\left|\left|\left|\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(e_{j_1,j_2}-e_{j_1',j_2'}\big)\right|\right|\right|_2\right] \tag{245}$$

$$\leq\ \frac{8\alpha D^2}{\mu'}\mathbb{E}\left[\left|\left|\left|\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(e_{j_1,j_2}-e_{j_1',j_2'}\big)\right|\right|\right|_2\right]\ , \tag{246}$$

where the second inequality follows for the Hölder's inequality and the last inequality follows from $\mu'|||\Delta|||_{\mathrm{nuc}}\leq D^2$ for all $\Delta\in\mathcal{B}'(D)$. To bound the expected spectral norm of the random matrix, we use matrix Bernstein's inequality. Note that $\left|\left|\left|\xi_{i,j_1,j_2,j_1',j_2'}c\right|\right|\right|_2\leq\sqrt{2}$ almost surely, $\mathbb{E}[(e_{j_1,j_2}-e_{j_1',j_2'})(e_{j_1,j_2}-e_{j_1',j_2'})^T]\preceq(2/d_1)\mathbf{I}_{d_1\times d_1}$, and $\mathbb{E}[(e_{j_1,j_2}-e_{j_1',j_2'})^T(e_{j_1,j_2}-e_{j_1',j_2'})]\preceq(2/d_2)\mathbf{I}_{d_2\times d_2}$. It follows that $\sigma^2=2n|\mathcal{T}_\ell|/\min\{d_1,d_2\}$, where $|\mathcal{T}_\ell|\leq\min\{k_1,k_2\}$. It follows that

$$\mathbb{P}\left\{\left|\left|\left|\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(e_{j_1,j_2}-e_{j_1',j_2'}\big)\right|\right|\right|_2>t\right\}$$
$$\leq(d_1+d_2)\exp\left\{\frac{-t^2/2}{2n\min\{k_1,k_2\}/\min\{d_1,d_2\}+\sqrt{2}t/3}\right\},$$

Choosing $t=\max\{\sqrt{64n(\min\{k_1,k_2\}/\min\{d_1,d_2\})\log d},(16\sqrt{2}/3)\log d\}$, we obtain a bound on the spectral norm of t with probability at least $1-2d^{-7}$. From the fact that $\left|\left|\left|\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(e_{j_1,j_2}-e_{j_1',j_2'}\big)\right|\right|\right|_2\leq(n/\sqrt{2})\min\{k_1,k_2\}$, it follows that

$$\mathbb{E}\left[\left|\left|\left|\sum_{i=1}^{n}\sum_{(j_1,j_2,j_1',j_2')\in\mathcal{T}_\ell}\xi_{i,j_1,j_2,j_1',j_2'}\big(e_{j_1,j_2}-e_{j_1',j_2'}\big)\right|\right|\right|_2\right] \tag{247}$$

$$\leq\ \max\left\{\sqrt{\frac{64\,n\ \min\{k_1,k_2\}\log d}{\min\{d_1,d_2\}}},(16\sqrt{2}/3)\log d\right\}+\frac{2n\min\{k_1,k_2\}}{\sqrt{2}d^7} \tag{248}$$

$$\leq\ \sqrt{\frac{66\,n\ \min\{k_1,k_2\}\log d}{\min\{d_1,d_2\}}} \tag{249}$$

which follows form the assumption that $n\min\{k_1,k_2\}\geq\min\{d_1,d_2\}\log d$ and $n\leq d^5\log d$. Substituting this bound in (242), and (246), we get that

$$\mathbb{E}[Z]\ \leq\ \frac{16e^{-2\alpha}\alpha D^2}{\mu'}\sqrt{\frac{66\log d}{n\min\{k_1,k_2\}\min\{d_1,d_2\}}} \tag{250}$$

$$\leq\ \frac{e^{-2\alpha}D^2}{4\,d_1d_2}\,. \tag{251}$$

Appendix H. Proof of the Information-theoretic Lower Bound in Theorem 16

This proof follow closely the proof of Theorem 10 in Appendix E. We apply the generalized Fano's inequality in the same way to get Eq. (160)

$$\mathbb{P}\left\{\widehat{L}\neq L\right\}\ \geq\ 1-\frac{\binom{M}{2}^{-1}\sum_{\ell_1,\ell_2\in[M]}D_{\mathrm{KL}}(\Theta^{(\ell_1)}\|\Theta^{(\ell_2)})+\log 2}{\log M}\,, \tag{252}$$

The main challenge in this case is that we can no longer directly apply the RUM interpretation to compete $D_{\mathrm{KL}}(\Theta^{(\ell_1)}\|\Theta^{(\ell_2)})$. This will result in over estimating the KL-divergence, because this approach does not take into account that we only take the top winner, out of those k_1k_2 alternatives. Instead, we compute the divergence directly, and provide an appropriate bound. Let the set of k_1 rows and k_2 columns chosen in one of the n samples

be $S \subset [d_1]$ and $T \subset [d_2]$ respectively. Then,

$$
\begin{aligned}
& D_{\mathrm{KL}}(\Theta^{(\ell_1)} \| \Theta^{(\ell_2)}) \\
\overset{(a)}{=} & \frac{n}{\binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \sum_{\substack{i\in S \\ j\in T}} \frac{e^{\Theta_{ij}^{(\ell_1)}}}{\sum_{\substack{i'\in S \\ j'\in T}} e^{\Theta_{i'j'}^{(\ell_1)}}} \log \left(\frac{e^{\Theta_{ij}^{(\ell_1)}} \sum_{\substack{i'\in S \\ j'\in T}} e^{\Theta_{i'j'}^{(\ell_2)}}}{e^{\Theta_{ij}^{(\ell_2)}} \sum_{\substack{i'\in S \\ j'\in T}} e^{\Theta_{i'j'}^{(\ell_1)}}} \right) \\
\overset{(b)}{\le} & \frac{n}{\binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \left(\sum_{i,j} \frac{e^{2\Theta_{ij}^{(\ell_1)}} \sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_2)}} - e^{\Theta_{ij}^{(\ell_1)}+\Theta_{ij}^{(\ell_2)}} \sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_1)}}}{e^{\Theta_{ij}^{(\ell_2)}} \left(\sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_1)}} \right)^2} \right) \\
\overset{(c)}{\le} & \frac{ne^{2\alpha}}{k_1^2 k_2^2 \binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \sum_{i,j} \left(e^{2\Theta_{ij}^{(\ell_1)}-\Theta_{ij}^{(\ell_2)}} \sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_2)}} - e^{\Theta_{ij}^{(\ell_1)}} \sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_1)}} \right) \\
= & \frac{ne^{2\alpha}}{k_1^2 k_2^2 \binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \left(\sum_{i',j'} e^{\Theta_{i'j'}^{(\ell_2)}} \sum_{i,j} \frac{\left(e^{\Theta_{ij}^{(\ell_1)}} - e^{\Theta_{ij}^{(\ell_2)}} \right)^2}{e^{\Theta_{ij}^{(\ell_2)}}} - \Big(\sum_{i,j} (e^{\Theta_{ij}^{(\ell_1)}} - e^{\Theta_{ij}^{(\ell_2)}}) \Big)^2 \right) \\
\overset{(d)}{\le} & \frac{ne^{4\alpha}}{k_1 k_2 \binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \sum_{i,j} \left(e^{\Theta_{ij}^{(\ell_1)}} - e^{\Theta_{ij}^{(\ell_2)}} \right)^2 \\
\overset{(e)}{\le} & \frac{ne^{5\alpha}}{k_1 k_2 \binom{d_1}{k_1}\binom{d_2}{k_2}} \sum_{S,T} \sum_{i,j} \left(\Theta_{ij}^{(\ell_1)} - \Theta_{ij}^{(\ell_2)} \right)^2 \\
\overset{(f)}{=} & \frac{ne^{5\alpha}}{d_1 d_2} \left\| \Theta_{ij}^{(\ell_1)} - \Theta_{ij}^{(\ell_2)} \right\|_{\mathrm{F}}^2
\end{aligned}
$$

Here (a) is by definition of KL-distance and the fact that S, T are chosen uniformly from all possible such sets and (b) is due to the fact that $\log(x) \le x - 1$ with $x = (e^{\Theta_{ij}^{(\ell_1)}} \sum_{i'\in S, j'\in T} e^{\Theta_{i'j'}^{(\ell_2)}})/(e^{\Theta_{ij}^{(\ell_2)}} \sum_{i'\in S, j'\in T} e^{\Theta_{i'j'}^{(\ell_1)}})$. The constants at (c) is due to the fact that each element of $\Theta^{(\ell_1)}$ is upper bounded by α and lower bounded by $-\alpha$. We can get (d) by removing the second term which is always negative, and using the bond of α. (e) is obtained because e^x where $-\alpha \le x \le \alpha$ is Lipschitz continuous with Lipschitz constant e^α. At last (f) is obtained by simple counting of the occurrences of each ij. Thus we have,

$$
\mathbb{P}\left\{ \widehat{L} \neq L \right\} \;\ge\; 1 - \frac{\binom{M}{2}^{-1} \sum_{\ell_1,\ell_2\in[M]} \frac{ne^{5\alpha}}{d_1 d_2} \left\| \Theta_{ij}^{(\ell_2)} - \Theta_{ij}^{(\ell_2)} \right\|_{\mathrm{F}}^2 + \log 2}{\log M}, \tag{253}
$$

The remainder of the proof relies on the following probabilistic packing.

Lemma 43 *Let $d_2 \ge d_1$ be sufficiently large positive integers. Then for each $r \in \{1, \dots, d_1\}$, and for any positive $\delta > 0$ there exists a family of $d_1 \times d_2$ dimensional matrices $\{\Theta^{(1)}, \dots, \Theta^{(M(\delta))}\}$ with cardinality $M(\delta) = \lfloor (1/4)\exp(rd_2/576) \rfloor$ such that each matrix is rank r and the following bounds hold:*

$$
\begin{aligned}
\left\| \Theta^{(\ell)} \right\|_{\mathrm{F}} &\le \delta \;, \text{ for all } \ell \in [M] && (254)\\
\left\| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right\|_{\mathrm{F}} &\ge \frac{1}{2}\delta \;, \text{ for all } \ell_1, \ell_2 \in [M] && (255)\\
\Theta^{(\ell)} &\in \Omega_{\tilde{\alpha}} \;, \text{ for all } \ell \in [M] \,, && (256)
\end{aligned}
$$

with $\tilde{\alpha} = (8\delta/d_2)\sqrt{2\log d}$ for $d = (d_1 + d_2)/2$.

Suppose $\delta \le \alpha d_2/(8\sqrt{2\log d})$ such that the matrices in the packing set are entry-wise bounded by α, then the above lemma 43 implies that $\left\| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right\|_{\mathrm{F}}^2 \le 4\delta^2$, which gives

$$
\mathbb{P}\left\{ \widehat{L} \neq L \right\} \;\ge\; 1 - \frac{\frac{e^{5\alpha} n 4\delta^2}{d_1 d_2} + \log 2}{\frac{rd_2}{576} - 2\log 2} \;\ge\; \frac{1}{2} \,, \tag{257}
$$

where the last inequality holds for $\delta^2 \le (rd_1 d_2^2/(1152 e^{5\alpha} n))$ and assuming $rd_2 \ge 1600$. Together with (257) and (255), this proves that for all $\delta \le \min\{\alpha d_2/(8\sqrt{2\log d}), \sqrt{rd_1 d_2^2/(9216\, e^{5\alpha} n)}\}$,

$$
\inf_{\widehat{\Theta}} \sup_{\Theta^* \in \Omega_\alpha} \mathbb{E}\Big[\left\| \widehat{\Theta} - \Theta^* \right\|_{\mathrm{F}} \Big] \;\ge\; \delta/4 \,.
$$

Choosing δ appropriately to maximize the right-hand side finishes the proof of the desired claim. Also by symmetry, we can apply the same argument to get similar bound with d_1 and d_2 interchanged.

H.1 Proof of Lemma 43

We show that the following procedure succeeds in producing the desired family with probability at least half, which proves its existence. Let $d = (d_1 + d_2)/2$, and suppose $d_2 \ge d_1$ without loss of generality. For the choice of $M' = e^{rd_2/576}$, and for each $\ell \in [M']$, generate a rank-r matrix $\Theta^{(\ell)} \in \mathbb{R}^{d_1 \times d_2}$ as follows:

$$
\Theta^{(\ell)} \;=\; \frac{\delta}{\sqrt{rd_2}} U (V^{(\ell)})^T - \frac{\delta}{\sqrt{rd_2}} \frac{\mathbb{1}^T U (V^{(\ell)})^T \mathbb{1}}{d_1 d_2} \mathbb{1}\mathbb{1}^T \,, \tag{258}
$$

where $U \in \mathbb{R}^{d_1 \times r}$ is a random orthogonal basis such that $U^T U = \mathbf{I}_{r\times r}$ and $V^{(\ell)} \in \mathbb{R}^{d_2 \times r}$ is a random matrix with each entry $V_{ij}^{(\ell)} \in \{-1, +1\}$ chosen independently and uniformly at random. By construction, notice that $\left\| \Theta^{(\ell)} \right\|_{\mathrm{F}} \le (\delta/\sqrt{rd_2}) \left\| U (V^{(\ell)})^T \right\|_{\mathrm{F}} = \delta$.

Now, by triangular inequality, we have

$$
\begin{aligned}
& \left\| \Theta^{(\ell_1)} - \Theta^{(\ell_2)} \right\|_{\mathrm{F}} \\
\ge\; & \frac{\delta}{\sqrt{rd_2}} \left\| U (V^{(\ell_1)} - V^{(\ell_2)})^T \right\|_{\mathrm{F}} - \frac{\delta \left| \mathbb{1}^T U (V^{(\ell_1)} - V^{(\ell_2)})^T \mathbb{1} \right|}{d_1 d_2 \sqrt{rd_2}} \left\| \mathbb{1}\mathbb{1}^T \right\|_{\mathrm{F}} \\
\ge\; & \frac{\delta}{\sqrt{rd_2}} \underbrace{\left\| V^{(\ell_1)} - V^{(\ell_2)} \right\|_{\mathrm{F}}}_{A} - \frac{\delta}{\sqrt{r\, d_1\, d_2^2}} \big(\underbrace{\left| \mathbb{1}^T U (V^{(\ell_1)})^T \mathbb{1} \right|}_{B} + \left| \mathbb{1}^T U (V^{(\ell_2)})^T \mathbb{1} \right| \big) \,.
\end{aligned}
$$

We will prove that the first term is bounded by $A \geq \sqrt{rd_2}$ with probability at least 7/8 for all M' matrices, and we will show that we can find M matrices such that the second term is bounded by $B \leq 8\sqrt{2rd_2 \log(32r)\log(32d)}$ with probability at least 7/8. Together, this proves that with probability at least 3/4, there exists M matrices such that

$$\left|\left|\left|\Theta^{(\ell_1)} - \Theta^{(\ell_2)}\right|\right|\right|_{\mathrm{F}} \;\geq\; \delta\Big(1 - \sqrt{\frac{2^7 \log(32r)\, \log(32d)}{d_1 d_2}}\Big) \;\geq\; \frac{1}{2}\delta\,,$$

for all ℓ_1, $\ell_2 \in [M]$ and for sufficiently large d_1 and d_2.

Applying similar McDiarmid's inequality as Eq. (122) in Appendix E, it follows that $A^2 \geq rd_2$ with probability at least 7/8 for $M' = e^{rd_2/576}$ and a sufficiently large d_2.

To prove a bound on B, we will show that for a given ℓ,

$$\mathbb{P}\Big\{|\mathbb{1}^T U (V^{(\ell)})^T \mathbb{1}| \leq 8\sqrt{2rd_2 \log(32r)\log(32d)}\Big\} \;\geq\; \frac{7}{8}\,. \tag{259}$$

Then using the similar technique as in (125), it follows that we can find $M = (1/4)M'$ matrices all satisfying this bound and also the bound on the max-entry in (260). We are left to prove (259). We apply a series of concentration inequalities. Let H_1 be the event that $\{|\langle\!\langle V_i^{(\ell)}, \mathbb{1}\rangle\!\rangle| \leq \sqrt{2d_2 \log(32r)}$ for all $i \in [r]\}$. Then, applying the standard Hoeffding's inequality, we get that $\mathbb{P}\{H_1\} \geq 15/16$, where $V_i^{(\ell)}$ is the i-th column of $V^{(\ell)}$. We next change the variables and represent $\mathbb{1}^T U$ as $\sqrt{d_1} u^T \tilde{U}$, where u is drawn uniformly at random from the unit sphere and $\tilde{U}$ is a r dimensional subspace drawn uniformly at random. By symmetry, $\sqrt{d_1} u^T \tilde{U}$ have the same distribution as $\mathbb{1}^T U$. Let H_2 be the event that $\{|\langle\!\langle \tilde{U}_i, (V^{(\ell)})^T \mathbb{1}\rangle\!\rangle| \leq \sqrt{16r(d_2/d_1)\log(32r)\log(32d)}$ for all $i \in [d_1]\}$, where $\tilde{U}_i$ is the i-th row of $\tilde{U}$. Then, applying Levy's theorem for concentration on the sphere Ledoux (2005), we have $\mathbb{P}\{H_2|H_1\} \geq 15/16$. Finally, let H_3 be the event that $\{|\sqrt{d_1}\langle\!\langle u, \tilde{U}(V^{(\ell)})^T\rangle\!\rangle \mathbb{1}| \leq 8\sqrt{2rd_2 \log(32r)\log(32d)}\}$. Then, again applying Levy's concentration, we get $\mathbb{P}\{H_3|H_1, H_2\} \geq 15/16$. Collecting all three concentration inequalities, we get that with probability at least 13/16, $|\mathbb{1}^T U (V^{(\ell)})^T \mathbb{1}| \leq 8\sqrt{2rd_2 \log(32r)\log(32d)}$, which proves Eq. (259).

We are left to prove that $\Theta^{(\ell)}$'s are in $\Omega_{(8\delta/d_2)\sqrt{2\log d_2}}$ as defined in (51). Similar to Eq. (124), applying Levy's concentration gives

$$\mathbb{P}\Big\{\max_{i,j} |\Theta_{ij}^{(\ell)}| \leq \frac{2\delta\sqrt{32\log d_2}}{d_2}\Big\} \;\geq\; 1 - 2\exp\Big\{-2\log d_2\Big\} \;\geq\; \frac{1}{2}\,, \tag{260}$$

for a fixed $\ell \in [M']$. Then using the similar technique as in (125), it follows that there exists $M = (1/4)M'$ matrices all satisfying this bound and also the bound on B in Eq. (259).

References

A. Agarwal, S. Negahban, and M. Wainwright. Fast global convergence rates of gradient methods for high-dimensional statistical recovery. In *In NIPS*, pages 37–45, 2010.

A. Agarwal, P. Patil, and S. Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pages 70–79, 2018.

H. Azari Soufiani, D. C. Parkes, and L. Xia. Random utility theory for social choice. In *NIPS*, pages 126–134, 2012.

H. Azari Soufiani, W. Chen, D. C. Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems 26*, pages 2706–2714, 2013.

H. Azari Soufiani, D. Parkes, and L. Xia. Computing parametric ranking models via rank-breaking. In *Proceedings of The 31st International Conference on Machine Learning*, pages 360–368, 2014.

J. Barzilai and J. M. Borwein. Two-point step size gradient methods. *IMA journal of numerical analysis*, 8(1):141–148, 1988.

M. E. Ben-Akiva and S. R. Lerman. *Discrete choice analysis: theory and application to travel demand*, volume 9. MIT press, 1985.

A. R. Benson, R. Kumar, and A. Tomkins. A discrete choice model for subset selection. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pages 37–45. ACM, 2018.

S. Bhojanapalli and P. Jain. Universal matrix completion. *arXiv preprint arXiv:1402.2324*, 2014.

J. Blanchet, G. Gallego, and V. Goyal. A markov chain approximation to choice modeling. In *EC*, pages 103–104, 2013.

C. Borgs, J. Chayes, C. E. Lee, and D. Shah. Iterative collaborative filtering for sparse matrix estimation. *arXiv preprint arXiv:1712.00710*, 2017.

V. S. Borkar, N. Karamchandani, and S. Mirani. Randomized Kaczmarz for rank aggregation from pairwise comparisons. In *Information Theory Workshop (ITW), 2016 IEEE*, pages 389–393. IEEE, 2016.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1955.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Foundations of Computational Mathematics*, 9(6):717–772, 2009.

X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202. ACM, 2013.

Y. Chen and C. Suh. Spectral mle: Top-k rank aggregation from pairwise comparisons. In *International Conference on Machine Learning*, pages 371–380, 2015.

Y. Chen, J. Fan, C. Ma, and K. Wang. Spectral method and regularized mle are both optimal for top-k ranking. *arXiv preprint arXiv:1707.09971*, 2017.

F. Chierichetti, R. Kumar, and A. Tomkins. Learning a mixture of two multinomial logits. In *International Conference on Machine Learning*, pages 960–968, 2018.

C. Chu, P. Leslie, and A. Sorensen. Bundle-size pricing as an approximation to mixed bundling. *The American Economic Review*, pages 263–303, 2011.

S. Clémençon, G. Lugosi, and N. Vayatis. Ranking and scoring using empirical risk minimization. In *International Conference on Computational Learning Theory*, pages 1–15. Springer, 2005.

M. A. Davenport, Y. Plan, E. van den Berg, and M. Wootters. 1-bit matrix completion. *Information and Inference*, 3(3):189–223, 2014.

P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.

M. Falahatgar, A. Jain, A. Orlitsky, V. Pichapati, and V. Ravindrakumar. The limits of maxing, ranking, and preference learning. In *International Conference on Machine Learning*, pages 1426–1435, 2018.

V. F. Farias, S. Jagabathula, and D. Shah. A data-driven approach to modeling choice. In *NIPS*, pages 504–512, 2009.

R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems*, pages 2973–2981, 2016.

D. F. Gleich and L.-h. Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 60–68. ACM, 2011.

K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 4(2):133–151, 2001.

I. C. Gormley and T. B. Murphy. A grade of membership model for rank data. *Bayesian Analysis*, 4(2):265–295, 2009.

P. M. Guadagni and J. D. Little. A logit model of brand choice calibrated on scanner data. *Marketing science*, 2(3):203–238, 1983.

B. Hajek, S. Oh, and J. Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems*, pages 1475–1483, 2014.

R. Heckel, M. Simchowitz, K. Ramchandran, and M. J. Wainwright. Approximate ranking from pairwise comparisons. *arXiv preprint arXiv:1801.01253*, 2018.

D. R. Hunter. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics*, pages 384–406, 2004.

P. Jain and S. Oh. Provable tensor factorization with missing data. preprint arXiv:1406.2784, 2014.

P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *STOC*, pages 665–674, 2013.

M. Jang, S. Kim, C. Suh, and S. Oh. Top-k ranking from pairwise comparisons: When spectral ranking is optimal. *arXiv preprint arXiv:1603.04153*, 2016.

M. Jang, S. Kim, C. Suh, and S. Oh. Optimal sample complexity of m-wise data for top-k ranking. In *Advances in Neural Information Processing Systems*, pages 1685–1695, 2017.

L. R. F. Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957.

S. Katariya, L. Jain, N. Sengupta, J. Evans, and R. Nowak. Adaptive sampling for coarse ranking. *arXiv preprint arXiv:1802.07176*, 2018.

J. Katz-Samuels and C. Scott. Nonparametric preference completion. *arXiv preprint arXiv:1705.08621*, 2017.

E. Kazemi, L. Chen, S. Dasgupta, and A. Karbasi. Comparison based learning from weak oracles. *arXiv preprint arXiv:1802.06942*, 2018.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *Information Theory, IEEE Transactions on*, 56(6):2980–2998, 2010a.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11(2057-2078):1, 2010b.

A. Khetan and S. Oh. Data-driven rank breaking for efficient rank aggregation. In *Proceedings of the 33rd International Conference on Machine Learning*, volume 48, 2016a.

A. Khetan and S. Oh. Computational and statistical tradeoffs in learning to rank. In *Advances in Neural Information Processing Systems 29*, pages 739–747, 2016b.

M. Ledoux. *The concentration of measure phenomenon*. American Mathematical Soc., 2005.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: isoperimetry and processes*. Springer Science & Business Media, 2013.

D. R. Luce. *Individual Choice Behavior*. Wiley, New York, 1959.

L. v. d. Maaten and G. Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

J. Marschak. Binary-choice constraints and random utility indicators. In *Proceedings of a symposium on mathematical methods in the social sciences*, volume 7, pages 19–38, 1960.

A. K. Massimino and M. A. Davenport. As you like it: Localization via paired comparisons. *arXiv preprint arXiv:1802.10489*, 2018.

L. Maystre and M. Grossglauser. Fast and accurate inference of Plackett–Luce models. In *Advances in Neural Information Processing Systems*, pages 172–180, 2015.

D. McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1973.

D. McFadden. Econometric models for probabilistic choice among products. *Journal of Business*, 53(3):S13–S29, 1980.

D. McFadden and K. Train. Mixed mnl models for discrete response. *Journal of applied Econometrics*, 15(5):447–470, 2000.

T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013a.

T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013b.

S. Negahban and M. J. Wainwright. Restricted strong convexity and (weighted) matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 2012.

S. Negahban, B. Yu, M. J. Wainwright, and P. K. Ravikumar. A unified framework for high-dimensional analysis of m-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *NIPS*, pages 2483–2491, 2012.

S. Oh and D. Shah. Learning mixed multinomial logit model from ordinal data. In *Advances in Neural Information Processing Systems*, pages 595–603, 2014.

G. Ongie, L. Balzano, D. Pimentel-Alarcón, R. Willett, and R. D. Nowak. Tensor methods for nonlinear matrix completion. *arXiv preprint arXiv:1804.10266*, 2018.

A. Pananjady, C. Mao, V. Muthukumar, M. J. Wainwright, and T. A. Courtade. Worst-case vs average-case design for estimation from fixed pairwise comparisons. *arXiv preprint arXiv:1707.06217*, 2017.

D. Park, J. Neeman, J. Zhang, S. Sanghavi, and I. S. Dhillon. Preference completion: Large-scale collaborative ranking from pairwise comparisons. *Proceedings of The 32nd International Conference on Machine Learning*, 2015.

R. L. Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

S. Ragain and J. Ugander. Pairwise choice Markov chains. In *Advances in Neural Information Processing Systems*, pages 3198–3206, 2016.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of The 31st International Conference on Machine Learning*, pages 118–126, 2014.

P. Ray. Independence of irrelevant alternatives. *Econometrica: Journal of the Econometric Society*, pages 987–991, 1973.

B. Recht, M. Fazel, and P. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

W. Rui, J. Xu, S. Rayadurgam, M. Lelarge, L. Massoulié, and B. Hajek. Clustering and inference from pairwise comparisons. In *SIGMETRICS'15 Proceedings of the 2015 ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems*, volume 43, page 2, 2015.

N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. Wainwright. When is it better to compare than to score? *arXiv preprint arXiv:1406.6618*, 2014.

N. B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J. Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *The Journal of Machine Learning Research*, 17(1):2049–2095, 2016a.

N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, 2016b.

P. Sham and D. Curtis. An extended transmission/disequilibrium test (tdt) for multi-allele marker loci. *Annals of human genetics*, 59(3):323–336, 1995.

O. Tamuz, C. Liu, S. Belongie, O. Shamir, and A. T. Kalai. Adaptively learning the crowd kernel. *arXiv preprint arXiv:1105.1033*, 2011.

L. L. Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

K. Train. *Qualitative choice analysis: Theory, econometrics, and an application to automobile demand*, volume 10. MIT press, 1986.

J. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Comput. Math.*, 2011.

J. A. Tropp. An introduction to matrix concentration inequalities. *arXiv preprint arXiv:1501.01571*, 2015.

A. Tsokos, S. Narayanan, I. Kosmidis, G. Baio, M. Cucuringu, G. Whitaker, and F. J. Király. Modeling outcomes of soccer matches. *arXiv preprint arXiv:1807.01623*, 2018.

S. Van De Geer. *Empirical Processes in M-estimation*, volume 6. Cambridge university press, 2000.

M. Vojnovic and S. Yun. Parameter estimation for generalized thurstone choice models. In *International Conference on Machine Learning*, pages 498–506, 2016.

J. Walker and M. Ben-Akiva. Generalized random utility model. *Mathematical Social Sciences*, 43:303–343, 2002.

M. Wilber, S. Kwak, and S. Belongie. Cost-effective HITs for relative similarity comparisons. In *Human Computation and Crowdsourcing (HCOMP)*, Pittsburgh, 2014. URL `http://arxiv.org/abs/1404.3291`.

M. Yuan and C. Zhang. On tensor completion via nuclear norm minimization. *arXiv preprint arXiv:1405.1773*, 2014.

E. Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1929.

Journal of Machine Learning Research 19 (2018) 1-42 Submitted 3/18; Published 9/18

Connections with Robust PCA and the Role of Emergent Sparsity in Variational Autoencoder Models

Bin Dai DAIB13@MAILS.TSINGHUA.EDU.CN
Institute for Advanced Study
Tsinghua University
Beijing, China

Yu Wang YW323@CAM.AC.UK
John Aston J.ASTON@STATSLAB.CAM.AC.UK
Department of Pure Mathematics and Mathematical Statistics
University of Cambridge
Cambridge, UK

Gang Hua GANGHUA@MICROSOFT.COM
Microsoft Research
Redmond, USA

David Wipf DAVIDWIPF@GMAIL.COM
Microsoft Research
Beijing, China

Editor: Sebastian Nowozin

Abstract

Variational autoencoders (VAE) represent a popular, flexible form of deep generative model that can be stochastically fit to samples from a given random process using an information-theoretic variational bound on the true underlying distribution. Once so-obtained, the model can be putatively used to generate new samples from this distribution, or to provide a low-dimensional latent representation of existing samples. While quite effective in numerous application domains, certain important mechanisms which govern the behavior of the VAE are obfuscated by the intractable integrals and resulting stochastic approximations involved. Moreover, as a highly non-convex model, it remains unclear exactly how minima of the underlying energy relate to original design purposes. We attempt to better quantify these issues by analyzing a series of tractable special cases of increasing complexity. In doing so, we unveil interesting connections with more traditional dimensionality reduction models, as well as an intrinsic yet underappreciated propensity for robustly dismissing sparse outliers when estimating latent manifolds. With respect to the latter, we demonstrate that the VAE can be viewed as the natural evolution of recent robust PCA models, capable of learning nonlinear manifolds of unknown dimension obscured by gross corruptions.

Keywords: Variational Autoencoder, Deep Generative Model, Robust PCA

1. Introduction

We begin with a data set $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^n$ composed of n i.i.d. samples of some random variable $\boldsymbol{x} \in \mathbb{R}^d$ of interest, with the goal of estimating a tractable approximation for $p_{\boldsymbol{\theta}}(\boldsymbol{x})$, knowledge of which would allow us to generate new samples of $\boldsymbol{x}$. Moreover we assume that each sample is governed by unobserved latent variables $\boldsymbol{z} \in \mathbb{R}^\kappa$, such that $p_{\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})p(\boldsymbol{z})d\boldsymbol{z}$, where $\boldsymbol{\theta}$ are the parameters defining the distribution we would like to estimate.

Given that this integral is intractable in all but the simplest cases, variational autoencoders (VAE) represent a powerful means of optimizing with respect to $\boldsymbol{\theta}$ a tractable upper bound on $-\log p_{\boldsymbol{\theta}}(\boldsymbol{x})$ (Kingma and Welling, 2014; Rezende et al., 2014). Once these parameters are obtained, we can then generate new samples from $p_{\boldsymbol{\theta}}(\boldsymbol{x})$ by first drawing some $\boldsymbol{z}^{(i)}$ from $p(\boldsymbol{z})$, and then a new $\boldsymbol{x}^{(i)}$ from $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}^{(i)})$. The VAE upper bound itself is constructed as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;=\; \sum_i \left\{ \mathbb{KL}\left[q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) || p_{\boldsymbol{\theta}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)\right] - \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}) \right\} \;\geq\; -\sum_i \log p_{\boldsymbol{\theta}}(\boldsymbol{x}^{(i)}), \tag{1}$$

where $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ defines an arbitrary approximating distribution, parameterized by $\boldsymbol{\phi}$, and $\mathbb{KL}\left[\cdot||\cdot\right]$ denotes the KL divergence between two distributions, which is always a non-negative quantity. For optimization purposes, it is often convenient to re-express this bound as

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;\equiv\; \sum_i \left(\mathbb{KL}\left[q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right) || p(\boldsymbol{z})\right] - \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{z}\right)\right]\right). \tag{2}$$

In these expressions, $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ can be viewed as an *encoder* model that defines a conditional distribution over the latent 'code' $\boldsymbol{z}$, while $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})$ can be interpreted as a *decoder* model since, given a code $\boldsymbol{z}$ it quantifies the distribution over $\boldsymbol{x}$.

By far the most common distributional assumptions are that $p(\boldsymbol{z}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ and the encoder model satisfies $q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \mathcal{N}(\boldsymbol{z}; \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z)$, where the mean $\boldsymbol{\mu}_z$ and covariance $\boldsymbol{\Sigma}_z$ are some function of model parameters $\boldsymbol{\phi}$ and the random variable $\boldsymbol{x}$. Likewise, for the decoder model we assume $p_{\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) = \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_x, \boldsymbol{\Sigma}_x)$ for continuous data, with means and covariances defined analogously.[1]

For arbitrarily parameterized moments $\boldsymbol{\mu}_z$, $\boldsymbol{\Sigma}_z$, $\boldsymbol{\mu}_x$, and $\boldsymbol{\Sigma}_x$, the KL divergence in (2) computes to

$$2\mathbb{KL}\left[q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})||p(\boldsymbol{z})\right] \;\equiv\; \mathrm{tr}\left[\boldsymbol{\Sigma}_z\right] + \|\boldsymbol{\mu}_z\|_2^2 - \log|\boldsymbol{\Sigma}_z|, \tag{3}$$

excluding irrelevant constants. However, the remaining integral from the expectation term admits no closed-form solution, making direct optimization over $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ intractable. Likewise, any detailed analysis of the underlying objective function becomes problematic as well.

At least for practical purposes, one way around this is to replace the troublesome expectation with a Monte Carlo stochastic approximation (Kingma and Welling, 2014; Rezende et al., 2014). More specifically we utilize

$$\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{z}\right)\right] \;\approx\; \tfrac{1}{\tau}\sum_{t=1}^{\tau} \log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{z}^{(i,t)}\right), \tag{4}$$

where $\boldsymbol{z}^{(i,t)}$ are samples drawn from $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$. Using a simple reparameterization trick, these samples can be constructed such that gradients with respect to $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ can be

1. For discrete data, a Bernoulli distribution is sometimes adopted instead.

propagated through the righthand side of (4). Therefore, assuming all the required moments $\boldsymbol{\mu}_z$, $\boldsymbol{\Sigma}_z$, $\boldsymbol{\mu}_x$, and $\boldsymbol{\Sigma}_x$ are differentiable with respect to $\boldsymbol{\phi}$ and $\boldsymbol{\theta}$, the entire model can be updated using SGD (Bottou, 2010).

While quite effective in numerous application domains that can apply generative models, e.g., semi-supervised learning (Kingma et al., 2014; Maaløe et al., 2016; Mansimov et al., 2016), certain important mechanisms which dictate the behavior of the VAE are obfuscated by the required stochastic approximation and the opaque underlying objective with high-dimensional integrals. Moreover, it remains unclear to what extent minima remain anchored at desirable locations in the non-convex energy landscape.

We take a step towards better quantifying such issues by probing the basic VAE model under a few simplifying assumptions of increasing complexity whereby closed-form integrations are (partially) possible. This process unveils a number of interesting connections with more transparent, established generative models, each of which shed light on how the VAE may perform under more challenging conditions. This mirrors the rich tradition of analyzing deep networks under various simplifications such as linear layers or i.i.d. random activation patterns (Choromanska et al., 2015a,b; Goodfellow et al., 2016; Kawaguchi, 2016; Saxe et al., 2014), and results in the following key contributions:

1. We demonstrate that the canonical form of the VAE, including the Gaussian distributional assumptions described above, harbors an innate agency for robust outlier removal in the context of learning inlier points constrained to a manifold of unknown dimension. In fact, when the decoder mean $\boldsymbol{\mu}_x$ is restricted to an affine function of $\boldsymbol{z}$, we prove that the VAE model collapses to a form of robust PCA (RPCA) (Candès et al., 2011; Chandrasekaran et al., 2011), a recently celebrated technique for separating data into low-rank (low-dimensional) inlier and sparse outlier components.[2]

2. We elucidate two central, albeit underappreciated roles of the VAE encoder covariance $\boldsymbol{\Sigma}_z$. First, through subtle multi-tasking efforts in both terms of (2), it facilitates learning the correct inlier manifold dimension. Secondly, $\boldsymbol{\Sigma}_z$ can help to smooth out undesirable minima in the energy landscape of what would otherwise resemble a more traditional deterministic autoencoder (AE) (Bengio, 2009). This is true even in certain situations where it provably does not actually alter the globally optimal solution itself. Note that prior to this work the AE could ostensibly be viewed as the most natural candidate for instantiating extensions of RPCA to handle outlier-robust nonlinear manifold learning. However, our results suggest that the VAE maintains pivotal advantages in mitigating the effects of bad local solutions and over-parameterized latent representations, even in completely deterministic settings that require no generative model per se.

As we will soon see, these points can have profound practical repercussions in terms of how VAE models are interpreted and deployed. For example, one immediate consequence is that even if the decoder capacity is not sufficient to capture the generative distribution *within* some fixed, unknown manifold, the VAE can nonetheless still often find the correct manifold itself, which is sufficient for deterministic recovery of uncorrupted inlier points.

2. RPCA represents a rather dramatic departure from vanilla PCA and is characterized by a challenging, combinatorial optimization problem. A formal definition will be provided in Section 3.

This is exactly analogous to RPCA recovery results, whereby it is possible to correctly estimate an unknown low-dimensional linear subspace heavily corrupted with outliers even if in doing so we do not obtain an actual generative model for the inliers within this subspace. We emphasize that this is *not* a job description for which the VAE was originally motivated, but a useful hidden talent nonetheless.

The remainder of this paper is organized as follows. In Section 2 we consider two affine decoder models and connections with past probabilistic PCA-like approaches. Note that the seminal work from (Rezende et al., 2014) mentions in passing that a special case of their VAE decoder model reduces to factor analysis (Bartholomew and Knott, 1999), a cousin of probabilistic PCA; however, no rigorous, complementary analysis is provided, such as how latent-space sparsity can emerge as we will introduce shortly. Next we examine various partially affine decoder models in Section 3, whereby only the mean $\boldsymbol{\mu}_x$ is affine while $\boldsymbol{\Sigma}_x$ has potentially unlimited complexity; all encoder quantities are likewise unconstrained. We precisely characterize how minimizers of the VAE cost, although not available in closed form, nonetheless are capable of optimally decomposing data into low-rank and sparse factors akin to RPCA while avoiding bad local optima. This section also discusses extensions as well as interesting behavioral properties of the VAE.

Section 4 then considers degeneracies in the full VAE model that can arise even with a trivially simple encoder and corresponding latent representation. Section 5 concludes with experiments that directly corroborate a number of interesting, practically-relevant hypotheses generated by our theoretical analyses, suggesting novel usages (unrelated to generating samples) as a tool for deterministic manifold learning in the presence of outliers. We provide final conclusions in Section 6. Note that our prior conference paper has presented the basic demonstration that VAE models can be applied to tackling generalized robust PCA problems (Wang et al., 2017). However this work primarily considers empirical demonstrations and high-level motivations, with minimal analytical support.

Notation: We use a superscript $^{(i)}$ to denote quantities associated with the i-th sample, which at times may correspond with the columns of a matrix, such as the data $\boldsymbol{X}$ or related. For a general matrix $\boldsymbol{M}$, we refer to the i-th row as $\boldsymbol{m}_{i\cdot}$ and the j-th column as $\boldsymbol{m}_{\cdot j}$. Although technically speaking posterior moments are functions of the parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$, the random variables $\boldsymbol{x}$, and the latent $\boldsymbol{z}$, i.e., $\boldsymbol{\mu}_x \equiv \boldsymbol{\mu}_x(\boldsymbol{z};\boldsymbol{\theta})$, $\boldsymbol{\Sigma}_x \equiv \boldsymbol{\Sigma}_x(\boldsymbol{z};\boldsymbol{\theta})$, $\boldsymbol{\mu}_z \equiv \boldsymbol{\mu}_z(\boldsymbol{x};\boldsymbol{\phi})$, and $\boldsymbol{\Sigma}_z \equiv \boldsymbol{\Sigma}_z(\boldsymbol{x};\boldsymbol{\phi})$, except in cases where some ambiguity exists regarding the arguments, these dependencies are omitted to avoid undue clutter; likewise for $\boldsymbol{\mu}_z^{(i)} \triangleq \boldsymbol{\mu}_z(\boldsymbol{x}^{(i)};\boldsymbol{\phi})$ and $\boldsymbol{\Sigma}_z^{(i)} \triangleq \boldsymbol{\Sigma}_z(\boldsymbol{x}^{(i)};\boldsymbol{\phi})$. Also, with some abuse of notation, we will use $\mathcal{L}$ to denote a number of different VAE-related objective functions and bounds, with varying arguments and context serving as differentiating factors. Finally, the diag[$\cdot$] operator converts vectors to a diagonal matrix, and vice versa as in the Matlab computing environment.

2. Affine Decoder and Probabilistic PCA

If we assume that $\boldsymbol{\Sigma}_x$ is fixed at some $\lambda \boldsymbol{I}$, and force $\boldsymbol{\Sigma}_z = \boldsymbol{0}$ (while removing the now undefined $\log|\boldsymbol{\Sigma}_z|$ term), then it is readily apparent that the resultant VAE model reduces to a traditional AE with squared-error loss function (Bengio, 2009), a common practical assumption. To see this, note that if $\boldsymbol{\Sigma}_z = \boldsymbol{0}$, then $q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$ collapses to $\delta(\boldsymbol{\mu}_z)$, i.e., a

delta function at the posterior mean, and $\mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{z}\right)\right] = \log p_{\boldsymbol{\theta}}\left(\boldsymbol{x}^{(i)}|\boldsymbol{\mu}_z^{(i)}\right)$, which is just a standard AE with quadratic loss and representation $\boldsymbol{\mu}_x\left(\boldsymbol{\mu}_z\left[\boldsymbol{x}\right]\right)$. Moreover, the only remaining (non-constant) regularization from the KL term is $\sum_i \|\boldsymbol{\mu}_z^{(i)}\|_2^2$. However, given scaling ambiguities that may arise in the decoder when $\boldsymbol{\Sigma}_z = \mathbf{0}$, $\boldsymbol{\mu}_z^{(i)}$ can often be made arbitrarily small, and therefore the effect of this quadratic penalty is infinitesimal. With affine encoder and decoder models, the resulting deterministic network will simply learn principal components like vanilla PCA, a well-known special case of the AE (Bourlard and Kamp, 1988).

Therefore to understand the VAE, it is crucial to explore the role of non-trivial selections for the encoder and decoder covariances, that serve as both enlightening and differentiating factors. As a step in this direction, we will explore several VAE reductions that lead to more manageable (yet still representative) objective functions and strong connections to existing probabilistic models. In this section we begin with the following simplification:

Lemma 1 *Suppose that the decoder moments satisfy $\boldsymbol{\mu}_x = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}$ and $\boldsymbol{\Sigma}_x = \lambda \boldsymbol{I}$ for some parameters $\boldsymbol{\theta} = \{\boldsymbol{W}, \boldsymbol{b}, \lambda\}$ of appropriate dimensions. Furthermore, we assume for the encoder we have $\boldsymbol{\mu}_z = f(\boldsymbol{x};\boldsymbol{\phi})$, $\boldsymbol{\Sigma}_z = \boldsymbol{S}_z\boldsymbol{S}_z^\top$, and $\boldsymbol{S}_z = g(\boldsymbol{x};\boldsymbol{\phi})$, where f and g are any parameterized functional forms that include arbitrary affine transformations for some arrangement of parameters. Under these assumptions, the objective from (2) admits optimal, closed-form solutions for $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ in terms of $\boldsymbol{W}$, $\boldsymbol{b}$, and λ such that the resulting VAE cost collapses to*

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{b}, \lambda) \;=\; \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}, \boldsymbol{b}, \lambda \boldsymbol{I}) + n \log \left| \lambda \boldsymbol{I} + \boldsymbol{W}\boldsymbol{W}^\top \right|, \tag{5}$$

where

$$\boldsymbol{\Omega}^{(i)}(\boldsymbol{W}, \boldsymbol{b}, \boldsymbol{\Psi}) \;\triangleq\; \left(\boldsymbol{x}^{(i)} - \boldsymbol{b}\right)^\top \left(\boldsymbol{\Psi} + \boldsymbol{W}\boldsymbol{W}^\top\right)^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{b}\right). \tag{6}$$

Additionally, if we enforce that off-diagonal elements of $\boldsymbol{\Sigma}_z$ must be equal to zero (i.e., $[\boldsymbol{\Sigma}_z]_{ij} = 0$ for $i \neq j$), then (5) further decouples/separates to

$$\mathcal{L}_{sep}(\boldsymbol{W}, \boldsymbol{b}, \lambda) \;=\; \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}, \boldsymbol{b}, \lambda \boldsymbol{I}) + n \left[\sum_j \log\left(\lambda + \|\boldsymbol{w}_{\cdot j}\|_2^2\right) + (d - \kappa) \log \lambda \right]. \tag{7}$$

All proofs are deferred to the appendices. The objective (5) is the same as that used by certain probabilistic PCA models (Tipping and Bishop, 1999), even though the latter is originally derived in a completely different manner. Moreover, it can be shown that any minimum of this objective represents a globally optimal solution (i.e, no minima with suboptimal objective function value exist). And with $\boldsymbol{b}$ and λ fixed, the optimal $\boldsymbol{W}$ will be such that span$[\boldsymbol{W}]$ equals the span of the singular vectors of $\boldsymbol{X} - \boldsymbol{b}\mathbf{1}^\top$ associated with singular values greater than $\sqrt{\lambda}$. So the global optimum produces a principal subspace formed by soft-thresholding the singular values of $\boldsymbol{X} - \boldsymbol{b}\mathbf{1}^\top$, with the rank one offset often used to normalize samples to have zero mean.[3]

3. While the details are omitted here, optimal solutions for both $\boldsymbol{b}$ and λ can be analyzed as well.

In contrast, the alternative cost (7), which arises from the oft-used practical assumption that $\boldsymbol{\Sigma}_z$ is diagonal, represents a rigorous upper bound to (5), since

$$\sum_j \log\left(\lambda + \|\boldsymbol{w}_{\cdot j}\|_2^2\right) + (d - \kappa)\log\lambda \;\geq\; \log\left|\lambda \boldsymbol{I} + \boldsymbol{W}\boldsymbol{W}^\top\right| \tag{8}$$

by virtue of Hadamard's inequality (see proof of Theorem 2 below), with equality iff $\boldsymbol{W}^\top\boldsymbol{W}$ is diagonal. Interestingly, all minima of the modified cost nonetheless retain global optimality of the original; however, it can be shown that there will be a combinatorial increase in the actual number of distinct (disconnected) minima:[4]

Theorem 2 *Let $\boldsymbol{R} \in \mathbb{R}^{\kappa\times\kappa}$ denote an arbitrary rotation matrix and $\boldsymbol{P} \in \mathbb{R}^{\kappa\times\kappa}$ an arbitrary permutation matrix. Furthermore let $\boldsymbol{W}^*$ be a minimum of (5) and $\boldsymbol{W}^{**}$ any minimum of (7) with $\boldsymbol{b}$ and λ fixed. Then the following three properties hold:*

1. $\mathcal{L}(\boldsymbol{W}^*, \boldsymbol{b}, \lambda) = \mathcal{L}(\boldsymbol{W}^*\boldsymbol{R}, \boldsymbol{b}, \lambda) = \mathcal{L}_{sep}(\boldsymbol{W}^{**}, \boldsymbol{b}, \lambda)$
$$= \;\; \mathcal{L}(\boldsymbol{W}^{**}\boldsymbol{P}, \boldsymbol{b}, \lambda) = \mathcal{L}_{sep}(\boldsymbol{W}^{**}\boldsymbol{P}, \boldsymbol{b}, \lambda). \tag{9}$$
2. *For any $\boldsymbol{W}^{**}\left(\boldsymbol{W}^{**}\right)^\top$ with distinct nonzero eigenvalues, there will exist at least $\frac{\kappa!}{(\kappa-r)!}$ distinct (disconnected) minima of (7) located at some $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{P}$, where $\boldsymbol{U}\boldsymbol{\Lambda}^2\boldsymbol{U}^\top$ represents the SVD of $\boldsymbol{W}^{**}\left(\boldsymbol{W}^{**}\right)^\top$ and $r = rank[\boldsymbol{W}^{**}]$.*
3. *$\boldsymbol{W}^{**}$ will have at most r nonzero columns, while $\boldsymbol{W}^*$ can have any number in $\{r, \ldots, \kappa\}$.*

Although this result applies to relatively simplistic affine decoders (the encoder need not be so constrained however), it nonetheless highlights a couple interesting principles. First, the diagonalization of $\boldsymbol{\Sigma}_z$ collapses the space of globally minimizing solutions to a subset of the original. While the consequences of this may be minor in the fully affine decoder model where all the solutions are still equally good, we surmise that with more sophisticated parameterizations this partitioning of the energy landscape into distinct basins-of-attraction could potentially introduce suboptimal local extrema. And from a broader perspective, Theorem 2 provides tangible validation of prior conjectures that variational Bayesian factorizations of this sort can fragment the space of local minima (Hoffman, 2014).

But there is a second, potentially-advantageous counter-affect elucidated by Theorem 2 as well. Specifically, even if $\boldsymbol{W}$ is overparameterized, meaning that κ is unnecessarily large, there exists an inherent mechanism to prune superfluous columns to exactly zero, i.e., column-wise sparsity. And once columns of $\boldsymbol{W}$ become sparse, the corresponding elements of $\boldsymbol{\mu}_z$ can no longer influence the data fit. Consequently, the $\|\boldsymbol{\mu}_z\|_2^2$ factor from (3) serves as the only relevant influence, pushing these values to be exactly zero even though ℓ_2 norms in most regularization contexts tend to favor diverse, *non-sparse* representations (Rao et al., 2003).

So ultimately, sparsity of $\boldsymbol{\mu}_z$ is an artifact of the diagonal $\boldsymbol{\Sigma}_z$ assumption and the interaction of multiple VAE terms, a subtle influence we empirically demonstrate translates to more complex regimes in Section 5. In any event, we have shown that both variants of the affine decoder model lead to reasonable probabilistic PCA-like objectives regardless of how overparameterized $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ happen to be.

4. By disconnected we mean that, to traverse from one minimum to another, we must ascend the objective function at some point along the way.

3. Partially Affine Decoder and Robust PCA

Thus far we have considered tight limitations on the complexity allowable in the functional forms of both $\boldsymbol{\mu}_x$ and $\boldsymbol{\Sigma}_x$, while $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ were free-range variables granted arbitrary flexibility. We now turn our gaze to the case where $\boldsymbol{\Sigma}_x$ can also be any parameterized, diagonal matrix[5] while $\boldsymbol{\mu}_x$ remains restricted. Although this administers considerable capacity to the model at the potential risk of overfitting, we will soon see that the VAE is nonetheless able to self-regularize in a very precise sense: Global minimizers of the VAE objective will ultimately correspond with optimal solutions to

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad n \cdot \text{rank}\,[\boldsymbol{L}] + \|\boldsymbol{S}\|_0, \qquad \text{s.t. } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}, \tag{10}$$

where $\|\cdot\|_0$ denotes the ℓ_0 norm, or a count of the number of nonzero elements in a vector or matrix. This problem represents the canonical form of robust principal component analysis (RPCA) (Candès et al., 2011; Chandrasekaran et al., 2011), decomposing a data matrix $\boldsymbol{X}$ into low-rank principal factors $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$, with $\boldsymbol{U}$ and $\boldsymbol{V}$ low-rank matrices of appropriate dimension, and a sparse outlier component $\boldsymbol{S}$. However, we must emphasize that (10), unlike traditional PCA, represents an NP-hard, discontinuous optimization problem with a combinatorial number of potentially bad local minima. Still, it is seemingly quite remarkable that the probabilistic VAE model shares any kinship with (10), even more so given that some of the distracting *local* minimizers can be smoothed away, a key VAE advantage as we will later argue.

Before elucidating this relationship, we require one additional technical caveat. Specifically, since $\log 0$ and $\frac{1}{0}$ are both undefined, and yet we will soon require an alliance with degenerate (or nearly so) covariance matrices that mimic the behavior of sparse and low-rank factors through log-det and inverse terms, we must place the mildest of restrictions on the minimal allowable singular values of $\boldsymbol{\Sigma}_x$ and $\boldsymbol{\Sigma}_z$. For this purpose we define $\mathcal{S}_\alpha^m$ as the set of $m \times m$ covariance matrices with singular values all greater than or equal to α, and likewise $\bar{\mathcal{S}}_\alpha^m$ as the subset of $\mathcal{S}_\alpha^m$ containing only diagonal matrices. We also define $\text{supp}_\alpha(\boldsymbol{x}) = \{i : |x_i| > \alpha\}$, noting that per this definition, $\text{supp}_0(\boldsymbol{x}) = \text{supp}(\boldsymbol{x})$, meaning we recover the standard definition of support: the set of indices associated with nonzero elements.

3.1 Main Result and Interpretation

Given the affine assumption from above, and the mild restriction $\boldsymbol{\Sigma}_x \in \bar{\mathcal{S}}_\alpha^d$ and $\boldsymbol{\Sigma}_z \in \mathcal{S}_\alpha^\kappa$ for some small $\alpha > 0$, the resulting constrained VAE minimization problem can be expressed as

$$\min_{\boldsymbol{\theta},\boldsymbol{\phi}} \quad \mathcal{L}\left(\boldsymbol{W}, \boldsymbol{b} = \boldsymbol{0}, \boldsymbol{\Sigma}_x \in \bar{\mathcal{S}}_\alpha^d, \boldsymbol{\mu}_z, \boldsymbol{\Sigma}_z \in \mathcal{S}_\alpha^\kappa\right), \tag{11}$$

where now $\boldsymbol{\theta}$ includes $\boldsymbol{W}$ as well as all the parameters embedded in $\boldsymbol{\Sigma}_x$, while $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$ are parameterized as in Lemma 1. We have also set $\boldsymbol{b} = \boldsymbol{0}$ merely for ease of presentation as its role is minor. We then have the following:

5. A full covariance over $\boldsymbol{x}$ is infeasible given the high dimension, and can lead to undesirable degeneracies anyway. Therefore a diagonal covariance is typically, if not always, used in practice.

Theorem 3 *Suppose that $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^n$ admits a feasible decomposition $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{V} + \boldsymbol{S}$ that uniquely[6] optimizes (10). Then for some $\bar{\alpha}$ sufficiently small, and all $\alpha \in (0, \bar{\alpha}]$, any global minimum $\{\hat{\boldsymbol{W}}, \hat{\boldsymbol{\Sigma}}_x, \hat{\boldsymbol{\mu}}_z, \hat{\boldsymbol{\Sigma}}_z\}$ of (11) will be such that*[7]

$$\textit{span}[\hat{\boldsymbol{W}}] = \textit{span}[\boldsymbol{U}] \qquad \textit{and} \qquad \textit{supp}_\alpha\left(\textit{diag}\left[\hat{\boldsymbol{\Sigma}}_x\left(\hat{\boldsymbol{\mu}}_z\left[\boldsymbol{x}^{(i)}\right]\right)\right]\right) = \textit{supp}[\boldsymbol{s}^{(i)}] \tag{12}$$

for all i provided that the latent representation satisfies $\kappa \geq \textit{rank}\,[\boldsymbol{U}]$.

Several important remarks are warranted here regarding the consequences and interpretation of this result:

- The $\hat{\boldsymbol{W}}$ satisfying (12) forms a linear basis for each inlier component $\boldsymbol{l}^{(i)}$, and likewise, a sample-dependent basis denoted $\boldsymbol{E}^{(i)}$ can be trivially constructed for each outlier component $\boldsymbol{s}^{(i)}$ using $\hat{\boldsymbol{\Sigma}}_x$, and $\hat{\boldsymbol{\mu}}_z$. Specifically, each unique column of $\boldsymbol{E}^{(i)}$ is a vector of zeros with a one in the j-th position, with $j \in \text{supp}_\alpha\left(\text{diag}\left[\hat{\boldsymbol{\Sigma}}_x\left(\hat{\boldsymbol{\mu}}_z\left[\boldsymbol{x}^{(i)}\right]\right)\right]\right)$. It follows that

  $$\boldsymbol{x}^{(i)} = \boldsymbol{l}^{(i)} + \boldsymbol{s}^{(i)} = \left[\hat{\boldsymbol{W}} \;\; \boldsymbol{E}^{(i)}\right]\left[\hat{\boldsymbol{W}} \;\; \boldsymbol{E}^{(i)}\right]^\dagger \boldsymbol{x}^{(i)}, \quad \forall i = 1, \ldots, n. \tag{13}$$

  Therefore if we can globally optimize the VAE objective, we can recover the correct latent representation, or equivalently, the optimal solution to (10).

- The requirements $\boldsymbol{\Sigma}_x \in \bar{\mathcal{S}}_\alpha^d$ and $\boldsymbol{\Sigma}_z \in \mathcal{S}_\alpha^\kappa$ do not portend the need for specialized tuning or brittleness of the result; these are merely technical conditions for dealing with degenerate covariances that occur near optimal solutions. While it might seem natural that $\boldsymbol{\Sigma}_x$ has diagonal elements pushed to zero in regions where near perfect data fit is possible, less intuitively, global optima of (11) can be achieved with an arbitrarily small $\boldsymbol{\Sigma}_z$, e.g., $\boldsymbol{\Sigma}_z = \alpha\boldsymbol{I}$, at least along latent dimensions needed to represent $\boldsymbol{L}$ (see proof construction). And interestingly, this implies that in areas surrounding a global optimum, the VAE objective can *resemble that of a regular AE*. As we will discuss more below, desirable smoothing effects of integration over $\boldsymbol{\Sigma}_z$ occur *elsewhere* in the energy landscape while preserving extrema anchored at the correct latent representation.

- Even if κ is large, meaning $\boldsymbol{W}$ is possibly overcomplete, the VAE will not overfit in the sense that there exists an inherent regulatory effect pushing $\text{span}[\boldsymbol{W}]$ towards $\text{span}[\boldsymbol{U}]$.

- If the globally optimal solution to (10) is not unique (this is different from uniqueness regarding the VAE objective), then a low-rank-plus-sparse model may not be the most reasonable, parsimonious representation of the data to begin with, and exact recovery of $\boldsymbol{L}$ and $\boldsymbol{S}$ will not be possible by *any* algorithm without further assumptions. More concretely, an arbitrary data point $\boldsymbol{x}^{(i)} \in \mathbb{R}^d$ requires d degrees of freedom to represent; however, if the data succinctly adheres to the RPCA model, then for properly chosen $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{S}$, we can have $\boldsymbol{x}^{(i)} = \boldsymbol{U}\boldsymbol{v}^{(i)} + \boldsymbol{s}^{(i)}$, where $\|\boldsymbol{v}^{(i)}\|_0 + \|\boldsymbol{s}^{(i)}\|_0 < d$. Arbitrary

6. Obviously only $\boldsymbol{L}$ and $\boldsymbol{S}$ will be unique; the actual decomposition of $\boldsymbol{L}$ into $\boldsymbol{U}$ and $\boldsymbol{V}$ is indeterminate up to an inconsequential invertible transform.

7. Although somewhat cumbersome in print, the expression $\hat{\boldsymbol{\Sigma}}_x\left(\hat{\boldsymbol{\mu}}_z\left[\boldsymbol{x}^{(i)}\right]\right)$ refers to $\hat{\boldsymbol{\Sigma}}_x$ evaluated at $\hat{\boldsymbol{\mu}}_z$, where the latter is evaluated at $\boldsymbol{x}^{(i)}$, the i-th sample.

data in general position will never admit such a unique decomposition, and we should only expect such structure in data well-represented by our VAE model, or the original RPCA predecessor from (10).

- A number of celebrated results have stipulated conditions (Candès et al., 2011; Chandrasekaran et al., 2011) whereby global solutions of the convex relaxation into nuclear and ℓ_1 norm components given by

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad \sqrt{n} \cdot \text{rank}\, \|\boldsymbol{L}\|_* + \|\boldsymbol{S}\|_1, \quad \text{s.t. } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}, \tag{14}$$

will equal global solutions of (10). While elegant in theory, and practically relevant given that (10) is discontinuous, non-convex, and difficult to optimize, the required conditions for this equivalence to hold place strong restrictions on the allowable structure in $\boldsymbol{L}$ and support pattern in $\boldsymbol{S}$. In practice these conditions can never be verified and are unlikely to hold, so an alternative modeling approach such as the VAE, which can be viewed as a smoothed version of (10) when an affine decoder mean is used (more on this later), remains attractive. Additionally, there is no clear way to modify (14) to handle nonlinear manifolds, which is obviously the bread and butter of the VAE.

We emphasize that these conclusions are not the product of an overly contrived situation, given that a significant restriction is only placed on $\boldsymbol{\mu}_x$; *all* other posterior quantities are essentially unconstrained provided a sufficient lower complexity bound is exceeded, implying that the result will hold whenever a sufficiently complex deep network is used. Moreover, although we will defer to a formal treatment to future work for purposes of brevity here, with some mild additional conditions, Theorem 3 can naturally be extended to the case where the decoder mean function is generalized to subsume non-linear, union-of-subspace models as commonly assumed in subspace clustering problems (Elhamifar and Vidal, 2013; Rao et al., 2010). This then deviates substantially from any direct PCA-kinship, and buttresses the argument that the analysis presented here transitions to broader scenarios. The experiments from Section 5 will also provide complementary empirical confirmation.

Moving forward, as a point of further comparison it is also interesting to examine how a traditional AE, which emerges when $\boldsymbol{\Sigma}_z$ is forced to zero, behaves under analogous conditions to Theorem 3.

Corollary 4 *Under the same conditions as Theorem 3, if we remove the* $\log|\boldsymbol{\Sigma}_z|$ *term and assume* $\boldsymbol{\Sigma}_z = \boldsymbol{0}$ *elsewhere, then (11) admits a closed-form solution for* $\boldsymbol{\Sigma}_x$ *in terms of* $\boldsymbol{W}$ *and* $\boldsymbol{\mu}_z$ *such that minimizers of the VAE cost are minimizers of*

$$\mathcal{L}\left(\boldsymbol{W}, \boldsymbol{\mu}_z\right) = \sum_i \left\| \boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z\left(\boldsymbol{x}^{(i)}\right) \right\|_0 \quad \textit{in the limit } \alpha \to 0. \tag{15}$$

From this result we immediately observe that, provided $\boldsymbol{\mu}_z$ enjoys a sufficiently rich parameterization, minimization of (15) is just a constrained version of (10), exactly equivalent to solving

$$\min_{\boldsymbol{L},\boldsymbol{S}} \quad \|\boldsymbol{S}\|_0, \quad \text{s.t. } \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}, \;\; \text{rank}\,[\boldsymbol{L}] \leq \kappa. \tag{16}$$

This expression immediately exposes one weakness of the AE; namely, if κ is too large, there is no longer any operation in place to prune away unnecessary dimensions, and the trivial solution $\boldsymbol{L} = \boldsymbol{X}$ will be produced. In the large-κ regime then, global VAE and global AE solutions do in fact deviate, ultimately because of the removal of the $-\log|\boldsymbol{\Sigma}_z|$ term in the latter. So $\boldsymbol{\Sigma}_z$ plays a critical role in determining the correct, low-dimensional inlier structure, and ultimately it is this covariance that chaperons $\boldsymbol{W}$ during the learning process.

3.2 Additional Local Minima Smoothing Effects

There is also a more important, yet subtle, advantage of the VAE over both (16) and the original unconstrained RPCA model from (10). For both RPCA constructions, any feasible support pattern, even the trivial ones associated with non-interesting decompositions satisfying $\|\boldsymbol{v}^{(i)}\|_0 + \|\boldsymbol{s}^{(i)}\|_0 \geq d$ for some i, will necessarily represent a local minimum, since there is an infinite gradient to overcome to move from a zero-valued element of $\boldsymbol{S}$ to a nonzero one.

Unlike these deterministic approaches, the behavior of the VAE reflects a form of differential smoothing that rids the model of many of these pitfalls while retaining desirable minima that satisfy (12).[8] Based on details of the proof of Theorem 3, it can be shown that, excluding small-order terms dependent on other variables and a constant scale factor of $-\log\alpha$, then a representative bound on the VAE objective associated with each sample index i behaves like

$$\text{rank}[\boldsymbol{W}] + \text{supp}_\alpha\left(\text{diag}\left[\boldsymbol{\Sigma}_x\left(\boldsymbol{\mu}_z\left[\boldsymbol{x}^{(i)}\right]\right)\right]\right). \tag{17}$$

But crucially, *this behavior lasts only as long as (17) is strictly less than* d and $\boldsymbol{\Sigma}_z$ is forced to be small or degenerate. In contrast, when the value is at or above d, (17) no longer reflects the energy function, which becomes relatively flat because of smoothing via $\boldsymbol{\Sigma}_z$, avoiding the pitfalls described above. This phenomena then has the potential to smooth out a large constellation of bad locally optimal solutions.

To situate things in the narrative of (10), which is useful for illustration purposes, the VAE can be viewed (at least to first order approximation) as minimizing the alternative lower-bounding objective function

$$\begin{aligned} \sum_i \text{rank}\left[\boldsymbol{L}\boldsymbol{L}^\top + \text{diag}\left(\boldsymbol{s}^{(i)}\right)^2\right] &\leq \sum_i \text{rank}\left[\boldsymbol{L}\boldsymbol{L}^\top\right] + \sum_i \text{rank}\left[\text{diag}\left(\boldsymbol{s}^{(i)}\right)^2\right] \\ &= n \cdot \text{rank}\,[\boldsymbol{L}] + \|\boldsymbol{S}\|_0, \end{aligned} \tag{18}$$

or a smooth surrogate thereof, over the constraint set $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$. The advantages of this lower bound are substantial: As long as a unique solution exists to the RPCA problem, the globally optimal solution with $\|\boldsymbol{v}^{(i)}\|_0 + \|\boldsymbol{s}^{(i)}\|_0 < d$ for all i will be unchanged; however, *any* feasible solution with $\|\boldsymbol{v}^{(i)}\|_0 + \|\boldsymbol{s}^{(i)}\|_0 \geq d$ will have a constant cost via the expression on the left of the inequality, truncating the many erratic peaks that will necessarily occur with the energy on the righthand side.

8. A more rudimentary form of this smoothing has been observed in much simpler empirical Bayesian models derived using Fenchel duality (Wipf, 2012).

In fact, away from the strongly attractive basins of optimal VAE solutions, the KL term from (2) is likely to push $\mathbf{\Sigma}_z$ more towards

$$\arg\min_{\mathbf{\Sigma}_z \succ \mathbf{0}} \mathbb{KL}\left[q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}\right)||p(\boldsymbol{z})\right] \quad \equiv \quad \arg\min_{\mathbf{\Sigma}_z \succ \mathbf{0}} \operatorname{tr}\left[\mathbf{\Sigma}_z\right] - \log|\mathbf{\Sigma}_z| \quad = \quad \boldsymbol{I}. \tag{19}$$

Experiments presented in Section 5 confirm that this is indeed the case. And once $\mathbf{\Sigma}_z$ moves away from zero, it will generally contribute a strong smoothing effect via the expectation in (2). However, there exists an important previously unobserved caveat here: If the decoder mean function is excessively complex, it can potentially outwit all regulatory persuasions from $\mathbf{\Sigma}_z$, leading to undesirable degenerate solutions with no representational value as described next.

4. Degeneracies Arising from a Flexible Decoder Mean

In this section we consider the case where $\boldsymbol{\mu}_x$ is finally released from its affine captivity to join with posterior colleagues in the wild. That simultaneously granting $\boldsymbol{\mu}_x$, $\mathbf{\Sigma}_x$, $\boldsymbol{\mu}_z$, and $\mathbf{\Sigma}_z$ unlimited freedom leads to overfitting may not come as a surprise; however, it turns out that even if the latter three are severely constrained, overfitting will not be avoided when $\boldsymbol{\mu}_x$ is over-parameterized in a certain sense extending beyond a single affine layer. This is because, at least at a high level, the once-proud regulatory effects of $\mathbf{\Sigma}_z$ can be completely squashed in these situations leading to the following:

Theorem 5 *Suppose $\kappa = 1$ (i.e., a latent dimension of only one), $\mathbf{\Sigma}_z \equiv \sigma_z^2 = \lambda_z$ (a scalar), $\mu_z = \boldsymbol{a}^\top \boldsymbol{x}$ for some fixed vector $\boldsymbol{a}$, $\mathbf{\Sigma}_x = \lambda_x \boldsymbol{I}$, and $\boldsymbol{\mu}_x$ is an arbitrary piecewise linear function with n segments. Then the VAE objective is unbounded from below at a trivial solution $\{\hat{\lambda}_z, \hat{\boldsymbol{a}}, \hat{\lambda}_x, \hat{\boldsymbol{\mu}}_x\}$ such that the resulting posterior mean $\hat{\boldsymbol{\mu}}_x(z;\boldsymbol{\theta})$ will satisfy $\hat{\boldsymbol{\mu}}_x(z;\boldsymbol{\theta}) \in \{\boldsymbol{x}^{(i)}\}_{i=1}^n$ with probability one for any z.*

In this special case, $\mathbf{\Sigma}_x$, σ_z^2, and μ_z are all simple affine functions and the latent dimension is minimal, and yet an essentially useless, degenerate solution can arbitrarily optimize the VAE objective. This occurs because the VAE has limited power to corral certain types of heavily over-parameterized decoder mean functions, even when all other degrees of freedom are constrained, and in this regime the VAE essentially has no advantage over a traditional autoencoder (its natural self-regulatory agency may sometimes break down). In contrast, as we saw in a previous section, there is no problem taming the influences of an unlimited latent representation (meaning κ is large, e.g., even $\kappa > n$) and its huge, attendant parameterized mean function, provided the latter is affine, as in $\boldsymbol{\mu}_x = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}$.

Indeed then, the issue is clearly not the *degree* of over-parameterization in $\boldsymbol{\mu}_x$ per se, but the actual structures in place. And the key problem is that, at least in some situations, the model can circumvent the entire regulatory mechanism of the KL term, pushing the latent variances towards zero even around *undesirable* solutions. For example, in the context of Theorem 5, the piecewise linear structure of $\boldsymbol{\mu}_x$ allows the decoder to act much like a vector quantization process, encouraging z towards a scalar code that selects for piecewise linear segments matched to training samples $\boldsymbol{x}^{(i)}$. And because this will lead to perfect reconstruction error if an optimal segment is found for a particular $z^{(i)}$, $\mathbf{\Sigma}_x = \lambda_x \boldsymbol{I} \approx \mathbf{0}$ serves as a reasonable characterization of posterior uncertainty, pushing $p(\boldsymbol{x}^{(i)}|z^{(i)}) \rightarrow \delta\left(\boldsymbol{x}^{(i)}\right)$ provided that $z^{(i)} \approx \mu_z\left(\boldsymbol{x}^{(i)}; \boldsymbol{a}\right) = \boldsymbol{a}^\top \boldsymbol{x}^{(i)}$, meaning that $\sigma_z^2 = \lambda_z$ is not too large.

In this situation, loosely speaking the data term from (2) will behave like $nd\log\lambda_x$, bullying the over-matched KL term that will scale only as $-n\log\lambda_z$. This in turn leads to a useless, degenerate solution as $\lambda_x = \lambda_z \rightarrow 0$, either for the purposes of generating representative samples, or for outlier removal as we have described herein.

One helpful caveat though, is that actually implementing such a complex piecewise linear function $\hat{\boldsymbol{\mu}}_x(z;\boldsymbol{\theta})$ using typical neural network components would require extremely wide and/or deep structure beyond the first decoder mean layer. And the degrees of freedom in such higher-layer structures would need to scale proportionally with the size of the training data, which is not a practical VAE operational regime to begin with. In contrast, the first layer of the decoder mean network more or less self-regularizes, at least in the affine and related cases as described above. And we conjecture that this self-regularization preserves in more complex networks of reasonable practical size as will be empirically demonstrated in Section 5. So really it is *excessive* complexity in higher decoder mean layers, unrelated to the dimensionality of the latent $\boldsymbol{z}$ bottleneck, where overfitting problems are more likely to arise.

Of course an analogous issue exists with generative adversarial networks (GAN) as well, a popular competing deep generative model composed of a generator network analogous to the VAE decoder, and a discriminator network that replaces the VAE encoder in a loose sense (Goodfellow et al., 2014). If the generator network merely learns a segmentation of $\boldsymbol{z}$-space such that all points in the i-th partition map to $\boldsymbol{x}^{(i)}$, the discriminator will be helpless to avert this degenerate situation even in principle. But there is an asymmetry when it comes to the GAN discriminator network and the VAE encoder: Over-parameterization of the former can be problematic (e.g., it can easily out-wit an affine or other proportionally simple generator), but the latter not so, at least in the sense that a highly flexible VAE encoder need not bully a simple decoder into trivial solutions as we have shown in previous sections.

5. Experiments and Analysis

Theoretical examination of simplified cases can be viewed as a powerful vehicle for generating accessible hypotheses that describe likely behavior in more realistic, practical situations. In this section we empirically evaluate and analyze three concrete hypotheses that directly emanate from our previous technical results and the tight connections between RPCA and VAE models. In aggregate, these hypotheses have wide-ranging consequences in terms of how VAEs should be applied and interpreted.

Before stating these hypotheses, we summarize what can be viewed as two, theoretically-accessible boundary cases considered thus far. First, building on Section 2, Section 3 demonstrated that the VAE can self-regularize and produce useful, robust models provided that restrictions are placed on only the decoder mean network. Conversely, Section 4 demonstrated that, regardless of other model components, if the decoder mean network is unreasonably complex beyond the first layer, then overfitting emerges as a potential concern. But between these two extremes, there exists a large operational regime whereby practical VAE behavior is both worth exploring and likely still informed by the original analysis of these boundary cases.

Within this context then, we conjecture that the desirable VAE properties exposed in Sections 2 and 3 are inherited by models involving deeper decoder mean networks, but at least constrained to practically-sized hidden-layer $\boldsymbol{\mu}_x$ complexity such that the concerns from Section 4 are not a significant factor (e.g., no networks where the degrees of freedom in higher decoder mean layers scales as $d \times n$, an absurd VAE structure by any measure). More specifically, in this section will empirically examine the following three hypotheses:

(i) When the decoder mean function is allowed to have multiple hidden layers of sensible size/depth, the VAE should behave like a nonlinear extension of RPCA, but with natural regularization effects in place that help to avoid local minima and/or overfitting to outliers. It is therefore likely to outperform either RPCA algorithms or, more importantly, an AE on diverse manifold recovery/outlier discovery problems unrelated to the probabilistic generative modeling tasks the VAE was originally designed for.

(ii) If the VAE latent representation $\boldsymbol{z}$ is larger than needed (meaning its dimension κ is higher than the true data manifold dimension), we have proven that unnecessary columns of $\boldsymbol{W}$ in a certain affine decoder mean model $\boldsymbol{\mu}_x = \boldsymbol{W}\boldsymbol{z} + \boldsymbol{b}$ will automatically be pruned as desired. Analogously, in the extended nonlinear case we would then expect that columns of the weight matrix from the first layer of the decoder mean network should be pushed to zero, again effectively pruning away the impact of any superfluous elements of $\boldsymbol{z}$.

(iii) When granted sufficient capacity in both $\boldsymbol{\mu}_x\,(\boldsymbol{\mu}_z\,[\boldsymbol{x}])$ and $\boldsymbol{\Sigma}_x$ to model inliers and outliers respectively, the VAE should have a tendency to push elements of the encoder covariance $\boldsymbol{\Sigma}_z$ to arbitrarily near zero along latent dimensions needed for representing inlier points, *selectively* overriding the KL regularizer that would otherwise push these values towards one. This counterintuitive behavior directly facilitates the VAE's utility as a nonlinear outlier removal tool (per Hypothesis (i)) by preserving exact adherence to the manifold in the neighborhood of optimal solutions.

5.1 Hypothesis (i) Evaluation Using Specially-Designed Ground-Truth Manifolds

If our theory is generally applicable, then a VAE with suitable parameterization should be able to significantly outperform an analogous deterministic AE (i.e., an equivalent VAE but with $\boldsymbol{\Sigma}_z = \boldsymbol{0}$) on the task of recovering data points drawn from a low-dimensional nonlinear manifold, but corrupted with gross outliers. In other words, even if both models have equivalent capacity to capture the intrinsic underlying manifold in principle, the VAE is more likely to avoid bad minima and correctly estimate it. We demonstrate this VAE capability here for the first time across an array of manifold dimensions and corruption percentages, recreating a nonlinear version of what are commonly termed *phase transition plots* in the vast RPCA literature (Candès et al., 2011; Ding et al., 2011; Kim et al., 2013; Wipf, 2012). These plots evaluate the reconstruction quality of competing algorithms for every pairing of subspace dimension and outlier ratio, creating a heat map that differentiates success and failure regions.

Of course explicit knowledge of ground-truth low-dimensional manifolds is required to accomplish this. With linear subspaces it is trivial to generate appropriate synthetic data

by simply creating two low-rank random matrices $\boldsymbol{U} \in \mathbb{R}^{d\times\kappa}$ and $\boldsymbol{V} \in \mathbb{R}^{\kappa\times n}$, a sparse outlier matrix $\boldsymbol{S}$, and then computing $\boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}$ with $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$. Algorithms are presented with only $\boldsymbol{X}$ and attempt to reconstruct $\boldsymbol{L}$. Here we generalize this process to the nonlinear regime using deep networks and the following non-trivial steps. In this revised context, the generated $\boldsymbol{L}$ will now represent a data matrix with columns confined to a ground-truth nonlinear manifold.

Data Generation: First we draw n low-dimensional samples $\boldsymbol{z}^{(i)} \in \mathbb{R}^{\kappa}$ from $\mathcal{N}(\boldsymbol{z}; \boldsymbol{0}, \boldsymbol{I})$ and pass them through a 3-layer network with ReLU activations (Nair and Hinton, 2010). We express this structure as $\boldsymbol{z}(\kappa)$-$\boldsymbol{D}_1(r_1)$-$\boldsymbol{D}_2(r_2)$-$\boldsymbol{l}(d)$, where $\boldsymbol{D}_1$ and $\boldsymbol{D}_2$ are hidden layers, $\boldsymbol{l}$ here serves as the output layer, and the values inside parentheses denote the respective dimensionalities (these experiment-dependent values will be discussed later). Network weights are set using the initialization procedure from (He et al., 2015). The d-dimensional output produced by $\boldsymbol{z}^{(i)}$ is denoted as $\boldsymbol{l}^{(i)}$, the collection of which form a matrix $\boldsymbol{L}$, with columns effectively lying on a κ-dimensional nonlinear manifold. This network can be viewed as a *ground-truth decoder*, projecting $\boldsymbol{z}^{(i)}$ to clean samples $\boldsymbol{l}^{(i)}$.

But we must also verify that there exists a known *ground-truth encoder* that can correctly invert the decoder, otherwise we cannot be sure that any given VAE structure provably maintains an optimal encoder within its capacity (this is very unlike the linear RPCA case where an analogous condition is trivially satisfied). To check this, we learn the requisite inverse mapping by training something like an inverted autoencoder. Basically, the decoder described above now acts as an encoder, to which we append a new 3-layer ReLU network structured as $\boldsymbol{l}(d)$-$\boldsymbol{E}_1(r_2)$-$\boldsymbol{E}_2(r_1)$-$\hat{\boldsymbol{z}}(\kappa)$, where now $\boldsymbol{E}_1$ and $\boldsymbol{E}_2$ denote candidate hidden layers for a potentially optimal encoder. The entire intverted structure then becomes $\boldsymbol{z}(\kappa)$-$\boldsymbol{D}_1(r_1)$-$\boldsymbol{D}_2(r_2)$-$\boldsymbol{l}(d)$-$\boldsymbol{E}_1(r_2)$-$\boldsymbol{E}_2(r_1)$-$\hat{\boldsymbol{z}}(\kappa)$. If any $\boldsymbol{z}^{(i)}$ passes through this network with zero reconstruction error, it implies that the corresponding $\boldsymbol{l}^{(i)}$ can pass through the flipped network with zero reconstruction error, and we have verified our complete ground truth network.

We could train the entire system end-to-end to accomplish this, which should be easy since $\kappa \ll d$; however, we found that although $\boldsymbol{z}^{(i)} = \hat{\boldsymbol{z}}^{(i)}$ is obviously not difficult to achieve, the corresponding learned samples $\boldsymbol{l}^{(i)}$ are pushed to very near a low-rank matrix when assembled into $\boldsymbol{L}$. This would imply that non-linear manifold learning is not actually even required and RPCA would likely be sufficient.

To circumvent this issue, we instead hold the initial $\boldsymbol{z}(\kappa)$-$\boldsymbol{D}_1(r_1)$-$\boldsymbol{D}_2(r_2)$-$\boldsymbol{l}(d)$ structure fixed, which ensures that the rank of $\boldsymbol{L}$ cannot be altered, and only train the second half using a standard ℓ_2 loss. In doing so we are able to obtain an $\boldsymbol{L}$ matrix, extracted from the middle layer, that is both (a) *not* well-represented by a low-rank approximation, and (b) *does* lie on a *known* low-dimensional non-linear manifold. And given that essentially zero reconstruction error is in fact achievable (up to the expected small ripples introduced by stochastic gradient descent or a similar surrogate), the learned decoder from this process implicitly serves as the ground-truth encoder underlying the data structure. Hence any VAE that includes $\boldsymbol{l}(d)$-$\boldsymbol{E}_1(r_2)$-$\boldsymbol{E}_2(r_1)$-$\hat{\boldsymbol{z}}(\kappa)$ within its encoder mean network capacity, as well as $\boldsymbol{z}(\kappa)$-$\boldsymbol{D}_1(r_1)$-$\boldsymbol{D}_2(r_2)$-$\boldsymbol{l}(d)$ within its decoder mean network capacity, will at least in principle have the capability of zero reconstruction error as well.

Finally, once $\boldsymbol{L}$ has been created in this manner, we then generate the noisy data matrix $\boldsymbol{X}$ by randomly corrupting $100 \cdot \nu\%$ of the entries, replacing the original value with samples from a standardized Gaussian distribution. In doing so, the original 'signal' component from $\boldsymbol{L}$ is completely swamped out at these locations.

Experimental Design: Given a data matrix $\boldsymbol{X}$ as generated above, we test the relative performance of four competing models:

1. VAE: We form a VAE architecture with the cascaded encoder/decoder mean networks $\boldsymbol{\mu}_x(\boldsymbol{\mu}_z[\boldsymbol{x}])$ assembled as $\boldsymbol{x}(100)$-$\boldsymbol{E}_1(2000)$-$\boldsymbol{E}_2(1000)$-$\boldsymbol{\mu}_z(50)$-$\boldsymbol{D}_1(1000)$-$\boldsymbol{D}_2(2000)$-$\boldsymbol{\mu}_x(100)$. This mirrors the high-level structure used to generate the outlier-free data, and ultimately will ensure that the ground-truth manifold is included within the network parameterization. Consistent with the design in (Kingma and Welling, 2014), a diagonal encoder covariance $\boldsymbol{\Sigma}_z$ is produced by sharing just the first two mean network layers. An exponential layer is also appended at the output to produce non-negative values. For consistency with AE models, the decoder covariance $\boldsymbol{\Sigma}_x$ is addressed separately via a special process described below.

2. AE-ℓ_2: We begin with the VAE model from above and fix $\boldsymbol{\Sigma}_z = \boldsymbol{0}$. This reduces the KL regularization term from (3) to simply $\|\boldsymbol{\mu}_z\|_2^2$. If no other changes are included, then the scaling ambiguity between $\boldsymbol{\mu}_z$ and decoder layer $\boldsymbol{D}_1$ is such that $\boldsymbol{\mu}_z$ can be made arbitrarily small without any loss of generality, rendering any beneficial regularization effect from $\|\boldsymbol{\mu}_z\|_2^2$ completely moot as discussed at the beginning of Section 2. Therefore we add a standard weight decay term to the AE-ℓ_2 network parameters $\{\boldsymbol{\theta}, \boldsymbol{\phi}\}$ to ameliorate this scaling ambiguity, which is tantamount to including an additional penalty factor $C_1\|\{\boldsymbol{\theta}, \boldsymbol{\phi}\}\|_2^2$. We also balance $\|\boldsymbol{\mu}_z\|_2^2$ with a second tuning parameter C_2, i.e., $C_2\|\boldsymbol{\mu}_z\|_2^2$. For the experiments in this section, we choose $C_1 = 0.0005$, a typical default value for weight decay, and then tune C_2 for optimal performance.[9]

   Note also that once $\boldsymbol{\Sigma}_z = \boldsymbol{0}$, at every sample $\boldsymbol{\Sigma}_x$ can be solved for in closed form as $\left[\boldsymbol{\Sigma}_x^{(i)}\right]_{jj} = \left(x_j^{(i)} - \mu_{x_j}^{(i)}\right)^2$ for $j = 1, \ldots, d$ assuming sufficient capacity per Corollary 4. We then plug this value into the AE-ℓ_2 cost, effectively optimizing $\boldsymbol{\Sigma}_x^{(i)}$ out of the model altogether making it entirely deterministic. For direct comparison, we apply the same procedure to the VAE from above, which can be interpreted as efficiently modeling the infinite capacity limit for $\boldsymbol{\Sigma}_x$ (i.e., even with infinite capacity in $\boldsymbol{\Sigma}_x$, the VAE model could do no better than this).

3. AE-ℓ_1: To explicitly encourage sparse latent representations, which could potentially be helpful in learning the correct manifold dimension, we begin with the AE-ℓ_2 model from above and replace $\|\boldsymbol{\mu}_z\|_2^2$ with the ℓ_1 norm $\|\boldsymbol{\mu}_z\|_1$, a well-known sparsity-promoting penalty function (Donoho and Elad, 2003). The corresponding parameter C_2 is likewise independently tuned for optimal performance.

4. RPCA: As an additional baseline, we also apply the convex RPCA formulation from (14) to the same corrupted data. This model is implemented via an augmented Lagrangian method using code from (Lin et al., 2010).

For the VAE, AE-ℓ_2, and AE-ℓ_1 networks, all model weights were randomly initialized so as not to copy any information from the ground-truth template. Training was conducted over 200 epochs using the Adam optimization technique (Kingma and Ba, 2014) with a learning rate of 0.0001 and a batch size of 100. We chose $n = 10^6$ training samples for each separate experiment, across which we varied the manifold dimension from $\kappa = 2, 4, \ldots, 20$ while the outlier ratio ranged as $\nu = 0.05, 0.10, \ldots, 0.50$. For each pair of experimental conditions, we train/run all four models and measure performance recovering the true $\boldsymbol{L}$ as quantified by the normalized MSE metric

$$\text{NMSE} \triangleq \|\boldsymbol{L} - \hat{\boldsymbol{L}}\|_{\mathcal{F}}^2 / \|\boldsymbol{L}\|_{\mathcal{F}}^2. \tag{20}$$

Note that although in practice we will not generally know the true manifold dimension κ in advance, because we choose $\dim[\boldsymbol{\mu}_z] = 50 > \kappa$ when constructing encoder networks for all experiments, perfect reconstruction is still theoretically possible by any of the VAE or AE models provided that outlier contributions can be successfully mitigated.

Results: Figure 1 displays the results estimating $\boldsymbol{L}$, where the VAE outperforms RPCA and the AE models by a wide margin. Perhaps most notably, the VAE performance dominates both AE-ℓ_1 and AE-ℓ_2, supporting our theory that the smoothing effect of integrating over $\boldsymbol{\Sigma}_z$ has immense practical value in avoiding bad minimizing solutions through its unique form of differential regularization. In fact, the AE-ℓ_2 objective is identical to the VAE once $\boldsymbol{\Sigma}_z = \boldsymbol{0}$, at least up to the constant C_2 applied to $\|\boldsymbol{\mu}_z\|_2^2$ which is only tuned to benefit the former while remaining fixed for the latter.[10] So this smoothing effect is essentially the *only* difference between the VAE and AE-ℓ_2 models, and therefore, Figures 1(a) and 1(b) truly isolate the benefits of the VAE in this regard.

To summarize then, by design all VAE and AE network structures are equivalent in terms of their predictive capacity, but only the VAE is able to capitalize on the regularizing effect of $\boldsymbol{\Sigma}_z$ to actually reach a good solution in challenging conditions. Moreover, this is even possible without the hassle of tuning tedious hyperparameters to balance regularization effects as required by AE-ℓ_1 and AE-ℓ_2 models.[11] This confirms Hypothesis (i) and suggests that VAEs are a viable candidate for replacing existing RPCA algorithms (Candès et al., 2011; Ding et al., 2011; Kim et al., 2013; Wipf, 2012) in regimes where a single linear

9. For direct comparison, we include the same weight decay factor $C_1\|\{\boldsymbol{\theta}, \boldsymbol{\phi}\}\|_2^2$ with the VAE model even though there is no equivalent issue with scaling ambiguity. In fact, this can be viewed as an advantage of the VAE regularization mechanism, in that it directly prevents large decoder weights from compensating for arbitrarily small values of $\boldsymbol{\mu}_z$, killing regularization effects. This is because there exists a key dependence between the weights from $\boldsymbol{D}_1$ and the covariance $\boldsymbol{\Sigma}_z$ such that any large weights that would accommodate pushing $\boldsymbol{\mu}_z$ towards zero would equally *amplify* the random additive noise coming from the stochastic encoder model, nullifying any benefit to the overall cost.

10. If $C_2 = 1$, the default value as produced by the VAE KL term, the AE-ℓ_2 performance is much worse (not shown) than when using the tuned value of $C_2 = 10^3$ as was adopted in producing Figure 1. In contrast, the VAE requires no such tuning at all, with the default $C_2 = 1$ producing the results shown.

11. Of course we admittedly have not exhaustively ruled out the potential existence of some alternative regularizer capable of outperforming the VAE when carefully tuned to appropriate conditions; however, it is still nonetheless impressive that the VAE can naturally perform so well without such tuning on a task that it was not even originally motivated for.

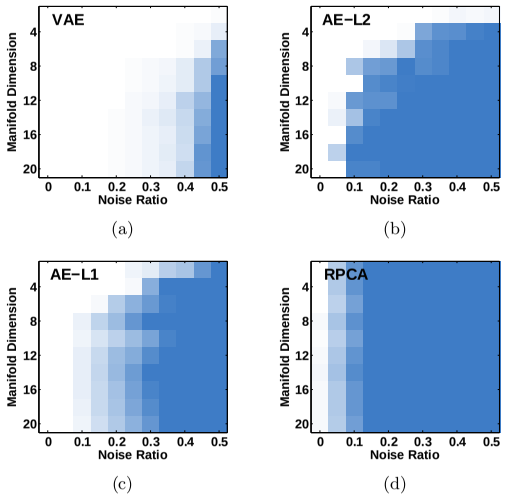


Figure 1: Results recovering synthesized low-dimensional manifolds across different outlier ratios (x-axis) and manifold dimensions (y-axis) for (a) the VAE, (b) the AE-ℓ_2, (c) the AE-ℓ_1, and (d) RPCA. In all cases, white color indicates normalized MSE near 0.0, while dark blue represents 1.0 or failure. The VAE is dramatically superior to each alternative, supporting Hypothesis (i). Additionally, it is crucial to note here that the AE and RPCA solutions perform poorly for quite different reasons. Not surprisingly, convex RPCA fails because it cannot accurately capture the underlying nonlinear manifold using a linear subspace inlier model. In contrast, both AE-ℓ_1 and AE-ℓ_2 have the *exact same* inlier model capacity as the VAE and can in principle represent uncorrupted points perfectly; however, they have inferior agency for pruning superfluous latent dimensions, discarding outliers, or generally avoiding bad locally-optimal solutions.

subspace is an inadequate signal representation. And we stress that, prior to the analysis herein, it was not at all apparent that a VAE could so dramatically outperform comparable AE models on this type of deterministic outlier removal task.

Note also that perfect reconstruction, as consistently exhibited by the VAE in Figure 1(a), does *not* actually require learning the correct generative model *within* the estimated manifold. Rather it only requires that as $\boldsymbol{\Sigma}_z \rightarrow \mathbf{0}$ selectively along appropriate dimensions (consistent with Hypothesis (iii) as will be discussed in Section 5.3), the encoder and decoder mean networks project onto the correct manifold while ignoring outliers. Hence although random samples $\boldsymbol{z}$ will likely lie on the true manifold when passed through the decoder network, they need not be perfectly distributed according to the full generative process

unless sufficient additional capacity exists beyond that needed to represent the manifold itself. We are not aware of this distinction being discussed in previous works, where VAE and related models are typically evaluated by either the overall quality of their generated samples (Dosovitskiy and Brox, 2016; Larsen et al., 2015; Oord et al., 2016), or by the value of the likelihood bound (Kingma and Welling, 2014; Kingma et al., 2016; Burda et al., 2015).[12]

Before proceeding to the next set of experiments, we address a tangential issue related to the RPCA performance as exhibited in Figure 1(d). When the outlier ratio is zero, RPCA can recover the ground-truth by simply defaulting to a full-rank inlier model without actually learning anything about the true manifold itself (the VAE and AE models do not have this luxury since they are forced to represent $\boldsymbol{L}$ using at most $\dim[\boldsymbol{\mu}_z] = 50 < \text{rank}[\boldsymbol{L}] = 100$ dimensions by design). In contrast, as the outlier ratio increases, it becomes increasingly difficult for RPCA to find any linear subspace representation that is both sufficiently high dimensional to include the majority of the inlier variance along the manifold while simultaneously excluding the outlier contributions. This explains the steep drop-off in performance moving from left to right within Figure 1(d). But there is noticeably no change in RPCA performance as we move from top to bottom in the same plot. This is because the clean data $\boldsymbol{L}$ is full-rank regardless of the manifold dimension κ, and so any linear subspace approximation is more or less equally bad across all κ.

5.2 Hypothesis (ii) Evaluation Using Ground-Truth Manifolds and MNIST Data

Synthetic Data Example: To evaluate Hypothesis (ii), we train analogous AE and VAE models as the number of decoder and encoder hidden layers vary, in each case with ground-truth available per the procedure described above. To generate each observed data point $\boldsymbol{x}^{(i)}$, we sample $\boldsymbol{z}^{(i)}$ from a 20-dimensional standard Gaussian distribution and pass it through a neural network structured as $\boldsymbol{z}(20)$-$\boldsymbol{D}_1(200)$-$\boldsymbol{D}_2(200)$-$\boldsymbol{x}(400)$, again with ReLU activations. We then train VAE models of variable depth, with concatenated mean networks $\boldsymbol{\mu}_x(\boldsymbol{\mu}_z[\boldsymbol{x}])$ designed as $\boldsymbol{x}(400)$-$\boldsymbol{E}_1(200)$-...-$\boldsymbol{E}_{N_e}(200)$-$\boldsymbol{\mu}_z(30)$-$\boldsymbol{D}_1(200)$-...-$\boldsymbol{D}_{N_d}(200)$-$\boldsymbol{\mu}_x(400)$, where N_e and N_d represent the number of hidden layers in the encoder and decoder respectively. The corresponding covariances are modeled as in Section 5.1, and likewise, the training protocol is unchanged. Note also that $\dim[\boldsymbol{\mu}_z] = 30$ is considerably larger than the ground-truth dimension of 20.

The first layer of the decoder mean network (before the nonlinearity) can be expressed as

$$\boldsymbol{h}_1 = \boldsymbol{W}_1\boldsymbol{z} + \boldsymbol{b}_1, \tag{21}$$

which in isolation is equivalent to the affine decoder mean model. If the VAE has the ability to find the true underlying manifold dimension, then the number of nonzero columns in $\boldsymbol{W}_1$ should be 20, indicating that $30 - 20 = 10$ dimensions of $\boldsymbol{z}$ are actually useless for

12. Learning the correct distribution within the manifold, as required for full recovery of the entire generative process and the production of realistic samples, is a topic largely orthogonal to the analysis presented herein. Still, to at least partially address these important issues, we have recently derived relatively broad conditions whereby provable recovery within a manifold itself is possible even in situations where $\boldsymbol{\Sigma}_z$ tends towards zero. However, we defer presentation of this topic, which involves many additional subtleties, to a future publication.

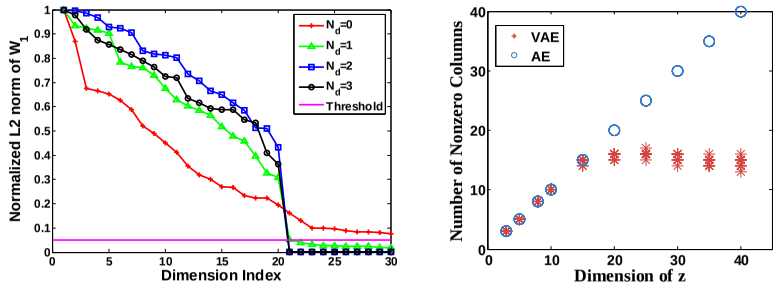
	$N_d = 0$	$N_d = 1$	$N_d = 2$	$N_d = 3$
$N_e = 0$	30.0	21.1	21.0	20.0
$N_e = 1$	30.0	21.0	20.0	20.0
$N_e = 2$	30.0	21.0	20.0	20.0
$N_e = 3$	30.0	20.4	20.0	20.0

Table 1: Number of nonzero columns in the VAE decoder mean first-layer weights $\boldsymbol{W}_1$ learned using different encoder and decoder depths applied to data with a ground-truth latent dimension of 20. Provided that the VAE model is sufficiently complex, the correct estimate is automatically obtained. We have not found an analogous AE model with similar capability.

any subsequent representation, i.e., we can estimate the intrinsic dimension of the latent code by counting the number of nonzero columns in $\boldsymbol{W}_1$, exactly analogous to the affine case. Of course in practice it is unlikely that a column of $\boldsymbol{W}_1$ converges all the way to exactly $\mathbf{0}$ via any stochastic optimization method. Therefore we define a simple threshold as $\text{thr} = 0.05 \times \max_{j=1}^{\kappa} \|\boldsymbol{w}_{\cdot j}\|_2$. If $\|\boldsymbol{w}_{\cdot j}\|_2 < \text{thr}$, we regard it as a zero column. But this heuristic notwithstanding, the partition between zero and non-zero columns is generally quite obvious as will be illustrated later.

Table 1 reports the estimated number of non-zero columns in $\boldsymbol{W}_1$ as N_e and N_d are varied, where we have run 10 trials for every pairing and averaged the results. When there is no hidden layer in the decoder (i.e., $N_d = 0$), which implies that the decoder mean is affine, all the columns are nonzero since the network is overly-simplistic and all degrees of freedom are being utilized to compensate. However, once we increase the depth, especially of the decoder within which $\boldsymbol{W}_1$ actually resides, the number of nonzero columns of $\boldsymbol{W}_1$ tends to exactly 20, which is the correct ground-truth manifold dimension by design, directly supporting Hypothesis (ii). Similar conclusions can be drawn from models of different sizes and configurations as well (not shown). In contrast, we did not find a corresponding AE model with this capability.

Note that prior work has loosely suggested that the KL regularizer indigenous to VAEs could potentially mute the impact of superfluous latent dimensions as part of the model optimization process (Burda et al., 2015; Sønderby et al., 2016). However, there has been no theoretical or empirical demonstration of why this should happen, nor any rigorous explanation of a precise pruning mechanism built into the aggregate VAE cost function itself. And as mentioned previously, the KL term is characterized by an ℓ_2 norm penalty on $\boldsymbol{\mu}_z$ (see (3)), which we would normally expect to promote low-energy latent representations with mostly small, but nonzero values (Chen et al., 1999), the exact *opposite* of any sparsity-promotion or pruning agency. But of course if columns of $\boldsymbol{W}_1$ are set to zero, then no information about $\boldsymbol{z}$ can pass through these dimensions to the hidden layers of the decoder. Therefore the KL term can now be minimized in isolation along these dimensions with the corresponding elements of $\boldsymbol{\mu}_z$ set to exactly zero. Hence it is only the counterintuitive co-mingling of *all* energy terms that leads to this desirable VAE pruning effect as we have meticulously characterized.



(a) (b)

Figure 2: (a) Validation of thresholding heuristic for determining nonzero columns in $\boldsymbol{W}_1$. With $N_e = 3$ and the settings from Table 1, the sorted column norms of $\boldsymbol{W}_1$ are plotted. Clearly for $N_d \in \{2, 3\}$ the gap between zero and nonzero values is extremely clear and any reasonable thresholding heuristic will suffice. (b) Number of nonzero columns in the decoder mean first-layer weights $\boldsymbol{W}_1$ as the latent dimension κ is varied for both AE and VAE models trained on MNIST data. Only the VAE automatically self-regularizes when κ becomes sufficiently large (here at $\kappa \approx 15$), consistent with Hypothesis (ii).

Finally, Figure 2(a) provides validation for our heuristic criterion for classifying columns of $\boldsymbol{W}_1$ as zero or not. Under the same experimental conditions as were used for creating Table 1, we plot the sorted column norms of $\boldsymbol{W}_1$ for the cases where $N_e = 3$ and $N_d \in \{0, 1, 2, 3\}$. Especially when $N_d \in \{2, 3\}$, meaning the model is of (or nearly of) sufficient capacity, zero and nonzero values are easily distinguishable and any reasonable thresholding heuristic would be adequate. Likewise for $N_d = 0$ it is clear that all values are significantly distant from zero. In contrast, when $N_d = 1$ (green curve) it is admittedly more subjective whether or not the smallest 9 or 10 elements should be classified as zero. Regardless, the overall trend is unequivocal, with any heuristic threshold only influencing the $N_d = 1$ boundary case.

MNIST Example: To further verify Hypothesis (ii), we train VAE models on the MNIST data set of handwritten digit images (Lecun et al., 1998) as κ is varied. We use all $n = 70000$ samples, each of size 28×28. We structure 4-layer cascaded VAE mean networks $\boldsymbol{\mu}_x(\boldsymbol{\mu}_z[\boldsymbol{x}])$ as $\boldsymbol{x}(d)$-$\boldsymbol{E}_1(1000)$-$\boldsymbol{E}_2(500)$-$\boldsymbol{E}_3(250)$-$\boldsymbol{\mu}_z(\kappa)$-$\boldsymbol{D}_1(250)$-$\boldsymbol{D}_2(500)$-$\boldsymbol{D}_3(1000)$-$\boldsymbol{\mu}_x(d)$, where $d = 28 \times 28 = 784$ and ReLU activations are used. Covariances and training protocols are handled as before. We draw values of κ from $\{3, 5, 8, 10, 15, 20, 25, 30, 35, 40\}$.

Figure 2(b) displays the number of nonzero columns in $\boldsymbol{W}_1$ produced by each κ-dependent model, again across 10 trials. We observe that when $\kappa > 15$, the number of nonzero columns plateaus for the VAE consistent with Hypothesis (ii). Of course unlike the

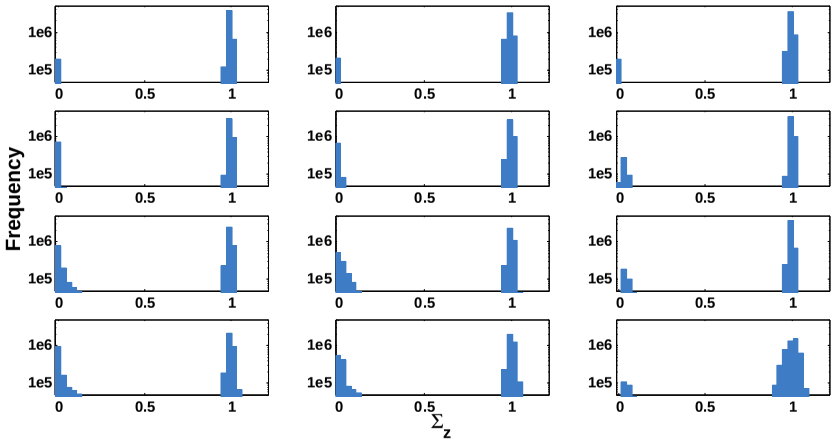


Figure 3: Log-scale histograms of $\left\{\boldsymbol{\Sigma}_z^{(i)}\right\}_{i=1}^n$ diagonal elements as outlier ratios and manifold dimensions are varying for the corrupted manifold recovery experiment corresponding with Figure 1. The three columns represent outlier ratios of $\nu \in \{0.0, 0.25, 0.50\}$ from left to right. The four rows represent manifold dimensions of $\kappa \in \{2, 8, 14, 20\}$ from top to bottom. All plots demonstrate the predicted clustering of variance values around either zero or one. Likewise, the relative sizes of these clusters, including observed changes across experimental conditions, conforms with our theoretical predictions (see detailed description in Section 5.3).

synthetic case, we no longer have access to ground truth for determining what the optimal manifold dimension should be.

We also applied an analogous AE model trained with $C_2 = 0$, i.e., a standard AE with no additional regularization penalty added. Not surprisingly, the number of nonzero columns in $\boldsymbol{W}_1$ is always equal to κ since there is no equivalent agency for column-wise pruning as implicitly instilled by the VAE. Note that tuning C_2 with either ℓ_1- or ℓ_2-norm penalties is of course always possible; however, the optimal value can be κ-dependent making subsequent results less interpretable. Moreover, in general we have not found a setting whereby the penalties lead to correct latent dimensionality estimation in situations where the ground-truth is known.

5.3 Hypothesis (iii) Evaluation Using Covariance Statistics from Corrupted Manifold Recovery Task

If some columns of $\boldsymbol{W}_1$ tend to zero as we have argued both empirically and theoretically, then the corresponding diagonal elements of $\boldsymbol{\Sigma}_z$, like $\boldsymbol{\mu}_z$, can no longer influence the decoder. And with only the lingering KL term to offer guidance, along these coordinates the optimal variance will then equal one by virtue of (19). But for nonzero columns of $\boldsymbol{W}_1$, the behavior of $\boldsymbol{\Sigma}_z$ is much more counter-intuitive. Despite the $-\log|\boldsymbol{\Sigma}_z|$ factor from the KL divergence that contributes an unbounded cost as any $[\boldsymbol{\Sigma}_z]_{jj} \to 0$, we nonetheless have proven for the affine decoder mean case a natural tendency of the VAE to push these variance values arbitrarily close to zero when approaching globally optimal solutions, at least along latent dimensions required for representing inlier points lying on ground-truth manifolds (i.e, dimensions where $\boldsymbol{W}_1$ is nonzero).

We now empirically verify that this same effect is inherited by general VAE models with more sophisticated, nonlinear decoder mean networks. For this purpose, we created histograms of all diagonal elements of $\left\{\boldsymbol{\Sigma}_z^{(i)}\right\}_{i=1}^n$ obtained from the experiments described in Section 5.1 where the outlier ratios and manifold dimensions vary. The results are plotted in Figure 3 for all pairs of outlier ratios $\nu \in \{0.0, 0.25, 0.50\}$ (columns) and ground-truth manifold dimensions $\kappa \in \{2, 8, 14, 20\}$ (rows). These results directly conform with our theoretical predictions per the following explanations.

First, consider the upper-left panel displaying the simplest case from an estimation standpoint, since $\nu = 0.0$ (no outliers) and $\kappa = 2$ (very low-dimensional manifold). Here we observe a clear partitioning between elements of $\left\{\boldsymbol{\Sigma}_z^{(i)}\right\}_{i=1}^n$ going to either zero or one. Moreover, given that $\dim[\boldsymbol{\mu}_z] = 50$ while the ground-truth involves $\kappa = 2$, 48 out of 50 dimensions are actually unnecessary. Hence we should expect that only about 4% of variance values should concentrate around zero, with the remainder forced towards one. In fact, this is precisely the general partitioning we observe (note the log scaling of the y-axis). Additionally, if we examine the other panels in the left-most column of Figure 3, we notice that as the ground-truth κ increases, the percentage of variance values shifts from one to zero roughly proportional to $\kappa/50$. In other words, as more dimensions are required to represent the more challenging, higher-dimensional manifolds, more diagonal elements of each $\boldsymbol{\Sigma}_z^{(i)}$ are pushed towards zero to enforce accurate reconstructions.

Next, we observe that in the top row of Figure 3, each of the three panels are more or less the same, indicating that the inclusion of outliers has not disrupted the VAE's ability to model the ground-truth manifold. In contrast, the bottom row presents a somewhat different story. Given the more challenging conditions with a much higher dimensional ground-truth manifold ($\kappa = 20$), the inclusion of additional outliers (as we move from left to right) shifts more variance elements from zero to one. This implies that the VAE, when confronted with both a higher-dimensional manifold and severe outliers (bottom-right panel), is settling on a relatively lower-dimensional approximation. This behavior is reasonable in the sense that accurately estimating a complex manifold via any method becomes problematic when 50% of the data is corrupted, and a low-dimensional approximation is all that is feasible to avoid simply fitting all the outliers. In this situation some manifold dimensions of lesser

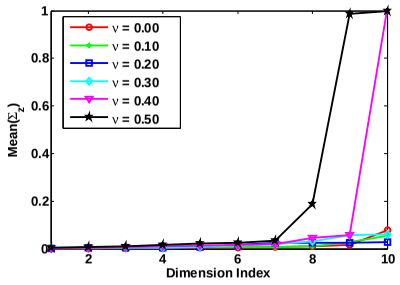


Figure 4: Diagonal values of $\frac{1}{n}\sum_{i=1}^{n}\mathbf{\Sigma}_z^{(i)}$ sorted in ascending order for a VAE model trained with ground-truth $\kappa = \dim[\boldsymbol{\mu}_z] = 10$ on the recovery task from Section 5.1. When the outlier proportion is $\nu \leq 0.30$, the average variance is near zero across all latent dimensions. However, for $\nu > 0.30$, some variance values are pushed towards one, indicating that the VAE is defaulting to a lower-dimensional approximation.

importance can be viewed as expendable, and consequently we will likely have additional elements of $\left\{\mathbf{\Sigma}_z^{(i)}\right\}_{i=1}^{n}$ tending to one.

To further elucidate this phenomena, we include one additional supporting visualization involving the special case where the ground-truth manifold dimension equals 10 while the outlier ratio ν varies. However, we slightly modify the testing conditions from Section 5.1. Instead of choosing $\dim[\boldsymbol{\mu}_z] = 50$, we set this value to the ground-truth value $\kappa = 10$. In this constrained setting, we expect that perfect recovery should require *all* diagonal elements of $\mathbf{\Sigma}_z$ to be pushed towards zero, since there are no longer any superfluous degrees of freedom. Therefore, if any covariance elements tend to one, we have isolated the emergence of a low-dimensional approximation as presumably necessitated by increasing outlier levels.

Figure 4 displays the diagonal values of $\frac{1}{n}\sum_{i=1}^{n}\mathbf{\Sigma}_z^{(i)}$ sorted in ascending order along the x-axis. When $\nu \leq 0.30$, the average variance is near zero across all latent dimensions. However, for $\nu > 0.30$, some variance values are pushed towards one, indicating that the VAE is defaulting to a lower-dimensional approximation.

To summarize then, the results of this section help to confirm a rather curious behavior of the VAE: If $\boldsymbol{\mu}_x(\boldsymbol{\mu}_z[\boldsymbol{x}])$ is suitably parameterized to model inlier samples, and $\mathbf{\Sigma}_x$ is sufficiently complex to model outlier locations, then elements of $\mathbf{\Sigma}_z$ can be selectively pushed towards zero in the neighborhood of global minima. This involves overpowering the $-\log|\mathbf{\Sigma}_z\|$ factor from the KL divergence that would otherwise seemingly prevent this from happening. Moreover, in this degenerate regime, the VAE will exhibit deterministic behavior and can perfectly represent original clean training data samples via a low-dimensional manifold provided that the outlier level is not too high.

6. Discussion

Although originally developed as a viable deep generative model or tractable bound on the data likelihood, in this work we have revealed certain properties and abilities of the VAE that are not obvious from first inspection. For example, in addition to its putative role in driving diversity into the learned generative process, the latent covariance $\mathbf{\Sigma}_z$ also serves as an important smoothing mechanism that aids in the robust recovery of corrupted samples, even if sometimes this requires exhibiting behavior (i.e., selective convergence towards zero) that may seem counterintuitive. And although the VAE only adopts an ℓ_2 norm penalty on $\boldsymbol{\mu}_z$ that in isolation should favor low energy solutions with all or mostly nonzero values, the latent mean estimator nonetheless tends to be highly sparse because of subtle, non-obvious interactions with other factors in the energy function such as the first-layer decoder mean network weights $\boldsymbol{W}_1$. Likewise, outliers can be estimated and completely removed via the action of $\mathbf{\Sigma}_x$ despite no traditional, additive sparsity penalty applied across each data point.

In general, our results speak to many under-appreciated aspects of VAE behavior, have wide ranging practical consequences, and suggest novel usages beyond the original VAE design principles. These include:

- The VAE can be applied to estimating deterministic nonlinear manifolds heavily corrupted with outliers.

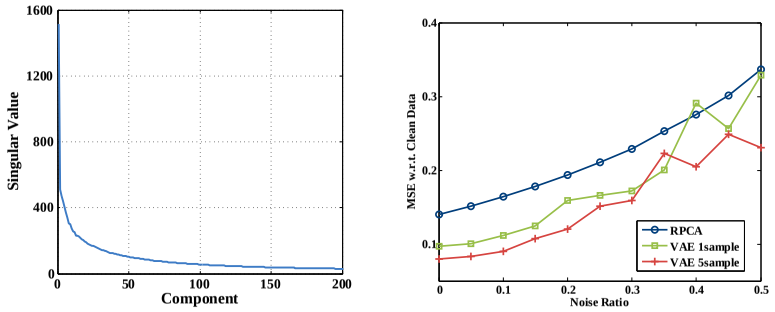
- The self-regularization effects of the VAE can largely handle excessive degrees of freedom when it comes to the latent representation $\boldsymbol{z}$ as produced by the full encoder and processed by the first layer of the decoder mean network, as well as an arbitrarily-parameterized decoder covariance $\mathbf{\Sigma}_x$. Conversely, only excessive complexity specifically localized in higher decoder mean network layers can, at least in principle, lead to potential problems with overfitting.

- The latent covariance $\mathbf{\Sigma}_z$ can serve as an approximate bellwether for determining the true dimensionality of a manifold, provided that excessive outliers/corruptions do not lead to an under-estimate. This is because typically near global solutions, we observe $[\mathbf{\Sigma}_z]_{jj} \to 0$ for *useful* dimensions, while for *useless* dimensions we have shown that $[\mathbf{\Sigma}_z]_{jj} \to 1$, a clear bifurcation.

Although the primary purpose of this paper is not to build a better generative model per se, we nevertheless hope that ideas introduced here will help to ensure that VAEs are not under or improperly utilized. Additionally, in closing we should also mention that the focus herein has been almost entirely on the analysis of the VAE energy function itself, independent of the specifics of how this energy function might ultimately be optimized in practice. But we believe the latter to be an equally-important, complementary topic, and further study is undeniably warranted. For example, if an optimization trajectory is somehow lured astray by the Siren's song of a bad local minimum in the VAE energy landscape, then obviously many of our conclusions predicated on global optima will not necessarily still hold.

Acknowledgments

This work was conducted while B. Dai was an intern and Y. Wang was a visiting researcher at Microsoft Research, Beijing. Y. Wang was also sponsored by the EPSRC Centre for Mathematical Imaging in Healthcare, University of Cambridge, EP/N014588/1.



(a) (b)

Figure 5: (a) Singular value spectrum of MNIST data revealing (approximately) low-rank structure. (b) Normalized MSE recovering MNIST digits from corrupted samples. The VAE is able to reduce the reconstruction error by better modeling more fine-grain details occupying the low end of the singular value spectrum.

Appendix A. Additional MNIST Data Set Experiment

Here we examine practical denoising of MNIST data corrupted with outliers using a VAE model. Outliers are added to MNIST handwritten digit data (Lecun et al., 1998) by randomly replacing from 5% to 50% of the pixels with a value uniformly sampled from $[0, 255]$ to create $\boldsymbol{X}$. We choose $\kappa = 30$ for the dimension of $\boldsymbol{z}$ and apply the same VAE structure as applied to MNIST data in Section 5.2. The model is trained using both $\tau = 1$ and $\tau = 5$ latent samples $\{\boldsymbol{z}^{(i,t)}\}_{t=1}^{\tau}$ for each $\boldsymbol{x}^{(i)}$, observing that the latter, which more closely approximates the posterior, should perform significantly better.

We compare the VAE against convex RPCA on the task of recovering the original, uncorrupted digits. Note that RPCA is commonly used for unsupervised cleaning of this type of data (Elhamifar and Vidal, 2013), and MNIST is known to have significant low-rank structure (Lu et al., 2013) as shown in Figure 5(a). Regardless, we observe in Figure 5(b) that the VAE performs significantly better in terms of normalized MSE by capturing additional manifold details that deviate from a purely low-rank representation. Furthermore, we hypothesize that using extra latent samples (the $\tau = 5$ case) may work better on outlier removal tasks given the strong need for accurate smoothing of the VAE objective as described previously.

Appendix B. Proof of Lemma 1

Proof Under the stated assumptions, the VAE cost can be simplified as

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) &= \sum_i \left\{ \mathbb{E}_{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}^{(i)})}\left[\tfrac{1}{\lambda}\left\|\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{z}-\boldsymbol{b}\right\|_2^2\right] + d\log\lambda \right. \\ &+ \left. \operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\right] - \log\left|\boldsymbol{\Sigma}_z^{(i)}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\}, \\ &= \sum_i \left\{ \tfrac{1}{\lambda}\left\|\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right\|_2^2 + \tfrac{1}{\lambda}\operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\boldsymbol{W}^\top\boldsymbol{W}\right] + d\log\lambda \right. \\ &+ \left. \operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\right] - \log\left|\boldsymbol{\Sigma}_z^{(i)}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\}, \end{aligned} \tag{22}$$

where $\boldsymbol{\mu}_z^{(i)} \triangleq \boldsymbol{\mu}_z\left(\boldsymbol{x}^{(i)};\boldsymbol{\phi}\right)$ and $\boldsymbol{\Sigma}_z^{(i)} \triangleq \boldsymbol{\Sigma}_z\left(\boldsymbol{x}^{(i)};\boldsymbol{\phi}\right)$. Given that

$$\log\left|\boldsymbol{A}\boldsymbol{A}^\top\right| = \arg\inf_{\boldsymbol{\Gamma}\succ\boldsymbol{0}} \operatorname{tr}\left[\boldsymbol{A}\boldsymbol{A}^\top\boldsymbol{\Gamma}^{-1}\right] + \log|\boldsymbol{\Gamma}|, \tag{23}$$

when optimization is carried out over positive definite matrices $\boldsymbol{\Gamma}$, minimization of (22) with respect to $\boldsymbol{\Sigma}_z^{(i)}$ leads to the revised objective

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) &\equiv \sum_i \left\{ \tfrac{1}{\lambda}\left\|\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right\|_2^2 + \log\left|\tfrac{1}{\lambda}\boldsymbol{W}^\top\boldsymbol{W}+\boldsymbol{I}\right| + d\log\lambda + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\}, \\ &= \sum_i \left\{ \tfrac{1}{\lambda}\left\|\boldsymbol{x}^{(i)}-\boldsymbol{W}\boldsymbol{\mu}_z^{(i)}-\boldsymbol{b}\right\|_2^2 + \log\left|\boldsymbol{W}\boldsymbol{W}^\top+\lambda\boldsymbol{I}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\}, \end{aligned} \tag{24}$$

ignoring constant terms. This expression only requires that $\boldsymbol{\Sigma}_z^{(i)} = \left[\frac{1}{\lambda}\boldsymbol{W}^\top\boldsymbol{W}+\boldsymbol{I}\right]^{-1}$, or a constant parameterization, independent of $\boldsymbol{x}^{(i)}$. Similarly we can optimize over $\boldsymbol{\mu}_z^{(i)}$ in terms of the other variables. This is just a ridge regression problem, with optimal solution

$$\boldsymbol{\mu}_z^{(i)} = \boldsymbol{W}^\top\left(\lambda\boldsymbol{I}+\boldsymbol{W}\boldsymbol{W}^\top\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{b}\right), \tag{25}$$

or a simple linear function of $\boldsymbol{x}^{(i)}$. Hence as long as the parameterization of both $\boldsymbol{\mu}_z^{(i)}$ and $\boldsymbol{\Sigma}_z^{(i)}$ allows for arbitrary affine functions as stipulated in the lemma statement, these optimal solutions are feasible. Plugging (25) into (24) and applying some basic linear algebra, we arrive at

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) \equiv \sum_i \left(\boldsymbol{x}^{(i)}-\boldsymbol{b}\right)^\top\left(\boldsymbol{W}\boldsymbol{W}^\top+\lambda\boldsymbol{I}\right)^{-1}\left(\boldsymbol{x}^{(i)}-\boldsymbol{b}\right) + n\log\left|\boldsymbol{W}\boldsymbol{W}^\top+\lambda\boldsymbol{I}\right|. \tag{26}$$

Finally, in the event that we enforce that $\boldsymbol{\Sigma}_z^{(i)}$ be diagonal, (24) must be modified via

$$\boldsymbol{\Sigma}_z^{(i)} = \left[\tfrac{1}{\lambda}\operatorname{diag}\left(\operatorname{diag}\left[\boldsymbol{W}^\top\boldsymbol{W}\right]\right)+\boldsymbol{I}\right]^{-1} = \sum_{j=1}^{\kappa}\log\left(\lambda+\|\boldsymbol{w}_{\cdot j}\|_2^2\right) - \kappa\log\lambda, \tag{27}$$

where the diag[·] operator converts vectors to diagonal matrices, and a matrix to a vector formed from its diagonal (just as in the Matlab computing environment), leading to the stated result. ■

Appendix C. Proof of Theorem 2

Proof First, for part 1 on the theorem, given that $\boldsymbol{W}\boldsymbol{R}\boldsymbol{R}^\top\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{P}\boldsymbol{P}^\top\boldsymbol{W}^\top = \boldsymbol{W}\boldsymbol{W}^\top$ for any rotation $\boldsymbol{R}$ and permutation $\boldsymbol{P}$, then obviously if $\boldsymbol{W}^*$ is a minimum of (5), $\boldsymbol{W}^*\boldsymbol{R}$ and $\boldsymbol{W}^*\boldsymbol{P}$ must also be. Likewise, since $\sum_{j=1}^{\kappa}\log\left(\lambda+\|\boldsymbol{w}_{\cdot j}\|_2^2\right)$ is invariant to the order of the summation, then if $\boldsymbol{W}^{**}$ is a minimum of (7), $\boldsymbol{W}^{**}\boldsymbol{P}$ must be as well.

We also have that

$$\begin{aligned}\mathcal{L}_{sep}(\boldsymbol{W}^{**},\boldsymbol{b},\lambda) &= \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{**},\boldsymbol{b},\lambda\boldsymbol{I}) + n\left[\sum_j \log\left(\lambda+\|\boldsymbol{w}_{\cdot j}^{**}\|_2^2\right) + (d-\kappa)\log\lambda\right] \\ &= \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{**},\boldsymbol{b},\lambda\boldsymbol{I}) + n\left[\sum_j \log\left(1+\tfrac{1}{\lambda}\|\boldsymbol{w}_{\cdot j}^{**}\|_2^2\right) + d\log\lambda\right] \\ &\geq \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{**},\boldsymbol{b},\lambda\boldsymbol{I}) + n\left[\log\left|\tfrac{1}{\lambda}\left(\boldsymbol{W}^{**}\right)^\top\boldsymbol{W}^{**}+\boldsymbol{I}\right| + d\log\lambda\right] \\ &= \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{**},\boldsymbol{b},\lambda\boldsymbol{I}) + n\log\left|\lambda\boldsymbol{I}+\boldsymbol{W}^{**}\boldsymbol{R}\left(\boldsymbol{W}^{**}\boldsymbol{R}\right)^\top\right| \\ &\geq \sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{*},\boldsymbol{b},\lambda\boldsymbol{I}) + n\log\left|\lambda\boldsymbol{I}+\boldsymbol{W}^{*}\boldsymbol{R}\left(\boldsymbol{W}^{*}\boldsymbol{R}\right)^\top\right|, \end{aligned} \tag{28}$$

where the the second inequality follows from the fact that $\boldsymbol{W}^*$ is an optimal solution to (5). The first inequality stems from Hadamard's inequality (Garling, 2007) applied to

$$\tfrac{1}{\lambda}\left(\boldsymbol{W}^{**}\right)^\top\boldsymbol{W}^{**}+\boldsymbol{I} = \boldsymbol{M}^\top\boldsymbol{M} \tag{29}$$

for some square matrix $\boldsymbol{M}$ of appropriate dimension. This results in

$$\log\left|\tfrac{1}{\lambda}\left(\boldsymbol{W}^{**}\right)^\top\boldsymbol{W}^{**}+\boldsymbol{I}\right| = 2\log|\boldsymbol{M}| \leq 2\log\left(\prod_j\|\boldsymbol{m}_{\cdot j}\|_2\right) = \sum_j\log\left(1+\tfrac{1}{\lambda}\|\boldsymbol{w}_{\cdot j}^{**}\|_2^2\right), \tag{30}$$

with equality iff $\boldsymbol{M}^\top\boldsymbol{M}$ is diagonal. We can further manipulate the log-det term in (28) via

$$\begin{aligned} n\log\left|\lambda\boldsymbol{I}+\boldsymbol{W}^*\boldsymbol{R}\left(\boldsymbol{W}^*\boldsymbol{R}\right)^\top\right| &= \log\left|\tfrac{1}{\lambda}\left(\boldsymbol{W}^*\boldsymbol{R}\right)^\top\boldsymbol{W}^*\boldsymbol{R}+\boldsymbol{I}\right| + d\log\lambda \\ &= \log\left|\tfrac{1}{\lambda}\left(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top\boldsymbol{R}\right)^\top\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top\boldsymbol{R}+\boldsymbol{I}\right| + d\log\lambda \\ &= \log\left|\tfrac{1}{\lambda}\boldsymbol{R}^\top\boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^\top\boldsymbol{R}+\boldsymbol{I}\right| + d\log\lambda, \end{aligned}$$

where $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ is the SVD of $\boldsymbol{W}^*$. Now if we choose $\boldsymbol{R}=\boldsymbol{V}$ and define $\bar{\boldsymbol{W}} \triangleq \boldsymbol{W}\boldsymbol{V}$, then $\Lambda_{jj} = \|\bar{\boldsymbol{w}}_{\cdot j}\|_2^2$ and this expression further reduces via

$$\begin{aligned}\log\left|\tfrac{1}{\lambda}\boldsymbol{R}^\top\boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^\top\boldsymbol{R}+\boldsymbol{I}\right| + d\log\lambda &= \sum_j\log\left(1+\tfrac{1}{\lambda}\Lambda_{jj}\right) + d\log\lambda \\ &= \sum_j\log\left(\lambda+\|\bar{\boldsymbol{w}}_{\cdot j}\|_2^2\right) + (d-\kappa)\log\lambda. \end{aligned} \tag{31}$$

Of course we cannot have

$$\sum_i \boldsymbol{\Omega}^{(i)}(\boldsymbol{W}^{**}, \boldsymbol{b}, \lambda \boldsymbol{I}) + n\left[\sum_j \log\left(\lambda + \|\boldsymbol{w}_{\cdot j}^{**}\|_2^2\right) + (d-\kappa)\log\lambda\right] \tag{32}$$
$$> \sum_i \boldsymbol{\Omega}^{(i)}(\bar{\boldsymbol{W}}, \boldsymbol{b}, \lambda \boldsymbol{I}) + n\left[\sum_j \log\left(\lambda + \|\bar{\boldsymbol{w}}_{\cdot j}\|_2^2\right) + (d-\kappa)\log\lambda\right],$$

otherwise $\boldsymbol{W}^{**}$ would not be a minimum of (7). Therefore, $\bar{\boldsymbol{W}}$ must also be a minimum of (7), from which the remaining parts of (9) immediately follows.

We next confront the arrangement of disconnected minima for part 2 of the theorem. It is not difficult to show that (5), and by virtue of the analysis above (7), will be uniquely minimized by $\boldsymbol{U}$ and $\boldsymbol{\Lambda}$ arising from the SVD of either $\boldsymbol{W}^*$ or equivalently $\boldsymbol{W}^{**}$. Let $\boldsymbol{W}^{**} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^\top$ via such a decomposition. So any partitioning into disconnected minimizers must come at the hands of $\boldsymbol{V}$, which only influences the $\sum_j \log\left(\lambda + \|\boldsymbol{w}_{\cdot j}\|_2^2\right)$ term in (7).

At stated above, if $\boldsymbol{W}^{**}$ is a minimum, then $\boldsymbol{W}^{**}\boldsymbol{P}$ must also be a minimum. Assume for the moment that $\boldsymbol{W}^{**}$ is full column rank. There will obviously be $r!$ unique permutations of its columns, with $r = \text{rank}[\boldsymbol{W}^{**}]$. Moreover, any transition from some permutation $\boldsymbol{P}'$ to another $\boldsymbol{P}''$ will necessarily involve some non-permutation-matrix rotation $\boldsymbol{V}$. Given our assumption of distinct eigenvalues, this will ensure that

$$\tfrac{1}{\lambda}\left(\boldsymbol{W}^{**}\right)^\top \boldsymbol{W}^{**} + \boldsymbol{I} = \tfrac{1}{\lambda}\boldsymbol{V}\boldsymbol{\Lambda}^2\boldsymbol{V}^\top + \boldsymbol{I} \tag{33}$$

is non-diagonal. While this will not increase (5), it *must* increase (7) when diagonalized by Hadamard's inequality. Therefore every permutation will reflect a distinct, disconnected minimizer. If $\boldsymbol{W}^{**}$ also has $\kappa - r$ zero-valued columns, then the resulting number of unique permutations increases to $\frac{\kappa!}{\kappa - r}$ by standard rules of combinatorics.

Finally, part 3 of the theorem follows directly from part 2: Given that any minimizer of (7) must be of the form $\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{P}$, then there cannot be more than r nonzero columns. In contrast, for (5) we may apply any arbitrary rotation to $\boldsymbol{W}^*$, and hence all columns can be nonzero even if the rank is smaller than κ. ■

Appendix D. Proof of Theorem 3

Proof For convenience we will adopt the notation $f(\alpha) = \mathrm{O}(h(\alpha))$ to indicate that there exists a positive $\bar{\alpha}$ and some constant C independent of α such that $|f(\alpha)| < Ch(\alpha)$ for all $\alpha \in (0, \bar{\alpha}]$. Similarly, we use $f(\alpha) = \Omega(h(\alpha))$ to convey that $|f(\alpha)| > Ch(\alpha)$ under equivalent conditions. We then say $f(\alpha) = \Theta(h(\alpha))$ iff $f(\alpha) = \mathrm{O}(h(\alpha))$ and $f(x) = \Omega(h(\alpha))$. Additionally, if the input argument to one of these expressions is a vector, the result is understood to apply element-wise.

The basic high-level strategy here is as follows: We first present a candidate solution that satisfies (12) and carefully quantify the achievable objective function value for $\alpha \in (0, \bar{\alpha}]$, and $\bar{\alpha}$ small. We then analyze a lower bound on the VAE cost and demonstrate that no solution can do significantly better, namely, any solution that can match the performance

of our original proposal must necessarily also satisfy (12). Given that this is a lower bound, this implies that no other solution can both minimize the VAE objective and not satisfy (12). We now proceed to the details.

Define $\boldsymbol{\mu}_z^{(i)} \triangleq \boldsymbol{\mu}_z\left(\boldsymbol{x}^{(i)}; \boldsymbol{\phi}\right)$ and $\boldsymbol{\Sigma}_z^{(i)} \triangleq \boldsymbol{\Sigma}_z\left(\boldsymbol{x}^{(i)}; \boldsymbol{\phi}\right)$. We first note that if $\boldsymbol{z} = \boldsymbol{\mu}_z^{(i)} + \boldsymbol{S}_z^{(i)}\boldsymbol{\epsilon}$, with $\boldsymbol{S}_z^{(i)}$ satisfying $\boldsymbol{\Sigma}_z^{(i)} = \boldsymbol{S}_z^{(i)}\left(\boldsymbol{S}_z^{(i)}\right)^\top$, and $\boldsymbol{\epsilon} \sim p(\boldsymbol{\epsilon}) = \mathcal{N}(\boldsymbol{\epsilon}; \boldsymbol{0}, \boldsymbol{I})$, then $\boldsymbol{z} \sim q_{\boldsymbol{\phi}}\left(\boldsymbol{z}|\boldsymbol{x}^{(i)}\right)$. With this reparameterization and

$$\begin{aligned}
\boldsymbol{\mu}_x^{(i)} &\triangleq \boldsymbol{W}\boldsymbol{\mu}_z^{(i)} + \boldsymbol{W}\boldsymbol{S}_z^{(i)}\boldsymbol{\epsilon}, \\
\text{diag}[\boldsymbol{\Sigma}_x^{(i)}] &\triangleq \nu\left(\boldsymbol{\mu}_z^{(i)} + \boldsymbol{S}_z^{(i)}\boldsymbol{\epsilon}; \boldsymbol{\theta}\right) \text{ for some function } \nu \\
\boldsymbol{\mu}_z^{(i)} &\triangleq f(\boldsymbol{x}^{(i)}; \boldsymbol{\phi}) \text{ for some function } f \\
\boldsymbol{S}_z^{(i)} &\triangleq g(\boldsymbol{x}^{(i)}; \boldsymbol{\phi}) \text{ for some function } g,
\end{aligned} \tag{34}$$

the equivalent VAE objective becomes

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;=\; & \sum_i \left\{ \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)} - \boldsymbol{W}\boldsymbol{S}_z^{(i)}\boldsymbol{\epsilon}\right)^\top \left(\boldsymbol{\Sigma}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)} - \boldsymbol{W}\boldsymbol{S}_z^{(i)}\boldsymbol{\epsilon}\right)\right] \right. \\
& \left. + \;\; \mathbb{E}_{p(\boldsymbol{\epsilon})}\left[\log\left|\boldsymbol{\Sigma}_x^{(i)}\right|\right] + \text{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\right] - \log\left|\boldsymbol{\Sigma}_z^{(i)}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\},
\end{aligned} \tag{35}$$

when $\boldsymbol{b} = \boldsymbol{0}$ as stipulated.[13] For now assume that κ, the dimension of the latent $\boldsymbol{z}$, satisfies $\kappa = \text{rank}[\boldsymbol{U}]$ (later we will relax this assumption).

D.1 A Candidate Solution

Here we consider a candidate solution that, by design, satisfies (12). For the encoder parameters we choose

$$\hat{\boldsymbol{\mu}}_z^{(i)} = \boldsymbol{\pi}^{(i)}, \qquad \hat{\boldsymbol{\Sigma}}_z^{(i)} = \alpha \boldsymbol{I}. \tag{36}$$

where α is a non-negative scalar and $\boldsymbol{\pi}^{(i)}$ is defined in conjunction with a matrix $\boldsymbol{\Psi}$ such that

$$\begin{aligned}
\text{supp}_\alpha\left[\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\right] &= \text{supp}\left[\boldsymbol{s}^{(i)}\right] \\
\text{span}\left[\boldsymbol{U}\right] &= \text{span}\left[\boldsymbol{\Psi}\right].
\end{aligned} \tag{37}$$

All quantities in (36) can be readily computed via $\boldsymbol{X}$ applied to an encoder module provided that $\kappa = \dim[\boldsymbol{z}] = \text{rank}\left[\boldsymbol{U}\right]$ as stipulated, and sufficient representational complexity for $\boldsymbol{\mu}_z$ and $\boldsymbol{\Sigma}_z$. Additionally, for the encoder we only need to define the posterior moments at specific points $\boldsymbol{x}^{(i)}$, hence the indexing via i in (36).

In contrast, for the decoder we consider the solution defined over any $\boldsymbol{z}$ given by

$$\begin{aligned}
\hat{\boldsymbol{W}} &= \boldsymbol{\Psi} \\
\hat{\boldsymbol{\mu}}_x &= \hat{\boldsymbol{W}}\boldsymbol{z} \\
\text{diag}\left[\hat{\boldsymbol{\Sigma}}_x\right] &= \boldsymbol{\Lambda}^{(h_\pi(\boldsymbol{z}))},
\end{aligned} \tag{38}$$

13. The extension to arbitrary $\boldsymbol{b}$ is trivial but clutters the presentation.

where $\boldsymbol{\Lambda}^{(i)} \in \mathbb{R}^{d\times d}$ is a diagonal matrix with

$$\left[\boldsymbol{\Lambda}^{(i)}\right]_{jj} = \begin{cases} \alpha, & \text{if } s_j^{(i)} = 0, \\ 1, & \text{otherwise}, \end{cases} \quad \forall j. \tag{39}$$

and $h_\pi : \mathbb{R}^\kappa \rightarrow \{1,\ldots,n\}$ is a function satisfying

$$h_\pi(\boldsymbol{z}) \triangleq \arg\min_{i\in\{1,..,n\}} \|\boldsymbol{z} - \boldsymbol{\pi}^{(i)}\|_2. \tag{40}$$

Again, given sufficient capacity, this function can always be learned by the decoder such that (38) is computable for any $\boldsymbol{z}$. Given these definitions, then the index-specific moments $\hat{\boldsymbol{\mu}}_x^{(i)}$ and $\hat{\boldsymbol{\Sigma}}_x^{(i)}$ are of course reduced to functions of $\boldsymbol{\epsilon}$ given by

$$\begin{aligned} \hat{\boldsymbol{\mu}}_x^{(i)} &= \hat{\boldsymbol{\mu}}_x\left(\hat{\boldsymbol{\mu}}_z^{(i)} + \hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}; \boldsymbol{\theta}\right) \\ \hat{\boldsymbol{\Sigma}}_x^{(i)} &= \hat{\boldsymbol{\Sigma}}_x\left(\hat{\boldsymbol{\mu}}_z^{(i)} + \hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}; \boldsymbol{\theta}\right). \end{aligned} \tag{41}$$

.

We next analyze the behavior of (35) at this specially parameterized solution as $\bar{\alpha}$ becomes small, in which case by design all covariances will be feasible by design. For this purpose, we first consider the integration across all cases where $\hat{\boldsymbol{\Sigma}}_x^{(i)}$ does not reflect the correct support, meaning $\boldsymbol{\epsilon} \notin \mathcal{S}^{(i)}$, where

$$\mathcal{S}^{(i)} \triangleq \left\{\boldsymbol{\epsilon} : \quad \left[\boldsymbol{\Sigma}_x^{(i)}\right]_{jj} = \alpha \;\text{ iff }\; s_j^{(i)} = 0, \;\; \forall j\right\}. \tag{42}$$

With this segmentation in mind, the VAE objection naturally partitions as

$$\mathcal{L}(\boldsymbol{\theta},\boldsymbol{\phi}) = \sum_i \left\{\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)}) + \mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)})\right\}, \tag{43}$$

where $\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)})$ denotes the cost for the i-th sample when integrated across those samples not in $\mathcal{S}^{(i)}$, and $\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)})$ is the associated complement.

D.2 Evaluation of $\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)})$

First we define

$$\rho = \min_{i,j\in\{1,..,n\}, i\neq j} \tfrac{1}{2}\|\boldsymbol{\pi}^{(i)} - \boldsymbol{\pi}^{(j)}\|_2, \tag{44}$$

which is just half the minimum distance between any two distinct coefficient expansions. If any $\boldsymbol{z}$ is within this distance of $\boldsymbol{\pi}^{(i)}$, it will necessarily be quantized to this value per our previous definitions. Therefore if $\|\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\|_2 < \rho$, we are guaranteed that the correct generating support pattern will be mapped to $\hat{\boldsymbol{\Sigma}}_x^{(i)}$, and so it follows that

$$P\left(\boldsymbol{\epsilon} \notin \mathcal{S}^{(i)}\right)) \leq P\left(\left\|\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right\|_2 > \rho\right) = P(\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 > \rho) \tag{45}$$

at our candidate solution. We also make use of the quantity

$$\eta \triangleq \max_{i\in\{1,..,n\}} \|\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\|_2^2, \tag{46}$$

which represents the maximum data-fitting error. Then for the i-th sample we have

$$\begin{aligned} &\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)}) \\ &= \int_{\boldsymbol{\epsilon}\notin\mathcal{S}^{(i)}} \left[\left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right)^\top \left(\hat{\boldsymbol{\Sigma}}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right)\right. \\ &\quad + \left.\log\left|\hat{\boldsymbol{\Sigma}}_x^{(i)}\right| + \mathrm{tr}\left[\hat{\boldsymbol{\Sigma}}_z^{(i)}\right] - \log\left|\hat{\boldsymbol{\Sigma}}_z^{(i)}\right| + \|\hat{\boldsymbol{\mu}}_z^{(i)}\|_2^2\right] p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \\ &\leq \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} \left[\left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right)^\top \left(\hat{\boldsymbol{\Sigma}}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right)\right. \\ &\quad + \left.\log\left|\hat{\boldsymbol{\Sigma}}_x^{(i)}\right| + \mathrm{tr}\left[\hat{\boldsymbol{\Sigma}}_z^{(i)}\right] - \log\left|\hat{\boldsymbol{\Sigma}}_z^{(i)}\right| + \|\hat{\boldsymbol{\mu}}_z^{(i)}\|_2^2\right] p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \\ &\leq \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} \left[\tfrac{1}{\alpha}\left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)} - \sqrt{\alpha}\boldsymbol{\Psi}\boldsymbol{\epsilon}\right)^\top \left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)} - \sqrt{\alpha}\boldsymbol{\Psi}\boldsymbol{\epsilon}\right)\right. \\ &\quad + \left.\kappa\alpha - \kappa\log\alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon}, \end{aligned} \tag{47}$$

where the second inequality comes from setting $\hat{\boldsymbol{\Sigma}}_x^{(i)} = \alpha\boldsymbol{I}$ (its smallest possible value) in the inverse term and $\hat{\boldsymbol{\Sigma}}_x^{(i)} = \boldsymbol{I}$ (its largest value) in the log-det term. Next, given that

$$\begin{aligned} \|\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\|_2^2 &\leq \eta \\ \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} \left(\boldsymbol{\pi}^{(i)}\right)^\top \boldsymbol{\Psi}^\top\boldsymbol{\Psi}\boldsymbol{\epsilon}\cdot p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} &= 0 \\ \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} \|\boldsymbol{\Psi}\boldsymbol{\epsilon}\|_2^2 p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} &\leq \mathrm{tr}\left[\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\right], \end{aligned} \tag{48}$$

it follows that the bound from (47) can be further reduced via

$$\begin{aligned} \mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)}) &\leq \mathrm{tr}\left[\boldsymbol{\Psi}^\top\boldsymbol{\Psi}\right] + \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} \left[\tfrac{1}{\alpha}\eta + \kappa\alpha - \kappa\log\alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \\ &= \Theta(1) + \left[\tfrac{1}{\alpha}\eta + \kappa\alpha - \kappa\log\alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} \\ &\leq \Theta(1) + \left[\tfrac{1}{\alpha}\eta + \kappa\alpha - \kappa\log\alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] \tfrac{\alpha}{\rho^2} \\ &= \Theta(1) + \Theta(\alpha^2) - \Theta(\alpha\log\alpha) \\ &= \Theta(1) \quad \text{as } \alpha \rightarrow 0, \end{aligned} \tag{49}$$

where the second inequality holds based on the vector version of Chebyshev's inequality, which ensures that

$$\int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2>\rho} p(\boldsymbol{\epsilon})d\boldsymbol{\epsilon} = P(\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 > \rho) \leq \tfrac{\alpha}{\rho^2}. \tag{50}$$

Clearly then, as α becomes small, we have established that

$$\mathcal{L}^{(i)}(\boldsymbol{\theta},\boldsymbol{\phi}; \epsilon \notin \mathcal{S}^{(i)}) \rightarrow O\left(1\right). \tag{51}$$

D.3 Evaluation of $\mathcal{L}^{(i)}(\boldsymbol{\theta}, \boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)})$

In analyzing $\mathcal{L}^{(i)}(\boldsymbol{\theta}, \boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)})$, we note that

$$\begin{aligned}
&\int_{\boldsymbol{\epsilon} \in \mathcal{S}^{(i)}} \left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right)^\top \left(\hat{\boldsymbol{\Sigma}}_x^{(i)}\right)^{-1} \left(\boldsymbol{x}^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{\mu}}_z^{(i)} - \hat{\boldsymbol{W}}\hat{\boldsymbol{S}}_z^{(i)}\boldsymbol{\epsilon}\right) p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\
&\quad \leq \int \left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)} - \sqrt{\alpha}\boldsymbol{\Psi}\boldsymbol{\epsilon}\right)^\top \left(\boldsymbol{\Lambda}^{(i)}\right)^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)} - \sqrt{\alpha}\boldsymbol{\Psi}\boldsymbol{\epsilon}\right) p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\
&\quad \leq \int \left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\right)^\top \left(\boldsymbol{\Lambda}^{(i)}\right)^{-1} \left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\right) p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} + \operatorname{tr}\left[\boldsymbol{\Psi}^\top \boldsymbol{\Psi}\right] \\
&\quad \leq \eta + \operatorname{tr}\left[\boldsymbol{\Psi}^\top \boldsymbol{\Psi}\right] \\
&\quad = \Theta(1)
\end{aligned} \tag{52}$$

given the alignment of $\boldsymbol{\Lambda}^{(i)}$ with zero-valued elements in $\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}$. Furthermore, the remaining terms in $\mathcal{L}^{(i)}(\boldsymbol{\theta}, \boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)})$ are independent of $\boldsymbol{\epsilon}$ giving

$$\begin{aligned}
&\int_{\boldsymbol{\epsilon} \in \mathcal{S}^{(i)}} \left[\log \left|\hat{\boldsymbol{\Sigma}}_x^{(i)}\right| + \operatorname{tr}\left[\hat{\boldsymbol{\Sigma}}_z^{(i)}\right] - \log \left|\hat{\boldsymbol{\Sigma}}_z^{(i)}\right| + \|\hat{\boldsymbol{\mu}}_z^{(i)}\|_2^2\right] p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\
&\quad = \left[\log \left|\boldsymbol{\Lambda}^{(i)}\right| + \kappa\alpha - \kappa \log \alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 < \rho} p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\
&\quad = \left[(r^{(i)} - \kappa)\log \alpha + \kappa\alpha + \|\boldsymbol{\pi}^{(i)}\|_2^2\right] \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 < \rho} p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \\
&\quad = \left[(r^{(i)} - \kappa)\log \alpha\right] \int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 < \rho} p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} + O(\alpha) + O(1),
\end{aligned} \tag{53}$$

where

$$r^{(i)} \triangleq \left|\left\{j : \Lambda_{jj}^{(i)} = \alpha\right\}\right| = d - \|\boldsymbol{s}^{(i)}\|_0. \tag{54}$$

Therefore, since $\int_{\|\sqrt{\alpha}\boldsymbol{\epsilon}\|_2 < \rho} p(\boldsymbol{\epsilon}) d\boldsymbol{\epsilon} \to 1$ as α becomes small, we may conclude that

$$\mathcal{L}^{(i)}(\boldsymbol{\theta}, \boldsymbol{\phi}; \epsilon \in \mathcal{S}^{(i)}) \to \left(d - \kappa - \|\boldsymbol{s}^{(i)}\|_0\right) \log \alpha + O(1). \tag{55}$$

D.4 Compilation of Candidate Solution Cost

After combining (51) and (55) across all i we find that

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \to \sum_i \left(d - \kappa - \|\boldsymbol{s}^{(i)}\|_0\right) \log \alpha + O(1) \tag{56}$$

for any $\alpha \in (0, \bar{\alpha}]$ as $\bar{\alpha}$ becomes small. If $d > \kappa + \|\boldsymbol{s}^{(i)}\|_0$, then this expression will tend towards minus infinity, indicative of an objective value that is unbounded from below, certainly a fertile region for candidate minimizers. Note that per the theorem statement, $\boldsymbol{L} = \boldsymbol{U}\boldsymbol{V}$ and $\boldsymbol{S}$ must represent a unique feasible solution to

$$\min_{\boldsymbol{L}, \boldsymbol{S}} \; d \cdot \operatorname{rank}[\boldsymbol{L}] + \|\boldsymbol{S}\|_0 \quad \text{s.t.} \quad \boldsymbol{X} = \boldsymbol{L} + \boldsymbol{S}. \tag{57}$$

Given that each column $\boldsymbol{x}^{(i)}$ has d degrees of freedom, then with $\boldsymbol{U}$ fixed there will be an infinite number of feasible solutions $\boldsymbol{x}^{(i)} = \boldsymbol{U}\boldsymbol{v}^{(i)} + \boldsymbol{s}^{(i)}$ such that $\dim[\boldsymbol{v}^{(i)}] + \|\boldsymbol{s}^{(i)}\|_0 = \kappa + \|\boldsymbol{s}^{(i)}\|_0 > d$ and a combinatorial number such that $k + \|\boldsymbol{s}^{(i)}\|_0 = d$. Therefore for uniqueness we require that $k + \|\boldsymbol{s}^{(i)}\|_0 < d$, so it follows that indeed $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ will be unbounded from below as $\bar{\alpha}$ and therefore α becomes small, with cost given by (56) as a candidate solution satisfying the conditions of the theorem.

Of course it still remains possible that some other candidate solution could exist that violates one of these conditions and yet still achieves (56) or an even lower cost. We tackle this issue next. For this purpose our basic strategy will be to examine a *lower* bound on $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ and show that essentially any candidate solution violating the theorem conditions will be worse than (56).

D.5 Evaluation of Other Candidate Solutions

To begin, we first observe that if granted the flexibility to optimize $\boldsymbol{\Sigma}_x^{(i)}$ independently over all values of $\boldsymbol{\epsilon}$ inside the integral for computing $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$, we immediately obtain a rigorous lower bound.[14] For this purpose we must effectively solve decoupled problems of the form

$$\inf_{\gamma > \alpha} \tfrac{c}{\gamma} + \log \gamma, \tag{58}$$

to which the optimal solution is just

$$\gamma^* = \xi_\alpha(x) \triangleq [c - \alpha]_+ + \alpha, \tag{59}$$

where the operator $[\cdot]_+$ retains only the positive part of its argument, setting negative values to zero. Plugging this solution back into (58), we find that

$$\inf_{\gamma > \alpha} \tfrac{c}{\gamma} + \log \gamma \quad = \quad \log \xi_\alpha(c) + O(1). \tag{60}$$

In the context of our bound, this leads to

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \quad \geq \quad & \sum_i \left\{ \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_j \log \xi_\alpha \left(\left[x_j^{(i)} - \boldsymbol{w}_{j\cdot} \boldsymbol{\mu}_z^{(i)} - \boldsymbol{w}_{j\cdot} \boldsymbol{S}_z^{(i)} \boldsymbol{\epsilon} \right]^2 \right) \right] \right. \\
& \left. + \; \operatorname{tr}\left[\boldsymbol{\Sigma}_z^{(i)}\right] - \log \left|\boldsymbol{\Sigma}_z^{(i)}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2 \right\} + O(1).
\end{aligned} \tag{61}$$

From this expression, it is clear that the lowest objective value we could ever hope to obtain cannot involve arbitrarily large values of $\boldsymbol{\Sigma}_z^{(i)}$ and $\boldsymbol{\mu}_z^{(i)}$ since the respective trace and quadratic terms grow faster than log-det terms. Likewise $\boldsymbol{\mu}_z^{(i)}$ cannot be unbounded for analogous reasons. Therefore, optimal solutions to (61) that will be unbounded from below must involve the first term becoming small, at least over a range of $\boldsymbol{\epsilon}$ values with significant probability measure. Although the required integral admits no closed-form solution, we can simplify things further using refinements of the above bound.

14. Note that this is never exactly achievable in practice, even with an infinite capacity network for computing $\boldsymbol{\Sigma}_x^{(i)}$, since it would require a unique network for each data sample; however, it nonetheless serves as a useful analysis tool.

For this purpose consider any possible candidate solution $\hat{\boldsymbol{W}} = \boldsymbol{\Psi}$ and $\hat{\boldsymbol{\mu}}_z^{(i)} = \boldsymbol{\pi}^{(i)}$ (not necessarily one that coincides with $\boldsymbol{U}$ and the optimal subspace), and define

$$\Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}) \triangleq \text{supp}_\alpha \left[\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi} \right]. \tag{62}$$

Without loss of generality we also specify that

$$\boldsymbol{S}_z^{(i)} \triangleq \boldsymbol{\Xi}^{(i)} \boldsymbol{D}^{(i)}, \tag{63}$$

where $\boldsymbol{\Xi}^{(i)} \in \mathbb{R}^{d \times \kappa}$ has orthonormal columns and $\boldsymbol{D}^{(i)}$ is a diagonal matrix with

$$\left[\boldsymbol{D}^{(i)} \right]_{kk} = \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right), \tag{64}$$

and $\boldsymbol{\sigma}^{(i)} = [\sigma_1^{(i)}, \ldots, \sigma_\kappa^{(i)}]^\top \in \mathbb{R}_+^\kappa$ is an arbitrary non-negative vector. Any general $\boldsymbol{\Sigma}_z^{(i)} = \boldsymbol{S}_z^{(i)} \left(\boldsymbol{S}_z^{(i)} \right)^\top$, with singular values bounded by α, is expressible via this format. We then reexpress (61) as

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;\geq\; & \sum_i \left\{ \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_{j \in \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\left[x_j^{(i)} - \boldsymbol{\psi}_{j\cdot} \boldsymbol{\pi}^{(i)} - \boldsymbol{\psi}_{j\cdot} \boldsymbol{\Xi}^{(i)} \boldsymbol{D}^{(i)} \boldsymbol{\epsilon} \right]^2 \right) \right] \right. \\
& + \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_{j \notin \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\left[O(\alpha) + \boldsymbol{\psi}_{j\cdot} \boldsymbol{\Xi}^{(i)} \boldsymbol{D}^{(i)} \boldsymbol{\epsilon} \right]^2 \right) \right] \qquad (65) \\
& \left. + \; \text{tr} \left[\boldsymbol{\Xi}^{(i)} \left(\boldsymbol{D}^{(i)} \right)^2 \left(\boldsymbol{\Xi}^{(i)} \right)^\top \right] - \log \left| \boldsymbol{\Xi}^{(i)} \left(\boldsymbol{D}^{(i)} \right)^2 \left(\boldsymbol{\Xi}^{(i)} \right)^\top \right| + \|\boldsymbol{\pi}^{(i)}\|_2^2 \right\} + O(1), \\
= \; & \sum_i \left\{ \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_{j \in \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\left[x_j^{(i)} - \boldsymbol{\psi}_{j\cdot} \boldsymbol{\pi}^{(i)} - \sum_k \bar{\psi}_{jk} \cdot \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right) \cdot \epsilon_k \right]^2 \right) \right] \right. \\
& + \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_{j \notin \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\left[O(\alpha) + \sum_k \bar{\psi}_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right) \cdot \epsilon_k \right]^2 \right) \right] \qquad (66) \\
& \left. + \; \sum_k \xi_\alpha \left[\left(\sigma_k^{(i)} \right)^2 \right] - \sum_k \log \xi_\alpha \left[\left(\sigma_k^{(i)} \right)^2 \right] + \|\boldsymbol{\pi}^{(i)}\|_2^2 \right\} + O(1),
\end{aligned}$$

where $\bar{\psi}_{jk}^{(i)}$ is the k-th element of the vector $\boldsymbol{\psi}_{j\cdot} \boldsymbol{\Xi}^{(i)}$. We can now analyze any given point $\{\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Xi}^{(i)}, \boldsymbol{\sigma}^{(i)}\}_{i=1}^n$ as α becomes small. The first term can be shown to be $\Theta(1)$ with all

other variables fixed,[15] leading to the revised bound

$$\begin{aligned}
\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;\geq\; & \sum_i \left\{ \mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\sum_{j \notin \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\left[O(\alpha) + \sum_k \bar{\psi}_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right) \cdot \epsilon_k \right]^2 \right) \right] \right. \qquad (67) \\
& \left. - \; \sum_k \log \xi_\alpha \left[\left(\sigma_k^{(i)} \right)^2 \right] \right\} + \Theta(1),
\end{aligned}$$

where the terms $\sum_k \xi_\alpha \left[\left(\sigma_k^{(i)} \right)^2 \right]$ and $\|\boldsymbol{\pi}^{(i)}\|_2^2$ have also been absorbed into $\Theta(1)$.

Given that

$$\mathbb{E}_{p(\boldsymbol{\epsilon})} \left[\log \xi_\alpha \left(\left[O(\alpha) + \boldsymbol{a}^\top \boldsymbol{\epsilon} \right]^2 \right) \right] = \log \xi_\alpha \left[\boldsymbol{a}^\top \boldsymbol{a} \right] + O(1) \geq \log \alpha + O(1) \tag{68}$$

for any vector $\boldsymbol{a}$, we have the new bound

$$\begin{aligned}
& \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \qquad (69) \\
\geq \; & \sum_i \left\{ \sum_{j \notin \Delta_\alpha^{(i)}(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)})} \log \xi_\alpha \left(\sum_k \left[\bar{\psi}_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right) \right]^2 \right) - \sum_k \log \xi_\alpha \left[\left(\sigma_k^{(i)} \right)^2 \right] \right\} + \Theta(1).
\end{aligned}$$

If then we choose $\sigma_k^{(i)} = 0$ for all $i = 1, \ldots n$ and $k = 1, \ldots, \kappa$, then

$$\log \xi_\alpha \left(\sum_k \left[\bar{\psi}_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}} \left(\sigma_k^{(i)} \right) \right]^2 \right) = \log \alpha + \Theta(1) \tag{70}$$

and we obtain the lower bound

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) \;\geq\; \sum_i \left(d - \kappa - \left| \Delta_\alpha^{(i)} \left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)} \right) \right| \right) \log \alpha + \Theta(1). \tag{71}$$

Additionally, if any set $\Delta_\alpha^{(i)} \left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)} \right)$ exists such that

$$\sum_i \left(d - \kappa - \left| \Delta_\alpha^{(i)} \left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)} \right) \right| \right) \leq \sum_i \left(d - \kappa - \|\boldsymbol{s}^{(i)}\|_0 \right), \tag{72}$$

then $\boldsymbol{s}^{(i)}$ cannot be part of the unique, feasible solution to (10), i.e., we could use the support pattern from each $\Delta_\alpha^{(i)} \left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)} \right)$ to find a different feasible solution with equal or lower value of $n \cdot \text{rank}[\boldsymbol{L}] + \|\boldsymbol{S}\|_0$, which would violate either the uniqueness or optimality of the original solution. Therefore, we have established that with $\sigma_k^{(i)} = O(\alpha)$ for all i and k, the resulting bound on $\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi})$ is essentially no better than (56), or the same bound we

15. Note that $x_j^{(i)} - \boldsymbol{\psi}_{j\cdot} \boldsymbol{\pi}^{(i)} = \Theta(1)$ for all $j \in \Delta_\alpha^{(i)} \left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)} \right)$ and $\int \log \xi_\alpha \left[(\Theta(1) + x)^2 \right] \mathcal{N}(x; 0, \gamma) dx = \Theta(1)$ for any variance $\gamma > 0$.

had before from our feasible trial solution. Moreover, the resulting $\hat{\boldsymbol{W}} = \boldsymbol{\Psi}$ that maximizes this bound, as well as the implicit

$$\hat{\boldsymbol{\Sigma}}_x^{(i)}\left(\hat{\boldsymbol{\mu}}_z\left[\boldsymbol{x}^{(i)}\right]\right) = \text{diag}\left[\left(\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\right)^2\right], \tag{73}$$

will necessarily satisfy (12). We then only need consider whether other choices for $\sigma_k^{(i)}$ can do better.

Let $\widetilde{\boldsymbol{\Psi}}^{(i)}$ denote the the rows of $\boldsymbol{\Psi}^{(i)}$ associated with row indeces $j \notin \Delta_\alpha^{(i)}\left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}\right)$, meaning the indices at which we assume no sparse corruption term exists. Additionally, define $\boldsymbol{B}^{(i)} \triangleq \widetilde{\boldsymbol{\Psi}}^{(i)}\boldsymbol{\Xi}^{(i)}$. This implies that

$$\begin{aligned}\sum_{j \notin \Delta_\alpha^{(i)}\left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}\right)} \log \xi_\alpha\left(\max_k\left[\bar{\psi}_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}}\left(\sigma_k^{(i)}\right)\right]^2\right) - \sum_k \log \xi_\alpha\left[\left(\sigma_k^{(i)}\right)^2\right] \\ = \sum_j \log \xi_\alpha\left(\max_k\left[B_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}}\left(\sigma_k^{(i)}\right)\right]^2\right) - \sum_k \log \xi_\alpha\left[\left(\sigma_k^{(i)}\right)^2\right].\end{aligned} \tag{74}$$

Contrary to our prior assumption $\boldsymbol{\sigma}^{(i)} = \boldsymbol{0}$, now consider any solution with $\|\boldsymbol{\sigma}^{(i)}\|_0 = \beta > 0$. For the time being, we also assume that $\boldsymbol{B}^{(i)}$ is full column rank. These conditions imply that

$$\sum_j \log \xi_\alpha\left(\max_k\left[B_{jk}^{(i)} \cdot \xi_{\sqrt{\alpha}}\left(\sigma_k^{(i)}\right)\right]^2\right) \geq \left(d - \beta - \left|\Delta_\alpha^{(i)}\left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}\right)\right|\right)\log \alpha + \Theta(1) \tag{75}$$

since at least β elements of the summation over j must now be order $\Theta(1)$. By assumption we also have $\sum_k \log \xi_\alpha\left[\left(\sigma_k^{(i)}\right)^2\right] = (\kappa - \beta)\log\alpha + \Theta(1)$. Combining with (75), we see that such a solution is equivalent or worse than (71). So the former is the best we can do at any value of $\{\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}, \boldsymbol{\Xi}^{(i)}, \boldsymbol{\sigma}^{(i)}\}_{i=1}^n$, provided that $\boldsymbol{B}^{(i)}$ is full rank, and obtaining the optimal value of $\Delta_\alpha^{(i)}\left(\boldsymbol{\Psi}, \boldsymbol{\pi}^{(i)}\right)$ implies that (12) holds.

However, if $\boldsymbol{B}^{(i)}$ is not full rank it would indeed entail that (74) could be reduced further, since a nonzero element of $\boldsymbol{\sigma}^{(i)}$ would not increase the first summation, while it would reduce the second. But if such a solution were to exist, it would violate the uniqueness assumption of the theorem statement. To see this, note that $\text{rank}[\boldsymbol{B}^{(i)}] = \text{rank}[\widetilde{\boldsymbol{\Psi}}^{(i)}]$ since $\boldsymbol{\Xi}^{(i)}$ is orthogonal, so if the former is not full column rank, neither is the latter. And if $\widetilde{\boldsymbol{\Psi}}^{(i)}$ is not full column rank, there will exist multiple solutions such that $\|\boldsymbol{x}^{(i)} - \boldsymbol{\Psi}\boldsymbol{\pi}^{(i)}\|_0 = \|\boldsymbol{s}^{(i)}\|_0$ or equivalently $\|\boldsymbol{x}^{(i)} - \boldsymbol{U}\boldsymbol{v}^{(i)}\|_0 = \|\boldsymbol{s}^{(i)}\|_0$ in direct violation of the uniqueness clause.

Therefore to conclude, a lower bound on the VAE cost is in fact the same order as that obtainable by our original trial solution. If this lower bound is not achieved, we cannot be at a minimizing solution, and any solution achieving this bound must satisfy (12).

D.6 Generalization to Case where $\kappa > \text{rank}[\boldsymbol{U}]$

Finally, we briefly consider the case where $\kappa > \text{rank}[\boldsymbol{U}] \triangleq \tau$, meaning that $\boldsymbol{W}$ contains redundant columns that are unnecessary in producing an optimal solution to (10). The candidate solution described in Section D.1 can be expanded via $\hat{\boldsymbol{W}} = \left[\boldsymbol{\Psi},\ \boldsymbol{0}_{[d\times(\kappa-\tau)]}\right]$, $\boldsymbol{\mu}_z^{(i)} = \left[(\boldsymbol{\pi}^{(i)})^\top,\ \boldsymbol{0}_{[1\times(\kappa-\tau)]}\right]^\top$, and $\hat{\boldsymbol{\Sigma}}_z^{(i)} = \text{diag}\left[\alpha\boldsymbol{1}_{[\tau\times 1]}^\top,\ \boldsymbol{1}_{[(\kappa-\tau)\times 1]}^\top\right]$ such that the same objective function value is obtained.

Now consider the general case where $\kappa \geq \text{rank}[\hat{\boldsymbol{W}}] > \tau$. If we review the lower bound described in Section D.5, with this general $\hat{\boldsymbol{W}}$ replacing $\boldsymbol{\Psi}$, it can be shown that $\hat{\boldsymbol{\Sigma}}_z^{(i)}$ will be forced to have additional diagonal elements lowered to α, increasing the achievable objective by at least $-\log\alpha$ per sample. The details are not especially enlightening and we omit them here for brevity. Consequently, at any minimizer we must have $\text{rank}[\hat{\boldsymbol{W}}] = \tau$. ■

Appendix E. Proof of Corollary 4

Proof Under the stated conditions, the partially-affine VAE cost simplifies to the function

$$\begin{aligned}\mathcal{L}(\boldsymbol{W}, \boldsymbol{\Sigma}_x, \boldsymbol{\mu}_z) &= \sum_i \left\{\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)}\right)^\top \left(\boldsymbol{\Sigma}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)}\right) + \log\left|\boldsymbol{\Sigma}_x^{(i)}\right| + \|\boldsymbol{\mu}_z^{(i)}\|_2^2\right\} \\ &= \sum_i \left\{\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\beta^{-1}\beta\boldsymbol{\mu}_z^{(i)}\right)^\top \left(\boldsymbol{\Sigma}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\beta^{-1}\beta\boldsymbol{\mu}_z^{(i)}\right)\right. \\ &+ \left.\log\left|\boldsymbol{\Sigma}_x^{(i)}\right| + \beta^2\|\boldsymbol{\mu}_z^{(i)}\|_2^2\right\},\end{aligned} \tag{76}$$

where $\beta > 0$ is an arbitrary scaler, $\boldsymbol{\Sigma}_x^{(i)} \triangleq \boldsymbol{\Sigma}_x\left(\boldsymbol{\mu}_z^{(i)}; \boldsymbol{\theta}\right)$, and $\boldsymbol{\mu}_z^{(i)} \triangleq \boldsymbol{\mu}_z(\boldsymbol{x}^{(i)}; \boldsymbol{\phi})$. Taking the limit as $\beta \to 0^+$, we can minimize (76) while ignoring the $\beta^2\|\boldsymbol{\mu}_z^{(i)}\|_2^2$ regularization factor. Consequently, we can without loss of generality consider minimization of

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{\Sigma}_x, \boldsymbol{\mu}_z) \equiv \sum_i \left\{\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)}\right)^\top \left(\boldsymbol{\Sigma}_x^{(i)}\right)^{-1}\left(\boldsymbol{x}^{(i)} - \boldsymbol{W}\boldsymbol{\mu}_z^{(i)}\right) + \log\left|\boldsymbol{\Sigma}_x^{(i)}\right|\right\}, \tag{77}$$

ignoring any explicit reparameterization by β for convenience. If we optimize over $\boldsymbol{\Sigma}_x^{(i)}$ in the feasible region $\bar{\mathcal{S}}_\alpha^d$ and plug in the resulting value, then (77) reduces to the new cost

$$\mathcal{L}(\boldsymbol{W}, \boldsymbol{\mu}_z) \equiv \sum_{i,j} \log \xi_\alpha\left(\left[x_j^{(i)} - \boldsymbol{w}_{j\cdot}\boldsymbol{\mu}_z^{(i)}\right]^2\right), \tag{78}$$

an immaterial constant notwithstanding. Given that $\lim_{t\to 0}\frac{1}{t}\left(|x|^t - 1\right) = \log|x|$, and $\lim_{t\to 0}\sum_j |x_j|^p = \|\boldsymbol{x}\|_0$, then up to an irrelevant scaling factor and additive constant, the stated result follows. ■

Appendix F. Proof of Theorem 5

Proof Based on the stated conditions, the VAE objective simplifies to

$$\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) = \sum_i \left\{\mathbb{E}_{q_{\boldsymbol{\phi}}(z|\boldsymbol{x}^{(i)})}\left[\frac{1}{\lambda_x}\left\|\boldsymbol{x}^{(i)} - \boldsymbol{\mu}_x(z; \boldsymbol{\theta})\right\|^2\right] + d\log\lambda_x + \lambda_z - \log\lambda_z + \left(\boldsymbol{a}^\top\boldsymbol{x}^{(i)}\right)^2\right\}. \tag{79}$$

Now choose some $\hat{\boldsymbol{a}}$ such that $\hat{\mu}_z^{(i)} = \hat{\boldsymbol{a}}^\top \boldsymbol{x}^{(i)}$ has a unique value for every sample $\boldsymbol{x}^{(i)}$ (here we assume that each sample is unique, although this assumption can be relaxed). We then define the function $h : \mathbb{R} \to \{1, \ldots, n\}$ as

$$h(z) \triangleq \arg \min_{i \in \{1,\ldots,n\}} \|z - \mu_z^{(i)}\|_2. \tag{80}$$

and the piecewise linear decoder mean function

$$\boldsymbol{\mu}_x(z; \boldsymbol{\theta}) = \boldsymbol{x}^{(h(z))}. \tag{81}$$

Given these definitions, (79) becomes

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) &= \sum_i \left\{ \mathbb{E}_{q_{\boldsymbol{\phi}}(z|\boldsymbol{x}^{(i)})} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h(z))} \right\|^2 \right] + d \log \lambda_x + \lambda_z - \log \lambda_z + \left(\hat{\mu}_z^{(i)} \right)^2 \right\} \\ &= \sum_i \left\{ \mathbb{E}_{p(\epsilon)} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h[\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon])} \right\|^2 \right] + d \log \lambda_x + \lambda_z - \log \lambda_z + \left(\hat{\mu}_z^{(i)} \right)^2 \right\}. \end{aligned} \tag{82}$$

Now define the set

$$\mathcal{S}^{(i)} \triangleq \left\{ \epsilon : h\left(\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon \right) = i \right\}, \tag{83}$$

which represents the set of ϵ that quantize to the correct index. We then have

$$\begin{aligned} &\mathbb{E}_{p(\epsilon)} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h[\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon])} \right\|^2 \right] \\ &= \int_{\epsilon \in \mathcal{S}^{(i)}} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h[\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon])} \right\|^2 \right] p(\epsilon) d\epsilon + \int_{\epsilon \notin \mathcal{S}^{(i)}} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h[\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon])} \right\|^2 \right] p(\epsilon) d\epsilon \\ &= \int_{\epsilon \notin \mathcal{S}^{(i)}} \left[\tfrac{1}{\lambda_x} \left\| \boldsymbol{x}^{(i)} - \boldsymbol{x}^{(h[\hat{\mu}_z^{(i)} + \sqrt{\lambda_z}\epsilon])} \right\|^2 \right] p(\epsilon) d\epsilon \\ &\leq \int_{\epsilon \notin \mathcal{S}^{(i)}} \tfrac{\eta}{\lambda_x} p(\epsilon) d\epsilon \\ &= \tfrac{\eta}{\lambda_x} P\left(\epsilon \notin \mathcal{S}^{(i)} \right), \end{aligned} \tag{84}$$

where

$$\eta \triangleq \max_{i,j \in \{1,\ldots,n\}, i \neq j} \|\boldsymbol{x}^{(i)} - \boldsymbol{x}^{(j)}\|_2^2, \tag{85}$$

the maximal possible quantization error. Now we also define

$$\rho \triangleq \max_{i,j \in \{1,\ldots,n\}, i \neq j} \frac{1}{2} \|\hat{\mu}_z^{(i)} - \hat{\mu}_z^{(j)}\|_2^2, \tag{86}$$

which is half the minimum distance between any two $\hat{\mu}_z^{(i)}$ and $\hat{\mu}_z^{(j)}$, with $i \neq j$. Then

$$\begin{aligned} P\left(\epsilon \notin \mathcal{S}^{(i)} \right) &\leq P\left(\sqrt{\lambda_z}\epsilon > \rho \right) \\ &\leq \tfrac{\lambda_z}{\rho^2} \end{aligned} \tag{87}$$

by Chebyshev's inequality as was used in proving Theorem 3. This implies that (82) can be bounded via

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\phi}) &\leq \sum_i \left\{ \tfrac{\eta}{\lambda_x} P\left(\epsilon \notin \mathcal{S}^{(i)} \right) + d \log \lambda_x + \lambda_z - \log \lambda_z + \left(\hat{\mu}_z^{(i)} \right)^2 \right\} \\ &= \sum_i \left\{ \tfrac{\eta}{\rho^2} + (d-1) \log \alpha + \alpha + \left(\hat{\mu}_z^{(i)} \right)^2 \right\} \end{aligned} \tag{88}$$

assuming we are at the trial solution $\hat{\lambda}_x = \hat{\lambda}_z = \alpha$. As we allow $\alpha \to 0$, this expression is unbounded from below, and as an upper bound on the VAE objective, the theorem follows. Incidentally, it should also be possible to prove that for α sufficiently small, no other solution can do appreciably better in terms of the dominate $(d-1)\log\alpha$ factor, but we will reserve this for future work. ■

References

D.J. Bartholomew and M. Knott. Latent variable models and factor analysis. In *Kendalls Library of Statistics 7, 2nd Edition*, 1999.

Y. Bengio. Learning deep architectures for AI. In *Foundations and Trends in Machine Learning*, 2009.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *International Conference on Computational Statistics (ICCS)*, 2010.

H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. In *Biological Cybernetics*, 1988.

Y. Burda, R. Grosse, and R. Salakhutdinov. Importance weighted autoencoders. *arXiv:1509.00519*, 2015.

E. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11, 2011.

V. Chandrasekaran, S. Sanghavi, P.A. Parrilo, and A.S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20(1), 1999.

A. Choromanska, M. Henaff, M. Mathieu, G.B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015a.

A. Choromanska, Y. LeCun, and G.B. Arous. Open problem: The landscape of the loss surfaces of multilayer networks. In *Conference on Learning Theory (CoLT)*, 2015b.

X. Ding, L. He, and L. Carin. Bayesian robust principal component analysis. *IEEE Transactions on Image Processing*, 20(12):3419–3430, 2011.

D.L. Donoho and M. Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100 (5):2197–2202, 2003.

A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 658–666, 2016.

E. Elhamifar and R. Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(11):2765–2781, 2013.

D.J.H. Garling. *Inequalities: A Journey into Linear Analysis*. Cambridge University Press, 2007.

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision (ICCV)*, 2015.

M. Hoffman. Why variational inference gives bad parameter estimates. In *Advances in Variational Inference, NIPS Worshop*, 2014.

K. Kawaguchi. Deep learning without poor local minima. In *Advances in Neural Information Processing Systems (NIPS)*, 2016.

T.H. Ohand H. Kim, Y.W. Tai, J.C. Bazin, and I.S. Kweon. Partial sum minimization of singular values in RPCA for low-level vision. In *International Conference on Computer Vision (ICCV)*, 2013.

D. Kingma and M. Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations (ICLR)*, 2014.

D. Kingma, D. Rezende, S. Mohamed, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems (NIPS)*, 2014.

D.P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

D.P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems (NIPS)*, pages 4743–4751, 2016.

A.B.L Larsen, S.K. Sønderby, H. Larochelle, and O. Winther. Autoencoding beyond pixels using a learned similarity metric. *arXiv preprint arXiv:1512.09300*, 2015.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998. ISSN 0018-9219.

Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

C. Lu, J. Feng, Z. Lin, and S. Yan. Correlation adaptive subspace segmentation by trace lasso. In *International Conference on Computer Vision (ICCV)*, 2013.

L. Maaløe, C.K. Sønderby, S.K. Sønderby, and O. Winther. Auxiliary deep generative models. In *International Conference on Machine Learning (ICML)*, 2016.

E. Mansimov, E. Parisotto, J.L. Ba, and R. Salakhutdinov. Generating images from captions with attention. In *International Conference on Learning Representations (ICLR)*, 2016.

V. Nair and G.E. Hinton. Rectified linear units improve restricted Boltzman machines. In *International Conference on Machine Learning (ICCV)*, 2010.

A.V.D. Oord, N. Kalchbrenner, and K. Kavukcuoglu. Pixel recurrent neural networks. *arXiv preprint arXiv:1601.06759*, 2016.

B.D. Rao, K. Engan, S. F. Cotter, J. Palmer, and K. Kreutz-Delgado. Subset selection in noise based on diversity measure minimization. *IEEE Transactions on Signal Processing*, 51(3):760–770, 2003.

S. Rao, R. Tron, R. Vidal, and Y. Ma. Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(10):1832–1845, 2010.

D.J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning (ICML)*, 2014.

A.M. Saxe, J.L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

C.K. Sønderby, T. Raiko, L. Maaløe, S.K. Sønderby, and O. Winther. How to train deep variational autoencoders and probabilistic ladder networks. In *arXiv:1602.02282*, 2016.

M. Tipping and C. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Y. Wang, B. Dai, G. Hua, J. Aston, and D. Wipf. Green generative modeling: Recycling dirty data using recurrent variational autoencoders. In *Uncertainty in Artificial Intelligence (UAI)*, 2017.

D. Wipf. Non-convex rank minimization via an empirical Bayesian approach. In *Uncertainty in Artificial Intelligence (UAI)*, 2012.

Journal of Machine Learning Research 19 (2018) 1-43 Submitted 3/18; Published 8/18

An Efficient and Effective Generic Agglomerative Hierarchical Clustering Approach

Julien Ah-Pine julien.ah-pine@univ-lyon2.fr
University of Lyon, Lyon 2
ERIC EA3083
5 avenue Pierre Mendès France
69676 Bron Cedex, France

Editor: Maya Gupta

Abstract

We introduce an agglomerative hierarchical clustering (AHC) framework which is generic, efficient and effective. Our approach embeds a sub-family of Lance-Williams (LW) clusterings and relies on inner-products instead of squared Euclidean distances. We carry out a constrained bottom-up merging procedure on a sparsified normalized inner-product matrix. Our method is named SNK-AHC for Sparsified Normalized Kernel matrix based AHC. SNK-AHC is more scalable than the classic dissimilarity matrix based AHC. It can also produce better results when clusters have arbitrary shapes. Artificial and real-world benchmarks are used to exemplify these points. From a theoretical standpoint, SNK-AHC provides another interpretation of the classic techniques which relies on the concept of weighted penalized similarities. The differences between group average, Mcquitty, centroid, median and Ward, can be explained by their distinct averaging strategies for aggregating clusters inter-similarities and intra-similarities. Other features of SNK-AHC are examined. We provide sufficient conditions in order to have monotonic dendrograms, we elaborate a stored data matrix approach for centroid and median, we underline the diagonal translation invariance property of group average, Mcquitty and Ward and we show to what extent SNK-AHC can determine the number of clusters.

Keywords: Agglomerative hierarchical clustering, Lance-Williams formula, Kernel methods, Scalability, Manifold learning.

1. Introduction

Clustering is the process of discovering homogeneous groups among a set of objects. There are many clustering methods and one way to differentiate one approach from another one is by the classification scheme they are based upon. On the one hand, flat clustering provides a partition of the elements. On the other hand, hierarchical clustering outputs a set of nested partitions. The latter classification type is represented by a binary tree named dendrogram.

Hierarchical clustering presents several advantages compared to flat clustering. Firstly, a dendrogram is more informative than a single partition because it provides more insights about the relationships between objects and clusters. Secondly, there is no requirement to set the number of clusters *a priori* unlike most of flat clustering techniques.

In this paper, we focus on hierarchical clustering methods. There are two kinds of procedures: agglomerative and divisive. The former type builds the dendrogram in a bottom-up fashion whereas the latter case uses a top-down approach. We focus on *Agglomerative Hierarchical Clustering (AHC)*. Suppose we are given a pairwise dissimilarity matrix between the elements we want to cluster. The AHC bottom-up procedure initializes a trivial partition composed of singletons then, iteratively merges the two closest clusters until all items are grouped together.

In any AHC method, after each merge, it is required to compute the dissimilarity measure between the newly formed group and other existing clusters. In fact, there are as many AHC methods as dissimilarity measures. Despite the great number of approaches found in the literature, Lance and Williams (1967) proposed a parametric formula (LW formula) that generalizes a lot of them.

The bottom-up strategy described above with the LW formula form the usual stored *Dissimilarities*[1] *based AHC (D-AHC)* framework. Due to its simplicity and flexibility, it has been studied in many research works, implemented in many programming languages and successfully applied in many domains.

However, D-AHC suffers from important scalability issues since, with respect to the number of objects, it has a quadratic memory complexity and a cubic time complexity. These drawbacks severely limit the application of D-AHC to very large data sets.

In this context, our work aims at designing an AHC approach that is equivalent to D-AHC but which can be extended in order to reduce the computational costs. Furthermore, the approach we define is able to take into account the natural geometry of the data. It is thus an unsupervised approach for manifold learning as well.

In a nutshell, the contributions of the paper are the following ones:

- We focus on a sub-part of the LW formula and we establish a more general model which relies on inner-products instead of squared Euclidean distances. In this case, we need two parametric recurrence equations instead of one. Since our model relies on inner-products, it encompasses Reproducing Kernel Hilbert Spaces (RKHS) through the use of kernel functions. This first model called *Kernel matrix based Agglomerative Hierarchical Clustering (K-AHC)* can be viewed as a kind of "dual" of D-AHC when squared Euclidean distances are used as dissimilarities.

- In the usual D-AHC framework, the geometric techniques centroid, median and Ward can be carried out by using data matrices instead of distance matrices. On the contrary, the graph methods group average and Mcquitty do not enjoy such a property. We show that K-AHC enables a stored data[2] matrix approach for the two latter schemes.

- The median and centroid schemes can suffer from pathological behaviors because they can produce reversals in the dendrogram. This phenomenon appears when at an iteration, the dissimilarity value between two clusters becomes lower than the dissimilarity values observed at previous iterations. Median and centroid are said to provide non-monotonic dendrograms. In fact, Ward can be seen as a modification of

1. The terms "stored dissimilarities/similarities" and "stored data" approaches were coined by Anderberg (1973). The former one means that the input is the pairwise proximity matrix whereas the latter one indicates that the input is the data matrix where objects are described by a set of attributes.
2. Defined in the previous footnote.

centroid which enables solving the non-monotonicity issue of the latter scheme. In the same spirit, we introduce a new scheme called w-median which solves the non-monotonicity problem of the median technique.

- We propose to project the data points on an hypersphere and to shift them in order to obtain non-negative inner-products values. As a result, we obtain a *Normalized Kernel (NK) matrix* which can also be interpreted as a similarity matrix satisfying several conditions such as maximal self-similarity. In this case, we can interpret our model in terms of *weighted penalized similarities* and we show that the main differences between classic techniques rely on distinct averaging operations of inter-similarities and of intra-similarities as well.

- Given a NK matrix, we can apply sparsification procedures in order to remove non-relevant similarity relationships between objects. The resulting output is called *Sparsified Normalized Kernel (SNK) matrix* and it can be viewed as the weighted adjacency matrix of a sparse similarity graph. Then, we apply K-AHC on a SNK matrix but with the *constraint* that two clusters can be merged together providing that they have a non-null inter-similarity value. Our approach is called *Sparsified Normalized Kernel matrix based AHC (SNK-AHC)*. SNK-AHC has much lower computational costs compared to K-AHC and D-AHC, both in terms of memory and running time. Moreover, the sparsification enables capturing the intrinsic geometry of the data.

- Unlike a NK matrix, a SNK matrix is not positive semi-definite. Therefore, from a general perspective, SNK-AHC can not be interpreted from a geometrical point of view unlike K-AHC. Nevertheless, we show that in the particular cases of group average, Mcquitty and Ward, SNK-AHC still implicitly acts in an Hilbert space. This is due to the fact that these schemes are invariant with respect to any translation of the diagonal of the SNK matrix.

- By interpreting SNK-AHC in the framework of graph theory, we demonstrate that the bottom-up procedure emulates the same kinds of operations employed in order to determine the connected components of an undirected graph. As a result, we show that SNK-AHC can automatically determine the number of clusters when the latter ones are seen as connected components of a similarity graph.

- We illustrate the aforementioned properties of K-AHC and SNK-AHC on two artificial data sets. In addition, we show the superiority of SNK-AHC over D-AHC on two real-world benchmarks. Our experimental results confirm that SNK-AHC is much more scalable than the classic D-AHC. Last but not least, SNK-AHC can also outperform D-AHC in terms of clustering quality. In fact, in many cases, our approach is both more efficient and more effective than D-AHC.

The remainder of the paper is organized as follows. In section 2, we introduce the notations and some useful definitions. In section 3, we review the basics of D-AHC and of the LW formula. Then, in section 4, we introduce our K-AHC model by establishing an inner-product based expression that embeds the LW sub-equation we are interested in. Several features of K-AHC are examined as well. Afterward, we present SNK-AHC and

study its properties in section 5. Section 6 is dedicated to the experiments which are carried out on both artificial and real-world data sets. After having introduced our approach and exhibited its properties, we present and discuss in section 7 related research works. Finally, in section 8, we conclude the paper and we sketch future work as well.

2. Notations and Definitions

The set of objects (or elements or items or points) to cluster is denoted by $\mathbb{O}$ and $|\mathbb{O}|$ represents its cardinal. We suppose throughout the paper that $|\mathbb{O}|= n$. The usual AHC algorithm takes as input a pairwise dissimilarity matrix.

Definition 1 (Dissimilarity matrix) *A pairwise dissimilarity matrix of elements in $\mathbb{O}$ is denoted $\mathbf{D}$. Given $|\mathbb{O}|= n$, $\mathbf{D}$ is a square matrix of order n satisfying the following conditions:*

$$\begin{cases} \mathbf{D}_{ab} \geq 0, \quad \forall a, b \in \mathbb{O} & \textit{(non-negativity)} \\ \mathbf{D}_{ab} = \mathbf{D}_{ba}, \quad \forall a, b \in \mathbb{O} & \textit{(symmetry)} \end{cases}, \quad \forall a, b \in \mathbb{O}$$

Let $2^{\mathbb{O}}$ denote the set of subsets of $\mathbb{O}$. The AHC procedure builds a set of nested partitions of $\mathbb{O}$. We denote by the letters a, b, c, d, e, f any singletons (or objects) of $\mathbb{O}$, whereas i, j, k, l, m correspond to any item (or clusters) in $2^{\mathbb{O}}$. The cardinal of k is denoted $|k|$. Given k and l, their fusion (or merge or union) is denoted by (kl).

The AHC algorithm is an iterative procedure with $n - 1$ steps. We denote by $\mathbb{T} = \{1, 2, \ldots, n-1\}$ the set of iterations and use t to designate any of its elements.

Let $\mathbb{C}^t$ denote the set of existing clusters at iteration t. It is a partition of $\mathbb{O}$ with $n-t+1$ subsets. We denote by $\mathbf{D}^t$, the dissimilarity matrix of clusters in $\mathbb{C}^t$. It is thus a symmetric square matrix of order $n - t + 1$ satisfying the conditions given in Definition 1.

The AHC bottom-up algorithm produces a set of nested partitions represented by a tree-diagram called dendrogram.

Definition 2 (Dendrogram) *A dendrogram representing a hierarchical clustering of $\mathbb{O}$ is denoted D. Given $|\mathbb{O}|= n$, D is a rooted binary tree with n leaves and $2n-1$ nodes in total. Each node i corresponds to a subset of objects. Any two distinct nodes i and j of D are such that $i \subset j$ or $j \subset i$ or $i \cap j = \emptyset$. Besides, each node i is assigned a non-negative value called the height denoted by $H(i)$.*

Since any node in a dendrogram represents a cluster, we adopt the same notations as precised above: a, b, c, d, e, f are nodes that designate singletons only, while i, j, k, l, m correspond to subsets of $\mathbb{O}$.

As an illustration, we give in Figure 1 an example of a dendrogram of the set $\mathbb{O} = \{a, b, c, d, e, f\}$.

Definition 3 (Monotonic dendrogram) *A dendrogram D is monotonic if and only if $i \subset j \Leftrightarrow H(i) \leq H(j)$, for any two distinct nodes i and j.*

The dendrogram represented in Figure 1 is monotonic. Indeed, the height of larger nodes are higher than smaller ones and any path from a leaf to the root has no reversal.

The following definition is used to compare dendrograms.

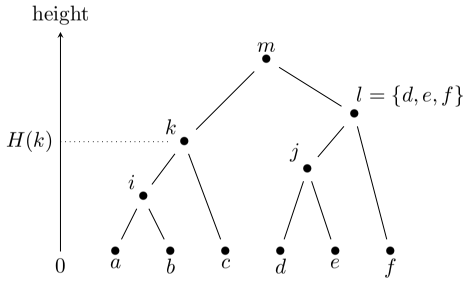


Figure 1: Illustration of a dendrogram.

Definition 4 (Sequence of merges) *A sequence of merges representing a hierarchical clustering of $\mathbb{O}$ is denoted M. Given $|\mathbb{O}|= n$, $M = (m_1, \ldots, m_{n-1})$ is a sequence of $n-1$ couples of disjoint subsets of elements of $\mathbb{O}$. For all $t \in \mathbb{T}$, $m_t = (m_t^1, m_t^2) \in \{(i,j) \in 2^{\mathbb{O}} \times 2^{\mathbb{O}}, i \cap j = \emptyset\}$. Any two elements m_s and m_t of M satisfy the following condition: if $s < t$ then $(m_s^1 \cup m_s^2) \subset (m_t^1 \cup m_t^2)$ or $(m_s^1 \cup m_s^2) \cap (m_t^1 \cup m_t^2) = \emptyset$.*

Whether an AHC technique produces a monotonic dendrogram or not, it always groups two clusters into a larger one at each iteration. If we consider the pairs of clusters that are fused at each step of the AHC bottom-up procedure (regardless the height values) then, it is clear that there is a one-to-one correspondence between dendrograms and sequences of merges.

For example, the sequence of merges in correspondence with the dendrogram provided in Figure 1 is $M = \{(\{a\},\{b\}),(\{d\},\{e\}),(\{a,b\},\{c\}),(\{d,e\},\{f\}),(\{a,b,c\},\{d,e,f\})\}$.

The following definition is used to establish the equivalence between two different AHC algorithms.

Definition 5 (Equivalent dendrograms) *Two dendrograms D and D' of a set of objects $\mathbb{O}$ are equivalent if their respective sequence of merges M and M' are identical.*

Eventually, we introduce the definition of a similarity matrix.

Definition 6 (Similarity matrix) *A pairwise similarity matrix of elements in $\mathbb{O}$ is denoted $\mathbf{S}$. Given $|\mathbb{O}|= n$, $\mathbf{S}$ is a square matrix of order n satisfying the following conditions:*

$$\begin{cases} \mathbf{S}_{ab} \geq 0 \ \textit{(non-negativity)} \\ \mathbf{S}_{ab} = \mathbf{S}_{ba} \ \textit{(symmetry)} \\ \mathbf{S}_{aa} \geq \mathbf{S}_{ab} \ \textit{(maximal self-similarity)} \end{cases}, \quad \forall a, b \in \mathbb{O}$$

The maximal self-similarity condition states that an object a can not be more similar to any other object b but to itself (except if $a = b$).

3. D-AHC : Dissimilarity based AHC

In this section, we review the basic concepts of AHC. Firstly, we introduce the usual LW formula based on dissimilarities. We also provide more detailed explanations about the bottom-up fusion mechanism that builds the dendrogram. Secondly, we review another equivalent way to express the LW formula. This formulation relies on a weighted version of the dissimilarities. In fact, the framework we introduce thereafter, is inspired from this latter expression.

3.1 The LW Formula and the Bottom-up Procedure

For $t = 1$, we initialize $\mathbb{C}^1$ to the set of n singletons with null height values and we set $\mathbf{D}^1 = \mathbf{D}$, the given dissimilarity matrix. Then, at each iteration $t \in \mathbb{T}$, D-AHC merges the[3] couple of clusters (k,l) that satisfies:

$$(k,l) = \underset{(i,j) \in \mathbb{C}^t \times \mathbb{C}^t, i \neq j}{\arg\min} \mathbf{D}_{ij}^t \tag{1}$$

Clusters k and l are fused into (kl) and the dendrogram D is amended with a new node whose height value $H((kl))$ is set to $\mathbf{D}_{kl}^t$. The new partition $\mathbb{C}^{t+1}$ is updated as follows:

$$\mathbb{C}^{t+1} = \mathbb{C}^t \setminus \{k,l\} \cup \{(kl)\} \tag{2}$$

Next, the dissimilarity values between the new cluster (kl) and the other clusters $m \in \mathbb{C}^{t+1}$ have to be computed in order to determine $\mathbf{D}^{t+1}$.

Several schemes were proposed and among the most famous ones we can cite: single linkage, complete linkage, group average (also named UPGMA[4]), Mcquitty (also named WPGMA[5]), centroid (also named UPGMC[6]), median (also named WPGMC[7]) and Ward. The first four techniques are known as graph methods whereas the three latter ones are named geometric methods.

Despite these numerous dissimilarity measures, the LW equation introduced in (Lance and Williams, 1967), is a parametric updating formula that generalizes all aforementioned cases. It is defined as follows:

$$\begin{aligned} \mathbf{D}_{(kl)m}^{t+1} = \alpha'(k,l,m)\mathbf{D}_{km}^t + \alpha'(l,k,m)\mathbf{D}_{lm}^t + \beta'(k,l,m)\mathbf{D}_{kl}^t + \gamma'|\mathbf{D}_{km}^t - \mathbf{D}_{lm}^t|, \\ \forall t \in \mathbb{T}, \forall m \in \mathbb{C}^{t+1}, m \neq (kl) \end{aligned} \tag{3}$$

where γ' is a scalar and α', β' are functions from the set of triples of disjoint subsets of $\mathbb{O}$ to $\mathbb{R}$.

In Table 1, we review the particular definitions of α', β' and γ' for the methods cited above. In this table, observe that:

- For all schemes, β' is symmetric in its two first arguments unlike α'.

3. Note that there might be several couples of clusters as solutions to (1) which could result in different dendrograms.
4. Unweighted Pair Group Method with Arithmetic mean
5. Weighted Pair Group Method with Arithmetic mean
6. Unweighted Pair Group Method Centroid
7. Weighted Pair Group Method Centroid

Method	$\alpha'(k,l,m)$	$\beta'(k,l,m)$	γ'
Single link.	$1/2$	0	$-1/2$
Complete link.	$1/2$	0	$1/2$
Group aver.	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	0	0
Mcquitty	$1/2$	0	0
Centroid	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	$-\frac{\lvert k\rvert\lvert l\rvert}{(\lvert k\rvert+\lvert l\rvert)^2}$	0
Median	$1/2$	$-1/4$	0
Ward	$\frac{\lvert k\rvert+\lvert m\rvert}{\lvert k\rvert+\lvert l\rvert+\lvert m\rvert}$	$-\frac{\lvert m\rvert}{\lvert k\rvert+\lvert l\rvert+\lvert m\rvert}$	0

Table 1: Particular settings of the LW formula (4).

- Except the Ward method, α' and β' do not depend on a third argument.
- Whatever the triple (k,l,m), α' is constant for single linkage, complete linkage, Mcquitty and median.
- Likewise, β' is constant for single linkage, complete linkage, group average, Mcquitty and median.
- Concerning γ', it is non-null only for single linkage and complete linkage.

In the rest of the paper, we only consider the sub-family of LW clusterings that satisfies $\gamma = 0$. This rules out the single and complete linkage techniques. In fact, these two latter schemes are peculiar since they reduce to the min and max operators respectively. Due to their specific features, single and complete linkages can be addressed using special algorithms (Gower and Ross, 1969; Sibson, 1973; Defays, 1977).

Consequently, we are interested in the following LW sub-formula in what follows:

$$\mathbf{D}^{t+1}_{(kl)m} = \alpha'(k,l,m)\mathbf{D}^t_{km} + \alpha'(l,k,m)\mathbf{D}^t_{lm} + \beta'(k,l,m)\mathbf{D}^t_{kl}, \quad \forall t \in \mathbb{T}, \forall m \in \mathbb{C}^{t+1}, m \neq (kl) \tag{4}$$

To wrap up this sub-section, we provide in Algorithm 1 the pseudo-code of D-AHC using the previous LW sub-formula.

3.2 An Equivalent Dissimilarity Based LW Sub-formula

Henceforth, we suppose that any object $a \in \mathbb{O}$ can be represented as a vector $\mathbf{x}^a$ in an Hilbert space $\mathcal{H}$. Moreover, we assume that dissimilarities are given by squared Euclidean distances. Thus, the general term of $\mathbf{D}$ is:

$$\mathbf{D}_{ab} = \|\mathbf{x}^a - \mathbf{x}^b\|^2, \quad \forall a, b \in \mathbb{O} \tag{5}$$

In this context, we review another dissimilarity based LW sub-formula which is equivalent to (4) and Table 1. Indeed, if objects are vectors in an Hilbert space then, the centroid

Algorithm 1: General procedure of D-AHC.

Input: $\mathbf{D}$ a dissimilarity matrix, an AHC method
Output: D a dendrogram
Initialize D with n leaves;
Set $\mathbf{D}^1 = \mathbf{D}$;
for $t = 1, \ldots, n-1$ **do**
  Find the pair of clusters (k,l) according to (1);
  Merge (k,l) into (kl) and update D;
  Compute $\mathbf{D}^{t+1}$ by applying (4) with the corresponding AHC method parameters values given in Table 1.
end

and Ward update equations can be expressed in terms of cluster representatives (see for e.g. (Murtagh and Contreras, 2012; Müllner, 2011)). Let $\mathbf{g}^i$ be the mean vector of cluster i:

$$\mathbf{g}^i = \frac{1}{|i|}\sum_{a \in i} \mathbf{x}^a, \quad i \in 2^{\mathbb{O}} \tag{6}$$

Then, for two clusters $i, j \in \mathbb{C}^t$, the dissimilarity used by the centroid scheme is:

$$\mathbf{D}^t_{ij} = \|\mathbf{g}^i - \mathbf{g}^j\|^2, \quad \forall t \in \mathbb{T} \tag{7}$$

Regarding the Ward approach, it is in fact, a weighted version of centroid since we have for the former scheme:

$$\mathbf{D}^t_{ij} = \frac{|i||j|}{|i|+|j|}\|\mathbf{g}^i - \mathbf{g}^j\|^2, \quad \forall t \in \mathbb{T}, \forall i, j \in \mathbb{C}^t \tag{8}$$

The following D-AHC iterative procedure is equivalent to Algorithm 1 (see for e.g. (Murtagh and Contreras, 2012)). For $t = 1$, let $\mathbf{D}^1$ be the input dissimilarity matrix $\mathbf{D}$ of squared Euclidean distances between data points. At each iteration, the pair (k,l) which gives the minimum *weighted dissimilarity* is merged:

$$(k,l) = \underset{(i,j) \in \mathbb{C}^t \times \mathbb{C}^t, i \neq j}{\arg\min} p(i,j)\mathbf{D}^t_{ij} \tag{9}$$

where p is a function from $\{(i,j) \in 2^{\mathbb{O}} \times 2^{\mathbb{O}}, i \cap j = \emptyset\}$, the set of pairs of disjoint subsets of $\mathbb{O}$, to $\mathbb{R}$. The definition of p for each dissimilarity scheme is provided in Table 2.

After each merge the dissimilarity matrix is updated as follows:

$$\mathbf{D}^{t+1}_{(kl)m} = \alpha(k,l)\mathbf{D}^t_{km} + \alpha(l,k)\mathbf{D}^t_{lm} + \beta(k,l)\mathbf{D}^t_{kl}, \quad \forall t \in \mathbb{T}, \forall m \in \mathbb{C}^{t+1}, m \neq (kl) \tag{10}$$

where, in this modeling, α and β are set functions whose definitions are also given in Table 2.

It is important to mention that, in the case of Ward, the set functions α and β do not depend on the third argument unlike α' and β'. Consequently, we can globally consider α and β as two-place set functions which only depend on the two clusters being fused at each iteration. More formally, α and β are functions from $\{(i,j) \in 2^{\mathbb{O}} \times 2^{\mathbb{O}}, i \cap j = \emptyset\}$ to $\mathbb{R}$

Method	$\alpha(k,l)$	$\beta(k,l)$	$p(i,j)$
Group aver.	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	0	1
Mcquitty	1/2	0	1
Centroid	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	$-\frac{\lvert k\rvert\lvert l\rvert}{(\lvert k\rvert+\lvert l\rvert)^2}$	1
Median	1/2	−1/4	1
Ward	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	$-\frac{\lvert k\rvert\lvert l\rvert}{(\lvert k\rvert+\lvert l\rvert)^2}$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$
W-Median	1/2	−1/4	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$

Table 2: Particular settings of the LW sub-formula (10) in the model defined by (9).

As mentioned previously and by observing (7) and (8), Ward can be interpreted as a weighted version of centroid. Similarly, we introduce a *weighted version of median (w-median)*. The parameters of this new method are defined in the last row of Table 2. W-median set functions α and β are the same as median. It is the set function p which is different: instead of a uniform weight, w-median uses the same weight function as Ward. As we shall demonstrate later on, the w-median method provides monotonic dendrograms unlike the median technique.

4. K-AHC: Kernel Matrix based AHC

We have supposed that the items are represented in an Hilbert space $\mathcal{H}$ and so far, squared Euclidean distances have been used to represent the proximity relationships between points. Henceforth, we also use the underlying inner-product of $\mathcal{H}$.

In this section, we start by introducing a model that generalizes the LW sub-equation (10). Our framework relies on inner-products and it amounts to a *Kernel matrix based AHC (K-AHC)*. Thereafter, we introduce several properties of our model. We provide sufficient conditions for a technique expressed in our modeling to provide monotonic dendrograms. Furthermore, we design a stored data matrix approach that generalizes to group average and Mcquitty schemes.

4.1 Inner-product Based LW Sub-formula

Let $\langle .,. \rangle$ denotes the inner-product of $\mathcal{H}$. The geometrical data representation we assume in this work is formally stated as follows:

$$\left\{ \begin{array}{l} \mathbf{S}_{ab} = \langle \mathbf{x}^a, \mathbf{x}^b \rangle \\ \mathbf{D}_{ab} = \mathbf{S}_{aa} + \mathbf{S}_{bb} - 2\mathbf{S}_{ab} \end{array} \right. , \quad \forall a,b \in \mathbb{O} \tag{C1}$$

This implies that the matrix $\mathbf{S}$ is a *kernel (or Gram) matrix* and satisfies:

$$\left\{ \begin{array}{ll} \mathbf{S}_{ab} = \mathbf{S}_{ba}, \quad \forall a,b \in \mathbb{O} & \text{(symmetry)} \\ \mathbf{S} \succeq 0 & \text{(positive semi-definite)} \end{array} \right. \tag{11}$$

This data representation encompasses Reproducing Kernel Hilbert Spaces (RKHS). Accordingly, our approach is a kernel method (see for e.g. (Scholkopf and Smola, 2001)) which

can benefit from a large spectrum of kernel functions in order to address diverse manifold learning problems, that is K-AHC is able to detect groups of items with arbitrary shapes.

In contrast to the LW sub-formula, our model requires two equations to update the $\mathbf{S}$ matrix: one for the *off-diagonal* elements and one for the *on-diagonal* entries.

Suppose that, for $t=1$, $\mathbf{S}$ is the input matrix of our procedure: $\mathbf{S}^1 = \mathbf{S}$. Assume that, at iteration $t \in \mathbb{T}$, clusters k and l are merged together. In our model, $\mathbf{S}^{t+1}$ is updated according to the two following recurrence equations:

$$\mathbf{S}^{t+1}_{(kl)m} = \mathfrak{a}(k,l)\mathbf{S}^t_{km} + \mathfrak{a}(l,k)\mathbf{S}^t_{lm}, \quad \forall t \in \mathbb{T}, \forall m \in \mathbb{C}^{t+1}, m \neq (kl) \tag{12a}$$

$$\mathbf{S}^{t+1}_{(kl)(kl)} = \mathfrak{b}(k,l)\mathbf{S}^t_{kl} + \mathfrak{c}(k,l)\mathbf{S}^t_{kk} + \mathfrak{c}(l,k)\mathbf{S}^t_{ll}, \quad \forall t \in \mathbb{T} \tag{12b}$$

where $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$ are functions from $\{(i,j) \in 2^{\mathbb{O}} \times 2^{\mathbb{O}}, i \cap j = \emptyset\}$ to $\mathbb{R}$.

Similarly to D-AHC, we assume that the updated matrix is symmetric all along the procedure and thus, $\mathbf{S}^t_{m(kl)} = \mathbf{S}^t_{(kl)m}, \forall t \in \mathbb{T}, \forall m \in \mathbb{C}^{t+1}$.

For all $t \in \mathbb{T}$, let $\mathbf{\Lambda}^t$ be the square matrix of order $n-1+t$ with general term:

$$\mathbf{\Lambda}^t_{ij} = \mathbf{S}^t_{ij} - \frac{1}{2}(\mathbf{S}^t_{ii} + \mathbf{S}^t_{jj}), \quad \forall t \in \mathbb{T}, \forall i,j \in \mathbb{C}^t \tag{13}$$

$\mathbf{\Lambda}^t$ compares each couple of clusters in $\mathbb{C}^t$ and plays a core role in our approach. The following result establishes the connection between the LW sub-equation (10) on the one hand, and our approach (13), (12a), (12b) on the other hand.

Lemma 7 *Let $\{\mathbf{D}^t\}_{t\in\mathbb{T}}$ and $\{\mathbf{\Lambda}^t\}_{t\in\mathbb{T}}$ be the sequences of square matrices with input elements $\mathbf{D}$ and $\mathbf{S}$ and subsequent elements defined by* (10) *and* (13), (12a),(12b), *respectively. Suppose that the related set functions α, β on the one hand and $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$ on the other hand, are such that:*

$$\begin{array}{l} \mathfrak{a} = \alpha \\ \mathfrak{b} = -2\beta \\ \mathfrak{c} = \alpha + \beta \end{array}$$

Then, under (C1) *and if $\mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1, \forall k,l \in 2^{\mathbb{O}}$, it holds:*

$$\mathbf{\Lambda}^t_{ij} = -\frac{1}{2}\mathbf{D}^t_{ij}, \quad \forall t \in \mathbb{T}, \forall i,j \in \mathbb{C}^t, i \neq j$$

Proof Under (C1), it is clear that for $t=1$:

$$\mathbf{\Lambda}^1_{ab} = \mathbf{S}^1_{ab} - \frac{1}{2}(\mathbf{S}^1_{aa} + \mathbf{S}^1_{bb}) = -\frac{1}{2}\mathbf{D}^1_{ab}, \quad \forall a,b \in \mathbb{O}$$

We assume that the property is true for t: $\mathbf{\Lambda}^t_{ij} = \mathbf{S}^t_{ij} - \frac{1}{2}(\mathbf{S}^t_{ii} + \mathbf{S}^t_{jj}) = -\frac{1}{2}\mathbf{D}^t_{ij}, \forall i,j \in \mathbb{C}^t, i \neq j$. Then, let us prove that it is true for $t+1$ as well. Let k and l be the two clusters that are fused at iteration t. We replace in (12a) and (12b) the set functions $\mathfrak{a}$, $\mathfrak{b}$ and $\mathfrak{c}$ with α, -2β and $\alpha+\beta$ respectively. It comes:

$$\mathbf{S}^t_{(kl)m} = \alpha(k,l)\mathbf{S}^t_{km} + \alpha(l,k)\mathbf{S}^t_{lm}$$

$$\mathbf{S}^t_{(kl)(kl)} = -2\beta(k,l)\mathbf{S}^t_{kl} + (\alpha(k,l) + \beta(k,l))\mathbf{S}^t_{kk} + (\alpha(l,k) + \beta(l,k))\mathbf{S}^t_{ll}$$

Method	$\mathfrak{a}(k,l)$	$\mathfrak{b}(k,l)$	$\mathfrak{c}(k,l)$	$\mathfrak{p}(i,j)$
Group average	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	0	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	1
Mcquitty	1/2	0	1/2	1
Centroid	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	$\frac{2\lvert k\rvert\lvert l\rvert}{(\lvert k\rvert+\lvert l\rvert)^2}$	$\frac{\lvert k\rvert^2}{(\lvert k\rvert+\lvert l\rvert)^2}$	1
Median	1/2	1/2	1/4	1
Ward	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}$	$\frac{2\lvert k\rvert\lvert l\rvert}{(\lvert k\rvert+\lvert l\rvert)^2}$	$\frac{\lvert k\rvert^2}{(\lvert k\rvert+\lvert l\rvert)^2}$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$
W-Median	1/2	1/2	1/4	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$

Table 3: Particular settings in our model defined by (12a), (12b) and (14).

Next, assuming $\alpha(k,l)+\alpha(l,k)=1$, $\forall k,l\in\mathbb{C}^t$, we have:

$$\mathbf{S}^t_{mm}=(\alpha(k,l)+\alpha(l,k))\mathbf{S}^t_{mm}$$

If we put all these ingredients into (13) for $t+1$, $i=(kl)$ and $j=m$; regroup terms with respect to α and β; replace $\mathbf{S}^t_{ii}+\mathbf{S}^t_{jj}-2\mathbf{S}^t_{ij}$ with $\mathbf{D}^t_{ij}$, for all $i,j\in\{k,l,m\}, i\neq j$; then we have, $\forall m\in\mathbb{C}^{t+1}$:

$$\begin{aligned}\mathbf{\Lambda}^{t+1}_{(kl)m}&=-\frac{1}{2}\left(\alpha(k,l)\mathbf{D}^t_{km}+\alpha(l,k)\mathbf{D}^t_{lm}+\beta(k,l)\mathbf{D}^t_{kl}\right)\\&=-\frac{1}{2}\mathbf{D}^{t+1}_{(kl)m}\end{aligned}$$

■

Next, we introduce our approach which also proceeds in a bottom-up manner. However, unlike D-AHC, K-AHC performs a *maximum search* at each iteration. Indeed, after having initialized a dendrogram D with n leaves, for each $t\in\mathbb{T}$, K-AHC fuses the pair of clusters (k,l) that satisfies:

$$(k,l)=\underset{(i,j)\in\mathbb{C}^t\times\mathbb{C}^t,i\neq j}{\arg\max}\ \mathfrak{p}(i,j)\mathbf{\Lambda}^t_{ij} \tag{14}$$

where $\mathfrak{p}$ is also a real-valued function whose domain is the set of pairs of disjoint subsets of $\mathbb{O}$.

At iteration t, clusters k and l are merged into (kl). The latter subset is represented by a new node in D and its "height" value is $H((kl))=\mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl}$. $\mathbb{C}^{t+1}$ is updated similarly to (2) and $\mathbf{S}^{t+1}$ is determined from $\mathbf{S}^t$ by applying (12a) and (12b).

In order to clearly state in our model the schemes under study, we provide in Table 3 the definitions of their respective set functions $\mathfrak{a}$, $\mathfrak{b}$, $\mathfrak{c}$ and $\mathfrak{p}$. Furthermore, we summarize in Algorithm 2 the K-AHC procedure.

From Lemma 7 and assuming $\mathfrak{p}=p$, it is clear that Algorithm 1 and Algorithm 2 merge the same[8] couple of clusters at each iteration. Therefore, they have equivalent dendrograms (see Definition 5). The only difference is that the dendrogram provided by K-AHC assigns

8. Assuming that if there are several (but same) solutions to (9) and (14), both algorithms pick the same pair.

Algorithm 2: General procedure of K-AHC.

Input: $\mathbf{S}$ a kernel matrix, an AHC method
Output: D a dendrogram
Initialize D with n leaves;
Set $\mathbf{S}^1=\mathbf{S}$;
for $t=1,\ldots,n-1$ **do**
  Find the pair of clusters (k,l) according to (14) with the corresponding AHC method parameters values given in Table 3 ;
  Merge (k,l) into (kl) and update D;
  Compute $\mathbf{S}^{t+1}$ by applying (12a) and (12b) with the corresponding AHC method parameters values given in Table 3.
end

to each node a "height" value which equals minus one half times the height value assigned to the same node of the dendrogram obtained by D-AHC. As a consequence, we have the following result.

Theorem 8 *Suppose that the conditions in Lemma 7 are satisfied. Suppose in addition, that p in (9) and $\mathfrak{p}$ in (14) are the same. Then, Algorithm 1 and Algorithm 2 provide equivalent dendrograms.*

Note that, since for all techniques listed in Table 3, $\mathfrak{a}(k,l)+\mathfrak{a}(l,k)=1, \forall k,l\in 2^{\mathbb{O}}$ and $p=\mathfrak{p}$ then, for all particular schemes we examine, K-AHC is equivalent to D-AHC.

4.2 Monotonicity

In hierarchical clustering, it is important to know whether a method can provide reversals while building the dendrogram. Indeed, in practice, non-monotonic dendrograms can be difficult to interpret and are thus undesirable.

In the classic AHC framework described by Algorithm 1, a technique provides a monotonic dendrogram if and only if the following condition holds:

$$\mathbf{D}^{t+1}_{(kl)m}\geq\mathbf{D}^t_{kl},\quad\forall t\in\mathbb{T},\forall k,l,m\in\mathbb{C}^t \tag{15}$$

Milligan (1979), provided sufficient conditions for a method expressed in the usual LW equation (4), to output a monotonic dendrogram. It has to satisfy the following relationships:

$$\begin{cases}\alpha'(k,l,m),\alpha'(l,k,m)\geq 0\\ \alpha'(k,l,m)+\alpha'(l,k,m)+\beta'(k,l,m)\geq 1\\ (\gamma'\geq 0)\vee(\gamma'\leq 0\wedge|\gamma'|\geq\alpha'(k,l,m),\alpha'(l,k,m))\end{cases},\quad\forall k,l,m\in 2^{\mathbb{O}} \tag{16}$$

In our approach described in Algorithm 2, the "height" value is rather a *depth value* since it varies in the opposite way than in Algorithm 1. Consequently, the monotonicity definition given previously translates as follows in the K-AHC case:

$$\mathfrak{p}((kl),m)\mathbf{\Lambda}^{t+1}_{(kl)m}\leq\mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl},\quad\forall t\in\mathbb{T},\forall k,l,m\in\mathbb{C}^t \tag{17}$$

We give below sufficient conditions for a method expressed in our model to give monotonic dendrograms.

Proposition 9 *Let $\{\mathbf{\Lambda}^t\}_{t\in\mathbb{T}}$ be the sequence of square matrices with input element $\mathbf{S}$ and subsequent elements defined by* (13), (12a), (12b). *Suppose that the set functions $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ and $\mathfrak{p}$ satisfy:*

$$\begin{cases} \mathfrak{a}(k,l), \mathfrak{b}(k,l), \mathfrak{c}(k,l), \mathfrak{p}(k,l) \geq 0 \\ \mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1 \\ \mathfrak{b}(k,l) - \mathfrak{b}(l,k) = 0 \\ \mathfrak{c}(k,l) - \mathfrak{a}(k,l) + \frac{1}{2}\mathfrak{b}(k,l) = 0 \\ \frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)} \geq 0 \\ \mathfrak{p}((kl),m)\left(\frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)}\right) \geq 1 \end{cases}, \quad \forall k,l,m \in 2^{\mathbb{O}}$$

Then, under (C1), $\{\mathfrak{p}\mathbf{\Lambda}^t\}_{t\in\mathbb{T}}$ *satisfies* (17).

Proof From the definition of $\mathbf{\Lambda}^t$ given in (13), it comes:

$$\mathbf{\Lambda}^{t+1}_{(kl)m} = \mathfrak{a}(k,l)\mathbf{S}^t_{km} + \mathfrak{a}(l,k)\mathbf{S}^t_{lm} - \frac{1}{2}(\mathfrak{b}(k,l)\mathbf{S}^t_{kl} + \mathfrak{c}(k,l)\mathbf{S}^t_{kk} + \mathfrak{c}(l,k)\mathbf{S}^t_{ll} + \mathbf{S}^t_{mm})$$

By using $\mathfrak{c}(k,l) = \mathfrak{a}(k,l) - \frac{1}{2}\mathfrak{b}(k,l)$ and $\mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1$, we obtain:

$$\mathbf{\Lambda}^{t+1}_{(kl)m} = \mathfrak{a}(k,l)\mathbf{S}^t_{km} + \mathfrak{a}(l,k)\mathbf{S}^t_{lm} - \frac{1}{2}\left(\mathfrak{b}(k,l)\mathbf{S}^t_{kl} + (\mathfrak{a}(k,l) - \frac{1}{2}\mathfrak{b}(k,l))\mathbf{S}^t_{kk} + (\mathfrak{a}(l,k) - \frac{1}{2}\mathfrak{b}(l,k))\mathbf{S}^t_{ll} + (\mathfrak{a}(k,l) + \mathfrak{a}(l,k))\mathbf{S}^t_{mm}\right)$$

Then, by assuming $\mathfrak{b}(k,l) - \mathfrak{b}(l,k) = 0$ and by regrouping terms with respect to $\mathfrak{a}$ and $\mathfrak{b}$, we get:

$$\mathbf{\Lambda}^{t+1}_{(kl)m} = \mathfrak{a}(k,l)\mathbf{\Lambda}^t_{km} + \mathfrak{a}(l,k)\mathbf{\Lambda}^t_{lm} - \frac{1}{2}\mathfrak{b}(k,l)\mathbf{\Lambda}^t_{kl}$$

Next, since k and l are the clusters that have been merged at iteration t, according to (14), we have $\mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl} \geq \mathfrak{p}(k,m)\mathbf{\Lambda}^t_{km}, \mathfrak{p}(l,m)\mathbf{\Lambda}^t_{lm}$. Therefore, it holds:

$$\begin{aligned} \mathbf{\Lambda}^{t+1}_{(kl)m} &\leq \mathfrak{a}(k,l)\frac{\mathfrak{p}(k,l)}{\mathfrak{p}(k,m)}\mathbf{\Lambda}^t_{kl} + \mathfrak{a}(l,k)\frac{\mathfrak{p}(k,l)}{\mathfrak{p}(l,m)}\mathbf{\Lambda}^t_{kl} - \frac{1}{2}\mathfrak{b}(k,l)\mathbf{\Lambda}^t_{kl} \\ &\leq \left(\frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)}\right)\mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl} \end{aligned}$$

Under (C1), it is easy to see that $\mathbf{\Lambda}^1_{ab} \leq 0, \forall a,b \in \mathbb{O}$. Then, by assuming that $\mathfrak{p}(k,l) \geq 0$ and $\frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)} \geq 0$, $\forall k,l,m \in 2^{\mathbb{O}}$, it is easy to prove by induction that $\mathbf{\Lambda}^t_{ij} \leq 0, \forall t \in \mathbb{T}, \forall i,j \in \mathbb{C}^t$, using the inequality above.

Besides, by multiplying the same previous inequality by $\mathfrak{p}((kl),m)$, we obtain an upper bound for $\mathfrak{p}((kl),m)\mathbf{\Lambda}^{t+1}_{(kl)m}$:

$$\mathfrak{p}((kl),m)\mathbf{\Lambda}^{t+1}_{(kl)m} \leq \mathfrak{p}((kl),m)\left(\frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)}\right)\mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl}$$

Finally, since $\mathbf{\Lambda}^t_{kl}, \mathbf{\Lambda}^{t+1}_{(kl)m} \leq 0$ as stated previously, in order to have $\mathfrak{p}((kl),m)\mathbf{\Lambda}^{t+1}_{(kl)m} \leq \mathfrak{p}(k,l)\mathbf{\Lambda}^t_{kl}$, it is sufficient that:

$$\mathfrak{p}((kl),m)\left(\frac{\mathfrak{a}(k,l)}{\mathfrak{p}(k,m)} + \frac{\mathfrak{a}(l,k)}{\mathfrak{p}(l,m)} - \frac{\mathfrak{b}(k,l)}{2\mathfrak{p}(k,l)}\right) \geq 1$$

■

It is easy to check that all six studied techniques described in Table 3 satisfy the first fifth conditions in Proposition 9. However, unlike group average, Mcquitty and Ward, the methods centroid and median do not satisfy the last condition. These three former schemes are known to be monotonic. Regarding w-median, the new technique we introduced at the end of sub-section 3.2, we have the following property.

Proposition 10 *The w-median scheme is monotonic.*

Proof If we apply the w-median parameters values given in Table 3 to the last condition of Proposition 9, the left-hand side of the inequality reads:

$$\frac{(|k|+|l|)|m|}{|k|+|l|+|m|}\left(\frac{|k|+|m|}{2|k||m|} + \frac{|l|+|m|}{2|l||m|} - \frac{|k|+|l|}{4|k||l|}\right)$$

By developing this equation and after some manipulations, we obtain the following equivalent expression:

$$1 + \frac{|m|}{4(|k|+|l|+|m|)}\left(\frac{|l|}{|k|} + \frac{|k|}{|l|} - 2\right)$$

Let $r = \frac{\max(|k|,|l|)}{\min(|k|,|l|)}$ being a non-negative rational number. The term in parenthesis becomes $\left(r + \frac{1}{r} - 2\right)$. It is easy to see that $r + \frac{1}{r} \geq 2$ and thus $\left(\frac{|l|}{|k|} + \frac{|k|}{|l|} - 2\right) \geq 0$, which completes the proof. ■

4.3 Stored Data Matrix Approach Based on K-AHC

Let q be the dimension of the Hilbert space $\mathcal{H}$. Suppose that q is finite and let $\mathbf{X}$ be the data matrix of size $n \times q$ where each row represents a vector.

So far, we have assumed a stored proximity matrix approach where the input of the algorithms is either $\mathbf{D}$ or $\mathbf{S}$ which are both of size $O(n^2)$. However, it can be useful to operate on the data matrix $\mathbf{X}$ instead of $\mathbf{D}$ or $\mathbf{S}$. Indeed, if n is too large, it could be inefficient, or even impossible, to store the whole proximity matrix on a single machine.

If n is very large but q is much lower than n then, the *stored data matrix* approach can be carried out. It takes as input the data matrix $\mathbf{X}$, computes the dissimilarities between vectors on the fly, finds the pair of clusters to merge, represents the new cluster by a representative vector in $\mathcal{H}$ and repeat these latter steps for $t \in \mathbb{T}$. Note that such an approach allows alleviating the storage complexity but not the computational one, since, in the worst case, the proximities between all pairs of objects need to be evaluated.

In order to carry out the stored data approach, any dissimilarity scheme needs to be formulated in terms of representative vectors in $\mathcal{H}$ (see for e.g. (Murtagh and Contreras, 2012)).

We already pointed out, in sub-section 3.2, that the centroid and Ward methods can be performed by using mean vectors. Note that we can determine the latter representative vectors in an iterative fashion. For $t = 1$, we set $\mathbf{g}^a = \mathbf{x}^a, \forall a \in \mathbb{O}$. For $t > 1$, if k and l are the clusters that are fused then, the mean vector $\mathbf{g}^{(kl)}$ can be computed as follows:

$$\mathbf{g}^{(kl)} = \frac{|k|}{|k|+|l|}\mathbf{g}^k + \frac{|l|}{|k|+|l|}\mathbf{g}^l \tag{18}$$

The median and w-median schemes can also be carried out in a similar way. In these cases, clusters are represented by mid-points. Let $\mathbf{g}^i$ denote the representative vector of cluster $i \in 2^{\mathbb{O}}$. For $t = 1$, all objects are considered as singletons and we set again $\mathbf{g}^a = \mathbf{x}^a, \forall a \in \mathbb{O}$. Then, for $t > 1$, the newly formed cluster (kl) is given as follows, for median and w-median:

$$\mathbf{g}^{(kl)} = \frac{1}{2}\mathbf{g}^k + \frac{1}{2}\mathbf{g}^l \tag{19}$$

Then, for the median and w-median methods, the dissimilarity values satisfy:

$$\mathbf{D}^t_{ij} = \|\mathbf{g}^i - \mathbf{g}^j\|^2, \quad \forall t \in \mathbb{T}, \forall i, j \in \mathbb{C}^t, i \neq j \tag{20}$$

For the graph methods group average and Mcquitty, there is no such equivalent way to express their dissimilarity update equation using representative points using the usual LW sub-equation 4. Yet, our approach makes it possible to obtain such a property for these two latter schemes.

In order to introduce this property, let us first discuss the stored data matrix approach for geometric schemes in our inner-product based modeling. We have the following results.

Proposition 11 *Let $\mathbf{g}^i = \frac{1}{|i|}\sum_{a\in i}\mathbf{x}^a, \forall i \in 2^{\mathbb{O}}$. Then, for the centroid and Ward schemes, it holds:*

$$\mathbf{S}^t_{ij} = \langle \mathbf{g}^i, \mathbf{g}^j \rangle, \quad \forall t \in \mathbb{T}, \forall i, j \in \mathbb{C}^t$$

Proof Since $\mathbf{S}$ is a kernel matrix, $\mathbf{S}^1_{ab} = \langle \mathbf{g}^a, \mathbf{g}^b \rangle, \forall a, b \in \mathbb{C}^1$. Assume that $\mathbf{S}^t_{ij} = \langle \mathbf{g}^i, \mathbf{g}^j \rangle, \forall i, j \in \mathbb{C}^t$ holds for t and let us prove that it holds for $t + 1$ as well. The proof simply uses the linear property of the inner-product. Concerning $\mathbf{S}^{t+1}_{(kl)m}$, we have:

$$\begin{aligned}
\mathbf{S}^{t+1}_{(kl)m} &= \mathfrak{a}(k,l)\mathbf{S}^t_{km} + \mathfrak{a}(l,k)\mathbf{S}^t_{lm} \\
&= \frac{|k|}{|k|+|l|}\langle \mathbf{g}^k, \mathbf{g}^m \rangle + \frac{|l|}{|k|+|l|}\langle \mathbf{g}^l, \mathbf{g}^m \rangle \\
&= \frac{|k|}{|k|+|l|}\langle \frac{1}{|k|}\sum_{a\in k}\mathbf{x}^a, \mathbf{g}^m \rangle + \frac{|l|}{|k|+|l|}\langle \frac{1}{|l|}\sum_{b\in l}\mathbf{x}^b, \mathbf{g}^m \rangle \\
&= \langle \frac{1}{|k|+|l|}(\sum_{a\in k}\mathbf{x}^a + \sum_{b\in l}\mathbf{x}^b), \mathbf{g}^m \rangle \\
&= \langle \mathbf{g}^{(kl)}, \mathbf{g}^m \rangle
\end{aligned}$$

Regarding $\mathbf{S}^{t+1}_{(kl)(kl)}$, we have:

$$\begin{aligned}
\mathbf{S}^{t+1}_{(kl)(kl)} &= \mathfrak{b}(k,l)\mathbf{S}^t_{kl} + \mathfrak{c}(k,l)\mathbf{S}^t_{kk} + \mathfrak{c}(l,k)\mathbf{S}^t_{ll} \\
&= \frac{2|k||l|}{(|k|+|l|)^2}\langle \mathbf{g}^k, \mathbf{g}^l \rangle + \frac{|k|^2}{(|k|+|l|)^2}\langle \mathbf{g}^k, \mathbf{g}^k \rangle \\
&+ \frac{|l|^2}{(|k|+|l|)^2}\langle \mathbf{g}^l, \mathbf{g}^l \rangle \\
&= \langle \frac{1}{|k|+|l|}(\sum_{a\in k}\mathbf{x}^a + \sum_{b\in l}\mathbf{x}^b), \frac{1}{|k|+|l|}(\sum_{a\in k}\mathbf{x}^a + \sum_{b\in l}\mathbf{x}^b) \rangle \\
&= \langle \mathbf{g}^{(kl)}, \mathbf{g}^{(kl)} \rangle
\end{aligned}$$

■

Proposition 12 *Let $\mathbf{g}^a = \mathbf{x}^a, \forall a \in \mathbb{O}$ and $\forall t \in \mathbb{T}$, if k and l are merged, let $\mathbf{g}^{(kl)}$ be defined by* (19)*. Then, for the median and w-median schemes, it holds:*

$$\mathbf{S}^t_{ij} = \langle \mathbf{g}^i, \mathbf{g}^j \rangle, \quad \forall t \in \mathbb{T}, \forall i, j \in \mathbb{C}^t$$

Proof The proof is similar than for Proposition 11. ■

Propositions 11 and 12 states that for centroid, Ward, median and w-median, both off-diagonal and on-diagonal entries of $\mathbf{S}^t$ can be determined by inner-products of representative vectors. As far as group average and Mcquitty techniques are concerned, this latter property is only valid for off-diagonal elements of $\mathbf{S}^t$. Indeed, in Table 3, it is clear that group average and Mcquitty respectively have the same weights vectors $\{\mathfrak{a}(k,l), \mathfrak{a}(l,k)\}$ than centroid (or Ward) and median (or w-median). Regarding the on-diagonal entries, we can actually compute these values for group average and Mcquitty efficiently, providing that we initially store, in an extra vector, the squared norm of each element vector. Let $\mathbf{s}$ be the vector of size n with general term:

$$\mathbf{s}_a = \langle \mathbf{x}^a, \mathbf{x}^a \rangle, \quad \forall a \in \mathbb{O} \tag{21}$$

Let $\mathbf{s}^1 = \mathbf{s}$ and for $t = 2, \ldots, n-1$, let $\mathbf{s}^t$ be a vector of size $n - t + 1$ whose component $\mathbf{s}^t_i$ is associated to cluster $i \in \mathbb{C}^t$. At each iteration $t \in \mathbb{T}$, suppose that k and l are the clusters that are merged then, $\mathbf{s}^{t+1}_{(kl)}$ is determined[9] using the following recurrence formula:

$$\mathbf{s}^{t+1}_{(kl)} = \mathfrak{c}(k,l)\mathbf{s}^t_k + \mathfrak{c}(l,k)\mathbf{s}^t_l, \quad \forall t \in \mathbb{T} \tag{22}$$

where $\mathfrak{c}$ is the set function defined for group average and Mcquitty in Table 3.

Then, it is easy to check the following property.

Proposition 13 *Let $\mathbf{s}^1_a = \langle \mathbf{x}^a, \mathbf{x}^a \rangle, \forall a \in \mathbb{O}$ and $\forall t \in \mathbb{T}$, if k and l are merged, let $\mathbf{s}^t_{(kl)}$ be defined by* (22)*. Then, for the group average and Mcquitty schemes, it holds:*

$$\mathbf{S}^t_{ii} = \mathbf{s}^t_i, \quad \forall t \in \mathbb{T}, \forall i \in \mathbb{C}^t$$

All previously discussed results are summarized in Table 4 and Algorithm 3 which provide a K-AHC based stored data matrix approach for all six methods we examine.

9. Note that similarly to $\mathbf{D}^t$ or $\mathbf{S}^t$, $\mathbf{s}^t$ loses one dimension at each iteration.

Method	$\mathbf{S}^t_{ij}, i \neq j$	$\mathbf{S}^t_{ii}$	$\mathfrak{p}(i,j)$	$\mathbf{g}^{(kl)}$	$\mathbf{s}^{t+1}_{(kl)}$
Group average	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\mathbf{s}^t_i$	1	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^k + \frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^l$	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{s}^t_k + \frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{s}^t_l$
Mcquitty	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\mathbf{s}^t_i$	1	$\frac{1}{2}\mathbf{g}^k + \frac{1}{2}\mathbf{g}^l$	$\frac{1}{2}\mathbf{s}^t_k + \frac{1}{2}\mathbf{s}^t_l$
Centroid	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\langle \mathbf{g}^i, \mathbf{g}^i \rangle$	1	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^k + \frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^l$	NA
Median	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\langle \mathbf{g}^i, \mathbf{g}^i \rangle$	1	$\frac{1}{2}\mathbf{g}^k + \frac{1}{2}\mathbf{g}^l$	NA
Ward	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\langle \mathbf{g}^i, \mathbf{g}^i \rangle$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$	$\frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^k + \frac{\lvert k\rvert}{\lvert k\rvert+\lvert l\rvert}\mathbf{g}^l$	NA
W-Median	$\langle \mathbf{g}^i, \mathbf{g}^j \rangle$	$\langle \mathbf{g}^i, \mathbf{g}^i \rangle$	$\frac{\lvert i\rvert\lvert j\rvert}{\lvert i\rvert+\lvert j\rvert}$	$\frac{1}{2}\mathbf{g}^k + \frac{1}{2}\mathbf{g}^l$	NA

Table 4: Particular settings in the stored data matrix based on K-AHC and defined by (14), (13) and representative vectors updates.

Algorithm 3: General procedure of the K-AHC based stored data matrix approach.

Input: $\mathbf{X}$ a data matrix, an AHC method
Output: D a dendrogram

Initialize D with n leaves;
Set $\mathbf{g}^a = \mathbf{x}^a, \forall a \in \mathbb{O}$;
Set $\mathbf{s}_a = \langle \mathbf{x}^a, \mathbf{x}^a \rangle, \forall a \in \mathbb{O}$, if appropriate;
for $t = 1, \ldots, n-1$ **do**
  Compute the inner-product matrix of representative vectors;
  Find the pair of clusters (k,l) according to (14) and (13) with the corresponding AHC method definitions given in Table 4 ;
  Merge (k,l) into (kl) and update D;
  Compute the representative vector $\mathbf{g}^{(kl)}$ by applying the corresponding AHC method formula given in Table 4;
  Compute $\mathbf{s}^{t+1}_{(kl)}$ by applying the corresponding AHC method formula given in Table 4, if appropriate.
end

5. SNK-AHC: Sparsified Normalized Kernel Matrix Based AHC

Another important property of K-AHC is that it offers a way to address the scalability issues of stored proximity matrix based AHC procedures. Our main idea can be stated as follows. Given the distance matrix $\mathbf{D}$, it is reasonable to assume that pairs of items whose distance measure is high are unlikely to be grouped together at an early stage. Consequently, in the goal of reducing the storage complexity, these values could be discarded and we may replace them with zero in order to have a sparse $\mathbf{D}$ matrix. However, this is not sound since a zero distance measure would mean that points are identical while they are far away[10]. In order to avoid this drawback, we propose to use the inner-product matrix $\mathbf{S}$ instead, as we shall explain in what follows.

We now introduce our approach called *Sparsified Normalized Kernel matrix based AHC (SNK-AHC)*. Firstly, we introduce the normalization procedure which transform a kernel matrix $\mathbf{S}$ so that it has a constant diagonal and non-negative values. This preliminary step makes it possible to interpret the inner-product matrix $\mathbf{S}$ in terms of similarities (see Definition 6). Next, we present the sparsification procedure which aims at thresholding $\mathbf{S}$ by setting to zero the lowest values. After this, we introduce the SNK-AHC algorithm and its properties. In particular, we study an interesting feature of group average, Mcquitty and Ward methods and we show in what context, SNK-AHC is able to determine the number of clusters.

5.1 Normalized Kernel Matrix

In our perspective, the term *Normalized Kernel (NK) matrix* designates a kernel matrix with a constant diagonal and non-negative terms. In other words, we assume that the points belong to the intersection between an hypersphere and the positive quadrant of $\mathcal{H}$:

$$\mathbf{S}_{aa} = \mathbf{S}_{bb}, \quad \forall a, b \in \mathbb{O} \tag{C2}$$

$$\mathbf{S}_{ab} \geq 0, \quad \forall a, b \in \mathbb{O} \tag{C3}$$

If the kernel matrix $\mathbf{S}$ does not have a constant diagonal then, we can always apply the cosine normalization (or any generalization proposed in (Ah-Pine, 2010)):

$$\mathbf{S}_{ab} \leftarrow \frac{\mathbf{S}_{ab}}{\sqrt{\mathbf{S}_{aa}\mathbf{S}_{bb}}}, \quad \forall a, b \in \mathbb{O} \tag{23}$$

Next, let v be the minimal value in $\mathbf{S}$. If $v < 0$ then we propose to perform a simple translation in order to obtain non-negative values:

$$\mathbf{S}_{ab} \leftarrow \mathbf{S}_{ab} + |v|, \quad \forall a, b \in \mathbb{O} \tag{24}$$

It is worth noting that such a translation does not change the results of Algorithm 2. In fact, the procedure is invariant under any positive linear transformation of the $\mathbf{S}$ matrix for all six schemes we are interested in.

10. Note that we could have replaced these values with a constant which would have been the maximal distance value but in this case, we would have lost the sparsity property.

Proposition 14 *Suppose that the set functions* $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}$ *satisfy:*

$$\begin{cases} \mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1 \\ \mathfrak{b}(k,l) + \mathfrak{c}(k,l) + \mathfrak{c}(l,k) = 1 \end{cases}, \quad \forall k,l,m \in 2^{\mathbb{O}}$$

Then, Algorithm 2 provides equivalent dendrograms for input similarity matrices $\mathbf{S}$ *and* $\mathbf{T} = u\mathbf{S} + v\mathbf{1}_n$*, with* $u > 0$*,* $v \in \mathbb{R}$ *and* $\mathbf{1}_n$ *being the square matrix of order* n *filled with 1.*

Proof Let $\mathbf{T}$ be a linear transformation of $\mathbf{S}$ with general term $\mathbf{T}_{ab} = u\mathbf{S}_{ab} + v$ where $u > 0$. For $t = 1$, we can write $\mathbf{T}^1_{ab} = u\mathbf{S}^1_{ab} + v$. Assume that $\mathbf{T}^t_{ij} = u\mathbf{S}^t_{ij} + v, \forall i,j \in \mathbb{C}^t$ and let us prove that $\mathbf{T}^{t+1}_{ij} = u\mathbf{S}^{t+1}_{ij} + v, \forall i,j \in \mathbb{C}^{t+1}$. Let k and l be the clusters that are fused at iteration t. First, we need to prove that $\mathbf{T}^{t+1}_{(kl)m} = u\mathbf{S}^{t+1}_{(kl)m} + v, \forall m \in \mathbb{C}^{t+1}, m \neq (kl)$. We have:

$$\begin{aligned} \mathbf{T}^{t+1}_{(kl)m} &= \mathfrak{a}(k,l)\mathbf{T}^t_{km} + \mathfrak{a}(l,k)\mathbf{T}^t_{lm} \\ &= \mathfrak{a}(k,l)(u\mathbf{S}^t_{km} + v) + \mathfrak{a}(l,k)(u\mathbf{S}^t_{lm} + v) \\ &= u(\mathfrak{a}(k,l)\mathbf{S}^t_{km} + \mathfrak{a}(l,k)\mathbf{S}^t_{lm}) + v(\mathfrak{a}(k,l) + \mathfrak{a}(l,k)) \\ &= u\mathbf{S}^{t+1}_{(kl)m} + v, \end{aligned}$$

providing that $\mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1$.

Next, we prove that $\mathbf{T}^{t+1}_{(kl)(kl)} = u\mathbf{S}^{t+1}_{(kl)(kl)} + v$. Indeed, we have:

$$\begin{aligned} \mathbf{T}^{t+1}_{(kl)(kl)} &= \mathfrak{b}(k,l)\mathbf{T}^t_{kl} + \mathfrak{c}(k,l)\mathbf{T}^t_{kk} + \mathfrak{c}(l,k)\mathbf{T}^t_{ll} \\ &= \mathfrak{b}(k,l)(u\mathbf{S}^t_{kl} + v) + \mathfrak{c}(k,l)(u\mathbf{S}^t_{kk} + v) + \mathfrak{c}(l,k)(u\mathbf{S}^t_{ll} + v) \\ &= u(\mathfrak{b}(k,l)\mathbf{S}^t_{kl} + \mathfrak{c}(k,l)\mathbf{S}^t_{kk} + \mathfrak{c}(l,k)\mathbf{S}^t_{ll}) + v(\mathfrak{b}(k,l) + \mathfrak{c}(k,l) + \mathfrak{c}(l,k)) \\ &= u\mathbf{S}^{t+1}_{(kl)(kl)} + v \end{aligned}$$

providing that $\mathfrak{b}(k,l) + \mathfrak{c}(k,l) + \mathfrak{c}(l,k) = 1$.

Denote respectively $\{\mathbf{\Lambda}^t\}_{t\in\mathbb{T}}$ and $\{\mathbf{\Delta}^t\}_{t\in\mathbb{T}}$, the sequences of square matrices defined by (13), (12a) and (12b), and obtained when $\mathbf{S}$ and $\mathbf{T} = u\mathbf{S} + v\mathbf{1}_n$ are the input kernel matrices respectively. We have, $\forall t \in \mathbb{T}, \forall i,j \in \mathbb{C}^t$:

$$\begin{aligned} \mathbf{\Delta}^t_{ij} &= \mathbf{T}^t_{ij} - \frac{1}{2}(\mathbf{T}^t_{ii} + \mathbf{T}^t_{jj}) \\ &= u\mathbf{S}^t_{ij} + v - \frac{1}{2}(u\mathbf{S}^t_{ii} + v + u\mathbf{S}^t_{jj} + v) \\ &= u(\mathbf{S}^t_{ij} - \frac{1}{2}(\mathbf{S}^t_{ii} + \mathbf{S}^t_{jj})) \\ &= u\mathbf{\Lambda}^t_{ij} \end{aligned}$$

Since $u > 0$ then, $\arg\max_{(i,j)\in\mathbb{C}^t, i\neq j} \mathfrak{p}(i,j)\mathbf{\Delta}^t_{ij} = \arg\max_{(i,j)\in\mathbb{C}^t, i\neq j} \mathfrak{p}(i,j)\mathbf{\Lambda}^t_{ij}, \forall t \in \mathbb{T}$. ■

Moreover, let us remind that a positive linear transformation of the terms of a positive semi-definite matrix provides a positive semi-definite matrix. Therefore, $\mathbf{S}$ remains a kernel matrix after (24) is carried out.

Henceforth, we assume that $\mathbf{S}$ is a NK matrix. In fact, such a matrix enjoys a double interpretation. On the one hand, it gives the inner-products of points represented (on an hypersphere) in an Hilbert space. On the other hand, it can be seen as a similarity matrix satisfying the conditions[11] given in Definition 6.

As a consequence, in the rest of the paper, NK matrix and similarity matrix are terms that we use interchangeably.

5.2 Penalized Similarities: Aggregated Inter-similarities *Versus* Aggregated Intra-similarities

Supposing (C1), (C2) and (C3), we discuss a new interpretation of K-AHC based on similarities. Equation (14) is a maximum search over the set of couples of clusters in $\mathbb{C}^t$. The quality of a pair (i,j) depends on $\mathbf{\Lambda}^t_{ij}$ defined in (13) which is the difference between $\mathbf{S}^t_{ij}$ and the arithmetic mean of $\mathbf{S}^t_{ii}$ and $\mathbf{S}^t_{jj}$. Let us call $\mathbf{S}^t_{ij}$, the *inter-similarity* between clusters i and j, and $\mathbf{S}^t_{ii}$, the *intra-similarity* of cluster i. For the couple of clusters (i,j), $\mathbf{\Lambda}^t_{ij}$ can be seen as their inter-similarity value *penalized* by the arithmetic mean of their respective intra-similarities. According to (13), $\mathbf{\Lambda}^t_{ij}$ is great if the inter-similarity is high *and* the intra-similarities are low. Consequently, i and j are more likely to be merged together if their inter-similarity is high enough with respect to their intra-similarities.

In this context, it is important to formally state the properties of the set functions defining the six schemes we deal with. From Table 3, we can observe that for all methods:

$$\begin{cases} \mathfrak{a}(k,l), \mathfrak{b}(k,l), \mathfrak{c}(k,l) \geq 0 \\ \mathfrak{a}(k,l) + \mathfrak{a}(l,k) = 1 \\ \mathfrak{b}(k,l) + \mathfrak{c}(k,l) + \mathfrak{c}(l,k) = 1 \end{cases}, \quad \forall k,l \in 2^{\mathbb{O}} \tag{25}$$

Therefore, $\{\mathfrak{a}(k,l), \mathfrak{a}(l,k)\}$ and $\{\mathfrak{b}(k,l), \mathfrak{c}(k,l), \mathfrak{c}(l,k)\}$ can be seen as *weight vectors* and we interpret (12a) and (12b) as *averages* of inter-similarities and intra-similarities respectively. From this viewpoint, the differences between the techniques can be understood from their distinct *averaging strategies*.

In order to have a more precise view of the differences between the six methods, let us take an example. Suppose $\mathbb{O} = \{a,b,c,d,e,f,g\}$ and at iteration $t = 4$, $\mathbb{C}^4 = \{k,l,m\}$ with $k = \{a,b,c\}$, $l = \{d,e\}$, $m = \{f,g\}$. Assume that (k,l) is the couple of clusters to be merged. In Figure 2, we illustrate this situation using the input similarity matrix $\mathbf{S}$. The different elements involved in (12a) and (12b) are shown. They correspond to rectangular blocks for inter-similarities $(\mathbf{S}^4_{kl}, \mathbf{S}^4_{lm}, \mathbf{S}^4_{kl})$ and to square blocks for intra-similarities $(\mathbf{S}^4_{kk}, \mathbf{S}^4_{ll}, \mathbf{S}^4_{mm})$.

The inter-similarity $\mathbf{S}^5_{(kl)m}$ is an average of $\mathbf{S}^4_{km}$ and $\mathbf{S}^4_{lm}$ represented by dashed line blocks $k \times m$ and $l \times m$. Mcquitty, median and w-median assign the same weight $1/2$ to both terms whereas group average, centroid and Ward, assign weights which depend on one of the blocks side length.

The intra-similarity $\mathbf{S}^5_{(kl)(kl)}$ is an average of $\mathbf{S}^4_{kl}$, $\mathbf{S}^4_{kk}$ and $\mathbf{S}^4_{ll}$ which are highlighted by a dotted line block and two solid line blocks respectively. Since $\mathbf{S}^4$ is symmetric, it is equivalent to consider that $\mathbf{S}^5_{(kl)(kl)}$ depends on $\mathbf{S}^4_{kl}$, $\mathbf{S}^4_{lk}$, $\mathbf{S}^4_{kk}$ and $\mathbf{S}^4_{ll}$. We can see that these four elements depict a partition of the block $(k\cup l)\times(k\cup l)$. For geometric methods, all four sub-blocks contribute to $\mathbf{S}^5_{(kl)(kl)}$. In the median and w-median schemes, all sub-blocks are assigned a uniform weight of $1/4$ which amounts to an unweighted mean. Regarding centroid

11. Note that in this context, the maximal self-similarity property is due to the Cauchy-Schwartz inequality.

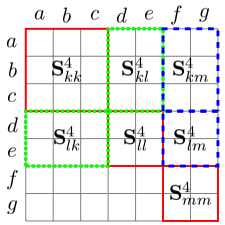


Figure 2: Illustration of inter-similarities and intra-similarities.

and Ward, the weights are distributed with respect to the blocks area. Conversely, for graph methods group average and Mcquitty, $\mathbf{S}^5_{(kl)(kl)}$ only depends on $\mathbf{S}^4_{kk}$ and $\mathbf{S}^4_{ll}$. Consequently, it is not difficult to see that only the on-diagonal terms of $\mathbf{S}$ are involved in the computation of the clusters intra-similarity for these two latter schemes. This observation was already underlined in sub-section 4.3.

5.3 Sparsified Normalized Kernel Matrix

In order to cope with the storage complexity of K-AHC, we sparsify the NK matrix by removing the lowest similarity values. We obtain a *Sparsified Normalized Kernel (SNK) matrix*.

The first sparsification procedure we introduce is a simple thresholding operator[12] based on a real parameter $\theta \geq 0$.

$$\mathbf{S}_{ab} \leftarrow \begin{cases} \mathbf{S}_{ab} & \text{if } \mathbf{S}_{ab} \geq \theta \\ 0 & \text{otherwise} \end{cases}, \quad \forall a, b \in \mathbb{O} \tag{26}$$

Note that under (C2), we have:

$$\begin{aligned} \mathbf{D}_{ab} &= \mathbf{S}_{aa} + \mathbf{S}_{bb} - 2\mathbf{S}_{ab} \\ &= 2(w - \mathbf{S}_{ab}), \quad \forall a, b \in \mathbb{O} \end{aligned}$$

where $\mathbf{S}_{aa} = w, \forall a \in \mathbb{O}$.

Thereby, the entries which correspond to the greatest values in $\mathbf{S}$ are exactly the same entries in $\mathbf{D}$ having the lowest values.

The second sparsification approach is based on the nearest neighbors. Let $\mathbb{NN}_k(a)$ be the set of the k elements closest to a according to $\mathbf{S}$ (or $\mathbf{D}$ equivalently). Then, we define:

$$\mathbf{S}_{ab} \leftarrow \begin{cases} \mathbf{S}_{ab} & \text{if } b \in \mathbb{NN}_k(a) \text{ or } a \in \mathbb{NN}_k(b) \\ 0 & \text{otherwise} \end{cases}, \quad \forall a, b \in \mathbb{O} \tag{27}$$

We point out that for each $a \in \mathbb{O}$, the number of non-null values in the similarity profile $\{\mathbf{S}_{ab}\}_{b \in \mathbb{O}}$ is lower bounded by k. Apart from the k closest items to a in $\mathbb{NN}_k(a)$, an item

12. When using distances, this sparsification procedure is equivalent to the epsilon-neighborhood method.

$c \notin \mathbb{NN}_k(a)$ could have a in its k nearest neighbors in which case $\mathbf{S}_{ca}$ but also $\mathbf{S}_{ac}$ would be non-null. Consequently, if $k = round(n/2)$ for instance, then $\mathbf{S}$ memory usage is not necessarily divided by 2, but by a factor which is lower or equal to 2.

Besides, it should be clear that determining the exact k nearest neighbors graph basically takes $O(n^2)$ time. However, there are different ways to speed-up this procedure (see for e.g. (Franti et al., 2006) and references therein).

Observe that after (26) or (27) is performed, the sparsified similarity matrix $\mathbf{S}$ is no longer positive semi-definite. Thereby, the geometric context we have assumed so far does not hold for a SNK matrix. Nonetheless, as we shall show in sub-section 5.5, three out of the six techniques are not concerned with this issue.

5.4 Performing K-AHC on a SNK Matrix in an Efficient and Effective Manner

We carry out K-AHC on a SNK matrix $\mathbf{S}$. However, owing to the distinct interpretations we can give to $\mathbf{S}$, as exposed in sub-section 5.2, we propose some substantial modifications to Algorithm 2 that lead to interesting properties.

In the stored proximities based AHC algorithms D-AHC or K-AHC, the bottleneck that causes an heavy computational cost is the search for the pair of clusters to fuse. This operation is carried out over the set of all possible pairs in $\mathbb{C}^t \times \mathbb{C}^t$ which has $O(n^2)$ cost. Since there are $n-1$ iterations, the overall time complexity is thus $O(n^3)$.

In our case, $\mathbf{S}$ is a sparse similarity matrix and we introduce the following subset:

$$\mathbb{S} = \{(a, b) \in \mathbb{O} \times \mathbb{O}, \mathbf{S}_{ab} > 0\} \tag{28}$$

Likewise, during the course of the bottom-up merging mechanism, we determine at each iteration, the following subsets:

$$\mathbb{S}^t = \{(i, j) \in \mathbb{C}^t \times \mathbb{C}^t, \mathbf{S}^t_{ij} > 0\}, \quad \forall t \in \mathbb{T} \tag{29}$$

Note that $\mathbb{S}^{t+1}$ can be easily updated from $\mathbb{S}^t$.

In contrast to K-AHC, for each $t \in \mathbb{T}$, SNK-AHC searches for the pair to merge among the elements in $\mathbb{S}^t$ only. Accordingly, we replace (14) with:

$$(k, l) = \underset{(i,j) \in \mathbb{S}^t, i \neq j}{\arg\max} \ \mathfrak{p}(i, j) \mathbf{\Lambda}^t_{ij} \tag{30}$$

Therefore, whatever the value $\mathfrak{p}(i, j)\mathbf{\Lambda}^t_{ij}$, two clusters i and j can not be merged together if they share no similarity at all. SNK-AHC is thus a *constrained AHC procedure*.

The SNK-AHC pseudo-code is given in Algorithm 4. It also describes a bottom-up algorithm but unlike K-AHC, the dendrogram grows as long as $\mathbb{S}^t \neq \emptyset$. As a consequence, the output of Algorithm 4 is not a tree in general but a *forest*. We investigate this point further in sub-section 5.6.

It is clear that restricting the search to $\mathbb{S}^t$ makes it possible to obtain a much more scalable dendrogram building procedure.

Proposition 15 *Let z be the number of non-null entries in $\mathbf{S}$ after the sparsification step in Algorithm 4 has been performed. Then, the bottom-up procedure of Algorithm 4 has $O(z)$ storage complexity and $O(nz)$ processing time complexity.*

Algorithm 4: General procedure of SNK-AHC.

Input: $\mathbf{S}$ a kernel matrix, a sparsification method, an AHC method
Output: D a dendrogram
if *the diagonal of* $\mathbf{S}$ *is not constant* **then**
  Normalize $\mathbf{S}$ using (23);
end
Translate $\mathbf{S}$ using (24);
Sparsify $\mathbf{S}$ using (26) or (27);
Initialize D with n leaves;
Set $\mathbf{S}^1 = \mathbf{S}$;
Determine $\mathbb{S}$ according to (28) and set $\mathbb{S}^1 = \mathbb{S}$;
while $\mathbb{S}^t \neq \emptyset$ **do**
  Find the pair of clusters (k, l) according to (30) with the corresponding AHC method parameters values given in Table 3 ;
  Merge (k, l) into (kl) and update D;
  Update $\mathbb{S}^{t+1}$ from $\mathbb{S}^t$;
  Compute $\mathbf{S}^{t+1}$ by applying (12a) and (12b) with the corresponding AHC method parameters values given in Table 3.
end

Note, however, that if $\mathbf{S}$ is a dense NK matrix which has not been sparsified, and $\mathbf{D}$ is the related distance matrix following (C1), then Algorithm 4 provides the exact same result as Algorithm 2 and thus an output equivalent to the one obtained with Algorithm 1 as well, according to Theorem 8.

As we shall see in section 6 dedicated to the experiments, not only SNK-AHC can be dramatically more efficient than D-AHC from a computational standpoint, but it also enables improving the quality of the clustering results on challenging problems.

5.5 Diagonal Translation Invariance

As highlighted in sub-section 5.3, the SNK matrix $\mathbf{S}$ is not positive semi-definite and we can not assume that the points belong to an Hilbert space any more. However, we can recover this feature quite easily. Indeed, since $\mathbf{S}$ is symmetric and all its diagonal entries are non-negative then, one simple way to make $\mathbf{S}$ positive semi-definite again, is to sufficiently augment the values of the diagonal entries in order to make $\mathbf{S}$ strictly diagonally dominant (see for e.g. (Horn and Johnson, 1986, Theorem 6.1.10)).

While we can always do this, we show, in what follows, that this is not necessary for some techniques.

Let us introduce the following matrix:

$$\mathbf{T} = \mathbf{S} + w\mathbf{I}_n \tag{31}$$

where $\mathbf{I}_n$ is the identity matrix of order n and $w > 0$ is chosen such that $\mathbf{T}$ is positive semi-definite.

Let $M = \{m_1, \ldots, m_{n-1}\}$ be a sequence of merges following Definition 4. By observing that (12a) does not depend on any on-diagonal entry of the SNK matrix then, it is easy to check the following result.

Lemma 16 *Let* $\{\mathbf{S}^t\}_{t\in\mathbb{T}}$ *and* $\{\mathbf{T}^t\}_{t\in\mathbb{T}}$ *be the sequences of square matrices with input elements* $\mathbf{S}$ *and* $\mathbf{T} = \mathbf{S} + w\mathbf{I}_n, w \in \mathbb{R}$ *and subsequent elements defined by* (12a) *and* (12b). *Suppose that* $\{\mathbf{S}^t\}_{t\in\mathbb{T}}$ *and* $\{\mathbf{T}^t\}_{t\in\mathbb{T}}$ *are determined with respect to the same sequence of merges* M. *Then, we have:*

$$\mathbf{T}^t_{ij} = \mathbf{S}^t_{ij}, \quad \forall t \in \mathbb{T}, \forall i, j \in \mathbb{C}^t, i \neq j$$

Lemma 16 indicates that when merging the same sequence of pairs of clusters, the off-diagonal entries of similarity matrices $\{\mathbf{S}^t\}_{t\in\mathbb{T}}$ and $\{\mathbf{T}^t\}_{t\in\mathbb{T}}$ are identical. It is only the intra-similarities that are influenced by the diagonal translation.

For group average, Mcquitty and Ward, we have the following relationships.

Lemma 17 *Let* $\{\mathbf{S}^t\}_{t\in\mathbb{T}}$ *and* $\{\mathbf{T}^t\}_{t\in\mathbb{T}}$ *be the sequences of square matrices with input elements* $\mathbf{S}$ *and* $\mathbf{T} = \mathbf{S} + w\mathbf{I}_n, w \in \mathbb{R}$ *and subsequent elements defined by* (12a) *and* (12b). *Suppose that* $\{\mathbf{S}^t\}_{t\in\mathbb{T}}$ *and* $\{\mathbf{T}^t\}_{t\in\mathbb{T}}$ *are determined with respect to the same sequence of merges* M. *Then, for group average and Mcquitty, we have:*

$$\mathbf{T}^t_{ii} = \mathbf{S}^t_{ii} + w, \quad \forall t \in \mathbb{T}, \forall i \in \mathbb{C}^t$$

Regarding Ward, we have:

$$\mathbf{T}^t_{ii} = \mathbf{S}^t_{ii} + \frac{w}{|i|}, \quad \forall t \in \mathbb{T}, \forall i \in \mathbb{C}^t$$

Proof Let us consider the group average technique and its parameters values given in Table 3. For $t = 1$, it follows from the definition given in (31) that $\mathbf{T}^1_{aa} = \mathbf{S}^1_{aa} + w, \forall a \in \mathbb{C}^1$. Assume that $\mathbf{T}^t_{ii} = \mathbf{S}^t_{ii} + w, \forall i \in \mathbb{C}^t$ for t. Then, let us prove that the latter relation is true for $t + 1$ as well. Suppose that at iteration t, the pair of clusters (k, l) is merged. By applying Lemma 16, it comes:

$$\begin{aligned}
\mathbf{T}^{t+1}_{(kl)(kl)} - \mathbf{S}^{t+1}_{(kl)(kl)} &= \mathfrak{b}(k,l)\mathbf{T}^t_{kl} + \mathfrak{c}(k,l)\mathbf{T}^t_{kk} + \mathfrak{c}(l,k)\mathbf{T}^t_{ll} - (\mathfrak{b}(k,l)\mathbf{S}^t_{kl} + \mathfrak{c}(k,l)\mathbf{S}^t_{kk} + \mathfrak{c}(l,k)\mathbf{S}^t_{ll}) \\
&= \mathfrak{c}(k,l)(\mathbf{T}^t_{kk} - \mathbf{S}^t_{kk}) + \mathfrak{c}(l,k)(\mathbf{T}^t_{ll} - \mathbf{S}^t_{ll}) \\
&= w(\mathfrak{c}(k,l) + \mathfrak{c}(l,k)) \\
&= w
\end{aligned}$$

since $\mathfrak{c}(k,l) + \mathfrak{c}(l,k) = 1$ for group average.
The proofs for Mcquitty and Ward are similar. ■

Theorem 18 *For group average, Mcquitty and Ward methods, Algorithm 4 provides equivalent dendrograms for input similarity matrices* $\mathbf{S}$ *and* $\mathbf{T} = \mathbf{S} + w\mathbf{I}_n$, $w \in \mathbb{R}$.

Proof Denote respectively $\{\mathbf{\Lambda}^t\}_{t\in\mathbb{T}}$ and $\{\mathbf{\Delta}^t\}_{t\in\mathbb{T}}$, the sequences of penalized similarity matrices obtained using $\mathbf{S}$ and $\mathbf{T}$ as input similarity matrices. We prove that for all $t \in \mathbb{T}$, the couple of clusters maximizing $\mathfrak{p}(i,j)\mathbf{\Delta}^t_{ij}$ is the same as the one maximizing $\mathfrak{p}(i,j)\mathbf{\Lambda}^t_{ij}$. To this end, note that a sufficient condition is $\mathfrak{p}(i,j)(\mathbf{\Delta}^t_{ij} - \mathbf{\Lambda}^t_{ij}) = c$, $\forall t \in \mathbb{T}$, where c is a constant. For $t = 1$, we have:

$$\begin{aligned}\mathfrak{p}(a,b)(\mathbf{\Delta}^1_{ab} - \mathbf{\Lambda}^1_{ab}) &= \mathfrak{p}(a,b)(\mathbf{T}^1_{ab} - \frac{1}{2}(\mathbf{T}^1_{aa} + \mathbf{T}^1_{bb}) - \mathbf{S}^1_{ab} + \frac{1}{2}(\mathbf{S}^1_{aa} + \mathbf{S}^1_{bb}))\\ &= -\frac{\mathfrak{p}(a,b)}{2}((\mathbf{T}^1_{aa} - \mathbf{S}^1_{aa}) + (\mathbf{T}^1_{bb} - \mathbf{S}^1_{bb}))\\ &= \underbrace{-\mathfrak{p}(1,1)w}_{c}\end{aligned}$$

where, by abusing the notation, $\mathfrak{p}(1,1)$ denotes $\mathfrak{p}(i,j)$ whenever $|i|=|j|=1$.

For $t = 1$, it is clear that using either $\mathbf{S}$ or $\mathbf{T}$ as input matrices, leads to the same merge. As a consequence, by applying Lemma 16, it is sufficient to prove by induction that:

$$-\frac{\mathfrak{p}(i,j)}{2}((\mathbf{T}^t_{ii} - \mathbf{S}^t_{ii}) + (\mathbf{T}^t_{jj} - \mathbf{S}^t_{jj})) = -\mathfrak{p}(1,1)w, \quad \forall t \in \mathbb{T}, \forall i,j \in \mathbb{C}^t, i \neq j \tag{32}$$

Concerning the group average and the Mcquitty techniques, for both cases $\mathfrak{p}(i,j) = 1, \forall i,j \in 2^{\mathbb{O}}$ and thus $c = -w$. Let assume that at iteration t, the pair of clusters (k,l) is merged. By applying Lemma 17, we have:

$$\begin{aligned}-\frac{1}{2}((\mathbf{T}^{t+1}_{(kl)(kl)} - \mathbf{S}^{t+1}_{(kl)(kl)}) + (\mathbf{T}^{t+1}_{mm} - \mathbf{S}^{t+1}_{mm})) &= -\frac{1}{2}(w + w)\\ &= -w\end{aligned}$$

In the case of Ward, $c = -\mathfrak{p}(1,1)w = -\frac{w}{2}$. By using Lemma 17 again, we have:

$$\begin{aligned}&-\frac{\mathfrak{p}((kl),m)}{2}((\mathbf{T}^{t+1}_{(kl)(kl)} - \mathbf{S}^{t+1}_{(kl)(kl)}) + (\mathbf{T}^{t+1}_{mm} - \mathbf{S}^{t}_{mm}))\\ &= -\frac{|(kl)||m|}{2(|(kl)|+|m|)}(\frac{w}{|(kl)|} + \frac{w}{|m|})\\ &= -\frac{w}{2}\end{aligned}$$

■

Consequently, for these three schemes, the geometrical representation of the objects lying in an Hilbert space is still valid when applying Algorithm 4, even though the SNK matrix $\mathbf{S}$ is not positive semi-definite.

For the other schemes, centroid, median and w-median, some preliminary empirical tests showed that making $\mathbf{S}$ positive semi-definite again is not recommended. Indeed, in these cases, increasing the diagonal provided worst performances in terms of clustering quality. It appears that such a transformation results in a space distortion to which, these latter methods are highly sensitive. Therefore, in Algorithm 4, we do not include a step for diagonal translation by default.

5.6 Clusters as Connected Components

One important issue in clustering is to determine the number of clusters. To some extent, Algorithm 4 is able to address this challenge. In order to detail this property, we place ourselves in the framework of graph theory.

Let G be an undirected graph with $\mathbb{O}$ being the set of nodes and $\mathbb{S}$, defined in (28), being the set of edges. G is connected if for every pair $(a,b) \in \mathbb{O}\times\mathbb{O}$, there is a path joining both nodes. If G is not connected then $\mathbb{O}$ can be separated with respect to its *connected components*. These latter subsets form a partition of $\mathbb{O}$. From a clustering viewpoint, the connected components can be seen as clusters.

One way to determine the connected components of an undirected graph is to use a disjoint sets data structure which typically: (i) puts nodes in a same set if there is a path joining each other and (ii) assigns a representative item to each set (see for e.g. (Cormen et al., 2009, Chapter 21)). In this context, three operations are employed:

- `make_set`(a): creates a set whose only member is a and takes a as its representative.
- `find_set`(a): finds the representative of the set a belongs to.
- `union`(a,b): unites the two disjoint sets that a and b belong to, removes the two latter sets and determine a representative for the new set.

We review in Algorithm 5 the pseudo-code that builds a disjoint sets data structure given a graph $G = (\mathbb{O}, \mathbb{S})$ and outputs the connected components.

Algorithm 5: Connected components determination.

```
Input: G = (𝕆, 𝕊)
Output: The connected components
for a ∈ 𝕆 do
    make_set(a)
end
for (a, b) ∈ 𝕊 do
    if find_set(a) ≠ find_set(b) then
        union(a,b)
    end
end
```

As an illustration, we provide an example in Figure 3. We represent a graph with seven nodes and four edges : $\mathbb{O} = \{a,b,c,d,e,f,g\}$ and $\mathbb{S} = \{(a,b),(a,c),(d,e),(e,f)\}$.

Applying Algorithm 5 to this example, provides the following connected components:

- $a : \{a,b,c\}$,
- $d : \{d,e,f\}$.

In this example, the representative item of a subset is chosen with respect to the lexical order.

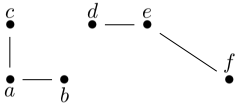


Figure 3: Illustration of a disconnected graph.

In fact, all AHC Algorithms 1-4 we have introduced so far, rely on a bottom-up fusion mechanism that reproduces the same operations than in Algorithm 5. The difference is that instead of scanning all unitary edges $(a, b) \in \mathbb{S}$ in the for loop in Algorithm 5, AHC procedures go through the consolidated edges between the representative items of the disjoint sets (the clusters). Moreover, unlike in Algorithm 5, the edges are not picked randomly in AHC algorithms but they are chosen in the goal of optimizing a criterion which is the weighted dissimilarity value in the case of D-AHC and the weighted penalized similarity value in the cases of K-AHC and SNK-AHC.

Furthermore, in Algorithms 1 and 2, the input proximity matrices are dense and the underlying graphs are thus fully connected. Therefore, these algorithms necessarily produce single trees as outputs. On the contrary, Algorithm 4 uses a sparse similarity matrix and in this case, G might not be connected, especially if $z << n^2$. In such a case, Algorithm 4 outputs a forest and each tree is a connected component which can be considered as a cluster.

This reasoning is summarized in the following statement.

Proposition 19 *Let* $\mathbf{S}$ *be the sparse similarity matrix obtained after the sparsification step of Algorithm 4 and let* $\mathbb{S}$ *be the set of pairs of objects defined by* (28)*. Let* $G = (\mathbb{O}, \mathbb{S})$ *be the associated undirected graph. If* G *is not connected and has* κ *connected components then Algorithm 4 stops at iteration* $n - \kappa - 1$*. Moreover, it outputs a forest where each tree is a connected component.*

Accordingly, note that SNK-AHC output is not a complete dendrogram in general.

6. Experiments

In this section we demonstrate the different properties and advantages of SNK-AHC over D-AHC using different benchmark data sets. We both use artificial and real-world problems which are freely available at (Franti and et al, 2015) and (Lichman, 2013) respectively.

Firstly, we compare the dendrograms obtained by D-AHC and SNK-AHC. Our first purpose is to verify that when no sparsification is performed, SNK-AHC is equivalent to D-AHC. Secondly, we are interested in assessing the proximity between the dendrograms given by D-AHC and SNK-AHC. Thirdly, on medium-size real-world data sets, we demonstrate that Algorithm 4 indeed allows reducing the D-AHC computational costs dramatically. Finally, for all data sets, we show that sparsifying the similarity matrix can also provide better clustering results.

We introduce below the different assessment criteria we used in our experiments, before presenting the benchmarks and the results we obtained.

6.1 Evaluation Measures

In order to measure the proximity between the dendrograms obtained by Algorithms 1 and 4, we use the cophenetic matrices and correlation coefficient (see for e.g. (Everitt et al., 2009, Section 4.4.2)). Given a dendrogram D, the derived cophenetic matrix denoted $\mathbf{C}(D)$, is a pairwise matrix of order n where, for each pair of items $(a, b) \in \mathbb{O} \times \mathbb{O}$, we record the height (D-AHC) or depth value (SNK-AHC) of the node that merges a and b for the first time. Then, let D_{dahc} and D_{snkahc} be the dendrograms obtained by both techniques. The cophenetic correlation is the product moment correlation between the vectorized upper triangular matrices of $\mathbf{C}(D_{dahc})$ and $\mathbf{C}(D_{snkahc})$. We take the opposite value of this measure which is denoted CC. In this case, CC=1 implies that the dendrograms are equivalent.

Next, in order to evaluate the quality of the clustering results we apply an external validation methodology since we are given the correct partition for all data sets. For each obtained dendrogram, we cut the forest so as to obtain the correct number of clusters denoted κ^*. Note that if κ, the number of clusters found by Algorithm 4, is greater than κ^* then, we keep the partition with κ clusters. Afterward, we compare the resulting partition and the ground-truth. The evaluation measure used in this case is the famous adjusted Rand index (Hubert and Arabie, 1985) which is denoted ARI. In this case as well, the greater the better, and ARI=1 means that the ground-truth was recovered perfectly.

Regarding scalability, our baselines are the D-AHC computational costs. Therefore, the memory and running time reductions are reported in comparison to the performances obtained by D-AHC. Relative storage and processing time decreases are thus examined. However, note that these points are mainly analyzed in the case of real-world benchmarks. Indeed, synthetic data sets are small-size and, in these cases it is not worth discussing computational gains in details.

6.2 Artificial Data

Small-size artificial data sets are used in the goal of illustrating the ability of SNK-AHC to address challenging clustering tasks.

In all experiments, before computing the inner-product matrix, we centered and scaled the data matrix with respect to the mean and standard deviation of the variables.

6.2.1 Aggregation Data

The first benchmark is taken from (Gionis et al., 2007). It consists of 788 points in a two-dimensional space. The objects and clusters are represented in Figure 4. There are seven different groups to identify. These clusters have different sizes and shapes. They can be non-convex and connected as well.

In (Gionis et al., 2007), the authors show the shortcomings of classic D-AHC methods such as the single linkage, complete linkage, group average and Ward schemes. The k-means algorithm also fails to recover the seven clusters. In this previous work, Euclidean distances were used.

In Table 5, we report the performance measures obtained by the different schemes used in the framework of SNK-AHC. A Gaussian kernel and the nearest neighbors sparsification operator $\mathbb{NN}_k$ were applied.

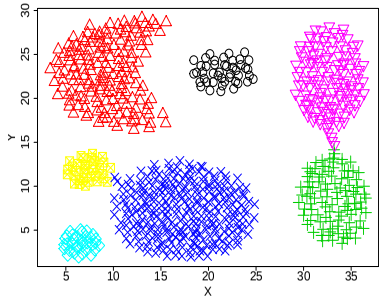


Figure 4: Aggregation data set.

Regarding the Gaussian kernel, we remind its definition below:

$$\mathbf{S}_{ab} = \exp(-\gamma \|\mathbf{x}^a - \mathbf{x}^b\|^2), \quad \forall a, b \in \mathbb{O}$$

We set $\gamma = 1/q$, q being the number of descriptive variables. This default setting is used in popular SVM tools like (Chang and Lin, 2011).

Concerning $\mathbb{NN}_k$, the distinct k values were successively set to (the nearest integer of) $\{100, 90, 75, 50, 25, 10, 1\}$ percent of n, the total number of items. This results in a sequence of sparser and sparser SNK matrices. The two opposite cases are the following ones. $k = n$ corresponds to 100% of the nearest neighbors and in that case, no sparsification is carried out. Applying SNK-AHC without any sparsification is equivalent to K-AHC or D-AHC. This situation corresponds to our baseline. By contrast, setting $k = round(n/100)$ refers to the sparsest similarity matrix that we used.

From Table 5, we observe that:

- For all schemes, when $k = 788$, CC=1. In other words, the obtained dendrograms are equivalent to the ones given by D-AHC. These observations empirically confirm Theorem 8.
- For all methods, a sparse $\mathbf{S}$ matrix that was reduced by up to half of its original memory size, does not alter the quality of SNK-AHC outputs. Therefore, we can improve the scalability without decreasing the ARI values. In addition, we note that the CC values are close to 1. It is likely that the dendrograms we obtain in theses cases are equivalent to the baseline.
- Below fifty percent of nearest neighbors, the performances are not stable and the ARI behavior depends on the method. For median and w-median, the sparsification beyond fifty percent, has a negative effect.
- Conversely, among the interesting cases, group average with only one percent of nearest neighbors was able to recover the correct partition with seven classes (ARI=1).

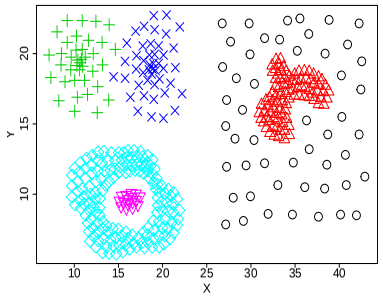


Figure 5: Compound data set.

With the same sparsification level, Ward dramatically improves over its baseline (dense $\mathbf{S}$) since the ARI value increases from 0.688 to 0.965.

- For all methods, κ, the number of clusters found by SNK-AHC, equals one except when $k = 8$. The latter setting corresponds to the sparsest similarity matrix. Only in this case, the underlying graph becomes disconnected and all schemes found five clusters. In Figure 4, the groups that were discovered are the three disconnected clusters (red triangles, black circles and cyan diamonds) and the two pairs of connected clusters which are respectively put together.

6.2.2 Compound Data

The second synthetic data set is a composition of several clustering tasks originally proposed in (Zahn, 1971). It consists of 399 points in a two-dimensional space. The data set is shown in Figure 5. There are six distinct groups of points that are identified with different symbols and colors. This task is particularly challenging since the clusters present very different patterns and are highly non-convex and non-linearly separable.

Similarly to the previous case, we applied a Gaussian kernel using the same default setting. However, the sparsification method we used here is based on a threshold following (26). The different θ values were chosen so that a certain level of sparsity is reached. Precisely, they correspond to the $\{100, 90, 75, 50, 25, 10, 1\}$th percentiles of the similarity values distribution. The 100th percentile does not yield any sparsification. Again, this case is considered as our baseline. On the contrary, the 1th percentile setting means that 99% of the similarity values were thresholded to zero. This latter case is the sparsest $\mathbf{S}$ matrix we experimented with.

The results we obtained are given in Table 6 and we can make the following comments:

- Many observations are actually similar to the ones we made for the previous benchmark. Firstly, when no sparsification is applied, we obtain equivalent dendrograms

Method	$\mathbb{NN}_k$	CC	ARI	κ
Group average	788	1.000	0.991	1
	709	1.000	0.991	1
	591	0.999	0.991	1
	394	0.999	0.991	1
	197	0.954	0.742	1
	79	0.758	0.746	1
	8	-0.834	**1.000**	5
Mcquitty	788	1.000	0.706	1
	709	1.000	0.706	1
	591	1.000	0.706	1
	394	0.998	0.706	1
	197	0.865	0.702	1
	79	0.811	0.675	1
	8	-0.697	0.760	5
Centroid	788	1.000	1.000	1
	709	1.000	**1.000**	1
	591	0.999	**1.000**	1
	394	0.994	**1.000**	1
	197	0.938	0.795	1
	79	0.346	0.815	1
	8	-0.818	0.804	5
Median	788	1.000	0.996	1
	709	1.000	0.996	1
	591	0.998	0.996	1
	394	0.994	0.996	1
	197	0.766	0.621	1
	79	0.450	0.415	1
	8	-0.694	0.798	5
Ward	788	1.000	0.688	1
	709	1.000	0.688	1
	591	0.977	0.688	1
	394	0.998	0.688	1
	197	0.986	0.679	1
	79	0.825	0.562	1
	8	-0.804	0.965	5
W-Median	788	NA	0.780	1
	709	NA	0.780	1
	591	NA	0.780	1
	394	NA	0.780	1
	197	NA	0.690	1
	79	NA	0.664	1
	8	NA	0.590	5

Table 5: Results for Aggregation data set using a Gaussian kernel.

Method	θ	CC	ARI	κ
Group average	0.010	1.000	0.811	1
	0.143	1.000	0.811	1
	0.245	1.000	0.811	1
	0.463	0.999	0.811	1
	0.819	0.947	0.802	1
	0.948	-0.766	0.818	3
	0.996	-0.741	**0.906**	99
Mcquitty	0.010	1.000	0.776	1
	0.143	1.000	0.776	1
	0.245	1.000	0.776	1
	0.463	0.998	0.776	1
	0.819	0.908	0.793	1
	0.948	-0.697	0.808	3
	0.996	-0.718	**0.906**	99
Centroid	0.010	1.000	0.812	1
	0.143	1.000	0.812	1
	0.245	0.999	0.812	1
	0.463	0.999	0.812	1
	0.819	0.836	0.785	1
	0.948	-0.800	0.747	3
	0.996	-0.753	**0.906**	99
Median	0.010	1.000	0.764	1
	0.143	0.981	0.764	1
	0.245	0.987	0.764	1
	0.463	0.997	0.764	1
	0.819	0.746	0.374	1
	0.948	-0.699	0.746	3
	0.996	-0.733	**0.906**	99
Ward	0.010	1.000	0.501	1
	0.143	1.000	0.501	1
	0.245	1.000	0.501	1
	0.463	1.000	0.501	1
	0.819	0.986	0.615	1
	0.948	-0.744	0.440	3
	0.996	-0.628	**0.906**	99
W-Median	0.010	NA	0.547	1
	0.143	NA	0.547	1
	0.245	NA	0.547	1
	0.463	NA	0.547	1
	0.819	NA	0.547	1
	0.948	NA	0.561	3
	0.996	NA	**0.906**	99

Table 6: Results for Compound data set using a Gaussian kernel.

between D-AHC and SNK-AHC. Secondly, an $\mathbf{S}$ matrix that was reduced by half of its original size, provides the same clustering quality than the dense $\mathbf{S}$ matrix. This is true for all techniques. As a consequence, it is possible to divide the computational costs by 2 without degrading the ARI values. Still, when 75% and 90% of similarity values are discarded, we do not obtain consistent improvements. Depending on the scheme, these particular thresholding levels do not always produce better ARI values.

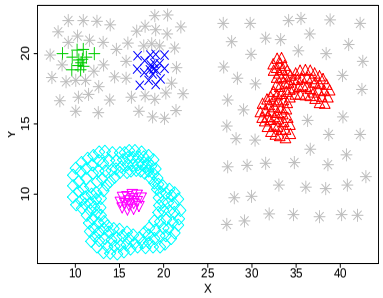


Figure 6: Compound data set results (clusters with size ≤ 3 are all represented by star symbols).

- Considering good performances, we underline the results obtained by the sparsest similarity matrix which only contains the 1% highest similarity values. The ARI value we obtain in this setting reaches 0.906 for all techniques which represents the best overall performance. All methods provided the same partition with 99 clusters. We provide in Figure 6 an illustration of the SNK-AHC outputs. For clarity reasons, we use the same star symbol to represent elements of clusters whose size is lower or equal to three. Note that among the 99 clusters, there are 89 singletons, 3 clusters of size two and 2 clusters of size three. We consider these groups as noise. Consequently, there are 5 "real" clusters that SNK-AHC was able to discover as depicted in Figure 6. Although the high ARI score of 0.906 is partly due to the fact that the SNK-AHC output is a partition with a number of clusters much larger than the ground-truth ($\kappa = 99$ *versus* $\kappa^* = 6$), it is important to precise that this result is a perfect sub-partition of the correct result.

- Finally, we emphasize the fact that SNK-AHC with the sparsest similarity matrix allows us to improve AHC from many viewpoints. Firstly, as we exposed previously, this case gives the best ARI scores and thus shows improvements in comparison with the baselines. The greatest refinement is again observed for the Ward technique whose ARI value increases from 0.501 to 0.906. Secondly, the computational costs of AHC are largely diminished. Last but not least, SNK-AHC was able to detect the core of five correct clusters which have diverse shapes and which were successfully separated from very small groups considered as noise.

6.3 Real-world Data

After having exemplified interesting properties of SNK-AHC on synthetic data sets, we address real-world clustering problems.

In this case, in addition to clustering quality, we discuss in more details the gains that SNK-AHC allow us to achieve in terms of scalability.

Likewise the previous set of experiments, the data matrix was centered and scaled before determining the inner-product matrices.

6.3.1 The Landsat Data Set

The first collection is called the landsat data set[13] which consists of 6,435 items. Each data unit corresponds to a set of 9 contiguous pixels disposed in a 3×3 patch. Each pixel is represented by its four spectral band values which are integer from 0 to 255. Consequently, the objects are described in a vectorial space of 36 dimensions. The task consists in recognizing the nature of an item among six different classes which are: red soil, cotton crop, grey soil, damp grey soil, soil with vegetation stubble, very damp grey soil.

Preliminary experimental results showed that the sparsification based on nearest neighbors provided better clustering results in comparison with the threshold based method. The Gaussian kernel also led to better performances as compared to the linear kernel. Consequently, we report the scores we obtained with these two best performing settings. Concerning $\mathbb{NN}_k$, the sequence of k values used in (27) was set to (the nearest integer of) $\{100, 90, 75, 50, 25, 10\}$ percent of n. As previously, these percentages give an estimation of the density of the $\mathbf{S}$ matrix.

The results we obtained are depicted in Figure 7. Several assessment measures are plotted. ARI and CC curves are represented by dotted lines with triangles and circles respectively. Moreover, the relative memory use and relative running time with respect to the computational costs of the baseline (dense $\mathbf{S}$) are represented. They correspond to solid lines with plus signs and dashed lines with cross signs respectively.

Below are the interesting outcomes we report from this set of experiments:

- On the scalability side, we verify that the sparser the SNK matrix, the lower the memory cost and the processing time since the curves of relative measurements of the two latter criteria clearly decrease. If z is the number of non-null entries in $\mathbf{S}$ then the relative time reduction is linear with respect to this latter variable. In other words, if we reduce $\mathbf{S}$ to 10% of its original memory size then the running time of SNK-AHC will also be reduced to 10% of its initial processing time. This result empirically illustrates Proposition 15.

- On the quality side, we can make the following comments. Firstly, if $k \geq round(n/2)$, the ARI values are not impacted very much whatever the AHC scheme. Likewise the previous benchmarks, we can save nearly half of the initial memory usage and running time without hurting the clustering quality. On the contrary, for the median and w-median approaches, the ARI values are even better. However, when $k < round(n/2)$, the clustering quality is not stable in general. Nevertheless, in the particular cases of group average and Mcquitty, the sparsest similarity matrix given by $k = round(n/10)$ provides the best performances for these techniques. The greatest gain concerns the group average scheme with an ARI score that is more than doubled since it jumped from 0.321 to 0.688.

13. `https://archive.ics.uci.edu/ml/datasets/Statlog+(Landsat+Satellite)`

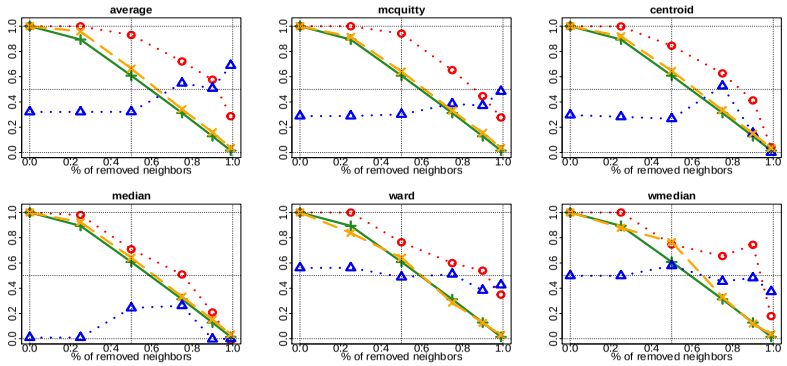


Figure 7: Results for the landsat data set using a Gaussian kernel. The x-axis corresponds to the % of removed neighbors. The y-axis corresponds to the observed values which all belong to $[0, 1]$. Solid lines with plus signs represent the relative memory use, dashed lines with cross signs show the relative running time, dotted lines with circles indicates the CC values, dotted lines with triangles give the ARI values.

- Regarding the new method w-median, it is worth mentioning that it plainly dominates the median scheme. It appears that the non-uniform weight we add to the median technique not only allows obtaining monotonic dendrograms, but it also enables boosting the clustering quality scores.

Note that all similarity graphs are connected even in the case of the sparsest similarity matrix. Thereby, whatever the setting, SNK-AHC always gave a single tree.

6.3.2 The Pendigits Data Set

The second collection we used, is called the pendigits data set[14]. This benchmark consists of handwritten digits that were collected from 44 different writers. Each one of them provided around 250 samples so that the entire collection is composed of 10,992 observations. Each sample is described by 16 numerical features. The 10 digits have equal frequency. Obviously, the task is to recognize groups of elements that correspond to the same digits.

In this case, we report the results we obtained with a linear kernel since they provided similar outcomes than a Gaussian kernel. However, the nearest neighbor sparsification outperformed the one relying on a threshold. Therefore, we report in Figure 8 the scores obtained with this former sparsification technique. We use the same sequence of neighborhood selection as before.

The results we obtain for this benchmark are pretty similar to the landsat case:

14. `https://archive.ics.uci.edu/ml/datasets/Pen-Based+Recognition+of+Handwritten+Digits`

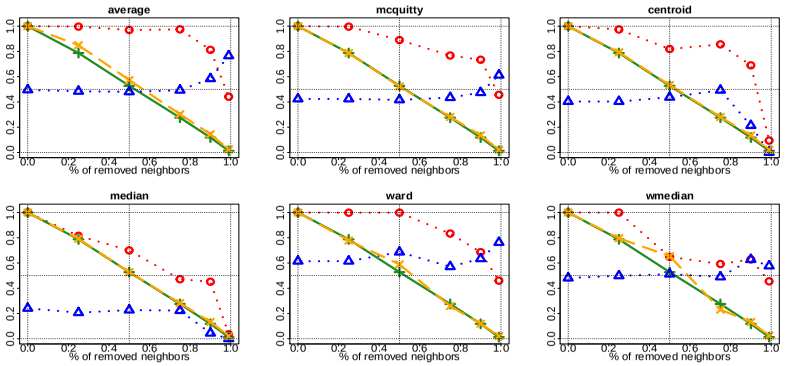


Figure 8: Results for the pendigits data set using a linear kernel. The x-axis corresponds to the % of removed neighbors. The y-axis corresponds to the observed values which all belong to $[0, 1]$. Solid lines with plus signs represent the relative memory use, dashed lines with cross signs show the relative running time, dotted lines with circles indicate the CC values, dotted lines with triangles give the ARI values.

- For all schemes the ARI curves are stable up to 75% of removed neighbors. In regard to scalability, this means that, for any technique, we can save up to ∼70% of memory usage and processing time without degrading the performances. Beyond 75% of removed neighbors, the clustering quality evolution depends on the method. For centroid and median it decreases whereas for the other approaches the ARI curves have a positive slope.

- For group average, Mcquitty and Ward, their respective best ARI scores are achieved with the sparsest $\mathbf{S}$ matrix. Precisely, if we keep only 10% of the nearest neighbors then, the ARI values observed for these techniques clearly outperform their respective baseline. Overall, the best gain and best ARI score is achieved by the group average with a clustering quality going from 0.495 to 0.765 which represents a 54% increase.

- For this data set as well, w-median is superior to median. Moreover, the ARI values we observe for the new method is pretty stable with respect to the level of sparsification.

Similarly to the landsat case, the similarity graph remained connected even with the strongest sparsification we applied.

7. Related Work and Discussion

Our approach is meant to be generic, scalable and effective with respect to challenging clustering tasks where objects belong to non-linear manifolds. Different types of previous

research works are relevant to our framework.

In order to face the inherent scalability issues of hierarchical clustering, several algorithms were introduced in the data mining community. BIRCH (Zhang et al., 1996) and CURE (Guha et al., 1998) are famous examples in that respect. These approaches use random sampling and/or a pre-clustering stage in order to reduce the number of elements to convey to the hierarchical clustering. These methods rely on a vectorial representation of the objects and use classic distances between points.

Carrying out an AHC approach on a sparse graph in the goal of speeding up the hierarchy construction was also studied in (Franti et al., 2006). In this work as well, objects are points in a vectorial space and weighted squared Euclidean distances serve as dissimilarity values. The authors use a directed k nearest neighbors graph. After each merge, the list of the k closest points to each centroid is approximately updated.

These research works provide efficient algorithms. However, they are not generic models of hierarchical clustering. In particular, they assume a feature based description of the items (stored data approach) and in this vectorial representation, the dissimilarities are all related to squared Euclidean distances. In our case, SNK-AHC relies on a generic model that allows designing different kinds of proximity relationships between clusters.

From this standpoint, our approach is in line with the works that were developed during the 1960's in the fields of statistics and data analysis. Overviews of these works can be found in (Gordon, 1987; Everitt et al., 2009; Mirkin, 1996; Murtagh and Contreras, 2012). The LW equation plays a core role in this landscape since it formally represents an infinite family of hierarchical clustering techniques. Furthermore, it makes it possible to algebraically study and define particular sub-families which satisfy different appealing conditions. The guarantee to output a monotonic dendrogram is an example of such conditions. In more general terms, these properties are named admissibility conditions (see for e.g. (Fisher and Van Ness, 1971; Chen and Van Ness, 1994, 1996; Mirkin, 1996)). More recently, Ackerman and Ben-David (2016) introduced other types of properties that characterize a class of linkage-based hierarchical clusterings.

In this context, several research papers have also addressed the scalability problems of D-AHC. We can highlight two research lines in that regard. The first one is essentially a matter of implementation and concerns the whole family of LW clusterings. In fact, by leveraging advanced data structure it is possible to speed-up D-AHC (Anderberg, 1973; Day and Edelsbrunner, 1984). By employing priority queues to efficiently store the nearest neighbors, the running time of the minimum search can be reduced. Maintaining the priority queues can also be done in an efficient fashion and overall, the time complexity of D-AHC can be reduced from cubic to a best-case $O(n^2)$ cost (Day and Edelsbrunner, 1984; Müllner et al., 2013).

In contrast to the previous research avenue, the second research line only involves a subfamily of AHC schemes. It is based on a property called reducibility which was stated by Bruynooghe (1978), and the resulting algorithm is usually named nearest neighbor chains. The reducibility condition is not satisfied by centroid and median for which the latter algorithm is not equivalent to D-AHC. For the other methods, it is an exact procedure that has a worst-case $O(n^2)$ time complexity instead of cubic. The nearest neighbor chains

procedure has been studied and implemented by several authors (de Rham, 1980; Juan, 1982; Benzécri, 1982; Murtagh, 1984; Müllner et al., 2013). More recently, Nguyen et al. (2014) has proposed a memory-efficient online hierarchical clustering called SparseHC which also relies on the reducibility property. However, only single, complete and group average linkages are considered.

These latter research works only focus on the computational efficiency of the usual D-AHC framework. SNK-AHC is able to effectively tackle a more general class of clustering problems.

Complex data such as texts, graphs, images and so on, do not necessarily lie on linear sub-spaces but rather on manifolds. Euclidean distances in the given descriptive space may fail to determine non-convex and arbitrary shapes. In order to better capture the underlying geometry of the data, other different approaches have been proposed in the literature.

In the context of non-parametric hierarchical clustering, one first group of papers, adopts a graph point of view of the clustering task. In particular, the nearest neighbors graph derived from the pairwise proximity values allows a better approximation of the natural geometries of groups of points. It is well-known that the single linkage leads to a chaining effect that is quite effective for arbitrary shapes detection as compared to other schemes. Gower and Ross (1969) pointed out the strong link between single linkage and the minimum spanning tree (MST) problem. Then Zahn (1971) analyzed more in details the application of the MST algorithm to the detection of groups that are non-convex and non-linearly separable. In this context, edge removal appears to be an effective mean to allow the MST to capture a large spectrum of shapes.

Other approaches use non-parametric proximity measures that rely on mutual nearest neighbors and rank-based linkages in order to recover arbitrary clusters shapes (Jarvis and Patrick, 1973; Gowda and Krishna, 1978). More recently, Balcan et al. (2014) uses common nearest neighbors and defines a two-step hierarchical clustering that is robust to outliers and which has interesting properties under some good neighborhood conditions.

Another related work in this context is the CHAMELEON algorithm introduced in (Karypis et al., 1999). It also proceeds in two stages and uses k nearest neighbors graphs. CHAMELEON presents another common point with SNK-AHC since it emphasizes the contrast between inter and intra connectivities of clusters. This so-called dynamic modeling is similar in spirit to our penalized similarities. However, the authors do not propose a generic model unlike our parametric recurrence equations (12a) and (12b).

Yet another research direction for manifold learning is through the use of kernel functions that map the data points from the original description space to a higher dimensional Hilbert space, called the feature space (see for e.g. (Lee and Verleysen, 2007)). It is expected that in the new space, the clusters are easier to detect. To our knowledge, only a few papers have extended D-AHC to kernels (Qin et al., 2003; Endo et al., 2004). In contrast to our model, these papers do not consider the scalability issues. Our previous work (Ah-Pine and Wang, 2016) also studies an inner-product based formulation of the LW equation. In addition, sparse kernel matrices are also employed. Nonetheless, this latter model is different from the framework we present in this paper. In particular, the concept of weighted penalized similarities and the theoretical results we provide are not examined in the framework introduced in (Ah-Pine and Wang, 2016).

Lastly, it is worth emphasizing the relationships between SNK-AHC and spectral clustering (Shi and Malik, 2000; Ng et al., 2001; Von Luxburg, 2007). In the latter family of techniques, kernel functions are employed to construct a similarity graph between objects. Then a sparsification method is applied and the Laplacian of the resulting graph is determined. Theoretical results from spectral graph theory (see for e.g. (Van Mieghem, 2010)) show the links between the eigen-decomposition of the Laplacian and the connected components of the graph. Spectral clustering is a two step procedure which performs a spectral embedding of the objects and subsequently applies a flat clustering method in the new space. The k-means algorithm is usually used in the second step. In this context, roughly speaking, we believe that SNK-AHC is to the classic D-AHC what spectral clustering is to the usual k-means: a significant extension of a conventional clustering method (a sub-family of LW clusterings in our case) which can recover groups of points with non-spherical shapes and which provide an interesting mean to guess the number of clusters. Besides, our approach has all the advantages that hierarchical clustering has over partitional clustering. In addition, since SNK-AHC is much more scalable than D-AHC, it does not have the major drawbacks of hierarchical clustering methods.

8. Conclusion and Future Work

We have introduced K-AHC a generic AHC model which relies on inner-products instead of squared Euclidean distances. Our approach is based on two recurrence formulas which embeds a sub-family of LW clustering techniques. In order to make our model efficient and effective for challenging clustering tasks, we apply K-AHC on a sparsified normalized kernel matrix. In that perspective, the two recurrence formulas highlight aggregation of inter-similarities on the one hand and of intra-similarities on the other hand. Our work can be viewed as a dynamic modeling of weighted penalized similarities of clusters. Moreover, by constraining the bottom-up merging procedure to only fuse pairs of clusters whose inter-similarity value is non-null, our method, SNK-AHC, not only is more scalable than the usual D-AHC, but it is also able to boost the clustering quality and to detect the number of clusters.

However, the performances that SNK-AHC can reach, depend on the way the similarity matrix is sparsified. Note that this is also the case for any method that relies on sparse similarity graphs such as spectral clustering. Therefore, one important future line of research is the study and design of more advanced sparsification techniques. From a clustering quality standpoint, the setting of the sparsification method is an important question to address in practice since it determines the connected components of the SNK matrix and thus the number of clusters our approach will recover. In order to investigate this problem from a theoretical point of view, the cluster tree framework introduced in (Chaudhuri and Dasgupta, 2010) and (Balakrishnan et al., 2013) could be of interest. Regarding the overall complexity of Algorithm 4, techniques that make it possible to exactly or approximately determine nearest neighbors graphs in an efficient manner are important to look at. Indeed, even though the dendrogram building procedure performed by SNK-AHC can be carried out efficiently, the basic computational cost for determining the k nearest neighbors graph remains quadratic and can be a bottleneck in practice.

Still, it is interesting to mention, that, as far as the tree building procedure is concerned, there are already pretty immediate ways to further improve the scalability of our approach. As underlined in the previous section, two directions could be considered. Firstly, we can enhance the complexity of SNK-AHC by using priority queues. Secondly, we can use the nearest neighbor chains approach. In that regard, note that the best performances we observed in our experiments concern schemes that satisfy the reducibility condition.

Finally, our model is generic but we have demonstrated that not all parameters settings are worth considering. From this standpoint, it is interesting to examine how other admissibility conditions are expressed in our framework. In that perspective, a new property which is peculiar to our work is the diagonal translation invariance. We proved that group average, Mcquitty and Ward satisfy this condition. These techniques are among the most effective ones from the experimental results we reported. Accordingly, a characterization of this sub-family of clustering techniques would be beneficial.

Acknowledgments

The author would like to thank the anonymous reviewers for their valuable comments.

References

Margareta Ackerman and Shai Ben-David. A characterization of linkage-based hierarchical clustering. *Journal of Machine Learning Research*, 17:1–17, 2016.

Julien Ah-Pine. Normalized kernels as similarity indices. In *Advances in Knowledge Discovery and Data Mining, 14th Pacific-Asia Conference, PAKDD 2010, Hyderabad, India, June 21-24, 2010. Proceedings. Part II*, pages 362–373, 2010.

Julien Ah-Pine and Xinyu Wang. Similarity based hierarchical clustering with an application to text collections. In *International Symposium on Intelligent Data Analysis*, pages 320–331. Springer, 2016.

Michael R Anderberg. *Cluster analysis for applications*. Academic Press, New York, 1973.

Sivaraman Balakrishnan, Srivatsan Narayanan, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Cluster trees on manifolds. In *Advances in Neural Information Processing Systems*, pages 2679–2687, 2013.

Maria-Florina Balcan, Yingyu Liang, and Pramod Gupta. Robust hierarchical clustering. *The Journal of Machine Learning Research*, 15(1):3831–3871, 2014.

Jean-Paul Benzécri. Construction d'une classification ascendante hiérarchique par la recherche en chaîne des voisins réciproques. *Les cahiers de l'analyse des données*, 7 (2):209–219, 1982.

M. Bruynooghe. Classification ascendante hiérarchique des grands ensembles de données : un algorithme rapide fondé sur la construction des voisinages réductibles. *Cahiers de l'analyse des données*, 3(1):7–33, 1978. URL `http://eudml.org/doc/87905`.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):27, 2011.

Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for the cluster tree. In *Advances in Neural Information Processing Systems*, pages 343–351, 2010.

Zhenmin Chen and John W. Van Ness. Space-contracting, space-dilating, and positive admissible clustering algorithms. *Pattern recognition*, 27(6):853–857, 1994.

Zhenmin Chen and John W Van Ness. Space-conserving agglomerative algorithms. *Journal of classification*, 13(1):157–168, 1996.

Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009. ISBN 0262033844, 9780262033848.

William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

C. de Rham. La classification hiérarchique ascendante selon la méthode des voisins réciproques. *Les cahiers de l'analyse des données*, 5(2):135–144, 1980.

Daniel Defays. An efficient algorithm for a complete link method. *The Computer Journal*, 20(4):364–366, 1977.

Yasunori Endo, Hideyuki Haruyama, and Takayoshi Okubo. On some hierarchical clustering algorithms using kernel functions. In *IEEE International Conference on Fuzzy Systems, FUZZ-IEEE 2004, Budapest, Hungary, July 25-29, 2004.*, pages 1513–1518, 2004.

Brian S. Everitt, Sabine Landau, and Morven Leese. *Cluster Analysis*. Wiley Publishing, 4th edition, 2009. ISBN 0340761199, 9780340761199.

Lloyd Fisher and John W. Van Ness. Admissible clustering procedures. *Biometrika*, 58(1): 91–104, 1971.

Pasi Franti and et al. Clustering datasets, 2015. URL `http://cs.uef.fi/sipu/datasets/`.

Pasi Franti, Olli Virmajoki, and Ville Hautamaki. Fast agglomerative clustering using a k-nearest neighbor graph. *IEEE Trans. Pattern Anal. Mach. Intell.*, 28(11):1875–1881, November 2006. ISSN 0162-8828. URL `http://dx.doi.org/10.1109/TPAMI.2006.227`.

Aristides Gionis, Heikki Mannila, and Panayiotis Tsaparas. Clustering aggregation. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):4, 2007.

Allan D Gordon. A review of hierarchical classification. *Journal of the Royal Statistical Society. Series A (General)*, pages 119–137, 1987.

K. Chidananda Gowda and G. Krishna. Agglomerative clustering using the concept of mutual nearest neighbourhood. *Pattern recognition*, 10(2):105–112, 1978.

John C. Gower and Gavin J.S. Ross. Minimum spanning trees and single linkage cluster analysis. *Applied statistics*, pages 54–64, 1969.

Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim. Cure: an efficient clustering algorithm for large databases. In *ACM Sigmod Record*, volume 27, pages 73–84. ACM, 1998.

Roger A. Horn and Charles R. Johnson, editors. *Matrix Analysis*. Cambridge University Press, New York, NY, USA, 1986. ISBN 0-521-30586-1.

Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of Classification*, 2 (1):193–218, 1985. ISSN 1432-1343. doi: 10.1007/BF01908075. URL `http://dx.doi.org/10.1007/BF01908075`.

Raymond Austin Jarvis and Edward A Patrick. Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on computers*, 100(11):1025–1034, 1973.

J. Juan. Programme de classification hiérarchique par l'algorithme de la recherche en chaîne des voisins réciproques. *Les cahiers de l'analyse des données*, 7(2):219–225, 1982.

George Karypis, Eui-Hong Han, and Vipin Kumar. Chameleon: Hierarchical clustering using dynamic modeling. *Computer*, 32(8):68–75, 1999.

Godfrey N. Lance and Williams T. Williams. A general theory of classificatory sorting strategies: 1. hierarchical systems. *The Computer Journal*, 9(4):373–380, 1967.

John A Lee and Michel Verleysen. *Nonlinear dimensionality reduction*. Springer Science & Business Media, 2007.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Glenn W Milligan. Ultrametric hierarchical clustering algorithms. *Psychometrika*, 44(3): 343–346, 1979.

Boris Mirkin. *Mathematical Classification and Clustering*. Kluwer Academic Publishers, London, 1996.

Daniel Müllner. Modern hierarchical, agglomerative clustering algorithms. *CoRR*, abs/1109.2378, 2011. URL `http://arxiv.org/abs/1109.2378`.

Daniel Müllner et al. fastcluster: Fast hierarchical, agglomerative clustering routines for r and python. *Journal of Statistical Software*, 53(9):1–18, 2013.

Fionn Murtagh. Complexities of hierarchic clustering algorithms: state of the art. *Computational Statistics Quarterly*, 1(2):101–113, 1984.

Fionn Murtagh and Pedro Contreras. Algorithms for hierarchical clustering: an overview. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 2(1):86–97, 2012.

Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. In *NIPS*, volume 14, pages 849–856, 2001.

Thuy-Diem Nguyen, Bertil Schmidt, and Chee-Keong Kwoh. SparseHC: a memory-efficient online hierarchical clustering algorithm. *Procedia Computer Science*, 29:8–19, 2014.

Jie Qin, Darrin P Lewis, and William Stafford Noble. Kernel hierarchical gene clustering from microarray expression data. *Bioinformatics*, 19(16):2097–2104, 2003.

Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2001.

Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 22(8):888–905, 2000.

Robin Sibson. SLINK: an optimally efficient algorithm for the single-link cluster method. *Comput. J.*, 16(1):30–34, 1973. doi: 10.1093/comjnl/16.1.30. URL `http://dx.doi.org/10.1093/comjnl/16.1.30`.

Piet Van Mieghem. *Graph spectra for complex networks*. Cambridge University Press, 2010.

Ulrike Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4): 395–416, 2007.

Charles T. Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *IEEE Transactions on computers*, 100(1):68–86, 1971.

Tian Zhang, Raghu Ramakrishnan, and Miron Livny. Birch: an efficient data clustering method for very large databases. In *ACM Sigmod Record*, volume 25, pages 103–114. ACM, 1996.

Journal of Machine Learning Research 19 (2018) 1-50 Submitted 1/14; Revised 7/18; Published 9/18

Markov Blanket and Markov Boundary of Multiple Variables

Xu-Qing Liu LIUXUQING688@163.COM
State Key Laboratory of Mechanics and Control of Mechanical Structures
Institute of Nano Science and Department of Mathematics
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
Faculty of Mathematics and Physics
Huaiyin Institute of Technology, Huai'an 223003, China

Xin-Sheng Liu* XSLIU@NUAA.EDU.CN
State Key Laboratory of Mechanics and Control of Mechanical Structures
Institute of Nano Science and Department of Mathematics
Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Editors: Marina Meila; Kevin Murphy; Joris Mooij

Abstract

Markov blanket (Mb) and Markov boundary (MB) are two key concepts in Bayesian networks (BNs). In this paper, we study the problem of Mb and MB for multiple variables. First, we show that Mb possesses the additivity property under the local intersection assumption, that is, an Mb of multiple targets can be constructed by simply taking the union of Mbs of the individual targets and removing the targets themselves. MB is also proven to have additivity under the local intersection assumption. Second, we analyze the cases of violating additivity of Mb and MB and then put forward the notions of Markov blanket supplementary (MbS) and Markov boundary supplementary (MBS). The properties of MbS and MBS are studied in detail. Third, we build two MB discovery algorithms and prove their correctness under the local composition assumption. We also discuss the ways of practically doing conditional independence tests and analyze the complexities of the algorithms. Finally, we make a benchmarking study based on six synthetic BNs and then apply MB discovery to multi-class prediction based on a real data set. The experimental results reveal our algorithms have higher accuracies and lower complexities than existing algorithms.

Keywords: Markov blanket, Markov boundary, Markov blanket supplementary, Markov boundary supplementary, Bayesian network

1. Introduction

Bayesian networks (BNs) are graphical structures used to represent the probabilistic relations among a large number of variables and to make the associated probabilistic inferences (Neapolitan, 2004; Pearl, 1988). In recent years, BNs have become one of the most powerful tools in encoding uncertain expert knowledge in expert systems (Daly et al., 2011; Parviainen and Koivisto, 2013) and also deeply influenced on many other actual domains such as medical diagnosis, financial analysis, bioinformatics, and industrial applications (Zhang and Guo, 2006).

*. Corresponding Author.

As two important concepts in BNs, Markov blanket (Mb) and Markov boundary (MB) play a key role in feature selection (FS; Fu and Desmarais, 2010; Pellet and Elisseeff, 2008; Aliferis et al., 2010a,b). Mathematically, Pearl (1988, pp. 218–221) showed the conditional probability for the target given other variables can be replaced by the MB as the conditional set. Pellet and Elisseeff (2008, pp. 1299, 1302) proved that an MB is the theoretically optimal set of features. Further, under certain assumptions about the learner and the loss function, MB is the solution to the variable selection problem (Tsamardinos and Aliferis, 2003; Statnikov et al., 2013).

So far most authors have focused on the problem of Mb or MB for a single variable. In this paper, we consider the problem of Mb and MB for multiple variables. This occurs if, for example, one wants to compute the joint probability of two or more variables conditioned on all other variables. The basic question for Mb of multiple variables is whether the additivity property holds, that is, can an Mb of multiple variables be constructed by simply taking the union of the Mbs of the individual variables and removing the target variables themselves? The same question is for MB. Further, if the additivity property is violated in some situation, how can we do it?

In the literature, there have been lots of MB discovery algorithms, such as the Koller-Sahami (`KS`) algorithm (Koller and Sahami, 1996), the grow-shrink (`GS`) algorithm (Margaritis and Thrun, 1999, 2000), the incremental association Markov boundary (`IAMB`) algorithm (Tsamardinos et al., 2003) and its several variants, the `HITON` algorithm (Aliferis et al., 2003), the max-min Markov boundary (`MMMB`) algorithm (Tsamardinos et al., 2006), the parents and children based Markov boundary (`PCMB`) algorithm and `KIAMB` algorithms (Peña et al., 2007), the `BFMB` algorithm (Fu and Desmarais, 2007), the algorithmic framework called generalized local learning (`GLL`, Aliferis et al., 2010a), and some others (Fu and Desmarais, 2010; Schlüter, 2014). For a single target variable, most of these algorithms are efficient to seek an approximate MB; for multiple target variables, if simply regarding them as a multivariate variable, these algorithms seem to be feasible. However, this will lead to low accuracies and high computational complexities. Hence, it is necessary to design more efficient MB discovery algorithms for multiple variables.

The remainder of this paper is organized as follows. Section 2 presents necessary preliminaries and the motivations of this paper. Subsection 3.1 shows additivity of Mb and MB under the local intersection assumption. In Subsection 3.2, we first analyze when additivity is violated and then put forward the notions of Markov blanket supplementary (MbS) and Markov boundary supplementary (MBS). The properties of MbS and MBS are studied detailedly. In Section 4, we design two MB discovery algorithms for multiple variables, and prove their correctness under the local composition assumption. In addition, we discuss the ways of practically doing conditional independence (CI) tests and analyze the complexities of the algorithms. Section 5 makes a benchmarking study based on six synthetic BNs, and Section 6 considers a practical application. The experimental results show the superiority of our algorithms with higher accuracies and lower complexities than existing algorithms. Section 7 concludes this paper and presents three remarks.

2. Preliminaries and Motivations

In the paper, we denote a variable and its value by upper-case and lower-case letters in italics (e.g., X, x), a set of variables and its value by upper-case and lower-case bold letters in italics (e.g., $\boldsymbol{X}$, $\boldsymbol{x}$). The difference between $\boldsymbol{X}$ and $\boldsymbol{Y}$ is denoted by $\boldsymbol{X} \setminus \boldsymbol{Y}$. For brevity, we write $(\boldsymbol{X} \setminus \boldsymbol{Y}) \setminus \boldsymbol{Z}$ as $\boldsymbol{X} \setminus \boldsymbol{Y} \setminus \boldsymbol{Z}$. In addition, we use $|\boldsymbol{X}|$ to denote the number of variables involved in $\boldsymbol{X}$.

2.1 Preliminaries

Suppose we have a joint probability distribution $\mathbb{P}$ over $\boldsymbol{V} \triangleq \{X_1, \cdots, X_p\}$ and a directed acyclic graph (DAG) $\mathbb{G}$ with the variables in $\boldsymbol{V}$ as its nodes. We say $(\mathbb{G}, \mathbb{P})$ satisfies the Markov condition if every $X \in \boldsymbol{V}$ is conditionally independent of its nondescendants given its parents; Further, $(\mathbb{G}, \mathbb{P})$ is called a Bayesian network (BN) if it satisfies the Markov condition; Furthermore, $(\mathbb{G}, \mathbb{P})$ satisfies the faithfulness condition if, based on the Markov condition, $\mathbb{G}$ entails all and only conditional independences (CIs) in $\mathbb{P}$ (Pearl, 1988; Neapolitan, 2004).

We write $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ ($\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$), if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are conditionally independent (dependent) given $\boldsymbol{Z}$ with respect to $\mathbb{P}$. The following properties describe the relations among CI statements (Pearl, 1988; Peña et al., 2007; Statnikov et al., 2013). For any $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}, \boldsymbol{W} \subseteq \boldsymbol{V}$, we have (i) *symmetry*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ is equivalent to $\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X} \,|\, \boldsymbol{Z}$; (ii) *decomposition*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \,|\, \boldsymbol{Z}$ implies $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \,|\, \boldsymbol{Z}$; (iii) *weak union*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \,|\, \boldsymbol{Z}$ implies $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z} \cup \boldsymbol{W}$; (iv) *contraction*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z} \cup \boldsymbol{W}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \,|\, \boldsymbol{Z}$ imply $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \,|\, \boldsymbol{Z}$; (v) *self-conditioning*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Y} \cup \boldsymbol{Z}$. Further, if $\mathbb{P}$ is strictly positive, then besides (i)∼(v) we also have (vi) *intersection*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z} \cup \boldsymbol{W}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \,|\, \boldsymbol{Z} \cup \boldsymbol{Y}$ imply $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \,|\, \boldsymbol{Z}$. Furthermore, if $\mathbb{P}$ is faithful to a DAG $\mathbb{G}$, then besides (i)∼(vi) we also have (vii) *composition*: $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ and $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{W} \,|\, \boldsymbol{Z}$ imply $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \cup \boldsymbol{W} \,|\, \boldsymbol{Z}$.

Among these properties, intersection and composition are two global ones. Statnikov et al. (2013, p. 504) provided a relaxed version for composition called *local composition*: one says $\boldsymbol{T} \subseteq \boldsymbol{V}$ satisfies the local composition property, if $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \,|\, \boldsymbol{Z}$ and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ imply $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \cup \boldsymbol{Y} \,|\, \boldsymbol{Z}$ for any $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \boldsymbol{T}$. We will provide a relaxed version for the intersection property.

Conditional mutual information (CMI) is one of the basic tools for testing CIs. Denote the CMI between $\boldsymbol{X}$ and $\boldsymbol{Y}$ conditioned on $\boldsymbol{Z}$ by $\mathbb{I}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z})$. Then $\mathbb{I}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \geqslant 0$, with equality holding if and only if $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ (Zhang and Guo, 2006). For a practical problem, we cannot access to the true CMI; instead, we use its empirical estimate, denoted by $\mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z})$, based on the data $\boldsymbol{D}$ (Cheng et al., 2002). Note that $\mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z}) \geqslant 0$ also holds for any $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$.

The *chain rule* for CMI (Cover and Thomas, 2006) is useful to prove the main results of this paper: $\mathbb{I}(\boldsymbol{X}; \boldsymbol{Y}_1 \cup \boldsymbol{Y}_2 \,|\, \boldsymbol{Z}) = \mathbb{I}(\boldsymbol{X}; \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}) + \mathbb{I}(\boldsymbol{X}; \boldsymbol{Y}_2 \,|\, \boldsymbol{Z} \cup \boldsymbol{Y}_1)$ holds for any four sets of variables $\boldsymbol{X}$, $\boldsymbol{Y}_1$, $\boldsymbol{Y}_2$, and $\boldsymbol{Z}$ from $\boldsymbol{V}$.

Another notion closely related to CI is d-separation (Pearl, 1988, p. 117). For a DAG $\mathbb{G}$ over $\boldsymbol{V}$, letting $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V}$ be disjoint, we say $\boldsymbol{Z}$ d-separates $\boldsymbol{X}$ and $\boldsymbol{Y}$ if it blocks every path between $\boldsymbol{X}$ and $\boldsymbol{Y}$, and if this is the case we write $\boldsymbol{X} \perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$. Here, $\boldsymbol{Z}$ blocking a path $\mathbb{c}$ means that $\mathbb{c}$ has a head-to-tail node or a tail-to-tail node belonging to $\boldsymbol{Z}$, or that $\mathbb{c}$ has a head-to-head node C such that C and its all descendants are not in $\boldsymbol{Z}$. As well known, $\boldsymbol{X} \perp \boldsymbol{Y} \,|\, \boldsymbol{Z} \Rightarrow \boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$, if $(\mathbb{G}, \mathbb{P})$ is a BN (Neapolitan, 2004, p. 74). This implication provides a convenient way of identifying CIs.

For example, consider a BN with the graph presented in Figure 1 as its DAG. It follows that: X_2 and X_8 are d-separated by $\{X_4, X_5\}$, meaning $X_2 \perp X_8 \,|\, \{X_4, X_5\}$ and thus $X_2 \perp\!\!\!\perp X_8 \,|\, \{X_4, X_5\}$; X_3 and X_4 are d-separated by $\emptyset$, meaning $X_3 \perp X_4$, so $X_3 \perp\!\!\!\perp X_4$. Note that these two probabilistic CIs can not be directly derived from the Markov condition.

In what follows, the concepts of Mb and MB are presented. They are a direct extension of Mb and MB for a single target variable (Pearl, 1988, p. 97; Neapolitan, 2004, pp. 108–109): an Mb of $\boldsymbol{T}$ is a set of variables shielding $\boldsymbol{T}$ from all other variables, so it carries all information of $\boldsymbol{T}$ that cannot be obtained from other variables, while an MB is a minimal Mb.

Definition 1 *Let $\boldsymbol{T} \subseteq \boldsymbol{V}$ and $\boldsymbol{M} \subseteq \boldsymbol{V} \setminus \boldsymbol{T}$. We call $\boldsymbol{M}$ a Markov blanket (Mb) of $\boldsymbol{T}$ if $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T} \,|\, \boldsymbol{M}$. Further, a Markov boundary (MB) of $\boldsymbol{T}$ is any Mb such that none of its proper subsets is an Mb.* ■

When $|\boldsymbol{T}| = 1$, the following results are well known in the literature (Pearl, 1988; Neapolitan, 2004; Statnikov et al., 2013): (a) if $(\mathbb{G}, \mathbb{P})$ is a BN, then for $T \in \boldsymbol{V}$ the set of its all parents, children, and spouses is an Mb of T (denoted by $\boldsymbol{M}_T$); (b) if $\mathbb{P}$ satisfies the intersection property, then T has a unique MB; (c) if $(\mathbb{G}, \mathbb{P})$ satisfies the faithfulness condition, then $\boldsymbol{M}_T$ is the unique MB of T.

Consider again the BN with the graph presented in Figure 1 as its DAG. In this BN, it is seen that $\boldsymbol{M}_{X_4} \triangleq \{X_2, X_6, X_3\}$ is an Mb of X_4; further, $\boldsymbol{M}_{X_4}$ is the unique MB of X_4 if the faithfulness condition is satisfied. Similarly, $\boldsymbol{M}_{X_2} \triangleq \{X_4, X_5\}$ is the unique MB of X_2 under the faithfulness condition.

The above result (b) points out that if the uniqueness of MB is violated, then the intersection property must be violated. Lemeire (2007) provided a case of violating intersection called *information equivalence*: $\boldsymbol{X}$ and $\boldsymbol{Y}$ are called information equivalent with respect to $\boldsymbol{T}$ if $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X}$, $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{Y}$, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \,|\, \boldsymbol{Y}$, and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{X}$. A related notion is *conditional information equivalence* (Lemeire et al., 2012; Statnikov et al., 2013): $\boldsymbol{X}$ and $\boldsymbol{Y}$ are called to be conditionally information equivalent with respect to $\boldsymbol{T}$ given $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \boldsymbol{X} \setminus \boldsymbol{Y} \setminus \boldsymbol{T}$, if $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X} \,|\, \boldsymbol{Z}$, $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \,|\, \boldsymbol{Y} \cup \boldsymbol{Z}$, and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{X} \cup \boldsymbol{Z}$. Lemeire et al. (2012, pp. 1309–1311) showed that (conditional) information equivalence is one of the two major cases in which *adjacency faithfulness* is violated. Here, the adjacency faithfulness condition (Ramsey et al., 2006; Lemeire et al., 2012) is defined as: if X and Y are adjacent, then $X \not\perp\!\!\!\perp Y \,|\, \boldsymbol{Z}$ for any $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \{X, Y\}$. Statnikov et al. (2013, p. 503) provided a local version for adjacency faithfulness by focusing on a specific variable.

Here, we employ the information flow metaphor (Cheng et al., 2002) to intuitively explain information equivalence: we can view a BN as a network of information channels, where each node is a *valve* that is either active or inactive; the valves are connected by *information channels*; information can flow through an active valve but not an inactive one; instantiating a node means this valve becomes inactive. We extend this metaphor by viewing a clique of one or more nodes as a valve. In this sense, a CI relation $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ means all the channels between $\boldsymbol{X}$ and $\boldsymbol{Y}$ are cut off by $\boldsymbol{Z}$ and thus the information between $\boldsymbol{X}$ and $\boldsymbol{Y}$ can not flow once $\boldsymbol{Z}$ becomes inactive. When information equivalence occurs, we further extend this information flow metaphor as follows: if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$ given $\boldsymbol{Z}$, then there exists an *information equivalent valve*, denoted by $\delta_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{T}\,|\,\boldsymbol{Z}}$, which connects $\boldsymbol{T}$ and $\boldsymbol{X}$ and connects $\boldsymbol{T}$ and $\boldsymbol{Y}$; *$\delta_{\boldsymbol{X},\boldsymbol{Y};\boldsymbol{T}\,|\,\boldsymbol{Z}}$ is active when and only when both $\boldsymbol{X}$ and $\boldsymbol{Y}$ are active.* Then, the relation "$\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$ given $\boldsymbol{Z}$" can be presented in Figure 2.

Information equivalence represents all the possible situations leading to the nonuniqueness of MB. In fact, we can show the following result, which indicates that violating the uniqueness of MB implies the presence of information equivalence. The proof is presented in Section B.

Lemma 1 *The intersection property holds if and only if no information equivalence occurs.* ■

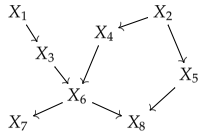


Figure 1: A simple DAG used to illustrate d-separation.

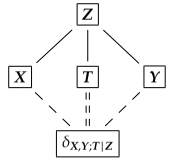


Figure 2: An intuitive illustration for information equivalence.

By analyzing the proof of Lemma 1, we present a relaxed version for the intersection property called *local intersection* as follows.

Definition 2 (Local Intersection) *Letting $\boldsymbol{T} \subseteq \boldsymbol{V}$, we say $\boldsymbol{T}$ satisfies the local intersection property, if the following two types of local conditions hold simultaneously:* (i) *type-I local condition: in the case of $|\boldsymbol{T}| \geqslant 2$, for any disjoint $\boldsymbol{T}_1, \boldsymbol{T}_2 \subseteq \boldsymbol{T}$, there are no disjoint $\boldsymbol{X}, \boldsymbol{Y} \subseteq \boldsymbol{V} \setminus \boldsymbol{T}$ such that $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ are information equivalent with respect to $\boldsymbol{X}$ conditioned on $\boldsymbol{Y}$; and* (ii) *type-II local condition: there are no disjoint $\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \boldsymbol{T}$ such that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$ conditioned on $\boldsymbol{Z}$.* ■

Clearly, *intersection* implies *local intersection* but not vice versa, because the former requires no any information equivalence while the latter only requires no information equivalence between the targets and the remaining variables. Here, we give a lemma concerning the uniqueness of MB under the local intersection assumption. The proof is presented in Appendix B.

Lemma 2 *For $\boldsymbol{T} \subseteq \boldsymbol{V}$, assume the type-II local condition defined in Definition 2 holds. Then $\boldsymbol{T}$ has a unique MB.* ■

To facilitate the identification of information equivalence, Lemeire (2007) ever introduced the notions of target partition (T-partition) and equivalent partition (E-partition), and then provided a relation among information equivalence, T-partition, and E-partition.

- T-partition: The domain, $\boldsymbol{X}_{\text{dom}}$, of $\boldsymbol{X}$ can be partitioned into disjoint subsets $\boldsymbol{X}_{\text{dom}}^{(k)}$ for which $\mathbb{P}(\boldsymbol{T}\,|\,\boldsymbol{x})$ is the same for all $\boldsymbol{x} \in \boldsymbol{X}_{\text{dom}}^{(k)}$. This is called the T-partition of $\boldsymbol{X}_{\text{dom}}$ with respect to $\boldsymbol{T}$.
- E-partition: A relation $\Re \subset \boldsymbol{X} \otimes \boldsymbol{Y}$ defines an E-partition in $\boldsymbol{Y}_{\text{dom}}$ to a partition of $\boldsymbol{X}_{\text{dom}}$, if: (i) $\neg(\boldsymbol{x}_2 \Re \boldsymbol{y}_1)$ holds for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{X}_{\text{dom}}$ belonging to different partitions and for any $\boldsymbol{y}_1 \in \boldsymbol{Y}_{\text{dom}}$ with $\boldsymbol{x}_1 \Re \boldsymbol{y}_1$; and (ii) for every $\boldsymbol{X}_{\text{dom}}^{(k)}$, there exist $\boldsymbol{x}_1 \in \boldsymbol{X}_{\text{dom}}^{(k)}$ and $y_1 \in \boldsymbol{Y}_{\text{dom}}$ such that $\boldsymbol{x}_1 \Re \boldsymbol{y}_1$.
- Relation among *information equivalence*, *T-partition*, and *E-partition*: If $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X}$ and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y}\,|\,\boldsymbol{X}$, then $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X}\,|\,\boldsymbol{Y}$ (meaning $\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$) if and only if the relation $\boldsymbol{x} \Re \boldsymbol{y}$ defined by $\mathbb{P}(\boldsymbol{x}, \boldsymbol{y}) > 0$ with $\boldsymbol{x} \in \boldsymbol{X}_{\text{dom}}$ and $y \in \boldsymbol{Y}_{\text{dom}}$ defines an E-partition in $\boldsymbol{Y}_{\text{dom}}$ to the T-partition of $\boldsymbol{X}_{\text{dom}}$ with respect to $\boldsymbol{T}$.

The graph shown in Figure 3, originally presented by Statnikov and Aliferis (2010), makes an intuitive illustration on T-partition and E-partition. As seen, $\{1, 2\}$ and $\{3\}$ constitute the T-partition of $A_{\text{dom}} \triangleq \{1, 2, 3\}$ with respect to C; $\{1, 2\}$ and $\{3\}$ are the E-partition of $B_{\text{dom}} \triangleq \{1, 2, 3\}$ to the T-partition of A_{dom}, Therefore, A and B are information equivalent with respect to C if $C \not\perp\!\!\!\perp A$, since $C \perp\!\!\!\perp B\,|\,A$ holds inherently because of the Markov condition.

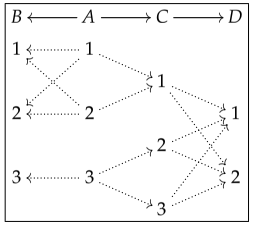


Figure 3: An illustration on T-partition and E-partition: in the DAG "$B \leftarrow A \rightarrow C \rightarrow D$", all variables take $\{1, 2, 3\}$ except for D taking $\{1, 2\}$, and dotted arrows denote all non-zero conditional probabilities of each variable given its parents.

Finally, the notion of *context-independent information equivalence* given by Statnikov et al. (2013) will be used in Example 2. $\boldsymbol{X}$ and $\boldsymbol{Y}$ are called context-independent information equivalent with respect to $\boldsymbol{T}$, if $\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$ given any $\boldsymbol{Z} \subseteq \boldsymbol{V} \setminus \boldsymbol{X} \setminus \boldsymbol{Y} \setminus \boldsymbol{T}$. For this notion, Statnikov et al. (2013) proved the following conclusion: if $\boldsymbol{M}$ is an Mb of $\boldsymbol{T}$ with $\boldsymbol{X} \subseteq \boldsymbol{M}$, and there is some $\boldsymbol{Y} \subseteq \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T}$ such that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are context-independent information equivalent with respect to $\boldsymbol{T}$, then $(\boldsymbol{M} \setminus \boldsymbol{X}) \cup \boldsymbol{Y}$ is also an Mb of $\boldsymbol{T}$.

2.2 Two Typical Algorithms: `IAMB` and `KIAMB`

This subsection concisely presents two typical MB discovery algorithms: `IAMB` (Tsamardinos et al., 2003) and `KIAMB` (Peña et al., 2007). We select them because of their high adaptability and time efficiency: (i) correctness of `IAMB` and `KIAMB` requires only the local composition assumption (Statnikov et al., 2013), while the correctness of the parents and children based algorithms, such as `PCMB` and the algorithms in the `GLL` framework, usually requires the faithfulness condition (Peña et al., 2007, Theorem 6; and Aliferis et al., 2010a, Theorem 1); (ii) `IAMB` and `KIAMB` are time efficient and thus suitable for the problem of MB for multiple variables, while the parents and children based algorithms have exponential complexities (Aliferis et al., 2010a, pp. 199–200), so they are hard to work when too many variables are involved, such as the problem of MB discovery for multiple variables.

`IAMB` is an enhanced variant of `GS`. In 2003, Tsamardinos et al. pointed out that `GS` uses a static and potentially inefficient heuristic in the growing phase, and then proposed `IAMB` by employing a dynamic heuristic. Tsamardinos et al. (2003) showed the correctness of `IAMB` under the faithfulness condition; Peña et al. (2007) relaxed the condition to the composition assumption; Statnikov et al. (2013) further relaxed the condition to the local composition assumption. Algorithm 3 describes the pseudo code for `IAMB`. See Appendix A for details.

In the algorithm, there is a function $f_{\boldsymbol{D}}$ (Line 3 of `IAMB` in Algorithm 3) denoting a heuristic used to measure the association between variables (Tsamardinos et al., 2003; Peña et al., 2007). Two widely used selections for $f_{\boldsymbol{D}}$ are CMI (Cheng et al., 2002; Tsamardinos et al., 2003) and the negative p-value (Tsamardinos et al., 2006; Aliferis et al., 2010a,b; Statnikov et al., 2013). Also, Yaramakala (2004, p. 41) suggested an equivalent version of the negative p-value. Subsection 4.3 will make a discussion about the ways of practically doing CI tests and the selections for $f_{\boldsymbol{D}}$.

KIAMB is a stochastic extension of IAMB. It embeds a randomization parameter $K \in [0, 1]$ which specifies the trade-off between greediness and randomness. If taking $K = 1$, KIAMB reduces to IAMB. Peña et al. (2007) proved the correctness of KIAMB under the composition assumption. By the proof, the local composition assumption is sufficient for KIAMB to be correct. Algorithm 3 describes the pseudo code for KIAMB.

For the case of $|\boldsymbol{T}| \geqslant 2$, IAMB and KIAMB can remain correct if strengthening the precondition. We present the correctness of them as follows, without presenting the proof since it is similar to that of the original IAMB and KIAMB (Tsamardinos et al., 2003; Peña et al., 2007; Statnikov et al., 2013). In what follows, we say a CI test for a hypothesis is correct if the statistical decision is correctly made by using a testing method. Subsection 4.3 gives a further discussion on this issue.

Theorem 1 (Correctness of IAMB and KIAMB) *Assume $\boldsymbol{T}$ satisfies the local composition property, and all CI tests are correct. Then* (i) *IAMB outputs an MB of $\boldsymbol{T}$;* (ii) *KIAMB outputs an MB of $\boldsymbol{T}$ for any $K \in [0, 1]$.* ■

2.3 Motivations

This subsection provides three motivations of this paper.

Let $\boldsymbol{M}$ be an MB of T. Then $\mathbb{P}(T \,|\, \boldsymbol{V} \setminus \{T\}) = \mathbb{P}(T \,|\, \boldsymbol{M})$. In other words, all information for predicting T is carried by $\boldsymbol{M}$. Further, $\boldsymbol{M}$ is a solution to the FS problem, if the algorithm that constructs the prediction model can learn any probability distribution, and the performance metric is strictly decreasing with the mean-squared loss with a preference for smaller subsets (Tsamardinos and Aliferis, 2003, Proposition 3). For this reason, MB for a single variable is sufficient.

However, there are the situations where MB for multiple variables is preferred. This occurs if we need the probability distribution of more than one variables given all the others. Let $\boldsymbol{M}_i$ be an MB of T_i for $i = 1, 2$. Denoting $\boldsymbol{T} = \{T_1, T_2\}$, it follows that

$$\mathbb{P}(\boldsymbol{T} \,|\, \boldsymbol{V} \setminus \boldsymbol{T}) = \begin{cases} \mathbb{P}(T_1 \,|\, \boldsymbol{M}_1)\mathbb{P}(T_2 \,|\, \boldsymbol{M}_2) & \text{if } T_1 \notin \boldsymbol{M}_2 \text{ or } T_2 \notin \boldsymbol{M}_1 \\ \mathbb{P}(T_1, T_2, \boldsymbol{V} \setminus \boldsymbol{T}) \,/\, \sum_{t_1, t_2} \mathbb{P}(t_1, t_2, \boldsymbol{V} \setminus \boldsymbol{T}) & \text{if } T_1 \in \boldsymbol{M}_2 \text{ and } T_2 \in \boldsymbol{M}_1 \end{cases}$$

As seen, in the case of $T_1 \in \boldsymbol{M}_2$ and $T_2 \in \boldsymbol{M}_1$, the computation is intractable, especially when the dimension is high. Nevertheless, if we have an MB for $\boldsymbol{T}$, denoted by $\boldsymbol{M}$, then $\mathbb{P}(\boldsymbol{T} \,|\, \boldsymbol{V} \setminus \boldsymbol{T}) = \mathbb{P}(\boldsymbol{T} \,|\, \boldsymbol{M})$ follows immediately, so the problem is simplified greatly. In this sense, it is meaningful to consider the problem of MB for multiple variables.

The second motivation is that we want to know whether the prediction for $\boldsymbol{T}$ will be affected if the observed values of some variables outside $\boldsymbol{T}$ (in a new observation) are missing. Denote these missing variables by $\boldsymbol{V}_{\mathrm{m}}$. This problem can be considered as follows: find an approximate MB (denoted by $\boldsymbol{M}_{\mathrm{m}}$) of $\boldsymbol{T}$ in $\boldsymbol{V} \setminus \boldsymbol{V}_{\mathrm{m}}$ by means of some method, then check if $\boldsymbol{M}_{\mathrm{m}}$ is an Mb in $\boldsymbol{V}$ via some criterion (e.g., a criterion based on Lemma 2 given by Statnikov et al., 2013); and finally assert $\boldsymbol{T}$ will not be affected if the above checking result is "yes". In this sense, it is also preferred to consider MB for multiple variables.

Figure 4 represents the DAG for the ALARM network (Beinlich et al., 1989), which is well known in the literature. Take $T_1 \triangleq X_{22}$ and $T_2 \triangleq X_{23}$. Then, $\boldsymbol{M}_{T_i}$ is the unique MB of T_i for $i = 1, 2$ under the faithfulness condition, with

$$\boldsymbol{M}_{T_1} \triangleq \{X_1, X_4, X_{15}, X_{21}, X_{23}, X_{27}, X_{29}\} \text{ and } \boldsymbol{M}_{T_2} \triangleq \{X_2, X_{22}, X_{24}, X_{25}, X_{27}, X_{29}\}, \tag{1}$$

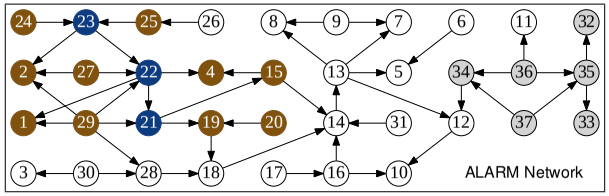


Figure 4: ALARM network (37 nodes and 46 edges): a logical alarm reduction mechanism.

respectively. This leads to intractable computations on the joint probability distribution of T_1 and T_2 given all other variables, consider that $T_1 \in \boldsymbol{M}_{T_2}$ and $T_2 \in \boldsymbol{M}_{T_1}$. Further, if the observed values of some variables (e.g., X_j for $j = 32, 33, \cdots, 37$) in a new observation are missing, can this observation be used any more for predicting T_1 and T_2? Furthermore, we have to face similar problems if three or more target variables are considered.

The third motivation concerns MB discovery algorithms. By Theorem 1, IAMB and KIAMB can be applied to the problem of MB for multiple variables if simply regarding the targets as a multivariate vector, under the strengthened local composition assumption. However, the assumption of local composition imposed on multiple targets may have more occasions to become invalid than imposed on single targets, due to the synergy effect in the sense that neither X nor Y carries information of T but together they contain some information of T (Rauh et al., 2014).

Here is an illustration: considering the BN with the graph in Figure 5 as its DAG, by direct computations using the FullBNT toolbox (Murphy, 2007), we find that

$$\begin{array}{rrll} A \perp\!\!\!\perp C, & A \perp\!\!\!\perp D, & \text{and} & A \perp\!\!\!\perp \{C, D\}; \\ B \perp\!\!\!\perp C, & B \perp\!\!\!\perp D, & \text{and} & B \perp\!\!\!\perp \{C, D\}; \\ \{A, B\} \perp\!\!\!\perp C, & \{A, B\} \perp\!\!\!\perp D, & \text{but} & \{A, B\} \not\perp\!\!\!\perp \{C, D\}. \end{array}$$

By this illustration, the idea of applying the existing MB discovery algorithms to multiple targets seems to be practically improper although it is theoretically feasible, because synergy effects may lead to potential inefficiency and even incorrectness. This motivates us to build some algorithms which are resistant to synergy effects and, further, are time efficient.

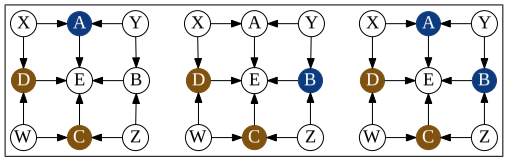


Figure 5: An illustration on synergy effects: each of $\{X, Y, Z, W\}$ takes $\{1, 2\}$ equiprobably; each of $\{A, B, C, D\}$ takes 1 with probabilities p_1, p_2, p_3, p_4 and takes 2 with probabilities $1 - p_1$, $1 - p_2$, $1 - p_3$, $1 - p_4$ given its parents, with $p_4 = p_1 - p_2 + p_3$; E has an arbitrary distribution.

3. Markov Blanket and Markov Boundary for Multiple Variables

This section presents the theoretical results on the problem of Mb and MB for multiple variables when the local intersection property is satisfied and when this property is violated. We study this problem following this way because we are trying to find a suitable approach to transform the problem of Mb and MB from multiple case to single cases, based on which we can build efficient algorithms with high accuracies and low complexities.

3.1 Additivity under Local Intersection

In this subsection, we consider the problem of Mb and MB for multiple variables under the local intersection assumption. We prove Mb and MB possess an ideal property called *additivity*. That is, an Mb of multiple variables can be constructed by simply taking the union of the Mbs of the individual variables and removing the target variables themselves (the same for MB). The results are presented in Theorem 2 and Theorem 3, respectively. Appendix B gives their proofs.

Theorem 2 (Additivity of Mb) *Let $(\mathbb{G}, \mathbb{P})$ be a BN over $\boldsymbol{V}$. The following two statements hold:*

(i) *Let $\boldsymbol{M}_i$ be an Mb of $\boldsymbol{T}_i \subseteq \boldsymbol{V}$ for $i = 1, 2$, and assume $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ satisfies the local intersection assumption. Then, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2)$ is an Mb of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$.*

(ii) *Let $\boldsymbol{M}_i$ be an Mb of $T_i \in \boldsymbol{V}$ for $i = 1, \cdots, k$, and assume $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$ satisfies the local intersection assumption. Then, $\bigcup_{i=1}^{k} \boldsymbol{M}_i \setminus \boldsymbol{T}$ is an Mb of $\boldsymbol{T}$.* ■

The additivity property of Mb can be intuitively described by the information flow metaphor (Cheng et al., 2002) using Figure 6: $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2)$ is enough to cut off all information channels from $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ to other valves, when no information equivalence associated with $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ occurs.

Let $\boldsymbol{T} \subseteq \boldsymbol{V}$ be the set of target variables. As we know, in the case of $|\boldsymbol{T}| = 1$ (denoting $\boldsymbol{T} = \{T\}$), the set $\boldsymbol{M}_T$ composed of the parents, children, and spouses of T is an Mb of it (Pearl, 1988), since $\boldsymbol{M}_T$ d-separates T from all other variables. For the case of $|\boldsymbol{T}| \geqslant 2$ (denoting $\boldsymbol{T} = \{T_1, \cdots, T_k\}$), Theorem 2 indicates that the union of all $\boldsymbol{M}_{T_i}$'s with $T_1, \cdots, T_k$ excluded is an Mb of $\boldsymbol{T}$.

Considering the ALARM network presented in Figure 4, we put $T_1 \triangleq X_{22}$ and $T_2 \triangleq X_{23}$. Then $\boldsymbol{M}_{T_i}$ is an Mb of T_i for $i = 1, 2$, where $\boldsymbol{M}_{T_1}$ and $\boldsymbol{M}_{T_2}$ are defined in (1). Assume $\boldsymbol{T}_{1,2} \triangleq \{T_1, T_2\}$ satisfies the local intersection property. It follows from Theorem 2 that

$$(\boldsymbol{M}_{T_1} \cup \boldsymbol{M}_{T_2}) \setminus \boldsymbol{T}_{1,2} = \{X_1, X_2, X_4, X_{15}, X_{21}, X_{24}, X_{25}, X_{27}, X_{29}\} \triangleq \boldsymbol{M}_{1,2} \tag{2}$$

is an Mb of $\boldsymbol{T}_{1,2}$. Those variables outside $\boldsymbol{M}_{1,2}$ contain no information about $\boldsymbol{T}_{1,2}$ conditioned on $\boldsymbol{M}_{1,2}$ and thereby $\mathbb{P}(\boldsymbol{T}_{1,2} \,|\, \boldsymbol{V} \setminus \boldsymbol{T}_{1,2})$ reduces to $\mathbb{P}(\boldsymbol{T}_{1,2} \,|\, \boldsymbol{M}_{1,2})$. Further, if the observed values of

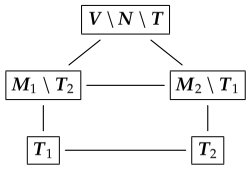


Figure 6: An illustration for additivity of Mb and MB with $\boldsymbol{T} = \boldsymbol{T}_1 \cup \boldsymbol{T}_2$ and $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}$.

X_j for $j = 32, \cdots, 37$ in a new observation are missing, this observation can still be used without affecting the prediction on $\boldsymbol{T}_{1,2}$. Further, $\boldsymbol{M}_{T_3} \triangleq \{X_{15}, X_{19}, X_{20}, X_{22}, X_{29}\}$ is an Mb of $T_3 \triangleq X_{21}$. Assume $\boldsymbol{T}_{1,2,3} \triangleq \{T_1, T_2, T_3\}$ satisfies the local intersection property. Then Theorem 2 shows

$$\boldsymbol{M}_{T_1} \cup \boldsymbol{M}_{T_2} \cup \boldsymbol{M}_{T_3} \setminus \boldsymbol{T}_{1,2,3} = \{X_1, X_2, X_4, X_{15}, X_{19}, X_{20}, X_{24}, X_{25}, X_{27}, X_{29}\} \triangleq \boldsymbol{M}_{1,2,3} \tag{3}$$

is an Mb of $\boldsymbol{T}_{1,2,3}$.

For additivity of Mb shown in (i) of Theorem 2, we have a useful remark (used to simplify our algorithms in Section 4), based on the fact that if $\boldsymbol{M}$ is an Mb of $\boldsymbol{T}$ then $\boldsymbol{M} \cup \boldsymbol{M}_0$ is also an Mb of $\boldsymbol{T}$ for any $\boldsymbol{M}_0 \subseteq \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T}$. By the remark, the local intersection assumption for additivity of Mb is not required in some special cases. The proof of this remark is given in Appendix B.

Remark 1 *In the case of either $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ or $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, the conclusion of* (i) *in Theorem 2 holds without requiring the local intersection assumption.* ■

Theorem 2 shows the additivity of Mb. A natural idea is to wonder if additivity is possessed by MB. Theorem 3 affirms this. Appendix B provides the proof. Note that the statements about the uniqueness of MB in this theorem follow from Lemma 2.

Theorem 3 (Additivity of MB) *Let $(\mathbb{G}, \mathbb{P})$ be a BN over $\boldsymbol{V}$. The following two statements hold:*

(i) *Assume $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ satisfies the local intersection assumption. Let $\boldsymbol{M}_i$ be the unique MB of $\boldsymbol{T}_i$ for $i = 1, 2$. Then, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2)$ is the unique MB of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$.*

(ii) *Assume $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$ satisfies the local intersection assumption. Let $\boldsymbol{M}_i$ be the unique MB of T_i for $i = 1, \cdots, k$. Then, $\bigcup_{i=1}^{k} \boldsymbol{M}_i \setminus \boldsymbol{T}$ is the uniqe MB of $\boldsymbol{T}$.* ■

According to Theorem 3, $\boldsymbol{M}_{1,2}$ defined in (2) is not only an Mb but also the unique MB of $\boldsymbol{T}_{1,2}$ in the ALARM network if the faithfulness condition is satisfied. Further, $\boldsymbol{M}_{1,2,3}$ defined in (3) is the unique MB of $\boldsymbol{T}_{1,2,3}$.

3.2 Theoretical Results in the General Case

Let $(\mathbb{G}, \mathbb{P})$ be a BN over $\boldsymbol{V}$, and assume $\boldsymbol{T}_i \subseteq \boldsymbol{V}$ with $|\boldsymbol{T}_i| \geqslant 1$ has an Mb or MB, $\boldsymbol{M}_i$, for $i = 1, 2$. Denote $\boldsymbol{T} = \boldsymbol{T}_1 \cup \boldsymbol{T}_2$ and $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}$. In the case that $\boldsymbol{M}_i$ is an MB of $\boldsymbol{T}_i$, Theorem 3 reveals that $\boldsymbol{N}$ is an MB of $\boldsymbol{T}$ if $\boldsymbol{T}$ satisfies the local intersection assumption. However, when the local intersection assumption does not hold (meaning information equivalence occurs, as Lemma 1 shows), $\boldsymbol{N}$ may be no longer an MB of $\boldsymbol{T}$, due to one of the following reasons: (i) $\boldsymbol{N}$ may be an Mb but it may not possess minimality, as shown by Example 2; (ii) $\boldsymbol{N}$ may be insufficient to shield $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ from all other variables, so it is no longer an Mb in this case, and some extra variables are required to enter into $\boldsymbol{N}$. Example 1 provides an illustration.

For the first case, we need only to optimize $\boldsymbol{N}$ by simply removing redundant variables from $\boldsymbol{N}$; however, for the second case, the additivity property of MB is thoroughly broken, and the problem of constructing an MB for $\boldsymbol{T}$ based on $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$ becomes complex. On the one hand, there are some variables in $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ needing to enter into $\boldsymbol{N}$; on the other hand, there may be some variables in $\boldsymbol{N}$ becoming redundant once some new members supplement $\boldsymbol{N}$. What we concern are which variables should enter into $\boldsymbol{N}$ and how we find them.

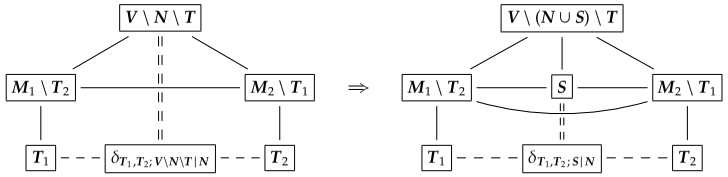


Figure 7: An illustration for the case of violating additivity of Mb and MB, caused by information equivalence.

Assume $\boldsymbol{N}$ is no longer an Mb of $\boldsymbol{T}$. Then, it is easily shown that

$$\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{T}_1 \,|\, \boldsymbol{N}, \quad \text{but} \quad \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{T}_2,$$
$$\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{T}_2 \,|\, \boldsymbol{N}, \quad \text{but} \quad \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{T}_1.$$

That is, $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ contain equivalent information about $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ given $\boldsymbol{N}$. See Figure 7 for an illustration: the valves $\boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ and $\boldsymbol{M}_2 \setminus \boldsymbol{T}_1$ can not cut off all information channels between $\boldsymbol{T}$ and $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$, because some information can flow through $\delta_{\boldsymbol{T}_1,\boldsymbol{T}_2;\boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}|\boldsymbol{N}}$, an information equivalent valve of $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ with respect to $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ given $\boldsymbol{N}$. In other words, $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ may exchange information directly; besides, they also share the equivalent information about $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$. This indicates we should continue to turn off some valves, $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$, besides $\boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ and $\boldsymbol{M}_2 \setminus \boldsymbol{T}_1$ such that $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ no longer exchange information through external valves and thus such that $\boldsymbol{T}$ has no information exchange with remaining valves.

This analysis motivates us to give the following definition:

Definition 3 *With the notations above, we call $\boldsymbol{S}$ ($\subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$) a Markov blanket supplementary (MbS) (of $\boldsymbol{T}$ to $\boldsymbol{N}$), if $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$. Further, a Markov boundary supplementary (MBS) is any MbS such that none of its proper subsets is an MbS.* ■

In what follows, we give the properties of MbS and MBS.

Theorem 4 *Assume $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$. Then, the following statements are equivalent:*

(i) *$\boldsymbol{S}$ is an MbS;*

(ii) $\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \min_{\boldsymbol{S}' \subseteq \boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}} \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}')$;

(iii) $\mathbb{I}(\boldsymbol{T}; \boldsymbol{S} \,|\, \boldsymbol{N}) = \max_{\boldsymbol{S}' \subseteq \boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}} \mathbb{I}(\boldsymbol{T}; \boldsymbol{S}' \,|\, \boldsymbol{N})$;

(iv) *$\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_1$ in $\boldsymbol{V} \setminus \boldsymbol{T}_2$ (or $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$).*

In addition, if $\boldsymbol{S}$ is an MbS, then it is also an MBS if and only if $\boldsymbol{T}_1 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ or $\boldsymbol{T}_2 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ holds for any $Y \in \boldsymbol{S}$. ■

The proof of this theorem is presented in Appendix B.

As seen, (ii) and (iii) of Theorem 4 explain the implication of MbS that the information flow metaphor illustrates in Figure 7: finding an MbS is equivalent to turning off some valves such that $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ no longer exchange information through external valves, or equivalent to finding all remaining equivalent information contained by $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$; (iv) and the property of MBS provide a practical way of building MBS discovery algorithms.

Here, we use an example to demonstrate the notions of MbS and MBS and their properties.

Example 1 *Consider the BN $(\mathbb{G}, \mathbb{P})$ over $\boldsymbol{V} = \{A, B, C, D\}$ presented in Figure 8, in which A, B, and C take $\{1, 2, 3\}$ while D takes $\{1, 2\}$. Put $\boldsymbol{T} = \{T_1, T_2\}$, $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} = \emptyset$, and $\boldsymbol{S} = \{C\}$, $\boldsymbol{S}_0 = \{C, D\}$ with $T_1 = A$, $T_2 = B$, $\boldsymbol{M}_1 = \{B\}$, $\boldsymbol{M}_2 = \{A\}$. Using the theory of information equivalence (Lemeire, 2007), we can show the following results (see Appendix B for the proofs):*

(i) *$\boldsymbol{M}_1$ is an MB of T_1 in $\boldsymbol{V}$: $\mathbb{I}(A; C, D \,|\, B) = 0$ and $\mathbb{I}(A; C, D) > 0$;*

(ii) *$\boldsymbol{M}_2$ is an MB of T_2 in $\boldsymbol{V}$: $\mathbb{I}(B; C, D \,|\, A) = 0$ and $\mathbb{I}(B; C, D) > 0$;*

(iii) *$\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$ in $\boldsymbol{V}$, so $\boldsymbol{S}$ is an MbS: $\mathbb{I}(A, B; D \,|\, C) = 0$;*

(iv) *$\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \min_{\boldsymbol{S}' \subseteq \boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}} \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}')$, because of $\mathbb{I}(A; B \,|\, C) = \mathbb{I}(A; B \,|\, C, D)$, $\mathbb{I}(A; B \,|\, C) \leqslant \mathbb{I}(A; B \,|\, D)$, and $\mathbb{I}(A; B \,|\, C) \leqslant \mathbb{I}(A; B)$;*

(v) *$\mathbb{I}(\boldsymbol{T}; \boldsymbol{S} \,|\, \boldsymbol{N}) = \max_{\boldsymbol{S}' \subseteq \boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}} \mathbb{I}(\boldsymbol{T}; \boldsymbol{S}' \,|\, \boldsymbol{N})$;*

(vi) *$\boldsymbol{N} \cup \boldsymbol{S}$ is an MB of T_1 in $\boldsymbol{V} \setminus \{T_2\}$: $\mathbb{I}(A; C, D) > 0$ and $\mathbb{I}(A; D \,|\, C) = 0$;*

(vii) *$\boldsymbol{N} \cup \boldsymbol{S}$ is an MB of T_2 in $\boldsymbol{V} \setminus \{T_1\}$: $\mathbb{I}(B; C, D) > 0$ and $\mathbb{I}(B; D \,|\, C) = 0$;*

(viii) *$\boldsymbol{S}$ is an MBS; $\boldsymbol{S}_0$ is an MbS (not an MBS): $\mathbb{I}(A, B; C, D) > 0$ and $\mathbb{I}(A; B \,|\, C, D) = \mathbb{I}(A; B \,|\, C)$.* ■

By Example 1, A and B share the equivalent information about C, so turning off the valve A (or B) means cutting off all the channels from B (or A) to C. This is why they can screen off each other from C. However, A and B lose the shield if they are integrated into a whole. In this case, we have to turn C off such that A and B no longer exchange information through external valves. This example reveals that an MBS is a minimal set of variables, $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$, such that $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ contain no equivalent information about the remaining variables given $\boldsymbol{N} \cup \boldsymbol{S}$.

When finding an MBS, $\boldsymbol{S}$, and letting the variables in $\boldsymbol{S}$ supplement $\boldsymbol{N}$, there may be some variables in $\boldsymbol{N}$ becoming redundant. In addition, $\boldsymbol{N}$ may be redundant even before supplementing $\boldsymbol{S}$. Example 2 gives an illustration. For both cases, we need to remove the redundant variables.

Example 2 *Consider the BN presented in Figure 9, in which any one variable from $\{A, B, C\}$ and another from $\{D, E, F\}$ (denoted by X and Y, respectively) contain context-independent equivalent information about G (see Statnikov et al., 2013, Example 3). Then, $\{X, Y\}$ is an MB of G. Put now $\boldsymbol{T}_1 = \{C, F\}$ and $\boldsymbol{T}_2 = \{G\}$, and take $\boldsymbol{M}_1 = \{B, E\}$ and $\boldsymbol{M}_2 = \{B, D\}$. Note that $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ (and also $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$). It concludes that $\boldsymbol{N} = \{B, D, E\}$ is not an MB but only an Mb of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$, since its proper subset $\{B, E\}$ is also an Mb (and also an MB) of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$. This shows why the process of refining $\boldsymbol{N}$ is necessary.* ■

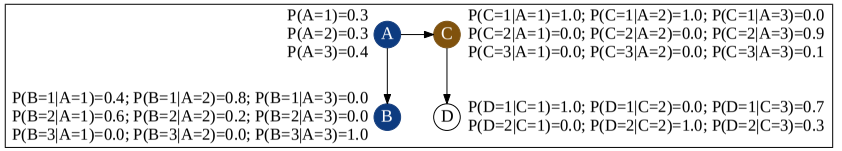


Figure 8: BN $(\mathbb{G}, \mathbb{P})$: $\mathbb{P}$ is a joint probability distribution over $\boldsymbol{V} = \{A, B, C, D\}$ with each variable taking values $\{1, 2, 3\}$ except for D taking $\{1, 2\}$; $\mathbb{G}$ is a DAG over $\boldsymbol{V}$.

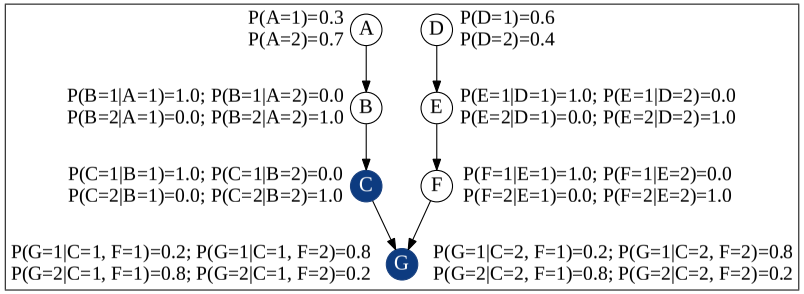


Figure 9: BN $(\mathbb{G},\mathbb{P})$: $\mathbb{P}$ is a joint probability distribution over $\boldsymbol{V}=\{A,B,C,D,E,F,G\}$ with all variables taking values $\{1,2\}$, $\mathbb{G}$ is a DAG with the variables in $\boldsymbol{V}$ as its nodes.

3.3 An Alternative Approach

Before building MB discovery algorithms for multiple targets, this subsection concisely presents an alternative approach to the *additivity based* and *MBS based* methods. In Section 5, we will apply this method as an FS strategy in multi-class prediction problems.

Let $\boldsymbol{T}\triangleq\{T_1,\cdots,T_k\}$ be the targets of interest and T be $\boldsymbol{T}$'s merged version, taking values $\{1,\cdots,t\}$ with $t\geqslant 3$. This procedure transforms the MB discovery for multiple targets $\boldsymbol{T}$ into the MB discovery for single target T, so all the existing MB discovery algorithms can be employed theoretically if the required conditions are satisfied. However, if t is large, selecting features of $\boldsymbol{T}$ or T directly will be difficult. Subsection 3.1 and Subsection 3.2 provide a way of solving this problem in different situations. In this case, an alternative strategy is to further convert T into a set of dummy variables denoted by $\left\{T_j^{(d)}\right\}_{j=1}^{t}$, where $T_j^{(d)}$ is a 0-1 variable defined as

$$T_j^{(d)}=\begin{cases}1, & \text{if } T=j\\ 0, & \text{if } T\neq j\end{cases}$$

This transformation produces a multiple-target $\boldsymbol{T}^{(d)}\triangleq\left(T_1^{(d)},\cdots,T_t^{(d)}\right)$. Clearly, $\boldsymbol{T}$, T, and $\boldsymbol{T}^{(d)}$ have the same MBs. In what follows, we show the MB of $\boldsymbol{T}^{(d)}$ can be derived by simply taking the union of MBs of $T_1^{(d)},\cdots,T_t^{(d)}$ and then removing the redundant variables in an efficient way. The proof will be given in Appendix B.

Theorem 5 *Let $\boldsymbol{M}_j$ be an MB of $T_j^{(d)}$ in $\boldsymbol{V}\setminus\boldsymbol{T}$ for $j=1,\cdots,t$. Then, $\boldsymbol{M}\triangleq\cup_{j=1}^{k}\boldsymbol{M}_j$ is an Mb of $\boldsymbol{T}$. Further, $\boldsymbol{M}$ is an MB of $\boldsymbol{T}$ iff for any $X\in\boldsymbol{M}$ there is some j such that $T_j^{(d)}\not\perp\!\!\!\perp X\mid\boldsymbol{M}\setminus\{X\}$.* ■

For why this transformative method is efficient, the fourth concluding remark in Section 7 will make a brief explanation.

4. Algorithms

This section builds MB discovery algorithms for multiple targets, $\{T_1,\cdots,T_k\}\triangleq\boldsymbol{T}$.

Let $\mathbb{A}$ be an MB discovery algorithm, assumed to perform well when used to discover an MB for a single target. In this paper, we employ `IAMB` and `KIAMB` as $\mathbb{A}$. Clearly, $\mathbb{A}$ can be directly used to find an MB for $\boldsymbol{T}$ if simply regarding $\boldsymbol{T}$ as the input of $\mathbb{A}$. Usually, this will lead to low accuracies and high complexities.

By Theorem 4, the MB discovery problem for multiple targets can be translated equivalently into a number of MB discovery problems for single targets, according to the following way: (i) use $\mathbb{A}$ to find an MB of T_i in $\boldsymbol{V}$ for $i=1,\cdots,k$, denoted by $\boldsymbol{M}_i$; (ii) find an Mb of T_2 in $\boldsymbol{V}\setminus\{T_1\}$ based on $(\boldsymbol{M}_1\cup\boldsymbol{M}_2)\setminus\{T_1,T_2\}$, and then get an MB of $\{T_1,T_2\}$, written as $\boldsymbol{M}_{1,2}$; (iii) find an Mb of T_3 in $\boldsymbol{V}\setminus\{T_1,T_2\}$ based on $(\boldsymbol{M}_{1,2}\cup\boldsymbol{M}_3)\setminus\{T_1,T_2,T_3\}$, and then get an MB of $\{T_1,T_2,T_3\}$, written as $\boldsymbol{M}_{1,2,3}$; (iv) the rest can be done in a similar manner. Following this way, the input of $\mathbb{A}$ for each use is a single variable, so this idea successfully avoids assigning an multivariate input to $\mathbb{A}$. Note that in the above process the equivalent information is extracted in a stepwise manner.

4.1 `IAMBS` and `KIAMBS`

Let $(\mathbb{G},\mathbb{P})$ be a BN over $\boldsymbol{V}$, and assume $\boldsymbol{T}_i\subseteq\boldsymbol{V}$ with $|\boldsymbol{T}_i|\geqslant 1$ has an MB, $\boldsymbol{M}_i$, for $i=1,2$. Denote $\boldsymbol{N}=(\boldsymbol{M}_1\cup\boldsymbol{M}_2)\setminus(\boldsymbol{T}_1\cup\boldsymbol{T}_2)$. This subsection presents the algorithms for discovering an MB of $\boldsymbol{T}_1\cup\boldsymbol{T}_2$. To design one such algorithm, we note that there may be some variables in $\boldsymbol{N}$ becoming redundant once an MBS, $\boldsymbol{S}$, is supplemented. Therefore, we need to first find $\boldsymbol{S}$ by setting $\boldsymbol{N}$ as a whitelist in $\mathbb{A}$ and then refine $\boldsymbol{N}$.

Applying this idea to `IAMB` and `KIAMB`, we obtain two algorithms called `IAMBS` and `KIAMBS`, in which "S" refers to as "supplementary". Their pseudo codes are presented in Algorithm 1. In order to differentiate these two algorithms, we set K in the `KIAMBS` algorithm as $K\in[0,1)$. It is mentioned here that $\boldsymbol{S}_2$ is a random subset of $\boldsymbol{S}_1$ with size $\max\{1,\lfloor|\boldsymbol{S}_1|\cdot K\rfloor\}$ in Line 5 of `KIAMBS`. As seen, these two algorithms first find an MbS, $\boldsymbol{S}$, in the growing phase and then refine $\boldsymbol{S}$ and $\boldsymbol{N}$ in sequence in the shrinking phase.

For example, based on a data set drawn from the BN in Example 1, the unique MB, $\{C\}$, of $\{A,B\}$ can be discovered by calling `IAMBS` or `KIAMBS` only once.

Theorem 1 presents the correctness of `IAMB` and `KIAMB` under the assumption that $\boldsymbol{T}_1\cup\boldsymbol{T}_2$ satisfies the local composition property. The theorem below shows `IAMBS` and `KIAMBS` are correct if $\boldsymbol{T}_2$ (instead of $\boldsymbol{T}_1\cup\boldsymbol{T}_2$) satisfies the local composition property. Appendix B gives the proof.

Theorem 6 (Correctness of `IAMBS` and `KIAMBS`) *Assume that $\boldsymbol{T}_2$ satisfies the local composition property, and that all CI tests are correct. Then* (i) *`IAMBS` outputs an MB of $\boldsymbol{T}_1\cup\boldsymbol{T}_2$;* (ii) *`KIAMBS` outputs an MB of $\boldsymbol{T}_1\cup\boldsymbol{T}_2$ for any $K\in[0,1)$.* ■

The following remark presents a relation among local intersection, local composition, and the adjacency faithfulness condition, under the *orientation faithfulness condition*. The proof is given in Appendix B. Here, the orientation faithfulness condition (Ramsey et al., 2006; Lemeire et al., 2012) is defined as: for any $X,Y,Z\in\boldsymbol{V}$ such that X and Z are adjacent to Y but X is not adjacent to Z, (i) if $X\rightarrow Y\leftarrow Z$, then $X\not\perp\!\!\!\perp Z\,|\,\boldsymbol{W}$ holds for any $\boldsymbol{W}\subseteq\boldsymbol{V}\setminus\{X,Z\}$ with $Y\in\boldsymbol{W}$; (ii) otherwise, $X\not\perp\!\!\!\perp Z\,|\,\boldsymbol{W}$ holds for any $\boldsymbol{W}\subseteq\boldsymbol{V}\setminus\{X,Y,Z\}$.

Remark 2 *The following two statements hold:* (a) *violating local intersection implies violating adjacency faithfulness;* (b) *under the orientation faithfulness condition, violating local composition at the end of the first phase of `IAMB` or `KIAMB` or `IAMBS` or `KIAMBS` means violating adjacency faithfulness.* ■

In addition, the lemma below is useful to explain the succeeding remarks.

Lemma 3 (a) *If there is* $\boldsymbol{P} \subseteq \boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ *such that* $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{P} | (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$, *then* $(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$ *is an Mb of* $\boldsymbol{T}_1$; (b) *If there is* $\boldsymbol{Q} \subseteq \boldsymbol{N} \setminus \boldsymbol{P}$ *such that* $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{Q} | (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$ *and* $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{Q} | (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$, *then* $(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$ *is an Mb of* $\boldsymbol{T}_1$, *and* $(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$ *is an Mb of* $\boldsymbol{T}_2$. ■

For Algorithm 1, we have three remarks below:

(i) In these two algorithms, the two CI tests for adding members to $\boldsymbol{S}$ and for refining $\boldsymbol{S}$ are based on $\boldsymbol{T}_2$ instead of $\boldsymbol{T}$, while the CI test for refining $\boldsymbol{N}$ is based on $\boldsymbol{T}$ instead of $\boldsymbol{T}_2$. (a) For the first two CI tests, $\boldsymbol{T}_2$ can be replaced with $\boldsymbol{T}$ without affecting the correctness of the algorithms, since $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{X} | \boldsymbol{N} \cup \boldsymbol{S}' \Leftrightarrow \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} | \boldsymbol{N} \cup \boldsymbol{S}'$ holds for any $\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ and $\boldsymbol{X} \subseteq \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}') \setminus \boldsymbol{T}$. However, if we replace $\boldsymbol{T}_2$ with $\boldsymbol{T}$, the resulting algorithms will need much longer time to run. This is why we use $\boldsymbol{T}_2$ in stead of $\boldsymbol{T}$ in these two places. (b) For the third CI test, $\boldsymbol{T}$ can not be replaced with $\boldsymbol{T}_2$, because of $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{X} | (\boldsymbol{N} \setminus \boldsymbol{X}) \cup \boldsymbol{S} \nRightarrow \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} | (\boldsymbol{N} \setminus \boldsymbol{X}) \cup \boldsymbol{S}$.

(ii) According to Remark 2, there may be some situations in which both local intersection and local composition are simultaneously violated. In this case, `IAMBS` and `KIAMBS` may not correctly work. Specifically, the *violation of local intersection* means $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ contain equivalent information about $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ given $\boldsymbol{N}$; while the *violation of local composition* indicates not all equivalent information are successfully extracted by $\boldsymbol{N}$. Let $\boldsymbol{P}$ and $\boldsymbol{Q}$ be defined as in Lemma 3, and assume $\boldsymbol{P} \cup \boldsymbol{Q} \neq \emptyset$. Then, it can be shown that $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ contain equivalent information about $\boldsymbol{P} \cup \boldsymbol{Q}$ given $\boldsymbol{N} \setminus (\boldsymbol{P} \cup \boldsymbol{Q})$. This means some equivalent information about $\boldsymbol{P} \cup \boldsymbol{Q}$ shared by $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ conditioned on $\boldsymbol{N} \setminus (\boldsymbol{P} \cup \boldsymbol{Q})$ may mask some equivalent information about $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ contained by $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$ conditioned on $\boldsymbol{N}$. This may be why not all equivalent information can be extracted by $\boldsymbol{N}$. According to this analysis, a potential remedy is to run `IAMBS` or `KIAMBS` by replacing $\boldsymbol{N}$ with a superset of $\boldsymbol{N} \setminus (\boldsymbol{P} \cup \boldsymbol{Q})$ that is a subset of $\boldsymbol{N}$.

(iii) By Remark 1, if $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ or $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, $\boldsymbol{N}$ must be an Mb of $\boldsymbol{T}$, so Lines 2∼14 of `IAMBS` and `KIAMBS` can be omitted. In this case, however, it is still necessary to refine $\boldsymbol{N}$, because $\boldsymbol{N}$ may not possess minimality. Example 2 illustrates this necessity.

Algorithm 1: `IAMBS` and `KIAMBS`

Procedure: $\boldsymbol{M} \leftarrow$ `IAMBS`$(\boldsymbol{D}; \boldsymbol{T}_1, \boldsymbol{T}_2; \boldsymbol{M}_1, \boldsymbol{M}_2)$
Input: a data matrix $\boldsymbol{D}$; two sets of targets $\boldsymbol{T}_1$ and $\boldsymbol{T}_2$; an MB $\boldsymbol{M}_i$ of $\boldsymbol{T}_i$ for $i = 1, 2$.
Output: an MB, $\boldsymbol{M}$, of $\boldsymbol{T} \triangleq \boldsymbol{T}_1 \cup \boldsymbol{T}_2$.

```
//Forward: Growing Phase
S ← ∅
while S has changed do
    M ← N ∪ S
    Y ← arg max_{X∈V\M\T} f_D(T_2; X | M)
    if T_2 ⊥⊥ Y | M then
        S ← S ∪ {Y}
    end
end
//Backward: Shrinking Phase
foreach X ∈ S do
    if T_2 ⊥⊥ Y | N ∪ (S \ {Y}) then
        S ← S \ {Y}
    end
end
foreach Y ∈ N do
    if T ⊥⊥ Y | (N \ {Y}) ∪ S then
        N ← N \ {Y}
    end
end
return M ← N ∪ S
```

Procedure: $\boldsymbol{M} \leftarrow$ `KIAMBS`$(\boldsymbol{D}; \boldsymbol{T}_1, \boldsymbol{T}_2; \boldsymbol{M}_1, \boldsymbol{M}_2; K)$
Input: Besides $\{\boldsymbol{D}, \boldsymbol{T}_i, \boldsymbol{M}_i\}$, $K \in [0, 1)$ is a randomization parameter.
Output: an MB, $\boldsymbol{M}$, of $\boldsymbol{T} \triangleq \boldsymbol{T}_1 \cup \boldsymbol{T}_2$.

```
//Forward: Growing Phase
S ← ∅
while S has changed do
    M ← N ∪ S
    if S_1 ← {X∈V\M\T : T_2 ⊥̸⊥ X | M} ≠ ∅ then
        Y ← arg max_{X∈S_2} f_D(T_2; X | M)
        S ← S ∪ {Y}
    end
end
//Backward: Shrinking Phase
foreach X ∈ S do
    if T_2 ⊥⊥ Y | N ∪ (S \ {Y}) then
        S ← S \ {Y}
    end
end
foreach Y ∈ N do
    if T ⊥⊥ Y | (N \ {Y}) ∪ S then
        N ← N \ {Y}
    end
end
return M ← N ∪ S
```


In addition, another problem that we concern is whether we can refine $\boldsymbol{N}$ before seeking $\boldsymbol{S}$ and, if this is the case, which variables in $\boldsymbol{N}$ can be removed directly. We consider this problem because any redundant variable in $\boldsymbol{N}$ can lead to unnecessary inaccuracies when using $\boldsymbol{N}$ as a part of the conditional set in practical computations. Lemma 3 indicates we can do like this. However, to avoid the danger of missing the information about $\boldsymbol{P} \cup \boldsymbol{Q}$ (this occurs if the equivalent information involved in $\boldsymbol{P} \cup \boldsymbol{Q}$ given $\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}$ is different in some sense from any part of the equivalent information involved in $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ given $\boldsymbol{N}$), we recommend to first search the members of $\boldsymbol{S}$ in $\boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ and then check if some variables in $\boldsymbol{P} \cup \boldsymbol{Q}$ are necessary to enter into $\boldsymbol{S}$ when implementing Lines 2∼8 of `IAMBS` and `KIAMBS`. Note that this will increase the total running time.

4.2 `MIAMB` and `MKIAMB`

In this subsection, we present two multivariate Markov boundary discovery algorithms, called `MIAMB` and `MKIAMB`, respectively.

Let $\{T_1, \cdots, T_k\} \in \boldsymbol{V}$ with $\boldsymbol{M}_i$ as its an MB for $i = 1, \cdots, k$. If the local intersection property is satisfied, Theorem 3 shows $\bigcup_{i=1}^{k} \boldsymbol{M}_i \setminus \boldsymbol{T}$ is an MB of $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$. Otherwise, $\boldsymbol{M}$ may be no longer an MB. In this case, we use `MIAMB` or `MKIAMB` to seek an MB for $\boldsymbol{T}$. Given an ordering of $T_1, \cdots, T_k$, saying $\tau \triangleq \{i_1, \cdots, i_k\}$, which determines the priorities of the variables in $\boldsymbol{T}$ entering into the queue whose an MB will be sought in the current step, we denote an MB of $\{T_{i_1}, \cdots, T_{i_\ell}\} \triangleq \boldsymbol{T}_{i_\ell}^*$ by $\boldsymbol{M}_{i_j}^*$.

With these notations, `MIAMB` and `MKIAMB` are pseudo-coded in Algorithm 2. Their correctness, shown by Theorem 7, is a direct consequence of Theorem 1 and Theorem 6. As seen, `MIAMB` or `MKIAMB` uses the following stepwise idea: it first finds an MB of two targets $\{T_{i_1}, T_{i_2}\} = \{T_{i_1}\} \cup \{T_{i_2}\}$, and then finds an MB of three targets $\{T_{i_1}, T_{i_2}, T_{i_3}\} = \{T_{i_1}, T_{i_2}\} \cup \{T_{i_3}\}$; the rest can be done in a similar manner until all the k target variables are considered.

Theorem 7 (Correctness of `MIAMB` and `MKIAMB`) *Assume that* T_i *satisfies the local composition property for* $i = 1, \cdots, k$, *and that all CI tests are correct. Denote* $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$. *Then* (i) `MIAMB` *outputs an MB of* $\boldsymbol{T}$; (ii) `MKIAMB` *outputs an MB of* $\boldsymbol{T}$ *for any* $K \in [0, 1)$. ■

As we know, for any real data, those preconditions (such as faithfulness or local composition) required by a learning algorithm are hard to hold exactly. However, our algorithms can be seen as an improvement over earlier methods. Specifically, `IAMB`/`KIAMB` algorithms require faithfulness or

local composition for multiple targets when used for MB discovery of multiple targets, while our MIAMB/MKIAMB only need local composition for single targets, which may be more close to real situations than faithfulness or local composition for multiple targets.

For MIAMB or MKIAMB, an ordering τ is set in Algorithm 2 mainly because different orderings may lead to different computational complexities. In Subsection 4.4, we will make a complexity analysis about the algorithms, based on which we present a feasible way of selecting τ, under the expectation that our algorithms should be run as quickly as possible. When $|\boldsymbol{T}| = 2$, however, τ is not necessary.

Besides MIAMB/MKIAMB algorithms (which are *MBS based*), we can consider additivity based (Theorem 3) and dummy variables based (Theorem 5) algorithms: (a) the *additivity based* MIAMB or MKIAMB simply takes the union of outputs of IAMB/KIAMB with respect to all single targets as the output; its correctness requires the conditions in Theorem 7 plus Theorem 3; and (b) the *dummy (variables based)* MIAMB/MKIAMB takes the union of the outputs of IAMB/KIAMB with respect to every dummy variable and removes redundant variables; its correctness requires the same condition as in Theorem 7. Throughout this paper, unless specified, MIAMB/MKIAMB denote the MBS based algorithms.

4.3 A Discussion on CI Test

As argued by Aliferis et al. (2010a, p. 200), the quality of an MB discovery algorithm highly depends on the selected CI testing methods. In this subsection, we discuss the ways of practically doing CI tests. Usually, the Pearson's X^2 test or the log-likelihood ratio G^2 test can be employed for this purpose (Yaramakala, 2004; Bromberg and Margaritis, 2009; Aliferis et al., 2010b; Statnikov et al., 2013). Here, the X^2 statistic and the G^2 statistic have the same asymptotic χ^2 distribution. We can also use some experimental testing methods such as the Akaike information criterion-based test (Cressie and Read, 1989; Scutari, 2010).

Algorithm 2: MIAMB and MKIAMB

Procedure: $\boldsymbol{M} \leftarrow$ MIAMB$(\boldsymbol{D}; \boldsymbol{T}; \tau)$
Input: a data matrix $\boldsymbol{D}$; a target set $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$; and an ordering $\tau \triangleq \{i_1 \cdots, i_k\}$.
Output: an MB, $\boldsymbol{M}$, of $\boldsymbol{T}$.

// MIAMB: $\boldsymbol{M} \leftarrow$ MIAMB$(\boldsymbol{D}; \boldsymbol{T}; \tau)$
for $\ell \leftarrow 1$ **to** k **do**
  $\boldsymbol{M}_{i_\ell} \leftarrow$ IAMB$(\boldsymbol{D}; \{T_{i_\ell}\})$
end
for $\ell \leftarrow 2$ **to** k **do**
  $\boldsymbol{M}^*_{i_\ell} \leftarrow$ IAMBS$(\boldsymbol{D}; \boldsymbol{T}^*_{i_{\ell-1}}, \{T_{i_\ell}\}; \boldsymbol{M}^*_{i_{\ell-1}}, \boldsymbol{M}_{i_\ell})$
end
return $\boldsymbol{M} \leftarrow \boldsymbol{M}^*_{i_k}$

Procedure: $\boldsymbol{M} \leftarrow$ MKIAMB$(\boldsymbol{D}; \boldsymbol{T}; K; \tau)$
Input: a data matrix $\boldsymbol{D}$; a target set $\boldsymbol{T}$; a randomization parameter $K \in [0, 1)$; and an ordering τ.
Output: an MB, $\boldsymbol{M}$, of $\boldsymbol{T}$.

// MKIAMB: $\boldsymbol{M} \leftarrow$ MKIAMB$(\boldsymbol{D}; \boldsymbol{T}; K; \tau)$
for $\ell \leftarrow 1$ **to** k **do**
  $\boldsymbol{M}_{i_\ell} \leftarrow$ KIAMB$(\boldsymbol{D}; \{T_{i_\ell}\}; K)$
end
for $\ell \leftarrow 2$ **to** k **do**
  $\boldsymbol{M}^*_{i_\ell} \leftarrow$ KIAMBS$(\boldsymbol{D}; \boldsymbol{T}^*_{i_{\ell-1}}, \{T_{i_\ell}\}; \boldsymbol{M}^*_{i_{\ell-1}}, \boldsymbol{M}_{i_\ell}; K)$
end
return $\boldsymbol{M} \leftarrow \boldsymbol{M}^*_{i_k}$

Recall that we are dealing with the MB discovery problem for *multiple* target variables. When the target set, namely $\boldsymbol{T}$, contains only a few variables (e.g., 1 or 2), the X^2 test or the G^2 test performs quite well in most situations. Unfortunately, when $\boldsymbol{T}$ contains too many variables (e.g., 5 or 6 or even more), X^2 or G^2 may not work well due to the overmany degrees of freedom. See Appendix C for a detailed discussion. In fact, as Cochran (1954, p. 420) recommended about the working rules for X^2 (also applicable to G^2), these two testing methods are unreliable if more than 20% of the cells in contingency tables have an expected count of less than 5 data points; however, such cases frequently arise in practice (Bromberg and Margaritis, 2009; Yaramakala, 2004).

Many authors have considered improving X^2 and G^2 by adjusting the statistics. Lawley (1956) showed that such tests can be improved by multiplying with a suitable scale factor; Hosmane (1986, 1987, 1990) and the pioneer scholars recommended the following two adjustment procedures (i) replace zero observed counts by a positive constant, leaving nonzero counts intact; and (ii) add a positive constant to all the observed counts. Brin et al. (1997) and Silverstein et al. (1998) used two heuristic "solutions" to the problem of low expected counts as follows: (i) simply ignore these cells when calculating X^2 or G^2; and (ii) use what is called *contingency table support* (CT-support): a set of items S has CT-support s at the t% level if at least t% of the cells in the contingency table for S have value s. Aliferis et al. (2010b) considered a similar heuristic called *heuristic power size*, which denotes the smallest sample size per cell in the contingency table of a reliable CI test.

The above ideas can lead to improvements on X^2 and G^2 to varying degrees if the dimensions are not very high. However, when working on the MB discovery problem for multiple targets, we need more suitable methods to do CI tests. For this reason, we suggest the following *practical operation*: when $|\boldsymbol{T}| \leqslant 2$, we can (i) use X^2 or G^2 or their variants mentioned above to do CI tests; otherwise, we consider the following testing method: (ii) use CMI and an experimental threshold, ε, to make statistical decisions as Cheng et al. (2002) did, in the sense that $\mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \geqslant \varepsilon$ asserts $\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ while $\mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) < \varepsilon$ concludes $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$, where $\varepsilon \triangleq |\boldsymbol{T}|^{a_1} \cdot \frac{100\, a_2}{n} \cdot \log_2 v$ is related to the the sample size, the average number of values that each variable takes, and the number of targets (denoted by n, v, and $|\boldsymbol{T}|$, respectively), in which a_1 and a_2 are two adjusting factors ($a_1 = 0.5$ and $a_2 \in (0.1, 0.5)$ are recommended). The association function, $f_{\boldsymbol{D}}$, can be selected as

$$f_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) = \mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \triangleq f^{(2)}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}). \tag{4}$$

Besides this experimental method, we can (iii) improve X^2 or G^2 by adjusting the number of the theoretical degrees of freedom.

For the above (iii), to be clear, we consider the G^2 statistic, $G^2(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \triangleq 2n \cdot \mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z})$, which approximates to the chi-square variate with $r \triangleq (r_{\boldsymbol{X}} - 1)(r_{\boldsymbol{Y}} - 1) r_{\boldsymbol{Z}}$ degrees of freedom, namely $\chi^2(r)$, where $r_{\boldsymbol{\xi}}$ represents the number of configurations for $\boldsymbol{\xi}$ (de Campos, 2006, p. 2158). Denote the p-value by

$$p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) = \mathbb{P}\{\chi^2(r) \geqslant G^2(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})\}.$$

Then, the G^2 test asserts $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ if $p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) > \alpha$ for a significance level α, and concludes $\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$ if $p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \leqslant \alpha$. In this paper, α is set to be 0.05. Aliferis et al. (2010a, pp. 200–201) provided a further discussion about this. Accordingly, the *negative p-value* is used as the association function, $f_{\boldsymbol{D}}$, as Tsamardinos et al. (2006), Aliferis et al. (2010a,b), and Statnikov et al. (2013) did:

$$f_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) = -p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) = -\mathbb{P}\{\chi^2(r) \geqslant G^2(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})\} \triangleq f^{(1)}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}). \tag{5}$$

Replace the theoretical value of r in $p(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z})$ with its a damped version of the form

$$g_{n,\kappa}(r) \triangleq r\left(1-\mathrm{e}^{-\frac{n}{\kappa r}}\right), \tag{6}$$

where $\kappa > 0$ is a constant, based on which $\frac{n}{\kappa}$ measures the amount of valid cells that n sample instances can support. For convenience, we will call the resulted p-value, denoted by $p_g(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z})$ instead of $p(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z})$, and the resulted testing method to be the *damped p-value* and the *damped log-likelihood ratio test* (or damped G^2 test). Further, we use the the following association function:

$$f_{\boldsymbol{D}}(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z}) = -p_g(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z}) = -\mathbb{P}\{\chi^2(g_{n,\kappa}(r)) \geqslant G^2(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z})\} \triangleq f^{(3)}_{\boldsymbol{D}}(\boldsymbol{X};\boldsymbol{Y}\,|\,\boldsymbol{Z}). \tag{7}$$

In Appendix C, we will provide the details for this damping procedure, and give some numerical illustrations about its reasonability. Clearly, the damped G^2 test approximately degenerates into the ordinary G^2 test when taking κ as a very small positive number.

4.4 Complexity Analysis

In the following, we analyze the computational complexities of the four algorithms: `IAMB`, `KIAMB`, `MIAMB`, and `MKIAMB`. Usually, the number of CI tests can be employed to measure the complexity of a CI-based MB discovery algorithm (Tsamardinos et al., 2003, 2006; Aliferis et al., 2010a), considering there exists efficient implementations of the CMI-based test or the association computation taking time $O(n\log n)$ if the conditional set is small. However, Aliferis et al. (2010a) also mentioned that the running time, denoted by $t_{n,q}$, for computing per CMI-based statistic is linear to the sample size, n, and exponential to the number, q, of variables in the conditional set. This means we should take $t_{n,q}$ into account, not simply using $O(n\log n)$ to measure the complexity.

Assume we are seeking an MB for $\boldsymbol{T} \triangleq \{T_1,\cdots,T_k\}$ according to the ordering τ. Without loss of generality, we assume $\tau = \{1,\cdots,k\}$. Consider the case of $k = 2$. Suppose $\boldsymbol{M}_i$ is an MB of T_i with $|\boldsymbol{M}_i| = m_i \geqslant 1$, and $\boldsymbol{S}$ is an MBS for $\boldsymbol{N} \triangleq \boldsymbol{M}_1 \cup \boldsymbol{M}_2 \setminus \{T_1,T_2\}$ with $|\boldsymbol{S}| = s \geqslant 0$. By Remark 1, we assume $T_1 \in \boldsymbol{M}_2$ and $T_2 \in \boldsymbol{M}_1$. Recall that the number of all variables is p. It follows that:

- In view of $|\boldsymbol{N}\cup\boldsymbol{S}| = m_1+m_2+s-2 \triangleq m$, `IAMB` takes time $O[(mp+m)t_{n,m}]$ to finish an execution. Thus, the complexity of `IAMB` is $O(mpt_{n,m})$. `KIAMB` has almost the same complexity.
- For `MIAMB`, it first takes time $O[(m_1p+m_1)t_{n,m_1}+(m_2p+m_2)t_{n,m_2}]$ to find $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$; then it seeks $\boldsymbol{S}$ and refines $\boldsymbol{N}$ taking time $O\{[s(p-m_1-m_2+2)+m]t_{n,m}\}$. Hence, `MIAMB` needs time $O\{(m_1p+m_1)t_{n,m_1}+(m_2p+m_2)t_{n,m_2}+[s(p-m_1-m_2+2)+m]t_{n,m}\}$ to finish an execution, so its complexity is $O(m_1pt_{n,m_1}+m_2pt_{n,m_2}+spt_{n,m})$. `MKIAMB` has almost the same complexity.

By this analysis, the complexity of `MIAMB` or `MKIAMB` is lower than that of `IAMB` or `KIAMB`. In fact, noting $t_{n,q}$ is exponential to q ($\leqslant m$; meaning $t_{n,q} \ll t_{n,m}$ in most situations) for $q = m_1, m_2$, this implies `MIAMB`/`MKIAMB` are expected to need much less time to run than `IAMB`/`KIAMB`, especially when $\boldsymbol{T}$ contains many variables. The evaluation section (Figure 15) confirms this expectation in the case of moderately large sample size.

For the general case, using the notations in Subsection 4.2 with $|\boldsymbol{M}_i| = m_i$ $(i = 1,\cdots,k)$, we assume $\boldsymbol{S}_i$ be an MBS for $\boldsymbol{M}^*_{i-1}$ and $\boldsymbol{M}_i$, with $|\boldsymbol{S}_i| = s_i$ $(i = 2,\cdots,k)$. Denote $m^*_i \triangleq \sum_{j=1}^{i} m_j + \sum_{j=2}^{i} s_j - i$. Note that, in general, $t_{n,m_a} \ll t_{n,m^*_a} \ll t_{n,m^*_b}$ for $a < b$. Then, the `IAMB` or `KIAMB` algorithm has the complexity $O(m^*_k pt_{n,m^*_k})$, while `MIAMB` or `MKIAMB` has a lower complexity $O(\sum_{i=1}^{k} m_i pt_{n,m_i} + \sum_{i=2}^{k} s_i pt_{n,m^*_i})$.

According to this theoretical result on complexities, we can use the ordering, $\tau \triangleq \{i_1\cdots,i_k\}$, in `MIAMB` or `MKIAMB` such that $m_{i_1} \leqslant \cdots \leqslant m_{i_k}$. This can reduce the complexities to some extent.

Besides, the additivity based `MIAMB`/`MKIAMB` algorithms have almost the same complexity as the MBS based `MIAMB`/`MKIAMB`, while the dummy `MIAMB`/`MKIAMB` have the complexity $O(mr_{\boldsymbol{T}}pt_{n,m})$, where $m = \sum_{j=1}^{k} m_j$, $r_{\boldsymbol{T}} = \prod_{j=1}^{k} r_{T_j}$, r_ξ denotes the number of configurations for ξ. It will be seen from Section 6 that, although the dummy `MIAMB`/`MKIAMB` are of high complexity theoretically, they usually perform well in multi-class prediction problems.

5. Benchmarking Study

This section makes a benchmarking study based on the data sets of six synthetic BNs. These data sets, generated by Tsamardinos et al. (2006) and Aliferis et al. (2010a), and the BNs are briefly described in Table 1. As Tsamardinos et al. (2006) and Aliferis et al. (2010a) stated, these BNs are representatives of a wide range of problem domains. Also, these BNs have different complexities (according to the number of nodes, the number of edges, maximal in-degree, maximal out-degree, and domain range). More details about the BNs and the used data sets are provided by Tsamardinos et al. (2006) and Aliferis et al. (2010a).

The following items are clarified before presenting the experimental results:

- *Measurements*: The primary measurement for the performance of an MB discovery algorithm used in our experiment is the weighted accuracy (WA), which is the average of the rate of true members and that of true nonmembers of an MB with respect to the truth. We also compute what we call the weighted precision (WP) as the average of the rate of true members and that of true nonmembers of an MB with respect to the output. In addition, we record the running time (RT) for every data set of each algorithm and for each BN. Here, RT refers to the single CPU time implemented on an Intel i7-3612QM 2.1 GHz and Windows 7 with 64 bits.

BN	Num. Nodes	Num. Edges	Maximal In-degree	Maximal Out-degree	Domain Range	Selected Targets	Sizes of Data Sets	Total RT (Hours)
Child10	200	257	2	7	2 ~ 6	$X_{131}, X_{132}, X_{98}, X_{194}$,	5×500	3.4297
						$X_{184}, X_{22}, X_{135}, X_{60}$	1×5000	8.2220
ALARM10	370	570	4	7	2 ~ 4	$X_{341}, X_{48}, X_{37}, X_{249}$,	5×500	4.6547
						$X_{209}, X_{188}, X_{192}, X_{161}$	1×5000	6.3721
Pigs	441	592	2	39	3 ~ 3	$X_{390}, X_{357}, X_{180}, X_{400}$,	5×500	11.7651
						$X_{199}, X_{241}, X_{228}, X_{176}$	1×5000	15.8443
Link	724	1125	3	14	2 ~ 4	$X_{369}, X_{293}, X_{303}, X_{457}$,	5×500	9.7932
						$X_{399}, X_{512}, X_{183}, X_{501}$	1×5000	16.1046
Lung Cancer	800	1476	4	28	2 ~ 3	$X_1, X_{416}, X_{345}, X_{641}$,	5×500	16.0301
						$X_{513}, X_{198}, X_{78}, X_{746}$	1×5000	16.3790
Gene	801	977	4	10	3 ~ 5	$X_{801}, X_{301}, X_{569}, X_{317}$,	5×500	17.2465
						$X_{185}, X_{622}, X_{516}, X_{577}$	1×5000	35.6790

Table 1: BNs and data sets.

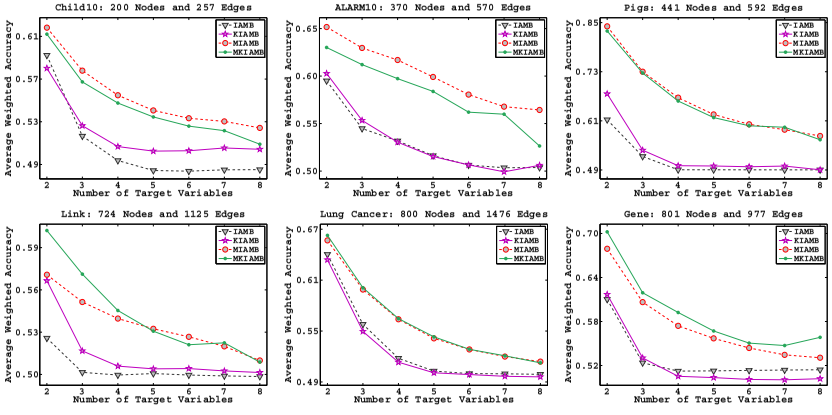


Figure 10: Average WA of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 500

- *Algorithms*: Four algorithms are used: `IAMB`, `KIAMB`, `MIAMB`, and `MKIAMB`. We take $K = 0.8$ as the randomization parameter in `KIAMB` and `MKIAMB` due to the following two reasons: (i) Peña et al. (2007, p. 227) asserted that $K \in [0.7, 0.9]$ performs best; and (ii) $K = 0.8$ is an appropriate tradeoff between WA (or WP) and RT.
- *Used CI Test*: Following the *practical operation* suggested in Subsection 4.3, we implemented the algorithms via the G^2 test, and found G^2 is suitable for small $|\boldsymbol{T}|$ but is not very suitable and even no longer works for large $|\boldsymbol{T}|$. Then, we used the experimental CMI-based test with a relatively rough $\varepsilon \approx \sqrt{|\boldsymbol{T}|} \cdot \varepsilon_0$, in which $\varepsilon_0 = 0.05$ if $n = 500$ and $\varepsilon_0 = 0.01$ if $n = 5000$; after that, we used the damped G^2 test by setting $\kappa = 5$. The results indicate both alternatives are desirable. Considering the association function, $f_{\boldsymbol{D}}^{(2)}$ defined in (4) corresponding to the CMI-based test, contains no the average number, v, of values that each variable takes, we may need to reselect ε_0 for a BN with a very different v. For these reasons, we eventually decided to use the damped G^2 test for the four algorithms in our experiment.
- *Data*: We use the data sets of sizes 500 and 5000, generated by Tsamardinos et al. (2006) and Aliferis et al. (2010a), which are available at `http://www.nyuinformatics.org/downloads/supplements/JMLR2009/index.html`.
- *Targets*: We employ eight of those variables selected by Aliferis et al. (2010a, p. 226) as the potential targets for each BN. See Table 1 for details. Then, $\boldsymbol{T}$ is any possible combination of k targets for $k = 2, \cdots, 8$.
- *Steps*: For each BN with eight selected targets, the steps of making simulation based on the data set of size 5000 are as follows: (a) for $k = 2, \cdots, 8$, call the four algorithms to obtain four MBs of $\boldsymbol{T} \triangleq \{T_{i_1}, \cdots, T_{i_k}\}$; (b) compute their WAs and WPs, and record the respective RTs; (c) take the average values of these $\binom{8}{k}$ WAs or WPs or RTs for each of the four algorithms. For the five data sets of size 500, each reported WA or WP or RT is the average value of the corresponding five results of an algorithm derived by (a) ~ (c) above.

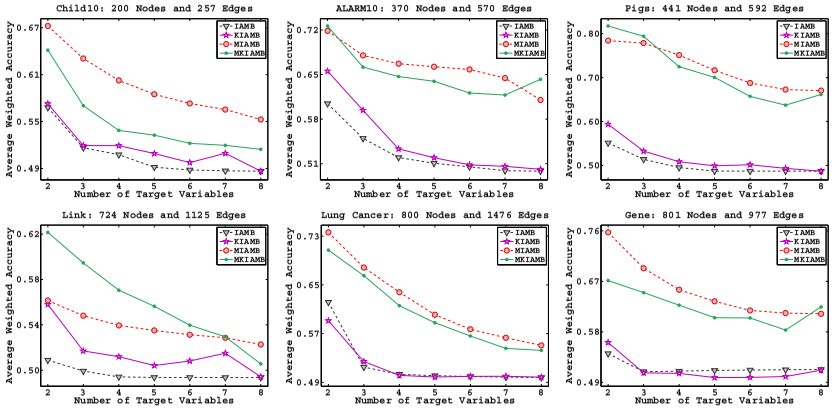


Figure 11: Average WA of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 5000

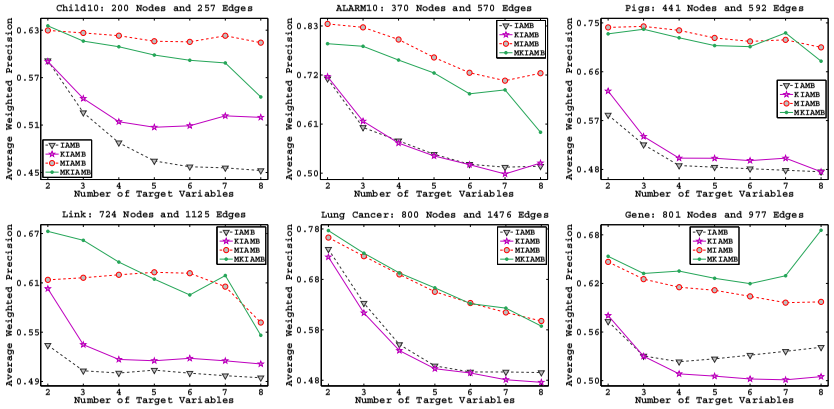


Figure 12: Average WP of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 500

According to the above description, we make computations with the aid of FullBNT (Murphy, 2007) and MIToolbox (Brown et al., 2012). The results of the WAs are presented in Figure 10 and Figure 11; the results of the WPs are given in Figure 12 and Figure 13; and the results of the RTs are shown in Figure 14, and Figure 15. The total RTs are presented in Table 1. By these figures, it is concluded that, on the whole, our `MIAMB` and `MKIAMB` have higher computational accuracies and lower time complexities than the existing `IAMB` and `KIAMB`.

Specifically, we have:

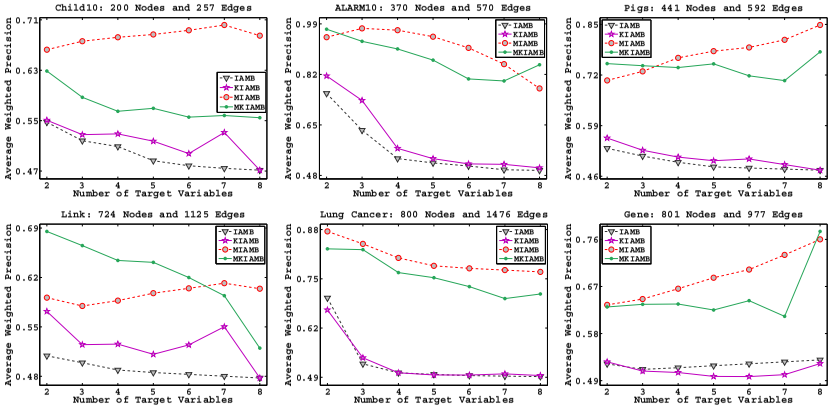


Figure 13: Average WP of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 5000

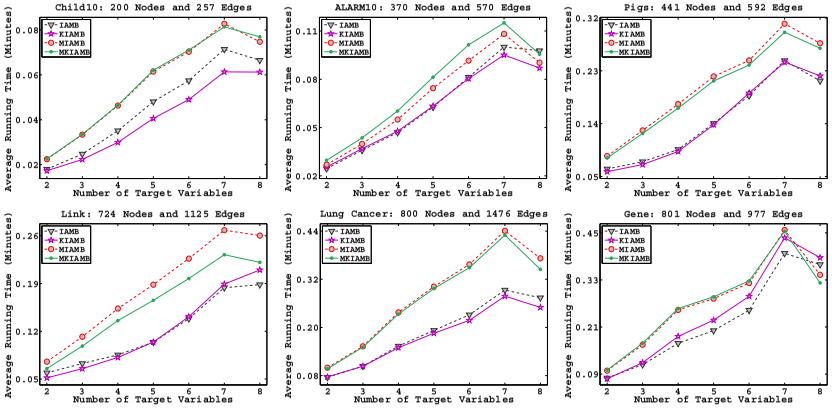


Figure 14: Average RT of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 500

(i) *Performance on WA*: (a) `MIAMB` and `MKIAMB` have larger WAs than `IAMB` and `KIAMB` for all the six BNs in any case of $|\boldsymbol{T}|$; (b) when $|\boldsymbol{T}|$ increases, WA declines quickly for `IAMB` and `KIAMB`, but it decreases gently for `MIAMB` and `MKIAMB`; and (c) the improvements of `MIAMB` and `MKIAMB` over `IAMB` and `KIAMB` tend to be gradually noticeable and then reduce slightly as $|\boldsymbol{T}|$ increases. The performance degradation along with the increase of $|\boldsymbol{T}|$ can be attributed to two possible aspects: one is that the local composition assumption may be more apt to be violated for a larger $|\boldsymbol{T}|$ because of synergy effects; and the other is that the assumption about

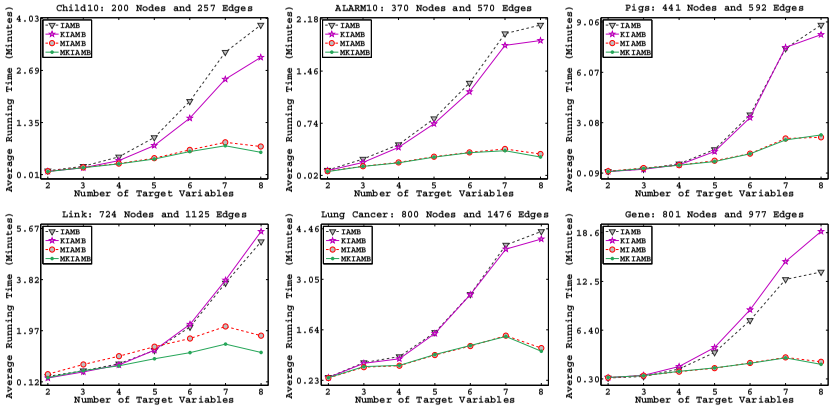


Figure 15: Average RT of the algorithms versus $|\boldsymbol{T}|$ with respect to the data sets of size 5000

the correctness of CI tests may also be more apt to be violated for a larger $|\boldsymbol{T}|$, due to the accumulation and propagation of the cascading errors (Bromberg and Margaritis, 2009). It is mentioned that (a)(b)(c) appear more evidently for the case of $n = 5000$ than for the case of $n = 500$.

(ii) *Performance on WP*: The similar interpretations to (a)(c) of (i) are valid.

(iii) *Performance on RT*: Here, we note that the real RT of an MB discovery algorithm is composed of two parts, in which the *part (I) is for CI tests*, and the *part (II) is for all other computations*. The part (I) is the major part used to measure the complexity of the MB discovery algorithm. Note also that the RT, $t_{n,q}$, of per CI test is linear to the sample size, n, and exponential to the number, q, of variables in the conditional set (see Subsection 4.4 for details).

This means that the part (II) of the real RT may dominate the part (I) if n is not large (for example, $n = 500$).

Let us now observe Figure 14 and Figure 15. First, both figures show the real RT that each algorithm needs is increasing along with the increase of $|\boldsymbol{T}|$. Also, Figure 14 indicates `MIAMB` and `MKIAMB` need slightly longer time to run than `IAMB` and `KIAMB`, because the running for CI tests is dominated by the running for all other computations in the case of a small sample size, while Figure 15 reveals that the real RTs of `IAMB` and `KIAMB` increase sharply as $|\boldsymbol{T}|$ increases and that the real RTs of `MIAMB` and `MKIAMB` increase slowly, just like the theoretical analyses about the complexities of the four algorithms show in Subsection 4.4.

In summary, the existing MB discovery algorithms, `IAMB` and `KIAMB`, can be approximately applied to the problem of MB discovery for multiple target variables when $|\boldsymbol{T}|$ is small, but they will perform poorly if $|\boldsymbol{T}|$ is moderately large. In comparison, our `MIAMB` and `MKIAMB` have higher accuracies and lower complexities for this problem, especially when $|\boldsymbol{T}|$ is large.

6. Application to FS in Multi-Class Prediction Problems

In this section, we apply the MB discovery for multiple targets to FS in multi-class prediction problems based on a real data set, HIVA. This data set is very challenging in WCCI 2006 (`http://www.modelselect.inf.ethz.ch`) and IJCNN 2007 (`http://www.agnostic.inf.ethz.ch`), because it contains many very unbalanced variables.

Let $T \in \boldsymbol{V}$ be a target variable taking values $\{1, \cdots, t\}$, $t \geqslant 3$. The multi-class prediction problem is to select features of T from $\boldsymbol{V} \setminus \{T\}$ such that T can be predicted as accurately as possible based on the chosen features. Let $\boldsymbol{T}^{(d)} \triangleq \left(T_1^{(d)}, \cdots, T_t^{(d)}\right)$ be the dummy version of T. Theoretically, $\boldsymbol{T}^{(d)}$ and T have the same MBs.

With these notations, the experiment is designed as follows:

- *Data*: HIVA contains 4229 data points and 1618 variables.
- *Targets*: In view of the fact that almost all variables in HIVA are binary, we randomly take k 2-class variables ($k = 2, \cdots, 5$) to create a merged 2^k-class target, T. Accordingly, we rearrange the original data to get a data set that is used only for FS of T. Repeat this step $n \triangleq 200$ times. Denote the resulting targets and their dummy versions by $T_1, \cdots, T_n$ and $\boldsymbol{T}_1^{(d)}, \cdots, \boldsymbol{T}_n^{(d)}$, respectively.
- *Algorithms*: We use the following 10 MB discovery algorithms to get the features for each T_j or $\boldsymbol{T}_j^{(d)} \triangleq \left(T_{j1}^{(d)}, \cdots, T_{jt_k}^{(d)}\right)$ with $t_k \triangleq 2^k$, $j = 1, \cdots, n$:
  a) `IAMB/KIAMB`-I: the `IAMB/KIAMB` algorithms working on T_j directly;
  b) `IAMB/KIAMB`-II: the `IAMB/KIAMB` algorithms working on $\boldsymbol{T}_j^{(d)}$ (that is, with $\boldsymbol{T}_j^{(d)}$ as its multiple targets);
  c) `MIAMB/MKIAMB`-I: the additivity based `MIAMB/MKIAMB` algorithms, taking the union of the outputs of `IAMB/KIAMB` with respect to $T_{ji}^{(d)}$ ($i = 1, \cdots, t_k$) as its output;
  d) `MIAMB/MKIAMB`-II: the MBS based `MIAMB/MKIAMB` algorithms, which are pseudo-coded in Algorithm 2;
  e) `MIAMB/MKIAMB`-III: the dummy `MIAMB/MKIAMB` algorithms, which take the union of the outputs of `IAMB/KIAMB` with respect to $T_{ji}^{(d)}$ ($i = 1, \cdots, t_k$) and removing redundant variables.
- *Classifier*: After making a number of preliminary experiments on the six benchmarking BNs, we found that the support vector machines (SVMs; implemented via `LibSVM` v3.22) perform the best in demonstrating the optimality of MBs for FS. This coincides with the assertion of Statnikov et al. (2013). Therefore, we use SVMs for our multi-class prediction problems. All the classifications are performed by 10-fold cross-validation.
- *Measurement* of an algorithm: For each target, the predictive quality of an MB is measured by the *balanced accuracy* defined as $\tau \triangleq \frac{1}{t_k}\sum_{\ell=1}^{t_k}\left(c_{\ell\ell}/\sum_{i=1}^{t_k} c_{i\ell}\right)$, where $\boldsymbol{C} \triangleq \left(c_{i\ell}\right)$ denotes the associated confusion matrix. As seen, τ is equal to one minus the *balanced error rate* used in WCCI 2006 and IJCNN 2007. We choose to use τ (instead of *ordinary accuracy*) because it trades off all values of the target in the sense that any unbalanced value (that the target takes) should not impact on the accuracy too much.[1] On the other hand, when two outputs of algorithms have the same total numbers of "true positives + true negatives", the balanced accuracy can identify the output that prefers to protect the scarce class as the better one, while the ordinary accuracy cannot. Finally, we compute the mean and standard deviation (std) of the n values of balanced accuracy, denoting them in the form of "(mean ± std)".

Problem	IAMB		MIAMB		
	I	II	I	II	III
4-class	0.9295 ± 0.082	0.9020 ± 0.113	0.9414 ± 0.068	0.9366 ± 0.073	0.9507 ± 0.057
8-class	0.9016 ± 0.102	0.8666 ± 0.133	0.9237 ± 0.091	0.9277 ± 0.087	0.9348 ± 0.077
16-class	0.8878 ± 0.105	0.8461 ± 0.143	0.9131 ± 0.087	0.9167 ± 0.086	0.9256 ± 0.075
32-class	0.8683 ± 0.113	0.8179 ± 0.157	0.9118 ± 0.078	0.9139 ± 0.078	0.9242 ± 0.067

Table 2: Balanced accuracy of `IAMB/MIAMB` algorithms in the form of "(mean ± std)".

Problem	KIAMB		MKIAMB		
	I	II	I	II	III
4-class	0.9283 ± 0.085	0.8972 ± 0.124	0.9281 ± 0.092	0.9444 ± 0.066	0.9501 ± 0.058
8-class	0.9007 ± 0.105	0.8631 ± 0.144	0.9245 ± 0.089	0.9280 ± 0.082	0.9340 ± 0.078
16-class	0.8886 ± 0.106	0.8494 ± 0.142	0.9159 ± 0.084	0.9168 ± 0.086	0.9263 ± 0.075
32-class	0.8687 ± 0.114	0.8241 ± 0.151	0.9111 ± 0.081	0.9151 ± 0.076	0.9239 ± 0.068

Table 3: Balanced accuracy of `KIAMB/MKIAMB` algorithms in the form of "(mean ± std)".

The experiment is then performed following the above procedures. Its results are summarized in Table 2 and Table 3. In these two tables, the backcolor indicates the performance of algorithms with black corresponding to the best while light blue to the worst. By the results, it can be seen that `MIAMB/MKIAMB` outperform `IAMB/KIAMB` in most situations. Specifically, we have:

- `IAMB/KIAMB` algorithms: `IAMB/KIAMB`-I are much more preferred than `IAMB/KIAMB`-II.
- `MIAMB/MKIAMB` algorithms: `MKIAMB`-I has almost equal performance to `KIAMB`-I in 4-class problems, and they performs slightly better than `IAMB/KIAMB`-I in 16- and 32-class problems;

1. For example, consider an unbalanced target T and its classification with the following two confusion matrices (the left is *extremely bad*, while the right is *very good*):

Test \ Truth	$T = 1$	$T = 2$
$T = 1$	948	49
$T = 2$	2	1

Test \ Truth	$T = 1$	$T = 2$
$T = 1$	899	0
$T = 2$	51	50

Then, we have: (a) for the left *bad* confusion matrix, the ordinary accuracy equals 94.90% (meaning it is impacted deeply by the unbalanced value 1 of T), while its balanced accuracy equals 50.89%; (b) for the right *good* confusion matrix, its ordinary accuracy also equals 94.90%, but its balanced accuracy equals 97.32%. This means balanced accuracy is more reasonable than ordinary accuracy to measure classification performance for a practical problem containing unbalanced variables (note that such problems may frequently occur in practice).

Null hypothesis (H_0)	Problem			
	4-class	8-class	16-class	32-class
MIAMB-I $\preccurlyeq$ IAMB-I	2.0349×10^{-4}	1.3225×10^{-7}	1.3816×10^{-10}	4.0634×10^{-19}
MIAMB-II $\preccurlyeq$ IAMB-I	1.4772×10^{-2}	9.3393×10^{-10}	4.0911×10^{-12}	8.2111×10^{-21}
MIAMB-III $\preccurlyeq$ IAMB-I	2.1276×10^{-10}	9.5876×10^{-16}	1.8800×10^{-20}	9.1061×10^{-30}
MKIAMB-I $\preccurlyeq$ KIAMB-I	0.6221	4.9365×10^{-9}	5.9117×10^{-12}	2.0651×10^{-19}
MKIAMB-I = KIAMB-I	0.7558	——	——	——
MKIAMB-II $\preccurlyeq$ KIAMB-I	2.2839×10^{-6}	5.9538×10^{-11}	5.4564×10^{-12}	2.7021×10^{-22}
MKIAMB-III $\preccurlyeq$ KIAMB-I	4.5857×10^{-10}	1.1261×10^{-15}	4.7802×10^{-20}	6.4206×10^{-30}

Table 4: p-values on paired t-test for comparison between MIAMB/MKIAMB and IAMB/KIAMB. Here, the notations are defined as follows: letting $\mathscr{A}_1$ and $\mathscr{A}_2$ be two algorithms and P be a problem, if $\mathscr{A}_1$ is better (in the sense of possessing higher accuracy) than $\mathscr{A}_2$ when used to solve P, we denote it by $\mathscr{A}_1 \succ \mathscr{A}_2$ (w.r.t. P); otherwise, we write it as $\mathscr{A}_1 \preccurlyeq \mathscr{A}_2$. In addition, we use $\mathscr{A}_1 = \mathscr{A}_2$ to denote $\mathscr{A}_1 \preccurlyeq \mathscr{A}_2$ and $\mathscr{A}_1 \succcurlyeq \mathscr{A}_2$.

MIAMB/MKIAMB-II significantly improve IAMB/KIAMB and even MIAMB/MKIAMB-I in most cases (although MIAMB/MKIAMB-II have larger std values than MIAMB/MKIAMB-I in some cases, the differences are slight). MIAMB/MKIAMB-III perform the best in all situations, with the highest mean values and the smallest std values.

Further, for any two algorithms, denote their balanced accuracy values as n (= 200) paired data points. Then, we can compute the p-values of *paired t-test* of associated hypotheses for one algorithm to be better (in the sense of possessing higher accuracy) than the other. The results are presented in Table 4. This table quantificationally shows the statistical significance of how much MIAMB/MKIAMB improve IAMB/KIAMB: in most cases, the improvement is more and more significant as the classification complex increases.

- The performance of each algorithm degrades with the increase of classification complexity. However, the degenerations of MIAMB/MKIAMB are slower than that of IAMB/KIAMB.

To compare IAMB/KIAMB and MIAMB/MKIAMB detailedly, we take the results of IAMB/KIAMB-I and MIAMB/MKIAMB-III to make a further analysis. For the 4-class prediction problem, denote the results of IAMB-I and MIAMB-III by $\tau_i^{(\mathtt{IAMB})}$ and $\tau_i^{(\mathtt{MIAMB})}$ for $i = 1, \cdots, n$, and draw them in (a) of Figure 16. Put

$$\begin{aligned} \boldsymbol{I}_1 &= \left\{i \in \{1, \cdots, n\} : \tau_i^{(\mathtt{IAMB})} > \tau_i^{(\mathtt{MIAMB})}\right\}, \\ \boldsymbol{I}_2 &= \left\{i \in \{1, \cdots, n\} : \tau_i^{(\mathtt{IAMB})} = \tau_i^{(\mathtt{MIAMB})}\right\}, \\ \boldsymbol{I}_3 &= \left\{i \in \{1, \cdots, n\} : \tau_i^{(\mathtt{IAMB})} < \tau_i^{(\mathtt{MIAMB})}\right\}. \end{aligned}$$

Draw the scatters of $\tau_i^{(\mathtt{IAMB})}$ and $\tau_i^{(\mathtt{MIAMB})}$ for $i \in \boldsymbol{I}_j$ in (a_j) of Figure 16. In addition, the information about (mean ± std) of IAMB-I vs that of MIAMB-III is annotated in each title. For other three K-class prediction problems (K = 8, 16, 32), repeat the above steps to get the scatters drawn in the other subplots of Figure 16. Similarly, Figure 17 draws the results of KIAMB-I versus MKIAMB-III.

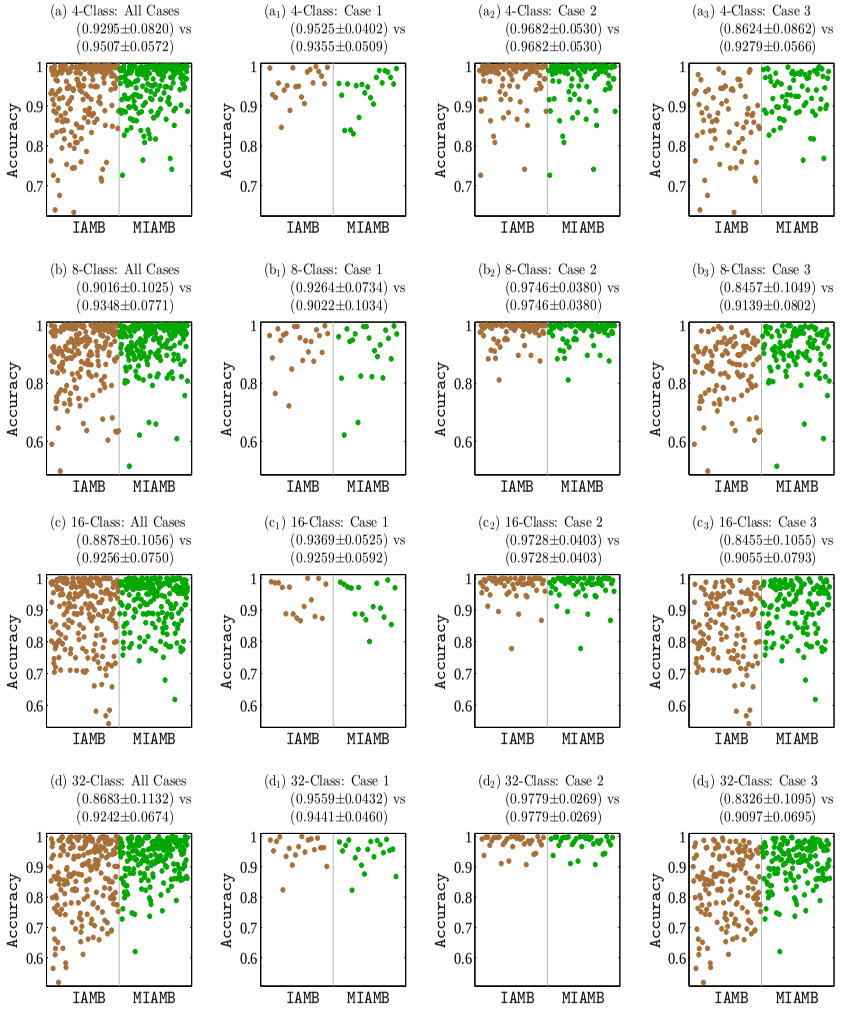


Figure 16: Balanced accuracy results on IAMB/MIAMB algorithms applied to 200 K-class prediction problems (K = 4, 8, 16, 32): the subplots in the first column for all the 200 results; the ones in the second column for the results that IAMB performs better than MIAMB; the ones in the third column for the results that IAMB and MIAMB perform equally well; the ones in the last column for the results that MIAMB performs better than IAMB.

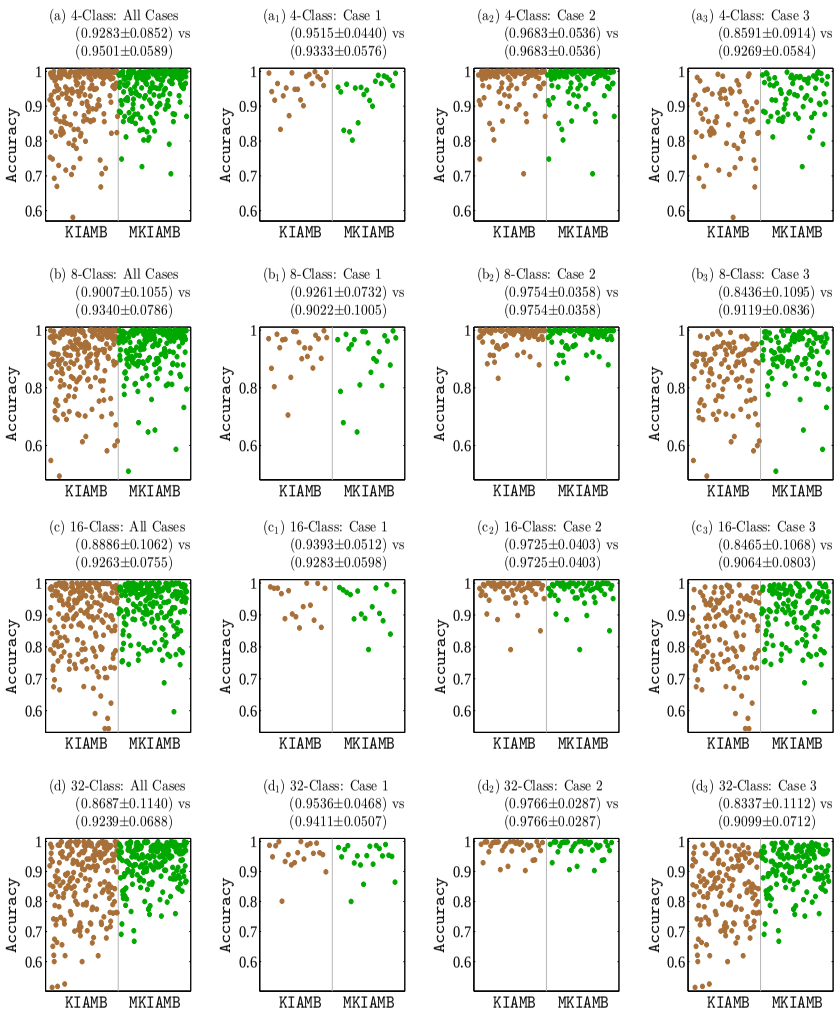


Figure 17: Balanced accuracy results on KIAMB/MKIAMB applied to 200 K-class prediction problems (K = 4, 8, 16, 32): the subplots in the first column for all the 200 results; the ones in the second column for the results that KIAMB performs better than MKIAMB; the ones in the third column for the results that KIAMB and MKIAMB perform equally well; the ones in the last column for the results that MKIAMB performs better than KIAMB.

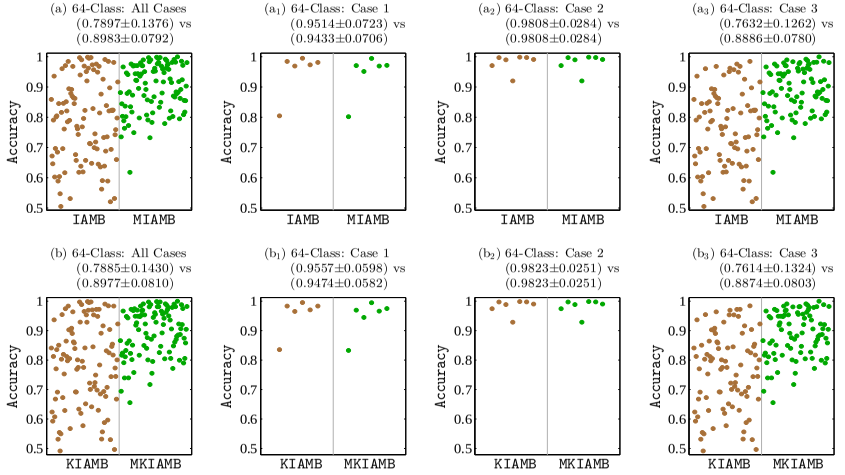


Figure 18: Balanced accuracy results on IAMB/MIAMB and KIAMB/MKIAMB applied to 100 64-class prediction problems: MIAMB/MKIAMB can improve IAMB/KIAMB substantially.

Figure 16 indicates that: (i) In most cases, MIAMB can improve IAMB to various degrees: MIAMB has higher mean values of balanced accuracy and smaller std values as well. Although there are a few of situations in which IAMB performs better than MIAMB, the difference of performance between them is very slight. In addition, there are some situations in which the two algorithms perform equally well (with very high mean and small std). (ii) MIAMB is more resistant to the classification complexity than IAMB: the improvements of MIAMB over IAMB become more and more visible with the increase of K (from 4 to 32). Figure 17 shows similar conclusions.

In brief, an FS problem for multi-class prediction can be transformed into a problem of MB discovery for multiple targets, and then get a more efficient solution. This idea may be particularly useful when the classification complexity is high or very high. To check this imagination, we apply the same procedures to 100 64-class prediction problems (also taken from the HIVA data set). The results are summarized in Figure 18, in which (a) and (a_j) are for IAMB/MIAMB while (b) and (b_j) are for KIAMB/MKIAMB, $j = 1, 2, 3$. By the figure, the improvement (nearly 14% on accuracy) of MIAMB/MKIAMB algorithms over IAMB/KIAMB is really desirable in the case of high classification complexity.

Finally, we apply LibSVM and the random forest (RF) algorithm (Breiman, 2001) to the whole HIVA data without any FS, considering LibSVM is of high classification performance while RF is a state-of-the-art FS algorithm. The results can be served as a baseline to see why FS (or equivalently, MB discovery) is necessary for a complex classification problem. Recall that HIVA contains many unbalanced variables, which enhance the classification complexity. Figure 19 draws the 95% confidence bands of LibSVM and FS, respectively.

By the figure, it follows that:

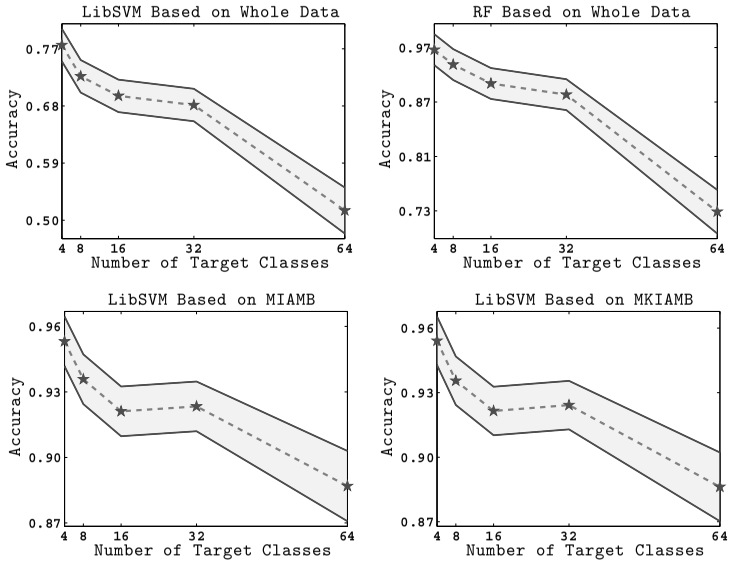


Figure 19: The 95% confidence bands of LibSVM and RF.

- Without any FS, LibSVM performs undesirably in all situations. This may be because too many noisy variables can lead to masking effects upon those unbalanced features such that LibSVM cannot classify targets expectedly. This shows the necessity of FS. In other words, LibSVM may not be suitable for some *high-dimension* problems, especially when there are many unbalanced variables.

- Without any FS, RF performs quite well when the classification complexity is not very high. However, with the increase of classification complexity, the performance of RF decreases gradually and then sharply. In other words, RF may not be suitable for those problems with too *high complexity*, especially when there are many unbalanced variables.

To observe why this happens, we check the results and then randomly take some targets (with extraordinarily low accuracy) to implement LibSVM and RF again by appropriately adjusting the algorithmic setting of LibSVM and increasing the number of trees of RF from 100 to 1000. However, the results change very little.

To make an intuitive comparison, the 95% confidence bands of MIAMB/MKIAMB-based LibSVM are also drawn in Figure 19. As seen, all methods degenerate with the increase of the classification complexity, but our methods degenerate far slower than LibSVM/RF based on the whole data. In a word, MB discovery (or equivalently, FS) is important to make classification, especially when the problem is of *high dimension* and of *high complexity*.

7. Concluding Remarks

In this paper, we considered the problem of Mb and MB of multiple variables. We first addressed their additivity under the local intersection assumption, and then studied this problem in the general case. The two algorithms that we proposed , MIAMB and MKIAMB, were proven to be correct under the local composition assumption with respect to single targets. The benchmarking study based on six synthetic BNs showed that MIAMB and MKIAMB have higher accuracies and lower complexities than the existing IAMB and KIAMB.

Before ending this paper, we present four concluding remarks as follows:

(i) The first remark concerns a method of using MIAMB and MKIAMB to find an MB for a single variable. Such an idea is motivated by the following two aspects: (a) the local composition assumption may be violated in practice, and if this is the case, IAMB and KIAMB may perform not very well in MB discovery even for a single variable; (b) randomness of a data set may result in a violation to the assumption that all the CI tests involved are correct. Naturally, it is useful to take a remedy for these two situations. One remedial strategy is described as follows: letting $T \in \boldsymbol{V}$ be the target variable, and $\boldsymbol{M}_1$ be a potential MB discovered by IAMB or KIAMB, take $T_0 \triangleq \arg\max_{X \in \boldsymbol{M}_1} J_{\boldsymbol{D}}^{(\ell)}(T; X \mid \boldsymbol{M}_1 \setminus \{X\})$ as a co-target of T for $\ell = 1$ or 2 or 3; then, employ MIAMB or MKIAMB to find a potential MB for $\{T, T_0\}$, saying $\boldsymbol{M}_2$. Finally, refine $\{T_0\} \cup \boldsymbol{M}_2$ to obtain $\boldsymbol{M}$, by virtue of the shrinking phase of IAMB or KIAMB, since this phase needs no the local composition precondition.

(ii) Our MIAMB and MKIAMB contain an ordering τ, which may affect the RT and even the WA or WP. A question arises here: is there an optimal selection of τ such that MIAMB or MKIAMB has the highest accuracy and the lowest complexity?

(iii) All the considered algorithms (IAMB, KIAMB, MIAMB, and MKIAMB) need the local composition assumption to theoretically guarantee their correctness. However, this precondition may be violated in practice and in this case only an approximate MB can be obtained by means of one of the above algorithms. Subsection 4.1 provides a potential remedy. We note that MIAMB and MKIAMB transform the problem of MB discovery for multiple targets into the ones for single targets. This idea provides a facilitation to use some stochastic optimization methods such as the particle swarm optimization algorithm (Kennedy and Eberhart, 1995, 1997).

(iv) In Subsection 3.3, we provided a method for MB discovery of a *complex* single variable based on an MB discovery of some *simple* multiple variables. Let us now explain why this transformation method is efficient. With the notations used in Subsection 3.3, let

$$\mathrm{MB}_T = \mathrm{MB}_{\boldsymbol{T}^{(d)}} \triangleq \boldsymbol{M}.$$

Then, a variable X can enter and stay in $\boldsymbol{M}$ (in the sense of MB_T) if $T \not\perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\}$. On the other hand, by Theorem 5, X can enter and stay in $\boldsymbol{M}$ (in the sense of $\mathrm{MB}_{\boldsymbol{T}^{(d)}}$) only if $T_j^{(d)} \not\perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\}$ holds for some j. That is, we need to test the following two pairs of hypotheses:

$$H_0^{(1)}: \quad T \perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\} \quad \leftrightarrow \quad H_1^{(1)}: \quad T \not\perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\};$$
$$H_0^{(2)}: \quad T_j^{(d)} \perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\} \quad \leftrightarrow \quad H_1^{(2)}: \quad T_j^{(d)} \not\perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\}.$$

Clearly, when T is high-dimensional, *the test for* $H_0^{(1)} \leftrightarrow H_1^{(1)}$ requires far more data points than *that for* $H_0^{(2)} \leftrightarrow H_1^{(2)}$, since the test statistic for the first pair of hypotheses contains far more free parameters than that for the second. In addition, the transformation from T to $\boldsymbol{T}^{(d)}$ can be easily made, with almost no running time. This explains why Theorem 5 is useful.

Acknowledgments

The authors are very grateful to the four anonymous reviewers and Prof. Marina Meila and Prof. Joris Mooij for their valuable comments and constructive suggestions which result in the present version. Thanks also to Prof. Kevin Murphy for all of his kind help.

This work was supported by the National Natural Science Foundation of China (61374183, 51472117, 51535005, 51675212), the Research Fund of State Key Laboratory of Mechanics and Control of Mechanical Structures (MCMS-0417G02, MCMS-0417G03), the Fundamental Research Funds for the Central Universities (NP2017101, NC2018001), the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and the Open Fund for the Key Laboratory for Traffic and Transportation Security of Jiangsu Province.

Appendix A. Pseudo Codes for IAMB and KIAMB

This appendix presents the pseudo codes for IAMB and KIAMB. We mention here that, in Line 4 of KIAMB, $\boldsymbol{M}_2$ denotes a random subset of $\boldsymbol{M}_1$ with size $|\boldsymbol{M}_2| = \max\{1, \lfloor |\boldsymbol{M}_1| \cdot K \rfloor\}$.

Algorithm 3: IAMB and KIAMB

	Procedure: $\boldsymbol{M} \leftarrow$ IAMB$(\boldsymbol{D}; \boldsymbol{T})$	**Procedure:** $\boldsymbol{M} \leftarrow$ KIAMB$(\boldsymbol{D}; \boldsymbol{T}; K)$
	Input: $\boldsymbol{D}$ is a data matrix; $\boldsymbol{T}$ is a set of target variables.	**Input:** Besides $\{\boldsymbol{D},\boldsymbol{T}\}$ as in IAMB, $K \in [0, 1]$ is a randomization parameter.
	Output: an MB, $\boldsymbol{M}$, of $\boldsymbol{T}$.	**Output:** an Mb, $\boldsymbol{M}$, of $\boldsymbol{T}$.
	//Forward: Growing Phase	//Forward: Growing Phase
1	$\boldsymbol{M} \leftarrow \emptyset$	$\boldsymbol{M} \leftarrow \emptyset$
2	**while** $\boldsymbol{M}$ has changed **do**	**while** $\boldsymbol{M}$ has changed **do**
3	$Y \leftarrow \arg\max_{X \in \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T}} f_{\boldsymbol{D}}(\boldsymbol{T}; X \mid \boldsymbol{M})$	**if** $\boldsymbol{M}_1 \leftarrow \{X \in \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T} : \boldsymbol{T} \not\perp\!\!\!\perp X \mid \boldsymbol{M}\} \neq \emptyset$ **then**
4	**if** $\boldsymbol{T} \not\perp\!\!\!\perp Y \mid \boldsymbol{M}$ **then**	$Y \leftarrow \arg\max_{X \in \boldsymbol{M}_2} f_{\boldsymbol{D}}(\boldsymbol{T}; X \mid \boldsymbol{M})$
5	$\boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{Y\}$	$\boldsymbol{M} \leftarrow \boldsymbol{M} \cup \{Y\}$
6	**end**	**end**
7	**end**	**end**
	//Backward: Shrinking Phase	//Backward: Shrinking Phase
8	**foreach** $X \in \boldsymbol{M}$ **do**	**foreach** $X \in \boldsymbol{M}$ **do**
9	**if** $\boldsymbol{T} \perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\}$ **then**	**if** $\boldsymbol{T} \perp\!\!\!\perp X \mid \boldsymbol{M} \setminus \{X\}$ **then**
10	$\boldsymbol{M} \leftarrow \boldsymbol{M} \setminus \{X\}$	$\boldsymbol{M} \leftarrow \boldsymbol{M} \setminus \{X\}$
11	**end**	**end**
12	**end**	**end**
13	**return** $\boldsymbol{M}$	**return** $\boldsymbol{M}$

Appendix B. Proofs

In this appendix, we give the proofs of the theoretical results.

Lemma 1 *The intersection property holds if and only if no information equivalence occurs.*

Proof Equivalently, we show that the intersection property is violated if and only if information equivalence occurs. The sufficiency holds clearly. To prove the necessity, we assume the intersection property is violated, that is, there are $\boldsymbol{T}$, $\boldsymbol{X}$, $\boldsymbol{Y}$, and $\boldsymbol{Z}$ such that $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \mid \boldsymbol{Z} \cup \boldsymbol{Y}$, and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z} \cup \boldsymbol{X}$, but $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X} \cup \boldsymbol{Y} \mid \boldsymbol{Z}$. Then, we can show $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X} \mid \boldsymbol{Z}$. In fact, if $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \mid \boldsymbol{Z}$, then combined with $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z} \cup \boldsymbol{X}$ and the contraction property, it concludes $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{X} \cup \boldsymbol{Y} \mid \boldsymbol{Z}$, which contradicts $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{X} \cup \boldsymbol{Y} \mid \boldsymbol{Z}$. Similarly, we can show $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{Y} \mid \boldsymbol{Z}$. Therefore, $\boldsymbol{X}$ and $\boldsymbol{Y}$ are information equivalent with respect to $\boldsymbol{T}$ given $\boldsymbol{Z}$. That is, information equivalence occurs. ■

Lemma 2 *For $\boldsymbol{T} \subseteq \boldsymbol{V}$, assume the type-II local condition holds. Then $\boldsymbol{T}$ has a unique MB.*

Proof Suppose $\boldsymbol{T}$ has two different MBs, $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$. Putting $\boldsymbol{M}_{12} \triangleq \boldsymbol{M}_1 \setminus \boldsymbol{M}_2$, $\boldsymbol{M}_{21} \triangleq \boldsymbol{M}_2 \setminus \boldsymbol{M}_1$, and $\boldsymbol{M} \triangleq \boldsymbol{M}_1 \cap \boldsymbol{M}_2 \subsetneq \boldsymbol{M}_i$ for $i = 1, 2$, we have

$$\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T} \mid \boldsymbol{M}_1 \;\Rightarrow\; \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T} \mid \boldsymbol{M} \cup \boldsymbol{M}_{12}, \tag{8}$$

$$\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_2 \setminus \boldsymbol{T} \mid \boldsymbol{M}_2 \;\Rightarrow\; \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_2 \setminus \boldsymbol{T} \mid \boldsymbol{M} \cup \boldsymbol{M}_{21}. \tag{9}$$

Now we show $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{M}_{12} \mid \boldsymbol{M}$. In fact, suppose we have $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{M}_{12} \mid \boldsymbol{M}$. This combined with (8) implies $\boldsymbol{T} \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}) \cup \boldsymbol{M}_{12} \mid \boldsymbol{M}$, in view of the contraction property. Equivalently, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T} \mid \boldsymbol{M}$, which contradicts the fact that $\boldsymbol{M}_1$ is an MB of $\boldsymbol{T}$, since $\boldsymbol{M} \subsetneq \boldsymbol{M}_1$. Hence, $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{M}_{12} \mid \boldsymbol{M}$. Similarly, we can show $\boldsymbol{T} \not\perp\!\!\!\perp \boldsymbol{M}_{21} \mid \boldsymbol{M}$. On the other hand, by the decomposition property, (9) and (8) indicate $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{M}_{12} \mid \boldsymbol{M} \cup \boldsymbol{M}_{21}$ and $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{M}_{21} \mid \boldsymbol{M} \cup \boldsymbol{M}_{12}$, respectively. Therefore, $\boldsymbol{M}_{12}$ and $\boldsymbol{M}_{21}$ are information equivalent with respect to $\boldsymbol{T}$ conditioned on $\boldsymbol{M}$. This contradicts the precondition. The uniqueness of MB of $\boldsymbol{T}$ is shown under the type-II local condition. ■

Theorem 2 (Additivity of Mb) *Let $(\mathbb{G}, \mathbb{P})$ be a BN over $\boldsymbol{V}$. The following two statements hold:*

(i) *Let $\boldsymbol{M}_i$ be an Mb of $\boldsymbol{T}_i \subseteq \boldsymbol{V}$ for $i = 1, 2$, and assume $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ satisfies the local intersection assumption. Then, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2)$ is an Mb of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$.*

(ii) *Let $\boldsymbol{M}_i$ be an Mb of $T_i \in \boldsymbol{V}$ for $i = 1, \cdots, k$, and assume $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$ satisfies the local intersection assumption. Then, $\bigcup_{i=1}^{k} \boldsymbol{M}_i \setminus \boldsymbol{T}$ is an Mb of $\boldsymbol{T}$.*

Proof It suffices to prove (i), since (ii) is a direct consequence of (i) using induction on the number of variables involved in $\boldsymbol{T}$.

Denote $\boldsymbol{V}_0 = \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ with $\boldsymbol{N} = \boldsymbol{N}_1 \cup \boldsymbol{N}_2$, in which $\boldsymbol{N}_1 = \boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ and $\boldsymbol{N}_2 = \boldsymbol{M}_2 \setminus \boldsymbol{T}_1$. Note that $\boldsymbol{N}$ can also be expressed as $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}$. First, we prove the following CI relationship, by means of the graphoid properties:

$$\boldsymbol{V}_0 \perp\!\!\!\perp \boldsymbol{T}_1 \mid \boldsymbol{N} \cup \boldsymbol{T}_2. \tag{10}$$

In fact, with the above notations, it is readily justified that $\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 = \boldsymbol{V}_0 \cup [(\boldsymbol{N}_2 \cup \boldsymbol{T}_2) \setminus \boldsymbol{M}_1]$. On the other hand, $\boldsymbol{M}_1$ is an Mb of $\boldsymbol{T}_1$. Therefore, $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \mid \boldsymbol{M}_1$, and thus we obtain $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V}_0 \cup [(\boldsymbol{N}_2 \cup \boldsymbol{T}_2) \setminus \boldsymbol{M}_1] \mid \boldsymbol{M}_1$. By the weak union property, $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V}_0 \mid \boldsymbol{M}_1 \cup [(\boldsymbol{N}_2 \cup \boldsymbol{T}_2) \setminus \boldsymbol{M}_1]$. This means (10) holds, since $\boldsymbol{M}_1 \cup [(\boldsymbol{N}_2 \cup \boldsymbol{T}_2) \setminus \boldsymbol{M}_1] = \boldsymbol{N} \cup \boldsymbol{T}_2$.

Similarly, $\boldsymbol{V}_0 \perp\!\!\!\perp \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{T}_1$, which combined with (10) indicates

$$\boldsymbol{V}_0 \perp\!\!\!\perp \boldsymbol{T} \,|\, \boldsymbol{N}$$

by the local intersection assumption. Or equivalently, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T} \,|\, \boldsymbol{N}$. That is, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} = \boldsymbol{N}$ is an Mb of $\boldsymbol{T}$. The proof is completed. ■

Remark 1 *In the case of either $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ or $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, the conclusion of* (i) *in Theorem 2 holds without requiring the local intersection assumption.*

Proof If $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ but $\boldsymbol{T}_2 \nsubseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, $\boldsymbol{M}_1 \cup \boldsymbol{M}_2$ is then an Mb of $\boldsymbol{T}_1$ according to the weak union property whereas $(\boldsymbol{M}_1 \setminus \boldsymbol{T}_2) \cup \boldsymbol{M}_2$ is an Mb of $\boldsymbol{T}_2$. Equivalently, we have

$$\begin{aligned}\boldsymbol{T}_1 &\perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1 \cup \boldsymbol{M}_2,\\ \boldsymbol{T}_2 &\perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{M}_1 \setminus \boldsymbol{T}_2) \cup \boldsymbol{M}_2] \setminus \boldsymbol{T}_2 \,|\, (\boldsymbol{M}_1 \setminus \boldsymbol{T}_2) \cup \boldsymbol{M}_2.\end{aligned}$$

By means of the contraction property and the decomposition property, it is seen that

$$\begin{aligned}\boldsymbol{V} \setminus [(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2] \setminus \boldsymbol{T} &\perp\!\!\!\perp \boldsymbol{T}_1 \,|\, [(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2] \cup \boldsymbol{T}_2,\\ \boldsymbol{V} \setminus [(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2] \setminus \boldsymbol{T} &\perp\!\!\!\perp \boldsymbol{T}_2 \,|\, (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2,\end{aligned}$$

so $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2] \setminus \boldsymbol{T} \,|\, (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2$. That is, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2) = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2$ is an Mb of $\boldsymbol{T}$. If $\boldsymbol{T}_1 \nsubseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ but $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, we can similarly show

$$(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2) = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_1$$

is an Mb of $\boldsymbol{T}$. Finally, if $\boldsymbol{T}_1 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_2$ and $\boldsymbol{T}_2 \subseteq \boldsymbol{V} \setminus \boldsymbol{M}_1$, imposing decomposition on

$$\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1 \cup \boldsymbol{M}_2$$

and weak union on $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{M}_1 \cup \boldsymbol{M}_2$, we get

$$\begin{aligned}\boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} &\perp\!\!\!\perp \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1 \cup \boldsymbol{M}_2,\\ \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} &\perp\!\!\!\perp \boldsymbol{T}_2 \,|\, (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \cup \boldsymbol{T}_1.\end{aligned}$$

By the contraction property, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} \,|\, \boldsymbol{M}_1 \cup \boldsymbol{M}_2$. That is,

$$(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2) = \boldsymbol{M}_1 \cup \boldsymbol{M}_2$$

is an Mb of $\boldsymbol{T}$. The conclusion is proved. ■

Theorem 3 (Additivity of MB) *Let $(\mathbb{G}, \mathbb{P})$ be a BN over $\boldsymbol{V}$. The following two statements hold:*

(i) *Assume $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ satisfies the local intersection assumption. Let $\boldsymbol{M}_i$ be the unique MB of $\boldsymbol{T}_i$ for $i = 1, 2$. Then, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus (\boldsymbol{T}_1 \cup \boldsymbol{T}_2)$ is the unique MB of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$.*

(ii) *Assume $\boldsymbol{T} \triangleq \{T_1, \cdots, T_k\}$ satisfies the local intersection assumption. Let $\boldsymbol{M}_i$ be the unique MB of T_i for $i = 1, \cdots, k$. Then, $\bigcup_{i=1}^{k} \boldsymbol{M}_i \setminus \boldsymbol{T}$ is the uniqe MB of $\boldsymbol{T}$.*

Proof We need only to prove (i), since (ii) is a direct consequence of (i).

Denote $\boldsymbol{N}_1 = \boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ and $\boldsymbol{N}_2 = \boldsymbol{M}_2 \setminus \boldsymbol{T}_1$. By Theorem 2, $(\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} = \boldsymbol{N}_1 \cup \boldsymbol{N}_2$ is an Mb of $\boldsymbol{T}$. Therefore, it suffices to prove the minimality of $\boldsymbol{N}_1 \cup \boldsymbol{N}_2$, based on Lemma 2. In fact, let $\boldsymbol{N}_0$ be any Mb of $\boldsymbol{T}$ which is a subset of $\boldsymbol{N}_1 \cup \boldsymbol{N}_2$. Note that $\boldsymbol{T}_i \cap \boldsymbol{N}_0 = \emptyset$ for $i = 1, 2$. Denote now $\boldsymbol{M} = \boldsymbol{N}_0 \cup (\boldsymbol{M}_1 \setminus \boldsymbol{N}_1) = \boldsymbol{N}_0 \cup (\boldsymbol{M}_1 \cap \boldsymbol{T}_2)$. It follows that

- $\boldsymbol{M}_1$ is the MB of $\boldsymbol{T}_1$: This implies $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1$, or equivalently, we have

  $$\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \,|\, (\boldsymbol{M}_1 \cap \boldsymbol{M}) \cup (\boldsymbol{M}_1 \setminus \boldsymbol{M}), \tag{11}$$

  in view of $\boldsymbol{M}_1 = (\boldsymbol{M}_1 \cap \boldsymbol{M}) \cup (\boldsymbol{M}_1 \setminus \boldsymbol{M})$.
- $\boldsymbol{N}_0$ is an Mb of $\boldsymbol{T}$: Equivalently, we have $\boldsymbol{T}_1 \cup \boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{N}_0 \setminus \boldsymbol{T}_2 \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{N}_0$, which gives

  $$\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N}_0 \cup \boldsymbol{T}_2) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{N}_0 \cup \boldsymbol{T}_2,$$

  according to the weak union property, and thus $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M} \cup \boldsymbol{T}_2) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M} \cup \boldsymbol{T}_2$ in view of $\boldsymbol{N}_0 \cup \boldsymbol{T}_2 = \boldsymbol{M} \cup \boldsymbol{T}_2$, or equivalently we have $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T}_1 \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{M} \cup \boldsymbol{T}_2$. By the self-conditioning property, this leads to $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cap \boldsymbol{M}) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M} \cup \boldsymbol{T}_2$, Therefore,

  $$\boldsymbol{T}_1 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1) \cup (\boldsymbol{M}_1 \setminus \boldsymbol{M}) \,|\, \boldsymbol{M} \cup \boldsymbol{T}_2,$$

  in terms of $\boldsymbol{V} \setminus (\boldsymbol{M}_1 \cap \boldsymbol{M}) \setminus \boldsymbol{T}_1 = (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1) \cup (\boldsymbol{M}_1 \setminus \boldsymbol{M})$. By the weak union property, this indicates $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{M}_1 \setminus \boldsymbol{M} \,|\, (\boldsymbol{M} \cup \boldsymbol{T}_2) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1)$. Consequently,

  $$\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{M}_1 \setminus \boldsymbol{M} \,|\, (\boldsymbol{M}_1 \cap \boldsymbol{M}) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1), \tag{12}$$

  due to $(\boldsymbol{M} \cup \boldsymbol{T}_2) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1) = (\boldsymbol{M}_1 \cap \boldsymbol{M}) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1)$.

By the local intersection property, (11)(12) indicate $\boldsymbol{T}_1 \perp\!\!\!\perp (\boldsymbol{M}_1 \setminus \boldsymbol{M}) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1) \,|\, \boldsymbol{M}_1 \cap \boldsymbol{M}$, so

$$\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cap \boldsymbol{M}) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1 \cap \boldsymbol{M},$$

since $(\boldsymbol{M}_1 \setminus \boldsymbol{M}) \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1) = \boldsymbol{V} \setminus (\boldsymbol{M}_1 \cap \boldsymbol{M}) \setminus \boldsymbol{T}_1$. Hence, $\boldsymbol{M}_1 \cap \boldsymbol{M}$ ($\subseteq \boldsymbol{M}_1$) is an Mb of $\boldsymbol{T}_1$. On the other hand, $\boldsymbol{M}_1$ is the MB of $\boldsymbol{T}_1$ and thereby $\boldsymbol{M}_1 \cap \boldsymbol{M} = \boldsymbol{M}_1$, or equivalently,

$$\boldsymbol{N}_1 \cup (\boldsymbol{M}_1 \cap \boldsymbol{T}_2) = \boldsymbol{M}_1 \subseteq \boldsymbol{M} = \boldsymbol{N}_0 \cup (\boldsymbol{M}_1 \cap \boldsymbol{T}_2),$$

which means $\boldsymbol{N}_1 \subseteq \boldsymbol{N}_0$. In a similar fashion, $\boldsymbol{N}_2 \subseteq \boldsymbol{N}_0$. Combined with $\boldsymbol{N}_0 \subseteq \boldsymbol{N}_1 \cup \boldsymbol{N}_2$, the expected relationship $\boldsymbol{N}_0 = \boldsymbol{N}_1 \cup \boldsymbol{N}_2$ follows. This indicates that $\boldsymbol{N}_1 \cup \boldsymbol{N}_2$ is an MB of $\boldsymbol{T}$. The proof is completed, since Lemma 2 shows the uniqueness of MB under the local intersection assumption. ■

Theorem 4 *Assume $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$. Then, the following statements are equivalent:*

(i) *$\boldsymbol{S}$ is an MbS;*

(ii) *$\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \min_{\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}} \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}')$;*

(iii) *$\mathbb{I}(\boldsymbol{T}; \boldsymbol{S} \,|\, \boldsymbol{N}) = \max_{\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}} \mathbb{I}(\boldsymbol{T}; \boldsymbol{S}' \,|\, \boldsymbol{N})$;*

(iv) *$\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_1$ in $\boldsymbol{V} \setminus \boldsymbol{T}_2$ (or $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$).*

In addition, if $\boldsymbol{S}$ is an MbS, then it is also an MBS if and only if $\boldsymbol{T}_1 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ or $\boldsymbol{T}_2 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ holds for any $Y \in \boldsymbol{S}$.

Proof We first prove (i) $\Leftrightarrow$ (ii). Put $\boldsymbol{Q} = \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T}$. First, note that $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{M}_1$ since $\boldsymbol{M}_1$ is an Mb of $\boldsymbol{T}_1$. In other words, $a \triangleq \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{Q} \cup (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \setminus \boldsymbol{Q}) \,|\, \boldsymbol{M}_1) = 0$. By the chain rule for CMI (Cover and Thomas, 2006), we have $\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \setminus \boldsymbol{Q} \,|\, \boldsymbol{M}_1) = 0$. It follows that

$$\begin{aligned} a &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \setminus \boldsymbol{Q} \,|\, \boldsymbol{M}_1) + \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{Q} \,|\, (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \setminus \boldsymbol{Q}) \cup \boldsymbol{M}_1) \\ &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{Q} \,|\, (\boldsymbol{V} \setminus \boldsymbol{M}_1 \setminus \boldsymbol{T}_1 \setminus \boldsymbol{Q}) \cup \boldsymbol{M}_1) \\ &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_2) \\ &= \mathbb{I}(\boldsymbol{T}; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) - \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) \qquad (13) \\ &\triangleq b - c, \end{aligned}$$

which combined with $a = 0$ gives $b = c$. Observing $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_2 \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{M}_2$ since $\boldsymbol{M}_2$ is an Mb of $\boldsymbol{T}_2$, we obtain $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{Q} \,|\, (\boldsymbol{V} \setminus \boldsymbol{M}_2 \setminus \boldsymbol{T}_2 \setminus \boldsymbol{Q}) \cup \boldsymbol{M}_2$ by using the weak union property, or equivalently, $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_1$, so $\mathbb{I}(\boldsymbol{T}_2; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_1) = 0$. This means

$$\begin{aligned} 0 \leqslant c &= \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) \\ &= \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \cup \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) - \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q}) \\ &= \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_1) - \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q}) \\ &= \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) - \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q}). \qquad (14) \end{aligned}$$

- (i) $\Leftarrow$ (ii): If $\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) \leqslant \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}')$ holds for any $\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$, then (14) indicates $c = 0$ since $0 \leqslant c = \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) - \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q}) \leqslant 0$. Therefore,

  $$\mathbb{I}(\boldsymbol{T}; \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \mathbb{I}(\boldsymbol{T}; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = b = 0,$$

  because of $b = c$. That is, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \,|\, \boldsymbol{N} \cup \boldsymbol{S}$, which means $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$, or equivalently, $\boldsymbol{S}$ is an MbS.
- (i) $\Rightarrow$ (ii): Observe that $\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{V} \setminus \boldsymbol{T}) = \min_{\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}} \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}')$ holds according to (14) holding for any $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ and $\boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q} = \boldsymbol{V} \setminus \boldsymbol{T}$. Then,

  $$\mathbb{I}(\boldsymbol{T}; \boldsymbol{Q} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \mathbb{I}(\boldsymbol{T}; \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = 0$$

  follows immediately if $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$. By (13) and (14), we have

  $$\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{V} \setminus \boldsymbol{T}) = \min_{\boldsymbol{S}' \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}} \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}'), \qquad (15)$$

  noting again $\boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{Q} = \boldsymbol{V} \setminus \boldsymbol{T}$. This means $\boldsymbol{N} \cup \boldsymbol{S}$ is an MbS.

To prove the equivalence between (ii) and (iii), we need only to show

$$\mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) = \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{M}_1) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{M}_2) - \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2) - \mathbb{I}(\boldsymbol{T}; \boldsymbol{N} \cup \boldsymbol{S}). \qquad (16)$$

In fact, using $\mathbb{I}(\boldsymbol{T}; \boldsymbol{N} \cup \boldsymbol{S}) = \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{N} \cup \boldsymbol{S} \,|\, \boldsymbol{T}_1)$, we have

$$\begin{aligned} d &\triangleq \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2) + \mathbb{I}(\boldsymbol{T}; \boldsymbol{N} \cup \boldsymbol{S}) \\ &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{T}_1) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{N} \cup \boldsymbol{S} \,|\, \boldsymbol{T}_1) \\ &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{T}_2 \cup \boldsymbol{N} \cup \boldsymbol{S}) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_1) \\ &= \mathbb{I}(\boldsymbol{T}_1; \boldsymbol{M}_1) + \mathbb{I}(\boldsymbol{T}_2; \boldsymbol{M}_2), \end{aligned}$$

which is equivalent to (16). This means (ii) $\Leftrightarrow$ (iii).

Now, we show that (i) is equivalent to (iv):

- (i) $\Rightarrow$ (iv): This implication holds clearly due to the decomposition property.
- (i) $\Leftarrow$ (iv): Assume $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_1$ in $\boldsymbol{V} \setminus \boldsymbol{T}_2$, that is, $\boldsymbol{T}_1 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{T}_2) \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S}$, or equivalently, $\boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{T}_1 \,|\, \boldsymbol{N} \cup \boldsymbol{S}$. On the other hand, $\boldsymbol{M}_2$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V}$, meaning $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M}_2 \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{M}_2$, which combined with the weak union property gives $\boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \perp\!\!\!\perp \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S} \cup \boldsymbol{T}_1$. By the contraction property, $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T} \,|\, \boldsymbol{N} \cup \boldsymbol{S}$. This means $\boldsymbol{S}$ is an MbS. Similarly, if $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$, we can show $\boldsymbol{S}$ is an MbS.

Finally, we prove that an MbS, $\boldsymbol{S}$, is an MBS if and only if $\boldsymbol{T}_2 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ holds for any $Y \in \boldsymbol{S}$. We first prove the necessity by reductio ad absurdum. Suppose there is some variable $Y \in \boldsymbol{S}$ such that $\boldsymbol{T}_2 \perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup \boldsymbol{R}$, in which $\boldsymbol{R} \triangleq \boldsymbol{S} \setminus \{Y\}$. Recall that $\boldsymbol{S}$ is an MbS, we get

$$\boldsymbol{T}_2 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{T}_1) \setminus (\boldsymbol{N} \cup \boldsymbol{S}) \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{S}.$$

Equivalently, $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{R}) \setminus \boldsymbol{T} \setminus \{Y\} \,|\, (\boldsymbol{N} \cup \boldsymbol{R}) \cup \{Y\}$, which combined with $\boldsymbol{T}_2 \perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup \boldsymbol{R}$ gives $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{R}) \setminus \boldsymbol{T} \,|\, \boldsymbol{N} \cup \boldsymbol{R}$, in view of the contraction property. That is, $\boldsymbol{T}_2 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{T}_1) \setminus (\boldsymbol{N} \cup \boldsymbol{R}) \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{R}$. Therefore, $\boldsymbol{N} \cup \boldsymbol{R}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$, and thus an MbS of $\boldsymbol{T}$ to $\boldsymbol{N}$. This contradicts the condition that $\boldsymbol{S}$ is an MBS of $\boldsymbol{T}$ to $\boldsymbol{N}$, and thus $\boldsymbol{T}_2 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ holds for any $Y \in \boldsymbol{S}$.

To prove the sufficiency, we suppose $\boldsymbol{S}$ is not a MBS of $\boldsymbol{T}$ to $\boldsymbol{N}$, that is, there is some $\boldsymbol{R} \subsetneq \boldsymbol{S}$ such that $\boldsymbol{R}$ is an MbS of $\boldsymbol{T}$ to $\boldsymbol{N}$. Take any given variable, Y, in $\boldsymbol{S} \setminus \boldsymbol{R}$. Then, $\boldsymbol{N} \cup \boldsymbol{R}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$. That is, $\boldsymbol{T}_2 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{T}_1) \setminus (\boldsymbol{N} \cup \boldsymbol{R}) \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup \boldsymbol{R}$. By the weak union property, we have

$$\boldsymbol{T}_2 \perp\!\!\!\perp (\boldsymbol{V} \setminus \boldsymbol{T}_1) \setminus [\boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})] \setminus \boldsymbol{T}_2 \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\}).$$

This combined with the decomposition property means $\boldsymbol{T}_2 \perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$, since

$$Y \in (\boldsymbol{V} \setminus \boldsymbol{T}_1) \setminus [\boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})] \setminus \boldsymbol{T}_2,$$

and thus leads to a contradiction to the condition that $\boldsymbol{T}_2 \not\perp\!\!\!\perp Y \,|\, \boldsymbol{N} \cup (\boldsymbol{S} \setminus \{Y\})$ holds for any $Y \in \boldsymbol{S}$.

The proof of Theorem 4 is completed. ■

Example 1 *Consider the BN $(\mathbb{G}, \mathbb{P})$ over $\boldsymbol{V} = \{A, B, C, D\}$ presented in Figure 8, in which A, B, and C take $\{1, 2, 3\}$ while D takes $\{1, 2\}$. Put $\boldsymbol{T} = \{T_1, T_2\}$, $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T} = \emptyset$, and $\boldsymbol{S} = \{C\}$, $\boldsymbol{S}_0 = \{C, D\}$ with $T_1 = A$, $T_2 = B$, $\boldsymbol{M}_1 = \{B\}$, $\boldsymbol{M}_2 = \{A\}$. By Figure 3, we can easily conclude that A and B are information equivalent with respect to C. This means*

$$\mathbb{I}(C; A) > 0, \;\; \mathbb{I}(C; B \,|\, A) = 0; \;\; \textit{and} \;\; \mathbb{I}(C; B) > 0, \;\; \mathbb{I}(C; A \,|\, B) = 0. \qquad (17)$$

It follows from the chain rule for CMI (Cover and Thomas, 2006) that

(i) *$\boldsymbol{M}_1$ is an MB of T_1 in $\boldsymbol{V}$: By (17), we have*

- *$\mathbb{I}(A; C, D \,|\, B) = \mathbb{I}(A; C \,|\, B) + \mathbb{I}(A; D \,|\, B, C) = 0$, since $\{B, C\}$ d-separates $\{A\}$ and $\{D\}$;*
- *$\mathbb{I}(A; C, D) \geqslant \mathbb{I}(A; C) > 0$.*

(ii) $\boldsymbol{M}_2$ *is an MB of* T_2 *in* $\boldsymbol{V}$*: By* (17)*, we have*

- $\mathbb{I}(B;C,D\,|\,A)=\mathbb{I}(B;C\,|\,A)+\mathbb{I}(B;D\,|\,A,C)=0$*, since* $\{A,C\}$ *d-separates* $\{B\}$ *and* $\{D\}$*;*
- $\mathbb{I}(B;C,D)\geqslant\mathbb{I}(B;C)>0$*.*

(iii) $\boldsymbol{N}\cup\boldsymbol{S}$ *is an Mb of* $\boldsymbol{T}$ *in* $\boldsymbol{V}$*, so* $\boldsymbol{S}$ *is an MbS: By* (17)*,*

$$\mathbb{I}(A,B;D\,|\,C)=\mathbb{I}(A;D\,|\,C)+\mathbb{I}(B;D\,|\,A,C)=0,$$

because $\{C\}$ *d-separates* $\{A\}$ *and* $\{D\}$*, while* $\{A,C\}$ *d-separates* $\{B\}$ *and* $\{D\}$*.*

(iv) $\mathbb{I}(\boldsymbol{T}_1;\boldsymbol{T}_2\,|\,\boldsymbol{N}\cup\boldsymbol{S})=\min_{\boldsymbol{S}'\subseteq\boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}}\mathbb{I}(\boldsymbol{T}_1;\boldsymbol{T}_2\,|\,\boldsymbol{N}\cup\boldsymbol{S}')$*: it suffices to show the following inequalities:*

- $\mathbb{I}(A;B\,|\,C)=\mathbb{I}(A;B\,|\,C,D)$*. In fact,*

$$\begin{aligned}\mathbb{I}(A;B\,|\,C,D)&=\mathbb{I}(A;B,D\,|\,C)-\mathbb{I}(A;D\,|\,C)\\&=\mathbb{I}(A;B\,|\,C)+\mathbb{I}(A;D\,|\,B,C)-\mathbb{I}(A;D\,|\,C)=\mathbb{I}(A;B\,|\,C),\end{aligned}$$

  since both $\{B,C\}$ *and* $\{C\}$ *d-separate* $\{A\}$ *and* $\{D\}$*;*

- $\mathbb{I}(A;B\,|\,C)\leqslant\mathbb{I}(A;B\,|\,D)$*; In fact,*

$$\begin{aligned}\mathbb{I}(A;B\,|\,C)&=\mathbb{I}(A;B\,|\,C,D)=\mathbb{I}(A;B,C\,|\,D)-\mathbb{I}(A;C\,|\,D)\\&=\mathbb{I}(A;B\,|\,D)+\mathbb{I}(A;C\,|\,B,D)-\mathbb{I}(A;C\,|\,D)\\&=\mathbb{I}(A;B\,|\,D)-\mathbb{I}(A;C\,|\,D)\leqslant\mathbb{I}(A;B\,|\,D),\end{aligned}$$

  due to $\mathbb{I}(A;C\,|\,B,D)=0$*, because of*

$$\begin{aligned}0&\leqslant\mathbb{I}(A;C\,|\,B,D)=\mathbb{I}(A;C,D\,|\,B)-\mathbb{I}(A;D\,|\,B)\\&=\mathbb{I}(A;C\,|\,B)+\mathbb{I}(A;D\,|\,B,C)-\mathbb{I}(A;D\,|\,B)\;=\;-\mathbb{I}(A;D\,|\,B)\leqslant 0,\end{aligned}$$

  since $\mathbb{I}(A;C\,|\,B)=0$ *(see Equation 17) and* $\{B,C\}$ *d-separates* $\{A\}$ *and* $\{D\}$*;*

- $\mathbb{I}(A;B\,|\,C)\leqslant\mathbb{I}(A;B)$*. In fact, by* (17)*, we have* $\mathbb{I}(A;C\,|\,B)=0$*. Thus,*

$$\begin{aligned}\mathbb{I}(A;B)&=\mathbb{I}(A;B,C)-\mathbb{I}(A;C\,|\,B)=\mathbb{I}(A;C)+\mathbb{I}(A;B\,|\,C)-\mathbb{I}(A;C\,|\,B)\\&=\mathbb{I}(A;C)+\mathbb{I}(A;B\,|\,C)\geqslant\mathbb{I}(A;B\,|\,C).\end{aligned}$$

(v) $\mathbb{I}(\boldsymbol{T};\boldsymbol{S}\,|\,\boldsymbol{N})=\max_{\boldsymbol{S}'\subseteq\boldsymbol{V}\setminus\boldsymbol{N}\setminus\boldsymbol{T}}\mathbb{I}(\boldsymbol{T};\boldsymbol{S}'\,|\,\boldsymbol{N})$*: the proof is omitted.*

(vi) $\boldsymbol{N}\cup\boldsymbol{S}$ *is an Mb of* T_1 *in* $\boldsymbol{V}\setminus\{T_2\}$*: By* (17)*, we have*

- $\mathbb{I}(A;C,D)\geqslant\mathbb{I}(A;C)>0$*;*
- $\mathbb{I}(A;D\,|\,C)=0$*, since* $\{C\}$ *d-separates* $\{A\}$ *and* $\{D\}$*.*

(vii) $\boldsymbol{N}\cup\boldsymbol{S}$ *is an Mb of* T_2 *in* $\boldsymbol{V}\setminus\{T_1\}$*: By* (17)*, we have*

- $\mathbb{I}(B;C,D)\geqslant\mathbb{I}(B;C)>0$*;*
- $\mathbb{I}(B;D\,|\,C)=0$*, since* $\{C\}$ *d-separates* $\{B\}$ *and* $\{D\}$*.*

(viii) $\boldsymbol{S}$ *is an MBS;* $\boldsymbol{S}_0$ *is an MbS (but not an MBS):* $\mathbb{I}(A;B\,|\,C,D)=\mathbb{I}(A;B\,|\,C)$*. In fact,*

$$\mathbb{I}(A;B\,|\,C,D)=\mathbb{I}(A;B,D\,|\,C)-\mathbb{I}(A;D\,|\,C)=\mathbb{I}(A;B\,|\,C)+\mathbb{I}(A;D\,|\,B,C)-0=\mathbb{I}(A;B\,|\,C),$$

since both $\{B,C\}$ *and* $\{C\}$ *d-separate* $\{A\}$ *and* $\{D\}$*.* ■

Theorem 5 *Let* $\boldsymbol{M}_j$ *be an MB of* $T_j^{(d)}$ *in* $\boldsymbol{V}\setminus\boldsymbol{T}$ *for* $j=1,\cdots,t$*. Then,* $\boldsymbol{M}\triangleq\cup_{j=1}^{k}\boldsymbol{M}_j$ *is an Mb of* $\boldsymbol{T}$*. Further,* $\boldsymbol{M}$ *is an MB of* $\boldsymbol{T}$ *iff for any* $X\in\boldsymbol{M}$ *there is some* j *such that* $T_j^{(d)}\not\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$*.*

Proof Recall that T is the merged version of $\boldsymbol{T}$, while $\boldsymbol{T}^{(d)}$ is the dummy version of T; all of them have the same MBs.

First, we have $T_j^{(d)}\perp\!\!\!\perp(\boldsymbol{V}\setminus\boldsymbol{T})\setminus\boldsymbol{M}_j\setminus\{T_j^{(d)}\}\,|\,\boldsymbol{M}_j$ for $j=1,\cdots,t$. Considering $T_j^{(d)}\notin\boldsymbol{V}\setminus\boldsymbol{T}$, it follows that $T_j^{(d)}\perp\!\!\!\perp\boldsymbol{V}\setminus\boldsymbol{T}\setminus\boldsymbol{M}_j\,|\,\boldsymbol{M}_j$, which combined with the weak union property gives

$$T_j^{(d)}\perp\!\!\!\perp\boldsymbol{V}\setminus\boldsymbol{T}\setminus\boldsymbol{M}\,|\,\boldsymbol{M},\quad j=1,\cdots,t,\tag{18}$$

since $\boldsymbol{M}_j\subseteq\boldsymbol{M}$. Putting $\boldsymbol{U}\triangleq\boldsymbol{V}\setminus\boldsymbol{T}\setminus\boldsymbol{M}$, the above independence statements imply

$$\mathbb{P}\big(T_j^{(d)}=1,\,\boldsymbol{U}=\boldsymbol{u}\,\big|\,\boldsymbol{M}\big)=\mathbb{P}\big(T_j^{(d)}=1\,\big|\,\boldsymbol{M}\big)\mathbb{P}\big(\boldsymbol{U}=\boldsymbol{u}\,\big|\,\boldsymbol{M}\big),\quad j=1,\cdots,t,$$

or equivalently, $\mathbb{P}(T=j,\,\boldsymbol{U}=\boldsymbol{u}\,|\,\boldsymbol{M})=\mathbb{P}(T=j\,|\,\boldsymbol{M})\mathbb{P}(\boldsymbol{U}=\boldsymbol{u}\,|\,\boldsymbol{M})$, meaning $T\perp\!\!\!\perp\boldsymbol{V}\setminus\boldsymbol{T}\setminus\boldsymbol{M}\,|\,\boldsymbol{M}$, and thus $\boldsymbol{T}\perp\!\!\!\perp\boldsymbol{V}\setminus\boldsymbol{T}\setminus\boldsymbol{M}\,|\,\boldsymbol{M}$. This shows $\boldsymbol{M}$ is an Mb of $\boldsymbol{T}$.

In what follows, we prove $\boldsymbol{M}$ is an MB of $\boldsymbol{T}$ if and only if, for any $X\in\boldsymbol{M}$, $T_j^{(d)}\not\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$ holds for some j:

"⇒" Assume $\boldsymbol{M}$ is an MB of $\boldsymbol{T}$. Suppose there is some variable X such that $T_j^{(d)}\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$ holds for any j. Then, by (18) and the contraction property, we get

$$T_j^{(d)}\perp\!\!\!\perp(\boldsymbol{V}\setminus\boldsymbol{T})\setminus(\boldsymbol{M}\setminus\{X\})\,|\,\boldsymbol{M}\setminus\{X\},\quad j=1,\cdots,t.\tag{19}$$

Similar to the proof of the first conclusion, it can be readily proven that (19) implies

$$\boldsymbol{T}\perp\!\!\!\perp(\boldsymbol{V}\setminus\boldsymbol{T})\setminus(\boldsymbol{M}\setminus\{X\})\,|\,\boldsymbol{M}\setminus\{X\},$$

meaning that $\boldsymbol{M}$ has a proper subset, $\boldsymbol{M}\setminus\{X\}$, which is an Mb of $\boldsymbol{T}$. This contradicts the minimality of $\boldsymbol{M}$, and thus proves the necessity.

"⇐" Suppose $\boldsymbol{M}$ is not an MB of $\boldsymbol{T}$ (or T). Then, there is some $X\in\boldsymbol{M}$ such that $T\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$. It follows that $\mathbb{P}(T=j,X=x\,|\,\boldsymbol{M}\setminus\{X\})=\mathbb{P}(T=j\,|\,\boldsymbol{M}\setminus\{X\})\mathbb{P}(X=x\,|\,\boldsymbol{M}\setminus\{X\})$ holds for any $j=1,\cdots,t$. Or equivalently, we have

$$\mathbb{P}\big(T_j^{(d)}=1,X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big)=\mathbb{P}\big(T_j^{(d)}=1\,\big|\,\boldsymbol{M}\setminus\{X\}\big)\mathbb{P}\big(X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big).\tag{20}$$

Futher, (20) indicates

$$\begin{aligned}\mathbb{P}\big(T_j^{(d)}=0,X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big)&=\mathbb{P}\big(X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big)-\mathbb{P}\big(T_j^{(d)}=1,X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big)\\&=\big[1-\mathbb{P}\big(T_j^{(d)}=1\,\big|\,\boldsymbol{M}\setminus\{X\}\big)\big]\mathbb{P}\big(X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big)\\&=\mathbb{P}\big(T_j^{(d)}=0\,\big|\,\boldsymbol{M}\setminus\{X\}\big)\mathbb{P}\big(X=x\,\big|\,\boldsymbol{M}\setminus\{X\}\big).\end{aligned}\tag{21}$$

By (20) and (21), we get $T_j^{(d)}\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$, which contradicts $T_j^{(d)}\not\perp\!\!\!\perp X\,|\,\boldsymbol{M}\setminus\{X\}$. This proves the sufficiency.

The proof is completed. ■

Theorem 6 (Correctness of IAMBS and KIAMBS) *Assume that $\boldsymbol{T}_2$ satisfies the local composition property, and that all CI tests are correct. Then* (i) *IAMBS outputs an MB of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$;* (ii) *KIAMBS outputs an MB of $\boldsymbol{T}_1 \cup \boldsymbol{T}_2$ for any $K \in [0, 1)$.*

Proof Clearly, $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}_2$ in $\boldsymbol{V} \setminus \boldsymbol{T}_1$ at the end of the growing phase of either IAMBS or KIAMBS under the local composition assumption, as in IAMB and KIAMB. Therefore, $\boldsymbol{S}$ is an MbS at the end of this stage. According to the last conclusion of Theorem 4, $\boldsymbol{S}$ is an MBS after it is refined. Finally, as a direct consequence of Lemma 4 (shown below), $\boldsymbol{N} \cup \boldsymbol{S}$ is an MB at the end of the algorithm, considering the process of refining $\boldsymbol{N}$ is similar to that of refining $\boldsymbol{S}$. ■

Remark 2 *The following two statements hold:* (a) *violating local intersection implies violating adjacency faithfulness;* (b) *under the orientation faithfulness condition, violating local composition at the end of the first phase of IAMB or KIAMB or IAMBS or KIAMBS means violating adjacency faithfulness.*

Proof By Lemma 1, the violation of the local intersection property means information equivalence occurs; further, Lemeire et al. (2012) showed that information equivalence is one of the cases of violating adjacency faithfulness. Hence, the violation of local intersection is one of the violations of adjacency faithfulness.

Now, we show that the violation of local composition, which is present at the end of the first phase of IAMB or KIAMB, is also one of the violations of adjacency faithfulness under the *orientation faithfulness condition*.

In fact, let $\boldsymbol{M}$ be the output of the first phase of IAMB or KIAMB, but not an Mb of $\boldsymbol{T}$. Without loss of generality, we assume $|\boldsymbol{T}| = 1$ and $\boldsymbol{T} = \{T\}$. Then, $T \perp\!\!\!\perp X \,|\, \boldsymbol{M}$ holds for any $X \in \boldsymbol{V} \setminus \boldsymbol{M} \setminus \{T\}$ but $T \not\perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M} \setminus \{T\} \,|\, \boldsymbol{M}$. Considering that the set $\boldsymbol{M}_T$ composed of the parents, children, and spouses of T is an Mb of T, we have $\boldsymbol{M} \not\supseteq \boldsymbol{M}_T$. Thus, there is some $X \in \boldsymbol{M}_T$ such that $X \notin \boldsymbol{M}$. If X is a spouse of T, then all the children of T and X are not in $\boldsymbol{M}$ (if not so, $T \not\perp\!\!\!\perp X \,|\, \boldsymbol{M}$ holds immediately following from the orientation faithfulness condition, and thus contradicts $T \perp\!\!\!\perp X \,|\, \boldsymbol{M}$ since $X \notin \boldsymbol{M}$). In this sense, we conclude that there is some node X adjacent to T such that $T \perp\!\!\!\perp X \,|\, \boldsymbol{M}$. This means the adjacency faithfulness condition is violated.

In short words, both the violation of the local composition property (present at the end of the first phase of IAMB or KIAMB) and the violation of the local intersection property are the violations of adjacency faithfulness, under the orientation faithfulness condition. ■

Lemma 3 (a) *If there is $\boldsymbol{P} \subseteq \boldsymbol{M}_1 \setminus \boldsymbol{T}_2$ such that $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{P} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$, then $(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$ is an Mb of $\boldsymbol{T}_1$;* (b) *If there is $\boldsymbol{Q} \subseteq \boldsymbol{N} \setminus \boldsymbol{P}$ such that $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{Q} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$ and $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{Q} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$, then $(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$ is an Mb of $\boldsymbol{T}_1$, and $(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$ is an Mb of $\boldsymbol{T}_2$.*

Proof Considering that $\boldsymbol{N} \cup \boldsymbol{T}_2$ is an Mb of $\boldsymbol{T}_1$, we have $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N} \cup \boldsymbol{T}_2) \setminus \boldsymbol{T}_1 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2 \cup \boldsymbol{P}$, which combined with $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{P} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$ implies $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2] \setminus \boldsymbol{T}_1 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$, in view of the contraction property. The first conclusion is proved.

For convenience, we denote now $\boldsymbol{N}_1 \triangleq \boldsymbol{M}_1 \setminus \boldsymbol{T}_2$. To show the second conclusion, we note that $\boldsymbol{P} \subseteq \boldsymbol{N}_1$, so $(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_1$ is an Mb of $\boldsymbol{T}_2$. It follows that: (i) $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2] \setminus \boldsymbol{T}_1 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_2$, which combined with $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{Q} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$ gives $\boldsymbol{T}_1 \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2] \setminus \boldsymbol{T}_1 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_2$; and (ii) $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_1] \setminus \boldsymbol{T}_2 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P}) \cup \boldsymbol{T}_1$, which combined with $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{Q} \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$ yields $\boldsymbol{T}_2 \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1] \setminus \boldsymbol{T}_2 \,|\, (\boldsymbol{N} \setminus \boldsymbol{P} \setminus \boldsymbol{Q}) \cup \boldsymbol{T}_1$. The second conclusion is also proved. ■

Lemma 4 *Let $\boldsymbol{T}_i$ be a subset of $\boldsymbol{V}$ with an Mb $\boldsymbol{M}_i$ for $i = 1, 2$, and $\boldsymbol{S} \subseteq \boldsymbol{V} \setminus \boldsymbol{N} \setminus \boldsymbol{T}$ be an MBS of $\boldsymbol{T}$ to $\boldsymbol{N}$, with $\boldsymbol{T} = \boldsymbol{T}_1 \cup \boldsymbol{T}_2$ and $\boldsymbol{N} = (\boldsymbol{M}_1 \cup \boldsymbol{M}_2) \setminus \boldsymbol{T}$. Assume $\boldsymbol{N}_0$ be a subset of $\boldsymbol{N}$ such that $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$. Then $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is an MB of $\boldsymbol{T}$ if and only if $\boldsymbol{T} \not\perp\!\!\!\perp Y \,|\, (\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}$ holds for any $Y \in \boldsymbol{N}_0$.*

Proof By the definition of MBS, $\boldsymbol{N} \cup \boldsymbol{R}$ and thus $\boldsymbol{N}_0 \cup \boldsymbol{R}$ will never be an Mb of $\boldsymbol{T}$ for any $\boldsymbol{N}_0 \subseteq \boldsymbol{N}$ and $\boldsymbol{R} \subsetneq \boldsymbol{S}$, in view of the weak union property.

- Necessity: Suppose there is some $Y \in \boldsymbol{N}_0$ such that $\boldsymbol{T} \perp\!\!\!\perp Y \,|\, (\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}$. By the precondition that $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$, we have $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus (\boldsymbol{N}_0 \cup \boldsymbol{S}) \setminus \boldsymbol{T} \,|\, [(\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}] \cup \{Y\}$. These two relationships combined with the contraction property imply
$$\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus [(\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}] \setminus \boldsymbol{T} \,|\, (\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S},$$
or equivalently, $(\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$. This contradicts that $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is an MB of $\boldsymbol{T}$.
- Sufficiency: Suppose $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is not an MB of $\boldsymbol{T}$, that is, there is some $\boldsymbol{N}_0' \subsetneq \boldsymbol{N}_0$ such that $\boldsymbol{N}_0' \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$. Take any given variable, Y, in $\boldsymbol{N}_0 \setminus \boldsymbol{N}_0'$. It can be shown that $\boldsymbol{T} \perp\!\!\!\perp Y \,|\, (\boldsymbol{N}_0 \setminus \{Y\}) \cup \boldsymbol{S}$, which leads to a contradiction. Hence, $\boldsymbol{N}_0 \cup \boldsymbol{S}$ is an MB of $\boldsymbol{T}$.

The proof is completed. ■

Appendix C. Improving the Log-Likelihood Ratio Test

In Subsection 4.3, we mentioned that the X^2 or G^2 test is suitable only for cases of small $|\boldsymbol{T}|$, and then summarized some improving methods proposed in the literature (Lawley, 1956; Hosmane, 1986, 1987, 1990; Brin et al., 1997; Silverstein et al., 1998; Aliferis et al., 2010b). However, we need more suitable CI testing methods when working on the MB discovery problem for multiple targets. In this appendix, we discuss a practical way of improving the G^2 test by damping the number of degrees of freedom for the G^2 statistic.

Consider the G^2 statistic, $G^2(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \triangleq 2n \cdot \mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \mid \boldsymbol{Z})$, which approximates to the chi-square variate with $r \triangleq (r_{\boldsymbol{X}} - 1)(r_{\boldsymbol{Y}} - 1) r_{\boldsymbol{Z}}$ degrees of freedom, namely $\chi^2(r)$, where $r_{\boldsymbol{\xi}}$ represents the number of configurations for $\boldsymbol{\xi}$ (de Campos, 2006, p. 2158).

Theoretically, $G^2(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})$ is a reasonable statistic for testing the hypothesis "$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$" *when n is large enough.* Unfortunately, this precondition is practically hard to be valid in many situations (Cochran, 1954; Yaramakala, 2004; Bromberg and Margaritis, 2009) due to the following reason: Let $\boldsymbol{X} = \{X_{i_1}, \cdots, X_{i_x}\}$, $\boldsymbol{Y} = \{X_{j_1}, \cdots, X_{j_y}\}$, and $\boldsymbol{Z} = \{X_{k_1}, \cdots, X_{k_z}\}$, in which each variable X_ℓ takes r_ℓ values. Then $r = (\prod_{\ell=1}^{x} r_{i_\ell} - 1)(\prod_{\ell=1}^{y} r_{j_\ell} - 1)(\prod_{\ell=1}^{z} r_{k_\ell})$, which is exponential with respect to x, y, and z. On the one hand, by the Wilson-Hilferty approximation for $\chi^2_\alpha(r)$ (de Campos, 2006; Gao, 2005), we obtain $\chi^2_\alpha(r) \approx c_{\alpha,r}\, r$, in which $c_{\alpha,r} \triangleq \left(1 - 2/(9r) + \sqrt{2/(9r)}\, z_\alpha\right)^3$ is a bit larger than 1, with z_α being the upper α-quantile of the standard normal distribution; on the other hand, we can show $\mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \leqslant \log_2 r_{\boldsymbol{X},\boldsymbol{Y}}$ with $r_{\boldsymbol{X},\boldsymbol{Y}} \triangleq \min\{\prod_{\ell=1}^{x} r_{i_\ell}, \prod_{\ell=1}^{y} r_{j_\ell}\}$. It follows that

$$p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) = \mathbb{P}\left\{\chi^2(r) \geqslant 2n \cdot \mathbb{I}_{\boldsymbol{D}}(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})\right\} \geqslant \mathbb{P}\left\{\chi^2(r) \geqslant 2n \cdot \log_2 r_{\boldsymbol{X},\boldsymbol{Y}}\right\}.$$

Suppose we are doing a G^2 test for the *false hypothesis* "$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$" (i.e., the truth is $\boldsymbol{X} \not\perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$). Then, at least $\frac{\chi^2_\alpha(r)}{2\log_2 r_{\boldsymbol{X},\boldsymbol{Y}}} \approx \frac{c_{\alpha,r}\, r}{2\log_2 r_{\boldsymbol{X},\boldsymbol{Y}}} = O\left(\frac{r}{\log_2 r_{\boldsymbol{X},\boldsymbol{Y}}}\right)$ instances are required if we expect the statistical

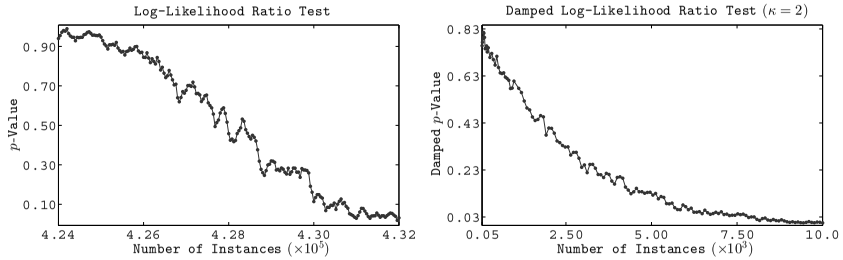


Figure 20: p-value/damped p-value versus the number of instances, n: the left subfigure illustrates why a very large n may still not be "large enough" for making a correct decision about the *false hypothesis* "$\boldsymbol{X}_1 \perp\!\!\!\perp \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}_1$" based on the G^2 test: at least 4.31×10^5 instances are required; while the right illustrates why the damped G^2 test is suitable for testing the same *false hypothesis*: about 8000 instances are sufficient.

decision can be correctly made with the significance level α (or equivalently, $p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z}) \leqslant \alpha$). In many practical situations, however, n may be far smaller than the required number of sample instances with the magnitude of at least $O(r/\log_2 r_{\boldsymbol{X},\boldsymbol{Y}})$, recalling that r is exponential with respect to x, y, and z, especially when too many variables are involved. In this case, the statistical decision made for the hypothesis will be wrong.

Taking the ALARM network presented in Figure 4 for example, we put

$$\boldsymbol{X}_1 = \{X_{36}\},\ \boldsymbol{Y}_1 = \{X_{11}, X_{34}, X_{35}, X_{37}\},\ \text{and}\ \ \boldsymbol{Z}_1 = \{X_4, X_{14}, X_{15}, X_{16}, X_{18}, X_{21}, X_{22}, X_{31}\};$$

then we compute the p-value versus the number of instances from 1000 to 1,000,000. The results are drawn in Figure 20 (averaged over 10 different samples with the same size). Note that the truth is $\boldsymbol{X}_1 \not\perp\!\!\!\perp \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}_1$ since $\boldsymbol{Y}_1$ is an Mb of $\boldsymbol{X}_1$. By the figure, the CI test for the *false hypothesis* "$\boldsymbol{X}_1 \perp\!\!\!\perp \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}_1$" is not correct unless at least $n_{\min} \approx 4.31 \times 10^5$ sample instances are available. It is mentioned that, in this example, $r/\log_2 r_{\boldsymbol{X}_1,\boldsymbol{Y}_1} \approx 2.42 \times 10^5$.

In short words, the precondition, "*when n is large enough*", for the theoretical assertion that "G^2 is a reasonable statistic for CI testing" may be hard to be guaranteed in practice because the above analysis and the numerical example indicate that a seemingly very large n may still not be "large enough". The problem is then how to improve on the G^2 test.

Observe that, for the G^2 test, the major reason for failing to make a correct statistical decision on CI testing is that the theoretical value of r is far larger than its data-driven value, due to the null cells frequently existing in the multi-contingency tables of $\boldsymbol{X}$ and $\boldsymbol{Y}$ given $\boldsymbol{Z}$ (e.g., Yaramakala, 2004, p. 34). In other words, the linear increase of n is hard to exponentially bring null cells into valid cells. Hence, a feasible way of improving the log-likelihood ratio G^2 test is to damp the increase of r such that the unmatched behaviours of n and r can get alleviated to a certain degree. Mathematically, we replace the theoretical value of r in $p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})$ with its a damped version, $g_{n,\kappa}(r)$, defined in (6), where $\kappa > 0$ is a constant, based on which $\frac{n}{\kappa}$ measures the amount of valid cells that n sample

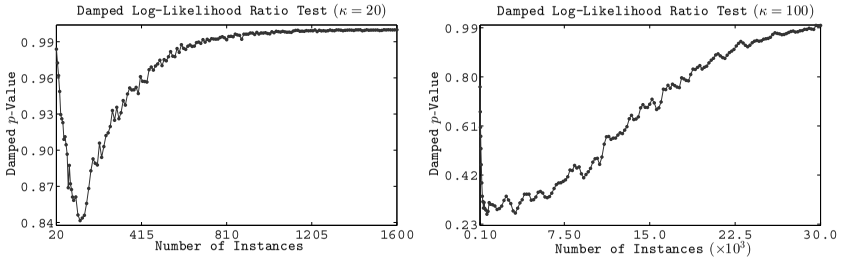


Figure 21: Damped p-value versus the number of instances: κ is taken as 20 and 100, respectively.

instances can support. It is easily seen that $g_{n,\kappa}(\cdot)$ possesses the following properties, which interpret the reasonability of employing such a damping procedure in the G^2 test:

- $g_{n,\kappa}(r)$ is monotonically increasing versus n for given r, and $\lim_{n\to+\infty} g_{n,\kappa}(r) = r$. This means more instances may generate more valid cells in the multi-contingency tables of $\boldsymbol{X}$ and $\boldsymbol{Y}$ given $\boldsymbol{Z}$, and all the theoretical degrees of freedom are valid when n is large enough.
- $g_{n,\kappa}(r)$ is monotonically increasing versus r for given n, and $\lim_{r\to+\infty} g_{n,\kappa}(r) = \frac{n}{\kappa}$. This means a larger r should correspond to a larger $g_{n,\kappa}(r)$, but not exceeding the supporting capacity of the data.
- For sufficient data, the damping function $g_{n,\kappa}(\cdot)$ only plays a little role; while for insufficient data, it trades off the theoretical r and the supporting capacity of the data.

For convenience, we call the resulted p-value, denoted by $p_g(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})$ instead of $p(\boldsymbol{X}; \boldsymbol{Y} \,|\, \boldsymbol{Z})$, and the resulted testing method to be the *damped p-value* and the *damped log-likelihood ratio test* (or damped G^2 test). Further, we use the negative damped p-value, $f_{\boldsymbol{D}}^{(3)}$ defined in (7), as the association function. It is mentioned here that the damped G^2 test approximately degenerates into the ordinary G^2 test when taking κ as a very small positive number.

The damped G^2 test may be more suitable than the ordinary G^2 test when too many variables are involved in the conditional set. We implement this testing method (by taking κ as $2, 3, \cdots, 10$, respectively) on the *false hypothesis* "$\boldsymbol{X}_1 \perp\!\!\!\perp \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}_1$", and find that the correct decision "$\boldsymbol{X}_1 \not\perp\!\!\!\perp \boldsymbol{Y}_1 \,|\, \boldsymbol{Z}_1$" is always made for $\kappa \geqslant 3$, even when the number of instances is smaller than 1000. For the case of $\kappa = 2$, we present the results in the right subfigure of Figure 20, from which it is seen that, for the damped G^2 test, about 8000 sample instances are sufficient to make the correct decision. Note also that the ordinary G^2 test needs at least 4.31×10^5 instances.

However, the damped G^2 test may also face a potential danger: it may excessively damp the theoretical value of r if a too large κ is inappropriately used. Here, "excessively damping r" means that a too large value of κ will lead to a too small $g_{n,\kappa}(r)$ such that the damped G^2 test incorrectly reject a *true hypothesis* "$\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} \,|\, \boldsymbol{Z}$". To illustrate this explanation, we put

$$\boldsymbol{X}_2 = \{X_{21}\},\ \ \boldsymbol{Y}_2 = \{X_{11}, X_{34}, X_{35}, X_{36}, X_{37}\},\ \ \text{and}\ \ \boldsymbol{Z}_2 = \{X_{15}, X_{19}, X_{20}, X_{22}, X_{29}\}$$

from the ALARM network. The truth is $\boldsymbol{X}_2 \perp\!\!\!\perp \boldsymbol{Y}_2 \,|\, \boldsymbol{Z}_2$ since $\boldsymbol{Z}_2$ is an Mb of $\boldsymbol{X}_2$. Now, use damped G^2 to test the *true hypothesis* "$\boldsymbol{X}_2 \perp\!\!\!\perp \boldsymbol{Y}_2 \,|\, \boldsymbol{Z}_2$" by taking κ as $3, 4, \cdots, 10, 20, 100$, respectively. All

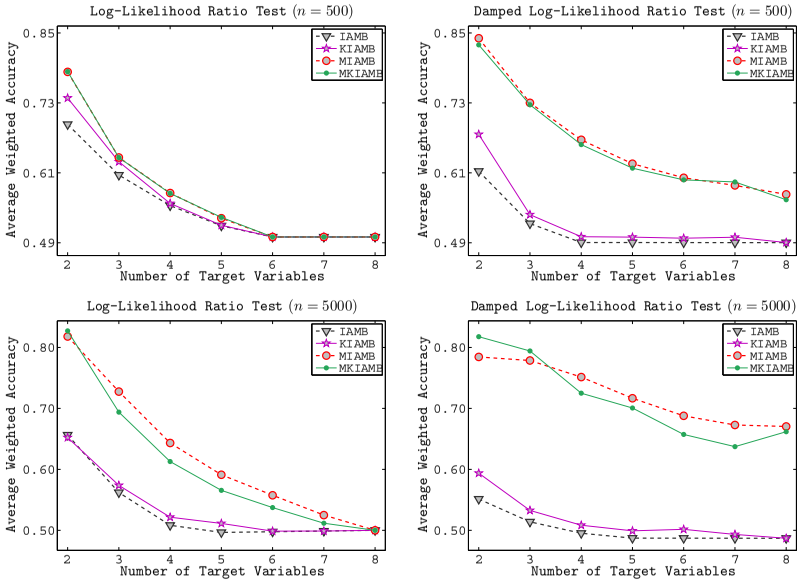


Figure 22: An illustration on why the damped G^2 test is more suitable than the ordinary G^2 test for the problem that involves too many variables, by virtue of the Pigs BN: the G^2 test no longer works when there are too many variables are involved, while the damped G^2 test remains valid in all considered cases.

the decisions are correctly made. However, a too large κ may be more apt to yield a relatively small damped p-value although it is still larger than α, as shown by Figure 21: 0.84 for the case of $\kappa = 20$ and only 0.27 for the case of $\kappa = 100$. Hence, to avoid the potential danger of excessively damping r, we conservatively recommend to take κ from the interval $[3, 10]$ in practice. In our benchmarking study, we employ $\kappa = 5$, which is large enough for testing *false hypotheses* and small enough for testing *true hypotheses*.

To further illustrate why the damped G^2 test is more suitable than the ordinary G^2 when working on a problem that involves too many variables, we make experiments on the six synthetical BNs based on `IAMB`, `KIAMB`, `MIAMB`, and `MKIAMB`.

For each algorithm, the G^2 test and the damped G^2 test are implemented for CI testing. Accordingly, the association functions, $f_{\boldsymbol{D}}^{(1)}$ and $f_{\boldsymbol{D}}^{(3)}$ defined by (5) and (7) are used. Figure 22 presents the results of the Pigs network, in which the left two are based on the ordinary G^2 test while the right two are based on the damped G^2 test. As seen, for the case of $n = 500$ the G^2 test becomes invalid when the target, $\boldsymbol{T}$, contains 6 or more variables, and for the case of $n = 5000$ this method no longer works when $|\boldsymbol{T}| = 8$. In comparison, the damped G^2 test is suitable for all cases. The results of the other five BNs show similar conclusions.

Appendix D. Used Acronyms

BFMB	breadth first search of Markov boundary algorithm (Fu and Desmarais, 2007).
BN	Bayesian network.
CI	conditional independence.
CMI	conditional mutual information.
CT-support	contingency table support: a set of items S has CT-support s at the t% level if at least t% of the cells in the contingency table for S have value s (Silverstein et al., 1998).
DAG	directed acyclic graph.
E-partition	equivalent partition (Lemeire, 2007): a relation $\Re \subset \boldsymbol{X} \otimes \boldsymbol{Y}$ defines an E-partition in $\boldsymbol{Y}_{\text{dom}}$ to a partition of $\boldsymbol{X}_{\text{dom}}$, if: (i) $\neg(\boldsymbol{x}_2 \Re \boldsymbol{y}_1)$ holds for any $\boldsymbol{x}_1, \boldsymbol{x}_2 \in \boldsymbol{X}_{\text{dom}}$ belonging to different partitions and for any $\boldsymbol{y}_1 \in \boldsymbol{Y}_{\text{dom}}$ with $\boldsymbol{x}_1 \Re \boldsymbol{y}_1$; and (ii) for every $\boldsymbol{X}_{\text{dom}}^{(k)}$, there exist $\boldsymbol{x}_1 \in \boldsymbol{X}_{\text{dom}}^{(k)}$ and $y_1 \in \boldsymbol{Y}_{\text{dom}}$ such that $\boldsymbol{x}_1 \Re \boldsymbol{y}_1$.
FS	feature selection.
GLL	generalized local learning: an algorithmic framework for local causal discovery and FS proposed by Aliferis et al. (2010a).
GS	grow-shrink algorithm (Margaritis and Thrun, 1999, 2000).
HITON	an MB discovery algorithm, pronounced hee-tón, from the Greek Χιτών, for "cover", "cloak", or "blanket" (Aliferis et al., 2003).
IAMB	incremental association Markov boundary algorithm (Tsamardinos et al., 2003); see Algorithm 3 for details.
IAMBS	an `IAMB`-based Markov boundary supplementary algorithm, outputting an MB for multiple targets (Algorithm 1).
KIAMB	a stochastic variant of `IAMB` (Peña et al., 2007); see Algorithm 3 for details.
KIAMBS	an `KIAMB`-based Markov boundary supplementary algorithm, outputting an MB for multiple targets (Algorithm 1).
KS	Koller-Sahami algorithm (Koller and Sahami, 1996).
LibSVM	a library for support vector machines contributed by Chang and Lin (2011).
Mb	Markov blanket: we call $\boldsymbol{M}$ an Mb of $\boldsymbol{T}$ if $\boldsymbol{T} \perp\!\!\!\perp \boldsymbol{V} \setminus \boldsymbol{M} \setminus \boldsymbol{T} \mid \boldsymbol{M}$ (Definition 1).
MB	Markov boundary: an MB of $\boldsymbol{T}$ is any Mb such that none of its proper subsets is an Mb of $\boldsymbol{T}$ (Definition 1).
MbS	Markov blanket supplementary: we call $\boldsymbol{S}$ an MbS of $\boldsymbol{T}$ to $\boldsymbol{N}$, if $\boldsymbol{N} \cup \boldsymbol{S}$ is an Mb of $\boldsymbol{T}$ (Definition 3).
MBS	Markov boundary supplementary: an MBS is any MbS such that none of its proper subsets is an MbS (Definition 3).
MIAMB	an `IAMB` and `IAMBS`-based algorithm, outputting an MB for multiple targets (see Algorithm 2 for details).
MKIAMB	an `KIAMB` and `KIAMBS`-based algorithm, outputting an MB for multiple targets (Algorithm 2).
MMMB	max-min Markov boundary algorithm (Tsamardinos et al., 2006).

PCMB	parents and children based Markov boundary algorithm (Peña et al., 2007).
RF	random forest algorithm.
RT	running time: the single CPU time implemented on an Intel i7-3612QM 2.1 GHz and Windows 7 with 64 bits.
SVM	support vector machine (in one-against-one approach).
T-partition	target partition (Lemeire, 2007): the domain, $\boldsymbol{X}_{\text{dom}}$, of $\boldsymbol{X}$ can be partitioned into some disjoint subsets $\boldsymbol{X}_{\text{dom}}^{(k)}$ for which $\mathbb{P}(\boldsymbol{T}\,|\,\boldsymbol{x})$ is the same for all $\boldsymbol{x} \in \boldsymbol{X}_{\text{dom}}^{(k)}$. This is called the T-partition of $\boldsymbol{X}_{\text{dom}}$ with respect to $\boldsymbol{T}$.
WA	weighted accuracy: WA is the average of the rate of true members and that of true nonmembers of an MB with respect to the truth.
WP	weighted precision: WP is the average of the rate of true members and that of true nonmembers of an MB with respect to the test.

References

Constantin F Aliferis, Ioannis Tsamardinos, and Alexander Statnikov. Hiton: a novel Markov blanket algorithm for optimal variable selection. In *AMIA 2003 Annual Symposium Proceedings*, pages 21–25. American Medical Informatics Association, 2003.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part I: Algorithms and empirical evaluation. *Journal of Machine Learning Research*, 11:171–234, 2010a.

Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and Markov blanket induction for causal discovery and feature selection for classification part II: Analysis and extensions. *Journal of Machine Learning Research*, 11:235–284, 2010b.

Ingo A Beinlich, H J Suermondt, R Martin Chavez, and Gregory F Cooper. The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks. In *Second European Conference on Artificial Intelligence in Medicine*, pages 247–256, London, 1989. Springer-Verlag.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324.

Sergey Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *SIGMOD'97 Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data*, volume 26, pages 265–276. ACM, 1997.

Facundo Bromberg and Dimitris Margaritis. Improving the reliability of causal discovery from small data sets using argumentation. *Journal of Machine Learning Research*, 10:301–340, 2009.

Gavin Brown, Adam Pocock, Ming-Jie Zhao, and Mikel Luján. Conditional likelihood maximisation: a unifying framework for information theoretic feature selection. *Journal of Machine Learning Research*, 13:27–66, 2012. (Version: MIToolbox-2.0).

Chih-Chung Chang and Chih-Jen Lin. LibSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Jie Cheng, Russell Greiner, Jonathan Kelly, David Bell, and Weiru Liu. Learning Bayesian networks from data: An information-theory based approach. *Artificial Intelligence*, 137(1–2):43–90, 2002.

William G Cochran. Some methods for strengthening the common χ^2 tests. *Biometrics*, 10(4): 417–451, 1954.

Thomas M Cover and Joy A Thomas. *Elements of Information Theory (Second Edition)*. John Wiley and Sons, 2006.

Noel Cressie and Timothy RC Read. Pearson's X^2 and the loglikelihood ratio statistic G^2: a comparative review. *International Statistical Review*, 57(1):19–43, 1989.

Rónán Daly, Qiang Shen, and Stuart Aitken. Learning Bayesian networks: Approaches and issues. *The Knowledge Engineering Review*, 26(2):99–157, 2011.

Luis M de Campos. A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7(Oct):2149–2187, 2006.

Shunkai Fu and Michel Desmarais. Local learning algorithm for Markov blanket discovery. In *AI 2007: Advances in Artificial Intelligence*, pages 68–79. Springer Berlin Heidelberg, 2007.

Shunkai Fu and Michel C Desmarais. Markov blanket based feature selection: a review of past decade. In *Proceedings of the World Congress on Engineering*, 2010.

Huixuan Gao. *Statistics Computation*. Peking University Press, Beijing, 2005.

Balakrishna Hosmane. Improved likelihood ratio test for multinomial goodness of fit. *Communications in Statistics-Theory and Methods*, 16(11):3185–3198, 1987.

Balakrishna S Hosmane. Smoothing of likelihood ratio statistic for equiprobable multinomial goodness-of-fit. *Annals of the Institute of Statistical Mathematics*, 42(1):133–147, 1990.

BS Hosmane. Improved likelihood ratio tests and Pearson chi-square tests for independence in two dimensional contingency tables. *Communications in Statistics-Theory and Methods*, 15(6): 1875–1888, 1986.

James Kennedy and Russell C Eberhart. Particle swarm optimization. In *Proceedings of 1995 IEEE International Conference on Neural Networks*, volume 4, pages 1942–1948, Perth, 1995.

James Kennedy and Russell C Eberhart. A discrete binary version of the particle swarm algorithm. In *IEEE International Conference on Systems, Man, and Cybernetics*, volume 5, pages 4104–4108, Orlando, 1997.

Daphne Koller and Mehran Sahami. Toward optimal feature selection. In *Thirteen International Conference in Machine Learning*. Stanford InfoLab, 1996.

D.N. Lawley. A general method for approximating to the distribution of likelihood ratio criteria. *Biometrika*, 43(3–4):295–303, 1956.

Jan Lemeire. *Learning Causal Models of Multivariate Systems and the Value of it for the Performance Modeling of Computer Programs*. ASP/VUBPRESS/UPA, PhD thesis, 2007.

Jan Lemeire, Stijn Meganck, Francesco Cartella, and Tingting Liu. Conservative independence-based causal structure learning in absence of adjacency faithfulness. *International Journal of Approximate Reasoning*, 53(9):1305–1325, 2012.

Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. Technical Report CMU-CS-99-134, 1999.

Dimitris Margaritis and Sebastian Thrun. Bayesian network induction via local neighborhoods. In *Advances in Neural Information Processing Systems*, volume 12, pages 505–511. Morgan Kaufmann, 2000.

Kevin P. Murphy. *Bayes Net Toolbox for Matlab*, 2007. (Version: FullBNT-1.0.7).

Richard E Neapolitan. *Learning Bayesian Networks*. Upper Saddle River: Prentice Hall, 2004.

Pekka Parviainen and Mikko Koivisto. Finding optimal Bayesian networks using precedence constraints. *Journal of Machine Learning Research*, 14(1):1387–1415, 2013.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. San Francisco: Morgan Kaufmann, 1988.

Jean-Philippe Pellet and André Elisseeff. Using Markov blankets for causal structure learning. *Journal of Machine Learning Research*, 9:1295–1342, 2008.

Jose M Peña, Roland Nilsson, Johan Björkegren, and Jesper Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45(2): 211–232, 2007.

Joseph Ramsey, Jiji Zhang, and Peter L Spirtes. Adjacency-faithfulness and conservative causal inference. In *Proceedings of the 22nd Annual Conference on Uncertainty in Artificial Intelligence (UAI-2006)*, pages 401–408, 2006.

Johannes Rauh, Nils Bertschinger, Eckehard Olbrich, and Jurgen Jost. Reconsidering unique information: Towards a multivariate information decomposition. In *2014 IEEE International Symposium on Information Theory (ISIT)*, pages 2232–2236. IEEE, 2014.

Federico Schlüter. A survey on independence-based Markov networks learning. *Artificial Intelligence Review*, 42:1069–1093, 2014.

Marco Scutari. Learning Bayesian networks with the bnlearn R package. *Journal of Statistical Software*, 35:1–22, 2010.

Craig Silverstein, Sergey Brin, and Rajeev Motwani. Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery*, 2(1):39–68, 1998.

Alexander Statnikov and Constantin F Aliferis. Analysis and computational dissection of molecular signature multiplicity. *PLoS Computational Biology*, 6(5):e1000790, 2010.

Alexander Statnikov, Nikita I. Lytkin, Jan Lemeire, and Constantin F Aliferis. Algorithms for discovery of multiple Markov boundaries. *Journal of Machine Learning Research*, 14(1):499–566, 2013.

Ioannis Tsamardinos and Constantin F Aliferis. Towards principled feature selection: Relevancy, filters and wrappers. In *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, 2003.

Ioannis Tsamardinos, Constantin F Aliferis, and Alexander Statnikov. Algorithms for large scale Markov blanket discovery. In *Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference (FLAIRS)*, pages 376–381, 2003.

Ioannis Tsamardinos, Laura E Brown, and Constantin F Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine learning*, 65(1):31–78, 2006.

Sandeep Yaramakala. *Fast Markov Blanket Discovery*. MS thesis, 2004.

Lianwen Zhang and Haipeng Guo. *Introduction to Bayesian Networks*. Science Press, Beijing, 2006.

Journal of Machine Learning Research 19 (2018) 1-29 Submitted 3/16; Revised 5/18; Published 9/18

Kernel Distribution Embeddings: Universal Kernels, Characteristic Kernels and Kernel Metrics on Distributions

Carl-Johann Simon-Gabriel cjsimon@tuebingen.mpg.de
Bernhard Schölkopf bs@tuebingen.mpg.de
MPI for Intelligent Systems
Spemannstrasse 41,
72076 Tübingen, Germany

Editor: Ingo Steinwart

Abstract

Kernel mean embeddings have become a popular tool in machine learning. They map probability measures to functions in a reproducing kernel Hilbert space. The distance between two mapped measures defines a semi-distance over the probability measures known as the maximum mean discrepancy (MMD). Its properties depend on the underlying kernel and have been linked to three fundamental concepts of the kernel literature: universal, characteristic and strictly positive definite kernels.

The contributions of this paper are three-fold. First, by slightly extending the usual definitions of universal, characteristic and strictly positive definite kernels, we show that these three concepts are essentially equivalent. Second, we give the first complete characterization of those kernels whose associated MMD-distance metrizes the weak convergence of probability measures. Third, we show that kernel mean embeddings can be extended from probability measures to generalized measures called Schwartz-distributions and analyze a few properties of these distribution embeddings.

Keywords: kernel mean embedding, universal kernel, characteristic kernel, Schwartz-distributions, kernel metrics on distributions, metrisation of the weak topology

1. Introduction

During the past decades, kernel methods have risen to a major tool across various areas of machine learning. They were originally introduced via the "kernel trick" to generalize linear regression and classification tasks by effectively transforming the optimization over a set of linear functions into an optimization over a so-called reproducing kernel Hilbert space (RKHS) $\mathcal{H}_k$, which is entirely defined by the kernel k. This lead to kernel (ridge) regression, kernel SVM and many other now standard algorithms. Besides these regression-type algorithms, another major family of kernel methods rely on kernel mean embeddings (KMEs). A KME is a mapping Φ_k that maps probability measures to functions in an RKHS via $\Phi_k : P \longmapsto \int_{\mathcal{X}} k(.,x)\,\mathrm{d}P(x)$. The RKHS-distance between two mapped measures defines a semi-distance over the set of probability measures, known as the Maximum Mean Discrepancy (MMD). It has numerous applications, ranging from homogeneity (Gretton et al., 2007), distribution comparison (Gretton et al., 2007, 2012) and (conditional) independence tests (Gretton et al., 2005, 2008; Fukumizu et al., 2008; Gretton and Györfi, 2010; Lopez-Paz et al., 2013) to generative adversarial networks (Dziugaite et al., 2015; Li et al., 2015). While KMEs have already been extended to embed not only probability measures, but also signed finite measures, a first contribution of this paper is to show that they can be extended even further to embed generalized measures called *Schwartz-distributions*. For an introduction to Schwartz-distributions—which we will now simply call a distribution, as opposed to a (signed) measure—see Appendix B. Furthermore, we show that for smooth and translation-invariant kernels, if the KME is injective over the set of probability measures, then it remains injective when extended to some Schwartz-distribution sets.

Our second contribution concerns the notions of universal, characteristic and strictly positive definite (s.p.d.) kernels. They are of prime importance to guarantee the consistency of many regression-type or MMD-based algorithms (Steinwart, 2001; Steinwart and Christmann, 2008). While these notions were originally introduced in very different contexts, they were shown to be connected in many ways which were eventually summarized in Figure 1 of Sriperumbudur et al. (2011). But by handling separately all the many variants of universal, characteristic and s.p.d. kernels that had been introduced, this figure—and the general machine learning literature—somehow missed the underlying very general duality principle that connects these notions. By giving a unified definition of these three concepts, we will make their link explicit, easy to remember, and immediate to generalize to Schwartz-distributions and other spaces.

Our third contribution concerns the MMD semi-metric. Through a series of articles, Sriperumbudur et al. (2010b; 2016) gave various sufficient conditions for a kernel to *metrize the weak-convergence of probability measures*, which means that a sequence of probability measures converges in MMD distance if and only if (iff) it converges weakly. Here, we generalize these results and give the first complete characterization of the kernels that metrize weak convergence when the underlying space $\mathcal{X}$ is locally compact.

Finally, we develop a few calculus rules to work with KMEs of Schwartz distributions. In particular, we prove the following formulae:

$$\left\langle f\,,\ \int k(.,x)\,\mathrm{d}D(x)\right\rangle_k = \int \langle f\,,\,k(.,x)\rangle_k\,\mathrm{d}D(x) \qquad \text{(Definition of KME)}$$

$$\left\langle \int k(.,y)\,\mathrm{d}D(y)\,,\ \int k(.,x)\,\mathrm{d}T(x)\right\rangle_k = \int k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x) \qquad \text{(Fubini)}$$

$$\int k(.,x)\,\mathrm{d}(\partial^p S)(x) = (-1)^{|p|}\int \partial^{(0,p)}k(.,x)\,\mathrm{d}S(x). \qquad \text{(Differentiation)}$$

The first and second lines are standard calculus rules for KMEs when applied with two probability measures D and T. We extend them to distributions. The third line however is specific to distributions. It uses the distributional derivative ('∂') which extends the usual derivative of functions to signed measures and distributions. For a quick introduction to Schwartz distributions and their derivatives see Appendix B.

The structure of this paper roughly follows this exposition. After fixing our notations, Section 2 introduces KMEs of measures and distributions. In Section 3 we define the concepts of universal, characteristic and s.p.d. kernels and prove their equivalence. Section 4 compares convergence in MMD with other modes of convergence for measures and distributions. Section 5 focuses specifically on KMEs of Schwartz-distributions, and Section 6 gives a brief overview of the related work and concludes.

1.1. Definitions and Notations

Let $\mathbb{N}$, $\mathbb{R}$ and $\mathbb{C}$ be the sets of non-negative integers, of reals and of complex numbers. The input set $\mathcal{X}$ of all considered kernels and functions will be locally compact and Hausdorff. This includes any Euclidian spaces or smooth manifolds, but no infinite-dimensional Banach-space. Whenever referring to differentiable functions or to distributions of order ≥ 1, we will *implicitly* assume that $\mathcal{X}$ is an open subset of $\mathbb{R}^d$ for some $d > 0$.

A *kernel* $k\colon\ \mathcal{X}\times\mathcal{X} \longrightarrow \mathbb{C}$ is a positive definite function, meaning that for all $n \in \mathbb{N}\backslash\{0\}$, all $\lambda_1, \ldots, \lambda_n \in \mathbb{C}$, and all $x_1, x_2, \ldots x_n \in \mathcal{X}$, $\sum_{i,j=1}^n \lambda_i k(x_i, x_j)\overline{\lambda_j} \geq 0$. For $p = (p_1, p_2, \ldots, p_d) \in \mathbb{N}^d$ and $f\colon\ \mathcal{X} \longrightarrow \mathbb{C}$, we define $|p| := \sum_{i=1}^d p_i$ and $\partial^p f := \frac{\partial^{|p|} f}{\partial^{p_1}x_1 \partial^{p_2}x_2 \cdots \partial^{p_d}x_d}$. For $m \in \mathbb{N}\cup\{\infty\}$, we say that f (resp. k) is m-times (resp. (m,m)-times) continuously differentiable and write $f \in \mathcal{C}^m$ (resp. $k \in \mathcal{C}^{(m,m)}$), if for any p with $|p| = m$, $\partial^p f$ (resp. $\partial^{(p,p)}k$) exists and is continuous. $\mathcal{C}_b^m$ (resp. $\mathcal{C}_0^m$, $\mathcal{C}_c^m$) is the subsets of $\mathcal{C}^m$ for which $\partial^p f$ is bounded (resp. converges to 0 at infinity, resp. has compact support) whenever $|p| \leq m$. Whenever $m = 0$, we may drop the superscript m. By default, we equip $\mathcal{C}_*^m$ ($* \in \{\emptyset, b, 0, c\}$) with their natural topologies (see Introduction of Simon-Gabriel and Schölkopf 2016 or Treves 1967). We write $k \in \mathcal{C}_0^{(m,m)}$ whenever k is bounded, (m,m)-times continuously differentiable and for all $|p| \leq m$ and $x \in \mathcal{X}$, $\partial^{(p,p)}k(.,x) \in \mathcal{C}_0$.

We call *space of functions* and denote by $\mathcal{F}$ any locally convex (loc. cv.) topological vector space (TVS) of functions (see Appendix C and Treves 1967). Loc. cv. TVSs include all Banach- or Fréchet-spaces and all function spaces defined in this paper.

The dual $\mathcal{F}'$ of a space of functions $\mathcal{F}$ is the space of *continuous* linear forms over $\mathcal{F}$. We denote $\mathcal{M}_\delta$, $\mathcal{E}^m$, $\mathcal{D}_{L^1}^m$ and $\mathcal{D}^m$ the duals of $\mathbb{C}^\mathcal{X}$, $\mathcal{C}^m$, $\mathcal{C}_0^m$ and $\mathcal{C}_c^m$ respectively. By identifying each signed measure μ with a linear functional of the form $f \longmapsto \int f\,\mathrm{d}\mu$, the Riesz-Markov-Kakutani representation theorem (see Appendix C) identifies $\mathcal{D}^0$ (resp. $\mathcal{D}_{L^1}^0$, $\mathcal{E}^0$ and $\mathcal{M}_\delta$) with the set $\mathcal{M}_r$ (resp. $\mathcal{M}_f$, $\mathcal{M}_c$, $\mathcal{M}_\delta$) of signed regular Borel measures (resp. with finite total variation, with compact support, with finite support). By definition, $\mathcal{D}^\infty$ is the set of all Schwartz-distributions, but all duals defined above can be seen as subsets of $\mathcal{D}^\infty$ and are therefore sets of Schwartz-distributions. Any element μ of $\mathcal{M}_r$ will be called a measure, any element of $\mathcal{D}^\infty$ a distribution. See Appendix B for a brief introduction to distributions and their connection to measures. We extend the usual notation $\mu(f) := \int f(x)\,\mathrm{d}\mu(x)$ for measures μ to distributions D: $D(f) =: \int f(x)\,\mathrm{d}D(x)$. Given a KME Φ_k and two embeddable distributions D, T (see Definition 1), we define

$$\langle D\,,\,T\rangle_k := \langle \Phi_k(D)\,,\,\Phi_k(T)\rangle_k \quad \text{and} \quad \|D\|_k := \|\Phi_k(D)\|_k\ .$$

where $\langle .\,,\,.\rangle_k$ is the inner product of the RKHS $\mathcal{H}_k$ of k. To avoid introducing a new name, we call $\|D\|_k$ the maximum mean discrepancy (MMD) of D, even though the term "discrepancy" usually specifically designates a distance between two distributions rather than the norm of a single one. Given two topological sets $\mathcal{S}_1, \mathcal{S}_2$, we write

$$\mathcal{S}_1 \hookrightarrow \mathcal{S}_2$$

and say that $\mathcal{S}_1$ *is continuously contained in* $\mathcal{S}_2$ if $\mathcal{S}_1 \subset \mathcal{S}_2$ and if the topology of $\mathcal{S}_1$ is stronger than the topology induced by $\mathcal{S}_2$. For a general introduction to topology, TVSs and distributions, we recommend Treves (1967).

2. Kernel Mean Embeddings of Distributions

In this section, we show how to embed general distribution spaces into an RKHS. To do so, we redefine the integral $\int k(.,x)\,\mathrm{d}\mu(x)$ so as to be well-defined even if μ is a distribution. It is often defined as a Bochner-integral; here we instead use the *weak-* (or *Pettis-*) integral:

Definition 1 (Weak Integral and KME) *Let D be a linear form over a space of functions $\mathcal{F}$. Let $\vec{\varphi}\colon\ \mathcal{X} \longrightarrow \mathcal{H}_k$ be an RKHS-valued function such that for any $f \in \mathcal{H}_k$, $x \longmapsto \langle f\,,\,\vec{\varphi}(x)\rangle_k \in \mathcal{F}$. Then $\vec{\varphi}\colon\ \mathcal{X} \longrightarrow \mathcal{H}_k$ is weakly integrable with respect to (w.r.t.) D if there exists a function in $\mathcal{H}_k$, written $\int \vec{\varphi}(x)\,\mathrm{d}D(x)$, such that*

$$\forall f \in \mathcal{H}_k, \quad \left\langle f\,,\,\int \vec{\varphi}(x)\,\mathrm{d}D(x)\right\rangle_k = \int \langle f\,,\,\vec{\varphi}(x)\rangle_k\,\mathrm{d}\bar{D}(x)\,, \tag{1}$$

where the right-hand-side stands for $\bar{D}\,(x \mapsto \langle f\,,\,\vec{\varphi}(x)\rangle_k)$ and $\bar{D}$ denotes the complex-conjugate of D. If $\vec{\varphi}(x) = k(.,x)$, we call $\int k(.,x)\,\mathrm{d}D(x)$ the kernel mean embedding (KME) of D and say that D embeds into $\mathcal{H}_k$. We denote $\Phi_{\vec{\varphi}}$ the map $\Phi_{\vec{\varphi}}\colon\ D \longmapsto \int \vec{\varphi}(x)\,\mathrm{d}D(x)$.

This definition extends the usual Bochner-integral: if $\vec{\varphi}$ is Bochner-integrable w.r.t. a measure $\mu \in \mathcal{M}_r$, then $\vec{\varphi}$ is weakly integrable w.r.t. μ and the integrals coincide (Schwabik, 2005, Prop. 2.3.1). In particular, if $x \longmapsto \|\vec{\varphi}(x)\|_k$ is Lebesgue-integrable, then $\vec{\varphi}$ is Bochner integrable, thus weakly integrable.

The general definition with $\vec{\varphi}$ instead of $k(.,x)$ will be useful in Section 5. But for now, let us concentrate on KMEs where $\vec{\varphi}(x) = k(.,x)$. Kernels satisfy the so-called *reproducing property*: for any $f \in \mathcal{H}_k$, $f(x) = \langle f\,,\,k(.,x)\rangle_k$. Therefore, the condition for all $f \in \mathcal{H}_k$ $x \longmapsto \langle f\,,\,\vec{\varphi}(x)\rangle_k \in \mathcal{F}$ reduces to $\mathcal{H}_k \subset \mathcal{F}$, and Equation (1) reads:

$$\forall f \in \mathcal{H}_k, \qquad \left\langle f\,,\,\int k(.,x)\,\mathrm{d}D(x)\right\rangle_k = \bar{D}(f)\,. \tag{2}$$

Thus, by the Riesz representation theorem (see Appendix C), D embeds into $\mathcal{H}_k$ iff it defines a continuous linear form over $\mathcal{H}_k$. And in that case, its KME $\int k(.,x)\,\mathrm{d}D(x)$ is the Riesz-representer of $\bar{D}$ restricted to $\mathcal{H}_k$. Thus, for an embeddable space of distributions $\mathcal{D}$, the embedding Φ_k can be decomposed as follows:

$$\Phi_k : \left\{\begin{array}{ccccc} \mathcal{D} & \longrightarrow & \mathcal{H}_k' & \longrightarrow & \mathcal{H}_k \\ & \text{Conjugate restriction} & & \text{Riesz representer} & \\ D & \longmapsto & \bar{D}\big|_{\mathcal{H}_k} & \longmapsto & \int k(.,x)\,\mathrm{d}D(x) \end{array}\right. . \tag{3}$$

To know if D is continuous over $\mathcal{H}_k$, we use the following lemma, and its applications.

Lemma 2 *If $\mathcal{H}_k \hookrightarrow \mathcal{F}$, then $\mathcal{F}'$ embeds into $\mathcal{H}_k$.*

Proof Suppose that $\mathcal{H}_k \hookrightarrow \mathcal{F}$. Let $D \in \mathcal{F}'$ and let $f, f_1, f_2, \ldots \in \mathcal{H}_k$. If $f_n \to f$ in $\mathcal{H}_k$ then $f_n \to f$ in $\mathcal{F}$, thus $D(f_n) \to D(f)$. Thus D is a continuous linear form over $\mathcal{H}_k$. ■

In practice we typically use one of the following two corollaries (proofs in Appendices A.1 and A.2). The space $(\mathcal{C}_b)_c$ that they mention will be introduced in the discussions following Theorem 6. It has the same elements as $\mathcal{C}_b$, but carries a weaker topology.

Corollary 3 (Embedding of Measures) *$\mathcal{H}_k \subset \mathcal{C}_0$ (resp. $\mathcal{H}_k \subset \mathcal{C}_b$, resp. $\mathcal{H}_k \subset \mathcal{C}$) iff the two following conditions hold.*

(i) For all $x \in \mathcal{X}$, $k(.,x) \in \mathcal{C}_0$ (resp. $k(.,x) \in \mathcal{C}_b$, resp. $k(.,x) \in \mathcal{C}$).
(ii) $x \longmapsto k(x,x)$ is bounded (resp. bounded, resp. locally *bounded, meaning that, for each $y \in \mathcal{X}$, there exists a (compact) neighborhood of y on which $x \longmapsto k(x,x)$ is bounded.).*

If so, then $\mathcal{H}_k \hookrightarrow \mathcal{C}_0$ (resp. $\mathcal{H}_k \hookrightarrow \mathcal{C}_b$, thus $\mathcal{H}_k \hookrightarrow (\mathcal{C}_b)_c$, resp. $\mathcal{H}_k \hookrightarrow \mathcal{C}$) and $\mathcal{M}_f$ (resp. $\mathcal{M}_f$, resp. $\mathcal{M}_c$) embeds into $\mathcal{H}_k$.

Corollary 4 (Embedding of Distributions)
If $k \in \mathcal{C}^{(m,m)}$, then $\mathcal{H}_k \hookrightarrow \mathcal{C}^m$, thus $\mathcal{E}^m$ embeds into $\mathcal{H}_k$.
If $k \in \mathcal{C}_0^{(m,m)}$, then $\mathcal{H}_k \hookrightarrow \mathcal{C}_0^m$, thus $\mathcal{D}_{L^1}^m$ embeds into $\mathcal{H}_k$.
If $k \in \mathcal{C}_b^{(m,m)}$, then $\mathcal{H}_k \hookrightarrow \mathcal{C}_b^m$, thus $\mathcal{H}_k \hookrightarrow (\mathcal{C}_b^m)_c$, thus $\mathcal{D}_{L^1}^m$ embeds into $\mathcal{H}_k$.

Corollary 3 applied to $\mathcal{C}_b$ shows that $\mathcal{H}_k$ is (continuously) contained in $\mathcal{C}_b$ iff k is bounded and separately continuous. As discovered by Lehtö (1952), there also exist kernels which are not continuous but whose RKHS $\mathcal{H}_k$ is contained in $\mathcal{C}_b$. So the conditions in Corollary 4 are sufficient, but in general not necessary. Concerning Lemma 2, note that it not only requires $\mathcal{H}_k \subset \mathcal{F}$, but also that $\mathcal{H}_k$ carries a stronger topology than $\mathcal{F}$. Otherwise there might exist a continuous form over $\mathcal{F}$ that is defined but non-continuous over $\mathcal{H}_k$. However, Corollary 3 shows that this cannot happen for $\mathcal{C}_*$, because if $\mathcal{H}_k \subset \mathcal{C}_*$ then $\mathcal{H}_k \hookrightarrow \mathcal{C}_*$. Although this also holds for $m = \infty$ (Simon-Gabriel and Schölkopf, 2016, Prop.4 & Comments), we do not know whether it extends to any $m > 0$.

3. Universal, Characteristic and S.P.D. Kernels

The literature distinguishes various variants of universal, characteristic and s.p.d. kernels, such as c-, $cc-$ or c_0-universal kernels, s.p.d. and integrally strictly positive definite ($\int$s.p.d.) kernels. They are all special cases of the following unifying definitions.

Definition 5 *Let k be a kernel, $\mathcal{F}$ be a space of functions such that $\mathcal{H}_k \subset \mathcal{F}$, and $\mathcal{D}$ be an embeddable subset of $\mathcal{F}'$ (e.g. an embeddable set of distributions). We say that k is*

- ▷ universal over $\mathcal{F}$ *if $\mathcal{H}_k$ is dense in $\mathcal{F}$.*
- ▷ characteristic to $\mathcal{D}$ *if the KME Φ_k is injective over $\mathcal{D}$.*
- ▷ strictly positive definite (s.p.d.) over $\mathcal{D}$ *if:* $\forall D \in \mathcal{D},\ \|\Phi_k(D)\|_k^2 = 0 \Rightarrow D = 0$.

A universal kernel over $\mathcal{C}^m$ (resp. $\mathcal{C}_0^m$) will be said c^m- (resp. c_0^m-) universal (without the superscript when $m = 0$). A characteristic kernel to the set $\mathcal{P}$ of probability measures will simply be called characteristic.

In general, instead of writing $\|\Phi_k(D)\|_k$ and $\langle \Phi_k(D), \Phi_k(T)\rangle_k$, we will write $\|D\|_k$ and $\langle D, T\rangle_k$. These definitions encompass the usual s.p.d. definitions. Denoting δ_x the Dirac measure concentrated on x, what is usually called

- ▷ s.p.d. corresponds to $\mathcal{D} = \mathcal{M}_\delta$, i.e.:

$$\forall \mu = \sum_{i=1}^n \lambda_i \delta_{x_i} \in \mathcal{M}_\delta : \quad \|\mu\|_k^2 = \sum_{i,j=1}^n \lambda_i k(x_i, x_j) \bar{\lambda}_j = 0 \quad \Rightarrow \quad \lambda_1 = \ldots = \lambda_n = 0 \ .$$

- ▷ conditionally s.p.d. corresponds to $\mathcal{D} = \mathcal{M}_\delta^0$ where $\mathcal{M}_\delta^0 := \{\mu \in \mathcal{M}_\delta : \mu(\mathcal{X}) = 0\}$, i.e.:

$$\left.\begin{array}{l} \forall \mu = \sum_{i=1}^n \lambda_i \delta_{x_i} \in \mathcal{M}_\delta \\ \text{s.t. } \sum_{i=1}^n \lambda_i = 0 \end{array}\right\} : \ \|\mu\|_k^2 = \sum_{i,j=1}^n \lambda_i k(x_i, x_j) \bar{\lambda}_j = 0 \quad \Rightarrow \quad \lambda_1 = \ldots = \lambda_n = 0 \ .$$

- ▷ $\int$s.p.d. corresponds to $\mathcal{D} = \mathcal{M}_f$, i.e.:

$$\forall \mu \in \mathcal{M}_f : \quad \|\mu\|_k^2 = \iint k(x,y) \, \mathrm{d}\mu(x) \, \mathrm{d}\bar{\mu}(y) = 0 \quad \Rightarrow \quad \mu = 0 \ .$$

Let us now state the general link between universal, characteristic and s.p.d. kernels, which is the key that underlies Figure 1 of Sriperumbudur et al. (2011).

Theorem 6 *If $\mathcal{H}_k \hookrightarrow \mathcal{F}$, then the following statements are equivalent.*

(i) k is universal over $\mathcal{F}$.
(ii) k is characteristic to $\mathcal{F}'$.
(iii) k is strictly positive definite over $\mathcal{F}'$.

Proof Equivalence of (ii) & (iii): Saying that $\|\Phi_k(D)\|_k = 0$ is equivalent to saying $\Phi_k(D) = 0$. Thus Φ_k is s.p.d. over $\mathcal{F}'$ iff the $\mathrm{Ker}(\Phi_k)$ (meaning the vector space that is mapped to 0 via Φ_k) is reduced to $\{0\}$, which happens iff Φ_k is injective over $\mathcal{F}'$.

Equivalence of (i) & (ii): Φ_k is the conjugate restriction operator $|_{\mathcal{H}_k} : D \longmapsto \bar{D}|_{\mathcal{H}_k}$ composed with the Riesz representer mapping (Diagram Eq.3). The Riesz representer map is injective, so Φ_k is injective iff $|_{\mathcal{H}_k}$ is injective. Now, if $\mathcal{H}_k$ is dense in $\mathcal{F}$, then, by continuity, any $D \in \mathcal{F}'$ is uniquely defined by its values taken on $\mathcal{H}_k$. Thus $|_{\mathcal{H}_k}$ is injective. Reciprocally, if $\mathcal{H}_k$ is not dense in $\mathcal{F}$, then, by the Hahn-Banach theorem (Treves, 1967, Thm.18.1, Cor.3), there exists two different elements in $\mathcal{F}'$ that coincide on $\mathcal{H}_k$ but not on the entire space $\mathcal{F}$. So $|_{\mathcal{H}_k}$ is not injective. Thus $|_{\mathcal{H}_k}$ is injective iff $\mathcal{H}_k$ is dense in $\mathcal{F}$. ■

To apply this theorem it suffices to find so-called *duality pairs* $(\mathcal{F}, \mathcal{F}')$ such that $\mathcal{H}_k \hookrightarrow \mathcal{F}$. Table 1 lists several such pairs. It shows in particular the well-known equivalence between c- (resp. c_0-) universal kernels and characteristic kernels to $\mathcal{M}_c$ (resp. $\mathcal{M}_f$) (Sriperumbudur et al., 2008). But we now discover that s.p.d. kernels over $\mathcal{M}_\delta$ can also be characterized in terms of universality over $\mathbb{C}^{\mathcal{X}}$, because $(\mathbb{C}^{\mathcal{X}})' = \mathcal{M}_\delta$ (Duc-Jacquet, 1973, p.II.35). And we directly get the generalization to distributions and c_*^m-universality.

However, Theorem 6 leaves open the important case where k is characteristic (to $\mathcal{P}$). Of course, as $\mathcal{P}$ is contained in $\mathcal{M}_f$, it shows that a c_0-universal kernel must be characteristic. But to really characterize characteristic kernels in terms of universality, we would need to find a predual of $\mathcal{P}$, meaning a space $\mathcal{F}$ such that $\mathcal{F}' = \mathcal{P}$. This is hardly possible, as $\mathcal{P}$ is not even a vector space. However, we will see in Theorem 8 that k is characteristic iff k is

Universal	Characteristic	S.P.D.	Name	Proof
$\mathscr{F}$	$\mathscr{F}'$	$\mathscr{F}'$	/	Thm. 6
$\mathbb{C}^{\mathcal{X}}$	$\mathcal{M}_\delta$	$\mathcal{M}_\delta$	s.p.d.	Thm. 6
$\mathbb{C}^{\mathcal{X}}/\mathbb{1}$	$\mathcal{M}_\delta^0$	$\mathcal{M}_\delta^0$	conditionally s.p.d.	Prop. 7
$\mathscr{C}$	$\mathcal{M}_c$	$\mathcal{M}_c$	c-universal (or cc-universal)	Thm. 6
$\mathscr{C}_0$	$\mathcal{M}_f$	$\mathcal{M}_f$	c_0-universal	Thm. 6
$(\mathscr{C}_b)_c$	$\mathcal{M}_f$	$\mathcal{M}_f$	$\int$spd	Thm. 6
$((\mathscr{C}_b)_c)/\mathbb{1}$	$\mathscr{P}$ (or $\mathcal{M}_f^0$)	$\mathcal{M}_f^0$	characteristic	Prop. 7
$\mathscr{C}^m$	$\mathscr{E}^m$	$\mathscr{E}^m$	c^m-universal	Thm. 6
$\mathscr{C}_0^m$	$\mathcal{D}_{L^1}^m$	$\mathcal{D}_{L^1}^m$	c_0^m-universal	Thm. 6
$(\mathscr{C}_b^m)_c$	$\mathcal{D}_{L^1}^m$	$\mathcal{D}_{L^1}^m$	/	Thm. 6

Table 1: Equivalence between the notions of universal, characteristic and s.p.d. kernels.

characteristic to the vector space $\mathcal{M}_f^0 := \{\mu \in \mathcal{M}_f : \mu(\mathcal{X}) = 0\}$. So if we find a predual of $\mathcal{M}_f^0$, then we get an analog of Theorem 6 applied to $\mathscr{P}$. Let us do so now.

As $\mathcal{M}_f^0$ is the hyperplane of $\mathcal{M}_f$ that is given by the equation $\int 1 \,\mathrm{d}\mu = 0$, our idea is to take a predual $\mathscr{F}$ of $\mathcal{M}_f$ and consider the quotient $\mathscr{F}/\mathbb{1}$ of $\mathscr{F}$ divided by the constant function $\mathbb{1}$. Proposition 35.5 of Treves (1967) would then show that $(\mathscr{F}/\mathbb{1})' = \mathcal{M}_f^0$. But if we take the usual predual of $\mathcal{M}_f$, $\mathscr{F} = \mathscr{C}_0$, then $\mathbb{1} \notin \mathscr{F}$, so the quotient $\mathscr{F}/\mathbb{1}$ is undefined. However, preduals are not unique, so let us try with another space $\mathscr{F}$ that contains $\mathbb{1}$, for example $\mathscr{F} = \mathscr{C}_b$. This time $\mathbb{1} \in \mathscr{F}$, but now the problem is that $\mathscr{F}'$ is in general strictly bigger than $\mathcal{M}_f$ (Fremlin et al., 1972, Sec. 2, §2) whereas we want $\mathscr{F}' = \mathcal{M}_f$. The trick now is to keep $\mathscr{C}_b$, but equip it with a weaker topology than the usual one, so that $\mathscr{F}'$ becomes smaller. Intuitively, the reason for this decrease of $\mathscr{F}'$ is that, by weakening the topology of $\mathscr{F}$, we let more sequences converge in $\mathscr{F}$. This makes it more difficult for a functional over $\mathscr{F}$ to be continuous, because for any converging sequence in $\mathscr{F}$, its images need to converge. Thus some of the linear functionals that were continuous for the original topology of $\mathscr{F}$ get "kicked out" of $\mathscr{F}'$ when $\mathscr{F}$ carries a weaker topology. Now the only remaining step is to find a topology such that $\mathscr{F}'$ shrinks exactly to $\mathcal{M}_f$. There are at least two such topologies: one defined by Schwartz (1954, p.100-101) and another, called the strict topology, whose definition can be found in Fremlin et al. (1972). Denoting τ_c either of these topologies, and $(\mathscr{C}_b)_c$ the space $\mathscr{C}_b$ equipped with τ_c, we finally get $((\mathscr{C}_b)_c)' = \mathcal{M}_f$, and thus:

Proposition 7 $((\mathscr{C}_b)_c/\mathbb{1})' = \mathcal{M}_f^0$. *Thus, if $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b)_c$, then k is characteristic to $\mathscr{P}$ iff k is universal over the quotient space $((\mathscr{C}_b)_c/\mathbb{1})$.*

Proof That $((\mathscr{C}_b)_c)' = \mathcal{M}_f$ is proven in Fremlin et al. (1972, Thm. 1) or Schwartz (1954, p.100-101). Proposition 35.5 of Treves (1967) then implies $((\mathscr{C}_b)_c/\mathbb{1})' = \mathcal{M}_f^0$ (because $\mathcal{M}_f^0$ is the so-called *polar set* of $\mathbb{1}$; see Treves 1967). Theorem 6 implies the rest. ■

For our purposes, the exact definition of τ_c does not matter. What matters more is that τ_c is weaker than the usual topology of $\mathscr{C}_b$, so that if $\mathcal{H}_k \hookrightarrow \mathscr{C}_b$, then $\mathcal{H}_k \hookrightarrow (\mathscr{C}_b)_c$. Proposition 7 thus applies every time that $\mathcal{H}_k \subset \mathscr{C}_b$ (see Corollaries 3 and 4). However, we do not know of any practical application of Proposition 7, except that it completes our overall picture of the equivalences between universal, characteristic and s.p.d. kernels. Let us also mention that, similarly to Proposition 7, as $(\mathbb{C}^{\mathcal{X}})' = \mathcal{M}_\delta$, we also have $(\mathbb{C}^{\mathcal{X}}/\mathbb{1})' = \mathcal{M}_\delta^0$. So conditionnally s.p.d. kernels (meaning s.p.d. over $\mathcal{M}_\delta^0$) are universal to $\mathbb{C}^{\mathcal{X}}/\mathbb{1}$.

We now prove what we announced and used earlier: a kernel is characteristic to $\mathscr{P}$ iff it is characteristic to $\mathcal{M}_f^0$. We add a few other characterisations which are probably more useful in practice. They rely on the following observation: as $\mathcal{M}_f^0$ is a hyperplane of $\mathcal{M}_f$, saying that k is characteristic to $\mathscr{P}$ is almost the same than saying that it is characteristic to $\mathcal{M}_f$, i.e. $\int$s.p.d. (Thm. 6): after all, there is only one dimension needed to go from $\mathcal{M}_f^0$ to $\mathcal{M}_f$. Thus there should be a way to construct an $\int$s.p.d. kernel out of any characteristic kernel. This is what is described here and proven in Appendix A.3.

Theorem 8 (Characteristic Kernels) *Let k_0 be a kernel. The following is equivalent.*

(i) *k_0 is characteristic to $\mathscr{P}$.*
(ii) *k_0 is characteristic to $\mathcal{M}_f^0$.*
(iii) *There exists $\epsilon \in \mathbb{R}$ such that the kernel $k(x,y) := k_0(x,y) + \epsilon^2$ is $\int$s.p.d..*
(iv) *For all $\epsilon \in \mathbb{R}\backslash\{0\}$, the kernel $k(x,y) := k_0(x,y) + \epsilon^2$ is $\int$s.p.d..*
(v) *There exists an RKHS $\mathcal{H}_k$ with kernel k and a measure $\nu_0 \in \mathcal{M}_f \backslash \mathcal{M}_f^0$ such that k is characteristic to $\mathcal{M}_f$ and $k_0(x,y) = \langle \delta_x - \nu_0 \,,\, \delta_y - \nu_0 \rangle_k$.*

Under these conditions, k_0 and k induce the same MMD semi-metric in $\mathcal{M}_f^0$ and in $\mathscr{P}$.

We will use this theorem to prove Theorem 12. Intuitively, a characteristic kernel guarantees that any two different signed measures μ_1, μ_2 with same total mass get mapped to two different functions in the RKHS. This is captured by (ii) which arbitrarily focuses on the special case where the total mass is 0. When they have different total masses however, they may still get mapped to a same function f, except if, like in (iii) and (iv), we add a positive constant to the kernel. In that case, μ_1 and μ_2 get mapped to the functions $f + \mu_1(\mathcal{X})\mathbb{1}$ and $f + \mu_2(\mathcal{X})\mathbb{1}$ which are now different, because $\mu_1(\mathcal{X}) \neq \mu_2(\mathcal{X})$. Intuively, by adding a positive constant to our kernel, we added one dimension to the RKHS (carried by the function $\mathbb{1}$) that explicitly 'checks' if two measures have the same mass. Finally, (v) tells us that, out of any $\int$s.p.d. kernel k, we can construct a characteristic kernel k_0 that is not $\int$s.p.d. anymore and vice-versa.

4. Topology Induced by k

Remember that for any distribution D of a set of embeddable distributions $\mathcal{D}$ we defined $\|D\|_k := \|\Phi_k(D)\|_k$ and called $\|D\|_k$ the Maximum Mean Discrepancy (MMD) of D. Doing this defines a new topology on $\mathcal{D}$, in which a net D_α converges to D iff $\|D_\alpha - D\|_k$ converges to 0. (A reader unfamiliar with nets may think of them as sequences where the index α

can be continuous; see Berg et al. 1984.) In this section, we investigate how convergence in MMD compares with other types of convergences defined on $\mathcal{D}$ that we now shortly present.

We defined $\mathcal{D}$ as a subset of a dual space $\mathcal{F}'$, so $\mathcal{D}$ will carry the topology induced by $\mathcal{F}'$. Many topologies can be defined on dual spaces, but the two most prominent ones, which we will consider here, are the *weak-** and the *strong* topology, denoted $w(\mathcal{F}', \mathcal{F})$ and $b(\mathcal{F}', \mathcal{F})$ respectively, or simply $w*$ and b. The weak-$*$ topology is the topology of pointwise convergence (where by 'point', we mean a function in $\mathcal{F}$), while the strong topology corresponds to the uniform convergence over the bounded subsets of $\mathcal{F}$ (see Eq. 4). Bounded sets of a TVS are defined in Appendix C (Definition 24). By default, we equip $\mathcal{F}'$ with the strong topology and sometimes write $\mathcal{F}'_b$ to emphasize it. When $\mathcal{F}$ is a Banach space, the strong topology of $\mathcal{F}'$ is the topology of the operator norm $\|D\|_{\mathcal{F}'} := \sup_{\|f\|_{\mathcal{F}} \leq 1} |D(f)|$. In particular, strong convergence in $\mathcal{M}_f = (\mathcal{C}_0)'$ means convergence in total variation (TV) norm and weak-$*$ convergence in $\mathcal{M}_f$ means convergence for any function $f \in \mathcal{C}_0$. On $\mathcal{M}_f$, we will also consider the topology of pointwise convergence over $\mathcal{C}_b$ (instead of $\mathcal{C}_0$). It is widely used in probability theory where it is known as the *weak* (or narrow) convergence topology. We will denote it by σ. Importantly, the weak and weak-$*$ topologies of $\mathcal{M}_f$ coincide on $\mathcal{P}$ (but not on $\mathcal{M}_f$) (Berg et al., 1984, Chap. 2, Cor. 4.3). Finally, we define the weak RKHS convergence of embeddable distributions, denoted by $w-k$, as the pointwise convergence over $\mathcal{H}_k$. Note that D_α converges in $w-k$ to D iff their embeddings converge weakly (or equivalently weakly-$*$) in $\mathcal{H}_k$, in the sense that, for any $f \in \mathcal{H}_k$, $\langle f\,,\, \Phi_k(D_\alpha)\rangle_k$ converges to $\langle f\,,\, \Phi_k(D)\rangle_k$. The following summarizes the different convergence types.

$$
\begin{array}{lrll}
D_\alpha \xrightarrow{b} D := & \sup_{f\in\mathcal{B}} |D_\alpha(f) - D(f)| \longrightarrow 0 & \forall \text{ bounded } \mathcal{B} \subset \mathcal{F} & D_\alpha \in \mathcal{F}' \\
D_\alpha \xrightarrow{w*} D := & |D_\alpha(f) - D(f)| \longrightarrow 0 & \forall f \in \mathcal{F} & D_\alpha \in \mathcal{F}' \\
\mu_\alpha \xrightarrow{\sigma} \mu := & |\mu_\alpha(f) - \mu(f)| \longrightarrow 0 & \forall f \in \mathcal{C}_b & \mu_\alpha \in \mathcal{M}_f \\
D_\alpha \xrightarrow{w-k} D := & |D_\alpha(f) - D(f)| \longrightarrow 0 & \forall f \in \mathcal{H}_k & D_\alpha \text{ embeddable} \\
D_\alpha \xrightarrow{\|.\|_k} D := & \|D_\alpha - D\|_k \longrightarrow 0 & & D_\alpha \text{ embeddable}
\end{array}
\tag{4}
$$

4.1. Embeddings of Dual Spaces are Continuous

In this section, we show that the MMD topology is often weaker than other topologies τ defined on $\mathcal{D}$, meaning that if D_α converges to D in τ, then it also converges to D in MMD. Note that this is equivalent to saying that the KME of $\mathcal{D}_\tau$ (read '$\mathcal{D}$ equipped with τ') is continuous. We start with the following pretty coarse, yet very general result.

Proposition 9 *If $\mathcal{H}_k \hookrightarrow \mathcal{F}$, then $D_\alpha \xrightarrow{b} D \;\Rightarrow\; D_\alpha \xrightarrow{\|.\|_k} D$ and $D_\alpha \xrightarrow{w*} D \;\Rightarrow\; D_\alpha \xrightarrow{w-k} D$.*

Proof Proposition 9 states that the KME is continuous when both $\mathcal{F}'$ and $\mathcal{H}_k$ carry their strong or their weak-$*$ topology, which we now show. From Diagram Eq.(3), we know that the KME is the composition of the conjugate restriction operator with the Riesz representer map. The Riesz representer map is a topological (anti-)isomorphism between $\mathcal{H}'_k$ and $\mathcal{H}_k$, thus continuous (see Appendix C). And the restriction map is the adjoint (or transpose) of the canonical embedding map $\begin{array}{rccc} \imath: & \mathcal{H}_k & \longrightarrow & \mathcal{F} \\ & f & \longmapsto & f \end{array}$, thus continuous when both $\mathcal{F}'$ and $\mathcal{H}'_k$ carry their weak-$*$ or strong topologies (Treves, 1967, Prop.19.5 & Corollary). ■

Let us briefly comment on this result. The statement $D_\alpha \xrightarrow{w*} D \;\Rightarrow\; D_\alpha \xrightarrow{w-k} D$ is actually obvious, because $\mathcal{H}_k \subset \mathcal{F}$. Concerning strong convergence, Proposition 9 implies that, if $\mathcal{F}$ is a Banach space, then any net that converges for the dual norm $\|\cdot\|_{\mathcal{F}'}$ converges in MMD. Applying this with $\mathcal{F} = \mathcal{C}_0$ and $\mathcal{F}' = \mathcal{M}_f$ shows that convergence in TV norm implies convergence in MMD, or equivalently, that the TV norm is stronger than the MMD. Similar reasoning can be used to show that the MMD is weaker than the so-called Kantorovich(-Wasserstein) and the Dudley norms (see Example 1 in Simon-Gabriel and Schölkopf 2016). These results can also be found in Sriperumbudur et al. (2010b). However, the authors there directly bounded the MMD semi-norm by the target norm. This has the advantage of giving concrete bounds, but is more difficult to generalize if $\mathcal{F}$ is not a Banach space.

Though very general, Proposition 9 is pretty weak, as it only compares a strong with a strong and a weak-$*$ with a weak(-$*$) topology. But how does the weak-$*$ topology on $\mathcal{F}'$ compare with the strong topology of $\mathcal{H}_k$: does weak-$*$ convergence imply convergence in MMD? This question is discussed in details in Simon-Gabriel and Schölkopf (2016, Sec.7). The short answer is: not always, but sometimes; it depends on the space $\mathcal{F}'$. For example, if $k \in \mathcal{C}^{(m,m)}$, then weak-$*$ convergence in $\mathcal{E}^m$ implies convergence in MMD; but weak-$*$ convergence in $\mathcal{D}^m_{L^1}$ usually does not imply MMD convergence when $\mathcal{X}$ is non-compact. For us, the only thing we will need later is to know what happens on $\mathcal{M}_+$, the set of finite positive measures. The following lemma shows that weak convergence in $\mathcal{M}_+$ usually implies MMD convergence.

Lemma 10 *A bounded kernel k is continuous iff: $\forall\, \mu_\alpha, \mu \in \mathcal{M}_+$, $\mu_\alpha \xrightarrow{\sigma} \mu \implies \mu_\alpha \xrightarrow{\|.\|_k} \mu$.*

Proof We assume k bounded to ensure that any probability measure is embeddable. Now, suppose that weak convergence implies MMD convergence and take $x, y, x_0, y_0 \in \mathcal{X}$ such that $x \to x_0$ and $y \to y_0$. Then $\delta_x \xrightarrow{\sigma} \delta_{x_0}$ and $\delta_y \xrightarrow{\sigma} \delta_{y_0}$, so $\Phi_k(\delta_x) \to \Phi_k(\delta_{x_0})$ and $\Phi_k(\delta_y) \to \Phi_k(\delta_{y_0})$ in $\mathcal{H}_k$. And by continuity of the inner product:

$$k(x,y) = \langle \Phi_k(\delta_y)\,,\, \Phi_k(\delta_x)\rangle_k \to \langle \Phi_k(\delta_{y_0})\,,\, \Phi_k(\delta_{x_0})\rangle_k = k(x_0, y_0)\ ,$$

so k is continuous. Conversely, suppose that k is continuous, and let $\mu_\alpha \xrightarrow{\sigma} \mu$ in $\mathcal{M}_+$. The tensor-product mapping $\begin{array}{ccc} \mathcal{M}_+(\mathcal{X}) & \longrightarrow & \mathcal{M}_+(\mathcal{X}\times\mathcal{X}) \\ \mu & \longmapsto & \mu\otimes\mu \end{array}$ is weakly continuous (Berg et al., 1984, Chap.2, Thm.3.3). So by applying $\bar{\mu}_\alpha \otimes \mu_\alpha$ to a bounded continuous kernel k, we get

$$
\begin{aligned}
\|\Phi_k(\mu_\alpha) - \Phi_k(\mu)\|_k^2 &= \iint k(x,y)\, \mathrm{d}(\mu_\alpha - \mu)(y)\, \mathrm{d}(\bar{\mu}_\alpha - \bar{\mu})(x) \\
&= [\bar{\mu}_\alpha \otimes \mu_\alpha](k) - [\bar{\mu} \otimes \mu_\alpha](k) - [\bar{\mu}_\alpha \otimes \mu](k) + [\bar{\mu}\otimes\mu](k) \longrightarrow 0\ . \quad ■
\end{aligned}
$$

4.2. When Does k Metrize the Topology of $\mathcal{F}'$?

So far we focused on the question: when does convergence in $\mathcal{D}$ imply convergence in MMD. We now seek the opposite: when does MMD-convergence imply convergence in $\mathcal{D}$?

First, the kernel *must* be characteristic to $\mathcal{D}$. Otherwise, the MMD does not define a distance but only a semi-distance, so that the induced topology would not be Hausdorff. Second, we will suppose that $\mathcal{F}$ is barreled. This is a technical, yet very general assumption

that we use in the next theorem. The definition of a barreled space is given in Appendix C for completeness, but all that the reader should remember is that all Banach, Fréchet, Limit-Fréchet and *all function spaces defined in this paper are barreled*,[1] except $(\mathscr{C}_b^m)_c$.

Lemma 11 *Suppose that $\mathcal{F}$ is barreled, k is universal over $\mathcal{F}$, $\mathcal{H}_k \hookrightarrow \mathcal{F}$ and let $(D_\alpha)_\alpha$ be a bounded net in $\mathcal{F}_b'$. Then $D_\alpha \xrightarrow{w-k} D$ iff $D_\alpha \xrightarrow{w*} D$. Hence $D_\alpha \xrightarrow{\|.\|_k} D \;\Rightarrow\; D_\alpha \xrightarrow{w*} D$.*

Proof Proposition 32.5 of Treves (1967) shows that the weak topologies of $\mathcal{F}'$ and of $\mathcal{H}_k'$ coincide on so-called *equicontinuous* sets of $\mathcal{F}'$, and the Banach-Steinhaus theorem (see Appendix C) states that if $\mathcal{F}$ is barreled, then the equicontinuous sets of $\mathcal{F}'$ are exactly its bounded sets. This precisely means that if the net D_α is bounded in $\mathcal{F}'$, then $D_\alpha(f) \to D(f)$ for all $f \in \mathcal{F}$ iff it converges for all $f \in \mathcal{H}_k$. Now, if $\|D_\alpha - D\|_k \to 0$, then, by continuity of the inner product, $D_\alpha(f) - D(f) = \langle f\,,\, D_\alpha - D\rangle_k \to 0$ for any $f \in \mathcal{H}_k$. ■

Lemma 11 says that the weak-$*$ topologies of $\mathcal{F}'$ and of $\mathcal{H}_k$ coincide on subsets of $\mathcal{F}'$ that are bounded in the strong topology. But from the Banach-Steinhaus theorem (see App. C) we know that on barreled spaces it is equivalent to be bounded in strong or in weak topology. Hence the net D_α of Lemma 11 is bounded iff $\sup_\alpha |D_\alpha(f)| < \infty$ for all $f \in \mathcal{F}$. Nevertheless, it is not enough in general to show that $\sup_\alpha \|D_\alpha\|_k < \infty$. A bounded set in $\mathcal{M}_f$ is also a set whose measures have uniformly bounded total variation. The total variation of any probability measure being 1, $\mathcal{P}$ is bounded. So Lemma 11 shows that for continuous c_0-universal kernels, convergence of probability measures in MMD distance implies weak-$*$ convergence, which on $\mathcal{P}$ is the same as weak-convergence. But by Lemma 10 the reverse is true as well. Thus, *for a continuous c_0-universal kernel k, probability measures converge weakly iff they converge in MMD distance.* Such kernels are said to *metrize* the weak convergence on $\mathcal{P}$.

However, the condition that k be c_0-universal seems slightly too restrictive. Indeed, it is needed in Lemma 11 to ensure that the KME be characteristic to $\mathcal{M}_f$ (by Thm. 6 applied to $\mathcal{F} = \mathscr{C}_0$) so that the MMD be a metric over $\mathcal{M}_f$ (not only a semi-metric). But, to be a metric over $\mathcal{P}$, it would suffice that k be characteristic to $\mathcal{P}$, which is a slightly coarser assumption than c_0-universality. Is this condition enough to guarantee the metrization of weak-convergence in $\mathcal{P}$? The following theorem shows that it is.

Theorem 12 *A bounded kernel over a locally compact Hausdorff space $\mathcal{X}$ metrizes the weak convergence of probability measures iff it is continuous and characteristic (to $\mathcal{P}$).*

Proof [Theorem 12] If k metrizes the weak convergence over $\mathcal{P}$, then, by Lemma 10, k is continuous, and, for $\|.\|_k$ to be a norm, k needs to be characteristic. Conversely, if k is continuous, then by Lemma 10 weak convergence implies convergence in MMD. So it remains to show that MMD convergence implies weak convergence. To do so, we use Lemma 20 of the appendix, which states that for an $\int$s.p.d. kernel, MMD convergence of probability measures implies their weak convergence. Now k might not be $\int$s.p.d., but using Theorem 8(iv), we can transform it to a kernel $k_1 := k+1$ which induces the same MMD metric

1. $\mathbb{C}^{\mathcal{X}}$ is barreled, because it is a topological product $\prod_{\mathcal{X}} \mathbb{C}$ of barreled spaces. All other mentioned spaces are either Banach, Fréchet or Limit-Fréchet spaces, thus barreled (Treves, 1967, Prop. 33.2 & Cor.1-3).

over probability measures than k, but which is $\int$s.p.d. This concludes. ■

To the best of our knowledge, this is the first characterization of the class of kernels that metrize the weak-convergence of probability measures. For example Gaussian, Laplace, inverse-multiquadratic or Matérn kernels are continuous and characteristic, so they all metrize the weak convergence over $\mathcal{P}$. In general however, even if a kernel metrizes the weak convergence over $\mathcal{P}$, it usually does not metrize weak convergence over $\mathcal{M}_+$ or $\mathcal{M}_f$ (see Simon-Gabriel and Schölkopf 2016).

5. Kernel Mean Embeddings of Schwartz-Distributions

We extended KMEs of measures to Schwartz-distributions and showed that they are continuous, but we hardly said anything about what to do and how to work with distributions. We will now catch up by focusing on distributions only. In Section 5.1, we discuss and prove the Fubini and the Differentiation formulae featured in the introduction. In Section 5.2 we provide sufficient conditions for a translation-invariant kernel to be c_*^m-universal.

5.1. Distributional Calculus

Proposition 13 (Fubini) *Let D, T be two embeddable distributions into $\mathcal{H}_k$. Then:*

$$\begin{aligned}\langle D\,,\,T\rangle_k &= \iint k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x) = \iint k(x,y)\,\mathrm{d}\bar{T}(x)\,\mathrm{d}D(y) \\ \|D\|_k^2 &= \iint k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{D}(x) = \iint k(x,y)\,\mathrm{d}\bar{D}(x)\,\mathrm{d}D(y)\,,\end{aligned} \tag{5}$$

where $\iint k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x)$ is to be understood as $\bar{T}(\mathcal{I})$ with $\mathcal{I}(x) = \int k(x,y)\,\mathrm{d}D(y)$.

Proof Definition 1 of a KME, together with the property that $k(y,x) = \overline{k(x,y)}$ leads to:

$$\begin{aligned}\langle D\,,\,T\rangle_k &= \int_x \left\langle \int_y k(.,y)\,\mathrm{d}D(y)\,,\,k(.,x)\right\rangle_k \mathrm{d}\bar{T}(x) \\ &= \int_x \overline{\left\langle k(.,x)\,,\,\int_y k(.,y)\,\mathrm{d}D(y)\right\rangle_k}\,\mathrm{d}\bar{T}(x) \\ &= \int_x \overline{\int_y \langle k(.,x)\,,\,k(.,y)\rangle_k\,\mathrm{d}\bar{D}(y)}\,\mathrm{d}\bar{T}(x) \\ &= \iint k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x).\end{aligned}$$

To prove the right-most part of (5), use $\langle D\,,\,T\rangle_k = \overline{\langle T\,,\,D\rangle}_k$. ■

These formulae are well-known when D and T are probability measures. They show that if you know how to integrate a function (the kernel) w.r.t. a measure or a distribution, then you can compute its MMD norm. However, integrating w.r.t. a distribution that is not a measure can be tedious. But the following proposition gives us a way to convert an integration w.r.t. a distribution into an integration w.r.t. a measure.

Proposition 14 (Differentiation) *Let $k \in \mathscr{C}^{(m,m)}$ and $p \in \mathbb{N}^d$ such that $|p| \leq m$. A distribution D embeds into $\mathcal{H}_k$ via $\partial^{(0,p)}k$ iff $\partial^p D$ embeds into $\mathcal{H}_k$ via k. In that case,*

$$\Phi_k(\partial^p D) = (-1)^{|p|} \int [\partial^{(0,p)}k](.,x)\,\mathrm{d}D(x) = (-1)^{|p|}\,\Phi_{\partial^{(0,p)}k}(D)\,. \tag{6}$$

If moreover k is translation-invariant, then

$$\Phi_k(\partial^p D) = \partial^p[\Phi_k(D)]. \tag{7}$$

Proof The proof holds in the following equalities. For any $f \in \mathcal{H}_k$,

$$\begin{aligned}
\left\langle f\,,\, \int k(.,x)\,\mathrm{d}[\partial^p D](x) \right\rangle_k &= \int \langle f\,,\, k(.,x)\rangle_k\,\mathrm{d}[\partial^p \bar{D}](x) = [\partial^p \bar{D}](f) \\
&= (-1)^{|p|}\bar{D}(\partial^p f) \\
&= (-1)^{|p|}\bar{D}(\left\langle f\,,\, \partial^{(0,p)}k(.,x)\right\rangle_k) \\
&= (-1)^{|p|}\int \left\langle f\,,\, \partial^{(0,p)}k(.,x)\right\rangle_k \mathrm{d}\bar{D}(x) \\
&= \left\langle f\,,\, (-1)^{|p|}\int \partial^{(0,p)}k(.,x)\,\mathrm{d}D(x)\right\rangle_k .
\end{aligned}$$

The first line uses the definition of KMEs (1), the second the definition of distributional derivatives (see App. B), the third Lemma 19, the fourth line rewrites the previous line with our notation convention, and the fifth one uses again the definition of a weak integral (1). ■

Equation (7) describes a commutative diagram pictured in Figure 1: it states that with translation-invariant kernels, it is equivalent to take the (distributional) derivative of a distribution and embed it, or to embed it and take the (usual) derivative of the embedding. See Appendix B for an introduction to distributional derivatives. Note that for a signed measure μ with a $|p|$-times differentiable density q, the distributional derivative $\partial^p\mu$ is the signed measure with density $\partial^p_u q$, where ∂^p_u is the usual partial derivative operator. However, Proposition 14 becomes most useful when μ has no differentiable density, for example when μ is an empirical measure. Then there is no analytical formula for the derivative of μ, but we can still compute its KME analytically by using (6) or (7).

Example 1 *Let us illustrate Proposition 14 on KMEs of Gaussian probability measures μ_σ with density $q_\sigma(x) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-x^2/\sigma^2}$ using a Gaussian kernel $k(x,y) = e^{-(x-y)^2}$. When σ goes to zero, μ_σ gets more and more peaked around 0 and converges weakly to the Dirac measure $\mu_0 := \delta_0$. The KME of μ_σ is easy to compute and using (7) we get*

$$\begin{aligned}
\Phi_k(\mu_\sigma)(x) &= \frac{1}{\sqrt{1+2\sigma^2}}\,e^{-\frac{x^2}{1+2\sigma^2}} \\
\Phi_k(\partial\mu_\sigma)(x) &= \partial[\Phi_k(\mu_\sigma)] = -\frac{2x}{(1+2\sigma^2)^{3/2}}\,e^{-\frac{x^2}{1+2\sigma^2}}\,,
\end{aligned}$$

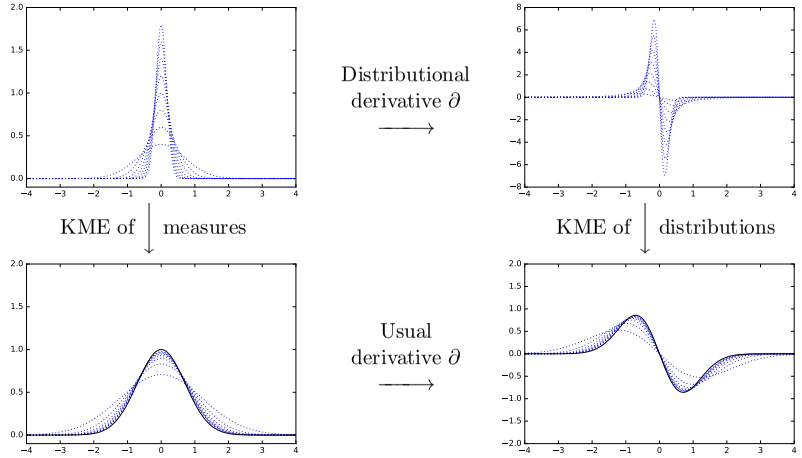


Figure 1: Densities of more and more peaked Gaussian probability measures μ_σ (top left) with their derivatives (top right) and their embeddings (below) using a Gaussian kernel (see Example 1). Equation (7) states that the diagram is commutative. When σ goes to 0, the Gaussians converge (weakly) to a Dirac mass δ_0, which has no density, but who's embedding is the solid black line (bottom left). The derivatives converge (weakly) to the Schwartz-distribution $\partial\delta_0$, which is not even a signed measure, but whose embedding (bottom right, black solid line) can easily be computed using (6) or (7). Moreover, the embeddings of μ_σ and $\partial\mu_\sigma$ converge (weakly) to the embeddings of μ_0 and $\partial\mu_0$, which illustrates Proposition 9.

where the formulae still hold when $\sigma = 0$. Figure 1 plots these embeddings for different σ's. Note that contrary to $\partial\mu_\sigma$ with $\sigma > 0$, $\partial\mu_0$ is not a signed measure (but a Schwartz-distribution) but it has a KME which, moreover, can easily be computed using (7). Notice also that on Figure 1 both the embeddings of μ_σ and $\partial\mu_\sigma$ converge (weakly) to the embeddings of μ_0 and $\partial\mu_0$. This illustrates Proposition 9.

Theoretically, (6) can be used to convert the KME of *any* distribution into a sum of KMEs of measures. In other words, the integral w.r.t. a distribution appearing in (1) can be converted into a sum of integrals w.r.t. signed measures. Here is how. Given a measure $\mu \in \mathcal{M}_f = \mathcal{D}^0_{L^1}(\mathbb{R})$, we may differentiate μ and get a new distribution $\partial\mu$ which may or may not be itself a measure.[2] But in any case, what will follow shows that $\partial\mu$ is in $\mathcal{D}^1_{L^1}(\mathbb{R})$. Thus the space of distributions that can be written as a sum $\mu_0 + \partial\mu_1$ of two finite measures μ_1, μ_2 is a subspace of $\mathcal{D}^1_{L^1}(\mathbb{R})$ and we may wonder how big exactly it is. Schwartz (1954, around p.100) showed that it is exactly the space $\mathcal{D}^1_{L^1}(\mathbb{R})$. More generally, he showed:

Lemma 15 (Schwartz) *For any $m \leq \infty$ and any distribution in $D \in \mathcal{D}^m_{L^1}$ (resp. $D \in \mathscr{C}^m$) there exists a* finite *family of measures $\mu_p \in \mathcal{M}_f$ (resp. $\mu_p \in \mathcal{M}_c$) such that $D = \sum_{|p|\leq m}\partial^p\mu_p$.*

2. Think for example of the Dirac measure: it is a measure, but not its derivative. See App. B.

Using (6), this means that the KME can be computed as $\sum_{|p|\leq m}\int \partial^{(0,p)}k(.,x)\,\mathrm{d}\mu_p(x)$, which gives a way to numerically compute the KME of distributions. As most distributions encountered in practice happen to be defined as measures or derivatives of some measures, this method is highly relevant in practice.

By combining Propositions 13 and 14, we get the following corollary.

Corollary 16 *Let $k \in \mathscr{C}^{(m,m)}$, $p \in \mathbb{N}^d$ with $|p| \leq m$, and let D, T be two distributions such that $\partial^p D$ and $\partial^p T$ embed into $\mathcal{H}_k$. Then*

$$\langle \partial^p D\,,\, \partial^p T\rangle_k = \langle D\,,\, T\rangle_{\partial^{(p,p)}k} \quad \textit{and} \quad \|\partial^p D\|_k = \|D\|_{\partial^{(p,p)}k}\ .$$

Proof The proof reduces to the following equations.

$$\begin{aligned}
\langle \partial^p D\,,\, \partial^p T\rangle_k &\overset{(a)}{=} \left\langle \int \partial^{(0,p)}k(.,x)\,\mathrm{d}D(x)\,,\, \int \partial^{(0,p)}k(.,y)\,\mathrm{d}T(y)\right\rangle_k \\
&\overset{(b)}{=} \int \left\langle \partial^{(0,p)}k(.,y)\,,\, \partial^{(0,p)}k(.,x)\right\rangle_k \mathrm{d}D(y)\,\mathrm{d}\bar{T}(x) \\
&\overset{(c)}{=} \int \partial^{(p,p)}k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x) \\
&\overset{(d)}{=} \langle D\,,\, T\rangle_{\partial^{(p,p)}k}\ ,
\end{aligned}$$

Equality (a) uses Proposition 14, (b) uses twice (on the left and on the right of the inner product) the definition of the weak integral (1), (c) uses Equation (9) proven in Appendix A which states that $\left\langle \partial^{(0,p)}k(.,y)\,,\, \partial^{(0,p)}k(.,x)\right\rangle_k = \partial^{(p,p)}k(x,y)$, and (d) uses (5) applied to the kernel $\partial^{(p,p)}k$. ■

Corollary 16 tells us that if we use $\partial^{(p,p)}k$—which is a kernel—to compute the MMD between two probability distributions D, T, then we are actually computing the MMD distance between their derivatives $\partial^p D$ and $\partial^p T$ with the kernel k. One could extend this corollary from (p,p) to (p,q) with $|q| \leq m$, yielding $\langle \partial^p D\,,\, \partial^q T\rangle_k = \int \partial^{(q,p)}k(x,y)\,\mathrm{d}D(y)\,\mathrm{d}\bar{T}(x)$. But in that case, $\partial^{(q,p)}k$ might not be a kernel anymore.

5.2. c^m- and c_0^m-Universal Kernels

Theorem 6 shows the equivalence between c_*^m-universality and characteristicness over $\mathscr{D}_{L^1}^m$ or $\mathscr{E}^m$. But neither the universality, nor the characteristic assumption seems easy to check in general. However, for translation-invariant kernels, meaning kernels that can be written as $k(x,y) = \psi(x-y)$ for some function ψ, we will now show that being characteristic to $\mathscr{P}$ or to $\mathscr{D}_{L^1}^m$ is one and the same thing, provided that $k \in \mathscr{C}_b^{(m,m)}$. Thus, any technique to prove that a kernel is characteristic may also be used to prove that it is characteristic to the much wider space $\mathscr{D}_{L^1}^m$. One of these techniques consists in verifying that the distributional Fourier transform $\mathcal{F}\psi$ has full support. The reader unfamiliar with distributional Fourier transforms may think of them as an extension of the usual Fourier transform—which is usually only defined on L^1, L^2 or $\mathcal{M}_f$—to wider function and distribution spaces. Let us mention that $\mathcal{F}\psi$ is exactly the unique positive, symmetric, finite measure appearing in Bochner's theorem (Wendland, 2004, Thm.6.6), and whose (usual) Fourier transform is ψ. We now successively present the result for $\mathscr{D}_{L^1}^m$, then for $\mathscr{E}^m$.

Theorem 17 *Let $k \in \mathscr{C}^{(m,m)}$ be a translation-invariant kernel $k(x,y) = \psi(x-y)$ with $\mathcal{X} = \mathbb{R}^d$, and $\mathcal{F}\psi$ its distributional Fourier transform. Then $\mathscr{D}_{L^1}^m$ embeds into $\mathcal{H}_k$ and the following are equivalent.*

(i) k is characteristic (to $\mathscr{P}$).
(ii) k is characteristic to $\mathscr{D}_{L^1}^m$.
(iii) $\mathcal{F}\psi$ has full support.

If moreover $\psi \in \mathscr{C}_0^m$, then k is c_0^m-universal iff it is c_0-universal.

Theorem 18 *Let $k \in \mathscr{C}^{(m,m)}$ be a translation-invariant kernel $k(x,y) = \psi(x-y)$ with $\mathcal{X} = \mathbb{R}^d$. If the support of $\mathcal{F}\psi$ has Lebesgue-measure > 0, then k is characteristic to $\mathscr{E}^m$.*

Proof [of Theorem 17] First, note that $\partial^{(p,p)}k(x,y) \leq \partial^{(p,p)}k(x,x)\partial^{(p,p)}k(y,y) = (\partial^{2p}\psi(0))^2$ for any $|p| \leq m$ (see Lemma 19 in Appendix A). Hence $k \in \mathscr{C}_b^{(m,m)}$, which, by Corollary 4, proves that $\mathscr{D}_{L^1}^m$ embeds into $\mathcal{H}_k$. Now suppose that (i) and (ii) are equivalent, then they are also equivalent to k being characteristic to $\mathcal{M}_f$. Using Theorem 6, we thus proved the last sentence. Now, (ii) clearly implies (i) and Theorem 9 of Sriperumbudur et al. (2010b) states that (i) and (iii) are equivalent. So it remains to show that (iii) implies (ii). We now sketch its proof and relegate the details to Appendix A.5. Let Λ be the finite positive measure from Bochner's theorem, such that $\psi = \mathcal{F}\Lambda$ and let $D \in \mathscr{D}_{L^1}^m$. Then

$$\begin{aligned}
\|D\|_k^2 &= \iint \left(\int e^{i(x-y)\cdot\xi}\,\mathrm{d}\Lambda(\xi)\right)\mathrm{d}\bar{D}(x)\,\mathrm{d}D(y) \\
&\overset{(a)}{=} \int \left(\iint (e^{i(x-y)\cdot\xi})\,\mathrm{d}\bar{D}(x)\,\mathrm{d}D(y)\right)\mathrm{d}\Lambda(\xi) \\
&\overset{(b)}{=} \int |[\mathcal{F}D](\xi)|^2\,\mathrm{d}\Lambda(\xi)\,,
\end{aligned}$$

where $\cdot$ denotes the Euclidian inner-product on $\mathbb{R}^d$. Λ being positive, if it has full support, then $[\mathcal{F}D](\xi) = 0$ for almost all $\xi \in \mathcal{X}$. Thus $D = 0$. Assuming that (a) and (b) indeed hold, we just showed that if (iii), then $\|D\|_k = 0$ implies $D = 0$, meaning that k is s.p.d. to $\mathscr{D}_{L^1}^m$, which, with Theorem 6, proves (ii). We relegate the proof of (a) and (b) to Appendix A.5. ■

Proof [of Theorem 18] For any $D \in \mathscr{E}^m$, we can write, like before: $\|D\|_k^2 = \int |[\mathcal{F}D](\xi)|^2\,\mathrm{d}\Lambda(\xi)$. But now, the Paley-Wiener-Schwartz theorem (Treves, 1967, Thm. 29.2) states that $\mathcal{F}D$ is an analytical function, so if its set of zeros has Lebesgue-measure > 0, then $\mathcal{F}D$ is the 0 function, so $D = 0$, showing that Φ_k is injective over $\mathscr{E}^m$. ■

These theorems show for example that Gaussian kernels are c_0^∞-universal and that the sinc kernel, defined on $\mathcal{X} = \mathbb{R}$ by $k(x,y) = \sin(x-y)/(x-y)$ (and 1 on the diagonal), is c^∞- but not c_0^∞-universal. When $\mathcal{X} = \mathbb{R}$, one can refine the conditions on the Fourier transform in Theorem 18 so that they become necessary and sufficient (Simon-Gabriel and Schölkopf, 2016, Theorem 41).

6. Conclusion

We first discuss how this work relates and contributes to the existing machine learning literature and then conclude.

6.1. Related Machine Learning Literature

Universal and characteristic kernels play an essential role in kernel methods and their theory. Universal kernels ensure consistency of many RKHS-based estimators in the context of regression and classification (Steinwart, 2001; Steinwart and Christmann, 2008), whereas characteristic kernels are of prime interest in any MMD-based algorithm, such as kernel two-sample tests (Gretton et al., 2007, 2012), HSIC independence tests (Gretton et al., 2008; Gretton and Györfi, 2010; Fukumizu et al., 2008), kernel density estimators (Sriperumbudur, 2016) and MMD-type GANs (Li et al., 2015; Dziugaite et al., 2015). The machine learning community gradually introduced more and more variants of universal kernels (Steinwart, 2001; Micchelli et al., 2006; Carmeli et al., 2006; Caponnetto et al., 2008), but instead of also introducing variants of characteristic kernels, it stuck to the original definition given by Fukumizu et al. (2004) which considered only characteristicness to $\mathscr{P}$. As a result, the literature started proving various links between the various variants of universal kernels and the only notion of characteristic kernels that it had. Eventually these notions were linked to $\int$s.p.d. and conditionally $\int$s.p.d. kernels (Fukumizu et al., 2004, 2008, 2009b,a; Gretton et al., 2007; Sriperumbudur et al., 2008, 2010a,b) and all known relations got summarized in a superb overview article by Sriperumbudur et al. (2011). However, by not introducing the notion of a characteristic kernel to something else than $\mathscr{P}$, the literature oversaw the fundamental dual link between universal, characteristic and s.p.d. kernels shown in Theorem 6 of this paper, which easily explains all the previously reported links.

Concerning the study of kernels that metrize the weak convergence of probability measures, in mathematics it dates back at least to Guilbart (1978), but it got introduced into the machine learning community only many years later by Sriperumbudur et al. (2010b). They gave new sufficient conditions to metrize the weak convergence, which then got improved by Sriperumbudur (2016)[Thm. 2]. However, by generalizing these sufficient conditions even further, Theorem 12 of this work is the first to provide conditions that are both sufficient *and necessary*, and that holds on any locally compact Hausdorff space $\mathcal{X}$ (which is more general than in the existing literature).

6.2. Future Work and Closing Remarks

This paper grouped various notions of universal, characteristic and s.p.d. kernels into three fundamental definitions—one for each—and showed that they are essentially equivalent: they describe the same family of kernels, but from dual perspectives. Using this duality link, we could systematically recover most of the previously known links, but also discovered new ones, such as the equivalence between characteristicness to $\mathscr{P}$ and universality over $(\mathscr{C}_b)_c/\mathbb{1}$; or between strict positive definiteness (over $\mathcal{M}_\delta$) and universality over $\mathbb{C}^{\mathcal{X}}$. We then compared the convergence in MMD with other convergence types of distributions and measures. Importantly, we showed that a bounded kernel metrizes the weak convergence of probability measures iff it is continuous and characteristic. Incidentally, we also showed that

KMEs over probability measures can be extended to generalized measures called Schwartz-distributions. For translation-invariant kernels, this extension preserves characteristicness, in the sense that a characteristic kernel to $\mathscr{P}$ will also be characteristic to $\mathcal{D}^m_{L^1}$. In all this work, we assumed $\mathcal{X}$ to be locally compact. Although this assumption fits many very general spaces, unfortunately, it does not contain any infinite-dimensional Banach space. So a main open question of this paper is whether our characterization of kernels that metrize the weak convergence of probability measures also applies to more general spaces, such as so-called Polish spaces, which are very standard spaces in probability theory. Finally, we also proved a few results that are specific to KMEs of distributions. Proposition 14 and its Corollary 16 on the embedding of derivatives for example show that these KMEs of distributions naturally appear when considering KMEs w.r.t. derivatives of kernels. We hope that they will in future lead to new insights and applications in machine learning.

Acknowledgments

We are deeply indebted to Alphabet for their support via a Google European Fellowship in Causal Inference accorded to Carl-Johann Simon-Gabriel. We thank Bharath Sriperumbudur for enlightening remarks, Jonas Peters for his kind and helpful advice, Ilya Tolstikhin for useful discussions and encouragements, and the two anonymous reviewers for their extensive, useful and contructive feedback.

Appendix A. Proofs

In this section, we gather all the complements to non fully proved theorems, propositions, corollaries or lemmas appearing in the main text. We start with a lemma that essentially follows from Corollary 4.36 of Steinwart and Christmann (2008), and which we will need a few times for the proofs.

Lemma 19 *Let* $k \in \mathscr{C}_b^{(m,m)}$ *and let* $\Phi : \begin{array}{ccc} \mathcal{X} & \longrightarrow & \mathcal{H}_k \\ x & \longmapsto & k(.,x) \end{array}$ *. Then for any* $p \in \mathbb{N}^d$ *with* $|p| \leq m$*, the partial derivative* $\partial^p\Phi$ *exists, belongs to* $\mathcal{H}_k$*, is continuous and verifies* $\partial^p\Phi(x) = \partial^{(0,p)}k(.,x)$*. Moreover, for any* $f \in \mathcal{H}_k$*,* $\partial^p f$ *exists, belongs to* $\mathcal{H}_k$ *and verifies:*

$$\partial^p f(x) = \Big\langle f\,,\, \partial^{(0,p)}k(.,x)\Big\rangle_k \ . \tag{8}$$

Applied with $f = \partial^{(0,q)}k(.,y)$ *where* $|q| \leq m$ *also proves that*

$$\partial^{(p,q)}k(x,y) = \Big\langle \partial^{(0,q)}k(.,y)\,,\, \partial^{(0,p)}k(.,x)\Big\rangle_k \ . \tag{9}$$

Proof This Lemma is essentially proven in Corollary 4.36 and in its proof of Steinwart and Christmann (2008). We only added Equation (9), which is a straightforward consequence of (8), and the part stating that $\partial^p\Phi(x) = \partial^{(0,p)}k(.,x)$. This can be shown as follows. Steinwart and Christmann (2008) prove that $\partial^p\Phi$ exists and belongs to $\mathcal{H}_k$. Thus

$$[\partial^p\Phi(x)](y) = \langle \partial^p\Phi(x)\,,\, k(.,y)\rangle_k$$

$$= \left\langle \lim_{h\to 0} (\Phi(x+he_i) - \Phi(x))/h \,,\, k(.,y) \right\rangle_k$$
$$= \lim_{h\to 0} (k(y, x+he_i) - k(y,x))/h$$
$$= \partial^{(0,p)} k(y,x) \;,$$

where we used the continuity of the inner product to swap limit and bracket signs. ■

A.1. Proof of Corollary 3

Proof Suppose that $\mathcal{H}_k \subset \mathcal{C}_0$. (i) clearly holds. Suppose (ii) was not met. Then let $x_n \in \mathcal{X}$ such that $k(x_n, x_n) = \|k(.,x_n)\|_k^2 \to \infty$. Thus $k(.,x_n)$ is unbounded. But $\langle f\,,\, k(.,x_n)\rangle_k = f(x_n)$ is bounded for any $f \in \mathcal{H}_k$, thus $k(.,x_n)$ is bounded (Banach-Steinhaus Theorem). Contradiction. Thus (ii) is met.

Conversely, suppose that (i) and (ii) hold. Let $\mathcal{H}_k^{\text{pre}} := \text{span}\{k(.,x)\,|\,x \in \mathcal{X}\}$. Then, $\mathcal{H}_k^{\text{pre}} \subset \mathcal{C}_0$, and for any $f, g \in \mathcal{H}_k$, $\|f-g\|_\infty \leq \|f-g\|_k \, \|k\|_\infty$. Thus $\mathcal{H}_k^{\text{pre}}$ continuously embeds into the *closed* $\mathcal{C}_0$, thus so does its $\|.\|_k$-closure, $\mathcal{H}_k$. The proof of the cases $\mathcal{H}_k \subset \mathcal{C}$ and $\mathcal{H}_k \subset \mathcal{C}_b$ are similar (see also Berlinet and Thomas-Agnan, 2004, Thm. 17). ■

A.2. Proof of Corollary 4

Proof Suppose that $k \in \mathcal{C}_b^{(m,m)}$. Then $\mathcal{H}_k^{\text{pre}} \subset \mathcal{C}_b^m$ (Steinwart and Christmann, 2008, Corollary 4.36) and for any $x \in \mathcal{X}$, $f \in \mathcal{H}_k^{\text{pre}}$, and $|p| \leq m$, we have $\|\partial^p f\|_\infty \leq \|f\|_k \left\| \sqrt{\partial^{(p,p)} k} \right\|_\infty$. Thus $\mathcal{H}_k^{\text{pre}}$ continuously embeds into the *closed* space $\mathcal{C}_b^m$, thus so does its $\|.\|_k$-closure, $\mathcal{H}_k$. But, by definition of $(\mathcal{C}_b^m)_c$ is the space $\mathcal{C}_b$ equipped with a weaker topology (see Section 3), thus $\mathcal{C}_b^m \hookrightarrow (\mathcal{C}_b^m)_c$. Thus $\mathcal{H}_k \hookrightarrow (\mathcal{C}_b^m)_c$, which concludes. The proofs when $k \in \mathcal{C}$ or $k \in \mathcal{C}_0$ are similar. ■

A.3. Proof of Theorem 8

Proof Equivalence between (i) & (ii). As KMEs are linear over $\mathcal{M}_f$, a kernel k is characteristic to $\mathcal{P}$ iff it is characteristic to $\mathcal{P} - P := \{\mu - P : \mu \in \mathcal{P}\}$, where P can be any fixed probability measure. This is equivalent to being characteristic to the linear span of $\mathcal{P} - P$. But the linear span of $\mathcal{P} - P$ is precisely $\mathcal{M}_f^0$, which concludes.

Equivalence of (ii) & (v): First of all, notice that, if (v), then k and k_0 define the same MMD on $\mathcal{M}_f^0$, because, for any $\mu \in \mathcal{M}_f^0$, $\mu(\mathbb{1}) = 0$, thus:

$$\|\mu\|_{k_0}^2 = \iint \langle \delta_x - \nu_0 \,,\, \delta_y - \nu_0 \rangle_k \, \mathrm{d}\bar{\mu}(x) \, \mathrm{d}\mu(y)$$
$$= \iint k(x,y) \, \mathrm{d}\bar{\mu}(x) \, \mathrm{d}\mu(y) - \int \langle \delta_x \,,\, \nu_0 \rangle_k \, \mathrm{d}\bar{\mu}(x) \int \mathrm{d}\mu(y)$$
$$- \int \mathrm{d}\bar{\mu}(x) \int \langle \nu_0 \,,\, \delta_y \rangle_k \, \mathrm{d}\mu(y) - \|\nu_0\|_k^2 \iint \mathrm{d}\bar{\mu}(x) \, \mathrm{d}\mu(y)$$

$$= \|\mu\|_k^2 \;,$$

Thus k_0 is characteristic to $\mathcal{M}_f^0$ iff k is also. Thus (v) implies (ii). Conversely, if k_0 is characteristic to $\mathcal{M}_f^0$, then k_0 is either characteristic to $\mathcal{M}_f$, in which case choosing $k_0 = k$ and $\nu_0 = 0$ fulfills the requirements of (v); or there exists a non zero measure $\nu_0 \in \mathcal{M}_f$ such that $\Phi_{k_0}(\nu_0) = 0$. As Φ_{k_0} is linear, we can choose $\nu_0(\mathbb{1}) = 1$ without loss of generality. Supposing now that we are in the latter case, the proof proceeds as follows.

(a) Show that the constant function $\mathbb{1} \notin \mathcal{H}_{k_0}$.
(b) Construct a new Hilbert space of functions of the form $\mathcal{H}_k = \text{span}\,\mathbb{1} \oplus \mathcal{H}_{k_0}$.
(c) Show that it has a reproducing kernel k.
(d) Show that k_0 and k fulfill the requirements of (v).

(a) Suppose that $\mathbb{1} \in \mathcal{H}_{k_0}$. Then $1 = \bar{\nu}_0(\mathbb{1}) = \int \langle \mathbb{1}\,,\, k_0(.,x)\rangle_{k_0} \, \mathrm{d}\bar{\nu}_0(x) \overset{(*)}{=} \left\langle \mathbb{1}\,,\, \int k_0(.,x) \, \mathrm{d}\nu_0(x) \right\rangle_{k_0} = \langle \mathbb{1}\,,\, \Phi_{k_0}(\nu_0)\rangle_{k_0} = 0$, where in $(*)$ we use the definition of KMEs (1) . Contradiction. Thus $\mathbb{1} \notin \mathcal{H}_{k_0}$.

(b) Define $\mathcal{H} := \text{span}\,\mathbb{1} \oplus \mathcal{H}_{k_0}$ and equip it with the inner product $\langle .\,,\, .\rangle$ that extends the inner product of $\mathcal{H}_{k_0}$ so, that

$$\mathbb{1} \perp \mathcal{H}_{k_0} \quad \text{and} \quad \|\mathbb{1}\| = 1 \,. \tag{10}$$

In other words, for any $f = c_f\mathbb{1} + f^\perp \in \mathcal{H}$ and any $g = c_g\mathbb{1} + g^\perp \in \mathcal{H}$:

$$\langle f\,,\, g\rangle := \left\langle f^\perp \,,\, g^\perp \right\rangle_{k_0} + c_f \bar{c_g}. \tag{11}$$

Obviously $\mathcal{H}$ is a Hilbert space of functions.

(c) We now construct k by first defining an injective embedding Φ and then showing that $k(x,y) := \langle \Phi(\delta_x)\,,\, \Phi(\delta_y)\rangle$ is a reproducing kernel with KME Φ.
As $\mathcal{M}_f^0$ is a hyperplane in $\mathcal{M}_f$ and $\nu_0 \in \mathcal{M}_f \backslash \mathcal{M}_f^0$, each measure $\mu \in \mathcal{M}_f$ can be decomposed uniquely in a sum: $\mu = \mu^\perp + \mu(\mathbb{1})\nu_0$ where $\mu^\perp = \mu - \mu(\mathbb{1})\nu_0 \in \mathcal{M}_f^0$. We may thus define the following linear embedding $\Phi : \mathcal{M}_f \longrightarrow \mathcal{H}$ by

$$\Phi(\mu) := \begin{cases} \Phi_{k_0}(\mu) & \text{if } \mu \in \mathcal{M}_f^0 \\ \mathbb{1} & \text{if } \mu = \nu_0 \end{cases} \quad \text{i.e.} \quad \begin{array}{rcl} \Phi(\mu) & := & \Phi_{k_0}(\mu^\perp) + \mu(\mathbb{1})\mathbb{1} \\ & = & \Phi_{k_0}(\mu) + \mu(\mathbb{1})\mathbb{1} \end{array} . \tag{12}$$

Noting that $\Phi(\mu)^\perp = \Phi(\mu^\perp) = \Phi_{k_0}(\mu^\perp) = \Phi_{k_0}(\mu)$ and using (11), we get

$$\forall f \in \mathcal{H},\ \forall x \in \mathcal{X},\ \ \langle f\,,\, \Phi(\delta_x)\rangle = \left\langle f^\perp \,,\, \Phi(\delta_x)^\perp \right\rangle_{k_0} + c_f = f^\perp(x) + c_f\mathbb{1}(x) = f(x)\,. \tag{13}$$

So by defining $k(x,y) := \langle \Phi(\delta_y)\,,\, \Phi(\delta_x)\rangle$ and applying (13) to $f = \Phi(\delta_y)$, we see that $\Phi(\delta_y) = k(.,y)$. Thus (13) may be rewritten as

$$\forall f \in \mathcal{H}, \forall x \in \mathcal{X}, \qquad \langle f\,,\, k(.,x)\rangle = f(x).$$

Thus $\mathcal{H}$ is an RKHS with reproducing kernel and Φ is its associated KME.

(d) As k_0 is characteristic to $\mathcal{M}_f^0$, Φ is injective over $\mathcal{M}_f^0$. And $\Phi(\nu_0) \in \mathcal{H}\backslash\Phi(\mathcal{M}_f^0)$. Thus Φ is injective over $\mathcal{M}_f$, so k is characteristic to $\mathcal{M}_f$. To conclude, (12) shows that

$$\begin{aligned}\langle \delta_y - \nu_0\,,\, \delta_x - \nu_0\rangle &= \langle \Phi_{k_0}(\delta_y) + (\delta_y - \nu_0)(\mathbb{1})\mathbb{1}\,,\, \Phi_{k_0}(\delta_x) + (\delta_x - \nu_0)(\mathbb{1})\mathbb{1}\rangle \\ &= \langle \Phi_{k_0}(\delta_y) + 0\,,\, \Phi_{k_0}(\delta_x) + 0\rangle \\ &= k_0(x,y)\ .\end{aligned}$$

Equivalence of (v) with (iii) & (iv): First, notice that the kernel k constructed in the proof of (v) $\Rightarrow$ (ii) verifies:

$$\begin{aligned}k(x,y) &= \langle \Phi(\delta_x)\,,\, \Phi(\delta_y)\rangle \\ &= \langle \Phi_{k_0}(\delta_x) + \delta_x(\mathbb{1})\mathbb{1}\,,\, \Phi_{k_0}(\delta_y) + \delta_y(\mathbb{1})\mathbb{1}\rangle \\ &= \langle \Phi_{k_0}(\delta_x)\,,\, \Phi_{k_0}(\delta_y)\rangle + \|\mathbb{1}\|^2 \\ &= k_0(x,y) + 1\ ,\end{aligned}$$

where we used (10), (12) and the fact that by construction $\langle .\,,\,.\rangle$ coincides with $\langle .\,,\,.\rangle_{k_0}$ on $\mathcal{M}_f^0$. Thus the proof of (v) $\Rightarrow$ (ii) shows that, if k_0 characteristic to $\mathcal{M}_f^0$, then the kernel $k_0(x,y)+1$ is characteristic to $\mathcal{M}_f$, thus $\int$s.p.d. (Thm. 6). $k(x,y) := k_0(x,y)+1$ is $\int$s.p.d.. More generally, if instead of fixing $\|\mathbb{1}\|_k = 1$ in (10) we fixed $\|\mathbb{1}\|_k = \epsilon$ for some real $\epsilon > 0$, then we would have ended up with an $\int$s.p.d. kernel k verifying $k(x,y) := k_0(x,y) + \epsilon^2$. Thus (ii) implies (iii) and (iv). Conversely, given any kernel k of the previous form, the inner products defined by k and k_0 coincide on $\mathcal{M}_f^0$. So if k is characteristic to $\mathcal{M}_f^0$, then so is k_0. Thus (iii) or (iv) implies (ii). ■

A.4. Proof of Theorem 12 Continued

The proof of Theorem 12 used the following lemma.

Lemma 20 *Let k be a continuous, $\int$s.p.d. kernel and let $(\mu_\alpha)_\alpha$ be bounded in $\mathcal{M}_+$ (meaning $\sup_\alpha \|\mu_\alpha\|_{TV} < \infty$). Then $\mu_\alpha \xrightarrow{w-k} \mu \;\Rightarrow\; \mu_\alpha \xrightarrow{\sigma} \mu$. Consequently: $\mu_\alpha \xrightarrow{\|.\|_k} \mu \;\Rightarrow\; \mu_\alpha \xrightarrow{\sigma} \mu$.*

Proof We will show that $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathcal{C}_c$. As $\mathcal{C}_c$ is a dense subset of $\mathcal{C}_0$ and μ_α is bounded, combining Prop. 32.5 and Thm. 33.2 of Treves (1967) then shows that $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathcal{C}_0$ (weak-$*$ convergence), which implies weak-convergence, $\mu_\alpha \xrightarrow{\sigma} \mu$ (Berg et al., 1984, Chap. 2, Cor. 4.3), and thus concludes.

Let K be a compact subset of $\mathcal{X}$. First, we show that there exists a function $h \in \mathcal{H}_k$ such that $h(x) > 0$ for any $x \in K$. To do so, let $f \in \mathcal{C}_b$ such that $f \geq 1$ on K. k being $\int$s.p.d. and $\mathcal{M}_f$ being the dual of $(\mathcal{C}_b)_c$, $\mathcal{H}_k$ is dense in $(\mathcal{C}_b)_c$ (Thm. 6). So we can find a sequence of functions $f_n \in \mathcal{H}_k$ that converges to f for the topology of $(\mathcal{C}_b)_c$. By definition of the topology of $(\mathcal{C}_b)_c$, this implies in particular that the restrictions of f_n to K converge in infinity norm, meaning: $\sup_{x\in K} |f_n(x) - f(x)| \to 0$. Thus, for a sufficiently large n, $f_n > 0$ on K, so we can take $h = f_n$.

Now, let us define the measures $h.\mu_\alpha$ as $[h.\mu_\alpha](f) = \mu_\alpha(hf)$ for any $f \in \mathcal{C}_b$. Then $\|h.\mu_\alpha\|_{TV} \leq \|h\|_\infty \|\mu_\alpha\|_{TV}$, so the new net $(h.\mu_\alpha)_\alpha$ is bounded. But bounded sets are

relatively compact for the weak-$*$ topology $w(\mathcal{M}_f, \mathcal{C}_0)$. (Treves 1967, Thm. 33.2, or Banach-Alaoglu theorem). So we can extract a subnet $h.\mu_\beta$ of $h.\mu_\alpha$ that converges in weak-$*$ topology. Then $h.\mu_\beta$ is also a Cauchy-net for the weak-$*$ topology, meaning that for any $\epsilon > 0$ and any sufficiently large β, β':

$$|\mu_\beta(hf) - \mu_{\beta'}(hf)| \leq \epsilon, \quad \forall f \in \mathcal{C}_0\ .$$

This inequality holds in particular for functions f whose support is contained in K, which we denote $f \in \mathcal{C}_c(K)$. But the mapping $f \longmapsto g := hf$ is a bijective map from $\mathcal{C}_c(K)$ to itself (because $h > 0$ on K), so we actually have $|\mu_\beta(g) - \mu_{\beta'}(g)| \leq \epsilon$ for any $g \in \mathcal{C}_c(K)$. But this holds for any compact subset K of $\mathcal{X}$. So the inequality also holds for any function $g \in \mathcal{C}_c(\mathcal{X})$, which shows that μ_β is a Cauchy-net for the topology of pointwise convergence in $\mathcal{C}_c(\mathcal{X})$, also known as the *vague* topology. But $\mathcal{M}_+$ is vaguely complete (Bourbaki, 2007, Chap.III, §1, n.9, Prop.14), so μ_β converges to a measure $\mu' \in \mathcal{M}_+$. But for any $f \in \mathcal{C}_c(\mathcal{X})$, $\mu'(f) = \lim_\beta \mu_\beta(f) = \lim_\alpha \mu_\alpha(f) = \mu(f)$, thus μ' and μ coincide on $\mathcal{C}_c(\mathcal{X})$, which is a dense subset of $\mathcal{C}_0$. Thus $\mu' = \mu$, and $\mu_\alpha(f) \to \mu(f)$ for any $f \in \mathcal{C}_c$. ■

Note that if we additionally supposed that $\mathcal{H}_k \hookrightarrow \mathcal{C}_0$ (meaning that k is c_0-universal), then Lemma 20 is a simple consequence of Lemma 11 and the fact that weak-$*$ and weak convergence coincide on $\mathscr{P}$.

A.5. Proof of Theorem 17 Continued

Proof We are left with proving (a) and (b). To do so, we will use the decomposition $D = \sum_{|p|\leq m} \partial^p \mu_p$ of Lemma 15. Indeed, k being in $\mathcal{C}_b^{(m,m)}$, by Corollary 4, $\partial^p \mu_p$ embeds into $\mathcal{H}_k$ for any $|p| \leq m$ and $\mu_p \in \mathcal{M}_f$. Thus

$$\begin{aligned}\langle \partial^p\mu_p\,,\, \partial^q\mu_q\rangle_k &= \langle \Phi_{\partial^{(0,p)}k}(\mu_p)\,,\, \Phi_{\partial^{(0,q)}k}(\mu_q)\rangle_k \\ &= \iint \left\langle \partial^{(0,p)}k(.,y)\,,\, \partial^{(0,q)}k(.,x)\right\rangle_k \mathrm{d}\bar{\mu}_q(x)\,\mathrm{d}\mu_p(y) \\ &= \iint \partial^{(q,p)}k(x,y)\,\mathrm{d}\bar{\mu}_q(x)\,\mathrm{d}\mu_p(y) \\ &= \iiint i^{|p+q|}\xi^{p+q}e^{i(x-y)\cdot\xi}\,\mathrm{d}\Lambda(\xi)\,\mathrm{d}\bar{\mu}_q(x)\,\mathrm{d}\mu_p(x)\ ,\end{aligned}$$

where for $\xi = (\xi_1, \dots \xi_d) \in \mathbb{R}^d$, we defined $\xi^p := \xi_1^{p_1}\xi_2^{p_2}\cdots\xi_d^{p_d}$. The first line uses Proposition 14, the second line uses twice the definition of a weak integral (1), the third uses (9) from Lemma 19 and the fourth line uses the fact that $\partial^{(q,p)}k(x,y) = (-1)^{|p|}\partial^{p+q}\psi(x-y)$ and $\mathcal{F}\partial^{p+q}\,\psi = i^{|p+q|}\xi^{p+q}\,\mathcal{F}\psi = i^{|p+q|}\xi^{p+q}\Lambda$.

Let us denote $\xi^p\Lambda$ the measure defined by $\xi^p\Lambda(A) := \int_A \xi^p\,\mathrm{d}\Lambda(\xi)$. We will now show that $\xi^{p+q}\Lambda$ is finite, so that we can apply the usual Bochner theorem and permute the order of integrations. To do so, notice that $\partial^{(p,p)}k(x,y) = (-1)^{|p|}\partial^{2p}\psi(x-y)$ is a continuous kernel, thus, by Bochner's theorem, its associated measure Λ_∂ is finite and verifies $\mathcal{F}\Lambda_\partial = \partial^{2p}\psi$. But the usual calculus rules with Fourier transforms show that $\partial^{2p}\psi = (-i)^{|2p|}\xi^{2p}\Lambda$. Thus $\Lambda_\partial = i^{|p|}\xi^{2p}\Lambda$, showing that $\tilde{\Lambda}$ is a finite measure. Noting now that $2|\xi^{p+q}| \leq \xi^{2p} + \xi^{2q}$, this also implies that $\xi^{p+q}\Lambda$ is a finite measure. Consequently:

$$\langle \partial^p\mu_p\,,\, \partial^q\mu_q\rangle_k = \iiint i^{|p+q|}e^{i(x-y)}\,\mathrm{d}[\xi^{p+q}\Lambda](\xi)\,\mathrm{d}\bar{\mu}_q(x)\,\mathrm{d}\mu_p(x)$$

$$= \iiint i^{|p+q|} e^{i(x-y)} \,\mathrm{d}\bar{\mu}_q(x)\,\mathrm{d}\mu_p(x)\,\mathrm{d}\tilde{\Lambda}(\xi)$$
$$= \int i^{|p+q|} \xi^{p+q}\, \mathscr{F}\mu_q(\xi)\overline{\mathscr{F}\mu_p(\xi)}\,\mathrm{d}\Lambda(\xi)$$
$$= \int [\mathscr{F}(\,\partial^p \mu_p)](\xi)\overline{[\mathscr{F}(\,\partial^q \mu_q)](\xi)}\,\mathrm{d}\Lambda(\xi).$$

Thus, with the decomposition $D = \sum_{|p|\leq m} \partial^p \mu_p$, we get

$$\|D\|_k^2 = \left\| \sum_{|p|\leq m} \partial^p \mu_p \right\|_k^2 = \int \sum_{|p|,|q|\leq m} [\mathscr{F}(\,\partial^p \mu_p)](\xi)\overline{[\mathscr{F}(\,\partial^q \mu_q)](\xi)}\,\mathrm{d}\Lambda(\xi)$$
$$= \int \Big| \sum_{|p|\leq m} [\mathscr{F}(\,\partial^p \mu_p)](\xi) \Big|^2 \mathrm{d}\Lambda(\xi)$$
$$= \int |\,\mathscr{F}D(\xi)|^2\,\mathrm{d}\Lambda(\xi)\ ,$$

where we used the linearity of the Fourier operator on the last line. ■

Appendix B. Short Introduction to Schwartz-Distributions

To introduce Schwartz-distributions, the first step is to notice that any continuous function f is uniquely characterized by the values taken by $f(\varphi) := \int \varphi(x) f(x)\,\mathrm{d}x$ when φ goes through $\mathscr{C}_c$. Rather than seeing f as a function that acts on points x in $\mathcal{X}$, we could thus equivalently see f as a linear functional that acts on other functions φ in $\mathscr{C}_c$ and takes its values in $\mathbb{C}$. Such functionals are called *linear forms*. We could do the same for measures: a signed measure μ is also characterized by the values of $\mu(\varphi) := \int \varphi(x)\,\mathrm{d}\mu(x)$. So we could also see it as a linear functional that acts on functions φ in $\mathscr{C}_c$. Doing so effectively identifies f with the signed measure μ_f that has density f, because both define the same linear form $\varphi \longmapsto \int \varphi(x) f(x)\,\mathrm{d}x$. So from this perspective, a function f becomes a particular kind of measure, and a measure μ a sort of 'generalized function'. Moreover, seen as linear forms over $\mathscr{C}_c$, f and μ are continuous in the sense that if φ_α converges to φ, then $\mu(\varphi_\alpha)$ converges to $\mu(\varphi)$. Thus, by definition, we just identified f and μ with elements of the dual of $\mathscr{C}_c$.

We may now ask whether there are other continuous linear forms over $\mathscr{C}_c$. The answer is negative and is given by the Riesz-Markov-Kakutani representer theorem (see Appendix C). It states that the dual of $\mathscr{C}_c$ is exactly the set of signed regular Borel measures $\mathcal{M}_r$, meaning that any continuous linear form over $\mathscr{C}_c$ can be written as $\varphi \longmapsto \int \varphi\,\mathrm{d}\mu(x)$ for some $\mu \in \mathcal{M}_r$, and can thus be identified with a measure μ. So it seems that our generalization of functions to measures using continuous linear forms is as general as it can get. But this is forgetting the following detail. To distinguish a measure μ from all the others in $\mathcal{M}_r$, we do not need to know the values $\mu(\varphi)$ for *all* functions φ of $\mathscr{C}_c$. Actually, it suffices to know them for all φ in $\mathscr{C}_c^\infty$. This is because $\mathscr{C}_c^\infty$ is a dense subset of $\mathscr{C}_c$. Thus for any $\varphi \in \mathscr{C}_c$, even if $\varphi \notin \mathscr{C}_c^\infty$, we can reconstruct the value $\mu(\varphi)$ by taking a sequence φ_α in $\mathscr{C}_c^\infty$ that converges to φ and noticing that, by continuity, $\mu(\varphi)$ is the limit of $\mu(\varphi_\alpha)$. So instead of

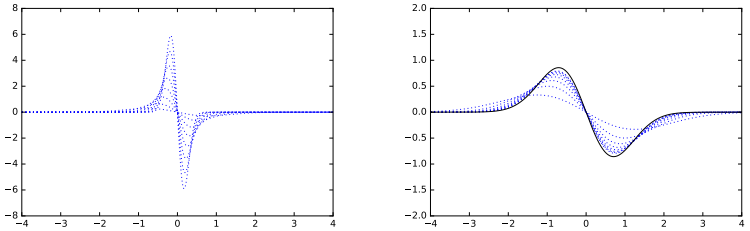


Figure 2: Left: the difference f_σ of two Gaussians that get closer and closer and more and more peaked with decreasing σ. Right: the KMEs of f_σ. Note the difference in the y-axis scale. f_σ converges to a *dipole*, which is not a measure, but a Schwartz-distribution. It cannot be represented as a function, but its KME can (black solid line). Note that the KMEs of f_σ seem to converge to the KME of the dipole.

seeing a function or a measure as an element of $(\mathscr{C}_c)'$, we could also see it as an element of $(\mathscr{C}_c^\infty)'$.

But do we gain anything from it? Yes indeed, because now, we can define linear functionals over $\mathscr{C}_c^\infty$ that we could not define over $\mathscr{C}_c$. For example, suppose that $\mathcal{X} = \mathbb{R}$ and consider the linear form d_x that, to each function φ associates its derivative $\partial\varphi(x)$ evaluated at x. This is a valid (continuous) linear form over $\mathscr{C}_c^\infty$—called a dipole in x—but it cannot be defined over $\mathscr{C}_c$, because not all continuous functions are differentiable. This example shows that, although each measure in $(\mathscr{C}_c)'$ can be seen as an element of $(\mathscr{C}_c^\infty)'$, the latter space contains many more linear forms which do not correspond to a signed measure. This bigger set of linear forms, which we denote $\mathcal{D}^\infty$, is called the set of Schwartz-distributions.

Now, why are distributions useful? First of all, because they can all be seen as limits of functions (Schwartz, 1978)[Theo. XV, Chap. III]. As an example, consider the sequence of functions

$$f_\sigma : \quad x \longmapsto \tfrac{1}{\sigma} g(\tfrac{x+\sigma}{\sigma}) - \tfrac{1}{\sigma} g(\tfrac{x-\sigma}{\sigma})\ ,$$

where g is a Gaussian (see Figure 2). f_σ is the difference of two Gaussians that get closer and closer and more and more peaked with decreasing σ. Now, applying f_σ to a function $\varphi \in \mathscr{C}_c^\infty$, it is not difficult to see that $f_\sigma(\varphi)$ converges to $\partial\varphi(0) = d_0(\varphi)$ when $\sigma \to 0$. The dipole d_0 can thus be seen as a weak limit of the functions f_σ, although it is itself neither a function nor even a signed measure.

Another reason to use distributions is that many common linear operations can be extended to them (or to big subsets of them), such as differentiation, Fourier transformation and convolution. Let us show for example how to extend differentiation. If we want the distributional derivative ∂ to be an extension of the usual derivative, then of course we should require that $\partial\mu_f = \mu_{f'}$ whenever f is a continuously differentiable function over

$\mathcal{X} = \mathbb{R}$ whose usual derivative is f'. Now, by integration by part, we get, for any $\varphi \in \mathscr{C}_c^\infty$:

$$\mu_{f'}(\varphi) = \int f'\varphi = -\int f\varphi' = -\mu_f(\varphi') \ .$$

This suggests to define the derivative of any $D \in \mathscr{D}^\infty$ as $\partial^p D(\varphi) := (-1)^{|p|} D(\partial^p \varphi)$ for any $\varphi \in \mathscr{C}_c^\infty$. Doing so, we just defined a notion of differentiation that is compatible with the usual differentiation and makes *any* distribution infinitely many times differentiable. In particular, any function and any measure is infinitely differentiable in this distributional sense. Moreover, if a sequence of differentiable functions f_n converges to a distribution D (in the sense that $f_n(\varphi)$ converges to $D(\varphi)$ for any φ), then their usual derivatives f_n' converges to ∂D (in the same *distributional* sense). All this makes distributions extremely useful for solving linear differential equations and more generally for physicists. Last but not least, note that, by construction, if Q is a probability measure with smooth density q, then $\partial^p Q$ is the signed measure with density $\partial^p q$.

Appendix C. Other Background Material

Formally, a topological vector space (TVS) $\mathcal{E}$ is a vector space equipped with a topology that is *compatible* with its linear structure, in the sense that the addition $\mathcal{E} \times \mathcal{E} \longrightarrow \mathcal{E}$ and scalar multiplication $\mathbb{C} \times \mathcal{E} \longrightarrow \mathcal{E}$ become continuous for this topology (when their domains are equipped with the product topology). This makes the topology translation-invariant and hence completely defined by the neighborhoods of the origin. A TVS is locally convex (loc. cv.) if there exists a basis of (origin-) neighborhoods consisting of convex sets only. Obviously, the origin-centered balls of any semi-norm are convex. But interestingly, one can show that a TVS is loc. cv. iff its topology can be defined by a family of (continuous) semi-norms. So we can think of loc. cv. TVSs as "multi-normed" spaces, i.e. where convergence is given by a family of possibly multiple semi-norms $(\|.\|_\alpha)_{\alpha \in \mathcal{I}}$ (where the index set $\mathcal{I}$ can be uncountable). If this family contains only a single norm, $\mathcal{E}$ is a normed space. The origin-centered balls of these semi-norms are actually not only convex, they are *barrels*.

Definition 21 (Barrel) *A subset T of a TVS $\mathcal{E}$ is called a* barrel *if it is*

(i) absorbing*: for any $f \in \mathcal{E}$, there exists $c_f > 0$ such that $f \in c_f T$;*
(ii) balanced*: for any $f \in \mathcal{E}$, if $f \in T$ then $\lambda f \in T$ for any $\lambda \in \mathbb{C}$ with $|\lambda| \leq 1$;*
(iii) convex *;*
(iv) closed.

Given that the topology of loc. cv. TVS can be defined by a family of semi-norms, it is not surprising that in loc. cv. spaces there always exists a basis of origin-neighborhoods consisting only of barrels. However, there might be barrels that are not a neighborhood of 0. This leads to

Definition 22 (Barreled spaces) *A TVS is* barreled *if any barrel is a neighborhood of the origin.*

Although many authors include local convexity in the definition, in general, a barreled space need not be loc. cv. Barreled spaces were introduced by Bourbaki, because they were well-suited for the following generalization of the celebrated *Banach-Steinhaus* theorem.

Theorem 23 (Banach-Steinhaus) *Let $\mathcal{E}$ be a barreled TVS, $\mathcal{F}$ be a loc. cv. TVS, and let $L(\mathcal{E},\mathcal{F})$ be the set of continuous linear maps form $\mathcal{E}$ to $\mathcal{F}$. For any $H \subset L(\mathcal{E},\mathcal{F})$ the following properties are equivalent:*

(i) H is equicontinuous.
(ii) H is bounded for the topology of pointwise convergence.
(iii) H is bounded for the topology of bounded convergence.

When $\mathcal{E}$ is a normed space and $\mathcal{F} = \mathbb{C}$, then $L(\mathcal{E},\mathcal{F})$ is by definition $\mathcal{E}'$. With $\|.\|_{\mathcal{E}'}$ being the dual norm in $\mathcal{E}'$, the equivalence of (ii) and (iii) states that

$$\left(\forall f \in \mathcal{E},\ \sup_{h \in H} |h(f)| < \infty\right) \iff \sup_{h \in H} \|h\|_{\mathcal{E}'} < \infty \, .$$

Obviously, to understand the content of the Banach-Steinhaus theorem, one needs the definition of a bounded set. Let us define them now.

When $\mathcal{E}$ is a normed space, then a subset B of $\mathcal{E}$ is called *bounded* if $\sup_{f \in B} \|f\|_{\mathcal{E}} < \infty$. In a more general loc. cv. TVS $\mathcal{E}$, where the topology is given by a family of semi-norms $(\|.\|_\alpha)_{\alpha \in \mathcal{I}}$, a subset B of $\mathcal{E}$ is called *bounded* if, for any $\alpha \in \mathcal{I}$, $\sup_{f \in B} \|f\|_\alpha < \infty$. This can be shown equivalent to the following, more usual definition.

Definition 24 (Bounded Sets in a TVS) *A subset B of a TVS $\mathcal{E}$ is* bounded*, if, for any neighborhood $U \subset \mathcal{E}$ of the origin, there exists a real $c_B > 0$ such that $B \subset c_B U$.*

Note that the notion of boundedness depends on the underlying topology. By default, a bounded set of some dual space $\mathcal{E} = \mathcal{F}'$ designates a set that is bounded for the strong dual topology. We now move on to an unrelated topic: the Riesz Representation theorem for Hilbert spaces. Most of this paper relies on this one theorem.

Theorem 25 (Riesz Representation Theorem for Hilbert Spaces) *A Hilbert space $\mathcal{H}$ and its topological dual $\mathcal{H}'$ are isometrically (anti-) isomorphic via the Riesz representer map*

$$\imath : \begin{array}{ccc} \mathcal{H} & \longrightarrow & \mathcal{H}' \\ f & \longmapsto & D_f := \left\{ \begin{array}{ccc} \mathcal{H} & \longrightarrow & \mathbb{C} \\ g & \longmapsto & \langle g \, , \, f \rangle \end{array} \right. \end{array} \ .$$

In particular, for any continuous linear form $D \in \mathcal{H}'$, there exists a unique element $f \in \mathcal{H}$, called the Riesz representer of D*, such that*

$$\forall g \in \mathcal{H}, \qquad D(g) = \langle g \, , \, f \rangle \ .$$

Note that "anti" in "anti-isomorphic" simply means that, instead of being linear, $\imath$ is anti-linear: for any $\lambda \in \mathbb{C}$ and $f \in \mathcal{H}$, $\imath(\lambda f) = \bar{\lambda}\, \imath(f)$. Often, we prefer to say that $\mathcal{H}$ is isometrically isomorphic to $\overline{\mathcal{H}}'$, where $\overline{\mathcal{H}}'$ denotes the conjugate of $\mathcal{H}$, where the scalar

multiplication is replaced by $(\lambda, f) \longmapsto \bar{\lambda} f$. $\mathcal{H}_k'$ and $\overline{\mathcal{H}_k}'$ are obviously isomorphic via the complex conjugation map $D \longmapsto \bar{D}$.

The Riesz representation theorem for Hilbert spaces is not to be confounded with the following theorem, also known as the Riesz—or Riesz-Markov-Kakutani—representation theorem. In this paper, we always refer to the latter as the Riesz-Markov-Kakutani representation theorem. This theorem has numerous variants, depending on which dual pair $(\mathcal{E}, \mathcal{E}')$ one uses. Here we state it for $\mathcal{E} = \mathscr{C}_0$.

Theorem 26 (Riesz-Markov-Kakutani) *Let $\mathcal{X}$ be a locally compact Hausdorff space. The spaces $\mathcal{M}_f(\mathcal{X})$ and $(\mathscr{C}_0(\mathcal{X}))'$ are isomorphic, both algebraically and topologically via the map*

$$\imath : \begin{array}{ccc} \mathcal{M}_f(\mathcal{X}) & \longrightarrow & (\mathscr{C}_0(\mathcal{X}))' \\ \mu & \longmapsto & D_\mu := \left\{ \begin{array}{ccc} \mathscr{C}_0 & \longrightarrow & \mathbb{C} \\ \varphi & \longmapsto & \int \varphi \, \mathrm{d}\mu \end{array} \right. \end{array} .$$

In other words, for any continuous linear form D over $\mathscr{C}_0(\mathcal{X})$, there exists a unique finite Borel measure $\mu \in \mathcal{M}_f$ such that, for any test function $\varphi \in \mathscr{C}_0(\mathcal{X})$, $D(\varphi) = \int \varphi \, \mathrm{d}\mu$. Moreover, $\sup_{\|\varphi\|_\infty \leq 1} D(\varphi) = |\mu|(\mathcal{X})$, or in short: $\|D\|_{(\mathscr{C}_0)'} = \|\mu\|_{TV}$, where $\|\mu\|_{TV}$ denotes the total variation norm of μ. This is why, in this paper, we identify $\mathcal{M}_f$—a space of σ-additive set functions—with $\mathcal{M}_f$—a space of linear functionals.

In this paper, to embed a space of measures into an RKHS $\mathcal{H}_k$ we successively apply both Riesz representation theorems: If $\mathcal{H}_k$ embeds continuously into $\mathscr{C}_0$, then $(\mathscr{C}_0)'$ embeds continuously into $\overline{\mathcal{H}_k}'$, via the embedding map Φ_k. But $(\mathscr{C}_0)' = \mathcal{M}_f$ (Riesz-Markov-Kakutani Representation) and $\overline{\mathcal{H}_k}' = \mathcal{H}_k$ (Riesz Representation). Thus Φ_k may also be seen as an embedding of $\mathcal{M}_f$ into $\mathcal{H}_k$.

For a further introduction to TVSs and the theorems mentioned here, we suggest Treves (1967).

References

C. Berg, J. P. R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups Theory of Positive Definite and Related Functions*. Springer, 1984.

A. Berlinet and C. Thomas-Agnan. *Reproducing Kernel Hilbert Spaces in Probability and Statistics*. Springer, 2004.

N. Bourbaki. *Intégration - Chapitres 1-4*. Springer, reprint of the 1965 original edition, 2007.

A. Caponnetto, C. Micchelli, M. Pontil, and Y. Ying. Universal multi-task kernels. *Journal of Machine Learning Research*, 9(7):1615–1646, 2008.

C. Carmeli, E. De Vito, and A. Toigo. Vector valued reproducing kernel Hilbert spaces of integrable functions and Mercer theorem. *Analysis and Applications*, 4(4):377–408, 2006.

M. Duc-Jacquet. *Approximation des Fonctionnelles Linéaires sur les Espaces Hilbertiens Autoreproduisants*. PhD thesis, Université Joseph-Fourier - Grenoble I, 1973.

G.K. Dziugaite, D. M. Roy, and Z. Ghahramani. Training generative neural networks via maximum mean discrepancy optimization. In *Uncertainty in Artificial Intelligence*, 2015.

D. H. Fremlin, D. J. H. Garling, and R. G. Haydon. Bounded measures on topological spaces. *Proceedings of the London Mathematical Society*, s3-25(1):115–136, 1972.

K. Fukumizu, F. Bach, and M. Jordan. Kernel dimensionality reduction for supervised learning. *Journal of Machine Learning Research*, 5(12):73–99, 2004.

K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Neural Information Processing Systems*, 2008.

K. Fukumizu, A. Gretton, G. R. Lanckriet, B. Schölkopf, and B. K. Sriperumbudur. Kernel choice and classifiability for RKHS embeddings of probability distributions. In *Neural Information Processing Systems*, 2009a.

K. Fukumizu, A. Gretton, B. Schölkopf, and B. Sriperumbudur. Characteristic kernels on groups and semigroups. In *Neural Information Processing Systems*, 2009b.

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

A. Gretton, O. Bousquet, A. J. Smola, and B. Schölkopf. Measuring statistical dependence with Hilbert-Schmidt norms. In *Algorithmic Learning Theory*, 2005.

A. Gretton, K. M. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Neural Information Processing Systems*, 2007.

A. Gretton, K. Fukumizu, C. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Neural Information Processing Systems*, 2008.

A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. Smola. A kernel two-sample test. *Journal of Machine Learning Research*, 13:723–773, 2012.

C. Guilbart. *Etude des Produits Scalaires sur l'Espace des Mesures: Estimation par Projections*. PhD thesis, Université des Sciences et Techniques de Lille, 1978.

O. Lehtö. Some remarks on the kernel function in Hilbert function space. *Annales Academiae Scientiarum Fennicae*, 109:6, 1952.

Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *International Conference on Machine Learning*, 2015.

D. Lopez-Paz, P. Hennig, and B. Schölkopf. The randomized dependence coefficient. In *Neural Information Processing Systems*, 2013.

C. A. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, 7(12):2651–2667, 2006.

Š. Schwabik. *Topics in Banach Space Integration*. Number 10 in Series in Real Analysis. World Scientific, 2005.

L. Schwartz. Espaces de fonctions différentiables à valeurs vectorielles. *Journal d'Analyse Mathématique*, 4(1):88–148, 1954.

L. Schwartz. *Théorie des Distributions*. Hermann, 1978.

C.-J. Simon-Gabriel and B. Schölkopf. Kernel distribution embeddings: Universal kernels, characteristic kernels and kernel metrics on distributions – arXiv version. arXiv: 1604.05251, 2016.

B. Sriperumbudur. On the optimal estimation of probability measures in weak and strong topologies. *Bernoulli*, 22(3):1839–1893, 2016.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, G. R. G. Lanckriet, and B. Schölkopf. Injective Hilbert space embeddings of probability measures. In *Conference On Learning Theory*, 2008.

B. K. Sriperumbudur, K. Fukumizu, and G. Lanckriet. On the relation between universality, characteristic kernels and RKHS embedding of measures. In *International Conference on Artificial Intelligence and Statistics*, 2010a.

B. K. Sriperumbudur, A. Gretton, K. Fukumizu, B. Schölkopf, and G. R.G. Lanckriet. Hilbert space embeddings and metrics on probability measures. *Journal of Machine Learning Research*, 11:1517–1561, 2010b.

B. K. Sriperumbudur, K. Fukumizu, and G. R. G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12: 2389–2410, 2011.

I. Steinwart. On the influence of the kernel on the consistency of support vector machines. *Journal of Machine Learning Research*, 2(12):67–93, 2001.

I. Steinwart and A. Christmann. *Support Vector Machines*. Information Science and Statistics. Springer, 2008.

F. Treves. *Topological Vector Spaces, Distributions and Kernels*. Academic Press, 1967.

H. Wendland. *Scattered Data Approximation*. Cambridge University Press, 2004.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 9/16; Revised 7/18; Published 9/18

Random Forests, Decision Trees, and Categorical Predictors: The "Absent Levels" Problem

Timothy C. Au TIMAU@GOOGLE.COM
Google LLC
1600 Amphitheatre Parkway
Mountain View, CA 94043, USA

Editor: Sebastian Nowozin

Abstract

One advantage of decision tree based methods like random forests is their ability to natively handle categorical predictors without having to first transform them (e.g., by using feature engineering techniques). However, in this paper, we show how this capability can lead to an inherent "absent levels" problem for decision tree based methods that has never been thoroughly discussed, and whose consequences have never been carefully explored. This problem occurs whenever there is an indeterminacy over how to handle an observation that has reached a categorical split which was determined when the observation in question's level was absent during training. Although these incidents may appear to be innocuous, by using Leo Breiman and Adele Cutler's random forests `FORTRAN` code and the `randomForest R` package (Liaw and Wiener, 2002) as motivating case studies, we examine how overlooking the absent levels problem can systematically bias a model. Furthermore, by using three real data examples, we illustrate how absent levels can dramatically alter a model's performance in practice, and we empirically demonstrate how some simple heuristics can be used to help mitigate the effects of the absent levels problem until a more robust theoretical solution is found.

Keywords: absent levels, categorical predictors, decision trees, CART, random forests

1. Introduction

Since its introduction in Breiman (2001), random forests have enjoyed much success as one of the most widely used decision tree based methods in machine learning. But despite their popularity and apparent simplicity, random forests have proven to be very difficult to analyze. Indeed, many of the basic mathematical properties of the algorithm are still not completely well understood, and theoretical investigations have often had to rely on either making simplifying assumptions or considering variations of the standard framework in order to make the analysis more tractable—see, for example, Biau et al. (2008), Biau (2012), and Denil et al. (2014).

One advantage of decision tree based methods like random forests is their ability to natively handle categorical predictors without having to first transform them (e.g., by using feature engineering techniques). However, in this paper, we show how this capability can lead to an inherent "absent levels" problem for decision tree based methods that has, to the best of our knowledge, never been thoroughly discussed, and whose consequences have never been carefully explored. This problem occurs whenever there is an indeterminacy over how to handle an observation that has reached a categorical split which was determined when the observation in question's level was absent during training—an issue that can arise in three different ways:

1. The levels are present in the population but, due to sampling variability, are absent in the training set.
2. The levels are present in the training set but, due to bagging, are absent in an individual tree's bootstrapped sample of the training set.
3. The levels are present in an individual tree's training set but, due to a series of earlier node splits, are absent in certain branches of the tree.

These occurrences subsequently result in situations where observations with absent levels are unsure of how to proceed further down the tree—an intrinsic problem for decision tree based methods that has seemingly been overlooked in both the theoretical literature and in much of the software that implements these methods.

Although these incidents may appear to be innocuous, by using Leo Breiman and Adele Cutler's random forests `FORTRAN` code and the `randomForest R` package (Liaw and Wiener, 2002) as motivating case studies,[1] we examine how overlooking the absent levels problem can systematically bias a model. In addition, by using three real data examples, we illustrate how absent levels can dramatically alter a model's performance in practice, and we empirically demonstrate how some simple heuristics can be used to help mitigate their effects.

The rest of this paper is organized as follows. In Section 2, we introduce some notation and provide an overview of the random forests algorithm. Then, in Section 3, we use Breiman and Cutler's random forests `FORTRAN` code and the `randomForest R` package to motivate our investigations into the potential issues that can emerge when the absent levels problem is overlooked. And although a comprehensive theoretical analysis of the absent levels problem is beyond the scope of this paper, in Section 4, we consider some simple heuristics which may be able to help mitigate its effects. Afterwards, in Section 5, we present three real data examples that demonstrate how the treatment of absent levels can significantly influence a model's performance in practice. Finally, we offer some concluding remarks in Section 6.

2. Background

In this section, we introduce some notation and provide an overview of the random forests algorithm. Consequently, the more knowledgeable reader may only need to review Sections 2.1.1 and 2.1.2 which cover how the algorithm's node splits are determined.

2.1 Classification and Regression Trees (CART)

We begin by discussing the Classification and Regression Trees (CART) methodology since the random forests algorithm uses a slightly modified version of CART to construct the

[1] Breiman and Cutler's random forests FORTRAN code is available online at:
`https://www.stat.berkeley.edu/~breiman/RandomForests/`

individual decision trees that are used in its ensemble. For a more complete overview of CART, we refer the reader to Breiman et al. (1984) or Hastie et al. (2009).

Suppose that we have a training set with N independent observations

$$(x_n, y_n), \quad n = 1, 2, \ldots, N,$$

where $x_n = (x_{n1}, x_{n2}, \ldots, x_{nP})$ and y_n denote, respectively, the P-dimensional feature vector and response for observation n. Given this initial training set, CART is a greedy recursive binary partitioning algorithm that repeatedly partitions a larger subset of the training set $\mathcal{N}_\mathcal{M} \subseteq \{1, 2, \ldots, N\}$ (the "mother node") into two smaller subsets $\mathcal{N}_\mathcal{L}$ and $\mathcal{N}_\mathcal{R}$ (the "left" and "right" daughter nodes, respectively). Each iteration of this splitting process, which can be referred to as "growing the tree," is accomplished by determining a decision rule that is characterized by a "splitting variable" $p \in \{1, 2, \ldots, P\}$ and an accompanying "splitting criterion" set $\mathcal{S}_p$ which defines the subset of predictor p's domain that gets sent to the left daughter node $\mathcal{N}_\mathcal{L}$. In particular, any splitting variable and splitting criterion pair $(p, \mathcal{S}_p)$ will partition the mother node $\mathcal{N}_\mathcal{M}$ into the left and right daughter nodes which are defined, respectively, as

$$\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p) = \{n \in \mathcal{N}_\mathcal{M} : x_{np} \in \mathcal{S}_p\} \quad \text{and} \quad \mathcal{N}_\mathcal{R}(p, \mathcal{S}_p) = \left\{n \in \mathcal{N}_\mathcal{M} : x_{np} \in \mathcal{S}'_p\right\}, \tag{1}$$

where $\mathcal{S}'_p$ denotes the complement of the splitting criterion set $\mathcal{S}_p$ with respect to predictor p's domain. A simple model useful for making predictions and inferences is then subsequently fit to the subset of the training data that is in each node.

This recursive binary partitioning procedure is continued until some stopping rule is reached—a tuning parameter that can be controlled, for example, by placing a constraint on the minimum number of training observations that are required in each node. Afterwards, to help guard against overfitting, the tree can then be "pruned"—although we will not discuss this further as pruning has not traditionally been done in the trees that are grown in random forests (Breiman, 2001). Predictions and inferences can then be made on an observation by first sending it down the tree according to the tree's set of decision rules, and then by considering the model that was fit in the furthest node of the tree that the observation is able to reach.

The CART algorithm will grow a tree by selecting, from amongst all possible splitting variable and splitting criterion pairs $(p, \mathcal{S}_p)$, the optimal pair $\left(p^*, \mathcal{S}^*_{p^*}\right)$ which minimizes some measure of "node impurity" in the resulting left and right daughter nodes as defined in (1). However, the specific node impurity measure that is being minimized will depend on whether the tree is being used for regression or classification.

In a regression tree, the responses in a node $\mathcal{N}$ are modeled using a constant which, under a squared error loss, is estimated by the mean of the training responses that are in the node—a quantity which we denote as:

$$\hat{c}(\mathcal{N}) = \text{ave}(y_n \mid n \in \mathcal{N}). \tag{2}$$

Therefore, the CART algorithm will grow a regression tree by partitioning a mother node $\mathcal{N}_\mathcal{M}$ on the splitting variable and splitting criterion pair $\left(p^*, \mathcal{S}^*_{p^*}\right)$ which minimizes the squared error resulting from the two daughter nodes that are created with respect to a

$(p, \mathcal{S}_p)$ pair:

$$\left(p^*, \mathcal{S}^*_{p^*}\right) = \underset{(p, \mathcal{S}_p)}{\arg\min} \left(\sum_{n \in \mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)} [y_n - \hat{c}(\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p))]^2 + \sum_{n \in \mathcal{N}_\mathcal{R}(p, \mathcal{S}_p)} [y_n - \hat{c}(\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p))]^2 \right), \tag{3}$$

where the nodes $\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)$ and $\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p)$ are as defined in (1).

Meanwhile, in a classification tree where the response is categorical with K possible response classes which are indexed by the set $\mathcal{K} = \{1, 2, \ldots, K\}$, we denote the proportion of training observations that are in a node $\mathcal{N}$ belonging to each response class k as:

$$\hat{\pi}_k(\mathcal{N}) = \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} I(y_n = k), \quad k \in \mathcal{K},$$

where $|\cdot|$ is the set cardinality function and $I(\cdot)$ is the indicator function. Node $\mathcal{N}$ will then classify its observations to the majority response class

$$\hat{k}(\mathcal{N}) = \underset{k \in \mathcal{K}}{\arg\max}\ \hat{\pi}_k(\mathcal{N}), \tag{4}$$

with the Gini index

$$G(\mathcal{N}) = \sum_{k=1}^{K} [\hat{\pi}_k(\mathcal{N}) \cdot (1 - \hat{\pi}_k(\mathcal{N}))]$$

providing one popular way of quantifying the node impurity in $\mathcal{N}$. Consequently, the CART algorithm will grow a classification tree by partitioning a mother node $\mathcal{N}_\mathcal{M}$ on the splitting variable and splitting criterion pair $\left(p^*, \mathcal{S}^*_{p^*}\right)$ which minimizes the weighted Gini index resulting from the two daughter nodes that are created with respect to a $(p, \mathcal{S}_p)$ pair:

$$\left(p^*, \mathcal{S}^*_{p^*}\right) = \underset{(p, \mathcal{S}_p)}{\arg\min} \left(\frac{|\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)| \cdot G(\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)) + |\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p)| \cdot G(\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p))}{|\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)| + |\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p)|} \right), \tag{5}$$

where the nodes $\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p)$ and $\mathcal{N}_\mathcal{R}(p, \mathcal{S}_p)$ are as defined in (1).

Therefore, the CART algorithm will grow both regression and classification trees by partitioning a mother node $\mathcal{N}_\mathcal{M}$ on the splitting variable and splitting criterion pair $(p^*, \mathcal{S}^*_{p^*})$ which minimizes the requisite node impurity measure across all possible $(p, \mathcal{S}_p)$ pairs—a task which can be accomplished by first determining the optimal splitting criterion $\mathcal{S}^*_p$ for every predictor $p \in \{1, 2, \ldots, P\}$. However, the specific manner in which any particular predictor p's optimal splitting criterion $\mathcal{S}^*_p$ is determined will depend on whether p is an ordered or categorical predictor.

2.1.1 Splitting on an Ordered Predictor

The splitting criterion $\mathcal{S}_p$ for an ordered predictor p is characterized by a numeric "split point" $s_p \in \mathbb{R}$ that defines the half-line $\mathcal{S}_p = \{x \in \mathbb{R} : x \leq s_p\}$. Thus, as can be observed from (1), a $(p, \mathcal{S}_p)$ pair will partition a mother node $\mathcal{N}_\mathcal{M}$ into the left and right daughter nodes that are defined, respectively, by

$$\mathcal{N}_\mathcal{L}(p, \mathcal{S}_p) = \{n \in \mathcal{N}_\mathcal{M} : x_{np} \leq s_p\} \quad \text{and} \quad \mathcal{N}_\mathcal{R}(p, \mathcal{S}_p) = \{n \in \mathcal{N}_\mathcal{M} : x_{np} > s_p\}.$$

Therefore, determining the optimal splitting criterion $\mathcal{S}_p^* = \{x \in \mathbb{R} : x \leq s_p^*\}$ for an ordered predictor p is straightforward—it can be greedily found by searching through all of the observed training values in the mother node in order to find the optimal numeric split point $s_p^* \in \{x_{np} \in \mathbb{R} : n \in \mathcal{N}_\mathcal{M}\}$ that minimizes the requisite node impurity measure which is given by either (3) or (5).

2.1.2 Splitting on a Categorical Predictor

For a categorical predictor p with Q possible unordered levels which are indexed by the set $\mathcal{Q} = \{1, 2, \ldots, Q\}$, the splitting criterion $\mathcal{S}_p \subset \mathcal{Q}$ is defined by the subset of levels that gets sent to the left daughter node $\mathcal{N}_\mathcal{L}$, while the complement set $\mathcal{S}_p' = \mathcal{Q} \setminus \mathcal{S}_p$ defines the subset of levels that gets sent to the right daughter node $\mathcal{N}_\mathcal{R}$. For notational simplicity and ease of exposition, in the remainder of this section we assume that all Q unordered levels of p are present in the mother node $\mathcal{N}_\mathcal{M}$ during training since it is only these present levels which contribute to the measure of node impurity when determining p's optimal splitting criterion $\mathcal{S}_p^*$. Later, in Section 3, we extend our notation to also account for any unordered levels of a categorical predictor p which are absent from the mother node $\mathcal{N}_\mathcal{M}$ during training.

Consequently, there are are $2^{Q-1}-1$ non-redundant ways of partitioning the Q unordered levels of p into the two daughter nodes, making it computationally expensive to evaluate the resulting measure of node impurity for every possible split when Q is large. However, this computation simplifies in certain situations.

In the case of a regression tree with a squared error node impurity measure, a categorical predictor p's optimal splitting criterion $\mathcal{S}_p^*$ can be determined by using a procedure described in Fisher (1958). Specifically, the training observations in the mother node are first used to calculate the mean response within each of p's unordered levels:

$$\gamma_p(q) = \text{ave}(y_n \mid n \in \mathcal{N}_\mathcal{M} \text{ and } x_{np} = q)\,, \quad q \in \mathcal{Q}. \tag{6}$$

These means are then used to assign numeric "pseudo values" $\tilde{x}_{np} \in \mathbb{R}$ to every training observation that is in the mother node according to its observed level for predictor p:

$$\tilde{x}_{np} = \gamma_p(x_{np}), \quad n \in \mathcal{N}_\mathcal{M}. \tag{7}$$

Finally, the optimal splitting criterion $\mathcal{S}_p^*$ for the categorical predictor p is determined by doing an ordered split on these numeric pseudo values $\tilde{x}_{np}$—that is, a corresponding optimal "pseudo splitting criterion" $\tilde{\mathcal{S}}_p^* = \{\tilde{x} \in \mathbb{R} : \tilde{x} \leq \tilde{s}_p^*\}$ is greedily chosen by scanning through all of the assigned numeric pseudo values in the mother node in order to find the optimal numeric "pseudo split point" $\tilde{s}_p^* \in \{\tilde{x}_{np} \in \mathbb{R} : n \in \mathcal{N}_\mathcal{M}\}$ which minimizes the resulting squared error node impurity measure given in (3) with respect to the left and right daughter nodes that are defined, respectively, by

$$\mathcal{N}_\mathcal{L}(p, \tilde{\mathcal{S}}_p^*) = \{n \in \mathcal{N}_\mathcal{M} : \tilde{x}_{np} \leq \tilde{s}_p^*\} \quad \text{and} \quad \mathcal{N}_\mathcal{R}(p, \tilde{\mathcal{S}}_p^*) = \{n \in \mathcal{N}_\mathcal{M} : \tilde{x}_{np} > \tilde{s}_p^*\}\,. \tag{8}$$

Meanwhile, in the case of a classification tree with a weighted Gini index node impurity measure, whether the computation simplifies or not is dependent on the number of response classes. For the $K > 2$ multiclass classification context, no such simplification is possible, although several approximations have been proposed (Loh and Vanichsetakul, 1988). However, for the $K = 2$ binary classification situation, a similar procedure to the one that was

just described for regression trees can be used. Specifically, the proportion of the training observations in the mother node that belong to the $k = 1$ response class is first calculated within each of categorical predictor p's unordered levels:

$$\gamma_p(q) = \frac{|\{n \in \mathcal{N}_\mathcal{M} : x_{np} = q \text{ and } y_n = 1\}|}{|\{n \in \mathcal{N}_\mathcal{M} : x_{np} = q\}|}, \quad q \in \mathcal{Q}, \tag{9}$$

and where we note here that $\gamma_p(q) \geq 0$ for all q since these proportions are, by definition, nonnegative. Afterwards, and just as in equation (7), these $k = 1$ response class proportions are used to assign numeric pseudo values $\tilde{x}_{np} \in \mathbb{R}$ to every training observation that is in the mother node $\mathcal{N}_\mathcal{M}$ according to its observed level for predictor p. And once again, the optimal splitting criterion $\mathcal{S}_p^*$ for the categorical predictor p is then determined by performing an ordered split on these numeric pseudo values $\tilde{x}_{np}$—that is, a corresponding optimal pseudo splitting criterion $\tilde{\mathcal{S}}_p^* = \{x \in \mathbb{R} : x \leq \tilde{s}_p^*\}$ is greedily found by searching through all of the assigned numeric pseudo values in the mother node in order to find the optimal numeric pseudo split point $\tilde{s}_p^* \in \{\tilde{x}_{np} \in \mathbb{R} : n \in \mathcal{N}_\mathcal{M}\}$ which minimizes the weighted Gini index node impurity measure given by (5) with respect to the resulting two daughter nodes as defined in (8). The proof that this procedure gives the optimal split in a binary classification tree in terms of the weighted Gini index amongst all possible splits can be found in Breiman et al. (1984) and Ripley (1996).

Therefore, in both regression and binary classification trees, we note that the optimal splitting criterion $\mathcal{S}_p^*$ for a categorical predictor p can be expressed in terms of the criterion's associated optimal numeric pseudo split point $\tilde{s}_p^*$ and the requisite means or $k = 1$ response class proportions $\gamma_p(q)$ of the unordered levels $q \in \mathcal{Q}$ of p as follows:

- The unordered levels of p that are being sent *left* have means or $k = 1$ response class proportions $\gamma_p(q)$ that are *less than or equal to* $\tilde{s}_p^*$:

$$\mathcal{S}_p^* = \{q \in \mathcal{Q} : \gamma_p(q) \leq \tilde{s}_p^*\}\,. \tag{10}$$

- The unordered levels of p that are being sent *right* have means or $k = 1$ response class proportions $\gamma_p(q)$ that are *greater than* $\tilde{s}_p^*$:

$$\mathcal{S}_p^{*\prime} = \{q \in \mathcal{Q} : \gamma_p(q) > \tilde{s}_p^*\}\,. \tag{11}$$

As we later discuss in Section 3, equations (10) and (11) lead to inherent differences in the left and right daughter nodes when splitting a mother node on a categorical predictor in CART—differences that can have significant ramifications when making predictions and inferences for observations with absent levels.

2.2 Random Forests

Introduced in Breiman (2001), random forests are an ensemble learning method that corrects for each individual tree's propensity to overfit the training set. This is accomplished through the use of bagging and a CART-like tree learning algorithm in order to build a large collection of "de-correlated" decision trees.

2.2.1 Bagging

Proposed in Breiman (1996a), bagging is an ensembling technique for improving the accuracy and stability of models. Specifically, given a training set

$$Z = \{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\},$$

this is achieved by repeatedly sampling N' observations with replacement from Z in order to generate B bootstrapped training sets $Z_1, Z_2, \ldots, Z_B$, where usually $N' = N$. A separate model is then trained on each bootstrapped training set Z_b, where we denote model b's prediction on an observation x as $\hat{f}_b(x)$. Here, showing each model a different bootstrapped sample helps to de-correlate them, and the overall bagged estimate $\hat{f}(x)$ for an observation x can then be obtained by averaging over all of the individual predictions in the case of regression

$$\hat{f}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x),$$

or by taking the majority vote in the case of classification

$$\hat{f}(x) = \underset{k \in \mathcal{K}}{\arg\max} \left(\sum_{b=1}^{B} I\left(\hat{f}_b(x) = k\right) \right).$$

One important aspect of bagging is the fact that each training observation n will only appear "in-bag" in a subset of the bootstrapped training sets Z_b. Therefore, for each training observation n, an "out-of-bag" (OOB) prediction can be constructed by only considering the subset of models in which n did not appear in the bootstrapped training set. Moreover, an OOB error for a bagged model can be obtained by evaluating the OOB predictions for all N training observations—a performance metric which helps to alleviate the need for cross-validation or a separate test set (Breiman, 1996b).

2.2.2 CART-Like Tree Learning Algorithm

In the case of random forests, the model that is being trained on each individual bootstrapped training set Z_b is a decision tree which is grown using the CART methodology, but with two key modifications.

First, as mentioned previously in Section 2, the trees that are grown in random forests are generally not pruned (Breiman, 2001). And second, instead of considering all P predictors at a split, only a randomly selected subset of the P predictors is allowed to be used—a restriction which helps to de-correlate the trees by placing a constraint on how similarly they can be grown. This process, which is known as the random subspace method, was developed in Amit and Geman (1997) and Ho (1998).

3. The Absent Levels Problem

In Section 1, we defined the absent levels problem as the inherent issue for decision tree based methods occurring whenever there is an indeterminacy over how to handle an observation that has reached a categorical split which was determined when the observation in question's

level was absent during training, and we described the three different ways in which the absent levels problem can arise. Then, in Section 2.1.2, we discussed how the levels of a categorical predictor p which were present in the mother node $\mathcal{N}_\mathcal{M}$ during training were used to determine its optimal splitting criterion $\mathcal{S}_p^*$. In this section, we investigate the potential consequences of overlooking the absent levels problem where, for a categorical predictor p with Q unordered levels which are indexed by the set $\mathcal{Q} = \{1, 2, \ldots, Q\}$, we now also further denote the subset of the levels of p that were present or absent in the mother node $\mathcal{N}_\mathcal{M}$ during training, respectively, as follows:

$$\begin{aligned} \mathcal{Q}_\mathcal{P} &= \{q \in \mathcal{Q} : |\{n \in \mathcal{N}_\mathcal{M} : x_{np} = q\}| > 0\}, \\ \mathcal{Q}_\mathcal{A} &= \{q \in \mathcal{Q} : |\{n \in \mathcal{N}_\mathcal{M} : x_{np} = q\}| = 0\}. \end{aligned} \tag{12}$$

Specifically, by documenting how absent levels have been handled by Breiman and Cutler's random forests `FORTRAN` code and the `randomForest R` package, we show how failing to account for the absent levels problem can systematically bias a model in practice. However, although our investigations are motivated by these two particular software implementations of random forests, we emphasize that the absent levels problem is, first and foremost, an intrinsic methodological issue for decision tree based methods.

3.1 Regression

For regression trees using a squared error node impurity measure, recall from our discussions in Section 2.1.2 and equations (6), (10), and (11), that the split of a mother node $\mathcal{N}_\mathcal{M}$ on a categorical predictor p can be characterized in terms of the splitting criterion's associated optimal numeric pseudo split point $\tilde{s}_p^*$ and the means $\gamma_p(q)$ of the unordered levels $q \in \mathcal{Q}$ of p as follows:

- The unordered levels of p being sent *left* have means $\gamma_p(q)$ that are *less than or equal to* $\tilde{s}_p^*$.
- The unordered levels of p being sent *right* have means $\gamma_p(q)$ that are *greater than* $\tilde{s}_p^*$.

Furthermore, recall from (2), that a node's prediction is given by the mean of the training responses that are in the node. Therefore, because the prediction of each daughter node can be expressed as a weighted average over the means $\gamma_p(q)$ of the present levels $q \in \mathcal{Q}_\mathcal{P}$ that are being sent to it, it follows that *the left daughter node $\mathcal{N}_\mathcal{L}$ will always give a prediction that is smaller than the right daughter node $\mathcal{N}_\mathcal{R}$ when splitting on a categorical predictor p in a regression tree that uses a squared error node impurity measure.*

In terms of execution, both the random forests `FORTRAN` code and the `randomForest` R package employ the pseudo value procedure for regression that was described in Section 2.1.2 when determining the optimal splitting criterion $\mathcal{S}_p^*$ for a categorical predictor p. However, the code that is responsible for calculating the mean $\gamma_p(q)$ within each unordered level $q \in \mathcal{Q}$ as in equation (6) behaves as follows:

$$\gamma_p(q) = \begin{cases} \text{ave}(y_n \mid n \in \mathcal{N}_\mathcal{M} \text{ and } x_{np} = q) & \text{if } q \in \mathcal{Q}_\mathcal{P} \\ 0 & \text{if } q \in \mathcal{Q}_\mathcal{A} \end{cases},$$

where $\mathcal{Q}_\mathcal{P}$ and $\mathcal{Q}_\mathcal{A}$ are, respectively, the present and absent levels of p as defined in (12).

Although this "zero imputation" of the means $\gamma_p(q)$ for the absent levels $q \in \mathcal{Q}_{\mathcal{A}}$ is inconsequential when determining the optimal numeric pseudo split point $\tilde{s}_p^*$ during training, it can be highly influential on the subsequent predictions that are made for observations with absent levels. In particular, by (10) and (11), the absent levels $q \in \mathcal{Q}_{\mathcal{A}}$ will be sent left if $\tilde{s}_p^* \geq 0$, and they will be sent right if $\tilde{s}_p^* < 0$. But, due to the systematic differences that exist amongst the two daughter nodes, this arbitrary decision of sending the absent levels left versus right can significantly impact the predictions that are made on observations with absent levels—even though the model's final predictions will also depend on any ensuing splits which take place after the absent levels problem occurs, observations with absent levels will tend to be biased towards smaller predictions when they are sent to the left daughter node, and they will tend to be biased towards larger predictions when they are sent to the right daughter node.

In addition, this behavior also implies that the random forest regression models which are trained using either the random forests `FORTRAN` code or the `randomForest R` package are sensitive to the set of possible values that the training responses can take. To illustrate, consider the following two extreme cases when splitting a mother node $\mathcal{N}_{\mathcal{M}}$ on a categorical predictor p:

- If the training responses $y_n > 0$ for all n, then the pseudo numeric split point $\tilde{s}_p^* > 0$ since the means $\gamma_p(q) > 0$ for all of the present levels $q \in \mathcal{Q}_{\mathcal{P}}$. And because the "imputed" means $\gamma_p(q) = 0 < \tilde{s}_p^*$ for all $q \in \mathcal{Q}_{\mathcal{A}}$, the absent levels will always be sent to the left daughter node $\mathcal{N}_{\mathcal{L}}$ which gives smaller predictions.
- If the training responses $y_n < 0$ for all n, then the pseudo numeric split point $\tilde{s}_p^* < 0$ since the means $\gamma_p(q) < 0$ for all of the present levels $q \in \mathcal{Q}_{\mathcal{P}}$. And because the "imputed" means $\gamma_p(q) = 0 > \tilde{s}_p^*$ for all $q \in \mathcal{Q}_{\mathcal{A}}$, the absent levels will always be sent to the right daughter node $\mathcal{N}_{\mathcal{R}}$ which gives larger predictions.

And although this sensitivity to the training response values was most easily demonstrated through these two extreme situations, the reader should not let this overshadow the fact that the absent levels problem can also heavily influence a model's performance in more general circumstances (e.g., when the training responses are of mixed signs).

3.2 Classification

For binary classification trees using a weighted Gini index node impurity measure, recall from our discussions in Section 2.1.2 and equations (9), (10), and (11), that the split of a mother node $\mathcal{N}_{\mathcal{M}}$ on a categorical predictor p can be characterized in terms of the splitting criterion's associated optimal numeric pseudo split point $\tilde{s}_p^*$ and the $k = 1$ response class proportions $\gamma_p(q)$ of the unordered levels $q \in \mathcal{Q}$ of p as follows:

- The unordered levels of p being sent *left* have $k = 1$ response class proportions $\gamma_p(q)$ that are *less than or equal to* $\tilde{s}_p^*$.
- The unordered levels of p being sent *right* have $k = 1$ response class proportions $\gamma_p(q)$ that are *greater than* $\tilde{s}_p^*$.

In addition, recall from (4), that a node's classification is given by the response class that occurs the most amongst the training observations that are in the node. Therefore, because

the response class proportions of each daughter node can be expressed as a weighted average over the response class proportions of the present levels $q \in \mathcal{Q}_{\mathcal{P}}$ that are being sent to it, it follows that *the left daughter node $\mathcal{N}_{\mathcal{L}}$ is always less likely to classify an observation to the $k = 1$ response class than the right daughter node $\mathcal{N}_{\mathcal{R}}$ when splitting on a categorical predictor p in a binary classification tree that uses a weighted Gini index node impurity measure.*

In terms of implementation, the `randomForest R` package uses the pseudo value procedure for binary classification that was described in Section 2.1.2 when determining the optimal splitting criterion $\mathcal{S}_p^*$ for a categorical predictor p with a "large" number of unordered levels.[2] However, the code that is responsible for computing the $k = 1$ response class proportion $\gamma_p(q)$ within each unordered level $q \in \mathcal{Q}$ as in equation (9) executes as follows:

$$\gamma_p(q) = \begin{cases} \frac{|\{n \in \mathcal{N}_{\mathcal{M}} : x_{np} = q \text{ and } y_n = 1\}|}{|\{n \in \mathcal{N}_{\mathcal{M}} : x_{np} = q\}|} & \text{if } q \in \mathcal{Q}_{\mathcal{P}} \\ 0 & \text{if } q \in \mathcal{Q}_{\mathcal{A}} \end{cases}.$$

Therefore, the issues that arise here are similar to the ones that were described for regression.

Even though this "zero imputation" of the $k = 1$ response class proportions $\gamma_p(q)$ for the absent levels $q \in \mathcal{Q}_{\mathcal{A}}$ is unimportant when determining the optimal numeric pseudo split point $\tilde{s}_p^*$ during training, it can have a large effect on the subsequent classifications that are made for observations with absent levels. In particular, since the proportions $\gamma_p(q) \geq 0$ for all of the present levels $q \in \mathcal{Q}_{\mathcal{P}}$, it follows from our discussions in Section 2.1.2 that the numeric pseudo split point $\tilde{s}_p^* \geq 0$. And because the "imputed" proportions $\gamma_p(q) = 0 \leq \tilde{s}_p^*$ for all $q \in \mathcal{Q}_{\mathcal{A}}$, the absent levels will always be sent to the left daughter node. But, due to the innate differences that exist amongst the two daughter nodes, this arbitrary choice of sending the absent levels left can significantly affect the classifications that are made on observations with absent levels—although the model's final classifications will also depend on any successive splits which take place after the absent levels problem occurs, the classifications for observations with absent levels will tend to be biased towards the $k = 2$ response class. Moreover, this behavior also implies that the random forest binary classification models which are trained using the `randomForest R` package may be sensitive to the actual ordering of the response classes: since observations with absent levels are always sent to the left daughter node $\mathcal{N}_{\mathcal{L}}$ which is more likely to classify them to the $k = 2$ response class than the right daughter node $\mathcal{N}_{\mathcal{R}}$, the classifications for these observations can be influenced by interchanging the indices of the two response classes.

Meanwhile, for cases where the pseudo value procedure is not or cannot be used, the random forests `FORTRAN` code and the `randomForest R` package will instead adopt a more brute force approach that either exhaustively or randomly searches through the space of possible splits. However, to understand the potential problems that absent levels can cause in these situations, we must first briefly digress into a discussion of how categorical splits are internally represented in their code.

Specifically, in their code, a split on a categorical predictor p is both encoded and decoded as an integer whose binary representation identifies which unordered levels go left

[2] The exact condition for using the pseudo value procedure for binary classification in version 4.6-12 of the `randomForest R` package is when a categorical predictor p has $Q > 10$ unordered levels. Meanwhile, although the random forests `FORTRAN` code for binary classification references the pseudo value procedure, it does not appear to be implemented in the code.

(the bits that are "turned on") and which unordered levels go right (the bits that are "turned off"). To illustrate, consider the situation where a categorical predictor p has four unordered levels, and where the integer encoding of the split is 5. In this case, since 0101 is the binary representation of the integer 5 (because $5 = [0] \cdot 2^3 + [1] \cdot 2^2 + [0] \cdot 2^1 + [1] \cdot 2^0$), levels 1 and 3 get sent left while levels 2 and 4 get sent right.

Now, when executing an exhaustive search to find the optimal splitting criterion $\mathcal{S}_p^*$ for a categorical predictor p with Q unordered levels, the random forests `FORTRAN` code and the `randomForest R` package will both follow the same systematic procedure:[3] *all $2^{Q-1} - 1$ possible integer encodings for the non-redundant partitions of the unordered levels of predictor p are evaluated in increasing sequential order starting from 1 and ending at $2^{Q-1} - 1$, with the choice of the optimal splitting criterion $\mathcal{S}_p^*$ being updated if and only if the resulting weighted Gini index node impurity measure strictly improves.*

But since the absent levels $q \in \mathcal{Q}_\mathcal{A}$ are not present in the mother node $\mathcal{N}_\mathcal{M}$ during training, *sending them left or right has no effect on the resulting weighted Gini index.* And because turning on the bit for any particular level q while holding the bits for all of the other levels constant will always result in a larger integer, it follows that *the exhaustive search that is used by these two software implementations will always prefer splits that send all of the absent levels right since they are always checked before any of their analogous Gini index equivalent splits that send some of the absent levels left.*

Furthermore, in their exhaustive search, the leftmost bit corresponding to the Q^{th} indexed unordered level of a categorical predictor p is always turned off since checking the splits where this bit is turned on would be redundant—they would amount to just swapping the "left" and "right" daughter node labels for splits that have already been evaluated. Consequently, the Q^{th} indexed level of p will also always be sent to the right daughter node and, as a result, the classifications for observations with absent levels will tend to be biased towards the response class distribution of the training observations in the mother node $\mathcal{N}_\mathcal{M}$ that belong to this Q^{th} indexed level. Therefore, although it may sound contradictory, this also implies that the random forest multiclass classification models which are trained using either the random forests `FORTRAN` code or the `randomForest R` package may be sensitive to the actual ordering of a categorical predictor's unordered levels—a reordering of these levels could potentially interchange the "left" and "right" daughter node labels, which could then subsequently affect the classifications that are made for observations with absent levels since they will always be sent to whichever node ends up being designated as the "right" daughter node.

Finally, when a categorical predictor p has too many levels for an exhaustive search to be computationally efficient, both the random forests `FORTRAN` code and the `randomForest R` package will resort to approximating the optimal splitting criterion $\mathcal{S}_p^*$ with the best split that was found amongst a large number of randomly generated splits.[4] This is accomplished by randomly setting all of the bits in the binary representations of the splits to either a

[3] The random forests `FORTRAN` code will use an exhaustive search for both binary and multiclass classification whenever $Q < 25$. In version 4.6-12 of the `randomForest R` package, an exhaustive search will be used for both binary and multiclass classification whenever $Q < 10$.

[4] The random forests `FORTRAN` code will use a random search for both binary and multiclass classification whenever $Q \geq 25$. In version 4.6-12 of the `randomForest R` package, a random search will only be used when $Q \geq 10$ in the multiclass classification case.

0 or a 1—a procedure which ultimately results in each absent level being randomly sent to either the left or right daughter node with equal probability. As a result, although the absent levels problem can still occur in these situations, it is difficult to determine whether it results in any systematic bias. However, it is still an open question as to whether or not such a treatment of absent levels is sufficient.

4. Heuristics for Mitigating the Absent Levels Problem

Although a comprehensive theoretical analysis of the absent levels problem is beyond the scope of this paper, in this section we briefly consider several heuristics which may be able to help mitigate the issue. Later, in Section 5, we empirically evaluate and compare how some of these heuristics perform in practice when they are applied to three real data examples.

4.1 Missing Data Heuristics

Even though absent levels are fully observed and known, the missing data literature for decision tree based methods is still perhaps the area of existing research that is most closely related to the absent levels problem.

4.1.1 Stop

One straightforward missing data strategy for dealing with absent levels would be to simply stop an observation from going further down the tree whenever the issue occurs and just use the mother node for prediction—a missing data approach which has been adopted by both the `rpart R` package for CART (Therneau et al., 2015) and the `gbm R` package for generalized boosted regression models (Ridgeway, 2013). Even with this missing data functionality already in place, however, the `gbm R` package has still had its own issues in readily extending it to the case of absent levels—serving as another example of a software implementation of a decision tree based method that has overlooked and suffered from the absent levels problem.[5]

4.1.2 Distribution-Based Imputation (DBI)

Another potential missing data technique would be to send an observation with an absent level down both daughter nodes—perhaps by using the distribution-based imputation (DBI) technique which is employed by the C4.5 algorithm for growing decision trees (Quinlan, 1993). In particular, an observation that encounters an infeasible node split is first split into multiple pseudo-instances, where each instance takes on a different imputed value and weight based on the distribution of observed values for the splitting variable in the mother node's subset of the training data. These pseudo-instances are then sent down their appropriate daughter nodes in order to proceed down the tree as usual, and the final prediction is derived from the weighted predictions of all the terminal nodes that are subsequently reached (Saar-Tsechansky and Provost, 2007).

[5] See, for example, `https://code.google.com/archive/p/gradientboostedmodels/issues/7`

4.1.3 Surrogate Splits

Surrogate splitting, which the `rpart` `R` package also supports, is arguably the most popular method of handling missing data in CART, and it may provide another workable approach for mitigating the effects of absent levels. Specifically, if $(p^*, \mathcal{S}^*_{p^*})$ is found to be the optimal splitting variable and splitting criterion pair for a mother node $\mathcal{N}_\mathcal{M}$, then the first surrogate split is the $(p', \mathcal{S}_{p'})$ pair where $p' \neq p^*$ that yields the split which most closely mimics the optimal split's binary partitioning of $\mathcal{N}_\mathcal{M}$, the second surrogate split is the $(p'', \mathcal{S}_{p''})$ pair where $p'' \notin \{p^*, p'\}$ resulting in the second most similar binary partitioning of $\mathcal{N}_\mathcal{M}$ as the optimal split, and so on. Afterwards, when an observation reaches an indeterminate split, the surrogates are tried in the order of decreasing similarity until one of them becomes feasible (Breiman et al., 1984).

However, despite its extensive use in CART, surrogate splitting may not be entirely appropriate for ensemble tree methods like random forests. As pointed out in Ishwaran et al. (2008):

> Although surrogate splitting works well for trees, the method may not be well suited for forests. Speed is one issue. Finding a surrogate split is computationally intensive and may become infeasible when growing a large number of trees, especially for fully saturated trees used by forests. Further, surrogate splits may not even be meaningful in a forest paradigm. [Random forests] randomly selects variables when splitting a node and, as such, variables within a node may be uncorrelated, and a reasonable surrogate split may not exist. Another concern is that surrogate splitting alters the interpretation of a variable, which affects measures such as [variable importance].

Nevertheless, surrogate splitting is still available as a non-default option for handling missing data in the `partykit` `R` package (Hothorn and Zeileis, 2015), which is an implementation of a bagging ensemble of conditional inference trees that correct for the biased variable selection issues which exist in several tree learning algorithms like CART and C4.5 (Hothorn et al., 2006).

4.1.4 Random/Majority

The `partykit` `R` package also provides some other functionality for dealing with missing data that may be applicable to the absent levels problem. These include the package's default approach of randomly sending the observations to one of the two daughter nodes with the weighting done by the number of training observations in each node or, alternatively, by simply having the observations go to the daughter node with more training observations. Interestingly, the `partykit` `R` package does appear to recognize the possibility of absent levels occurring, and chooses to handle them as if they were missing—its reference manual states that "Factors in test samples whose levels were empty in the learning sample are treated as missing when computing predictions." Whether or not such missing data heuristics adequately address the absent levels problem, however, is still unknown.

4.2 Feature Engineering Heuristics

Apart from missing data methods, feature engineering techniques which transform the categorical predictors may also be viable approaches to mitigating the effects of absent levels.

However, feature engineering techniques are not without their own drawbacks. First, transforming the categorical predictors may not always be feasible in practice since the feature space may become computationally unmanageable. And even when transformations are possible, they may further exacerbate variable selection issues—many popular tree learning algorithms such as CART and C4.5 are known to be biased in favor of splitting on ordered predictors and categorical predictors with many unordered levels since they offer more candidate splitting points to choose from (Hothorn et al., 2006). Moreover, by recoding a categorical predictor's unordered levels into several different predictors, we forfeit a decision tree based method's natural ability to simultaneously consider all of the predictor's levels together at a single split. Thus, it is not clear whether feature engineering techniques are preferable when using decision tree based methods.

Despite these potential shortcomings, transformations of the categorical predictors is currently required by the `scikit-learn` `Python` module's implementation of random forests (Pedregosa et al., 2011). There have, however, been some discussions about extending the module so that it can support the native categorical split capabilities used by the random forests `FORTRAN` code and the `randomForest` `R` package.[6] But, needless to say, such efforts would also have the unfortunate consequence of introducing the indeterminacy of the absent levels problem into another popular software implementation of a decision tree based method.

4.2.1 One-Hot Encoding

Nevertheless, one-hot encoding is perhaps the most straightforward feature engineering technique that could be applied to the absent levels problem—even though some unordered levels may still be absent when determining a categorical split during training, any uncertainty over where to subsequently send these absent levels would be eliminated by recoding the levels of each categorical predictor into separate dummy predictors.

5. Examples

Although the actual severity of the absent levels problem will depend on the specific data set and task at hand, in this section we present three real data examples which illustrate how the absent levels problem can dramatically alter the performance of decision tree based methods in practice. In particular, we empirically evaluate and compare how the seven different heuristics in the set

$$\mathcal{H} = \{\textit{Left}, \textit{Right}, \textit{Stop}, \textit{Majority}, \textit{Random}, \textit{DBI}, \textit{One-Hot}\}$$

perform when confronted with the absent levels problem in random forests.

In particular, the first two heuristics that we consider in our set $\mathcal{H}$ are the systematically biased approaches discussed in Section 3 which have been employed by both the random

[6] See, for example, `https://github.com/scikit-learn/scikit-learn/pull/3346`

forests `FORTRAN` code and the `randomForest R` package due to having overlooked the absent levels problem:

- Left: Sending the observation to the left daughter node.
- Right: Sending the observation to the right daughter node.

Consequently, these two "naive heuristics" have been included in our analysis for comparative purposes only.

In our set of heuristics $\mathcal{H}$, we also consider some of the missing data strategies for decision tree based methods that we discussed in Section 4:

- Stop: Stopping the observation from going further down the tree and using the mother node for prediction.
- Majority: Sending the observation to the daughter node with more training observations, with any ties being broken randomly.
- Random: Randomly sending the observation to one of the two daughter nodes, with the weighting done by the number of training observations in each node.[7]
- Distribution-Based Imputation (DBI): Sending the observation to both daughter nodes using the C4.5 tree learning algorithm's DBI approach.

Unlike the two naive heuristics, these "missing data heuristics" are all less systematic in their preferences amongst the two daughter nodes.

Finally, in our set $\mathcal{H}$, we also consider a "feature engineering heuristic" which transforms all of the categorical predictors in the original data set:

- One-Hot: Recoding every categorical predictor's set of possible unordered levels into separate dummy predictors

Under this heuristic, although unordered levels may still be absent when determining a categorical split during training, there is no longer any uncertainty over where to subsequently send observations with absent levels.

Code for implementing the naive and missing data heuristics was built on top of version 4.6-12 of the `randomForest R` package. Specifically, the `randomForest R` package is used to first train the random forest models as usual. Afterwards, each individual tree's in-bag training data is sent back down the tree according to the tree's set of decision rules in order to record the unordered levels that were absent at each categorical split. Finally, when making predictions or inferences, our code provides some functionality for carrying out each of the naive and missing data heuristics whenever the absent levels problem occurs.

Each of the random forest models that we consider in our analysis is trained "off-the-shelf" by using the `randomForest R` package's default settings for the algorithm's tuning

[7] We also investigated an alternative "unweighted" version of the Random heuristic which randomly sends observations with absent levels to either the left or right daughter node with equal probability (analogous to the random search procedure that was described at the end of Section 3.2). However, because this unweighted version was found to be generally inferior to the "weighted" version described in our analysis, we have omitted it from our discussions for expositional clarity and conciseness.

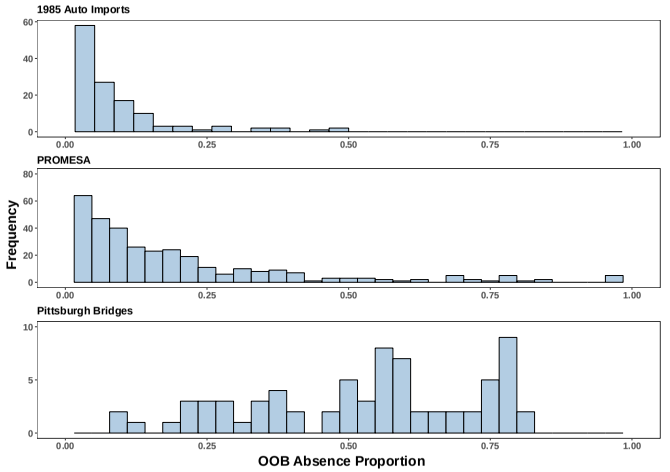


Figure 1: Histograms for each example's distribution of OOB absence proportions.

parameters. Moreover, to account for the inherent randomness in the random forests algorithm, we repeat each of our examples 1000 times with a different random seed used to initialize each experimental replication. However, because of the way in which we have structured our code, we note that our analysis is able to isolate the effects of the naive and missing data heuristics on the absent levels problem since, within each experimental replication, their underlying random forest models are identical with respect to each tree's in-bag training data and differ only in terms of their treatment of the absent levels. As a result, the predictions and inferences obtained from the naive and missing data heuristics will be positively correlated across the 1000 experimental replications that we consider for each example—a fact which we exploit in order to improve the precision of our comparisons.

Recall from Section 4, however, that the random forest models which are trained on feature engineered data sets are intrinsically different from the random forest models which are trained on their original untransformed data set counterparts. Therefore, although we use the same default `randomForest R` package settings and the same random seed to initialize each of the One-Hot heuristic's experimental replications, we note that its predictions and inferences will be essentially uncorrelated with the naive and missing data heuristics across each example's 1000 experimental replications.

5.1 1985 Auto Imports

For a regression example, we consider the 1985 Auto Imports data set from the UCI Machine Learning Repository (Lichman, 2013) which, after discarding observations with missing data, contains 25 predictors that can be used to predict the prices of 159 cars. Categorical predictors for which the absent levels problem can occur include a car's make (18 levels),

Statistic	1985 Auto Imports	PROMESA	Pittsburgh Bridges
Min	0.003	0.001	0.080
1st Quartile	0.021	0.020	0.368
Median	0.043	0.088	0.564
Mean	0.076	0.162	0.526
3rd Quartile	0.093	0.206	0.702
Max	0.498	0.992	0.820

Table 1: Summary statistics for each example's distribution of OOB absence proportions.

body style (5 levels), drive layout (3 levels), engine type (5 levels), and fuel system (6 levels). Furthermore, because all of the car prices are positive, we know from Section 3.1 that the random forests `FORTRAN` code and the `randomForest R` package will both always employ the Left heuristic when faced with absent levels for this particular data set.[8]

The top panel in Figure 1 depicts a histogram of this example's OOB absence proportions, which we define for each training observation as the proportion of its OOB trees across all 1000 experimental replications which had the absent levels problem occur at least once when using the training set with the original untransformed categorical predictors. Meanwhile, Table 1 provides a more detailed summary of this example's distribution of OOB absence proportions. Consequently, although there is a noticeable right skew in the distribution, we see that most of the observations in this example had the absent levels problem occur in less than 5% of their OOB trees.

Let $\hat{y}_{nr}^{(h)}$ denote the OOB prediction that a heuristic h makes for an observation n in an experimental replication r. Then, within each experimental replication r, we can compare the predictions that two different heuristics $h_1, h_2 \in \mathcal{H}$ make for an observation n by considering the difference $\hat{y}_{nr}^{(h_1)} - \hat{y}_{nr}^{(h_2)}$. We summarize these comparisons for all possible pairwise combinations of the seven heuristics in Figure 2, where each panel plots the mean and middle 95% of these differences across all 1000 experimental replications as a function of the OOB absence proportion. From the red intervals in Figure 2, we see that significant differences in the predictions of the heuristics do exist, with the magnitude of the point estimates and the width of the intervals tending to increase with the OOB absence proportion—behavior that agrees with our intuition that the distinctive effects of each heuristic should become more pronounced the more often the absent levels problem occurs.

In addition, we can evaluate the overall performance of each heuristic $h \in \mathcal{H}$ within an experimental replication r in terms of its root mean squared error (RMSE):

$$\text{RMSE}_r^{(h)} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_n - \hat{y}_{nr}^{(h)}\right)^2}.$$

[8] This is the case for versions 4.6-7 and earlier of the `randomForest R` package. Beginning in version 4.6-9, however, the `randomForest R` package began to internally mean center the training responses prior to fitting the model, with the mean being subsequently added back to the predictions of each node. Consequently, the Left heuristic isn't always used in these versions of the `randomForest R` package since the training responses that the model actually considers are of mixed sign. Nevertheless, such a strategy still fails to explicitly address the underlying absent levels problem.

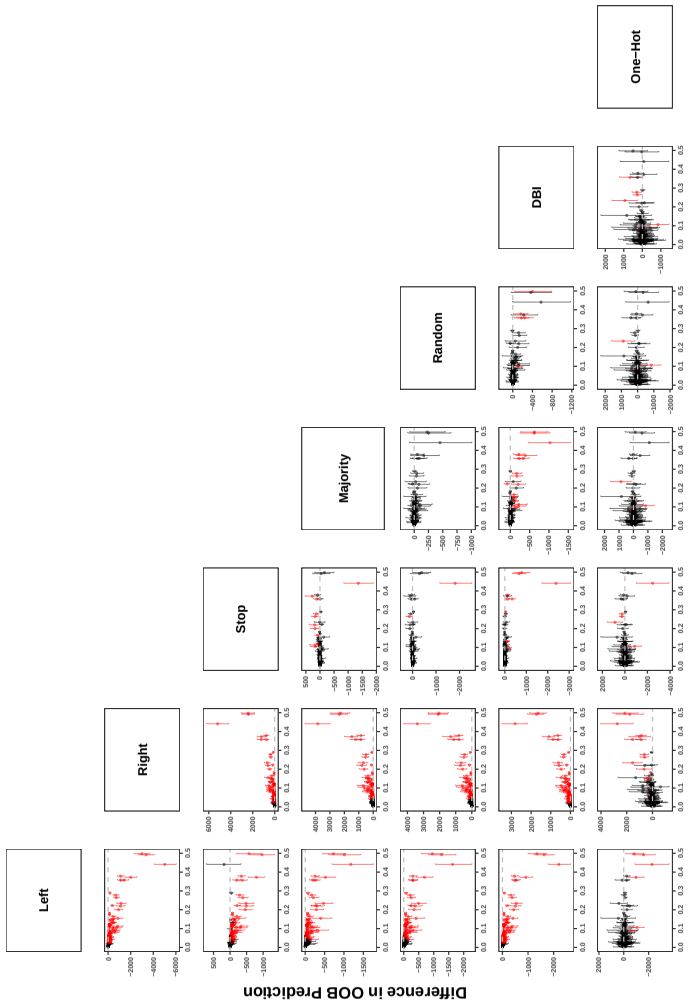


Figure 2: Pairwise differences in the OOB predictions as a function of the OOB absence proportions in the 1985 Auto Imports data set. Each panel plots the mean and middle 95% of the differences across all 1000 experimental replications when the OOB predictions of the heuristic that is labeled at the right of the panel's row is subtracted from the OOB predictions of the heuristic that is labeled at the top of the panel's column. Differences were taken within each experimental replication in order to account for the positive correlation that exists between the naive and missing data heuristics. Intervals containing zero (the horizontal dashed line) are in black, while intervals not containing zero are in red.

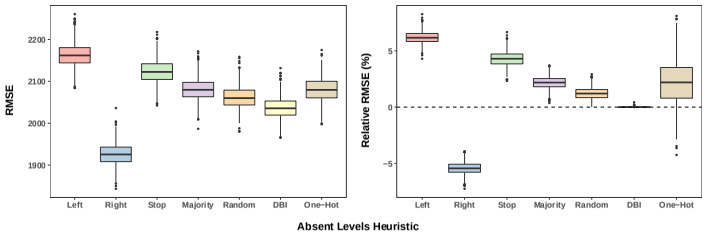


Figure 3: RMSEs for the OOB predictions of the seven heuristics in the 1985 Auto Imports data set. The left panel shows boxplots of each heuristic's marginal distribution of RMSEs across all 1000 experimental replications, which ignores the positive correlation that exists between the naive and missing data heuristics. The right panel accounts for this positive correlation by comparing the RMSEs of the heuristics relative to the best RMSE that was obtained amongst the missing data heuristics within each of the 1000 experimental replications as in (13).

Boxplots displaying each heuristic's marginal distribution of RMSEs across all 1000 experimental replications are shown in the left panel of Figure 3. However, these marginal boxplots ignore the positive correlation that exists between the naive and missing data heuristics. Therefore, within every experimental replication r, we also compare the RMSE for each heuristic $h \in \mathcal{H}$ relative to the best RMSE that was achieved amongst the missing data heuristics $\mathcal{H}_m = \{Stop, Majority, Random, DBI\}$:

$$\text{RMSE}_r^{(h\|\mathcal{H}_m)} = \frac{\text{RMSE}_r^{(h)} - \min_{h\in\mathcal{H}_m} \text{RMSE}_r^{(h)}}{\min_{h\in\mathcal{H}_m} \text{RMSE}_r^{(h)}}. \tag{13}$$

Here we note that the Left and Right heuristics were not considered in the definition of the best RMSE achieved within each experimental replication r due to the issues discussed in Section 3, while the One-Hot heuristic was excluded from this definition since it is essentially uncorrelated with the other six heuristics across all 1000 experimental replications. Boxplots of these relative RMSEs are shown in the right panel of Figure 3.

5.1.1 Naive Heuristics

Relative to all of the other heuristics and consistent with our discussions in Section 3.1, we see from Figure 2 that the Left and Right heuristics have a tendency to severely underpredict and overpredict, respectively. Furthermore, for this particular example, we notice from Figure 3 that the random forests `FORTRAN` code and the `randomForest R` package's behavior of always sending absent levels left in this particular data set substantially underperforms relative to the other heuristics—it gives an RMSE that is, on average, 6.2% worse than the best performing missing data heuristic. And although the Right heuristic appears to

perform exceptionally well, we again stress the misleading nature of this performance—its tendency to overpredict just coincidentally happens to be beneficial in this specific situation.

5.1.2 Missing Data Heuristics

As can be seen from Figure 2, the predictions obtained from the four missing data heuristics are more aligned with one another than they are with the Left, Right, and One-Hot heuristics. Considerable disparities in their predictions do still exist, however, and from Figure 3 we note that amongst the four missing data heuristics, the DBI heuristic clearly performs the best. And although the Majority heuristic fares slightly worse than the Random heuristic, they both perform appreciably better than the Stop heuristic.

5.1.3 Feature Engineering Heuristic

Recall that the One-Hot heuristic is essentially uncorrelated with the other six heuristics across all 1000 experimental replications—a fact which is reflected in its noticeably wider intervals in Figure 2 and in its larger relative RMSE boxplot in Figure 3. Nevertheless, it can still be observed from Figure 3 that although the One-Hot heuristic's predictions are sometimes able to outperform the other heuristics, on average, it yields an RMSE that is 2.2% worse than than the best performing missing data heuristic.

5.2 PROMESA

For a binary classification example, we consider the June 9, 2016 United States House of Representatives vote on the Puerto Rico Oversight, Management, and Economic Stability Act (PROMESA) for addressing the Puerto Rican government's debt crisis. Data for this vote was obtained by using the `Rvoteview R` package to query the Voteview database (Lewis, 2015). After omitting those who did not vote on the bill, the data set contains four predictors that can be used to predict the binary "No" or "Yes" votes of 424 House of Representative members. These predictors include a categorical predictor for a representative's political party (2 levels), a categorical predictor for a representative's state (50 levels), and two ordered predictors which quantify aspects of a representative's political ideological position (McCarty et al., 1997).

The "No" vote was taken to be the $k = 1$ response class in our analysis, while the "Yes" vote was taken to be the $k = 2$ response class. Recall from Section 3.2, that this ordering of the response classes is meaningful in a binary classification context since the `randomForest R` package will always use the Left heuristic which biases predictions for observations with absent levels towards whichever response class is indexed by $k = 2$ (corresponding to the "Yes" vote in our analysis).

From Figure 1 and Table 1, we see that the absent levels problem occurs much more frequently in this example than it did in our 1985 Auto Imports example. In particular, the seven House of Representative members who were the sole representatives from their state had OOB absence proportions that were greater than 0.961 since the absent levels problem occurred for these observations every time they reached an OOB tree node that was split on the state predictor.

For random forest classification models, the predicted probability that an observation belongs to a response class k can be estimated by the proportion of the observation's trees

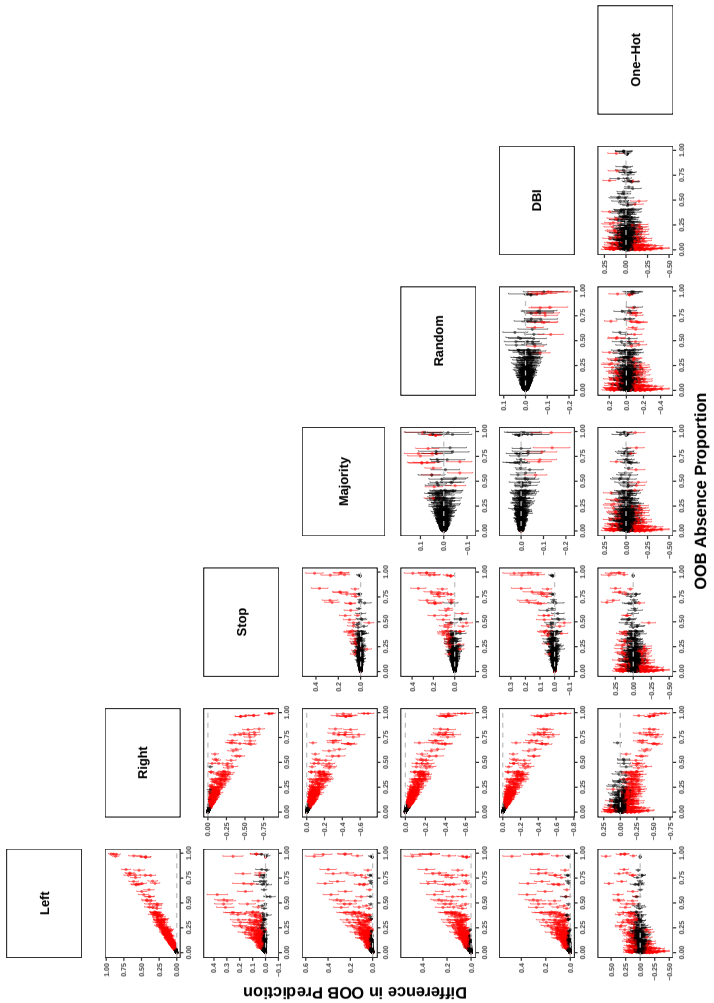


Figure 4: Pairwise differences in the OOB predicted probabilities of voting "Yes" as a function of the OOB absence proportion in the PROMESA data set. Each panel plots the mean and middle 95% of the pairwise differences across all 1000 experimental replications when the OOB predicted probabilities of the heuristic that is labeled at the right of the panel's row is subtracted from the OOB predicted probabilities of the heuristic that is labeled at the top of the panel's column. Differences were taken within each experimental replication to account for the positive correlation that exists between the naive and missing data heuristics. Intervals containing zero (the horizontal dashed line) are in black, while intervals not containing zero are in red.

which classify it to class k.[9] Let $\hat{p}_{nkr}^{(h)}$ denote the OOB predicted probability that a heuristic h assigns to an observation n of belonging to a response class k in an experimental replication r. Then, within each experimental replication r, we can compare the predicted probabilities that two different heuristics $h_1, h_2 \in \mathcal{H}$ assign to an observation n by considering the difference $\hat{p}_{nkr}^{(h_1)} - \hat{p}_{nkr}^{(h_2)}$. We summarize these differences in the predicted probabilities of voting "Yes" for all possible pairwise combinations of the seven heuristics in Figure 4, where each panel plots the mean and middle 95% of these differences across all 1000 experimental replications as a function of the OOB absence proportion.

The large discrepancies in the predicted probabilities that are observed in Figure 4 are particularly concerning since they can lead to different classifications. If we let $\hat{y}_{nr}^{(h)}$ denote the OOB classification that a heuristic h makes for an observation n in an experimental replication r, then Cohen's kappa coefficient (Cohen, 1960) provides one way of measuring the level of agreement between two different heuristics $h_1, h_2 \in \mathcal{H}$:

$$\kappa_r^{(h_1,h_2)} = \frac{o_r^{(h_1,h_2)} - e_r^{(h_1,h_2)}}{1 - e_r^{(h_1,h_2)}}, \tag{14}$$

where

$$o_r^{(h_1,h_2)} = \frac{1}{N}\sum_{n=1}^{N} I\left(\hat{y}_{nr}^{(h_1)} = \hat{y}_{nr}^{(h_2)}\right)$$

is the observed probability of agreement between the two heuristics, and where

$$e_r^{(h_1,h_2)} = \frac{1}{N^2}\sum_{k=1}^{K}\left[\left(\sum_{n=1}^{N} I\left(\hat{y}_{nr}^{(h_1)} = k\right)\right)\cdot\left(\sum_{n=1}^{N} I\left(\hat{y}_{nr}^{(h_2)} = k\right)\right)\right]$$

is the expected probability of the two heuristics agreeing by chance. Therefore, within an experimental replication r, we will observe $\kappa_r^{(h_1,h_2)} = 1$ if the two heuristics are in complete agreement, and we will observe $\kappa_r^{(h_1,h_2)} \approx 0$ if there is no agreement amongst the two heuristics other than what would be expected by chance. In Figure 5, we plot histograms of the Cohen's kappa coefficient for all possible pairwise combinations of the seven heuristics across all 1000 experimental replications when the random forests algorithm's default majority vote discrimination threshold of 0.5 is used.

More generally, the areas underneath the receiver operating characteristic (ROC) and precision-recall (PR) curves can be used to compare the overall performance of binary classifiers as the discrimination threshold is varied between 0 and 1. Specifically, as the discrimination threshold changes, the ROC curve plots the proportion of positive observations that a classifier correctly labels as a function of the proportion of negative observations that a classifier incorrectly labels, while the PR curve plots the proportion of a classifier's positive labels that are truly positive as a function of the proportion of positive observations that a classifier correctly labels (Davis and Goadrich, 2006).

[9] This is the approach that is used by the `randomForest R` package. The `scikit-learn Python` module uses an alternative method of calculating the predicted response class probabilities which takes the average of the predicted class probabilities over the trees in the random forest, where the predicted probability of a response class k in an individual tree is estimated using the proportion of a node's training samples that belong to the response class k.

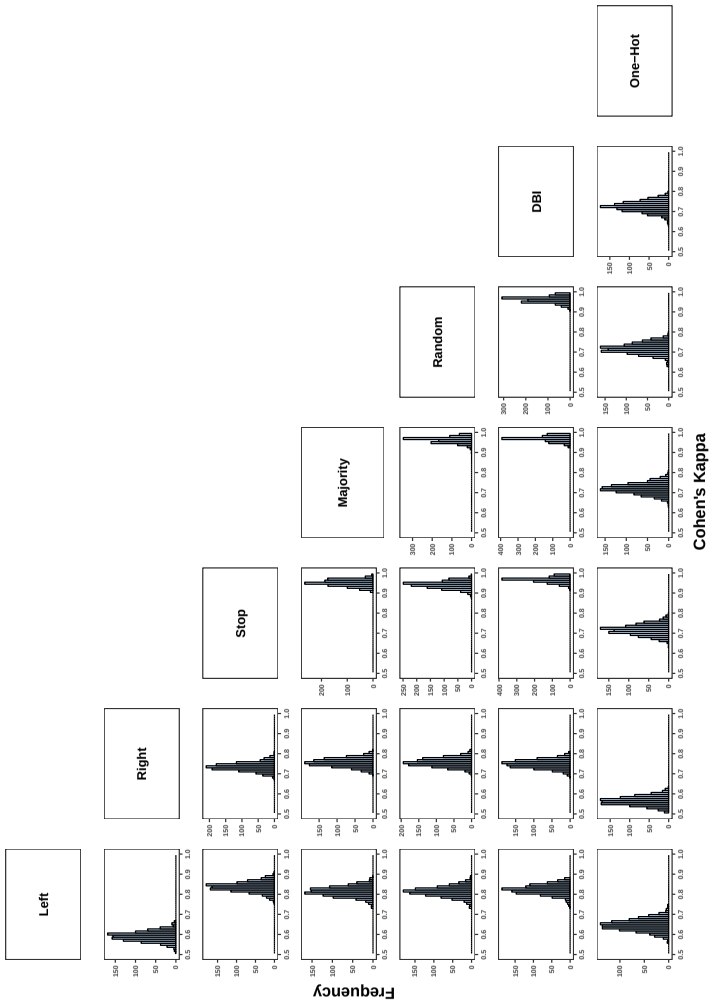


Figure 5: Pairwise Cohen's kappa coefficients for the seven different heuristics as defined in (14) when the random forests algorithm's default majority vote discrimination threshold of 0.5 is used in the PROMESA data set. Each panel plots the histogram of coefficients across all 1000 experimental replications when the OOB classifications of the heuristic that is labeled at the right of the panel's row are compared against the OOB classifications of the heuristic that is labeled at the top of the panel's column. Cohen's kappa coefficients were calculated within each of the 1000 experimental replications to account for the positive correlation between the naive and missing data heuristics.

Taking the "Yes" vote to be the positive response class in our analysis, we calculate the areas underneath the ROC and PR curves for each heuristic $h \in \mathcal{H}$ within each experimental replication r. Boxplots depicting each heuristic's marginal distribution of these two areas across all 1000 experimental replications are shown in the left panels of Figure 6. However, these marginal boxplots ignore the positive correlation that exists between the naive and missing data heuristics. Therefore, within every experimental replication r and similar to what was previously done in our 1985 Auto Imports example, we also compare the areas for each heuristic $h \in \mathcal{H}$ relative to the best area that was achieved amongst the missing data heuristics $\mathcal{H}_m = \{Stop, Majority, Random, DBI\}$:

$$\text{AUC}_r^{(h \| \mathcal{H}_m)} = \frac{\text{AUC}_r^{(h)} - \max_{h \in \mathcal{H}_m} \text{AUC}_r^{(h)}}{\max_{h \in \mathcal{H}_m} \text{AUC}_r^{(h)}}, \tag{15}$$

where, depending on the context, $\text{AUC}_r^{(h)}$ denotes the area that is underneath either the ROC or PR curve for heuristic h in experimental replication r. Boxplots of these relative areas across all 1000 experimental replications are displayed in the right panels of Figure 6.

5.2.1 Naive Heuristics

As expected given our discussions in Section 3.2 and how we have chosen to index the response classes in our analysis, we see from Figure 4 that the Left heuristic results in significantly higher predicted probabilities of voting "Yes" than the other heuristics, while the Right heuristic yields predicted probabilities of voting "Yes" that are substantially lower. The consequences of this behavior in terms of making classifications can be observed in Figure 5, where we note that both the Left and Right heuristics tend to exhibit a high level of disagreement when compared against any other heuristic's classifications. Moreover, Figure 6 illustrates that the `randomForest R` package's practice of always sending absent levels left in binary classification is noticeably detrimental here—relative to the best performing missing data heuristic, it gives areas underneath the ROC and PR curves that are, on average, 1.5% and 3.7% worse, respectively. And although the Right heuristic appears to do well in terms of the area underneath the PR curve, we once again emphasize the spurious nature of this performance and caution against taking it at face value.

5.2.2 Missing Data Heuristics

Similar to what was previously seen in our 1985 Auto Imports example, Figures 4 and 5 show that the four missing data heuristics tend to exhibit a higher level of agreement with one another than they do with the Left, Right, and One-Hot heuristics. However, significant differences do still exist, and we see from Figure 6 that the relative performances of the heuristics will vary depending on the specific task at hand—the Majority heuristic slightly outperforms the three other missing data heuristics in terms of the area underneath the ROC curve, while the Random heuristic does considerably better than all of its missing data counterparts with respect to the area underneath the PR curve.

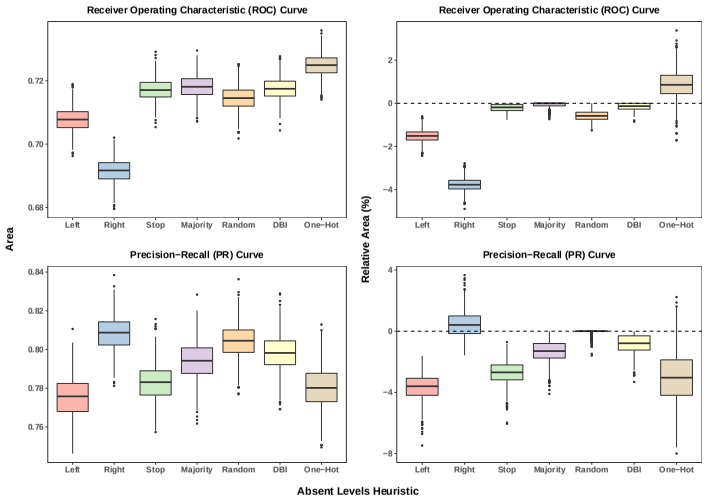


Figure 6: Areas underneath the ROC and PR curves for the seven heuristics in the PROMESA data set. The left panels show boxplots of each heuristic's marginal distribution of areas across all 1000 experimental replications, which ignores the positive correlation that exists between the naive and missing data heuristics. The right panels account for this positive correlation by comparing the areas of the heuristics relative to the best area that was obtained amongst the missing data heuristics within each of the 1000 experimental replications as in (15).

5.2.3 Feature Engineering Heuristic

Although it is essentially uncorrelated with the other six heuristics across all 1000 experimental replications, Figures 4 and 5 still suggest that the One-Hot heuristic's predicted probabilities and classifications can greatly differ from the other six heuristics. Furthermore, Figure 6 shows that even though the One-Hot heuristic may appear to perform well in terms of the area underneath its ROC curve relative to the missing data heuristics, its performance in the PR context is rather lackluster.

5.3 Pittsburgh Bridges

For a multiclass classification example, we consider the Pittsburgh Bridges data set from the UCI Machine Learning Repository which, after removing observations with missing data, contains seven predictors that can be used to classify 72 bridges to one of seven different bridge types. The categorical predictors in this data set for which the absent levels problem can occur include a bridge's river (3 levels), purpose (3 levels), and location (46

levels). Consequently, recall from Section 3.2, that the random forests `FORTRAN` code and `randomForest R` package will both employ an exhaustive search that always sends absent levels right when splitting on either the river or purpose predictors, and that they will both resort to using a random search that sends absent levels either left or right with equal probability when splitting on the location predictor since it has too many levels for an exhaustive search to be computationally efficient. The OOB absence proportions for this example are summarized in the bottom panel of Figure 1 and in Table 1.

Within each experimental replication r, we can use the log loss

$$\text{LogLoss}_r^{(h)} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{k=1}^{K}\left[I(y_n = k)\cdot log\left(\hat{p}_{nkr}^{(h)}\right)\right]$$

to evaluate the overall performance of each heuristic $h \in \mathcal{H}$, where we once again let $\hat{p}_{nkr}^{(h)}$ denote the OOB predicted probability that a heuristic h assigns to an observation n of belonging to a response class k in an experimental replication r. The left panel of Figure 7 displays the marginal distribution of each heuristic's log losses across all 1000 experimental replications. However, to once again account for the positive correlation that exists amongst the naive and missing data heuristics, within every experimental replication r, we also compare the log losses for each heuristic $h \in \mathcal{H}$ relative to the best log loss that was achieved amongst the missing data heuristics $\mathcal{H}_m = \{Stop, Majority, Random, DBI\}$:

$$\text{LogLoss}_r^{(h\|\mathcal{H}_m)} = \frac{\text{LogLoss}_r^{(h)} - \min_{h\in\mathcal{H}_m}\text{LogLoss}_r^{(h)}}{\min_{h\in\mathcal{H}_m}\text{LogLoss}_r^{(h)}}. \tag{16}$$

Boxplots of these relative log losses are depicted in the right panel of Figure 7.

5.3.1 Naive Heuristics

Although we once again stress the systematically biased nature of the Left and Right heuristics, we note from Figure 7 that the two naive heuristics are sometimes able to outperform the missing data heuristics. Nevertheless, on average, the Left and Right heuristics resulted in log losses that are 0.7% and 1.9% worse than the best performing missing data heuristic, respectively.

5.3.2 Missing Data Heuristics

Figure 7 shows that for this particular example, the Majority and Random heuristics perform roughly on par with one another, and that they both also significantly outperform the Stop and DBI heuristics—the smallest log loss amongst all of the missing data heuristics was achieved by either the Majority or the Random heuristic in 999 out of the 1000 experimental replications.

5.3.3 Feature Engineering Heuristic

It can also be observed from Figure 7 that, although the One-Hot heuristic can occasionally outperform the missing data heuristics, on average, it yields a log loss that is 4.5% worse than the best performing missing data heuristic.

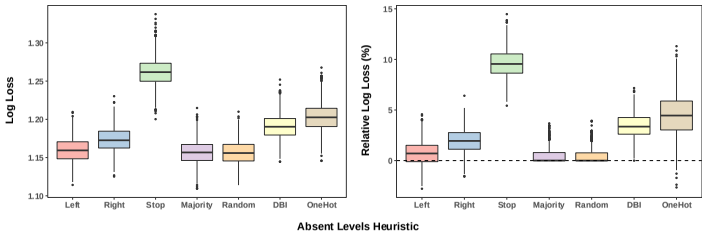


Figure 7: Log losses for the OOB predicted response class probabilities of the seven heuristics in the Pittsburgh Bridges data. The left panel shows boxplots of each heuristic's marginal distribution of log losses set across all 1000 experimental replications, which ignores the positive correlation that exists between the naive and missing data heuristics. The right panel accounts for this positive correlation by comparing the log losses of the heuristics relative to the best log loss that was obtained amongst the missing data heuristics within each of the 1000 experimental replications as in (16).

6. Conclusion

In this paper, we introduced and investigated the absent levels problem for decision tree based methods. In particular, by using Breiman and Cutler's random forests `FORTRAN` code and the `randomForest R` package as motivating case studies, we showed how overlooking the absent levels problem could systematically bias a model. Furthermore, we presented three real data examples which illustrated how absent levels can dramatically alter a model's performance in practice.

Even though a comprehensive theoretical analysis of the absent levels problem was beyond the scope of this paper, we empirically demonstrated how some simple heuristics could be used to help mitigate the effects of absent levels. And although none of the missing data and feature engineering heuristics that we considered performed uniformly better than all of the others, they were all shown to be superior to the biased naive approaches that are currently being employed due to oversights in the software implementations of decision tree based methods.

Consequently, until a more robust theoretical solution is found, we encourage the software implementations which support the native categorical split capabilities of decision trees to incorporate the Random heuristic as a provisional measure given its reliability—in all of our examples, the Random heuristic was always competitive in terms of its performance. Moreover, based on our own personal experiences, we note that the Random heuristic was one of the easier heuristics to implement on top of the `randomForest R` package. In the meantime, while waiting for these mitigations to materialize, we also urge users who rely on decision tree based methods to feature engineer their data sets when possible in order to circumvent the absent levels problem—although our empirical results suggest that this may

sometimes be detrimental to a model's performance, we believe this to still be preferable to the alternative of having to rely on biased approaches which do not adequately address absent levels.

Finally, although this paper primarily focused on the absent levels problem for random forests and a particular subset of the types of analyses in which random forests have been used, it is important to recognize that the issue of absent levels applies much more broadly. For example, decision tree based methods have also been employed for clustering, detecting outliers, imputing missing values, and generating variable importance measures (Breiman, 2003)—tasks which also depend on the terminal node behavior of the observations. In addition, several extensions of decision tree based methods have been built on top of software which currently overlook absent levels—such as the quantile regression forests algorithm (Meinshausen, 2006, 2012) and the infinitesimal jackknife method for estimating the variance of bagged predictors (Wager et al., 2014), which are both implemented on top of the `randomForest R` package. Indeed, given how extensively decision tree based methods have been used, a sizable number of these models have almost surely been significantly and unknowingly affected by the absent levels problem in practice—further emphasizing the need for the development of both theory and software that accounts for this issue.

Acknowledgements

The author is extremely grateful to Art Owen for numerous valuable discussions and insightful comments which substantially improved this paper. The author would also like to thank David Chan, Robert Bell, the action editor, and the anonymous reviewers for their helpful feedback. Finally, the author would like to thank Jim Koehler, Tim Hesterberg, Joseph Kelly, Iván Díaz, Jingang Miao, and Aiyou Chen for many interesting discussions.

References

Yali Amit and Donald Geman. Shape quantization and recognition with randomized trees. *Neural Computation*, 9:1545–1588, 1997.

Gérard Biau. Analysis of a random forests model. *Journal of Machine Learning Research*, 13(1):1063–1095, 2012.

Gérard Biau, Luc Devroye, and Gábor Lugosi. Consistency of random forests and other averaging classifiers. *Journal of Machine Learning Research*, 9(1):2015–2033, 2008.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

Leo Breiman. Out-of-bag estimation. Technical report, Department of Statistics, U.C. Berkeley, 1996b. URL `https://www.stat.berkeley.edu/~breiman/OOBestimation.pdf`.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Leo Breiman. Manual—setting up, using, and understanding random forest v4.0. 2003. URL `https://www.stat.berkeley.edu/~breiman/Using_random_forests_v4.0.pdf`.

Leo Breiman, Jerome. H. Friedman, Richard. A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, 1984. ISBN 0-534-98053-8.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

Jesse Davis and Mark Goadrich. The relationship between precision-recall and ROC curves. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 233–240, 2006.

Misha Denil, David Matheson, and Nando de Freitas. Narrowing the gap: Random forests in theory and in practice. In *Proceedings of the 31th International Conference on Machine Learning*, pages 665–673, 2014.

Walter D. Fisher. On grouping for maximum homogeneity. *Journal of the American Statistical Association*, 53(284):789–798, 1958.

Trevor J Hastie, Robert J. Tibshirani, and Jerome H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition*. Springer Series in Statistics. Springer, New York, 2009.

Tin Kam Ho. The random subspace method for constructing decision forests. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(8):832–844, 1998.

Torsten Hothorn and Achim Zeileis. partykit: A modular toolkit for recursive partytioning in R. *Journal of Machine Learning Research*, 16:3905–3909, 2015.

Torsten Hothorn, Kurt Hornik, and Achim Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical Statistics*, 15 (3):651–674, 2006.

Hemant Ishwaran, Udaya B. Kogalur, Eugene H. Blackstone, and Michael S. Lauer. Random survival forests. *Annals of Applied Statistics*, 2(3):841–860, 2008.

Jeff Lewis. *Rvoteview: Voteview Data in R*, 2015. `https://github.com/JeffreyBLewis/Rvoteview`, `http://voteview.polisci.ucla.edu`, `http://voteview.com`.

Andy Liaw and Matthew Wiener. Classification and regression by randomForest. *R News*, 2(3):18–22, 2002.

Moshe Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Wei-Yin Loh and Nunta Vanichsetakul. Tree-structured classification via generalized discriminant analysis. *Journal of the American Statistical Association*, 83(403):715–725, 1988.

Nolan M. McCarty, Keith T. Poole, and Howard Rosenthal. *Income Redistribution and the Realignment of American Politics*. AEI Press, publisher for the American Enterprise Institute, 1997.

Nicolai Meinshausen. Quantile regression forests. *Journal of Machine Learning Research*, 7:983–999, 2006.

Nicolai Meinshausen. *quantregForest: Quantile Regression Forests*, 2012. URL `http://CRAN.R-project.org/package=quantregForest`. R package version 0.2-3.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

John R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc., 1993.

Greg Ridgeway. *gbm: Generalized Boosted Regression Models*, 2013. URL `http://CRAN.R-project.org/package=gbm`. R package version 2.1.

Brian D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge, 1996. ISBN 0-521-46086-7.

Maytal Saar-Tsechansky and Foster Provost. Handling missing values when applying classification models. *Journal of Machine Learning Research*, 8:1623–1657, 2007.

Terry Therneau, Beth Atkinson, and Brian Ripley. *rpart: Recursive Partitioning and Regression Trees*, 2015. URL `http://CRAN.R-project.org/package=rpart`. R package version 4.1-10.

Stefan Wager, Trevor Hastie, and Bradley Efron. Confidence intervals for random forests: The jackknife and the infinitesimal jackknife. *Journal of Machine Learning Research*, 15 (1):1625–1651, 2014.

Journal of Machine Learning Research 19 (2018) 1-48 Submitted 1/17; Revised 10/18; Published 10/18

On Tight Bounds for the Lasso

Sara van de Geer geer@stat.math.ethz.ch
Seminar for Statistics
ETH Zürich
8092 Zürich, Switzerland

Editor: Francis Bach

Abstract

We present upper and lower bounds for the prediction error of the Lasso. For the case of random Gaussian design, we show that under mild conditions the prediction error of the Lasso is up to smaller order terms dominated by the prediction error of its noiseless counterpart. We then provide exact expressions for the prediction error of the latter, in terms of compatibility constants. Here, we assume the active components of the underlying regression function satisfy some "betamin" condition. For the case of fixed design, we provide upper and lower bounds, again in terms of compatibility constants. As an example, we give an up to a logarithmic term tight bound for the least squares estimator with total variation penalty.

Keywords: Compatibility, Lasso, Linear Model, Lower Bound

1. Introduction

Let $X \in \mathbb{R}^{n\times p}$ be an input matrix and $\beta^0 \in \mathbb{R}^p$ a vector of unknown coefficients. Consider an n-vector of noisy observations

$$Y = X\beta^0 + \epsilon$$

where the noise $\epsilon \in \mathbb{R}^n$ is a vector of i.i.d. standard Gaussians independent of X. The Lasso estimator $\hat{\beta}$ is

$$\hat{\beta} \in \arg\min_{b\in\mathbb{R}^p}\Big\{\|Y - Xb\|_2^2 + 2\lambda\|b\|_1\Big\} \tag{1}$$

with $\lambda > 0$ a regularization parameter (Tibshirani (1996)). Its prediction error is $\|X(\hat{\beta} - \beta^0)\|_2^2$. Main aim of this paper is to provide lower bounds for this prediction error, bounds which show that compatibility constants necessarily enter into the picture.

The results of this paper can be summarized as follows. Firstly, suppose the design is random and that $\Sigma_0 := \mathbb{E}X^TX/n$ exists. Let β^* be the noiseless Lasso for random design

$$\beta^* \in \arg\min_{b\in\mathbb{R}^p}\Big\{n\|\Sigma_0^{1/2}(b - \beta^0)\|_2^2 + 2\lambda\|b\|_1\Big\}. \tag{2}$$

For the case where the rows of X are i.i.d $\mathcal{N}(0, \Sigma_0)$, we compare $\|X(\hat{\beta} - \beta^0)\|_2$ with $\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2$ in Theorem 11. We assume here some mild condition on the growth of the compatibility constants as n increases. The theorem has as an important corollary that $\|X(\hat{\beta} - \beta^0)\|_2$ is up to lower order terms equal to $\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2$ whenever (after normalizing the co-variance matrix Σ_0 to having bounded entries) the largest eigenvalue $\Lambda_{\max}^2$ of Σ_0 is of small order $\log n$, see Corollary 12. Secondly, we provide in Theorem 14 exact expressions for the prediction error of the noiseless Lasso in terms of compatibility constants. We require here "betamin" conditions, which roughly say that the non-zero coefficients of β^0 should have the appropriate signs and remain above the noise level in absolute value. Thirdly, for the case of fixed design, we present upper and lower bounds for the prediction error $\|X(\hat{\beta} - \beta^0)\|_2^2$ in terms of weighted compatibility constants. Theorem 17 states the lower bounds, assuming again certain betamin conditions. The upper bounds we present are similar to those obtained the literature and presented for completeness. They are stated as a consequence of Theorem 18 in Corollary 19. Another application of Theorem 18 is given in Corollary 20. It presents an upper bound for $\|X(\hat{\beta} - \beta^*)\|_2$ where β^* is now the counterpart of (2) for the fixed design case. As an illustration we consider least squares estimation with a (one-dimensional) total variation penalty. For this case we arrive in Corollary 22 at lower and upper bounds that are the same up to a logarithmic term.

There are general upper bounds in the literature, in particular *sharp oracle bounds* as in Koltchinskii et al. (2011) (see also Giraud, 2014, Theorem 4.1 or van de Geer, 2016, Theorem 2.2). The oracle bounds involve a compatibility constant, and an improved version of this constant has been developed in Sun and Zhang (2012), Belloni and Wang (2014) and Dalalyan et al. (2017).

Main theme of this paper is to gain further insight into the role of the compatibility constant when applying the Lasso and to see how it occurs in lower bounds. In Zhang et al. (2014) it is shown that for a given sparsity level, there is a design and a lower bound for the mean prediction error in the noisy case, that holds for any polynomial time algorithm. This lower bound is close to the known upper bounds and in particular shows that compatibility conditions or restricted eigenvalue conditions cannot be avoided. This has also been shown by Bellec (2017), where a choice of the particular vector of regression coefficients β^0 leads to a lower bound matching the upper bound. We further elaborate on this issue, and provide lower bounds that hold for a large class of vectors β^0.

To get an idea of the flavour of the type of bounds we are after, we present in Theorem 1 the case of random design. Details of its proof can be found in Subsection 11.9. We provide more explicit statements in Theorem 11.

Throughout the paper, the active set of β^0 is denoted by $S_0 := \{j : \beta_j^0 \neq 0\}$. Its size is denoted by $s_0 := |S_0|$. Our betamin condition is as follows (its meaning should become more clear after looking at Section 3 where compatibility constants are defined).

Betamin condition *Let*

$$b^* \in \arg\min\Big\{\|\Sigma_0^{1/2}b\|_2 : \sum_{j\in S_0}|b_j| - \sum_{j\notin S_0}|b_j| = 1\Big\}$$

and for $j \in S_0$ let z_j^* be the sign of b_j^*. We say that β^0 satisfies the betamin condition for the noiseless case with random design if

$$z_j^* \beta_j^0 > \frac{z_j^* b_j^*}{\|\Sigma_0^{1/2} b^*\|_2^2} \frac{\lambda}{n}, \ \forall j \in S_0. \tag{3}$$

We will make asymptotic statements with the sample size n tending to infinity and apply (stochastic) order symbols. All quantities in the paper are allowed to depend on n unless otherwise stated.

Theorem 1 *Let the rows of X be i.i.d. $\mathcal{N}(0, \Sigma_0)$, let $\|\Sigma_0\|_\infty$ be the maximal entry in the co-variance matrix Σ_0 and $\Lambda_{\max}^2$ be its largest eigenvalue. For $S \subset \{1, \ldots, p\}$, let $\kappa^2(S)$ be the compatibility constant defined in Definition 2. Suppose that*

$$\Lambda_{\max}^2 / \|\Sigma_0\|_\infty = o(\log(2p)),$$

and

$$\max\left\{ \left(\frac{\|\Sigma_0\|_\infty}{\kappa^2(S)} \right) \frac{\log(2p)|S|}{n} : \ S \subset \{1, \ldots, p\}, \ |S| \leq \left(\frac{\Lambda_{\max}^2}{\kappa^2(S_0)} \right) 4 s_0 \right\} = o(1).$$

For some $t > 0$, take the tuning parameter λ to satisfy

$$3\|\Sigma_0\|_\infty^{1/2} \left(\sqrt{2n(\log(2p) + t)} + 2(\log(2p) + t) \right) \leq \lambda = \mathcal{O}\left(\sqrt{\|\Sigma_0\|_\infty^{1/2} \log(2p)} \right).$$

Then, under condition (3) (the betamin condition for the noiseless case with random design), we have

$$\|X(\hat{\beta} - \beta^0)\|_2^2 = \frac{\lambda^2 / n}{\|\Sigma_0^{1/2} b^*\|_2^2} (1 + o_{\mathbf{P}}(1)) + \mathcal{O}_{\mathbf{P}}(1)$$

(where in fact $s_0 \|\Sigma_0^{1/2} b^*\|_2^2 = \kappa^2(S_0)$).

2. Organization of the Paper

In Section 3 the definition of compatibility constants is given and also some of their properties are discussed. Section 4 shows that for the case of random design the squared "bias" of the Lasso dominates its "variance". Section 5 then gives expressions for this "bias", i.e. for the noiseless Lasso. Here, we examine fixed design but the results carry over immediately to random design. In Section 6 the result of Section 5 is illustrated with the total variation penalty (in one dimension). Section 7 presents lower bounds for the noisy case with fixed design, and Section 8 presents some upper bounds. Corollary 19 is essentially as in the papers Sun and Zhang (2012), Belloni and Wang (2014) and Dalalyan et al. (2017), albeit that do not consider the approximately sparse case to avoid digressions. Section 9 has upper and lower bounds for the least squares estimator with total variation penalty in the noisy case. Section 10 concludes. Section 11 contains the proofs.

3. Compatibility Constants

We introduce some notation in order to be able to define the compatibility constants. This notation will also be helpful at other places. For $S \subset \{1, \ldots, p\}$ and a vector $b \in \mathbb{R}^p$ let $b_S \in \mathbb{R}^p$ be the vector with entries $b_{j,S} := b_j 1\{j \in S\}$, $j = 1, \ldots, p$. We apply the same notation for the $|S|$-dimensional vector $\{b_j\}_{j \in S}$. We moreover write $b_{-S} := b_{S^c}$ where S^c is the the complement of the set S.

3.1. Theoretical Compatibility Constants

The population version of the compatibility constant will be used for the case of random design X. We call the population version the theoretical compatibility constant.

Definition 2 *Let $\Sigma_0 := \mathbb{E} X^T X / n$ (assumed to exist). Let $S \subset \{1, \ldots, p\}$ be a set of indices and $u \geq 0$ be a constant. The theoretical compatibility constant is*

$$\kappa^2(u, S) := \min\left\{ |S| \|\Sigma_0^{1/2} b\|_2^2 : \ \|b_S\|_1 - u\|b_{-S}\|_1 = 1 \right\}.$$

For $u = 1$ we write $\kappa(1, S) =: \kappa(S)$.

3.2. Empirical Compatibility Constants

For a vector w we let $W := \mathrm{diag}(w)$ be the diagonal matrix with w on the diagonal.

Definition 3 *(Belloni and Wang, 2014, Dalalyan et al., 2017) Let $S \subset \{1, \ldots, p\}$ be a set of indices and $w \in \mathbb{R}^{p-|S|}$ be a vector of non-negative weights. The (empirical) compatibility constant is is*

$$\hat{\kappa}^2(w, S) := \min\left\{ |S| \|Xb\|_2^2 / n : \ \|b_S\|_1 - \|W b_{-S}\|_1 = 1 \right\}.$$

For the case where $w = \mathbf{1}$ where $\mathbf{1}$ denotes a vector with all entries equal to one, put $\hat{\kappa}^2(S) := \hat{\kappa}^2(\mathbf{1}, S)$.

3.3. Some Properties of Compatibility Constants

One readily sees that the theoretical and empirical compatibility constants differ only in terms of the matrix used in the quadratic form (which is Σ_0 in the theoretical case and the Gram matrix $\hat{\Sigma} := X^T X / n$ in the empirical case). Thus, when discussing their basic properties it suffices to deal with only one of the two. In this section, we therefore restrict attention to the empirical version $\hat{\kappa}(w, S)$. Note that we have generalized the empirical version as compared to the theoretical one, by considering general weight vectors, not just constant vectors. With some abuse of notation, we write $\hat{\kappa}(u, S) = \hat{\kappa}(u\mathbf{1}, S)$ when the weights are the constant vector $u\mathbf{1}$ (it should be clear from the context what is meant).

The empirical compatibility constant as given in Definition 3 is from Belloni and Wang (2014) or Dalalyan et al. (2017). Another version, from for instance van de Geer (2007) or van de Geer (2016) and its references, is presented in the next definition.

Definition 4 *Let $S \subset \{1,\ldots,p\}$ be a set of indices and $u > 0$ be a constant. The (older) compatibility constant is*

$$\hat{\phi}^2(u,S) := \min\Big\{ |S| \|Xb\|_2^2/n : \ \|b_S\|_1 = 1, \ \|b_{-S}\|_1 \leq 1/u \Big\}.$$

Let $\hat{\phi}^2(S) := \hat{\phi}^2(1,S)$ be the compatibility constant for the case $u = 1$.

The constant $\hat{\phi}(u,S)$ compares, for b's satisfying a "cone condition" $\|b_{-S}\|_1 \leq \|b_S\|_1/u$, the ℓ_2-norm $\|Xb\|_2$ with the ℓ_1-norm $\|b_S\|_1$. The constant $\hat{\kappa}(u,S)$ is similar, but takes in the comparison more advantage of a "cone condition" $\|b_S\|_1 - u\|b_{-S}\|_1 > 0$. When $\hat{\kappa}^2(S) > 0$ the null space property holds (Donoho and Tanner, 2005). We will need throughout that the compatibility constant is strictly positive at S_0 (if it is zero our results cease to be of any interest). This means that we implicitly require throughout

Invertibility condition

$$\textit{The matrix } X_{S_0}^T X_{S_0} \textit{ is invertible.} \tag{4}$$

Here, for any $S \subset \{1,\ldots,p\}$ the matrix $X_S = \{X_j\}_{j\in S}$ is the $n \times |S|$ matrix consisting of the columns of X corresponding to the set S.

The newer version $\hat{\kappa}(u,S)$ is an improvement over $\hat{\phi}(u,S)$ in the sense that $\hat{\kappa}(u,S)$ is the larger of the two.

Lemma 5 *For all $u > 0$ it is true that*

$$\hat{\kappa}^2(u,S) \geq \hat{\phi}^2(u,S).$$

Let now for some $v > 0$

$$b^* \in \arg\min\Big\{ \|Xb\|_2^2/n : \ \|b_S\|_1 - v\|b_{-S}\|_1 = 1 \Big\}.$$

Then by definition

$$\hat{\kappa}^2(v,S) = |S| \|Xb^*\|_2^2/n.$$

The restriction $\|b_S\|_1 - v\|b_{-S}\|_1 = 1$ does not put any bound on the ℓ_1-norm of b_S^*. However, if there is a little room to spare, its ℓ_1-norm is bounded. This will be useful to understand the betamin conditions (conditions (3) and (8)). For simplicity we examine only the value $v = 1$.

Lemma 6 *Let*

$$b^* \in \arg\min\Big\{ \|Xb\|_2^2/n : \ \|b_S\|_1 - \|b_{-S}\|_1 = 1 \Big\}.$$

Then for $0 \leq u < 1$

$$\|b_S^*\|_1 \leq \frac{\hat{\kappa}(S) - u\hat{\kappa}(u,S)}{(1-u)\hat{\kappa}(u,S)}.$$

3.4. Comparing Empirical and Theoretical and Compatibility

Having random quadratic forms in mind, the fact that $\|b_S\|_1 - \|b_{-S}\|_1 = 1$ gives no bound on the ℓ_1-norm can be a problem. Again, if there is a little room to spare in the value of u in the compatibility constant, one *does* get a bound on the ℓ_1-norm. We show this in Lemma 7, and with this tool in hand we lower bound the empirical compatibility constant in terms of the theoretical one in Lemma 8.

Lemma 7 *Let $v > u > 0$. Then*

$$\hat{\kappa}^2(v,S) \geq \min\Big\{ |S| \|Xb\|_2^2/n : \ \|b_S\|_1 - u\|b_{-S}\|_1 = 1, \ \|b\|_1 \leq 1 + (1+u)/(v-u) \Big\}.$$

The following lemma will be applied when bounding the prediction error of $\hat{\beta}$ in terms of that of the noiseless Lasso β^*. The lemma may also be of interest in itself with applications elsewhere.

Lemma 8 *Suppose the rows of X are i.i.d. $\mathcal{N}(0,\Sigma_0)$. Let $\|\Sigma_0\|_\infty$ be the largest entry in the matrix Σ_0. For $v > u$, $(1+u)/(v-u) = \mathcal{O}(1)$ and*

$$\left(\frac{\|\Sigma_0\|_\infty}{\kappa^2(u,S)} \right) \frac{s \log(2p)}{n} = o(1),$$

it is true with probability tending to one that

$$\hat{\kappa}^2(v,S) \geq (1-\eta)^2 \kappa^2(u,S).$$

where $\eta = o(1)$.

4. Comparison With the Noiseless Lasso When the Design is Random

In this section we assume that the rows of X are i.i.d. copies of a Gaussian row vector with mean zero and co-variance matrix Σ_0. We denote the largest eigenvalue of Σ_0 by $\Lambda_{\max}^2$ and let $\|\Sigma_0\|_\infty$ be its largest entry. We define a noiseless version β^* of the Lasso where also the random design is replaced by its population counterpart:

$$\beta^* \in \arg\min_{b\in\mathbb{R}^p} \Big\{ n\|\Sigma_0^{1/2}(b - \beta^0)\|_2^2 + 2\lambda\|b\|_1 \Big\}.$$

The normalization with n is to put things on the scale of the empirical version, as $\mathbb{E}X^T X = n\Sigma_0$. One may think of $\|X(\beta^* - \beta^0)\|_2$ as "bias" and $\|X(\hat{\beta} - \beta^*)\|_2^2$ as "variance". We first investigate in some detail the "variance" part in Theorems 9 and 10. Then we apply the triangle inequality as a way to establish that the squared "bias" dominates the "variance", see Theorem 11.

Theorem 9 *Suppose that*

$$\rho^2 := \max\left\{ \left(\frac{\|\Sigma_0\|_\infty}{\kappa^2(S)} \right) \frac{\log(2p)|S|}{n} : \ S \subset \{1,\ldots,p\}, \ |S| \leq \left(\frac{\Lambda_{\max}^2}{\kappa^2(S_0)} \right) 4s_0 \right\} = o(1).$$

Take for some $t > 0$

$$\lambda \geq 3\|\Sigma_0\|_\infty^{1/2}\Big(\sqrt{2n(\log(2p)+t)} + 2(\log(2p)+t)\Big)$$

and define

$$\gamma := (2\Lambda_{\max})\sqrt{n}/\lambda + (2/\|\Sigma_0\|_\infty^{1/2})\rho\lambda/\sqrt{n\log(2p)}.$$

Then we have for all $x > 0$ with probability at least $1 - 4\exp[-t] - \exp[-x] - o(1)$ that

$$\|X(\hat\beta - \beta^*)\|_2 \leq \gamma\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2 + \sqrt{2x}.$$

Using concentration of measure, one can remove the dependency of the confidence level on the value of t. This value appears in the choice of the tuning parameter λ. We make some rather arbitrary choices for the constants.

Theorem 10 *With the conditions and notations of Theorem 9, and assuming in addition that $4\exp[-t] < 1/8$ (say), for n large enough and for all $x > 0$, with probability at least $1 - 2\exp[-x]$,*

$$\|X(\hat\beta - \beta^*)\|_2 \leq \gamma\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2 + 4\sqrt{\log 2} + \sqrt{2x}.$$

We can now make a type of bias-variance decomposition. The triangle inequality tells us that

$$\Big|\|X(\hat\beta - \beta^0)\|_2 - \|X(\beta^* - \beta^0)\|_2\Big| \leq \|X(\hat\beta - \beta^*)\|_2.$$

We then approximate the empirical "bias" $\|X(\beta^* - \beta^0)\|_2$ by the theoretical "bias" $\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta_0)\|_2$ (which is easy as β^* and β^0 are non-random vectors), and use Theorem 9 or 10 to bound the "variance" $\|X(\hat\beta - \beta^*)\|_2^2$.

Theorem 11 *With the conditions and notations of Theorem 10, we have for n sufficiently large, for all $x > 0$ with probability at least $1 - 2\exp[-x]$*

$$\Big|\|X(\hat\beta - \beta^0)\|_2 - \sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2\Big|$$

$$\leq (\gamma + o(1))\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2 + 4\sqrt{\log 2} + \sqrt{2x}.$$

Corollary 12 *Recall that we defined γ as*

$$\gamma := (2\Lambda_{\max})\sqrt{n}/\lambda + (2/\|\Sigma_0\|_\infty^{1/2})\rho\lambda/\sqrt{n\log(2p)}.$$

Therefore, with the conditions and notations of Theorem 11, and assuming in addition
- $\Lambda_{\max}^2/\|\Sigma_0\|_\infty = o(\log(2p))$,
and
- $\lambda = o(\sqrt{\|\Sigma_0\|_\infty n\log(2p)})/\rho$,
we get with probability at least $1 - 2\exp[-x]$

$$\Big|\|X(\hat\beta - \beta^0)\|_2 - \sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2\Big| = o(\sqrt{n}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2) + 4\sqrt{\log 2} + \sqrt{2x}.$$

In words: the squared "bias" dominates the "variance".

Remark 13 *With the help of Lemma 45, one may also prove bounds for $\sqrt{n}\|\Sigma_0(\hat\beta - \beta^0)\|_2$ to complete those for of $\|X(\hat\beta - \beta^0)\|_2$. We refrain from doing this here to avoid digressions.*

5. The Noiseless Case with Fixed Design

In this section we study fixed design X and the noiseless Lasso

$$\beta^* \in \arg\min_{b\in\mathbb{R}^p}\Big\{\|X(b - \beta^0)\|_2^2 + 2\lambda^*\|b\|_1\Big\}. \tag{5}$$

In principle the noiseless Lasso considered here differs from (2), although one can say that for fixed design $\hat\Sigma = \mathbf{E}\hat\Sigma =: \Sigma_0$, with $\hat\Sigma := X^TX/n$ being the Gram matrix. In what follows in this section, we do not use any specific properties of $\hat\Sigma$ and the theory goes through for any positive semi-definite matrix, Σ say. In the upcoming illustration on functions of bounded variation, the fixed design setup is the natural one.

Note that we supplied the tuning parameter λ^* with a supscript $*$. This is because in Theorem 18 we consider a case with different tuning parameters for the noisy and the noiseless case, say λ and λ^*.

The Karush-Kuhn-Tucker (KKT) conditions for the noiseless Lasso read

$$X^TX(\beta^* - \beta^0) + \lambda^*\zeta^* = 0,\ \zeta^* \in \partial\|\beta^*\|_1, \tag{6}$$

where $\partial\|b\|_1$ denotes the sub-differential of $b \mapsto \|b\|_1$:

$$\partial\|b\|_1 = \Big\{z \in \mathbb{R}^p :\ z^Tb = \|b\|_1,\ \|z\|_\infty \leq 1\Big\}.$$

Recall that

$$\hat\kappa^2(S) = |S|/\|Xb^*\|_2^2/n$$

where

$$b^* \in \arg\min_{b\in\mathbb{R}^p}\Big\{\|Xb\|_2 :\ \|b_S\|_1 - \|b_{-S}\|_1 = 1\Big\}. \tag{7}$$

Note that b^* given in (7) is not unique, for example we can flip the signs of b^* (i.e., replace b^* by $-b^*$).

In Theorem 14 below we give a tight result for the noiseless case under the condition that the active coefficients in β^0 are sufficiently large in absolute value: Condition 8. Here sufficiently large depends on the magnitude of the entries of a solution b^* of (7) with $S = S_0$. Therefore, it is of interest to know how large b^* is. Lemma 6 considers its ℓ_1-norm, and in view of this lemma we conclude that if there is a little room to spare, the ℓ_1-norm of $\|b_S^*\|_1$ is bounded, or - in other words - $\{b_j^*|S|\}_{j\in S}$ is bounded "on average".

For the next condition it is useful to know that we show in Lemma 28 that for b^* given in (7), each coefficient b_j^* with $j \in S$ is nonzero (provided $\hat\kappa^2(S) > 0$).

Betamin condition *Suppose $\hat{\kappa}^2(S_0) > 0$. Let b^* satisfy (7) with $S = S_0$. Denote, for $j \in S_0$, the sign of b_j^* as z_j^*. We say that β^0 satisfies the betamin condition for the noiseless case with fixed design if*

$$z_j^* \beta_j^0 > \frac{z_j^* b_j^* s_0}{\hat{\kappa}^2(S_0)} \frac{\lambda^*}{n} \ \forall \ j \in S_0. \tag{8}$$

Here is the main theorem for the noiseless case.

Theorem 14 *Suppose $\hat{\kappa}^2(S_0) > 0$. Let b^* satisfy (7) with $S = S_0$. If β^0 satisfies condition (8) (the betamin condition for the noiseless case with fixed design), then there exists a solution β^* of the KKT conditions (6) such that*

$$\|X(\beta^* - \beta^0)\|_2^2 = \frac{s_0}{\hat{\kappa}^2(S_0)} \frac{\lambda^{*2}}{n}.$$

6. The Total Variation Penalty in the Noiseless Case

In this section Theorem 14 is illustrated with the total variation penalty. For a vector $f \in \mathbb{R}^n$, its total variation is defined as

$$\mathrm{TV}(f) := \sum_{i=2}^n |f_i - f_{i-1}|.$$

Fix a vector $f^0 \in \mathbb{R}^n$ and let $f^* \in \mathbb{R}^n$ is the least squares approximation of f^0 with total variation penalty:

$$f^* \in \arg\min_{f \in \mathbb{R}^n} \Big\{ \|f - f^0\|_2^2 + 2\lambda^* \mathrm{TV}(f) \Big\}. \tag{9}$$

Theorem 15 presents an explicit expression for the compatibility constant $\hat{\kappa}^2(S_0)$ where S_0 is the set consisting of the locations of the jumps of f^0. Invoking Theorem 14 one then arrives at an explicit expression for $\|f^* - f^0\|_2^2$ provided the jumps of f^0 are sufficiently large, see Corollary 16.

First, we need to rewrite problem (9) as a (noiseless) Lasso problem. Indeed, for $j = 1 \ldots, n$,

$$f_j = \sum_{i=1}^n (f_i - f_{i-1}) 1\{j \geq i\} =: (Xb)_j,$$

where $X_{j,i} = 1\{j \geq i\}$ and $b_i = f_i - f_{i-1}$, with $f_0 := 0$. Hence we can say that $f^0 = X\beta^0$ and $f^* = X\beta^*$ with

$$\beta^* := \arg\min_{b \in \mathbb{R}^n} \Big\{ \|X(b - \beta^0)\|_2^2 + 2\lambda^* \sum_{i=2}^n |b_i| \Big\}.$$

Note that the first coefficient b_1 is not penalized. It is therefore typically active, and we consider the active set as the location of the jumps augmented with the index $\{1\}$. We

slightly adjust the definition of the compatibility constant to deal with the a coefficient without penalty: we set for $S \subset \{2, \ldots, n\}$

$$\kappa^2(S) := \min \Big\{ |S \cup \{1\}| \|Xb\|_2^2 : \|b_S\|_1 - \|b_{-(S \cup \{1\})}\|_1 = 1 \Big\}. \tag{10}$$

Let now $S := \{d_1 + 1, d_1 + d_2 + 1, \ldots, d_1 + \cdots + d_s + 1\}$ for some $\{d_j\}_{j=1}^s \subset \{2, \ldots, n\}$ satisfying $\sum_{j=1}^s d_j + 2 < n$. The set S represents locations of jumps, d_1 is the location of the first jump and $\{d_j\}_{j=2}^s$ are the distances between jumps. Let $d_{s+1} := n - \sum_{j=1}^s d_j$ the distance between the last jump and the end point. For simplicity we assume that d_j is even for all $j \in \{2, \ldots, s\}$.

Theorem 15 *The compatibility constant $\hat{\kappa}^2(S)$ is, up the constant 4 and the scaling by $1/n$, the harmonic mean of of the distances between jumps, including the distance between starting point and first jump and last jump and endpoint:*

$$\hat{\kappa}^2(S) = \frac{s+1}{\frac{n}{d_1} + \sum_{j=2}^s \frac{4n}{d_j} + \frac{n}{d_{s+1}}}.$$

In fact

$$\hat{\kappa}^2(S) = (s+1)\|Xb^*\|_2^2/n$$

where $b_j^* = 0$ for all $j \notin S$ and $b^* = \tilde{b}/\|\tilde{b}\|_1$ with

$$\begin{aligned}
\tilde{b}_{d_1+1} &= \frac{n}{d_1} + \frac{2n}{d_2}, \\
\tilde{b}_{d_2+1} &= -\Big(\frac{2n}{d_2} + \frac{2n}{d_3}\Big), \\
&\vdots \\
\tilde{b}_{d_s} &= (-1)^{s+1}\Big(\frac{2n}{d_s} + \frac{n}{d_{s+1}}\Big).
\end{aligned}$$

Corollary 16 *Suppose f^0 jumps at $S_0 := S = \{d_1 + 1, d_1 + d_2 + 1, \ldots, d_1 + \cdots + d_s + 1\}$, with $s = s_0$. Assume f^0 alternates between jumps up and jumps down. Suppose moreover that*

$$\begin{aligned}
|f^0_{d_1+1} - f^0_{d_1}| &\geq \Big(\frac{n}{d_1} + \frac{2n}{d_2}\Big)\frac{\lambda^*}{n}, \\
|f^0_{d_2+1} - f^0_{d_2}| &\geq \Big(\frac{2n}{d_2} + \frac{2n}{d_3}\Big)\frac{\lambda^*}{n}, \\
&\vdots \\
|f^0_{d_{s_0}+1} - f^0_{d_{s_0}}| &\geq \Big(\frac{2n}{d_{s_0}} + \frac{n}{d_{s_0+1}}\Big)\frac{\lambda^*}{n}.
\end{aligned}$$

Then by Theorem 14 combined with Theorem 15

$$\|f^* - f^0\|_2^2 = \Big(\frac{n}{d_1} + \sum_{j=2}^{s_0} \frac{4n}{d_j} + \frac{n}{d_{s_0+1}}\Big)\frac{\lambda^{*2}}{n}.$$

At this point it may be helpful to look how this normalizes. Say we choose $\lambda^* = \sqrt{n \log n}$. *Suppose* $\max_{1 \leq j \leq s_0+1} n/d_j = \mathcal{O}(s_0+1)$. *Then the jumps of* f^0 *are required to be of order at least* $(s_0+1)\sqrt{\log n/n}$. *We then obtain*

$$\|f^* - f^0\|_2^2 = \mathcal{O}\Big((s_0+1)^2 \log n\Big).$$

7. A Lower Bound in the Noisy Case with Fixed Design

We now turn to the Lasso $\hat{\beta}$ in the noisy case, given by

$$\hat{\beta} \in \arg\min_{b \in \mathbb{R}^p} \Big\{ \|Y - Xb\|_2^2 + 2\lambda \|b\|_1 \Big\}$$

where

$$Y = X\beta^0 + \epsilon.$$

We investigate the case of fixed design X. Recall that we assume throughout i.i.d. standard Gaussian noise.

7.1. Towards Betamin Conditions

Consider some vector $\bar{v} \in \mathbb{R}^{p-s_0}$ with $0 < \bar{v}_j < 1$ for all j. This vector represents the "noise" that is to be overruled by the penalty. Define the collection of weights

$$\mathcal{W}(\bar{v}) := \Big\{ w \in \mathbb{R}^{p-s_0} : \ \ 1 - \bar{v}_j \leq w_j \leq 1 + \bar{v}_j \ \forall \ j \Big\}.$$

Let for $\bar{W} := \mathrm{diag}(1 + \bar{v})$

$$b^*(\bar{v}) \in \arg\min \Big\{ \|Xb\|_2^2 : \ \|b_{S_0}\|_1 - \|\bar{W} b_{-S_0}\|_1 = 1 \Big\}, \ z_j^*(\bar{v}) := \mathrm{sign}(b_j^*(\bar{v})), \ j \in S_0.$$

Then by definition $\hat{\kappa}^2(1+\bar{v}, S_0) = s_0 \|Xb^*(\bar{v})\|_2^2/n$. We remark here that by a slight adjustment of Lemma 28, the assumption $\hat{\kappa}(1+\bar{v}, S_0) > 0$ ensures that $b_j^*(\bar{v}) \neq 0$ for all $j \in S_0$.

For $w \in \mathcal{W}(\bar{v})$ we define the convex problem with linear and convex constraints

$$b(w) \in \arg\min \Big\{ \|Xb\|_2^2 : \ z_{S_0}^{*T}(\bar{v}) b_{S_0} - \|W b_{-S_0}\|_1 \geq 1 \Big\}.$$

Finally, define

$$\mathbf{b}_j(\bar{v}) := \max_{w \in \mathcal{W}(\bar{v})} |b_j(w)| / \|Xb(w)\|_2^2, \ j \in S_0.$$

7.2. Projections

We denote the projection of X_{-S_0} on the space spanned by the columns of X_{S_0} by $X_{-S_0} \mathrm{P} X_{S_0}$. The projection is always defined but as it is implicitly assumed that $X_{S_0}^T X_{S_0}$ is invertible (condition (4)), we can clarify what we mean by projection by writing

$$X_{-S_0} \mathrm{P} X_{S_0} := X_{S_0} (X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T X_{-S_0}.$$

The anti-projection is denoted by

$$X_{-S_0} \mathrm{A} X_{S_0} = X_{-S_0} - X_{-S_0} \mathrm{P} X_{S_0}.$$

We define the matrix

$$\begin{aligned} V_{-S_0,-S_0} &:= \Big(X_{-S_0} \mathrm{A} X_{S_0} \Big)^T \Big(X_{-S_0} \mathrm{A} X_{S_0} \Big) \\ &= X_{-S_0}^T \Big(I - X_{S_0} (X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \Big) X_{-S_0}, \end{aligned}$$

and let $\{v_j^2\}_{j \notin S_0}$ be the diagonal elements of this matrix.

7.3. A Lower Bound

The main result for the noisy case is presented in the next theorem. Here, we use the notations and definitions of the previous two subsections.

Theorem 17 *Take for some* $t > 0$,

$$\lambda > \|v_{-S_0}\|_\infty \sqrt{2(\log(2p) + t)}. \tag{11}$$

Define

$$\bar{v}_j := v_j \sqrt{2(\log(2p) + t)}/\lambda, \ j \notin S_0$$

and

$$\bar{u}_j := u_j \sqrt{2(\log(2p) + t)}/\lambda, \ j \in S_0.$$

where $\{u_j\}_{j \in S_0}$ *are the diagonal elements of the matrix* $(X_{S_0}^T X_{S_0})^{-1}$. *Assume that* $\hat{\kappa}(1 + \bar{v}, S_0) > 0$ *and that the following betamin condition holds:*

$$|\beta_j^0| > \lambda(\mathbf{b}_j(\bar{v}) + \bar{u}_j), \ \mathrm{sign}(\beta_j^0) = z_j^*(\bar{v}) \ \forall j \in S_0.$$

Then for all $x > 0$ *with probability at least* $1 - \exp[-t] - \exp[-x]$ *there is a solution* $\hat{\beta}$ *of the KKT conditions such that*

$$\|X(\hat{\beta} - \beta^0)\|_2 \geq \sqrt{\frac{s_0}{\hat{\kappa}^2(1+\bar{v}, S_0)}} \sqrt{\frac{\lambda^2}{n}} - \sqrt{s_0} - \sqrt{2x}. \tag{12}$$

Note that for $j \in S_0$, the quantity u_j is the variance of the ordinary least squares estimator of β_j^0 for the case S_0 is known. Thus the betamin condition of Theorem 17 needs that the magnitude of the active coefficients should exceed the noise level of the ordinary least squares estimator for known S_0.

8. Comparison with the Noiseless Lasso when the Design is Fixed

This section studies the case of fixed design and compares the noisy Lasso

$$\hat{\beta} := \arg\min_{b \in \mathbb{R}^p} \left\{ \|Y - Xb\|_2^2 + 2\lambda \|b\|_1 \right\}$$

with the noiseless Lasso

$$\beta^* := \arg\min_{b \in \mathbb{R}^p} \left\{ \|X(b - \beta^0)\|_2^2 + 2\lambda^* \|b\|_1 \right\}$$

where $\lambda^* \leq \lambda$. We let S_* be active set of β^* and its cardinality $s_* := |S_*|$. We investigate the error $\|X(\hat{\beta} - \beta^*)\|_2$ in Theorem 18. For $\lambda^* = 0$ we see that $\beta^* = \beta^0$ and then Theorem 18 gives a bound for $\|X(\hat{\beta} - \beta^0)\|_2$. This is elaborated upon in Corollary 19. The case $\lambda^* = \lambda$ is detailed in Corollary 20. The error $\|X(\hat{\beta} - \beta^*)\|_2^2$ can then seen as "variance" and $\|X(\beta^* - \beta^0)\|_2$ as "bias".

8.1. Projections

We now introduce some notations and definitions similar to the ones in Subsections 7.2, now for general S instead of just $S = S_0$. The projection of X_{-S} on the space spanned by the columns of X_S is denoted by $X_{-S}\mathrm{P}X_S$. Recall that such projections are defined, also if X_S does not have full column rank. The anti-projection is

$$X_{-S}\mathrm{A}X_S := X_{-S} - X_{-S}\mathrm{P}X_S.$$

Define the matrix

$$V^S_{-S,-S} := \left(X_{-S}\mathrm{A}X_S \right)^T \left(X_{-S}\mathrm{A}X_S \right)$$

and let $\{(v_j^S)^2\}_{j \notin S}$ be the diagonal elements of this matrix.

8.2. Upper Bound

Recall the KKT conditions for β^* as given in (6), involving the vector ζ^* in the sub-differential $\partial \|\beta^*\|_1$.

Theorem 18 *Fix a set S with cardinality $|S| = s$. Assume that that for some $t > 0$*

$$\lambda > \|v^S_{-S}\|_\infty \sqrt{2(\log(2p) + t)} \tag{13}$$

and write

$$\bar{v}_j^S := v_j^S \sqrt{2(\log(2p) + t)}/\lambda, \ j \notin S. \tag{14}$$

Suppose that

$$\lambda^* |\zeta_j^*|/\lambda < 1 - \bar{v}_j^S \ \forall \ j \notin S.$$

Define

$$\bar{w}_j^S := \frac{1 - \bar{v}_j^S - \lambda^* |\zeta_j^*|/\lambda}{1 - \lambda^*/\lambda}, \ j \notin S.$$

We have for all x with probability at least $1 - \exp[-t] - \exp[-x]$

$$\|X(\hat{\beta} - \beta^*)\|_2 \leq \sqrt{\frac{s}{\hat{\kappa}^2(\bar{w}^S, S)}} \sqrt{\frac{(\lambda - \lambda^*)^2}{n}} + \sqrt{s} + \sqrt{2x}. \tag{15}$$

Corollary 19 *If we take the tuning parameter λ^* of the noiseless Lasso equal to zero, Theorem 18 gives the following: with probability at least $1 - \exp[-t] - \exp[-x]$*

$$\|X(\hat{\beta} - \beta^0)\|_2 \leq \sqrt{s_0/\hat{\kappa}^2(1 - \bar{v}, S_0)} \sqrt{\lambda^2/n} + \sqrt{s_0} + \sqrt{2x}.$$

This result is comparable to results in Sun and Zhang (2012), Belloni and Wang (2014) and Dalalyan et al. (2017), albeit that we do not deal with the extension to the approximately sparse case. One may check that the the combined conclusions of this corollary with that of Theorem 17 also hold with probability at least $1 - \exp[-t] - \exp[-x]$.

Corollary 20 *We can also take $\lambda^* = \lambda$ in Theorem 18. We then formally put $\bar{w}_j^S = \infty$ for all $j \notin S$ and we put $\hat{\kappa}(\bar{w}) = \infty$ as well. Let S with $|S| = s$. Assume that*

$$|\zeta_j^*| < 1 - \bar{v}_j^S \ \forall \ j \notin S \tag{16}$$

(this implies $S \supset S_*$). We have with probability at least $1 - \exp[-t] - \exp[-x]$

$$\|X(\hat{\beta} - \beta^*)\|_2 \leq \sqrt{s} + \sqrt{2x}.$$

This result is as in van de Geer (2016), Problem 2.4.

Corollary 20 is of interest only when $\sqrt{s}$ is small enough This is the case if $\hat{\Sigma} := X^T X/n$ has a well behaved maximal eigenvalue $\hat{\Lambda}^2_{\max}$. Indeed, one can show in the same way as in Lemma 24 (where $\hat{\Sigma}$ is replaced by Σ_0) that

$$s \leq \left(\frac{\hat{\Lambda}^2_{\max}}{(1 - \|\bar{v}^S\|_\infty)^2} \right) \frac{n}{\lambda^2} \|X(\beta^* - \beta^0)\|_2^2.$$

Thus if $\hat{\Lambda}^2_{\max}/(\|\hat{\Sigma}\|_\infty (1 - \|\bar{v}^S\|_\infty)^2) = o(\log(2p))$, then $s = o(\|X(\beta^* - \beta^0)\|_2^2)$. However, for the case of fixed design, one might not want to impose such eigenvalue conditions. Alternatively, one may want to resort to irrepresentable conditions. To this end, fix a set $S \supset S_0$. Let for $j \notin S$, the projection of the j^{th} column X_j on X_S be denoted by

$$X_j \mathrm{P} X_S := X_S \gamma_{S,j}.$$

Then it is not difficult to see that for $j \notin S$ $|\zeta_j^*| \leq \|\gamma_{S,j}\|_1$. In other words, a sufficient condition for (16) to hold is the irrepresentable condition

$$\|\gamma_{S,j}\|_1 \leq 1 - \bar{v}_j^S, \ \forall j \notin S.$$

We conclude that under irrepresentable conditions the squared "bias" $\|X(\beta^* - \beta^0)\|_2^2$ dominates the "variance" $\|X(\hat{\beta} - \beta^*)\|_2^2$.

9. The Total Variation Penalty in the Noisy Case

We continue with the total variation penalty of Section 6, but now in a noisy setting:

$$Y = f^0 + \epsilon,$$

where $f^0 \in \mathbb{R}^n$ is an unknown vector. The least squares estimator with total variation penalty is

$$\hat{f} \in \arg\min_{f \in \mathbb{R}^n} \Big\{ \|Y - f\|_2^2 + 2\lambda \mathrm{TV}(f) \Big\}. \tag{17}$$

As has become clear from the previous sections, to assess the prediction error in the noisy case one needs to evaluate the compatibility constant $\hat{\kappa}(w, S)$ with weights $w_j \neq 1$ for $j \notin S$. For the upper bound on the prediction error, we need lower bounds on $\hat{\kappa}(w, S)$. These are derived in Dalalyan et al. (2017), Proposition 2. We re-derive (and slightly improve) their result using a different proof (the proof in Dalalyan et al., 2017 applies a probabilistic argument).

Suppose as in Section 6 that the locations of the jumps are $S := \{d_1+1, d_1+d_2+1, \ldots, d_1+\cdots+d_s+1\}$ for some $\{d_j\}_{j=1}^s \subset \{2, \ldots, n\}$ satisfying $\sum_{j=1}^s d_j + 2 < n$. Let $d_{s+1} := n - \sum_{j=1}^s d_j$. Assume again for simplicity that d_j is even for all $j \in \{2, \ldots, s\}$.

Lemma 21 *Let $w_1, \ldots, w_n$ be non-negative weights. We have*

$$\frac{\sqrt{s+1}}{\hat{\kappa}(w, S)} \leq \|w\|_\infty \frac{\sqrt{s+1}}{\hat{\kappa}(S)} + \sqrt{n \sum_{i=2}^n (w_i - w_{i-1})^2},$$

where as in Theorem 15

$$\frac{s+1}{\hat{\kappa}^2(S)} = \frac{n}{d_1} + \sum_{j=2}^s \frac{4n}{d_j} + \frac{n}{d_{s+1}}.$$

Corollary 22 *Using the notation of Section 8 suppose that λ satisfies (13) with and let $\bar{v} = \bar{v}^{S_0}$ be given in (14), both with $S := S_0$. Define $\bar{v}_i = 0$ for all $i \in S_0$. We then have with $w_i := 1 - \bar{v}_i$, $j \notin S_0 \cup \{1\}$, $w_1 = w_2$ and $w_i = 1$, $i \in S_0$ that*

$$|w_i - w_{i-1}| \leq |v_i - v_{i-1}| / \|v\|_\infty, \quad i = \{2, \ldots, n\}.$$

In Dalalyan et al. (2017) it is shown in their Proposition 3 that

$$\sum_{i=2}^n (v_i - v_{i-1})^2 / \|v\|_\infty^2 \leq (s_0+1) \log n / n.$$

Hence one obtains from Lemma 21 with $S = S_0$, combined with Corollary 19,

$$\frac{\sqrt{s_0+1}}{\hat{\kappa}(1-\bar{v}, S_0)} \leq \sqrt{\frac{s_0+1}{\hat{\kappa}(S_0)}} + \sqrt{(s_0+1)\log n}$$

where as before

$$\frac{s_0+1}{\hat{\kappa}^2(S_0)} = \frac{n}{d_1} + \sum_{j=2}^{s_0} \frac{4n}{d_j} + \frac{n}{d_{s_0+1}}.$$

Thus, with probability at least $1 - \exp[-t] - \exp[-x]$

$$\|\hat{f} - f^0\|_2 \leq \lambda \Big(\sqrt{\frac{(s_0+1)}{n\hat{\kappa}^2(S_0)}} + \sqrt{\frac{(s_0+1)\log n}{n}} \Big) + \sqrt{s_0} + \sqrt{2x}.$$

Theorem 15 implies that

$$\hat{\kappa}(1+\bar{v}, S_0) \leq \hat{\kappa}(S_0).$$

Recall that for the combined conclusion of Theorem 17 and Corollary 19 we do not have to change the confidence level (which is $1 - \exp[-t] - \exp[-x]$). We therefore obtain that if the jumps of f^0 are sufficiently large in absolute value, as given in Theorem 17, then with probability at least $1 - \exp[-t] - \exp[-x]$

$$\begin{aligned} \lambda \sqrt{\frac{s_0+1}{n\hat{\kappa}^2(S_0)}} - \sqrt{s_0} - \sqrt{2x} \quad &\leq \quad \|\hat{f} - f^0\|_2 \leq \lambda \sqrt{\frac{s_0+1}{n\hat{\kappa}^2(S_0)}} + \sqrt{s_0} + \sqrt{2x} \\ &+ \quad \lambda \sqrt{\frac{(s_0+1)\log n}{n}}. \end{aligned}$$

10. Conclusion

This paper establishes that in a sense the squared "bias" of the Lasso dominates the "variance". Moreover, lower bounds for the prediction error are given. These lower bounds often match up to constants or logarithmic factors the upper bounds, or are in fact tight up to smaller order terms. The bounds show that compatibility constants necessarily enter into the picture. The lower bounds require "betamin" conditions, and - for the case of random design - also certain sparsity conditions. It is as yet unclear what can be said when betamin conditions fail to hold. In combination with this, it would also be of great interest to know what happens when the regression coefficients are not (approximately) sparse. The question to what extent the Lasso will have large prediction error when sparseness assumptions are violated (i.e. when the Lasso is used in a scenario not meant for it) still has some open ends.

11. Proofs

11.1. Proofs of the Lemmas in Section 3

Proof of Lemma 5. We have to show that $\hat{\kappa}^2(u, S) \geq \hat{\phi}^2(u, S)$. Write

$$A := \Big\{ b : \ \|b_{-S}\|_1 \leq \|b_S\|_1 / u, \ \|b_S\|_1 > 0 \Big\}$$

and

$$B := \Big\{ b : \ \|b_S\|_1 - u\|b_{-S}\|_1 > 0 \Big\}.$$

Then

$$B \subset A.$$

Thus

$$\begin{aligned} \hat{\phi}^2(u,S) &= \min\Big\{ \frac{|S|\|Xb\|_2^2/n}{\|b_S\|_1^2} : \ b \in A \Big\} \\ &\leq \min\Big\{ \frac{|S|\|Xb\|_2^2/n}{\|b_S\|_1^2} : \ b \in B \Big\} \\ &= \hat{\kappa}^2(u,S). \end{aligned}$$

□

Proof of Lemma 6. This lemma bounds the ℓ_1-norm of the minimizer b^* if there is a little room to spare. We have

$$\begin{aligned} \|b_S^*\|_1 - u\|b_{-S}^*\|_1 &\leq \sqrt{|S|/n}\|Xb^*\|_2/\hat{\kappa}(u,S) \\ &= \hat{\kappa}(S)/\hat{\kappa}(u,S). \end{aligned}$$

On the other hand

$$\begin{aligned} \|b_S^*\|_1 - u\|b_{-S}^*\|_1 &= \|b_S^*\|_1 - \|b_{-S}^*\|_1 + (1-u)\|b_{-S}^*\|_1 \\ &= 1 + (1-u)\|b_{-S}^*\|_1. \end{aligned}$$

Thus

$$\|b_{-S}^*\|_1 \leq \frac{\hat{\kappa}(S) - \hat{\kappa}(u,S)}{(1-u)\hat{\kappa}(u,S)},$$

yielding

$$\|b_S^*\|_1 = 1 + \|b_{-S}^*\|_1 \leq \frac{\hat{\kappa}(S) - u\hat{\kappa}(u,S)}{(1-u)\hat{\kappa}(u,S)}.$$

□

Proof of Lemma 7. This lemma shows that one has a bound for the ℓ_1-norm in the "cone condition" if there is a little room to spare. Consider a vector $b \in \mathbb{R}^p$ satisfying

$$\|b_S\|_1 - v\|b_{-S}\|_1 = 1.$$

Since

$$\|b_S\|_1 - v\|b_{-S}\|_1 = \|b_S\|_1 - u\|b_{-S}\|_1 - (v-u)\|b_{-S}\|_1$$

we obtain

$$(v-u)\|b_{-S}\|_1 = \|b_S\|_1 - u\|b_{-S}\|_1 - 1 \leq \|b_S\|_1 - u\|b_{-S}\|_1.$$

Moreover, clearly

$$\|b_S\|_1 - u\|b_{-S}\|_1 = (v-u)\|b_{-S}\|_1 + 1 \geq 1.$$

It follows that

$$\min\Big\{ \|Xb\|_2 : \ \|b_S\|_1 - v\|b_{-S}\|_1 = 1 \Big\}$$

$$\geq \min\Big\{ \|Xb\|_2 : \ (v-u)\|b_{-S}\|_1 \leq \|b_S\|_1 - u\|b_{-S}\|_1, \ \|b_S\|_1 - u\|b_{-S}\|_1 \geq 1 \Big\}.$$

Suppose now that for some $c > 1$

$$(v-u)\|b_{-S}\|_1 \leq \|b_S\|_1 - u\|b_{-S}\|_1, \ \|b_S\|_1 - u\|b_{-S}\|_1 = c.$$

Define

$$\tilde{b} := b/c.$$

Then

$$(v-u)\|\tilde{b}_{-S}\|_1 \leq 1, \ \|\tilde{b}_S\|_1 - u\|\tilde{b}_{-S}\|_1 = 1.$$

Moreover

$$\|Xb\|_2 = c\|X\tilde{b}\|_2 > \|X\tilde{b}\|_2.$$

Therefore

$$\begin{aligned} &\min\Big\{ \|Xb\|_2 : \ (v-u)\|b_{-S}\|_1 \leq \|b_S\|_1 - u\|b_{-S}\|_1, \ \|b_S\|_1 - u\|b_{-S}\|_1 \geq 1 \Big\} \\ = \ &\min\Big\{ \|Xb\|_2 : \ (v-u)\|b_{-S}\|_1 \leq 1, \ \|b_S\|_1 - u\|b_{-S}\|_1 = 1 \Big\}. \end{aligned}$$

But if $(v-u)\|b_{-S}\|_1 \leq 1$ and $\|b_S\|_1 - u\|b_{-S}\|_1 = 1$ we see that

$$\begin{aligned} \|b\|_1 &\leq \|b_S\|_1 + \|b_{-S}\|_1 = 1 + (1+u)\|b_{-S}\|_1 \\ &\leq 1 + (1+u)/(v-u). \end{aligned}$$

□

Proof of Lemma 8. This lemma lower bounds the empirical compatibility constant by the theoretical one. Here is a proof. If $\|b_S\|_1 - u\|b_{-S}\|_1 = 1$ we know that

$$1 \leq \|\Sigma_0^{1/2} b\|_2 \sqrt{s}/\kappa(u,S).$$

It therefore follows from Lemma 7 that

$$\hat{\kappa}^2(v,S) \geq \Big\{ |S|\|Xb\|_2^2/n : \ \|b_S\|_1 - u\|b_{-S}\|_1 = 1, \ \|b\|_1 \leq M(u,v)\|\Sigma_0^{1/2} b\|_2 \Big\}$$

where

$$M(u,v) := (1 + (1+u)/(v-u))\sqrt{s}/\kappa(u,S) = o(\sqrt{n/(\|\Sigma_0\|_\infty \log(2p))}).$$

In view of Lemma 45 we know that when $M = o(\sqrt{n/(\|\Sigma_0\|_\infty \log(2p))})$, then with probability tending to one

$$\inf_{\|b\|_1 \leq M\|\Sigma_0^{1/2} b\|_2} \frac{\|Xb\|_2^2/n}{\|\Sigma_0^{1/2} b\|_2^2} \geq (1 - \eta_M)^2$$

for suitable $\eta_M = o(1)$. Hence with probability tending to one

$$\min\Big\{\|Xb\|_2^2/n:\ \|b_S\|_1 - u\|b_{-S}\|_1 = 1,\ \|b\|_1 \le M(u,v)\|\Sigma_0^{1/2}b\|_2\Big\}$$

$$\ge (1-\eta_{M(u,v)})^2 \min\Big\{\|\Sigma_0^{1/2}b\|_2^2:\ \|b_S\|_1 - u\|b_{-S}\|_1 = 1\Big\} = (1-\eta_{M(u,v)})^2\kappa^2(u,S).$$

□

11.2. Proof of Theorem 9

The proof is organized as follows. We first present a bound for $\|\Sigma_0(\beta^* - \beta_0)\|_2$ in Lemma 23. This will be used to bound later the number of active variables s_* of β^*, or rather some extended version of it involving sub-differential calculus, see Lemma 24. We then establish in Lemma 25 a deterministic bound assuming we are on some subset of the underlying probability space. Then in Lemma 26 we show that this subset has large probability.

The noiseless Lasso β^* given in (2) satisfies the KKT conditions

$$n\Sigma_0(\beta^* - \beta^0) + \lambda\zeta^* = 0,\ \zeta^* \in \partial\|\beta^*\|_1, \tag{18}$$

where $\partial\|b\|_1$ is the sub-differential of $b \mapsto \|b\|_1$:

$$\partial\|b\|_1 := \Big\{z:\ \|z\|_\infty \le 1,\ z^T b = \|b\|_1\Big\}.$$

This will be used in Lemma 24 and again in Lemma 25. In the latter we also invoke the KKT conditions for $\hat\beta$

$$X^T X(\hat\beta - \beta^0) + \lambda\hat\zeta = X^T\epsilon,\ \hat\zeta \in \partial\|\hat\beta\|_1. \tag{19}$$

11.2.1. A Bound for the Number of Active Variables of β^*

First we bound the prediction error of β^*.

Lemma 23 *Suppose $\kappa^2(S_0) > 0$. Then*

$$n\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2 \le \frac{s_0}{\kappa^2(S_0)}\frac{\lambda^2}{n}.$$

Proof of Lemma 23. This follows from results in the literature and also from a slight adjustment of Theorem 18 in this paper. Let us present a self-contained proof as well. By the KKT conditions (18)

$$-(\beta^* - \beta^0)^T\zeta^* \le \|\beta^0\|_1 - \|\beta^*\|_1 \le \|\beta^*_{S_0} - \beta^0\|_1 - \|\beta^*_{-S_0}\|_1.$$

So if $\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2 > 0$ we obtain by the definition of the compatibility constant $\kappa^2(S_0)$ that

$$n\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2 \le \lambda\sqrt{s_0}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2/\kappa(S_0).$$

This yields the result of the lemma. □

Consider the set $S_* := \{\beta_j^* \ne 0\}$ of active coefficients of β^*. We bound the size of this set. In fact we look at bound for the size of a potentially larger set, namely the set $S_*(\nu) := \{j : |\zeta_j^*| \ge 1-\nu\}$ where $0 \le \nu < 1$ is arbitrary. Note that indeed $S_* \subset S_*(\nu)$. We pin down the value of ν to $\nu = 1/2$ but the argument goes through for other values if one adjusts the constants accordingly. We still keep the symbol ν at places to facilitate tracking the constants.

Lemma 24 *We have that*

$$|S_*(\nu)| \le \frac{\Lambda_{\max}^2}{(1-\nu)^2}\frac{n^2}{\lambda^2}\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2 \le \frac{\Lambda_{\max}^2}{(1-\nu)^2}\frac{s_0}{\kappa^2(S_0)}.$$

Proof of Lemma 24. Since

$$\|\zeta^*\|_2^2 \ge \|\zeta^*_{S_*(\nu)}\|_2^2 \ge (1-\nu)^2|S_*(\nu)|$$

it follows from the KKT conditions (18) that

$$(1-\nu)^2|S_*(\nu)| \le \|\Sigma_0(\beta^* - \beta^0)\|_2^2\frac{n^2}{\lambda^2} \le \Lambda_{\max}^2\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2\frac{n^2}{\lambda^2}.$$

The proof is completed by applying the upper bound of Lemma 23

$$\|\Sigma_0^{1/2}(\beta^* - \beta^0)\|_2^2 \le \frac{s_0}{\kappa^2(S_0)}\frac{\lambda^2}{n^2}.$$

□

11.2.2. Projections

Let $S := S_*(\nu)$, $s := |S|$ (where $\nu = 1/2$). Set

$$\mathbf{U}(S) := \|\epsilon \mathrm{P} X_S\|_2$$

where $\epsilon \mathrm{P} X_S$ is the projection of ϵ on the space spanned by the columns of X_S. Denote the anti-projection of X_{-S} on this space by

$$X_{-S}\mathrm{A}X_S := X_{-S} - X_{-S}\mathrm{P}X_S.$$

11.2.3. Choice of λ

Recall we take for some $t > 0$

$$\lambda \ge 3\|\Sigma_0\|_\infty^{1/2}\Big(\sqrt{2(\log(2p)+t)} + 2(\log(2p)+t)\Big).$$

11.2.4. The Sets $\mathcal{T}_1$, $\mathcal{T}_2$ and $\mathcal{T}_3$

Write

$$v_0 := \|\Sigma_0\|_\infty^{1/2}\Big(\sqrt{2n(\log(2p)+t)}+2(\log(2p)+t)\Big)/\lambda.$$

We now define a suitable subset of the underlying probability space, on which we can derive the searched for inequality. This subset will be the intersection of the following sets:

$$\begin{aligned}
\mathcal{T}_1 &:= \Big\{\|(X_{-S}\mathrm{A}X_S)^T\epsilon\|_\infty \le \lambda v_0,\ \mathbf{U}(S)\le \sqrt{s}+\sqrt{2x}\Big\},\\
\mathcal{T}_2 &:= \Big\{\ \|(X^TX-n\Sigma_0)(\beta^*-\beta^0)\|_\infty \le \lambda\delta\Big\},\\
\mathcal{T}_3 &:= \Big\{\hat\kappa^2((v-v_0-\delta)/\delta,S)\ge(1-\eta)^2\kappa^2(S)\Big\},
\end{aligned}$$

where $x>0$ is arbitrary, $\delta := \|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2$, and where $\eta\in(0,1)$ is arbitrary. We pin down η to $\eta=1/2$ like we did with ν. We require that $\nu-v_0-2\delta>0$. Since $\nu=1/2$ and $v_0\le 1/3$ this is the case for $\delta\le 1/(12)$. In view of Lemma 23, Theorem 9 is about the case $\delta=o(1)$, so $\delta\le 1/(12)$ will be true for n sufficiently large.

11.2.5. Deterministic Part

Lemma 25 *On $\mathcal{T}_1\cap\mathcal{T}_2\cap\mathcal{T}_3$ it holds that*

$$\|X(\hat\beta-\beta^*)\|_2 \le \left(\frac{\Lambda_{\max}}{(1-\nu)}\frac{\sqrt{n}}{\lambda}+\sqrt{\frac{s}{\kappa^2(S)}}\frac{\lambda}{(1-\eta)n}\right)\sqrt{n}\delta+\sqrt{2x}.$$

Proof of Lemma 25. The KKT conditions (18) and (19), for β^* and $\hat\beta$ respectively, are

$$X^TX(\beta^*-\beta^0)+\lambda\zeta^* = Z,$$

with $Z := (X^TX-n\Sigma_0)(\beta^*-\beta^0)$, and

$$X^TX(\hat\beta-\beta^0)+\lambda\hat\zeta = X^T\epsilon.$$

So subtracting the first from the second

$$X^TX(\hat\beta-\beta^*)+\lambda\hat\zeta-\lambda\zeta^* = X^T\epsilon - Z.$$

Multiplying with $\hat\beta-\beta^*$ yields

$$\|X(\hat\beta-\beta^*)\|_2^2+\lambda(\hat\beta-\beta^*)^T(\hat\zeta-\zeta^*) = (\hat\beta-\beta^*)^T(X^T\epsilon-Z). \tag{20}$$

We write (as in the proof of Theorem 18 ahead) with $S:=S_*(\nu)$, $s:=|S|$,

$$X_S\hat b_S := X_S(\hat\beta_S-\beta_S^*)+(X_{-S}\mathrm{P}X_S)\hat\beta_{-S}.$$

Since $|\zeta_j^*|\le 1-\nu<1$ for all $j\notin S$, it must be true that $\beta^*_{-S}=0$. Therefore

$$X(\hat\beta-\beta^*) = X_S\hat b_S+(X_{-S}\mathrm{A}X_S)\hat\beta_{-S}.$$

So

$$(\hat\beta-\beta^*)^TX^T\epsilon = \hat b_S^TX_S^T\epsilon+\hat\beta_{-S}^T(X_{-S}\mathrm{A}X_S)^T\epsilon.$$

We use that (on $\mathcal{T}_1$)

$$\begin{aligned}
\hat b_S^TX_S^T\epsilon &\le \mathbf{U}(S)\|X_S\hat b_S\|_2\\
&\le \mathbf{U}(S)\|X(\hat\beta-\beta^*)\|_2\\
&\le (\sqrt{s}+\sqrt{2x})\|X(\hat\beta-\beta^*)\|_2
\end{aligned}$$

and

$$\hat\beta_{-S}^T(X_{-S}\mathrm{A}X_S)^T\epsilon \le \|\hat\beta_{-S}\|_1\|(X_{-S}\mathrm{A}X_S)^T\epsilon\|_\infty \le \lambda v_0\|\hat\beta_{-S}\|_1.$$

Moreover (on $\mathcal{T}_2$)

$$-(\hat\beta-\beta^*)^TZ \le \|\hat\beta-\beta^*\|_1\|Z\|_\infty \le \lambda\delta\|\hat\beta-\beta^*\|_1.$$

Then

$$\begin{aligned}
(\hat\beta-\beta^*)^T(\zeta^*-\hat\zeta) &= \hat\beta^T\zeta^*-\beta^{*T}\zeta^*+\beta^{*T}\hat\zeta-\hat\beta^T\hat\zeta\\
&= \hat\beta^T\zeta^*-\|\beta^*\|_1+\beta^{*T}\hat\zeta-\|\hat\beta\|_1\\
&\le \|\hat\beta_S\|_1-\|\beta_S^*\|_1+\|\beta_S^*\|_1-\|\hat\beta_S\|_1\\
&+ \hat\beta_{-S}^T\zeta^*_{-S}-\|\hat\beta_S\|_1\\
&= \hat\beta_{-S}^T\zeta^*_{-S}-\|\hat\beta_S\|_1\\
&\le (1-\nu)\|\hat\beta_{-S}\|_1-\|\hat\beta_{-S}\|_1\\
&= -\nu\|\hat\beta_{-S}\|_1.
\end{aligned}$$

Inserting these bounds in (20) gives

$$\|X(\hat\beta-\beta^*)\|_2^2+\lambda(\nu-v_0-\delta)\|\hat\beta_{-S}\|_1 \le (\sqrt{s}+\sqrt{2x})\|X(\hat\beta-\beta^*)\|_2+\lambda\delta\|\hat\beta_S-\beta_S^*\|_1.$$

If

$$\|X(\hat\beta-\beta^*)\|_2 \le (\sqrt{s}+\sqrt{2x})$$

we are done as by Lemma 24, $\sqrt{s}\le\Lambda_{\max}\delta n/((1-\nu)\lambda)$. If

$$\|X(\hat\beta-\beta^*)\|_2 > (\sqrt{s}+\sqrt{2x})$$

we get

$$(\nu-v_0-\delta)\|\hat\beta_{-S}\|_1 < \delta\|\hat\beta_S-\beta_S^*\|_1$$

or

$$\|\hat\beta_S-\beta^*\|_1-((\nu-v_0-\delta)/\delta)\|\hat\beta_{-S}\|_1 > 0.$$

But (on $\mathcal{T}_3$)

$$\begin{aligned} & \|\hat\beta_S-\beta^*_S\|_1-((\nu-v_0-\delta)/\delta)\|\hat\beta_{-S}\|_1 \\ \leq & \frac{\sqrt{s}\|X(\hat\beta-\beta^*)\|_2}{\sqrt{n}\hat\kappa((\nu-v_0-\delta)/\delta),S)} \\ \leq & \frac{\sqrt{s}\|X(\hat\beta-\beta^*)\|_2}{\sqrt{n}\kappa(S)(1-\eta)}. \end{aligned}$$

This gives

$$\|X(\hat\beta-\beta^*)\|_2\leq\sqrt{s}+\sqrt{2x}+\lambda\delta\sqrt{s}/(\sqrt{n}\kappa(S)(1-\eta)).$$

Again, by Lemma 24, $\sqrt{s}\leq\Lambda_{\max}\delta n/((1-\nu)\lambda)$. We see that

$$\|X(\hat\beta-\beta^*)\|_2\leq\left(\frac{\Lambda_{\max}}{(1-\nu)}\frac{\sqrt{n}}{\lambda}+\frac{\sqrt{s}}{\kappa(S)(1-\eta)}\frac{\lambda}{(1-\eta)n}\right)\sqrt{n}\delta+\sqrt{2x}.$$

□

11.2.6. Random Part

We apply the tools of Section 12.

Lemma 26 *It holds that*

$$\mathbb{P}\left(\mathcal{T}_1\cap\mathcal{T}_2\cap\mathcal{T}_3\right)\geq 1-4\exp[-t]-\exp[-x]-o(1).$$

Proof of Lemma 26 . We first show that $\mathbb{P}(\mathcal{T}_1)\geq 1-2\exp[-t]-\exp[-x]$. One component of this is to show that with probability at least $1-2\exp[-t]$

$$\|(X_{-S}\mathrm{A}X_S)^T\epsilon\|_\infty\leq\lambda v_0.$$

For a square matrix B, let $\mathrm{diag}(B)$ be its diagonal. By Lemma 41 we know that with probability at least $1-\exp[-t]$

$$\|(X_{-S}\mathrm{A}X_S)^T\epsilon\|_\infty\leq\|\mathrm{diag}((X_{-S}\mathrm{A}X_S)^T(X_{-S}\mathrm{A}X_S))\|_\infty^{1/2}\sqrt{2(\log(2p)+t)}.$$

But

$$\|\mathrm{diag}((X_{-S}\mathrm{A}X_S)^T(X_{-S}\mathrm{A}X_S))\|_\infty\leq\|\mathrm{diag}(X^TX)\|_\infty.$$

Moreover in view of Lemma 42, and using the union bound, with probability at least $1-\exp[-t]$

$$\left|\|\mathrm{diag}(X^TX)\|_\infty^{1/2}-\sqrt{n}\|\mathrm{diag}(\Sigma_0)\|_\infty^{1/2}\right|\leq\|\Sigma_0\|_\infty^{1/2}\sqrt{2(\log(2p)+t)}.$$

So with probability at least $1-2\exp[-t]$

$$\|(X_{-S}\mathrm{A}X_S)^T\epsilon\|_\infty\leq\|\Sigma_0\|_\infty^{1/2}\left(\sqrt{2n(\log(2p)+t)}+2(\log(2p)+t)\right)\leq\lambda v_0.$$

The second component is to show that

$$\mathbb{P}(\mathbf{U}(S)\leq\sqrt{s}+\sqrt{2x})\leq\exp[-x],$$

but this follows immediately from Lemma 42.

Next we show that $\mathbb{P}(\mathcal{T}_2)\leq 2\exp[-t]$. Set $Z:=(X^TX-n\Sigma_0)(\beta^*-\beta^0)$. Clearly $X(\beta^*-\beta^0)$ is a Gaussian vector with i.i.d. entries with mean zero and variance $\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2^2$. Hence, applying Lemma 43 with $\sigma_u^2\leq\|\Sigma_0\|_\infty$, $\sigma_v^2=\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2^2$ and using the union bound, we obtain that with probability at least $1-2\exp[-t]$

$$\|Z\|_\infty\leq 3\|\Sigma_0\|_\infty^{1/2}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\left(\sqrt{2n(\log(2p)+t}+\log(2p)+t\right).$$

Finally, the result $\mathbb{P}(\mathcal{T}_3)=1-o(1)$ follows from Lemma 8. □

11.2.7. Collecting the pieces

Combining Lemma 25 with Lemma 26 completes the proof of Theorem 9.

11.3. Proof of Theorems 10 and 11

We use concentration of measure, Lemma 44.

Proof of Theorem 10. Let $m^*:=\mathbb{E}(\|X(\hat\beta-\beta^*)\|_2|X)$. Then we have (by Lemma 44) that with probability at least $1-1/8-3/4-o(1)$

$$\|X(\hat\beta-\beta^*)\|_2\geq m^*-2\sqrt{\log 2}$$

as well as (by Theorem 9),

$$\|X(\hat\beta-\beta^*)\|_2\leq\gamma\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+2\sqrt{\log 2}.$$

Thus

$$m^*\leq\gamma\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+4\sqrt{\log 2}.$$

Applying again Lemma 44 we see that

$$\begin{aligned} & \mathbb{P}\left(\|X(\hat\beta-\beta^*)\|\geq\gamma\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+4\sqrt{\log 2}+\sqrt{2x}\right) \\ & \leq\mathbb{P}\left(\|X(\hat\beta-\beta^*)\|\geq m^*+\sqrt{2x}\right)\leq 2\exp[-x]. \end{aligned}$$

□

Proof of Theorem 11. By the triangle inequality

$$\left|\|X(\hat\beta-\beta^0)\|_2-\|X(\beta^*-\beta^0)\|_2\right|\leq\|X(\hat\beta-\beta^*)\|_2.$$

By Lemma 42, with with probability at least $1-2/n$

$$\left|\|X(\beta^*-\beta^0)\|_2-\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\right| \leq (\sqrt{2\log n})\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2.$$

So, invoking Theoerem 9, with probability at least $1-4\exp[-t]-\exp[-x]-o(1)-2/n$ (subtracting the term $2/n$ to follow the argument, as of course it can be included in the $o(1)$ term)

$$\left|\|X(\hat\beta-\beta^0)\|_2-\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\right| \leq (\gamma+\sqrt{2\log n/n})\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+\sqrt{2x}.$$

Let $m^0:=\mathbb{E}(\|X(\hat\beta-\beta^0)\|_2|X)$. Using the same arguments as in Theorem 10, we arrive at

$$m^0-2\sqrt{\log 2}\leq(1+\gamma+\sqrt{2\log n/n})\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+2\sqrt{\log 2}$$

and

$$(1-\gamma-\sqrt{2\log n/n})\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2-2\sqrt{\log 2}\leq m^0+2\sqrt{\log 2},$$

or

$$\left|m^0-\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\right|\leq\left(\gamma+\sqrt{2\log n/n}\right)\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+4\sqrt{\log 2}.$$

Thus, inserting the triangle inequality,

$$\begin{aligned}&\left|\|X(\hat\beta-\beta^0)\|_2-\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\right|\\ \leq\ &\left|\|X(\hat\beta-\beta^0)\|_2-m^0\right|+\left|m^0-\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2\right|\\ \leq\ &\left|\|X(\hat\beta-\beta^0)\|_2-m^0\right|+(\gamma+\sqrt{2\log n/n})\sqrt{n}\|\Sigma_0^{1/2}(\beta^*-\beta^0)\|_2+4\sqrt{\log 2}.\end{aligned}$$

Apply Lemma 44 again to finalize the result. □

11.4. Proof of Theorem 14

To establish Theorem 14, we first need to study the minimizer b^* in (7). The minimization

$$\min\left\{\|Xb\|_2^2:\ \|b_S\|_1-\|b_{-S}\|_1=1\right\}$$

has non-convex constraints. If we fix the signs within S of a possible solution b, one can reformulate it as a convex problem with convex constraints. This is done in Lemma 27. We then show that $b_j^*\neq 0$ for all $j\in S$ in Lemma 28. This is important because given the signs within S of a potential solution b, we want the restrictions on these signs to be non-active so that the Lagrangian formulation is of a similar form as the KKT conditions (6) for the noiseless Lasso. This Lagrangian form is then given in Lemma 31 with Lemma 30 serving as a preparation. The Lagrangian form of Lemma 31 with $S=S_0$ in a sense resembles the KKT conditions (6) when the active coefficients in the vector β_S^0 have appropriate signs and $|\beta_j^0|$ is for $j\in S_0$ large enough. This allows one to find a solution β^* of the KKT conditions (6) with the prescribed prediction error.

11.4.1. Non-Sparseness within S

Our first step is to ascertain that a solution

$$b^*\in\arg\min_{b\in\mathbb{R}^p}\left\{\|Xb\|_2:\ \|b_S\|_1-\|b_{-S}\|_1=1\right\}$$

can be found by searching over (at most) $2^{|S|}$ convex problems with convex constraints. This is done in the next lemma, where we also show that the equality constraint $\|b_S\|_1-\|b_{-S}\|_1=1$ can be replaced by an inequality constraint $\|b_S\|_1-\|b_{-S}\|_1\geq 1$.

Lemma 27 *We have*

$$\begin{aligned}&\min\left\{\|Xb\|_2^2:\ \|b_S\|_1-\|b_{-S}\|_1=1\right\}\\ =\ &\min\left\{\|Xb\|_2^2:\ \|b_S\|_1-\|b_{-S}\|_1\geq 1\right\}\\ =\ &\min_{z_S\in\{\pm1\}^{|S|}}\min_b\left\{\|Xb\|_2^2:\ z_S^Tb_S-\|b_{-S}\|_1\geq 1,\ z_jb_j\geq 0\ \forall\ j\in S\right\}.\end{aligned}$$

Proof of Lemma 27. To show that the equality constraint can be turned into an inequality constraint let us consider some $b\in\mathbb{R}^p$ for which it holds that $\|b_S\|_1-\|b_{-S}\|_1=c$, where c is a constant bigger than 1. Let $\tilde b:=b/c$. Then

$$\|\tilde b_S\|_1-\|\tilde b_{-S}\|_1=\left(\|b_S\|_1-\|b_{-S}\|_1\right)/c=1.$$

Moreover

$$\|X\tilde b\|_2=\|Xb\|_2/c<\|Xb\|_2.$$

Thus the first equality of the lemma must be true.

We now show the second equality of the lemma. If for some $z_S\in\{\pm1\}$ it holds that $z_jb_j\geq 0$ for all $j\in S$, we have $z_S^Tb_S=\|b_S\|_1$. Conversely, if we define for $j\in S$ with $b_j\neq 0$, $z_j:=b_j/|b_j|$ as the sign of b_j, and define $z_j\in\{\pm1\}$ arbitrarily for $j\in S$ with $b_j=0$, then we have $z_jb_j\geq 0$ for all $j\in S$. Thus

$$\left\{b:\ \|b_S\|_1-\|b_{-S}\|_1\geq 1\right\}=\cup_{z_S\in\{\pm1\}^{|S|}}\left\{b:\ z_S^Tb_S-\|b_{-S}\|_1\geq 1,\ z_jb_j\geq 0\right\}.$$

□

We establish in the next lemma that sign constraints on b_S^* are not active: b_S^* is so to speak maximally non-sparse. We assume that $\hat\kappa^2(S)>0$, so for $S=S_0$ we implicitly assume Condition 4.

Lemma 28 *Suppose that $\hat\kappa(S)\neq 0$. Then for any minimizer b^* of the problem*

$$\min\left\{\|Xb\|_2:\ \|b_S\|_1-\|b_{-S}\|_1=1\right\}$$

it holds that $b_j^*\neq 0$ for all $j\in S$.

Remark 29 *A (very) special case of Lemma 28 is the minimization problem*

$$\mathrm{b}_S^* \in \arg\min\Big\{\|b_S\|_2^2 :\ \|b_S\|_1 = 1\Big\}.$$

Clearly the solution has $|\mathrm{b}_j^*| = 1/|S| \neq 0$ *for all* $j \in S$*. More generally, for the case without "*b_{-S}*-part" one can apply a geometric argument to show that whenever* $X_S^T X_S$ *is non-singular*

$$\mathrm{b}_S^* \in \arg\min\{\|Xb_S\|_2 :\ \|b_S\|_1 = 1\}$$

must have all its components in S *nonzero.*

Proof of Lemma 28. We use the representation of Lemma 27. Let $z_S^* \in \{\pm 1\}^{|S|}$ satisfy $z_S^{*T} b_S^* = \|b_S^*\|_1$ and $z_j^* b_j^* \geq 0$ for all $j \in S$. Then b^* is a solution of the convex minimization problem with (linear and) convex constraints

$$\min\Big\{\|Xb\|_2^2 :\ z_S^{*T} b_S - \|b_{-S}\|_1 \geq 1, z_j^* b_j \geq 0,\ \forall\, j \in S\Big\}.$$

Note that in the minimization, one may replace the inequality constraint $z_S^{*T} b_S - \|b_{-S}\|_1 \geq 1$ by an inequality constraint $z_S^{*T} b_S - \|b_{-S}\|_1 = 1$. This follows from the same arguments as used in the proof of Lemma 27. A reason to replace the equality constraint by an inequality constraint is that the restrictions become convex.

The solution of the convex problem with convex constraints can be found using Lagrange multipliers $\tilde{\lambda}$ and μ_S, where $\tilde{\lambda} \geq 0$ and where μ_S is an $|S|$-vector with non-negative entries. The Lagrangian formulation is

$$\min\Big\{\|Xb\|_2^2 + 2\tilde{\lambda}\Big(\|b_{-S}\|_1 - z_S^{*T} b_S - 1\Big) - 2\sum_{j\in S} \mu_{j,S} z_j^* b_j\Big\}.$$

Because the inequality constraint can be replaced by an equality constraint, we know that in fact $\tilde{\lambda} > 0$. The Lagrangian formulation has has KKT conditions

$$X^T X b^* = \tilde{\lambda} z^* + \mathrm{diag}(\mu_S) z_S^*,$$

where z_{-S}^* is an element of the sub-differential

$$-\partial\|b_{-S}^*\|_1 = \Big\{z_{-S} :\ \|z_{-S}\|_1 \leq 1,\ z_{-S}^T b_{-s}^* = -\|b_{-S}^*\|_1\Big\}.$$

It follows that for $j \in S$

$$b_j^* \neq 0\ \Rightarrow \mu_{j,S} = 0.$$

Let $\mathcal{N} := \{j \in S :\ b_j^* = 0\}$. Then we have by the above argument

$$(X^T X b^*)_{-\mathcal{N}} = \tilde{\lambda} z_{-\mathcal{N}}^*$$

$$(X^T X b^*)_{\mathcal{N}} = \tilde{\lambda} z_{\mathcal{N}}^* + \mathrm{diag}(\mu_{\mathcal{N}}) z_{\mathcal{N}}^*.$$

The tangent plane of $\{b :\ \|Xb\|_2 = \|Xb^*\|_2\}$ at b^* is

$$\mathcal{U} := \{u = b^* + v :\ v^T X^T X b^* = 0\}.$$

The idea of the proof is now to take an element $u = b^* + tv$ in this tangent plane with $t > 0$ and with $v_j \neq 0$ for at least one $j \in \mathcal{N}$ and such that $v_j \neq 0$ has the same sign as b_j^* for all $j \in S\backslash\mathcal{N}$. For $j \notin S$ we take $v_j = 0$. Then $\tilde{b} := b^* + tv$ has $\|\tilde{b}_S\|_1 - \|\tilde{b}_{-S}\|_1 > 1$ and this leads for a suitable scale t to

$$\frac{\|X\tilde{b}\|_2}{\|\tilde{b}_S\|_1 - \|\tilde{b}_{-S}\|_1} < \|Xb^*\|_2.$$

Let us now work out this idea. It cannot be true that $b_j^* = 0$ for all $j \in S$ as $\|b_S^*\|_1 \geq 1$. Hence $S\backslash\mathcal{N} \neq \emptyset$. Take (for example) $v_j = z_j^*$ for all $j \in S\backslash\mathcal{N}$. Then

$$v_{S\backslash\mathcal{N}}^T z_{S\backslash\mathcal{N}}^* = z_{S\backslash\mathcal{N}}^{*T} z_{S\backslash\mathcal{N}}^* = |S\backslash\mathcal{N}|.$$

Now $\tilde{\lambda} > 0$ and the entries of $\mu_{\mathcal{N}}$ are all positive as well (since $\mu_j = 0$ for some $j \in \mathcal{N}$ would imply $b_j^* = 0$ for this j, which is not possible by the definition of $\mathcal{N}$). Therefore we can choose

$$v_{\mathcal{N}}^T(\tilde{\lambda} z_{\mathcal{N}}^* + \mathrm{diag}(\mu_{\mathcal{N}}) z_{\mathcal{N}}^*) = -\tilde{\lambda}|S\backslash\mathcal{N}|.$$

Then at least one entry of $v_{\mathcal{N}}$ has to be non-zero and moreover

$$\begin{aligned} v^T X^T X b^* &= \tilde{\lambda} v_{S\backslash\mathcal{N}}^T z_{S\backslash\mathcal{N}}^* + v_{\mathcal{N}}^T(\tilde{\lambda} z_{\mathcal{N}}^* + \mathrm{diag}(\mu_{\mathcal{N}}) z_{\mathcal{N}}^*) \\ &= \tilde{\lambda}|S\backslash\mathcal{N}| - \tilde{\lambda}|S\backslash\mathcal{N}| \\ &= 0. \end{aligned}$$

We thus have for all $t > 0$

$$\|X(b^* + tv)\|_2^2 = \|Xb^*\|_2^2 + t^2\|Xv\|_2^2.$$

Moreover

$$\begin{aligned} \|b_S^* + tv_S\|_1 &= \|b_{S\backslash\mathcal{N}}^*\|_1 + t\|v_{S\backslash\mathcal{N}}\|_1 + t\|v_{\mathcal{N}}\|_1 \\ &= \|b_S^*\|_1 + t\|v\|_1. \end{aligned}$$

Therefore

$$\begin{aligned} \|b_S^* + tv_S\|_1 - \|b_{-S}^*\|_1 &= \|b_S^*\|_1 - \|b_{-S}^*\|_1 + t\|v\|_1 \\ &= 1 + t\|v\|_1. \end{aligned}$$

It follows that

$$\begin{aligned} &\frac{\|X(b^* + tv)\|_2^2}{(\|b_S^* + tv_S\|_1 - \|b_{-S}^*\|_1)^2} \\ = \ &\frac{\|Xb^*\|_2^2 + t^2\|Xv\|_2^2}{(1 + t\|v\|_1)^2}. \end{aligned}$$

Define

$$\begin{aligned} \mathrm{A} &:= \|Xb^*\|_2^2 + t^2\|Xv\|_2^2 - \|Xb^*\|_2^2(1+t\|v\|_1)^2 \\ &= t^2\|Xv\|_2^2 - 2t\|Xb^*\|_2^2\|v\|_1 - t^2\|Xb^*\|_2^2\|v\|_1^2 \\ &= t^2(\|Xv\|_2^2 - \|Xb^*\|_2^2\|v\|_1^2) - 2t\|Xb^*\|_2^2\|v\|_1^2. \end{aligned}$$

We will show that for suitable $t > 0$ the constant A is strictly negative. This means

$$\|X(b^* + tv)\|_2^2 < \|Xb^*\|_2^2(\|b^*_S + tv_S\|_1 - \|b^*_{-S}\|_1)^2$$

and so we arrive at a contradiction. To show $\mathrm{A} < 0$ we distinguish two cases. If

$$\|Xv\|_2^2 \leq \|Xb^*\|_2^2\|v\|_1^2$$

then $\mathrm{A} < 0$ for all $t > 0$. If

$$\|Xv\|_2^2 > \|Xb^*\|_2^2\|v\|_1^2$$

then $\mathrm{A} < 0$ for all t satisfying

$$0 < t < \frac{2\|Xb^*\|_2^2\|v\|_1^2}{\|Xv\|_2^2 - \|Xb^*\|_2^2\|v\|_1^2}.$$

Here we used the assumption that $\|Xb^*\|_2^2 > 0$ so that the above right hand side is indeed strictly positive. □

11.4.2. Lagrangian Form

We now present the Lagrangian form given the signs within the set S and given that within the set S the solution has non-zero entries. Let for each $z_S \in \{\pm 1\}^{|S|}$

$$b^*(z_S) \in \arg\min\Big\{ \|Xb\|_2^2 : \ z_S^T b_S - \|b_{-S}\|_1 \geq 1, z_j b_j \geq 0, \ \forall\ j \in S \Big\}.$$

Define

$$\mathcal{Z}_S := \Big\{ z_S \in \{-1,1\}^{|S|} : \ z_j b_j^*(z_S) > 0 \ \forall\ j \in S \Big\}.$$

Lemma 30 *We have for all $z_S \in \mathcal{Z}_S$*

$$X^T X b^*(z_S) = z^*(z_S)\|Xb^*(z_S)\|_2^2$$

where $z_S^*(z_S) = z_S$ and $z^*_{-S}(z_S) \in -\partial\|b^*_{-S}(z_S)\|_1$.

Proof of Lemma 30. To prove this result it is useful to repeat some arguments of the proof of Lemma 28. The convex minimization problem with (linear and) convex constraints

$$\min\Big\{ \|Xb\|_2^2 : \ z_S^T b_S - \|b_{-S}\|_1 \geq 1, z_j b_j \geq 0, \ \forall\ j \in S \Big\}$$

can be solved using Lagrange multipliers $\tilde{\lambda}$ and μ_S, where $\tilde{\lambda} > 0$ and μ_S is an $|S|$-vector with non-negative entries. The Lagrangian formulation is

$$\min\Big\{ \|Xb\|_2^2 + 2\tilde{\lambda}\Big(\|b_{-S}\|_1 - z_S^T b_S - 1 \Big) - 2\sum_{j\in S} \mu_{j,S} z_j b_j \Big\}.$$

This has KKT conditions

$$X^T X b^*(z_S) = \tilde{\lambda} z^* + \mathrm{diag}(\mu_S) z_S,$$

where $z_S^* = z_S$ and $z^*_{-S} = z^*_{-S}(z_S)$ depends on z_S and is an element of the sub-differential

$$-\partial\|b^*_{-S}(z_S)\|_1 = \Big\{ z_{-S} : \ \|z_{-S}\|_\infty \leq 1, \ z_{-S}^T b^*_{-S}(z_S) = -\|b^*_{-S}\|_1 \Big\}.$$

It follows that for $j \in S$

$$b_j^*(z_S) \neq 0 \Rightarrow \mu_{j,S} = 0.$$

The assumption that $z_S \in \mathcal{Z}_S$ thus gives $\mu_S = 0$. The KKT conditions then read

$$X^T X b^*(z_S) = \tilde{\lambda} z^*.$$

One sees that

$$1 = z^{*T} b^*(z_S) = b^{*T}(z_S) X^T X b^*(z_S)/\tilde{\lambda} = \|Xb^*(z_S)\|_2^2/\tilde{\lambda}.$$

This gives

$$\tilde{\lambda} = \|Xb^*(z_S)\|_2^2 \ .$$

□

We apply the above lemma with $z_S := \partial\|b_S^*\|_1$. This gives the following result.

Lemma 31 *Suppose $\hat{\kappa}(S) \neq 0$. Let*

$$b^* \in \arg\min\Big\{ \|Xb\|_2^2 : \ \|b_S\|_1 - \|b_{-S}\|_1 = 1 \Big\}$$

Then

$$X^T X b^* = z^* \|Xb^*\|_2^2.$$

where $z_S^* = \partial\|b_S^*\|_1$ and $z^*_{-S} \in -\partial\|b^*_{-S}\|_1$.

Proof of Lemma 31. By Lemma 28, for each

$$b^* \in \arg\min\Big\{ \|Xb\|_2^2 : \ \|b_S\|_1 - \|b_{-S}\|_1 = 1 \Big\}$$

it holds that $b_j^* \neq 0$ for all $j \in S$. We can therefore define $z_j^* := b_j^*/|b_j^*|$ for all $j \in S$ and then $z_S^* = \partial\|b_S^*\|_1 \in \mathcal{Z}_S$. The result now follows from Lemma 30. □

11.4.3. Finalizing the Proof of Theorem 14

With the help of Lemma 31 we are now in the position to prove Theorem 14.

Proof of Theorem 14. Let b^* and z^* be as in Lemma 31, with $S = S_0$. Define

$$\beta' = \beta^0 - \frac{b^* s_0}{\hat{\kappa}^2(S_0)} \frac{\lambda^*}{n}.$$

Then

$$\begin{aligned} X^T X(\beta' - \beta^0) &= -\frac{\lambda^* X^T X b^* s_0}{n\hat{\kappa}^2(S_0)} \\ &= -\frac{\lambda^* X^T X b^*}{\|Xb^*\|_2^2} \\ &= -\lambda^* z^*. \end{aligned}$$

Let $S_* := \{j : \ b_j^* \neq 0\}$. Then by Lemma 28, $S_0 \subset S_*$. Furthermore

$$z_j^* \beta_j' = \begin{cases} z_j^* \beta_j^0 - \lambda z_j^* b_j^* / \|Xb^*\|_2^2 > 0 & j \in S_0 \\ -\lambda^* z_j^* b_j^* / \|Xb^*\|_2^2 > 0 & j \in S_* \backslash S_0 \\ 0 & j \notin S_* \end{cases}.$$

It follows that $z^* \in \partial \|\beta'\|_1$. Thus, $\beta' =: \beta^*$ is a solution of the KKT conditions (6) with $\zeta^* = z^*$. It holds moreover that

$$\|X(\beta^* - \beta^0)\|_2^2 = \frac{\lambda^{*2} \|Xb^*\|_2^2}{\|Xb^*\|_2^4} = \frac{\lambda^{*2} s_0}{n\hat{\kappa}^2(S_0)}.$$

□

11.5. Proof of Theorem 15

The proof of Theorem 15 consists of several steps. First we note that, given the sizes of its jumps, the total variation of a function is the smallest when this function is decreasing or increasing. This is stated in Lemma 32 as a trivial fact. As a consequence, if one subtracts from an arbitrary function value - or minus this value - the total variation, the result will be at most the average of the absolute values. This is shown in Lemma 33. Lemma 33 is then applied at each jump separately, as $\|b_S\|_1 - \|b_{-S\cup\{1\}}\|_1$ in this example amounts to subtracting at each jump some total variation to the left or to the right of this jump. Lemma 34 shows how this works for one jump. Then Theorem 15 is in part proved by applying this lemma to each jump. This leads to a lower bound for $\hat{\kappa}^2(S)$. The proof is completed by showing that this lower bound is achieved by the vector b^* as given in Theorem 15.

For $f \in \mathbb{R}^n$ we define the ordered vector

$$f_{(1)} \leq \cdots \leq f_{(n)},$$

with arbitrary ordering within ties.

Lemma 32 *It holds that*

$$TV(f) \geq f_{(n)} - f_{(1)}$$

with equality if f is increasing or decreasing.

Proof of Lemma 32. Trivial. □

Lemma 33 *It holds for any $j \in \{1, \ldots, n\}$ that*

$$f_j - \mathrm{TV}(f) \leq f_{(1)} \leq \frac{1}{n} \sum_{i=1}^n |f_i|,$$

and

$$-f_j - \mathrm{TV}(f) \leq -f_{(n)} \leq \frac{1}{n} \sum_{i=1}^n |f_i|.$$

Proof of Lemma 33. We have from Lemma 32 that $\mathrm{TV}(f) \geq f_{(n)} - f_{(1)}$. Moreover, $f_j \leq f_{(n)}$. Thus

$$\begin{aligned} f_j - \mathrm{TV}(f) &\leq f_j - (f_{(n)} - f_{(1)}) \\ &\leq f_{(n)} - (f_{(n)} - f_{(1)}) \\ &= f_{(1)}. \end{aligned}$$

Case 1: if $f_{(1)} < 0$ obviously $f_{(1)} < \frac{1}{n} \sum_{i=1}^n |f_i|$.
Case 2: if $f_{(1)} \geq 0$ then $f_i \geq 0$ for all i and then

$$f_{(1)} \leq \sum_{i=1}^n f_i/n = \sum_{i=1}^n |f_i|/n.$$

In the same way

$$\begin{aligned} -f_j - \mathrm{TV}(f) &\leq -f_j - (f_{(n)} - f_{(1)}) \\ &\leq -f_{(1)} - (f_{(n)} - f_{(1)}) \\ &= -f_{(n)}. \end{aligned}$$

Case 1: if $f_{(n)} > 0$ then $-f_{(n)} < \frac{1}{n} \sum_{i=1}^n |f_i|$.
Case 2: if $f_{(n)} \leq 0$ then $f_i \leq 0$ for all i and then

$$-f_{(n)} \leq -\sum_{i=1}^n f_i/n = \sum_{i=1}^n |f_i|/n.$$

□

Lemma 34 *Let $f \in \mathbb{R}^n$ with total variation $\mathrm{TV}(f) = \sum_{i=2}^n |f_i - f_{i-1}|$ and $g \in \mathbb{R}^m$ with total variation $\mathrm{TV}(g) = \sum_{i=2}^m |g_i - g_{i-1}|$. Then for any $j \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, m\}$*

$$|f_j - g_k| - \mathrm{TV}(f) - \mathrm{TV}(g) \leq \frac{1}{n} \sum_{i=1}^n |f_i| + \frac{1}{m} \sum_{i=1}^m |g_i|.$$

Proof of Lemma 34. Suppose without loss of generality that $f_j \geq g_k$. Then by Lemma 33

$$\begin{aligned} |f_j - g_k| - \mathrm{TV}(f) - \mathrm{TV}(g) &= \underbrace{(f_j - \mathrm{TV}(f))}_{\leq \sum_{i=1}^n |f_i|/n} + \underbrace{(-g_k - \mathrm{TV}(g))}_{\leq \sum_{i=1}^m |g_i|/m} \\ &\leq \frac{1}{n}\sum_{i=1}^n |f_i| + \frac{1}{m}\sum_{i=1}^m |g_i|. \end{aligned}$$

□

Proof of Theorem 15. Let for $j = 2, \ldots, s$, $u_j \in \mathbb{N}$ satisfy $1 \leq u_j \leq d_j - 1$. We may write for $f = Xb$,

$$\begin{aligned} & \|b_S\|_1 - \|b_{-(S\cup\{1\})}\|_1 \\ = \; & |f_{d_1+1} - f_{d_1}| - \sum_{i=2}^{d_1} |f_i - f_{i-1}| - \sum_{i=d_1+2}^{d_1+u_2} |f_i - f_{i-1}| \\ + \; & |f_{d_1+d_2+1} - f_{d_1+d_2}| - \sum_{i=d_1+u_2+1}^{d_1+d_2} |f_i - f_{i-1}| - \sum_{i=d_1+d_2+2}^{d_1+d_2+u_3} |f_i - f_{i-1}| \\ & \ldots \\ + \; & |f_{d_1+\cdots+d_{s-1}+1} - f_{d_1+\cdots+d_{s-1}}| \\ - \; & \sum_{i=d_1+\cdots+d_{s-2}+u_{s-1}+1}^{d_1+\cdots+d_{s-1}} |f_i - f_{i-1}| - \sum_{i=d_1+\cdots+d_{s-1}+2}^{d_1+\cdots+d_{s-1}+u_s} |f_i - f_{i-1}| \\ + \; & |f_{d_1+\cdots+d_s+1} - f_{d_1+\cdots+d_s}| \\ - \; & \sum_{i=d_1+\cdots+d_{s-1}+u_s+1}^{d_1+\cdots+d_s} |f_i - f_{i-1}| - \sum_{i=d_1+\cdots+d_s+2}^{n} |f_i - f_{i-1}| \end{aligned}$$

$$\begin{aligned} \leq \; & \frac{1}{d_1}\sum_{i=1}^{d_1} |f_i| + \frac{1}{u_2}\sum_{i=d_1+1}^{d_1+u_2} |f_i| \\ + \; & \frac{1}{d_2-u_2}\sum_{i=d_1+u_2+1}^{d_1+d_2} |f_i| + \frac{1}{u_3}\sum_{i=d_1+d_2+1}^{d_1+d_2+u_3} |f_i| \\ & \ldots \\ + \; & \frac{1}{d_{s-1}-u_{s-1}}\sum_{i=d_1+\cdots+d_{s-2}+u_{s-1}+1}^{d_1+\cdots+d_{s-1}} |f_i| + \frac{1}{u_s}\sum_{i=d_1+\cdots+d_{s-1}+1}^{d_1+\cdots+d_{s-1}+u_s} |f_i| \\ + \; & \frac{1}{d_s-u_s}\sum_{i=d_1+\cdots+d_{s-1}+u_s+1}^{d_1+\cdots+d_s} |f_i| + \frac{1}{d_{s+1}}\sum_{i=d_1+\cdots+d_s+1}^{n} |f_i| \end{aligned}$$

$$\begin{aligned} \leq \; & \sqrt{\frac{1}{d_1} + \frac{1}{u_2} + \frac{1}{d_2-u_2} + \cdots + \frac{1}{d_{s-1}-u_{s-1}} + \frac{1}{u_s} + \frac{1}{d_s-u_s} + \frac{1}{d_{s+1}}} \\ & \times \sqrt{\sum_{i=1}^n |f_i|^2}, \end{aligned}$$

where in the first inequality we applied Lemma 34 and the second one follows from the Cauchy-Schwarz inequality. The assumption that for all $j \in \{2, \ldots, s\}$ d_j is even allows us to take $u_j = d_j/2$ to arrive at

$$\kappa^2(S) \geq \frac{s+1}{\frac{n}{d_1} + \sum_{j=2}^s \frac{4n}{d_j} + \frac{n}{d_{s+1}}}.$$

Now for the reverse inequality, let $\tilde{b}$ be given as in the theorem and and $\tilde{f} := X\tilde{b}$. Then $\tilde{f}$ is equal to

$$\tilde{f}_i = \begin{cases} -\frac{n}{d_1} & i = 1, \ldots, d_1 \\ \frac{2n}{d_2} & i = d_1+1, \ldots, d_1+d_2 \\ \vdots & \\ (-1)^s \frac{2n}{d_s} & i = \sum_{j=1}^{s-1} d_j + 1, \ldots, \sum_{j=1}^s d_j \\ (-1)^{s+1} \frac{n}{d_{s+1}} & i = \sum_{j=1}^s d_j + 1, \ldots, n \end{cases}.$$

By the definition of $\tilde{f} = X\tilde{b}$,

$$\begin{aligned} \|\tilde{b}_S\|_1 = \sum_{j=1}^s |\tilde{f}_{d_j+1} - \tilde{f}_{d_j}| &= \frac{n}{d_1} + \frac{2n}{d_2} \\ &+ \frac{2n}{d_2} + \frac{2n}{d_3} \\ &\vdots \\ &+ \frac{2n}{d_{s-1}} + \frac{2n}{d_s} \\ &+ \frac{2n}{d_s} + \frac{n}{d_{s+1}} \\ &= \frac{n}{d_1} + \sum_{j=2}^s \frac{4n}{d_j} + \frac{n}{d_{s+1}}, \end{aligned}$$

and also

$$\begin{aligned} \sum_{i=1}^n \tilde{f}_i^2 &= d_1\tilde{f}_{t_1}^2 + \cdots + d_{s+1}\tilde{f}_{d_{s+1}}^2 \\ &= \frac{n^2}{d_1} + 4\sum_{j=2}^s \frac{n^2}{d_j} + \frac{n^2}{d_{s+1}}. \end{aligned}$$

Note also that

$$\begin{aligned} & \|\tilde{b}_{-(S\cup\{1\})}\|_1 \\ = & \sum_{i=2}^{d_1}|\tilde{f}_i-\tilde{f}_{i-1}|+\sum_{i=d_1+2}^{d_2}|\tilde{f}_i-\tilde{f}_{i-1}|+\cdots+\sum_{i=d_1+\cdots+d_s+2}^{n}|\tilde{f}_i-\tilde{f}_{i-1}| \\ = & 0 \end{aligned}$$

It follows that

$$\begin{aligned} \frac{(s+1)\|X\tilde{b}\|_2^2/n}{(\|\tilde{b}_S\|_1-\|\tilde{b}_{-(S\cup\{1\})}\|_1)^2} &= \frac{\sum_{i=1}^n \tilde{f}_i^2/n}{\left(\sum_{j=1}^s|\tilde{f}_{d_j+1}-\tilde{f}_{d_j}|\right)^2} \\ &= \frac{s+1}{\frac{n}{d_1}+\sum_{j=2}^s\frac{4n}{d_j}+\frac{n}{d_{s+1}}}. \end{aligned}$$

□

11.6. Proof of Theorem 17

To prove Theorem 17, we first establish the Lagrangian form of the minimization problem where we have the convex constraint $z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1\geq 1$. Then we recall the projections and we introduce a subset $\mathcal{T}$ of the underlying probability space where the lower bound of Theorem 17 holds. The latter is shown in Lemma 36. Finally, we show that the subset $\mathcal{T}$ has large probability.

11.6.1. Lagrangian Form

Recall for $w\in\mathcal{W}(\bar{v})$ the convex problem with linear and convex constraints

$$b(w)\in\arg\min\left\{\|Xb\|_2^2:\ z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1\geq 1\right\}.$$

Note that here we do not require the positivity constraint $z_j^{*T}(\bar{v})b_j\geq 0$ for all $j\in S_0$. The next lemma gives its Lagrangian form. This form plays in the proof of Theorem 17 the same role as in the proof of Theorem 14 for the noiseless version. We also show that for $w\in\mathcal{W}(\bar{v})$ the minimum $\|Xb(w)\|_2^2$ is not larger than $\|Xb^*(\bar{v})\|_2^2$ (recall that by definition $\hat{\kappa}^2(1+\bar{v},S_0)=s_0\|Xb^*(\bar{v})\|_2^2/n$).

Lemma 35 *We have*

$$X^TXb(w)=\|Xb(w)\|_2^2Wz(w),$$

with

$$z_{S_0}(w)=z_{S_0}^*(\bar{v}),\ z_{-S_0}(w)\in-\partial\|b_{-S_0}(w)\|_1.$$

Moreover, for $w\in\mathcal{W}(\bar{v})$

$$s_0\|Xb(w)\|_2^2/n\leq\hat{\kappa}^2(1+\bar{v},S_0).$$

Proof of Lemma 35. The problem

$$\min\left\{\|Xb\|_2^2:\ z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1\geq 1\right\}$$

has Lagrangian

$$X^TXb(w)=\tilde{\lambda}Wz(w)$$

with $z_{S_0}(w)=z_{S_0}^*(\bar{v})$ and $z_{-S_0}(w)\in-\partial\|b_{-S_0}(w)\|_1$. Moreover

$$\|Xb(w)\|_2^2=\tilde{\lambda}b(w)^TWz(w)=z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1=1$$

because the minimum is reached at the boundary. So

$$\tilde{\lambda}=\|Xb(w)\|_2^2.$$

To obtain the second statement of the lemma, we use similar arguments as in the proof of Lemma 5. We have

$$\|Xb(w)\|_2=\min_{b\in\mathbb{R}^p}\left\{\frac{\|Xb\|_2}{z_{S_0}^{*T}(\bar{v})b_{S_0}-\|W_{-S_0}b_{-S_0}\|_1}:\ z_{S_0}^{*T}(\bar{v})b_{S_0}-\|W_{-S_0}b_{-S_0}\|_1>0\right\}$$

But for $w\in\mathcal{W}$ and $\bar{w}:=1+\bar{v}$, we know

$$\|Wb_{-S_0}\|_1\leq\|\bar{W}b_{-S_0}\|_1$$

and so

$$z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1>z_{S_0}^{*T}(\bar{v})b_{S_0}-\|\bar{W}b_{-S_0}\|_1.$$

Let

$$A:=\left\{b:\ z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1>0\right\}$$

and

$$B:=\left\{b:\ z_{S_0}^{*T}(\bar{v})b_{S_0}-\|\bar{W}b_{-S_0}\|_1>0\right\}.$$

Then $B\subset A$. Hence

$$\begin{aligned} \|Xb(w)\|_2 &= \min_{b\in A}\frac{\|Xb\|_2}{z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1} \\ &\leq \min_{b\in B}\frac{\|Xb\|_2}{z_{S_0}^{*T}(\bar{v})b_{S_0}-\|Wb_{-S_0}\|_1} \\ &\leq \min_{b\in B}\frac{\|Xb\|_2}{z_{S_0}^{*T}(\bar{v})b_{S_0}-\|\bar{W}b_{-S_0}\|_1} \\ &= \|Xb^*(\bar{v})\|_2 \\ &= \frac{\sqrt{n}\hat{\kappa}(1+\bar{v},S_0)}{\sqrt{s_0}}. \end{aligned}$$

□

11.6.2. Projections

Recall the notation of Subsection 7.2 and that moreover the diagonal elements of the matrix $(X_{S_0}^T X_{S_0})^{-1}$ are denoted by $\{u_j^2\}_{j\in S_0}$. We write

$$\hat{u}_{S_0} := (X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon.$$

We denote the projection of ϵ on the space spanned by the columns of X_{S_0} by

$$\epsilon \mathrm{P} X_{S_0} := X_{S_0}(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon = X_{S_0}\hat{u}_{S_0}$$

and write

$$\mathbf{U}(S_0) := \|\epsilon \mathrm{P} X_{S_0}\|_2.$$

11.6.3. Choice of λ

Recall that we require that for some $t > 0$

$$\lambda > \|v_{-S_0}\|_\infty \sqrt{2(\log(2p) + t)}.$$

11.6.4. The Set $\mathcal{T}$

Recall

$$\bar{u}_j := u_j\sqrt{2(\log(2p)+t)}/\lambda,\ j \in S_0,\ \bar{v}_j := v_j\sqrt{2(\log(2p)+t)}/\lambda,\ j \notin S_0. \tag{21}$$

Let $\mathcal{T}$ be the set

$$\begin{aligned}\mathcal{T} &:= \left\{|\hat{u}_j| \le \lambda \bar{u}_j\ \forall j \in S_0\right\} \\ &\cap \left\{|\hat{v}_j| \le \lambda \bar{v}_j\ \forall j \notin S_0\right\} \cap \left\{\mathbf{U}(S_0) \le \sqrt{s_0} + \sqrt{2x}\right\}.\end{aligned}$$

We show in Subsection 11.6.6 that $\mathbb{P}(\mathcal{T}) \ge 1 - \exp[-t] - \exp[-x]$.

11.6.5. Deterministic Part

The idea is now to incorporate the noisy part of the KKT conditions for the noisy Lasso into a weighted sub-differential, creating in that way KKT conditions of the same for as the noiseless KKT conditions (see (22) in the proof). To do so, we first put part of the noise in the vector β^0 without adding additional non-zeros. This makes it possible not to change the sub-differential at S_0. The rewriting of the KKT conditions make them resemble the Lagrangian form of Lemma 35.

We will use the KKT conditions (19) for $\hat{\beta}$:

$$-X^T(Y - X\hat{\beta}) = -\lambda\hat{\zeta},\ \hat{\zeta} \in \partial\|\hat{\beta}\|_1.$$

Lemma 36 *Suppose we are on the set $\mathcal{T}$ defined in Subsection 11.6.4. Then under the conditions of Theorem 17*

$$\|X(\hat{\beta} - \beta^0)\|_n \ge \frac{\lambda\sqrt{s_0}}{\sqrt{n}\hat{\kappa}(1+\bar{v}, S)} + \sqrt{2x}$$

Proof of Lemma 36. Set

$$\hat{\beta}_{S_0}^0 := \beta^0 + \hat{u}_{S_0},\ \hat{\beta}_{-S_0}^0 := 0.$$

Then

$$\begin{aligned}Y &= X\beta^0 + \epsilon \\ &= X_{S_0}\beta_{S_0}^0 + X_{S_0}\hat{u}_{S_0} + \epsilon \mathrm{A} X_{S_0} \\ &= X\hat{\beta}^0 + \epsilon \mathrm{A} X_{S_0}.\end{aligned}$$

The KKT conditions (19) are

$$-X^T(Y - X\hat{\beta}) = -\lambda\hat{\zeta}.$$

We have

$$Y - X\hat{\beta} = -X(\hat{\beta} - \hat{\beta}^0) - \epsilon \mathrm{A} X_{S_0}.$$

Therefore

$$-X^T(Y - X\hat{\beta}) = X^T X(\hat{\beta} - \hat{\beta}^0) - X^T(\epsilon \mathrm{A} X_{S_0}).$$

But

$$X_{S_0}^T(\epsilon \mathrm{A} X_{S_0}) = 0,$$

and

$$\begin{aligned}X_{-S_0}^T(\epsilon \mathrm{A} X_{S_0}) &= X_{-S_0}^T - X_{-S_0}^T X_{S_0}(X_{S_0}^T X_{S_0})^{-1} X_{S_0}^T \epsilon \\ &= (X_{-S_0} \mathrm{A} X_{S_0})^T \epsilon.\end{aligned}$$

Hence the KKT conditions read

$$X^T X(\hat{\beta} - \hat{\beta}^0) = -\lambda\hat{\zeta} + \hat{v},$$

where

$$\hat{v}_{S_0} = 0,\ \hat{v}_{-S_0} = (X_{-S_0} \mathrm{A} X_{S_0})^T \epsilon.$$

Set $\hat{S} := \{j :\ \hat{\beta}_j \ne 0\}$ and define for all $j \in \hat{S}\backslash S_0$

$$\hat{w}_j := 1 + \hat{v}_j/(\lambda\hat{\zeta}_j).$$

By assumption (since we are on $\mathcal{T}$) $|\hat{v}_j| < \lambda\bar{v}_j$. so $\hat{w}_j \ge 1 - \bar{v}_j$ for all $j \in \hat{S}\backslash S_0$. For $j \notin \hat{S} \cup S_0$ we define

$$\hat{w}_j := \max\{|1 + \hat{v}_j/\lambda|, 1 - \bar{v}_j\}.$$

Then for $j \notin \hat{S} \cup S_0$

$$\begin{aligned}\lambda\hat{\zeta}_j + \hat{v}_j &= \lambda|\hat{\zeta}_j + \hat{v}_j/\lambda|\text{sign}(\hat{\zeta}_j + \hat{v}_j/\lambda) \\ &= \begin{cases}\hat{w}_j\text{sign}(\hat{\zeta}_j + \hat{v}_j/\lambda), & |\hat{\zeta}_j + \hat{v}_j/\lambda| \geq 1 - \bar{v}_j \\ \hat{w}_j\frac{|\hat{\zeta}_j+\hat{v}_j/\lambda|}{1-\bar{v}_j}\text{sign}(\hat{\zeta}_j + \hat{v}_j/\lambda) & |\hat{\zeta}_j + \hat{v}_j/\lambda| \leq 1 - \bar{v}_j\end{cases} \\ &= \hat{w}_j\tilde{\zeta}_j,\end{aligned}$$

where

$$\tilde{\zeta}_j := \begin{cases}\text{sign}(\hat{\zeta}_j + \hat{v}_j/\lambda), & |\hat{\zeta}_j + \hat{v}_j/\lambda| \geq 1 - \bar{v}_j \\ \frac{|\hat{\zeta}_j+\hat{v}_j/\lambda|}{1-\bar{v}_j}\text{sign}(\hat{\zeta}_j + \hat{v}_j/\lambda) & |\hat{\zeta}_j + \hat{v}_j/\lambda| \leq 1 - \bar{v}_j\end{cases}.$$

One readily verifies that (on $\mathcal{T}$) $\hat{w}_j \leq 1 + \bar{v}_j$ for all $j \notin S_0$. Taking $\tilde{\zeta}_j = \hat{\zeta}_j$ for $j \in S \cup S_0$ we arrive at the KKT conditions

$$X^T X(\hat{\beta} - \hat{\beta}^0) = -\lambda\hat{W}\tilde{\zeta},\ \tilde{\zeta} \in \partial\|\hat{\beta}\|_1 \tag{22}$$

and where $\hat{W} = \text{diag}(\hat{w})$ with $\hat{w} \in \mathcal{W}(\bar{v})$. Let now $S_0^+ := \{j \in S_0 :\ z_j^*(\bar{v})b_j(\hat{w}) > 0\}$ and $S_0^- := \{j \in S_0 :\ z_j^*(\bar{v})b_j(\hat{w}) \leq 0\}$. Take

$$\beta' = \hat{\beta}^0 - \lambda b_j(\hat{w})/\|Xb(\hat{w})\|_2^2.$$

Case 1 Let $j \in S_0$. By the condition on β^0 we know that $|\beta_j^0| > \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2 + |\hat{u}_{S_0}|$, so $|\hat{\beta}_j^0| \geq |\beta_j^0| - |\hat{u}_{S_0}| > \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2$. If $z_j^*(\bar{v}) = 1$ and $b_j(\hat{w}) > 0$, then $\hat{\beta}_j^0 > 0$ and

$$\beta_j' = |\hat{\beta}_j^0| - \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2 > 0.$$

If $z_j^*(\bar{v}) = 1$ and $b_j(\hat{w}) \leq 0$, then $\hat{\beta}_j^0 > 0$ and we have

$$\beta_j' = |\hat{\beta}_j^0| + \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2 > 0.$$

If $z_j^*(\bar{v}) = -1$ and $b_j(\hat{w}) < 0$, then $\hat{\beta}_j^0 < 0$ and

$$\beta_j' = -|\hat{\beta}_j^0| + \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2 < 0.$$

If $z_j^*(\bar{v}) = -1$ and $b_j(\hat{w}) \geq 0$, then $\hat{\beta}_j^0 < 0$ and

$$\beta_j' = -|\hat{\beta}_j^0| - \lambda|b_j(\hat{w})|/\|Xb(\hat{w})\|_2^2 < 0.$$

Case 2 Let now $j \notin S_0$. Then

$$\beta_j' = -\lambda b_j(\hat{w})/\|Xb(\hat{w})\|_2^2,$$

so

$$z_j(\hat{w})\beta_j' = -\lambda z_j(\hat{w})b_j(\hat{w})/\|Xb(\hat{w})\|_2^2 > 0.$$

Thus

$$z(\hat{w}) \in \partial\|\beta'\|_1.$$

Furthermore, by the first part of Lemma 35,

$$X^T X(\beta' - \hat{\beta}^0) = -\lambda X^T Xb(\hat{w})/\|Xb(\hat{w})\|_2 = -\lambda\hat{W}z(\hat{w}).$$

So $\beta' =: \hat{\beta}$ satisfies the KKT conditions with $\tilde{\zeta} = z(\hat{w})$. We further have

$$\begin{aligned}\|X(\hat{\beta} - \hat{\beta}^0)\|_2^2 &= \lambda^2 b^T(\hat{w})\hat{W}z(\hat{w})/\|Xb(\hat{w})\|_2^2 \\ &= \lambda^2/\|Xb(\hat{w})\|^2 \\ &\geq \lambda^2 s_0/(n\hat{\kappa}^2(1 + \bar{v}, S_0))\end{aligned}$$

where in the last step we used the second part of Lemma 35. Finally, by the triangle inequality

$$\begin{aligned}\|X(\hat{\beta} - \beta^0)\|_2 &\geq \|X(\hat{\beta} - \hat{\beta}^0)\|_2 - \mathbf{U}(S_0) \\ &\geq \frac{\lambda\sqrt{s_0}}{\sqrt{n}\hat{\kappa}(1 + \bar{v}, S_0)} - \mathbf{U}(S_0) \\ &\geq \frac{\lambda\sqrt{s_0}}{\sqrt{n}\hat{\kappa}(1 + \bar{v}, S_0)} - \sqrt{s_0} - \sqrt{2x}.\end{aligned}$$

□

11.6.6. Random Part

In Lemma 36, we showed that the conclusion (12) of Theorem 17 holds on the set $\mathcal{T}$. This subsection obtains that $\mathbb{P}(\mathcal{T}) \geq 1 - \exp[-t] + \exp[-x]$.

Lemma 37 *It holds that*

$$\mathbb{P}(\mathcal{T}) \geq 1 - \exp[-t] - \exp[-x].$$

Proof of Lemma 37. Apply Lemma 41 with $Z_j = \hat{u}_j/u_j$ for $j \in S_0$ and $Z_j = \hat{v}_j/v_j$ for $j \notin S_0$ to find that with probability at least $1 - \exp[-t]$

$$|\hat{u}_j| \leq \lambda\bar{u}_j\ \forall j \in S_0,\ |\hat{v}_j| \leq \lambda\bar{v}_j\ \forall j \notin S_0.$$

Furthermore, the random variable $\mathbf{U}^2(S_0)$ has a chi-squared distribution with s_0 degrees of freedom. Lemma 42 gives that with probability at least $1 - \exp[-x]$,

$$\mathbf{U}(S_0) \leq \sqrt{s_0} + \sqrt{2x}.$$

□

11.6.7. Collecting the Pieces

Combining Lemma 36 with Lemma 37 completes the proof of Theorem 17.

11.7. Proof of Theorem 18

The proof is along the lines of Theorem 9.

11.7.1. Comparing the KKT Conditions

We compare the KKT conditions for the noisy Lasso with those for the noiseless Lasso.

Lemma 38 *It holds that*

$$\|X(\hat\beta-\beta^*)\|_2^2+\lambda\|\hat\beta\|_1-\lambda^*\hat\beta^T z^*\leq(\hat\beta-\beta^*)^TX^T\epsilon+(\lambda-\lambda^*)\|\beta^*\|_1.$$

Proof of Lemma 38. The KKT conditions (19) for $\hat\beta$ can be written as

$$X^TX(\hat\beta-\beta^0)+\lambda\hat\zeta=X^T\epsilon.$$

where $\hat\zeta\in\partial\|\hat\beta\|_1$. By the KKT conditions (6) for β^*

$$X^TX(\beta^*-\beta^0)+\lambda^*\zeta^*=0.$$

Hence, taking the difference

$$X^TX(\hat\beta-\beta^*)+\lambda\hat\zeta-\lambda^*\zeta^*=X^T\epsilon.$$

Multiply by $(\hat\beta-\beta^*)^T$ to find

$$\|X(\hat\beta-\beta^*)\|_2^2+\lambda(\hat\beta-\beta^*)^T\hat\zeta-\lambda^*(\hat\beta-\beta^*)^T\zeta^*=(\hat\beta-\beta^*)^TX^T\epsilon.$$

But

$$\begin{aligned}&\lambda(\hat\beta-\beta^*)^T\hat\zeta-\lambda^*(\hat\beta-\beta^*)^T\zeta^*\\ =\;&\lambda\|\hat\beta\|_1-\lambda^*\hat\beta^T\zeta^*+\lambda^*\|\beta^*\|_1-\lambda\beta^{*T}\hat\zeta\\ =\;&\lambda\|\hat\beta\|_1-\lambda^*\hat\beta^T\zeta^*+\lambda\|\beta^*\|_1-\lambda\beta^{*T}\hat\zeta-(\lambda-\lambda^*)\|\beta^*\|_1\\ \geq\;&\lambda\|\hat\beta\|_1-\lambda^*\hat\beta^T\zeta^*-(\lambda-\lambda^*)\|\beta^*\|_1\end{aligned}$$

where we used that

$$\|\beta^*\|_1-\beta^{*T}\hat\zeta\geq 0.$$

Therefore

$$\|X(\hat\beta-\beta^*)\|_2^2+\lambda\|\hat\beta\|_1-\lambda^*\hat\beta^T z^*\leq(\hat\beta-\beta^*)^TX^T\epsilon+(\lambda-\lambda^*)\|\beta^*\|_1.$$

□

11.7.2. Projections

Recall the notation of Subsection 8.1. We let moreover $\hat v_{-S_0}$ be the vector

$$\hat v_{-S}^S:=(X_{-S}\mathrm{A}X_S)^T\epsilon.$$

As before, we denote the projection of ϵ on the space spanned by the columns of X_S by $\epsilon\mathrm{P}X_S$ and write

$$\mathbf{U}(S):=\|\epsilon\mathrm{P}X_S\|_2.$$

11.7.3. Choice of λ

Recall that we require that for some $t>0$

$$\lambda>\|v_{-S}^S\|_\infty\sqrt{2(\log(2p)+t)}.$$

11.7.4. The Set $\mathcal{T}^S$

Recall

$$\bar v^S:=v_j^S\sqrt{2(\log(2p)+t)}/\lambda,\ j\notin S.$$

Let

$$\mathcal{T}^S:=\{|\hat v_j|\leq\lambda\bar v_j\ \forall\ j\notin S\}\cap\{\mathbf{U}(S)\leq\sqrt{s}+\sqrt{2x}\}.$$

11.7.5. Deterministic Part

Lemma 39 *On the set $\mathcal{T}^S$ it holds that*

$$\|X(\hat\beta-\beta^*)\|_2\leq\sqrt{s}+\sqrt{2x}+(\lambda-\lambda^*)\sqrt{s/n}/\hat\kappa(\bar w^S,S).$$

Proof of Lemma 39. Since $S_*\subset S$

$$X(\hat\beta-\beta^*)=X_S\hat b_S+X_{-S}\mathrm{A}X_S\hat\beta_{-S}$$

where

$$X_S\hat b_S=X_S(\hat\beta_S-\beta_S^*)+(X_{-S}\mathrm{P}X_S)\hat\beta_{-S}.$$

In view of Lemma 38,

$$\begin{aligned}&\|X(\hat\beta-\beta^*)\|_2^2+\lambda\|\hat\beta\|_1-\lambda^*\beta^Tz^*\\ &\leq\hat b_S^TX_S^T\epsilon+\left[X_{-S}\mathrm{A}X_S\hat\beta_S\right]^T\epsilon+(\lambda-\lambda^*)\|\beta^*\|_1\end{aligned}$$

By the Cauchy-Schwarz inequality and since we are on $\mathcal{T}^S$

$$\hat b_S^TX_S^T\epsilon\leq\mathbf{U}(S)\|X\hat b_S\|_2\leq(\sqrt{s}+\sqrt{2x})\|X\hat b_S\|_2\leq(\sqrt{s}+\sqrt{2x})\|X(\hat\beta-\beta^*)\|_2$$

where in the last inequality we used Pythagoras rule. Moreover, by the definition of $\hat v_{-S}^S$ and since we are on the set $\mathcal{T}^S$

$$\left[X_{-S}\mathrm{A}X_S\hat\beta_{-S}\right]^T\epsilon=\hat\beta_{-S}^T\hat v_{-S}^S\leq\lambda\sum_{j\notin S}\bar v_{-S}^S|\hat\beta_j|.$$

On the other hand,

$$\lambda\|\hat\beta_{-S}\|_1-\lambda^*\zeta_{-S}^{*T}\hat\beta_{-S}\geq\lambda\sum_{j\notin S}(1-\lambda^*|\zeta_j^*|/\lambda)|\hat\beta_j|$$

and

$$(\lambda-\lambda^*)\|\beta^*\|_1-\lambda\|\hat\beta_S\|_1+\lambda^* z^{*T}\hat\beta_S\leq(\lambda-\lambda^*)\|\hat\beta_S-\beta^*_S\|_1.$$

If $\|X(\hat\beta-\beta^*)\|_2\leq\sqrt{s}+\sqrt{2x}$ we are done. Suppose therefore that $\|X(\hat\beta-\beta^*\|_2>\sqrt{s}+\sqrt{2x}$. Then we see that

$$\begin{aligned}&\|X(\hat\beta-\beta^*)\|_2^2-(\sqrt{s}+\sqrt{2x})\|X(\hat\beta-\beta^*)\|_2\\ =\;&\|X(\hat\beta-\beta^*)\|_2\Big(\|X(\hat\beta-\beta^*)\|_2-\sqrt{s}-\sqrt{2x}\Big)\\ >\;&0.\end{aligned}$$

But then

$$\lambda\sum_{j\notin S}(1-\bar v_j^S-\lambda^*|\zeta_j^*|/\lambda)|\hat\beta_j|<(\lambda-\lambda^*)\|\hat\beta_S-\beta^*\|_1.$$

or

$$\|\hat\beta_S-\beta^*_S\|_1-\|\bar W^S\hat\beta_{-S}\|_1>0.$$

Then

$$\|\hat\beta_S-\beta^*_S\|_1-\|\bar W^S\hat\beta_{-S}\|_1\leq(\sqrt{s/n})\|X(\hat\beta-\beta^*)\|_2/\hat\kappa(\bar w^S,S).$$

We thus arrive at

$$\begin{aligned}&\|X(\hat\beta-\beta^*)\|_2^2\\ \leq\;&\Big(\sqrt{s}+\sqrt{2x}+(\lambda-\lambda^*)\sqrt{s/n}/\hat\kappa(\bar w^S,S)\Big)\|X(\hat\beta-\beta^*)\|_2\end{aligned}$$

or

$$\|X(\hat\beta-\beta^*)\|_2\leq\sqrt{s}+\sqrt{2x}+(\lambda-\lambda^*)\sqrt{s/n}/\hat\kappa(\bar w^S,S).$$

□

11.7.6. Random Part

Lemma 40 *We have*

$$\mathbb{P}(\mathcal{T}^S)\geq 1-\exp[-t]-\exp[-x].$$

Proof of Lemma 40. This follows from Lemma 41 and Lemma 42. □

11.7.7. Finalizing the Proof of Theorem 18

Combine Lemma 39 with Lemma 40.

11.8. Proof of the Lemma in Section 9

Proof of Lemma 21. Write $g_i:=w_if_i$, $i=1,\ldots,n$ and $u_j:=d_j/2$, $j=2,\ldots,s$. Then we have

$$\begin{aligned}&\sum_{j=1}^s|g_{d_j+1}-g_{d_j}|-\sum_{i=2}^{d_1}|g_i-g_{i-1}|-\sum_{j=2}^{s-1}\sum_{i=d_j+1}^{d_{j+1}}|g_i-g_{i-1}|-\sum_{i=d_s+1}^{n}|g_i-g_{i-1}|\\ \leq\;&\frac{1}{d_1}\sum_{i=1}^{d_1}|g_i|+\frac{1}{u_2}\sum_{i=d_1+1}^{d_1+u_2}|g_i|\\ +\;&\frac{1}{d_2-u_2}\sum_{i=d_1+u_2+1}^{d_1+d_2}|g_i|+\frac{1}{u_3}\sum_{i=d_1+d_2+1}^{d_1+d_2+u_3}|g_i|\\ &\cdots\\ +\;&\frac{1}{d_{s-1}-u_{s-1}}\sum_{i=d_1+\cdots+d_{s-2}+u_{s-1}+1}^{d_1+\cdots+d_{s-1}}|g_i|+\frac{1}{u_s}\sum_{i=d_1+\cdots+d_{s-1}+1}^{d_1+\cdots+d_{s-1}+u_s}|g_i|\\ +\;&\frac{1}{d_s-u_s}\sum_{i=d_1+\cdots+d_{s-1}+u_s+1}^{d_1+\cdots+d_s}|g_i|+\frac{1}{d_{s+1}}\sum_{i=d_1+\cdots+d_s+1}^{n}|g_i|\\ \leq\;&\Big(\frac{1}{d_1^2}\sum_{i=1}^{d_1}w_i^2+\frac{1}{u_2^2}\sum_{i=d_1+1}^{d_1+u_2}w_i^2\\ +\;&\frac{1}{(d_2-u_2)^2}\sum_{i=d_1+u_2+1}^{d_1+d_2}w_i^2+\frac{1}{u_3^2}\sum_{i=d_1+d_2+1}^{d_1+d_2+u_3}w_i^2\\ &\cdots\\ +\;&\frac{1}{(d_{s-1}-u_{s-1})^2}\sum_{i=d_1+\cdots+d_{s-2}+u_{s-1}+1}^{d_1+\cdots+d_{s-1}}w_i^2+\frac{1}{u_s^2}\sum_{i=d_1+\cdots+d_{s-1}+1}^{d_1+\cdots+d_{s-1}+u_s}w_i^2\\ +\;&\frac{1}{(d_s-u_s)^2}\sum_{i=d_1+\cdots+d_{s-1}+u_s+1}^{d_1+\cdots+d_s}w_i^2+\frac{1}{d_{s+1}^2}\sum_{i=d_1+\cdots+d_s+1}^{n}w_i^2\Big)^{1/2}\\ &\times\Big(\sum_{i=1}^n f_i^2\Big)^{1/2}\\ \leq\;&\sqrt{\frac{n}{d_1}+\frac{n}{u_2}+\frac{n}{d_2-u_2}+\cdots+\frac{n}{d_{s-1}-u_{s-1}}+\frac{n}{u_s}+\frac{n}{d_s-u_s}+\frac{n}{d_{s+1}}}\\ &\times\sqrt{\sum_{i=1}^n|f_i|^2/n}\\ &\times\|w\|_\infty.\end{aligned}$$

Moreover

$$\begin{aligned}
&\sum_{j=1}^{s} w_{d_j+1}|f_{d_j+1}-f_{d_j}| - \sum_{i=2}^{d_1} w_i|f_i - f_{i-1}| \\
- &\sum_{j=2}^{s-1}\sum_{i=d_j+1}^{d_{j+1}} w_i|f_i-f_{i-1}| - \sum_{i=d_s+1}^{n} w_i|f_i-f_{i-1}| \\
\leq &\sum_{j=1}^{s}|g_{d_j+1}-g_{d_j}| - \sum_{i=2}^{d_1}|g_i-g_{i-1}| - \sum_{j=2}^{s-1}\sum_{i=d_j+1}^{d_{j+1}}|g_i-g_{i-1}| - \sum_{i=d_s+1}^{n}|g_i-g_{i-1}| \\
+ &\sum_{i=2}^{n}|w_i-w_{i-1}||f_{i-1}|,
\end{aligned}$$

and

$$\begin{aligned}
\sum_{i=2}^{n}|w_i-w_{i-1}||f_{i-1}| &\leq \sqrt{\sum_{i=2}^{n}(w_i-w_{i-1})^2}\sqrt{\sum_{i=2}^{n} f_{i-1}^2} \\
&\leq \sqrt{\sum_{i=2}^{n}(w_i-w_{i-1})^2}\sqrt{\sum_{i=1}^{n} f_i^2}
\end{aligned}$$

Thus we conclude

$$\begin{aligned}
&\sum_{j=1}^{s} w_{d_j+1}|f_{d_j+1}-f_{d_j}| \\
- &\sum_{i=2}^{d_1} w_i|f_i-f_{i-1}| - \sum_{j=2}^{s-1}\sum_{i=d_j+1}^{d_{j+1}} w_i|f_i-f_{i-1}| - \sum_{i=d_s+1}^{n} w_i|f_i-f_{i-1}| \\
\leq &\left(\|w\|_\infty\sqrt{\frac{n}{d_1}+\sum_{j=2}^{s}\frac{4n}{d_j}+\frac{n}{d_{s+1}}} + \sqrt{n\sum_{i=2}^{n}(w_i-w_{i-1})^2}\right)\sqrt{\sum_{i=1}^{n} f_i^2/n}.
\end{aligned}$$

□

11.9. Proof of Theorem 1

This follows from Corollary 12 combined with Theorem 14, where in the latter we replace $\hat{\Sigma} := X^T X/n$ by the population version Σ_0. This works because we replaced condition (8) by its population counterpart condition (3).

12. Tools from Probability Theory

We first present three standard lemmas for Gaussian random variables, Lemmas 41, 42 and 43. These three lemmas are followed by a concentration of measure result and a result for Gaussian quadratic forms.

Lemma 41 *Let $Z_1, \ldots, Z_p$ be standard normal random variables. Then it holds for all $t > 0$ that*

$$\mathbb{P}\left(\max_{1\leq j\leq p}|Z_j| \geq \sqrt{2(\log(2p)+t)}\right) \leq \exp[-t].$$

Proof of Lemma 41. For each $t > 0$

$$\mathbb{P}(|Z_1| \geq \sqrt{2t}) \leq 2\exp[-t].$$

So by the union bound, for any $t > 0$,

$$\begin{aligned}
\mathbb{P}\left(\max_{1\leq j\leq p}|Z_j| > \sqrt{2(\log(2p)+t)}\right) &\leq p\mathbb{P}(|Z_1| \geq \sqrt{2(\log(2p)+t)}) \\
&\leq 2p\exp[-(\log(2p+t)] = \exp[-t].
\end{aligned}$$

□

Lemma 42 *Let $Z := (Z_1, \ldots, Z_T)^T$ be a vector with i.i.d. standard Gaussian entries. Then it holds for all $x > 0$ that*

$$\mathbb{P}\left(\|Z\|_2 \geq \sqrt{T} + \sqrt{2x}\right) \leq \exp[-x]$$

and

$$\mathbb{P}\left(|\|Z\|_2 - \sqrt{T}| \geq \sqrt{2x}\right) \leq 2\exp[-x].$$

Proof of Lemma 42. This follows from concentration of measure (Borell, 1975, Giné and Nickl, 2015, Theorem 2.5.7) because the map $Z \mapsto \|Z\|_2$ is Lipschitz. Alternatively, one may apply Lemma 1 in Laurent and Massart (2000). □

Lemma 43 *Let $(U, V) \in \mathbb{R}^{n\times 2}$ have i.i.d Gaussian rows with mean zero and covariance matrix*

$$\begin{pmatrix} \sigma_u^2 & \sigma_{uv} \\ \sigma_{uv} & \sigma_v^2 \end{pmatrix}.$$

Then for all $t > 0$, with probability at least $1 - 4\exp[-t]$

$$|U^T V - n\sigma_{uv}| \leq 3\sigma_u\sigma_v\left(\sqrt{2nt} + t\right).$$

Proof of Lemma 43. By standard arguments (see van de Geer (2017) for tracking down some constants) one can derive that with probability at least $1 - 4\exp[-t]$

$$|U^T V - n\sigma_{uv}| \leq (\sigma_u\sigma_v + 2|\sigma_{u,v}|)\sqrt{2nt} + (\sigma_u\sigma_v + 2|\sigma_{u,v}|)t.$$

We simplify this to: with probability at least $1 - 4\exp[-t]$

$$|U^T V - n\sigma_{uv}| \leq 3\sigma_u \sigma_v \Big(\sqrt{2nt} + t\Big).$$

□

This is the concentration of measure lemma that we use in Section 4.

Lemma 44 *For any $b \in \mathbb{R}^p$ and all $x > 0$, we have*

$$\mathbb{P}\Big(\|X(\hat{\beta} - b)\|_2 \geq m_b + \sqrt{2x}\Big) \leq \exp[-x]$$

and

$$\mathbb{P}\Big(\Big|\|X(\hat{\beta} - b)\|_2 - m_b\Big| \geq \sqrt{2x}\Big) \leq 2\exp[-x]$$

where $m_b := \mathbb{E}(\|X(\hat{\beta} - b)\|_2 | X)$.

Proof of Lemma 44. This follows from concentration of measure see e.g. Borell (1975), or Giné and Nickl (2015), Theorem 2.5.7, as the map $\epsilon \mapsto \|X(\hat{\beta} - b)\|^2$ is Lipschitz, see also van de Geer and Wainwright (2017). □

Finally, we give a result for Gaussian quadratic forms.

Lemma 45 *Let X have i.i.d. $\mathcal{N}(0, \Sigma_0)$-distributed rows and let M be a (sequence of) constant(s) such that*

$$M^2 = o\Big(n/(\|\Sigma_0\|_\infty \log(2p))\Big).$$

Then, for a suitable sequence $\eta_M = o(1)$, with probability tending to one

$$\inf_{\|b\|_1 \leq M\|\Sigma_0^{1/2} b\|_2} \frac{\|Xb\|_2^2/n}{\|\Sigma_0^{1/2} b\|_2^2} \geq (1 - \eta_M)^2.$$

Proof of Lemma 45. See for example Chapter 16 in van de Geer (2016) and its references, or van de Geer and Muro (2014). □

Acknowledgments

We thank Rico Zenklusen from the Institute of Operations Research, ETH Zürich, and Hamza Fawzi from the Department of Applied Mathematics and Theoretical Physics at the University of Cambridge, for very helpful discussions. Research supported by Isaac Newton Institute for Mathematical Sciences, program *Statistical Scalability*, EPSRC Grant Number LNAG/036 RG91310.

References

P. C. Bellec. Optimistic lower bounds for convex regularized least-squares. *arXiv preprint arXiv:1703.01332*, 2017.

V. Belloni, A.and Chernozhukov and L. Wang. Pivotal estimation via square-root Lasso in nonparametric regression. *Annals of Statistics*, 42(2):757–788, 2014.

C. Borell. The Brunn-Minkowski inequality in Gauss space. *Inventiones Mathematicae*, 30 (2):207–216, 1975.

A. S. Dalalyan, M. Hebiri, and J. Lederer. On the prediction performance of the Lasso. *Bernoulli*, 23(1):552–581, 2017.

D.L. Donoho and J. Tanner. Neighborliness of randomly projected simplices in high dimensions. *Proceedings of the National Academy of Sciences of the United States of America*, 102(27):9452–9457, 2005.

E. Giné and R. Nickl. *Mathematical Foundations of Infinite-Dimensional Models*. Cambridge University Press, 2015.

C. Giraud. *Introduction to High-Dimensional Statistics*, volume 138. CRC Press, 2014.

V. Koltchinskii, K. Lounici, and A.B. Tsybakov. Nuclear-norm penalization and optimal rates for noisy low-rank matrix completion. *Annals of Statistics*, 39(5):2302–2329, 2011.

B. Laurent and P. Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 99:879–898, 2012.

R. Tibshirani. Regression analysis and selection via the Lasso. *Journal of the Royal Statistical Society Series B*, 58:267–288, 1996.

S. van de Geer. *Estimation and Testing Under Sparsity: École d'Eté de Probabilités de Saint Flour XLV-2016*. Springer Science & Business Media, 2016.

S. van de Geer. On the efficiency of the de-biased Lasso, 2017. arXiv:1708.07986.

S. van de Geer and A. Muro. On higher order isotropy conditions and lower bounds for sparse quadratic forms. *Electronic Journal of Statistics*, 8:3031–3061, 2014.

S. van de Geer and M. Wainwright. On concentration for (regularized) empircal risk minimization. *Sankhyā*, 79-A:159–200, 2017.

S.A. van de Geer. The deterministic Lasso. In *JSM proceedings, 2007, 140*. American Statistical Association, 2007.

Y. Zhang, M. Wainwright, and M. Jordan. Lower bounds on the performance of polynomial-time algorithms for sparse linear regression. In *COLT*, pages 921–948, 2014.

Journal of Machine Learning Research 19 (2018) 1-49 Submitted 4/17; Revised 2/18; Published 9/18

Harmonic Mean Iteratively Reweighted Least Squares for Low-Rank Matrix Recovery

Christian Kümmerle C.KUEMMERLE@TUM.DE
Department of Mathematics
Technische Universität München
Boltzmannstr. 3, 85748 Garching/Munich, Germany

Juliane Sigl JULIANE.SIGL@MA.TUM.DE
Department of Mathematics
Technische Universität München
Boltzmannstr. 3, 85748 Garching/Munich, Germany

Editor: Benjamin Recht

Abstract

We propose a new iteratively reweighted least squares (IRLS) algorithm for the recovery of a matrix $X \in \mathbb{C}^{d_1 \times d_2}$ of rank $r \ll \min(d_1, d_2)$ from incomplete linear observations, solving a sequence of low complexity linear problems. The easily implementable algorithm, which we call harmonic mean iteratively reweighted least squares (`HM-IRLS`), optimizes a non-convex Schatten-p quasi-norm penalization to promote low-rankness and carries three major strengths, in particular for the matrix completion setting. First, we observe a remarkable global convergence behavior of the algorithm's iterates to the low-rank matrix for relevant, interesting cases, for which any other state-of-the-art optimization approach fails the recovery. Secondly, `HM-IRLS` exhibits an empirical recovery probability close to 1 even for a number of measurements very close to the theoretical lower bound $r(d_1 + d_2 - r)$, i.e., already for significantly fewer linear observations than any other tractable approach in the literature. Thirdly, `HM-IRLS` exhibits a locally superlinear rate of convergence (of order $2 - p$) if the linear observations fulfill a suitable null space property. While for the first two properties we have so far only strong empirical evidence, we prove the third property as our main theoretical result.

Keywords: Iteratively Reweighted Least Squares, Low-Rank Matrix Recovery, Matrix Completion, Non-Convex Optimization

1. Introduction

The problem of recovering a low-rank matrix from incomplete linear measurements or observations has gained considerable attention in the last few years due to the omnipresence of low-rank models in different areas of science and applied mathematics. Low-rank models arise in a variety of areas such as system identification (Liu et al., 2013; Liu and Vandenberghe, 2010), signal processing (Ahmed and Romberg, 2015), quantum tomography (Gross et al., 2010; Gross, 2011) and phase retrieval (Candès et al., 2013; Candès et al., 2013; Gross et al., 2015). An instance of this problem of particular importance, e.g., in recommender systems (Srebro et al., 2005; Goldberg et al., 1992; Candès and Recht, 2009), is the *matrix completion* problem, where the measurements correspond to entries of the matrix to be recovered.

Although the low-rank matrix recovery problem is NP-hard in general, several tractable algorithms have been proposed that allow for provable recovery in many important cases. The *nuclear norm minimization* (`NNM`) approach (Fazel, 2002; Candès and Recht, 2009), which solves a surrogate semidefinite program, is particularly well-understood. For `NNM`, recovery guarantees have been shown for a number of measurements on the order of the information theoretical lower bound $r(d_1 + d_2 - r)$, if r denotes the rank of a $d_1 \times d_2$-matrix (Recht et al., 2010; Candès and Recht, 2009); i.e., for a number of measurements $m \geqslant \rho r(d_1 + d_2 - r)$ with some oversampling constant $\rho \geqslant 1$. Even though `NNM` is solvable in polynomial time, it can be computationally very demanding if the problem dimensions are large, which is the case in many potential applications. Another issue is that although the number of measurements necessary for successful recovery by nuclear norm minimization is of *optimal order*, it is not *optimal*. More precisely, it turns out that the oversampling factor ρ of nuclear norm minimization *needs to be much larger than the oversampling factor of some other, non-convex algorithmic approaches* (Zheng and Lafferty, 2015; Tanner and Wei, 2013).

These limitations of convex relaxation approaches have led to a rapidly growing line of research discussing the advantages of non-convex optimization for the low-rank matrix recovery problem (Jain et al., 2010; Tanner and Wei, 2013; Haldar and Hernando, 2009; Jain et al., 2013; Wen et al., 2012; Tanner and Wei, 2016; Vandereycken, 2013; Wei et al., 2016; Tu et al., 2016). For several of these non-convex algorithmic approaches, recovery guarantees comparable to those of `NNM` have been derived (Candès et al., 2015; Tu et al., 2016; Zheng and Lafferty, 2015; Sun and Luo, 2016). Their advantage is a higher empirical recovery rate and an often more efficient implementation. While there are some results about global convergence of first-order methods minimizing a non-convex objective (Ge et al., 2016; Bhojanapalli et al., 2016) so that a success of the method might not depend on a particular initialization, the assumptions of these results are not always optimal, e.g., in the scaling of the numbers of measurements m in the rank r (Ge et al., 2016, Theorem 5.3). In general, the success of many non-convex optimization approaches relies on a distinct, possibly expensive initialization step.

1.1 Contribution of this paper

In this spirit, we propose a new iteratively reweighted least squares (IRLS) algorithm for the low-rank matrix recovery problem[1] that strives to minimize a non-convex objective function based on the Schatten-p quasi-norm

$$\min_X \|X\|_{S_p}^p \text{ subject to } \Phi(X) = Y, \tag{1}$$

for $0 < p < 1$, where $\Phi : \mathbb{C}^{d_1 \times d_2} \to \mathbb{C}^m$ is the linear measurement operator and $Y \in \mathbb{C}^m$ is the data vector defining the problem. The overall strategy of the proposed IRLS algorithm is to mimic this minimization by a sequence of weighted least squares problems. This strategy

1. The algorithm and partial results were presented at the 12th International Conference on Sampling Theory and Applications in Tallinn, Estonia, July 3–7, 2017. The corresponding conference paper has been published in its proceedings (Kümmerle and Sigl, 2017).

is shared by the related previous algorithms of (Fornasier et al., 2011; Mohan and Fazel, 2012) which minimize (1) by defining iterates as

$$X^{(n+1)} = \min_X \|W_L^{(n)\frac{1}{2}} X\|_F^2 \text{ subject to } \Phi(X) = Y, \tag{2}$$

where $W_L^{(n)} \approx \left(X^{(n)} X^{(n)*}\right)^{\frac{p-2}{2}}$ is a so-called *weight matrix* which *reweights* the quadratic penalty by operating on the column space of the matrix variable. Thus, we call this column-reweighting type of IRLS algorithms `IRLS-col`. Due to the inherent symmetry, it is evident to conceive, still in the spirit of (Fornasier et al., 2011; Mohan and Fazel, 2012), the algorithm `IRLS-row`

$$X^{(n+1)} = \min_X \|W_R^{(n)\frac{1}{2}} X^*\|_F^2 \text{ subject to } \Phi(X) = Y \tag{3}$$

with $W_R^{(n)} \approx \left(X^{(n)*} X^{(n)}\right)^{\frac{p-2}{2}}$, which reweights the quadratic penalty by acting on the *row space* of the matrix variable. We note that even for square dimensions $d_1 = d_2$, `IRLS-col` and `IRLS-row` do not coincide.

In this paper, as an important innovation, we propose the use of a different type of weight matrices, which can be interpreted as the *harmonic mean* of the matrices $W_L^{(n)}$ and $W_R^{(n)}$ above. This motivates the name *harmonic mean iteratively reweighted least squares* (`HM-IRLS`) for the corresponding algorithm. The harmonic mean of the weight matrices of `IRLS-col` and of `IRLS-row` in `HM-IRLS` is able to use the information in both the column and the row space of the iterates, and it also gives rise to a *qualitatively better* behavior than the use of more obvious symmetrizations as, e.g., the arithmetic mean of weight matrices would allow for both in theory and in practice.

We argue that the choice of harmonic mean weight matrices as in `HM-IRLS` leads to an efficient algorithm for the low-rank matrix recovery problem with fast convergence and superior performance in terms of sample complexity, also compared to algorithms based on strategies different from IRLS.

On the one hand, we show that the accumulation points of the iterates of `HM-IRLS` converge to stationary points of a smoothed Schatten-p functional under the linear constraint, as it is known for, e.g., `IRLS-col`, c.f. (Fornasier et al., 2011; Mohan and Fazel, 2012). On the other hand, we extend the theoretical guarantees which are based on a Schatten-p *null space property* (NSP) of the measurement operator (Oymak et al., 2011; Foucart and Rauhut, 2013) to `HM-IRLS`.

Our main theoretical result is that `HM-IRLS` exhibits a locally superlinear convergence rate of order $2-p$ in the neighborhood of a low-rank matrix for the non-convexity parameter $0 < p < 1$ connected to the Schatten-p quasi-norm, if the measurement operator fulfills the mentioned NSP of sufficient order. For $p \ll 1$, this means that the convergence rate is *almost quadratic*.

Although parts of our theoretical results, as in the case of the IRLS algorithms algorithms of Fornasier et al. (2011) and Mohan and Fazel (2012), do not apply to the matrix completion setting, due to the popularity of the problem and for reasons of comparability with other algorithms, we conduct numerical experiments to explore the empirical performance of `HM-IRLS` also for this setting. Surprisingly enough we observe that the theoretical

results comply with our numerical experiments also for matrix completion. In particular, the theoretically predicted local convergence rate of order $2 - p$ can be observed very precisely for this important measurement model as well (see Figures 3 to 5).

This local superlinear convergence rate of `HM-IRLS` is unprecedented by previous IRLS variants such as `IRLS-col` or those that use the arithmetic mean of the one-sided weight matrices: this means that neither can a superlinear rate be verified numerically, nor is it possible to show such a rate by our proof techniques for any other previsouly considered IRLS variant designed for the low-rank matrix recovery problem.

To the best of our knowledge, `HM-IRLS` is the first algorithm for low-rank matrix recovery which achieves superlinear rate of convergence for low complexity measurements as well as for larger problems.

Additionally, we conduct extensive numerical experiments comparing the efficiency of `HM-IRLS` with previous IRLS algorithms as `IRLS-col`, Riemannian optimization techniques (Vandereycken, 2013), alternating minimization approaches (Haldar and Hernando, 2009; Tanner and Wei, 2016), algorithms based on iterative hard thresholding (Kyrillidis and Cevher, 2014; Blanchard et al., 2015), and others (Park et al., 2016), in terms of sample complexity, again for the important case of *matrix completion*.

The experiments lead to the following observation: `HM-IRLS` recovers low-rank matrices systematically with an optimal number of measurements that is very close to the theoretical lower bound on the number of measurements that is necessary for recovery with high empirical probability. We consider this result to be remarkable, as it means that for problems of moderate dimensionality (matrices of $\approx 10^7$ variables, e.g. $(d_1 \times d_2)$-matrices with $d_1 \approx d_2 \approx 3 \cdot 10^3$) *the proposed algorithm needs fewer measurements for the recovery of a low rank matrix than all the state-of-the-art algorithms we included in our experiments* (see Figure 6).

An important practical observation of `HM-IRLS` is that its performance is very robust to the choice of the initialization and that it can be used as a stand-alone algorithm to recover low-rank matrices also starting from a trivial initialization. This is suggested by our numerical experiments since even for random or adversary initializations, `HM-IRLS` converges to the low-rank matrix, even though it is based on an objective function which is highly non-convex. While a complete theoretical understanding of this behavior is not yet achieved, we regard the empirical evidence in a variety of interesting cases as strong. In this context, we consider a proof of the global convergence of `HM-IRLS` for non-convex penalizations under appropriate assumptions as an interesting open problem.

1.2 Organization of the paper

We proceed in the paper as follows. In Section 2, we introduce some notation to be used and provide some background about different reformulations of the Schatten-p quasi-norm in terms of weighted ℓ_2-norms. This leads to the derivation of the harmonic mean iteratively reweighted least squares (`HM-IRLS`) algorithm in Section 3. We present our main theoretical results, the convergence guarantees and the locally superlinear convergence rate for the algorithm in Section 4. Numerical experiments and comparisons to state-of-the-art methods for low-rank matrix recovery are carried out in Section 5. In Section 6, we interpret the algorithm's different steps as minimizations of an auxililary functional with respect to its

arguments and show theoretical guarantees for `HM-IRLS` extending similar guarantees for `IRLS-col`. After this, we detail the proof of the locally superlinear convergence rate under appropriate assumptions on the null space of the measurement operator.

In Appendix A, we provide a short overview about Kronecker and Hadamard products, and end with some deferred proofs in Appendix B and Appendix C.

2. Notation and background

2.1 General notation, Schatten-p and weighted norms

In this section, we explain some of the notation we use in the course of this paper.

The set of matrices $X \in \mathbb{C}^{d_1 \times d_2}$ is denoted by $M_{d_1 \times d_2}$. Unless stated otherwise, vectors $x \in \mathbb{C}^d$ are considered as column vectors. We also use the vectorized form $X_{\mathrm{vec}} = \begin{bmatrix} X_1^T, \dots, X_j^T, \dots, X_{d_2}^T \end{bmatrix}^T \in \mathbb{C}^{d_1 d_2}$ of a matrix $X \in M_{d_1 \times d_2}$ with columns X_j, $j \in \{1, \dots, d_2\}$. The reverse recast of a vector $x \in \mathbb{C}^{d_1 d_2}$ into a matrix of dimension $d_1 \times d_2$ is denoted by $x_{\mathrm{mat}(d_1,d_2)} = [X_1, \dots, X_j, \dots, X_{d_2}]$, where $X_j = [x_{(d_1-1)\cdot j+1}, \dots, x_{(d_1-1)\cdot j+d_1}]^T$, $j = 1, \dots, d_2$ are column vectors, or X_{mat} if the dimensions are clear from the context. Obviously, it holds that $X = (X_{\mathrm{vec}})_{\mathrm{mat}}$.

The identity matrix in dimension $d \times d$ is denoted by $\mathbf{I}_d$. With $\mathbf{0}_{d_1 \times d_2} \in M_{d_1 \times d_2}$ and $\mathbf{1}_{d_1 \times d_2} \in M_{d_1 \times d_2}$ we denote the matrices with only 0- or 1-entries respectively. The set of Hermitian matrices is denoted by $H_{d \times d} := \{X \in M_{d \times d} \mid X = X^*\}$. We write $X^+ \in M_{d_1 \times d_2}$ for the Moore-Penrose inverse of the matrix $X \in M_{d_1 \times d_2}$.

Let $\mathcal{U}_d = \{U \in \mathbb{C}^{d \times d}; UU^* = \mathbf{I}_d\}$ denote the set of unitary matrices. Then the singular value decomposition of a matrix $X \in M_{d_1 \times d_2}$ can be written as $X = U\Sigma V^*$ with $U \in \mathcal{U}_{d_1}$, $V \in \mathcal{U}_{d_2}$ and $\Sigma \in M_{d_1 \times d_2}$, where Σ is diagonal and contains the singular values of X such that $\Sigma_{ii} = \sigma_i(X) \geqslant 0$ for $i \in \{1, \dots, \min(d_1, d_2)\}$. We define the *Schatten-p (quasi-)norm* of $X \in M_{d_1 \times d_2}$ as

$$\|X\|_{S_p} := \begin{cases} \operatorname{rank}(X), & \text{for } p = 0, \\ \left[\sum_{j=1}^{\min(d_1,d_2)} \sigma_j^p(X)\right]^{1/p}, & \text{for } 0 < p < \infty, \\ \sigma_{\max}(X), & \text{for } p = \infty. \end{cases} \tag{4}$$

Note that for $p = 1$, the Schatten-p norm is also called *nuclear norm*, written as $\|X\|_* := \|X\|_{S_1}$. The *trace* $\operatorname{tr}[X]$ of a matrix $X \in M_{d_1 \times d_2}$ is defined by the sum of its diagonal elements, $\operatorname{tr}[X] = \sum_{j=1}^{\min(d_1,d_2)} X_{jj}$. It can be seen that the p-th power of the Schatten-p norm coincides with $\|X\|_{S_p}^p = \operatorname{tr}\left[(X^*X)^{p/2}\right]$. The Schatten-2 norm is also called *Frobenius norm* and has the property that it is induced by the Frobenius scalar product $\langle X, Y\rangle_F = \operatorname{tr}[X^*Y]$, i.e., $\|X\|_F = \|X\|_{S_2} = \sqrt{\langle X, X\rangle_F}$. We define the *weighted Frobenius scalar product* of two matrices $X, Y \in M_{d_1 \times d_2}$ weighted by the the positive definite weight matrix $W \in H_{d_1 \times d_1}$ as $\langle X, Y\rangle_{F(W)} := \langle WX, Y\rangle_F = \langle X, WY\rangle_F$. This scalar product induces the *weighted Frobenius norm* $\|X\|_{F(W)} = \sqrt{\langle X, X\rangle_{F(W)}} = \sqrt{\operatorname{tr}[(WX)^*X]}$. It is clear that the Frobenius norm of a matrix X coincides with the ℓ_2-norm of its vectorization X_{vec}, i.e., $\|X\|_F = \|X_{\mathrm{vec}}\|_{\ell_2}$.

Similar to weighted Frobenius norms, we define the *weighted ℓ_2-scalar product* of vectors $x, y \in \mathbb{C}^d$ weighted by the positive definite weight matrix $W \in H_{d \times d}$ as $\langle x, y\rangle_{\ell_2(W)} = x^*Wy = \overline{y^*Wx}$ and its induced *weighted ℓ_2-norm* as $\|x\|_{\ell_2(W)} = \sqrt{x^*Wx}$. We use the notation $X \succ 0$ for a positive definite matrix $X \in H_{d \times d}$. Furthermore, we denote the range of a linear map $\Phi : M_{d_1 \times d_2} \to \mathbb{C}^m$ by $\operatorname{Ran}(\Phi) = \left\{Y \in \mathbb{C}^m;\ \text{there is } X \in M_{d_1 \times d_2} \text{ such that } Y = \Phi(X)\right\}$ and its null space by $\mathcal{N}(\Phi) = \left\{X \in M_{d_1 \times d_2}; \Phi(X) = 0\right\}$.

2.2 Problem setting and characterization of S_p- and reweighted Frobenius norm minimizers

Given a linear map $\Phi : M_{d_1 \times d_2} \to \mathbb{C}^m$ such that $m \ll d_1 d_2$, we want to uniquely identify and reconstruct an unknown matrix X_0 from its linear image $Y := \Phi(X_0) \in \mathbb{C}^m$. However, basic linear algebra tells us that this is not possible without further assumptions, since Φ is not injective if $m < d_1 d_2$. Indeed, there is a $(d_1 d_2 - m)$-dimensional affine space $\{X_0\} + \mathcal{N}(\Phi)$ fulfilling the linear constraint

$$\Phi(X) = Y.$$

Nevertheless, under the additional assumption that the matrix $X_0 \in M_{d_1 \times d_2}$ has rank $r < \min(d_1, d_2)$ and under appropriate assumptions on the map Φ, the recovery of X_0 is possible by solving the affine rank minimization problem

$$\min \operatorname{rank}(X) \text{ subject to } \Phi(X) = Y. \tag{5}$$

The unique solvability of (5) is given with high probability if, for example, Φ is a linear map whose matrix representation has i.i.d. Gaussian entries (Eldar et al., 2012) and $m = \Omega(r(d_1 + d_2))$. Unfortunately, solving (5) is intractable in general, but the works (Candès and Recht, 2009; Recht et al., 2010; Candès and Plan, 2011) suggest solving the tractable convex optimization program

$$\min \|X\|_{S_1} \text{ subject to } \Phi(X) = Y, \tag{6}$$

also called *nuclear norm minimization (`NNM`)*, as a proxy.

As discussed in the introduction, there are empirical as well as theoretical results (e.g., in (Daubechies et al., 2010; Chartrand, 2007)) coming from the related *sparse vector recovery* problem that suggest alternative relaxation approaches. These results indicate that it might be even more advantageous to solve the non-convex problem

$$\min F^p(X) := \|X\|_{S_p}^p \text{ subject to } \Phi(X) = Y, \tag{7}$$

for $0 < p < 1$, i.e., minimizing the p-th power of the Schatten-p quasi-norms under the affine constraint. Heuristically, the choice of $p < 1$ relatively small can be motivated by the observation that by the definition (4) of the Schatten-p quasi-norm

$$\|X\|_{S_p}^p \xrightarrow{p \to 0} \operatorname{rank}(X) =: \|X\|_{S_0}.$$

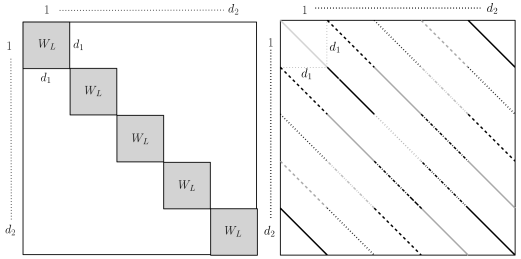
The above consideration suggests that the solution of (7) might be closer to (5) than (6) for small p. On the other hand, again, it is in general computationally intractable to find a global minimum of the non-convex optimization problem (7) if $p < 1$. Therefore it is a

natural and very relevant question to ask which optimization algorithm to use to find global minimizers of (7).

In this paper, we discuss an algorithm striving to solve (7) that is based on the following observations: Assume for the moment that we are given a square matrix $X \in M_{d_1 \times d_2}$ with $d_1 = d_2$ of full rank. Then, we can rewrite the p-th power of its Schatten-p quasi-norm as a squared weighted Frobenius norm, or, using Kronecker product notiation as explained in Appendix A, as a squared weighted ℓ_2-norm (if we use the vectorized notation X_{vec}): It turns out that

(i) $$\|X\|_{S_p}^p = \mathrm{tr}[(XX^*)^{\frac{p}{2}}] = \mathrm{tr}[(XX^*)^{\frac{p-2}{2}}(XX^*)] = \mathrm{tr}(W_L XX^*) = \|W_L^{\frac{1}{2}} X\|_F^2$$
$$= \|X\|_{F(W_L)}^2 = \|(\mathbf{I}_{d_2} \otimes W_L)^{\frac{1}{2}} X_{\mathrm{vec}}\|_{\ell_2}^2 = \|X_{\mathrm{vec}}\|_{\ell_2(\mathbf{I}_{d_2} \otimes W_L)}^2,$$
where W_L is the symmetric weight matrix $(XX^*)^{\frac{p-2}{2}}$ in $M_{d_1 \times d_1}$ and $\mathbf{I}_{d_2} \otimes W_L$ is the block diagonal weight matrix in $M_{d_1 d_2 \times d_1 d_2}$ with d_2 instances of W_L on the diagonal blocks, but also that

(ii) $$\|X\|_{S_p}^p = \mathrm{tr}[(X^*X)^{\frac{p}{2}}] = \mathrm{tr}[(X^*X)(X^*X)^{\frac{p-2}{2}}] = \mathrm{tr}(X^*XW_R) = \|XW_R^{\frac{1}{2}}\|_F^2$$
$$= \|X^*\|_{F(W_R)}^2 = \|(W_R \otimes \mathbf{I}_{d_1})^{\frac{1}{2}} X_{\mathrm{vec}}\|_{\ell_2}^2 = \|X_{\mathrm{vec}}\|_{\ell_2(W_R \otimes \mathbf{I}_{d_1})}^2,$$
where W_R is the symmetric weight matrix $(X^*X)^{\frac{p-2}{2}}$ in $M_{d_2 \times d_2}$. It follows from the definition of the Kronecker product that the weight matrix $W_R \otimes \mathbf{I}_{d_1} \in M_{d_1 d_2 \times d_1 d_2}$ is a block matrix of diagonal blocks of the type $\mathrm{diag}\big((W_R)_{ij}, \ldots, (W_R)_{ij}\big) \in M_{d_1 \times d_1}$, $i, j \in [d_2]$.



(a) $\mathbf{I}_{d_2} \otimes W_L$ (b) $W_R \otimes \mathbf{I}_{d_1}$

Figure 1: Sparsity structure of the weight matrices $\in M_{d_1 d_2 \times d_1 d_2}$

The sparsity structures of $\mathbf{I}_{d_2} \otimes W_L$ and $W_R \otimes \mathbf{I}_{d_1}$ are illustrated in Fig. 1. Note that a representation of $\|X\|_{S_p}^p$ by squares of Frobenius norms can be achieved by multiplying X by $W_L^{\frac{1}{2}}$ from the left in (i), or by $W_R^{\frac{1}{2}}$ from the right in (ii).

The above calculations are not well-defined if X is not of full rank or if $d_1 \neq d_2$, since in these cases at least one of the matrices $XX^* \in M_{d_1 \times d_1}$ or $X^*X \in M_{d_2 \times d_2}$ is singular, prohibiting the definition of the matrices $W_R = (X^*X)^{\frac{p-2}{2}}$ or $W_L = (XX^*)^{\frac{p-2}{2}}$ for $p < 2$. However, these issues can be overcome by introducing a smoothing parameter $\epsilon > 0$ and

smoothed weight matrices $W_L(X, \epsilon) \in M_{d_1 \times d_1}$ and $W_R(X, \epsilon) \in M_{d_2 \times d_2}$ defined by

$$W_L(X, \epsilon) := (XX^* + \epsilon^2 \mathbf{I}_{d_1})^{\frac{p-2}{2}}, \tag{8}$$
$$W_R(X, \epsilon) := (X^*X + \epsilon^2 \mathbf{I}_{d_2})^{\frac{p-2}{2}}. \tag{9}$$

Remark 1 *The weight matrices $W_L(X, \epsilon)$ and $W_R(X, \epsilon)$ are symmetric and positive definite.*

The possibility to rewrite the p-th power of the Schatten-p of a matrix as a squared weighted Frobenius norm gives rise to the general strategy of IRLS algorithms for low-rank matrix recovery: Weighted least squares problems of the type

$$\min_{\substack{X \in M_{d_1 \times d_2} \\ \Phi(X) = Y}} \|X\|_{F(W_L)}^2 \quad \text{or} \quad \min_{\substack{X \in M_{d_1 \times d_2} \\ \Phi(X) = Y}} \|X^*\|_{F(W_R)}^2$$

are solved and weight matrices W_L are updated alternatingly, leading to the algorithms column-reweighting `IRLS-col` and row-reweighting `IRLS-row`, respectively (Mohan and Fazel, 2012; Fornasier et al., 2011).

2.3 Averaging of weight matrices

While the algorithms `IRLS-col` and `IRLS-row` provide a tractable local minimization strategy of smoothed Schatten-p functionals under the linear constraint, we argue that it is suboptimal to follow either one of the two approaches as they do not exploit the symmetry of the problem in an optimal way: They *either* use low-rank information in the column space *or* in the row space.

A first intuitive approach towards a symmetric exploitation of the low-rank structure is inspired by the following identity, by combing the calculations (i) and (ii) carried out in Section 2.2.

Lemma 2 *Let $0 < p \leqslant 2$ and $X \in M_{d_1 \times d_2}$ with $d = d_1 = d_2$ be a matrix of full rank. Then*

$$\|X\|_{S_p}^p = \frac{1}{2}\left(\|W_L^{\frac{1}{2}} X\|_F^2 + \|XW_R^{\frac{1}{2}}\|_F^2\right) = \left\|\left(\frac{W_L \oplus W_R}{2}\right)^{\frac{1}{2}} X_{\mathrm{vec}}\right\|_{\ell_2}^2 = \|X_{\mathrm{vec}}\|_{\ell_2(W_{(arith)})}^2,$$

where

$$\frac{1}{2}\left(\mathbf{I}_{d_2} \otimes W_L + W_R \otimes \mathbf{I}_{d_1}\right) = \frac{W_L \oplus W_R}{2} =: W_{(arith)}$$

is the arithmetic mean matrix of the symmetric and positive definite weight matrices $\mathbf{I}_{d_2} \otimes W_L$ and $W_R \otimes \mathbf{I}_{d_1}$, $W_L := (XX^*)^{\frac{p-2}{2}}$, and $W_R := (X^*X)^{\frac{p-2}{2}}$.

Unfortunately, the introduction of arithmetic mean weight matrices does not prove to be particularly advantageous compared to one-sided reweighting strategies. Convincing improvements could be noted neither in numerical experiments nor in the theoretical investigations for the convergence rate of IRLS for low-rank matrix recovery, cf. also Section 5.2 and Remark 22.

In contrast, we want to promote the usage of the *harmonic mean of the weight matrices* $\mathbf{I}_{d_2} \otimes W_L$ and $W_R \otimes \mathbf{I}_{d_1}$, i.e., weight matrices of the type $2\left(W_R^{-1} \otimes \mathbf{I}_{d_1} + \mathbf{I}_{d_2} \otimes W_L^{-1}\right)^{-1} = 2\left(W_L^{-1} \oplus W_R^{-1}\right)^{-1} =: W_{(\text{harm})}$. In the remaining parts of the paper, we explain why $W_{(\text{harm})}$ is able to significantly outperform other weighting variants both theoretically and practically.

The following lemma verifies that the harmonic mean $W_{(\text{harm})}$ of the weight matrices $\mathbf{I}_{d_2} \otimes W_L$ and $W_R \otimes \mathbf{I}_{d_1}$ leads to a legitimate reformulation of the Schatten-p quasi-norm power, as it we already saw for the arithmetic mean $W_{(\text{arith})}$.

Lemma 3 *Let $0 < p \leqslant 2$ and $X \in \mathbb{C}^{d_1 \times d_2}$ with $d = d_1 = d_2$ be a full rank matrix. Then*

$$\|X\|_{S_p}^p = 2\left\|\left(W_L^{-1} \oplus W_R^{-1}\right)^{-\frac{1}{2}} X_{\text{vec}}\right\|_{\ell_2}^2 = \|X_{\text{vec}}\|_{\ell_2(W_{(harm)})}^2,$$

where

$$2\left(W_R^{-1} \otimes \mathbf{I}_{d_1} + \mathbf{I}_{d_2} \otimes W_L^{-1}\right)^{-1} = 2\left(W_L^{-1} \oplus W_R^{-1}\right)^{-1} =: W_{(harm)}$$

is the harmonic mean matrix of the symmetric and positive definite weight matrices $\mathbf{I}_{d_2} \otimes W_L$ and $W_R \otimes \mathbf{I}_{d_2}$, $W_L := (XX^*)^{\frac{p-2}{2}}$ and $W_R := (X^*X)^{\frac{p-2}{2}}$.

Proof Let $X = U\Sigma V^* = \sum_{i=1}^d \sigma_i u_i v_i^* \in M_{d\times d}$ be the singular value decomposition of X. Therefore for the vectorized version, $X_{\text{vec}} = (V \otimes U)\Sigma_{\text{vec}}$ holds true. By the definitions of W_L and W_R, we can write $W_L^{-1} = \sum_{i=1}^d \sigma_i^{2-p} u_i u_i^*$ and $W_R^{-1} = \sum_{i=1}^d \sigma_i^{2-p} v_i v_i^*$. Using the Kronecker sum inversion formula of Lemma 23 in Appendix A, we obtain

$$\begin{aligned}
\|X_{\text{vec}}\|_{\ell_2(W_{(\text{harm})})}^2 &= \|W_{(\text{harm})}^{\frac{1}{2}} X_{\text{vec}}\|_{\ell_2}^2 = 2\left\|\left(W_L^{-1} \oplus W_R^{-1}\right)^{-\frac{1}{2}} X_{\text{vec}}\right\|_{\ell_2}^2 \\
&= 2\,\text{tr}\left(\left(\left(W_L^{-1} \oplus W_R^{-1}\right)^{-1} X_{\text{vec}}\right)_{\text{mat}}^* X\right) \\
&= \sum_{i=1}^d \sum_{j=1}^d \sum_{k=1}^d \frac{2\sigma_k}{\sigma_i^{2-p} + \sigma_j^{2-p}} v_j v_i^* v_k u_k^* u_i u_i^* \sum_{l=1}^d \sigma_l u_l v_l^* \\
&= 2\left(\sum_{i=1}^d \frac{\sigma_i^2}{2\sigma_i^{2-p}}\right) = \|X\|_{S_p}^p,
\end{aligned}$$

which finishes the proof. ∎

3. Harmonic mean iteratively reweighted least squares algorithm

In this section, we use this idea to formulate a new iteratively reweighted least squares algorithm for low-rank matrix recovery. The so-called *harmonic mean iteratively reweighted least squares* algorithm (`HM-IRLS`) solves a sequence of weighted least squares problems to recover a low-rank matrix $X_0 \in M_{d_1\times d_2}$ from few linear measurements $\Phi(X_0) \in \mathbb{C}^m$. The weight matrices appearing in the least squares problems can be seen as the harmonic mean of the weight matrices in (8) and (9), i.e., the ones used by `IRLS-col` and `IRLS-row`.

More precisely, for $0 < p \leqslant 1$ and $d = \min(d_1, d_2)$, $D = \max(d_1, d_2)$, given a non-increasing sequence of non-negative real numbers $(\epsilon^{(n)})_{n=1}^\infty$ and the sequence of iterates $(X^{(n)})_{n=1}^\infty$ produced by the algorithm, we update our weight matrices such that

$$\widetilde{W}^{(n)} = 2\left[U^{(n)}(\bar{\Sigma}_{d_1}^{(n)})^{2-p}U^{(n)*} \oplus V^{(n)}(\bar{\Sigma}_{d_2}^{(n)})^{2-p}V^{(n)*}\right]^{-1}, \tag{10}$$

with the diagonal matrices $\bar{\Sigma}_{d_t}^{(n)} \in M_{d_t\times d_t}$ for $d_t = \{d_1, d_2\}$ such that

$$(\bar{\Sigma}_{d_t}^{(n)})_{ii} = \begin{cases} (\sigma_i(X^{(n)})^2 + \epsilon^{(n)2})^{\frac{1}{2}} & \text{if } i \leqslant d, \\ 0 & \text{if } d < i \leqslant D, \end{cases} \tag{11}$$

and the matrices $U^{(n)} \in \mathcal{U}_{d_1}$ and $V^{(n)} \in \mathcal{U}_{d_2}$ containing the left and right singular vectors of $X^{(n)}$ in its columns, respectively.

We note that this definition of $\widetilde{W}^{(n)}$ can be seen as a stabilized version of the harmonic mean weight matrix $W_{(\text{harm})}$ of Lemma 3. This stabilization is necessary as $\widetilde{W}^{(n)}$ becomes very ill-conditioned as soon as some of the singular values of $X^{(n)}$ approach zero and, related to that, $(X^{(n)}X^{(n)*})^{\frac{2-p}{2}} \oplus (X^{(n)*}X^{(n)})^{\frac{2-p}{2}}$ would even be singular as soon as $X^{(n)}$ is not of full rank.

Additionally, for the formulation of the algorithm it is convenient to define the linear operator $(\widetilde{\mathcal{W}}^{(n)})^{-1} : M_{d_1\times d_2} \to M_{d_1\times d_2}$ for any $n \in \mathbb{N}$ such that

$$(\widetilde{\mathcal{W}}^{(n)})^{-1}(X) := \frac{1}{2}\left[U^{(n)}(\bar{\Sigma}_{d_1}^{(n)})^{2-p}U^{(n)*}X + XV^{(n)}(\bar{\Sigma}_{d_2}^{(n)})^{2-p}V^{(n)*}\right], \tag{12}$$

describing the operation of the inverse of $\widetilde{W}^{(n)}$ on $M_{d_1\times d_2}$.

Finally, `HM-IRLS` can be formulated in pseudo code as follows.

Algorithm 1 Harmonic Mean IRLS for low-rank matrix recovery (`HM-IRLS`)

Input: A linear map $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$, image $Y = \Phi(X_0)$ of the ground truth matrix $X_0 \in M_{d_1\times d_2}$, rank estimate $\tilde{r}$, non-convexity parameter $0 < p \leqslant 1$.
Output: Sequence $(X^{(n)})_{n=1}^{n_0} \subset M_{d_1\times d_2}$.
Initialize $n = 0$, $\epsilon^{(0)} = 1$ and $\widetilde{W}^{(0)} = \mathbf{I}_{d_1d_2} \in M_{d_1d_2\times d_1d_2}$.
repeat

$$X^{(n+1)} = \underset{\Phi(X)=Y}{\arg\min} \|X_{\text{vec}}\|_{\ell_2(\widetilde{W}^{(n)})}^2 = ((\widetilde{\mathcal{W}}^{(n)})^{-1}\big(\Phi^*\big((\Phi \circ ((\widetilde{\mathcal{W}}^{(n)})^{-1} \circ \Phi^*)^{-1}(Y)\big)\big), \tag{13}$$

$$\epsilon^{(n+1)} = \min\left(\epsilon^{(n)}, \sigma_{\tilde{r}+1}(X^{(n+1)})\right), \tag{14}$$

$$\widetilde{W}^{(n+1)} = 2\left[U^{(n+1)}(\bar{\Sigma}_{d_1}^{(n+1)})^{2-p}U^{(n+1)*} \oplus V^{(n+1)}(\bar{\Sigma}_{d_2}^{(n+1)})^{2-p}V^{(n+1)*}\right]^{-1}, \tag{15}$$

where $U^{(n+1)} \in \mathcal{U}_{d_1}$ and $V^{(n+1)} \in \mathcal{U}_{d_2}$ are matrices containing the left and right singular vectors of $X^{(n+1)}$ in its columns, and the $\bar{\Sigma}_{d_t}^{(n+1)}$ are defined for $t \in \{1, 2\}$ according to (11).

$$n = n + 1,$$

until *stopping criterion is met.*
Set $n_0 = n$.

From a practical point of view, it is beneficial that the explicit calculation of the very large weight matrices $\widetilde{W}^{(n)} \in H_{d_1 d_2 \times d_1 d_2}$ (cf. (15)) is not necessary in implementations of Algorithm 1. As suggested by formulas (12) and (13), it can be seen that just the operation of its inverse $(\widetilde{W}^{(n)})^{-1}$ is needed, which can be implemented by matrix-matrix multiplications on the space $M_{d_1 \times d_2}$: For matrices $X, \tilde{X} \in M_{d_1 \times d_2}$, we have that $\widetilde{W}^{(n)} X_{\mathrm{vec}} = \tilde{X}_{\mathrm{vec}}$ if and only if $X_{\mathrm{vec}} = (\widetilde{W}^{(n)})^{-1} \tilde{X}_{\mathrm{vec}}$, which can be written in matrix variables as

$$X = \frac{1}{2}\left[U^{(n)} (\bar{\Sigma}_{d_1}^{(n)})^{2-p} U^{(n)*} \tilde{X} + \tilde{X} V^{(n)} (\bar{\Sigma}_{d_2}^{(n)})^{2-p} V^{(n)*}\right].$$

The last equivalence is due to the definitions of $\widetilde{W}^{(n)}$ and the Kronecker sum, cf. (15) and Appendix A.

Note that the smoothing parameters $\epsilon^{(n)}$ are chosen in dependence on a rank estimate $\tilde{r}$ here, which will be an important ingredient for the theoretical analysis of the algorithm. In practice, however, other choices of non-increasing sequences of non-negative real numbers $(\epsilon^{(n)})_{n=1}^{\infty}$ are possible and can as well lead to (a maybe even faster) convergence when tuned appropriately.

We refer to Section 5.4 for a further discussion of implementation details.

Example With a simple example, we illustrate the versatility of `HM-IRLS`: Let $d_1 = d_2 = 4$, and assume that we want to reconstruct the rank-1 matrix

$$X_0 = uv^* = \begin{pmatrix} 1 \\ 10 \\ -2 \\ 0.1 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 10 & 20 & 30 & 40 \\ -2 & -4 & -6 & -8 \\ 0.1 & 0.2 & 0.3 & 0.4 \end{pmatrix}$$

from $m = d_f = r(d_1 + d_2 - r) = 7$ sampled entries $\Phi(X_0)$, where Φ is the linear map $\Phi : M_{4\times 4} \to \mathbb{C}^7$, $\Phi(X) = \big(X_{2,1},\ X_{4,1},\ X_{3,2},\ X_{4,2},\ X_{4,3},\ X_{1,4},\ X_{2,4}\big)$. Since the linear map Φ samples some entries of matrices in $M_{4\times 4}$ and does not see the others, this is an instance of the problem that is called *matrix completion*.

In general, reconstructing a $(d_1 \times d_2)$ rank-r matrix from $m = r(d_1 + d_2 - r)$ entries is a hard problem, as it is known that if $m < r(d_1 + d_2 - r)$, there is always more than one matrix X such that $\Phi(X) = \Phi(X_0)$, and even for equality, the property that Φ is invertible on (most) rank-r matrices might be hard to verify (Király et al., 2015).

It can be argued that the specific matrix completion problem we consider is in some sense a hard one, since, e.g., the deterministic sufficient condition for unique completability of (Pimentel-Alarcón et al., 2016, Theorem 2) is not fulfilled (less then 2 observed entries in the third column), and since the classical coherence parameters $\mu(u) = d_1 \max_{1\leqslant i\leqslant 4} \frac{\|uu^* e_i\|_2^2}{\|u\|_2^4} \approx 3.81$ and $\mu(v) = d_2 \max_{1\leqslant i\leqslant 4} \frac{\|vv^* e_i\|_2^2}{\|v\|_2^4} \approx 2.13$ that are used to analyze the behavior of many matrix completion algorithms (Candès and Recht, 2009; Jain et al., 2013) are quite large, with $\mu(u)$ being quite close to the maximal value of 4.

On the other hand, as the problem is small and X_0 has rank $r = 1$, it is possible to impute the missing values of

$$\begin{pmatrix} * & * & * & 4 \\ 10 & * & * & 40 \\ * & -4 & * & * \\ 0.1 & 0.2 & 0.3 & * \end{pmatrix}$$

by solving very simple linear equations, since, for example, $X_{4,4} = u_4 v_4$, $X_{2,1} = u_2 v_1$, $X_{2,4} = u_2 v_4$, and $X_{4,1} = u_4 v_1$, and therefore $X_{4,4} = \frac{X_{4,1} X_{2,4}}{X_{2,1}} = 0.4$. This shows that the only rank-1 matrix compatible with $\Phi(X_0)$ is X_0.

It turns out that—without using the combinatorial simplicity of the problem—the classical `NNM` does not solve the problem, as the nuclear norm minimizer (solution of (6) for $Y = \Phi(X_0)$) produced by the semidefinite program of the convex optimization package `CVX` (Grant and Boyd, 2014) converges to

$$\bar{X}_{\mathrm{nuclear}} \approx \begin{pmatrix} 1 & 0.023 & 0.041 & 4 \\ 10 & 0.232 & 0.411 & 40 \\ -0.056 & -4 & -0.200 & -0.226 \\ 0.1 & 0.2 & 0.3 & 0.400 \end{pmatrix},$$

a matrix with $45.74 \approx \|\bar{X}_{\mathrm{nuclear}}\|_{S_1} < \|X_0\|_{S_1} = \sigma_1(X_0) \approx 56.13$ and a relative Frobenius error of $\frac{\|\bar{X}_{\mathrm{nuclear}} - X_0\|_F}{\|X_0\|_F} = 0.661$.

On the other hand, `HM-IRLS` is able to solve the problem—if p is chosen small enough—with very high precision already after few iterations, for example, up to a relative error of $4.18 \cdot 10^{-13}$ after 24 iterations if $p = 0.1$. This is in contrast to the behavior of `IRLS-col`, `IRLS-row` and also to the behavior of `AM-IRLS`, the IRLS variant that uses weight matrices derived from the *arithmetic mean* of the weights of `IRLS-col` and `IRLS-row`, cf. Lemma 2. The iterates $X^{(n)}$ for iteration $n = 2000$ of these algorithms exhibit relative errors of 0.240, 0.489 and 0.401, respectively, for the choice of $p = 0.1$. Furthermore, there is no choice of p that would lead to a convergence to X_0.

To understand this very different behavior, we note that the n-th iterate of any of the four IRLS variants can be written, using Appendix A, in a concise way as

$$X^{(n+1)} = \operatorname*{arg\,min}_{\Phi(X)=Y} \langle X_{\mathrm{vec}}, W^{(n)} X_{\mathrm{vec}} \rangle, \tag{16}$$

where

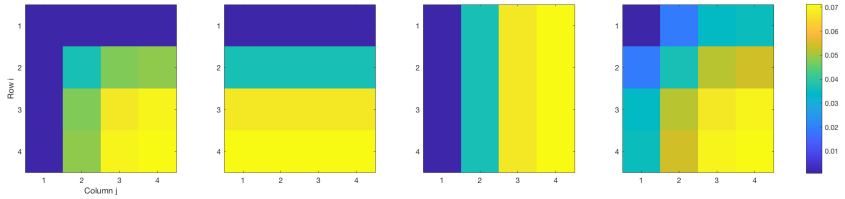
$$\langle X_{\mathrm{vec}}, W^{(n)} X_{\mathrm{vec}} \rangle = \langle X, U^{(n)} \big[H^{(n)} \circ (U^{(n)*} X V^{(n)})\big] V^{(n)*} \rangle_F = \sum_{i,j=1}^{4} H_{ij}^{(n)} |\langle u_i^{(n)}, X v_j^{(n)} \rangle|^2 \tag{17}$$

with $X^{(n)} = U^{(n)} \Sigma^{(n)} V^{(n)*} = \sum_{i=1}^{4} \sigma_i^{(n)} u_i^{(n)} v_i^{(n)}$ being the SVD of $X^{(n)}$, and

$$H_{i,j}^{(n)} = \begin{cases} 2\big[\big((\sigma_i^{(n)})^2 + (\epsilon^{(n)})^2\big)^{\frac{2-p}{2}} + \big((\sigma_j^{(n)})^2 + (\epsilon^{(n)})^2\big)\big)^{\frac{2-p}{2}}\big]^{-1} & \text{for } \texttt{HM-IRLS}, \\ \big((\sigma_i^{(n)})^2 + (\epsilon^{(n)})^2\big)^{\frac{p-2}{2}} & \text{for } \texttt{IRLS-col}, \\ \big((\sigma_j^{(n)})^2 + (\epsilon^{(n)})^2\big)^{\frac{p-2}{2}} & \text{for } \texttt{IRLS-row}, \text{ and} \\ 0.5 \cdot \big[\big((\sigma_i^{(n)})^2 + (\epsilon^{(n)})^2\big)^{\frac{p-2}{2}} + \big((\sigma_i^{(n)})^2 + (\epsilon^{(n)})^2\big)^{\frac{p-2}{2}}\big] & \text{for } \texttt{AM-IRLS}, \end{cases}$$

for $i, j \in \{1, 2, 3, 4\}$ and $\epsilon^{(n)} = \min(\sigma_2^{(n)}, \epsilon^{(n-1)})$.

The values of the matrix $H^{(1)}$ of weight coefficients after the first iteration in the above example are visualized in Figure 2, for each of the four IRLS versions above.



(a) `HM-IRLS` (b) `IRLS-col` (c) `IRLS-row` (d) `AM-IRLS`

Figure 2: Values of the matrix $H^{(1)}$ of "weight coefficients" corresponding to the orthonormal basis $(u_i^{(1)} v_j^{(1)*})_{i,j=1}^4$ after the first iteration in the example

The intuition for the superior behavior of `HM-IRLS` is now the following: Since large entries of $H^{(n)}$ *penalize* the corresponding parts of the space $M_{d_1 \times d_2} = \operatorname{span}\{u_i^{(n)} v_j^{(n)*}, i \in [d_1], j \in [d_2]\}$ in the minimization problem (16), large areas of *blue* and *dark blue* in Figure 2 indicate a benign optimization landscape where the minimizer $X^{(n+1)}$ of (16) *is able to improve considerably* on the previous iterate $X^{(n)}$.

In particular, it can be seen that in the case of `HM-IRLS`, the penalties on the *whole direct sum of column and row space of the best rank-r approximation of $X^{(n)}$*

$$T^{(n)} := \left\{ \begin{pmatrix} u_1^{(n)}, & \dots, & u_r^{(n)} \end{pmatrix} Z_1^* + Z_2 \begin{pmatrix} v_1^{(n)}, & \dots, & v_r^{(n)} \end{pmatrix}^* : Z_1 \in M_{d_1 \times r}, Z_2 \in M_{d_2 \times r} \right\},$$

are small compared to the other penalites, since the coefficients of $H^{(1)}$ corresponding to $T^{(1)}$ are exactly the ones in the first row and first column of the (4×4) matrices in Figure 2—a contrast that becomes more and more pronounced as $X^{(n)}$ approaches the rank-r ground truth X_0 (with $r = 1$ in the example).

On the other hand, `IRLS-col`, `IRLS-row` and `AM-IRLS` only have small coefficients on smaller parts of $T^{(n)}$, which, from a global perspective, explains why their usage might lead to non-global minima of the Schatten-p objective.

We note that the space $T^{(n)}$ plays also an important role in Riemannian optimization approaches for matrix recovery problems (see Vandereycken, 2013), since it is also the tangent space of the smooth manifold of rank-r matrices at the best rank-r approximation of $X^{(n)}$.

4. Convergence results

In the following part, we state our main theoretical results about convergence properties of the algorithm `HM-IRLS`. Furthermore, their relation to existing results for `IRLS-col` and `IRLS-row` is discussed.

It cannot be expected that a low-rank matrix recovery algorithm like `HM-IRLS` succeeds to converge to a low-rank matrix without any assumptions on the measurement operator Φ that defines the recovery problem (5). For the purpose of the convergence analysis of `HM-IRLS`, we introduce the following strong *Schatten-p null space property* (Fornasier et al., 2011; Oymak et al., 2011; Foucart and Rauhut, 2013).

Definition 4 (Strong Schatten-p null space property) *Let $0 < p \leqslant 1$. We say that a linear map $\Phi : M_{d_1 \times d_2} \to \mathbb{C}^m$ fulfills the strong Schatten-p null space property (Schatten-p NSP) of order r with constant $0 < \gamma_r \leqslant 1$ if*

$$\left(\sum_{i=1}^{r} \sigma_i^2(X) \right)^{p/2} < \frac{\gamma_r}{r^{1-\frac{p}{2}}} \left(\sum_{i=r+1}^{d} \sigma_i^p(X) \right) \tag{18}$$

for all $X \in \mathcal{N}(\Phi) \backslash \{0\}$.

Intuitively explained, if a map Φ fulfills the strong Schatten-p null space property of order r, there are no rank-r matrices in the null space and all the elements of the null space must not have a quickly decaying spectrum.

Null space properties have already been used to guarantee the success of nuclear norm minimization (6), or Schatten-1 minimization in our terminology, for solving the low-rank matrix recovery problem (Recht et al., 2011).

We note that the definitions of Schatten-p null space properties are quite analogous to the ℓ_p-null space property in classical compressed sensing (Foucart and Rauhut, 2013, Theorem 4.9), applied to the vector of singular values. In particular, (18) implies that

$$\sum_{i=1}^{r} \sigma_i^p(X) < \sum_{i=r+1}^{d} \sigma_i^p(X) \qquad \text{for all } X \in \mathcal{N}(\Phi) \backslash \{0\}, \tag{19}$$

since $\|X\|_{S_p} \leqslant r^{1/p - 1/2} \|X\|_{S_2}$ for X that is rank-r. This, in turn, ensures the existence of unique solutions to (7) if $Y = \Phi(X_0)$ are the measurements of a low-rank matrix X_0.

Proposition 5 (Foucart (2018)) *Let $\Phi : M_{d_1 \times d_2} \to \mathbb{C}^m$ be a linear map, let $0 < p \leqslant 1$ and $r \in \mathbb{N}$. Then every matrix $X_0 \in M_{d_1 \times d_2}$ such that $\operatorname{rank}(X_0) \leqslant r$ and $\Phi(X_0) = Y \in \mathbb{C}^m$ is the* unique *solution of Schatten-p minimization* (7) *if and only if Φ fulfills* (19).

Remark 6 *The sufficiency of the Schatten-p NSP* (19) *in Proposition 5 has already been pointed out by Oymak et al. (2011). The necessity as stated in the theorem, however, is due to a recent generalization of Mirsky's singular value inequalities to concave functions (Audenaert, 2014; Foucart, 2018).*

It can be seen that the (weak) Schatten-p NSP of (19) is a *stronger* property for larger p in the sense that if $0 < p' \leqslant p \leqslant 1$, the Schatten-$p$ property implies the Schatten-p' property. Very related to this, it can be seen that for any $0 < p \leqslant 1$, the strong Schatten-p null space property is implied by a sufficiently small *rank restricted isometry constant* δ_r, which is a classical tool in the analysis of low-rank matrix recovery algorithms (Recht et al., 2010; Candès and Plan, 2011).

Definition 7 (Restricted isometry property (RIP)) *The* restricted isometry constant $\delta_r > 0$ of order r *of the linear map* $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ *is defined as the smallest number such that*

$$(1-\delta_r)\|X\|_F^2 \leqslant \|\Phi(X)\|_{\ell_2}^2 \leqslant (1+\delta_r)\|X\|_F^2$$

for all matrices $X \in M_{d_1\times d_2}$ *of rank at most* r.

Indeed, it follows from the proof of Chavez-Dominguez and Kutzarova (2015, Theorem 4.1) that a restricted isometry constant of order $2r$ such that $\delta_{2r} < \frac{2}{\sqrt{2}+3} \approx 0.4531$ implies the strong Schatten-p NSP of order r with a constant $\gamma_r < 1$ for any $0 < p \leqslant 1$. More precisely, it can be seen that $\delta_{2r} < \frac{2}{\sqrt{2}+3}$ implies that the strong Schatten-p NSP (18) of order r holds with the constant $\gamma_r = \frac{(\sqrt{2}+1)^p}{2^p}\frac{\delta_{2r}^p}{(1-\delta_{2r})^p}$.

Linear maps that are instances drawn from certain random models are known to fulfill the restricted isometry property with high probability if the number of measurements is sufficiently large (Davenport and Romberg, 2016), and, a fortiori, the Schatten-p null space property. In particular, this is true for *(sub-)Gaussian* linear measurement maps $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ whose matrix representation is such that

$$\frac{1}{\sqrt{m}}\widetilde{\Phi} \in \mathbb{C}^{m\times d_1 d_2}, \text{ where } \widetilde{\Phi} \text{ has i.i.d. standard (sub-)Gaussian entries,} \tag{20}$$

as it is summarized in the following lemma.

Lemma 8 *For any* $0 < p \leqslant 1$, $0 < \gamma < 1$ *and any (sub-)Gaussian random operator* $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ *(e.g. as defined in* (20)*), there exist constants* $C_1 > 1$, $C_2 > 0$ *such that if* $m \geqslant C_1 r(d_1+d_2)$*, the strong Schatten-p null space property* (18) *of order* r *with constant* $\gamma_r < \gamma$ *is fulfilled with probability at least* $1 - e^{-C_2 m}$.

4.1 Local convergence for $p < 1$

In this section, we provide a convergence analysis for `HM-IRLS` covering several aspects. We show that the algorithm converges to stationary points of a smoothed Schatten-p functional g_ϵ^p as in (21) without any additional assumptions on the measurement map Φ. Such guarantees have already been obtained for IRLS algorithms with one-sided reweighting as `IRLS-col` and `IRLS-row`, in particular for $p = 1$ by Fornasier et al. (2011) and for $0 < p \leqslant 1$ by Mohan and Fazel (2012).

Beyond that, assuming the measurement operator fulfills an appropriate Schatten-p null space property as defined in Definition 4, we show the a-posteriori exact recovery statement that `HM-IRLS` converges to the low-rank matrix X_0 if $\lim_{n\to\infty} \epsilon_n = 0$, which only was shown for one-sided `IRLS` for the case $p = 1$ by Fornasier et al. (2011).

Moreover, we provide a local convergence guarantee stating that `HM-IRLS` recovers the low-rank matrix X_0 if we obtain an iterate $X^{(\bar{n})}$ that is close enough to X_0, which is novel for IRLS algorithms.

Let $0 < p \leqslant 1$ and $\epsilon > 0$. To state the theorem, we introduce the *ϵ-perturbed Schatten-p functional* $g_\epsilon^p : M_{d_1\times d_2} \to \mathbb{R}_{\geqslant 0}$ such that

$$g_\epsilon^p(X) = \sum_{i=1}^{d} (\sigma_i(X)^2 + \epsilon^2)^{\frac{p}{2}}, \tag{21}$$

where $\sigma(X) \in \mathbb{R}^d$ denotes the vector of singular values of $X \in M_{d_1\times d_2}$.

Theorem 9 *Let* $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ *be a linear operator and* $Y \in \mathrm{Ran}(\Phi)$ *a vector in its range. Let* $(X^{(n)})_{n\geqslant 1}$ *and* $(\epsilon^{(n)})_{n\geqslant 1}$ *be the sequences produced by Algorithm 1 for input parameters* Φ, Y, r *and* $0 < p \leqslant 1$*, let* $\epsilon = \lim_{n\to\infty} \epsilon^{(n)}$.

(i) If $\epsilon = 0$ *and if* Φ *fulfills the strong Schatten-p NSP* (18) *of order* r *with constant* $0 < \gamma_r < 1$*, then the sequence* $(X^{(n)})_{n\geqslant 1}$ *converges to a matrix* $\bar{X} \in M_{d_1\times d_2}$ *of rank at most* r *that is the unique minimizer of the Schatten-p minimization problem* (7)*. Moreover, there exists an absolute constant* $\hat{C} > 0$ *such that for any* X *with* $\Phi(X) = Y$ *and any* $\widetilde{r} \leqslant r$*, it holds that*

$$\|X - \bar{X}\|_F^p \leqslant \frac{\hat{C}}{r^{1-p/2}}\beta_{\widetilde{r}}(X)_{S_p},$$

where $\hat{C} = \frac{2^{p+1}\gamma_r^{1-p/2}}{1-\gamma_r}$ *and* $\beta_{\widetilde{r}}(X)_{S_p}$ *is the best rank-*$\widetilde{r}$ *Schatten-p approximation error of* X*, i.e.,*

$$\beta_{\widetilde{r}}(X)_{S_p} := \inf\left\{\|X - \widetilde{X}\|_{S_p}^p,\ \widetilde{X} \in M_{d_1\times d_2} \text{ has rank } \widetilde{r}\right\}. \tag{22}$$

(ii) If $\epsilon > 0$*, then each accumulation point* $\bar{X}$ *of* $(X^{(n)})_{n\geqslant 1}$ *is a stationary point of the* ϵ*-perturbed Schatten-p functional* g_ϵ^p *of* (21) *under the linear constraint* $\Phi(X) = Y$*. If additionally* $p = 1$*, then* $\bar{X}$ *is the unique global minimizer of* g_ϵ^p.

(iii) Assume that there exists a matrix $X_0 \in M_{d_1\times d_2}$ *with* $\Phi(X_0) = Y$ *such that* $\mathrm{rank}(X_0) = r \leqslant \frac{\min(d_1,d_2)}{2}$*, a constant* $0 < \zeta < 1$ *and an iteration* $\bar{n} \in \mathbb{N}$ *such that*

$$\|X^{(\bar{n})} - X_0\|_{S_\infty} \leqslant \zeta\sigma_{\widetilde{r}}(X_0)$$

and $\epsilon^{\bar{n}} = \sigma_{r+1}(X^{\bar{n}})$*. If* Φ *fulfills the strong Schatten-p NSP of order* $2r$ *with* $\gamma_{2r} < 1$ *and if the condition number* $\kappa = \frac{\sigma_1(X_0)}{\sigma_r(X_0)}$ *of* X_0 *and* ζ *are sufficiently small (see condition* (25) *and formula* (26)*), then*

$$X^{(n)} \to X_0 \quad \text{for } n \to \infty.$$

It is important to note that by using Lemma 8, it follows that the assertions of Theorem 9(i) and (iii) hold for (sub-)Gaussian operators (20) with high probability in the regime of measurements of optimal sample complexity order. In particular, there exist constant oversampling factors $\rho_1, \rho_2 \geqslant 1$ such that the assertions of (i) and (iii) hold with high probability if $m > \rho_k r(d_1+d_2)$, $k \in \{1,2\}$, respectively.

Remark 10 *However, if* $m < d_1 d_2$*, null space property-type assumptions as* (18) *or* (19) *do not hold for the important case of matrix completion-type measurements (Candès and Recht, 2009), where* $\Phi(X)$ *is given as* m *sample entries*

$$\Phi(X)_\ell = X_{i_\ell, j_\ell}, \qquad \ell = 1, \dots, m, \tag{23}$$

and $(i_\ell, j_\ell) \in [d_1]\times[d_2]$ *for all* $\ell \in [m]$*, of the matrix* $X \in M_{d_1\times d_2}$*, which also were considered in the example of Section 3.*

This means that parts (i) and (iii) of Theorem 9 do, unfortunately, not apply for matrix completion measurements, which define a very relevant class of low-rank matrix recovery problems. This problem is shared by any existing theory for IRLS algorithms for low-rank matrix recovery (Fornasier et al., 2011; Mohan and Fazel, 2012). However, in Section 5, we provide strong numerical evidence that `HM-IRLS` *exhibits properties as predicted by (i) and (iii) of Theorem 9 even for the matrix completion setting. We leave the extension of the theory of* `HM-IRLS` *to matrix completion measurements as an open problem to be tackled by techniques different from uniform null space properties (Davenport and Romberg, 2016, Section V).*

4.2 Locally superlinear convergence rate for $p < 1$

Next, we state the second main theoretical result of this paper, Theorem 11. It shows that in a neighborhood of a low-rank matrix X_0 that is compatible with the measurement vector Y, the algorithm `HM-IRLS` converges to X_0 with a *convergence rate* that is *superlinear of the order* $2-p$, if the operator Φ fulfills an appropriate Schatten-p null space property.

Theorem 11 (Locally Superlinear Convergence Rate) *Assume that the linear map* $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ *fulfills the strong Schatten-p NSP of order* $2r$ *with constant* $\gamma_{2r} < 1$ *and that there exists a matrix* $X_0 \in M_{d_1\times d_2}$ *with* $\operatorname{rank}(X_0) = r \leqslant \frac{\min(d_1,d_2)}{2}$ *such that* $\Phi(X_0) = Y$*, let* Φ, Y, r *and* $0 < p \leqslant 1$ *be the input parameters of Algorithm 1. Moreover, let* $\kappa = \frac{\sigma_1(X_0)}{\sigma_r(X_0)}$ *be the condition number of* X_0 *and* $\eta^{(n)} := X^{(n)} - X_0$ *be the error matrices of the n-th output of Algorithm 1 for* $n \in \mathbb{N}$*.*
Assume that there exists an iteration $\bar{n} \in \mathbb{N}$ *and a constant* $0 < \zeta < 1$ *such that*

$$\|\eta^{(\bar{n})}\|_{S_\infty} \leqslant \zeta \sigma_r(X_0) \tag{24}$$

and $\epsilon^{(\bar{n})} = \sigma_{r+1}(X^{(\bar{n})})$*. If additionally the condition number* κ *and* ζ *are small enough, or more precisely, if*

$$\mu \|\eta^{(\bar{n})}\|_{S_\infty}^{p(1-p)} < 1 \tag{25}$$

with the constant

$$\mu := 2^{5p}(1+\gamma_{2r})^p \Big(\frac{\gamma_{2r}(3+\gamma_{2r})(1+\gamma_{2r})}{(1-\gamma_{2r})}\Big)^{2-p} \Big(\frac{d-r}{r}\Big)^{2-\frac{p}{2}} r^p \frac{\sigma_r(X_0)^{p(p-1)}}{(1-\zeta)^{2p}} \kappa^p \tag{26}$$

then

$$\|\eta^{(n+1)}\|_{S_\infty} \leqslant \mu^{1/p} \left(\|\eta^{(n)}\|_{S_\infty}\right)^{2-p} \quad \text{and} \quad \|\eta^{(n+1)}\|_{S_p} \leqslant \mu^{1/p} \left(\|\eta^{(n)}\|_{S_p}\right)^{2-p}$$

for all $n \geqslant \bar{n}$*.*

We think that the result of Theorem 11 is remarkable, since there are only few low-rank recovery algorithms which exhibit either theoretically or practically verifiable superlinear convergence rates. In particular, although the algorithms of Mishra et al. (2013) and `NewtonSLRA` of Schost and Spaenlehauer (2016) do show superlinear convergence rates, the first is not competitive to `HM-IRLS` in terms of sample complexity and the second has neither applicable theoretical guarantees for most of the interesting problems nor the ability of solving medium size problems.

Remark 12 *It is interesting to compare Theorem 11 with a related result for an IRLS algorithm for the sparse vector recovery problem in Daubechies et al. (2010, Theorem 7.9). We observe that while the statement describes the observed rates of convergence very accurately (cf. Section 5.2), the assumption* (25) *on the neighborhood that enables convergence of a rate* $2-p$ *is more pessimistic than our numerical experiments suggest. Our experiments confirm that the local convergence rate of order* $2-p$ *also holds for matrix completion measurements, where the assumption of a Schatten-p null space property fails to hold, cf. Section 5.*

4.3 Discussion and comparison with existing IRLS algorithms

Optimally, we would like to have a statement in Theorem 9 about the accumulation points $\bar{X}$ being *global minimizers* of g_ϵ^p, instead of mere stationary points (Fornasier et al., 2011, Theorem 6.11), (Daubechies et al., 2010, Theorem 5.3). A statement that strong is, unfortunately, difficult to achieve due to the non-convexity of the Schatten-p quasi-norm and of the ϵ-perturbed version g_ϵ^p. Nevertheless, our theorems can be seen as analogues of Daubechies et al. (2010, Theorem 7.7), which discusses the convergence properties of an IRLS algorithm for sparse recovery based on ℓ_p-minimization with $p < 1$.

As already mentioned in previous sections, Fornasier et al. (2011) and Mohan and Fazel (2012) proposed IRLS algorithms for low-rank matrix recovery and analysed their convergence properties. The algorithm of Fornasier et al. (2011) corresponds (almost) to `IRLS-col` with $p = 1$ as explained in Section 3. In this context, Theorem 9 recovers the results of Fornasier et al. (2011, Theorem 6.11(i-ii)) for $p = 1$ and generalize them, with weaker conclusions due to the non-convexity, to the cases $0 < p < 1$. The algorithm `IRLS-`p of Mohan and Fazel (2012) is similar to the former, but differs in the choice of the ϵ-smoothing and also covers non-convex choices $0 < p < 1$. However, we note that in the non-convex case, its convergence result (Mohan and Fazel, 2012, Theorem 5.1) corresponds to Theorem 9(ii), but does not provide statements similar to (i) and (iii) of Theorem 9.

Theorem 11 with its analysis of the convergence rate is new in the sense that to the best of our knowledge, there are no convergence rate proofs for IRLS algorithms for the low-rank matrix recovery problem in the literature. Indeed, we refer to Remark 22 in Section 6.3 for an explanation why the variants of Fornasier et al. (2011) and Mohan and Fazel (2012) cannot exhibit superlinear convergence rates, unlike `HM-IRLS`.

We also note that there is a close connection between the statements of Theorems 9 and 11 and results that were obtained for an IRLS algorithm dedicated to the sparse vector recovery problem in Daubechies et al. (2010, Theorems 7.7 and 7.9).

5. Numerical experiments

In this section, we demonstrate first that the superlinear convergence rate that was proven theoretically for Algorithm 1 (`HM-IRLS`) in Theorem 11 can indeed be accurately verified in

numerical experiments, even beyond measurement operators fulfilling the strong null space property, and compare its performance to other variants of `IRLS`.

In Section 5.3, we then examine the recovery performance of `HM-IRLS` for the matrix completion setting with the performance of other state-of-the-art algorithms comparing the measurement complexities that are needed for successful recovery for many random instances.

The numerical experiments are conducted on Linux and Mac systems with MATLAB R2017b. An implementation of `HM-IRLS` for matrix completion including code reproducing many conducted experiments is available at `https://github.com/ckuemmerle/hm_irls`.

5.1 Experimental setup

In the experiments, we sample $(d_1 \times d_2)$ dimensional ground truth matrices X_0 of rank r such that $X_0 = U\Sigma V^*$, where $U \in \mathbb{R}^{d_1 \times r}$ and $V \in \mathbb{R}^{d_2 \times r}$ are independent matrices with i.i.d. standard Gaussian entries and $\Sigma \in \mathbb{R}^{r \times r}$ is a diagonal matrix with i.i.d. standard Gaussian diagonal entries, independent from U and V.

We recall that a rank-r matrix $X \in M_{d_1 \times d_2}$ has $d_f = r(d_1 + d_2 - r)$ degrees of freedom, which is the theoretical lower bound on the number of measurements that are necessary for exact reconstruction (Candès and Plan, 2011). The random measurement setting we use in the experiments can be described as follows: We take measurements of matrix completion type, sampling $m = \lfloor \rho d_f \rfloor$ entries of X_0 uniformly over its $d_1 d_2$ indices to obtain $Y = \Phi(X_0)$. Here, ρ is such that $\frac{d_1 d_2}{d_f} \geqslant \rho \geqslant 1$ and parametrizes the difficulty of the reconstruction problem, from very hard problems for $\rho \approx 1$ to easier problems for larger ρ.

However, this uniform sampling of Φ could yield instances of measurement operators whose information content is not large enough to ensure well-posedness of the corresponding low-rank matrix recovery problem, even if $\rho > 1$. More precisely, it is impossible to recover a matrix exactly if the number of revealed entries in any row or column is smaller than its rank r, which is explained and shown in the context of the proof of Pimentel-Alarcón et al. (2016, Theorem 1).

Thus, in order to provide for a sensible measurement model for small ρ, we exclude operators Φ that sample fewer than r entries in any row or column. Therefore, we adapt the uniform sampling model such that operators Φ are discarded and sampled again until the requirement of at least r entries per column and row is met and recovery can be achieved from a theoretical point of view.

We note that the described phenomenon is very related to the fact that matrix completion recovery guarantees for the uniform sampling model require at least one additional log factor, i.e., they require at least $m \geqslant \log(\max(d_1, d_2))d_f$ sampled entries (Davenport and Romberg, 2016, Section V).

While we detail the experiments for the matrix completion measurement setting just described in the remaining section, we add that Gaussian measurement models also lead to very similar results in experiments.

5.2 Convergence rate comparison with other IRLS algorithms

In this subsection, we vary the Schatten-p parameter between 0 and 1 and compare the corresponding convergence behavior of `HM-IRLS` with the IRLS variant `IRLS-col`, which

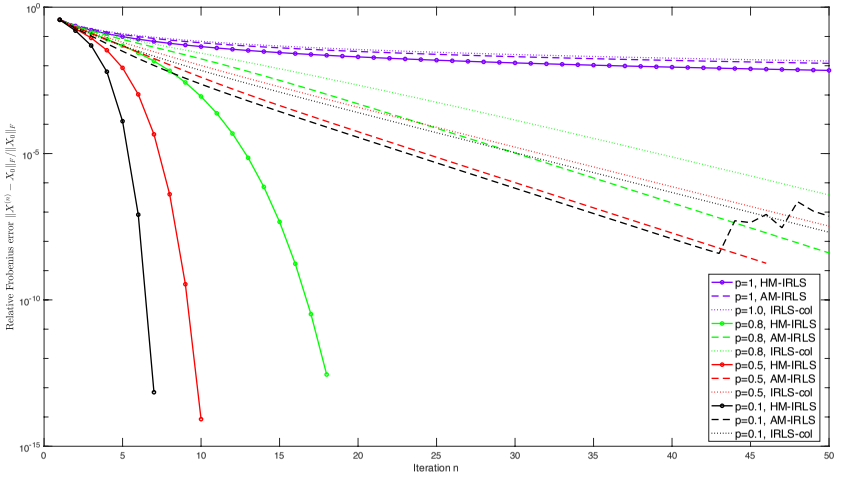


Figure 3: Relative Frobenius errors as a function of the iteration n for oversampling factor $\rho = 2$ (easy problem).

performs the reweighting just in the column space, and with the arithmetic mean variant `AM-IRLS`. The latter two coincide with Algorithm 1 except that the weight matrices are chosen as described in Equation (17) in Section 3.

We note that `IRLS-col` is very similar to the IRLS algorithms of Fornasier et al. (2011) and Mohan and Fazel (2012) and differs from them basically just in the choice of the ϵ-smoothing. We present the experiments with `IRLS-col` to isolate the influence of the weight matrix type, but very similar results can be observed for the algorithms of Fornasier et al. (2011) and Mohan and Fazel (2012).[2]

In the matrix completion setup of Section 5.1, we choose $d_1 = d_2 = 40$, $r = 10$ and distinguish easy, hard and very hard problems corresponding to oversampling factors ρ of 2.0, 1.2 and 1.0, respectively. The algorithms are provided with the ground truth rank r and are stopped whenever the relative change of Frobenius norm $\|X^{(n)} - X^{(n-1)}\|_F / \|X^{(n-1)}\|_F$ drops below the threshold of 10^{-10} or a maximal iteration of iterations $n_{\max}$ is reached.

5.2.1 Convergence rates

First, we study the behavior of the three IRLS algorithms for the easy setting of an oversampling factor of $\rho = 2$, which means that $\frac{2r(d_1+d_2-r)}{d_1 d_2} = 0.875$ of the entries are sampled, and parameters $p \in \{0.1, 0.5, 0.8, 1\}$.

In Figure 3, we observe that for $p = 1$, `HM-IRLS`, `AM-IRLS` and `IRLS-col` have a quite similar behavior, as the relative Frobenius errors $\|X^{(n)} - X_0\|_F / \|X_0\|_F$ decrease only slowly,

2. Implementations of the mentioned authors' algorithms were downloaded from `https://faculty.washington.edu/mfazel/` and `https://github.com/rward314/IRLSM`, respectively.

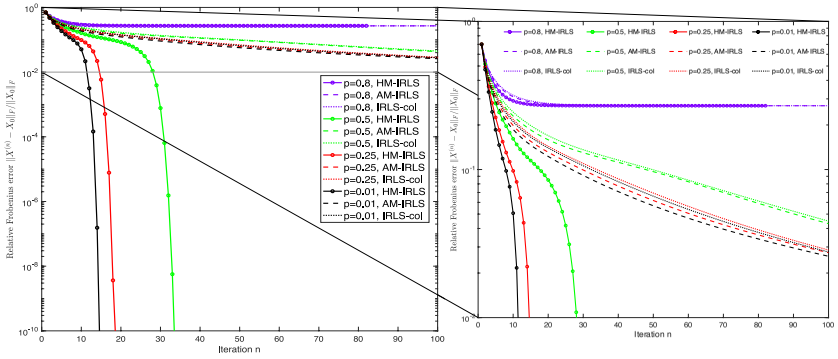


Figure 4: Relative Frobenius errors as a function of the iteration n for oversampling factor $\rho = 1.2$ (hard problem). Left column: y-range $[10^{-10}; 10^{0}]$. Right column: Enlarged section of left column corresponding to y-range of $[10^{-2}; 10^{0}]$.

i.e., even a linear rate is hardly identifiable. For choices $p < 1$ that correspond to non-convex objectives, we observe a very fast, superlinear convergence of `HM-IRLS`, as the iterates $X^{(n)}$ converge up to a relative error of less than 10^{-12} within fewer than 20 iterations for $p \in \{0.8, 0.5, 0.1\}$. Precise calculations verify that the rate of convergences are indeed of order $2 - p$, the order predicted by Theorem 11. We note that this fast convergence rate not only kicks in locally, but starting from the very first iteration.

On the other hand, it is easy to see that `AM-IRLS` and `IRLS-col` converge *linearly, but not superlinearly* to the ground truth X_0 for $p \in \{0.8, 0.5, 0.1\}$. The linear rate of `AM-IRLS` is slightly better than the one of `IRLS-col`, but the numerical stability of `AM-IRLS` deteriorates for $p = 0.1$ close to the ground truth (after iteration 43). This is due to a bad conditioning of the quadratic problems as the $X^{(n)}$ are close to rank-r matrices. In contrast, no numerical instability issues can be observed for `HM-IRLS`.

For the hard matrix completion problems with oversampling factor of $\rho = 1.2$, we observe that for $p = 0.8$, the three algorithms typically do not converge to ground truth. This can be seen in the example that is shown in Figure 4, where `HM-IRLS`, `AM-IRLS` and `IRLS-col` all exhibit a relative error of 0.27 after 100 iterations. We do not visualize the result for $p = 1$, as the iterates of the three algorithms do not converge to the ground truth either, which is to be expected: In some sense, they implement nuclear norm minimization, which is typically not able to recover a low-rank matrix from measurements with an oversampling factor as small as $\rho = 1.2$ (Donoho et al., 2013). The dramatic difference in behavior between `HM-IRLS` and the other approaches becomes very apparent for more non-convex choices of $p \in \{0.01, 0.25, 0.5\}$, where the former converges up to a relative Frobenius error of less than 10^{-10} within 15 to 35 iterations, while the others do not reach a relative error of 10^{-2} even after 100 iterations. For `HM-IRLS`, the convergence of order $2-p$ can be very well locally observed also here, it just takes some iterations until the superlinear convergence begins, which is due to the increased difficulty of the recovery problem.

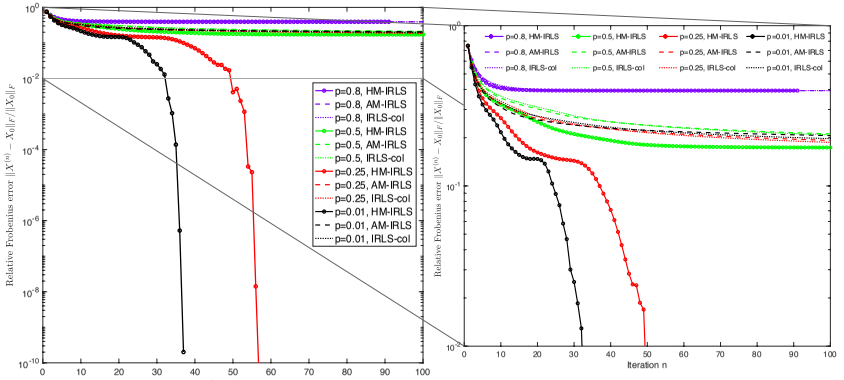


Figure 5: Relative Frobenius errors as a function of the iteration n for oversampling factor $\rho = 1.0$ (very hard problem). Left column: y-range $[10^{-10}; 10^{0}]$. Right column: Enlarged section of left column corresponding to y-range of $[10^{-2}; 10^{0}]$.

Finally, we see in the example shown in Figure 5 that even for the very hard problems where $\rho = 1$, which means that the number of sampled entries corresponds exactly to the degrees of freedom $r(d_1 + d_2 - r)$, `HM-IRLS` can be successful to recover the rank-r matrix if the parameter p is chosen small enough (here: $p \leqslant 0.25$). This is not the case for the algorithms `AM-IRLS` and `IRLS-col`.

5.2.2 `HM-IRLS` as the best extension of IRLS for sparse recovery

We summarize that among the three variants `HM-IRLS`, `AM-IRLS` and `IRLS-col`, only `HM-IRLS` is able to solve the low-rank matrix recovery problem for very low sample complexities corresponding to $\rho \approx 1$. Furthermore, it is the only IRLS algorithm for low-rank matrix recovery that exhibits a superlinear rate of convergence at all.

It is worthwhile to compare the properties of `HM-IRLS` with the behavior of the IRLS algorithm of Daubechies et al. (2010) designed to solve the sparse vector recovery problem by mimicking ℓ_p-minimization for $0 < p \leqslant 1$. While neither `IRLS-col` nor `AM-IRLS` are able to generalize the superlinear convergence behavior of Daubechies et al. (2010) (which is illustrated in Figure 8.3 of the same paper) to the low-rank matrix recovery problem, `HM-IRLS` is, as can be seen in Figures 3 to 5.

Taking the theoretical guarantees as well as the numerical evidence into account, we claim that *`HM-IRLS` is the presently best extension of IRLS for vector recovery in Daubechies et al. (2010) to the low-rank matrix recovery setting*, providing a substantial improvement over the reweighting strategies of Fornasier et al. (2011) and Mohan and Fazel (2012).

Moreover, we mention two observations which suggest that `HM-IRLS` has in some sense even more favorable properties than the algorithm of Daubechies et al. (2010): First, the discussion of Daubechies et al. (2010, Section 8) states that a superlinear convergence can only be observed locally after a considerable amount of iterations with just a linear error

decay. In contrast to that, HM-IRLS exhibits a superlinear error decay quite early (i.e., for example as early as after two iterations), at least if the sample complexity is large enough, cf. Figure 3.

Secondly, it can be observed that the convergence of the algorithm of Daubechies et al. (2010) to a sparse vector often breaks down if p is smaller than 0.5 (Daubechies et al., 2010, Section 8). In contrast to that, we observe that HM-IRLS does not suffer from this loss of global convergence for $p \ll 0.5$. Thus, a choice of very small parameters $p \approx 0.1$ or smaller is suggested as such a choice is accompanied by a very fast convergence.

5.3 Recovery performance compared to state-of-the-art algorithms

After comparing the performance of HM-IRLS with other IRLS variants, we now conduct experiments to compare the empirical performance of HM-IRLS also to that of low-rank matrix recovery algorithms different from IRLS.

To obtain a comprehensive picture, we consider not only the IRLS variants AM-IRLS and IRLS-col, but a variety of state-of-the-art methods in the experiments, as the Riemannian optimization technique Riemann_Opt (Vandereycken, 2013), the alternating minimization approaches AltMin (Haldar and Hernando, 2009), ASD (Tanner and Wei, 2016) and BFGD (Park et al., 2016), and finally the algorithms Matrix ALPS II (Kyrillidis and Cevher, 2014) and CGIHT_Matrix (Blanchard et al., 2015), which are based on iterative hard thresholding. As the IRLS variants we consider, all these algorithms use knowledge about the actual ground truth rank r.

In the experiments, we examine the empirical recovery probabilities of the different algorithms systematically for varying oversampling factors ρ, determining the difficulty of the low-rank recovery problem as the sample complexity fulfills $m = \lfloor \rho d_f \rfloor$. We recall that a large parameter ρ corresponds to an easy reconstruction problem, while a small ρ, e.g., $\rho \approx 1$, defines a very hard problem.

We choose $d_1 = d_2 = 100$ and the $r = 8$ as parameter of the experimental setting, conducting the experiments to recover rank-8 matrices $X_0 \in \mathbb{R}^{100\times 100}$. We remain in the matrix completion measurement setting described in Section 5.1, but sample now 150 random instances of X_0 and Φ for different numbers of measurements varying between $m_{\min} = 1500$ to $m_{\max} = 4000$. This means that the oversampling factor ρ increases from $\rho_{\min} = 0.975$ to $\rho_{\max} = 2.60$. For each algorithm, a successful recovery of X_0 is defined as a relative Frobenius error $\|X^{\text{out}} - X_0\|_F / \|X_0\|_F$ of the matrix X^{out} returned by the algorithm of smaller than 10^{-3}. The algorithms are run until stagnation of the iterates or until the maximal number of iterations $n_{\max} = 3000$ is reached. The number $n_{\max}$ is chosen large enough to ensure that a recovery failure is not due to a lack of iterations.

In the experiments, except for AltMin, for which we used our own implementation, we used implementations provided by the authors of the corresponding papers for the respective algorithms, using default input parameters provided by the authors. The respective code sources can be found in the references.

5.3.1 Beyond the state-of-the-art performance of HM-IRLS

The results of the experiment can be seen in Figure 6. We observe that HM-IRLS exhibits a very high empirical recovery probability for $p = 0.1$ and $p = 0.5$ as soon as the sample

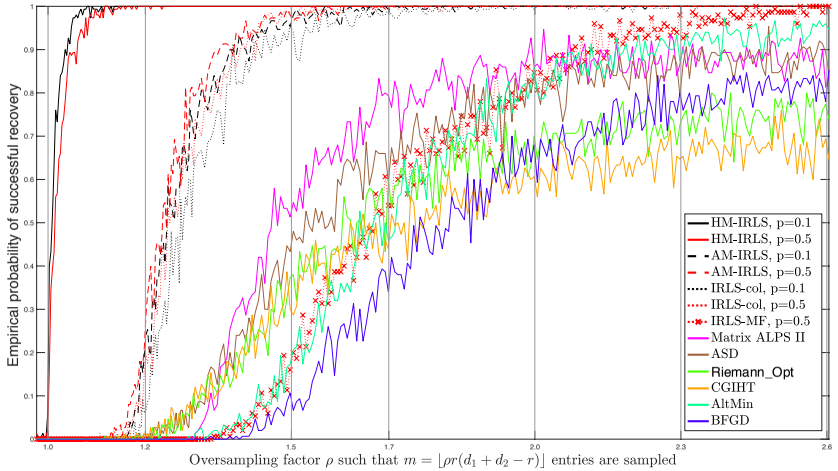


Figure 6: Comparison of empirical success rates of state-of-the-art algorithms, as a function of the oversampling factor ρ

complexity parameter ρ is slightly larger than 1.0, which means that $m = \lfloor \rho r(d_1 + d_2 - r) \rfloor$ measurements suffice to recover $(d_1 \times d_2)$-dimensional rank-r matrices with ρ close to 1. This is very close to the information theoretical lower bound of $d_f = r(d_1 + d_2 - r)$. Very interestingly, it can be observed that the empirical recovery probability reaches almost 1 already for an oversampling factor of $\rho \approx 1.1$, and remains at exactly 1 starting from $\rho \approx 1.2$.

Relatively good success rates can also be observed for the algorithms AM-IRLS and IRLS-col for non-convex parameter choices $p \in \{0.1, 0.5\}$, reaching an empirical success probability of almost 100% at around $\rho = 1.5$. AM-IRLS performs only marginally better than the classical IRLS strategy IRLS-col, which are both outperformed considerably by HM-IRLS. It is important to note that in accordance to what was observed in Section 5.2, in the successful instances, the error threshold that defines successful recovery is achieved already after a few dozen iterations for HM-IRLS, while typically only after several or many hundreds for AM-IRLS and IRLS-col. Furthermore, it is interesting to observe that the algorithm IRLS-MF, which corresponds to the variant studied and implemented by Mohan and Fazel (2012) and differs from IRLS-col mainly only in the choice of the ϵ-smoothing (14), has a considerably worse performance than the other IRLS methods. This is plausible since the smoothing influences severely the optimization landscape of the objective to be minimized.

The strong performance of HM-IRLS is in stark contrast to the behavior of all the algorithms that are based on different approaches than IRLS and that we considered in our experiments. They basically never recover any rank-r matrix if $\rho < 1.2$, and most of the algorithms need a sample complexity parameter of $\rho > 1.7$ to exceed a empirical recovery probability of a mere 0.5. A success rate of close to 0.8 is reached not before raising ρ above

2.0 in our experimental setting, and also only for a subset of the comparison algorithms, in particular for `Matrix ALPS II, ASD, AltMin`. The empirical probability of 1 is only reached for some of the IRLS methods, and not for any competing method in our experimental setting, even for rather large oversampling factors such as $\rho = 2.5$. While we do not rule out that a possible parameter tuning could improve the performance of any of the algorithms slightly, we conclude that for hard matrix completion problems, the experimental evidence for the vast differences in the recovery performance of `HM-IRLS` compared to other methods is very apparent.

Thus, our observation is that the proposed `HM-IRLS` algorithm *recovers low-rank matrices systematically with nearly the optimal number of measurements and needs fewer measurements than all the state-of-the-art algorithms we included in our experiments*, if the non-convexity parameter p is chosen such that $p \ll 1$.

We also note that the very sharp phase transition between failure and success that can be observed in Figure 6 for `HM-IRLS` indicates that the sample complexity parameter ρ is indeed the major variable determining the success of `HM-IRLS`. In contrast, the wider phase transitions for the other algorithms suggest that they might depend more on other factors, as the realizations of the random sampling model and the interplay of measurement operator Φ and ground truth matrix X_0.

Another conclusion that can be drawn from the empirical recovery probability of 1 is that, despite the severe non-convexity of the underlying Schatten-p quasi-norm for, e.g., $p = 0.1$, `HM-IRLS` with the initialization of $X^{(1)}$ as the Frobenius norm minimizer does not get stuck in stationary points if the oversampling factor is large enough. Further experiments conducted with random initializations as well as severely adversary initializations, e.g., with starting points chosen in the orthogonal complement of the spaces spanned by the singular vectors of the ground truth matrix X_0, lead to comparable results. Therefore, we claim that `HM-IRLS` exhibits a global convergence behavior in interesting application cases and for oversampling factor ranges for which competing non-convex low-rank matrix recovery algorithms fail to succeed. We consider a theoretical investigation of such behavior as an interesting open problem to explore.

5.4 Computational complexity

While the harmonic mean weight matrix $\widetilde{W}^{(n)}$, cf. (15), is an inverse of a $(d_1d_2 \times d_1d_2)$-matrix and therefore in general a dense $(d_1d_2 \times d_1d_2)$-matrix, it is important to note that it never has to be computed explicitly in an implementation of `HM-IRLS`; neither is it necessary to compute its inverse $(\widetilde{W}^{(n)})^{-1} = \frac{1}{2}\left(U^{(n)}(\bar{\Sigma}^{(n)})^{2-p}U^{(n)*} \oplus V^{(n)}(\bar{\Sigma}^{(n)})^{2-p}V^{(n)*}\right)$ explicitly.

Indeed, as it can be seen in (13) and by the definition of the Kronecker sum (55), the harmonic mean weight matrix appears just as the linear operator $(\mathcal{W}^{(n)})^{-1}$ on the space of matrices $M_{d_1\times d_2}$, whose action consists of a left- and right-sided matrix multiplication, cf. (12). Therefore, the application of $(\mathcal{W}^{(n)})^{-1}$ is $O(d_1d_2(d_1+d_2))$ by the naive matrix multiplication algorithm, and can be easily parallelized.

While this useful observation is helpful for the implementation of `HM-IRLS`, it is not true for `AM-IRLS`, as the action of $(W^{(n)}_{(\text{arith})})^{-1}$, the inverse of the arithmetic mean weight matrix at iteration n, is not representable as a sum of left- and right-sided matrix multiplication.

This means that even the execution of a fixed number of iterations of `HM-IRLS` is faster than computational advantage over `AM-IRLS`.

The cost to compute $\Phi \circ \widetilde{\mathcal{W}}^{(n)-1} \circ \Phi^* \in M_{m\times m}$ depends on the linear measurement operator Φ. In the matrix completion setting (23), no additional arithmetic operations have to be performed, as Φ is a just a selection operator in this case, and for `HM-IRLS`, this means that $\Phi \circ \widetilde{\mathcal{W}}^{(n)-1} \circ \Phi^*$ is a sparse matrix.

Thus, the algorithm `HM-IRLS` consists of basically of two computational steps per iteration: The computation of the SVD of the $d_1 \times d_2$-matrix $X^{(n)}$ and the solution of the linearly constrained least squares problem in (13). The first is of time complexity $O(d_1d_2\min(d_1,d_2))$. The time complexity of the second depends on Φ, but is dominated by the inversion of a symmetric, $m \times m$ sparse linear system in the matrix completion setting, if m is the number of given entries. This has a worst case time complexity of $O(\max(d_1,d_2)^3r^3)$ if ρ is just a constant oversampling factor.

For the matrix completion case, this allows us to recover low-rank matrices up to, e.g., $d_1 = d_2 = 3000$ on a single machine given very few entries with `HM-IRLS`.

Acceleration possibilities and extensions

To tackle higher dimensionalities in reasonable runtimes, a key strategy could be to address the computational bottleneck of `HM-IRLS`, the solution of the $m \times m$ linear system in (13), by using iterative methods. For IRLS algorithms designed for the related sparse recovery problem, the usage of conjugate gradient (CG) methods is discussed in Fornasier et al. (2016). By coupling the accuracy of the CG solutions to the outer IRLS iteration and using appropriate preconditioning, the authors obtain a competitive solver for the sparse recovery problem, also providing a convergence analysis. Similar ideas could be used for an acceleration of `HM-IRLS`.

It is interesting to see if further computational improvements can be achieved by combining the ideas of `HM-IRLS` with the usage of truncated and randomized SVDs (Halko et al., 2011), replacing the full SVDs of the $X^{(n)}$ that are needed to define the linear operator $(\mathcal{W}^{(n)})^{-1}$ in Algorithm 1.

6. Theoretical analysis

For the theoretical analysis of `HM-IRLS`, we introduce the following auxiliary functional $\mathcal{J}_p$, leading to a variational interpretation of the algorithm. In the whole section, we denote $d = \min(d_1,d_2)$ and $D = \max(d_1,d_2)$.

Definition 13 *Let $0 < p \leqslant 1$. Given a full rank matrix $Z \in M_{d_1\times d_2}$, let*

$$\widetilde{W}(Z) := 2\left[\mathbf{I}_{d_2} \otimes (ZZ^*)^{\frac{1}{2}}\right]\left[(ZZ^*)^{\frac{1}{2}} \oplus (Z^*Z)^{\frac{1}{2}}\right]^{-1}\left[(Z^*Z)^{\frac{1}{2}} \otimes \mathbf{I}_{d_1}\right] \in H_{d_1d_2\times d_1d_2}$$

be the harmonic mean matrix $\widetilde{W}$ associated to Z.

We define the auxiliary functional $\mathcal{J}_p : M_{d_1\times d_2} \times \mathbb{R}_{\geqslant 0} \times M_{d_1\times d_2} \to \mathbb{R}_{\geqslant 0}$ *as*

$$\mathcal{J}_p(X,\epsilon,Z) := \begin{cases} \frac{p}{2}\|X_{\text{vec}}\|^2_{\ell_2(\widetilde{W}(Z))} + \frac{\epsilon^2 p}{2}\sum_{i=1}^{d}\sigma_i(Z) + \frac{2-p}{2}\sum_{i=1}^{d}\sigma_i(Z)^{\frac{p}{(p-2)}} & \text{if } \operatorname{rank}(Z) = d, \\ +\infty & \text{if } \operatorname{rank}(Z) < d. \end{cases}$$

We note that the matrix $\widetilde{W}$ of Definition 13 is just the harmonic mean of the matrices $\widetilde{W}_1 := \mathbf{I}_{d_2} \otimes (ZZ^*)^{\frac{1}{2}}$ and $\widetilde{W}_2 = (Z^*Z)^{\frac{1}{2}} \otimes \mathbf{I}_{d_1}$, as introduced in Section 2.3, if $(ZZ^*)^{\frac{1}{2}}$ *and* $(Z^*Z)^{\frac{1}{2}}$ are positive definite. Indeed, in this case, $(ZZ^*)^{\frac{1}{2}} \oplus (Z^*Z)^{\frac{1}{2}} = \widetilde{W}_1 + \widetilde{W}_2$ is invertible and as $(A^{-1} + B^{-1})^{-1} = A(A+B)^{-1}B$ for any positive definite matrices A and B of the same dimensions,

$$\widetilde{W}(Z) = 2\widetilde{W}_1\big(\widetilde{W}_1 + \widetilde{W}_2\big)^{-1}\widetilde{W}_2 = 2(\widetilde{W}_1^{-1} + \widetilde{W}_2^{-1})^{-1}. \tag{27}$$

We use the more general definition $\widetilde{W}(Z)$ as it is well-defined for any full-rank $Z \in M_{d_1 \times d_2}$ and as it allows to handle the case of non-square matrices, i.e., the case $d_1 \neq d_2$, as in this case $(ZZ^*)^{\frac{1}{2}}$ or $(Z^*Z)^{\frac{1}{2}}$ has to be singular. Using the Moore-Penrose pseudo inverse $\widetilde{W}_1^+$ and $\widetilde{W}_2^+$ of the matrices $\widetilde{W}_1$ and $\widetilde{W}_2$, we can rewrite $\widetilde{W}(Z)$ from Definition 13 as

$$\widetilde{W}(Z) = 2\widetilde{W}_1\big(\widetilde{W}_1 + \widetilde{W}_2\big)^{-1}\widetilde{W}_2 = 2(\widetilde{W}_1^{+} + \widetilde{W}_2^{+})^{-1}.$$

With the auxiliary functional $\mathcal{J}_p$ at hand, we can interpret Algorithm 1 as an alternating minimization of the functional $\mathcal{J}_p(X, \epsilon, Z)$ with respect to its arguments X, ϵ and Z.

In the following, we derive the formula (15) for the weight matrix $\widetilde{W}^{(n+1)}$ as the evaluation $\widetilde{W}^{(n+1)} = \widetilde{W}\big(Z^{(n+1)}\big)$ of $\widetilde{W}$ from Definition 13 at the minimizer

$$Z^{(n+1)} = \operatorname*{arg\,min}_{Z \in M_{d_1 \times d_2}} \mathcal{J}_p(X^{(n+1)}, \epsilon^{(n+1)}, Z), \tag{28}$$

with the minimizer being unique. Similarly, formula (13) can be interpreted as

$$X^{(n+1)} = \operatorname*{arg\,min}_{\substack{X \in M_{d_1 \times d_2} \\ \Phi(X) = Y}} \|X_{\mathrm{vec}}\|^2_{\ell_2(\widetilde{W}(Z^{(n)}))} = \operatorname*{arg\,min}_{\substack{X \in M_{d_1 \times d_2} \\ \Phi(X) = Y}} \mathcal{J}_p(X, \epsilon^{(n)}, Z^{(n)}). \tag{29}$$

These observations constitute the starting point of the convergence analysis of Algorithm 1, which is detailed subsequently after the verification of the optimization steps.

6.1 Optimization of $\mathcal{J}_p$ with respect to Z and X

We fix $X \in M_{d_1 \times d_2}$ with singular value decomposition $X = \sum_{i=1}^{d} \sigma_i u_i v_i^*$, where $u_i \in \mathbb{C}^{d_1}$ $v_i \in \mathbb{C}^{d_2}$ are the left and right singular vectors respectively and $\sigma_i = \sigma_i(X)$ denote its singular values for $i \in [d]$.

Our objective in the following is the justification of formula (15). To yield the building blocks of the weight matrix $\widetilde{W}^{(n+1)}$, we consider the minimization problem

$$\operatorname*{arg\,min}_{Z \in M_{d_1 \times d_2}} \mathcal{J}_p(X, \epsilon, Z) \tag{30}$$

for $\epsilon > 0$.

Lemma 14 *The unique minimizer of* (30) *is given by*

$$Z_{\mathrm{opt}} = \sum_{i=1}^{d} (\sigma_i(X)^2 + \epsilon^2)^{\frac{p-2}{2}} u_i v_i^*.$$

Furthermore, the value of $\mathcal{J}_p$ at the minimizer Z_{opt} is

$$\mathcal{J}_p(X, \epsilon, Z_{\mathrm{opt}}) = \sum_{i=1}^{d} (\sigma_i(X)^2 + \epsilon^2)^{\frac{p}{2}} =: g_\epsilon^p(X) \tag{31}$$

for $p > 0$.

The proof of Lemma 14 is detailed in Appendix B.

Remark 15 *We note that the value of $\mathcal{J}_p(X, \epsilon, Z_{\mathrm{opt}})$ can be interpreted as a smooth ϵ-perturbation of a p-th power of a Schatten-p quasi-norm of the matrix X. In fact, for $\epsilon = 0$ we have*

$$\mathcal{J}_p(X, 0, Z_{\mathrm{opt}}) = \|X\|_{S_p}^p = g_0^p(X).$$

Now, we show that our definition rule (13) of $X^{(n+1)}$ in Algorithm 1 can be interpreted as a minimization of the auxiliary functional $\mathcal{J}_p$ with respect to the variable X. Additionally, this minimization step can be formulated as the solution of a weighted least squares problem with weight matrix $\widetilde{W}^{(n)}$. This is summarized in the following lemma.

Lemma 16 *Let $0 < p \leqslant 1$. Given a full-rank matrix $Z \in M_{d_1 \times d_2}$, let $\widetilde{W}(Z) := 2([[(ZZ^*)^{\frac{1}{2}}]^+ \oplus [(Z^*Z)^{\frac{1}{2}}]^+)^{-1} \in H_{d_1 d_2 \times d_1 d_2}$ be the matrix from Definition 13 and $\mathcal{W}^{-1} : M_{d_1 \times d_2} \to M_{d_1 \times d_2}$ the linear operator of its inverse*

$$\mathcal{W}^{-1}(X) := \frac{1}{2}\left[[(ZZ^*)^{\frac{1}{2}}]^+ X + X[(Z^*Z)^{\frac{1}{2}}]^+\right].$$

Then the matrix

$$X_{\mathrm{opt}} = \big(\mathcal{W}^{-1} \circ \Phi^* \circ (\Phi \circ \mathcal{W}^{-1} \circ \Phi^*)^{-1}\big)\big(Y\big) \in M_{d_1 \times d_2}$$

is the unique minimizer of the optimization problems

$$\operatorname*{arg\,min}_{\Phi(X) = Y} \mathcal{J}_p(X, \epsilon, Z) = \operatorname*{arg\,min}_{\Phi(X) = Y} \|X_{\mathrm{vec}}\|^2_{\ell_2(\widetilde{W})}. \tag{32}$$

Moreover, a matrix $X_{\mathrm{opt}} \in M_{d_1 \times d_2}$ is a minimizer of the minimization problem (32) *if and only if it fulfills the property*

$$\langle \widetilde{W}(Z)(X_{\mathrm{opt}})_{\mathrm{vec}}, H_{\mathrm{vec}} \rangle_{\ell_2} = 0 \quad \text{for all} \quad H \in \mathcal{N}(\Phi) \quad \text{and} \quad \Phi(X_{\mathrm{opt}}) = Y. \tag{33}$$

In Appendix B, the interested reader can find a sketch of the proof of this lemma.

6.2 Basic properties of the algorithm and convergence results

In the following subsection, we will have a closer look at Algorithm 1 and point out some of its properties, in particular, the boundedness of the iterates $(X^{(n)})_{n \in \mathbb{N}}$ and the fact that two consecutive iterates are getting arbitrarily close as $n \to \infty$. These results will be used to show convergence and to determine the rate of convergence of Algorithm 1 under conditions determined along the way.

Lemma 17 *Let $(X^{(n)}, \epsilon^{(n)})_{n\in\mathbb{N}}$ be the sequence of iterates and smoothing parameters of Algorithm 1. Let $X^{(n)} = \sum_{i=1}^{d} \sigma_i^{(n)} u_i^{(n)} v_i^{(n)*}$ be the SVD of the n-th iterate $X^{(n)}$. Let $(Z^{(n)})_{n\in\mathbb{N}}$ be a corresponding sequence such that*

$$Z^{(n)} = \sum_{i=1}^{d} (\sigma_i^{(n)2} + \epsilon^{(n)2})^{\frac{p-2}{2}} u_i^{(n)} v_i^{(n)*}$$

for $n \in \mathbb{N}$. Then the following properties hold:

(a) $\mathcal{J}_p(X^{(n)}, \epsilon^{(n)}, Z^{(n)}) \geqslant \mathcal{J}_p(X^{(n+1)}, \epsilon^{(n+1)}, Z^{(n+1)})$ *for all* $n \geqslant 1$,

(b) $\|X^{(n)}\|_{S_p}^p \leqslant \mathcal{J}_p(X^{(1)}, \epsilon^{(0)}, Z^{(0)}) =: \mathcal{J}_{p,0}$ *for all* $n \geqslant 1$,

(c) The iterates $X^{(n)}, X^{(n+1)}$ come arbitrarily close as $n \to \infty$, i.e., $\lim_{n\to\infty} \|(X^{(n)} - X^{(n+1)})_{\mathrm{vec}}\|_{\ell_2}^2 = 0$.

At this point we notice that, assuming $X^{(n)} \to \bar{X}$ and $\epsilon^{(n)} \to \bar{\epsilon}$ for $n \to \infty$ with the limit point $(\bar{X}, \bar{\epsilon}) \in M_{d_1\times d_2} \times \mathbb{R}_{\geqslant 0}$, it would follow that

$$\mathcal{J}_p(X^{(n)}, \epsilon^{(n)}, Z^{(n)}) \to g_{\bar{\epsilon}}^p(\bar{X})$$

for $n \to \infty$ by equation (31).

Now, let $\epsilon > 0$, a measurement vector $Y \in \mathbb{C}^m$ and the linear operator Φ be given and consider the optimization problem

$$\min_{\substack{X\in M_{d_1\times d_2} \\ \Phi(X)=Y}} g_\epsilon^p(X) \tag{34}$$

with $g_\epsilon^p(X) = \sum_{i=1}^{d} (\sigma_i(X)^2 + \epsilon^2)^{\frac{p}{2}}$ and $\sigma_i(X)$ being the i-th singular value of X, cf. (31). If $g_\epsilon^p(X)$ is non-convex, which is the case for $p < 1$, one might practically only be able to find critical points of the problem.

Lemma 18 *Let $X \in M_{d_1\times d_2}$ be a matrix with the SVD such that $X = \sum_{i=1}^{d} \sigma_i u_i v_i^*$, let $\epsilon > 0$. If we define*

$$\widetilde{W}(X, \epsilon) = 2\left[\Big(\sum_{i=1}^{d} (\sigma_i^2 + \epsilon^2)^{\frac{2-p}{2}} u_i u_i^* \Big) \oplus \Big(\sum_{i=1}^{d} (\sigma_i^2 + \epsilon^2)^{\frac{2-p}{2}} v_i v_i^* \Big) \right]^{-1} \in H_{d_1 d_2 \times d_1 d_2},$$

then $\widetilde{W}(X^{(n)}, \epsilon^{(n)}) = \widetilde{W}^{(n)}$, with $\widetilde{W}^{(n)}$ defined as in Algorithm 1, cf. (10). Furthermore, X is a critical point of the optimization problem (34) if and only if

$$\langle \widetilde{W}(X, \epsilon) X_{\mathrm{vec}}, H_{\mathrm{vec}} \rangle_{\ell_2} = 0 \quad \text{for all} \quad H \in \mathcal{N}(\Phi) \quad \text{and} \quad \Phi(X) = Y. \tag{35}$$

In the case that g_ϵ^p is convex, i.e., if $p = 1$, (35) implies that X is the unique minimizer of (34).

Now, we have some basic properties of the algorithm at hand that allow us, together with the strong nullspace property in Definition 4 to carry out the proof of the convergence result in Theorem 9. The proof is sketched in Appendix C using the results above.

6.3 Locally superlinear convergence

In the proof of Theorem 11 we use the following bound on perturbations of the singular value decomposition, which is originally due to Wedin (1972). It bounds the alignment of the subspaces spanned by the singular vectors of two matrices by their norm distance, given a gap between the first singular values of the one matrix and the last singular values of the other matrix that is sufficiently pronounced.

Lemma 19 (Wedin's bound (Stewart, 2006)) *Let X and $\bar{X}$ be two matrices of the same size and their singular value decompositions*

$$X = \begin{pmatrix} U_1 & U_2 \end{pmatrix} \begin{pmatrix} \Sigma_1 & 0 \\ 0 & \Sigma_2 \end{pmatrix} \begin{pmatrix} V_1^* \\ V_2^* \end{pmatrix} \quad \text{and} \quad \bar{D} = \begin{pmatrix} \bar{U}_1 & \bar{U}_2 \end{pmatrix} \begin{pmatrix} \bar{\Sigma}_1 & 0 \\ 0 & \bar{\Sigma}_2 \end{pmatrix} \begin{pmatrix} \bar{V}_1^* \\ \bar{V}_2^* \end{pmatrix},$$

where the submatrices have the sizes of corresponding dimensions. Suppose that δ, α satisfying $0 < \delta \leqslant \alpha$ are such that $\alpha \leqslant \sigma_{\min}(\Sigma_1)$ and $\sigma_{\max}(\bar{\Sigma}_2) < \alpha - \delta$. Then

$$\|\bar{U}_2^* U_1\|_{S_\infty} \leqslant \sqrt{2} \frac{\|X - \bar{X}\|_{S_\infty}}{\delta} \text{ and } \|\bar{V}_2^* V_1\|_{S_\infty} \leqslant \sqrt{2} \frac{\|X - \bar{X}\|_{S_\infty}}{\delta}. \tag{36}$$

As a first step towards the proof of Theorem 11, we show the following lemma.

Lemma 20 *Let $(X^{(n)})_n$ be the output sequence of Algorithm 1 for parameters Φ, Y, r and $0 < p \leqslant 1$, and $X_0 \in M_{d_1\times d_2}$ be a matrix such that $\Phi(X_0) = Y$.*

(i) Let $\eta_{2r}^{(n+1)}$ be the best rank-$2r$ approximation of $\eta^{(n+1)} = X^{(n+1)} - X_0$. Then

$$\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^{2p} \leqslant 2^{2-p} \Bigg(\sum_{i=r+1}^{d} \big(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\big)^{\frac{p}{2}} \Bigg)^{2-p} \|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^{2p},$$

where $\widetilde{W}^{(n)}$ denotes the harmonic mean weight matrix from (10).

(ii) Assume that the linear map $\Phi : M_{d_1\times d_2} \to \mathbb{C}^m$ fulfills the strong Schatten-p NSP of order $2r$ with constant $\gamma_{2r} < 1$. Then

$$\|\eta^{(n+1)}\|_{S_2}^{2p} \leqslant 2^p \frac{\gamma_{2r}^{2-p}}{r^{2-p}} \Bigg(\sum_{i=r+1}^{d} \big(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\big)^{\frac{p}{2}} \Bigg)^{2-p} \|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^{2p}. \tag{37}$$

(iii) Under the same assumption as for (ii), it holds that

$$\|\eta^{(n+1)}\|_{S_p}^{2p} \leqslant (1 + \gamma_{2r})^2 2^{2-p} \Bigg(\sum_{i=r+1}^{d} \big(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\big)^{\frac{p}{2}} \Bigg)^{2-p} \|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^{2p}.$$

Proof (i) Let the $X^{(n)} = \widetilde{U}^{(n)} \Sigma^{(n)} \widetilde{V}^{(n)*}$ be the (full) singular value decomposition of $X^{(n)}$, i.e., $\widetilde{U}^{(n)} \in \mathcal{U}_{d_1}$ and $\widetilde{V}^{(n)} \in \mathcal{U}_{d_2}$ are unitary matrices and $\Sigma^{(n)} = \mathrm{diag}(\sigma_1(X^{(n)}), \ldots, \sigma_r(X^{(n)})) \in M_{d_1\times d_2}$. We define $U_T^{(n)} \in M_{d_1\times r}$ as the matrix of the first r columns of $\widetilde{U}^{(n)}$ and $U_{T_c}^{(n)} \in$

$M_{d_1\times(d_1-r)}$ as the matrix of its last d_1-r columns, so that $\widetilde{U}^{(n)} = \begin{pmatrix} U_T^{(n)} & U_{T_c}^{(n)} \end{pmatrix}$, and similarly $V_T^{(n)}$ and $V_{T_c}^{(n)}$.

As $\mathbf{I}_{d_1} = U_T^{(n)}U_T^{(n)*} + U_{T_c}^{(n)}U_{T_c}^{(n)*}$ and $\mathbf{I}_{d_2} = V_T^{(n)}V_T^{(n)*} + V_{T_c}^{(n)}V_{T_c}^{(n)*}$, we note that

$$U_{T_c}^{(n)}U_{T_c}^{(n)*}\eta^{(n+1)}V_{T_c}^{(n)}V_{T_c}^{(n)*} = \eta^{(n+1)} - U_T^{(n)}U_T^{(n)*}\eta^{(n+1)} + U_{T_c}^{(n)}U_{T_c}^{(n)*}\eta^{(n+1)}V_T^{(n)}V_T^{(n)*},$$

while $U_T^{(n)}U_T^{(n)*}\eta^{(n+1)} + U_{T_c}^{(n)}U_{T_c}^{(n)*}\eta^{(n+1)}V_T^{(n)}V_T^{(n)*}$ has a rank of at most $2r$. This implies that

$$\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p} \leqslant \|U_{T_c}^{(n)}U_{T_c}^{(n)*}\eta^{(n+1)}V_{T_c}^{(n)}V_{T_c}^{(n)*}\|_{S_p} = \|U_{T_c}^{(n)*}\eta^{(n+1)}V_{T_c}^{(n)}\|_{S_p}. \tag{38}$$

Using the definitions of $\widetilde{U}^{(n)}$ and $\widetilde{V}^{(n)}$, we write the harmonic mean weight matrices of the n-th iteration (10) as

$$\widetilde{W}^{(n)} = 2(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})\big(\bar{\Sigma}_{d_1}^{(n)2-p}\oplus\bar{\Sigma}_{d_2}^{(n)2-p}\big)^{-1}(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})^*, \tag{39}$$

where $\bar{\Sigma}_{d_1}^{(n)} \in M_{d_1\times d_1}$ and $\bar{\Sigma}_{d_2}^{(n)} \in M_{d_2\times d_2}$ are the diagonal matrices with the smoothed singular values of $X^{(n)}$ from (11), but filled up with zeros if necessary. Using the abbreviation

$$\Omega := (\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})^*\widetilde{W}^{(n)\frac{1}{2}}\eta_{\mathrm{vec}}^{(n+1)} \in \mathbb{C}^{d_1d_2}, \tag{40}$$

we rewrite

$$\begin{aligned}\eta_{\mathrm{vec}}^{(n+1)} &= \widetilde{W}^{(n)-\frac{1}{2}}\widetilde{W}^{(n)\frac{1}{2}}\eta_{\mathrm{vec}}^{(n+1)} = 2^{-1/2}(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})\big(\bar{\Sigma}_{d_1}^{(n)2-p}\oplus\bar{\Sigma}_{d_2}^{(n)2-p}\big)^{1/2}\Omega\\ &= 2^{-1/2}(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})\left[(\mathbf{I}_{d_2}\otimes\bar{\Sigma}_{d_1}^{(n)\frac{2-p}{2}})D_L + (\bar{\Sigma}_{d_2}^{(n)\frac{2-p}{2}}\otimes\mathbf{I}_{d_1})D_R\right]\Omega\end{aligned} \tag{41}$$

with the diagonal matrices $D_L, D_R \in M_{d_1d_2\times d_1d_2}$ such that

$$(D_L)_{i+(j-1)d_1,i+(j-1)d_1} = \Big(1+\Big(\frac{\sigma_j^2(X^{(n)})+\epsilon^{(n)2}}{\sigma_i^2(X^{(n)})+\epsilon^{(n)2}}\Big)^{\frac{2-p}{2}}\Big)^{-1/2}$$

and

$$(D_R)_{i+(j-1)d_1,i+(j-1)d_1} = \Big(\Big(\frac{\sigma_i^2(X^{(n)})+\epsilon^{(n)2}}{\sigma_j^2(X^{(n)})+\epsilon^{(n)2}}\Big)^{\frac{2-p}{2}}+1\Big)^{-1/2}$$

for $i\in[d_1]$ and $j\in[d_2]$. This can be seen from the definitions of the Kronecker product $\otimes$ and the Kronecker sum $\oplus$ (cf. Appendix A), as

$$\begin{aligned}&\Big(\big(\bar{\Sigma}_{d_1}^{(n)2-p}\oplus\bar{\Sigma}_{d_2}^{(n)2-p}\big)^{1/2}\Big)_{i+(j-1)d_1,i+(j-1)d_1} = (s_i+s_j)^{1/2}\\ &= s_i(s_i+s_j)^{-1/2} + s_j(s_i+s_j)^{-1/2} = s_i^{1/2}(1+\frac{s_j}{s_i})^{-1/2} + s_j^{1/2}(\frac{s_i}{s_j}+1)^{-1/2}\end{aligned}$$

if s_ℓ denotes the ℓ-th diagonal entry of $\bar{\Sigma}_{d_2}^{(n)2-p}$ and $\bar{\Sigma}_{d_1}^{(n)2-p}$ for $\ell\in[\max(d_1,d_2)]$.

If we write $\bar{\Sigma}_{d_1,T_c}^{(n)\frac{2-p}{2}} \in M_{(d_1-r)\times(d_1-r)}$ for the diagonal matrix containing the d_1-r last diagonal elements of $\bar{\Sigma}_{d_1}^{(n)2-p}$ and $\bar{\Sigma}_{d_2,T_c}^{(n)\frac{2-p}{2}} \in M_{(d_1-r)\times(d_1-r)}$ for the diagonal matrix containing the d_2-r last diagonal elements of $\bar{\Sigma}_{d_2}^{(n)2-p}$, it follows from (41) that

$$\begin{aligned}\big\|U_{T_c}^{(n)*}\eta^{(n+1)}V_{T_c}^{(n)}\big\|_{S_p}^p &= 2^{-\frac{p}{2}}\Big\|U_{T_c}^{(n)*}\widetilde{U}^{(n)}\Big[\bar{\Sigma}_{d_1}^{(n)\frac{2-p}{2}}(D_L\Omega)_{\mathrm{mat}} + (D_R\Omega)_{\mathrm{mat}}\bar{\Sigma}_{d_2}^{(n)\frac{2-p}{2}}\Big]\widetilde{V}^{(n)*}V_{T_c}^{(n)}\Big\|_{S_p}^p\\ &\leqslant 2^{-\frac{p}{2}}\Big\|\bar{\Sigma}_{d_1,T_c}^{(n)\frac{2-p}{2}}\big[(D_L\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\Big\|_{S_p}^p + \Big\|\big[(D_R\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\bar{\Sigma}_{d_2,T_c}^{(n)\frac{2-p}{2}}\Big\|_{S_p}^p\end{aligned}$$

with the notation that M_{T_c,T_c} denotes the submatrix of M which contains the intersection of the last d_1-r rows of M with its last d_2-r columns.

Now, Hölder's inequality for Schatten-p quasi-norms (e.g., Gohberg et al. (2000, Theorem 11.2)) can be used to see that

$$\Big\|\bar{\Sigma}_{d_1,T_c}^{(n)\frac{2-p}{2}}\big[(D_L\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\Big\|_{S_p}^p \leqslant \Big\|\bar{\Sigma}_{T_c}^{(n)\frac{2-p}{2}}\Big\|_{S_{\frac{2p}{2-p}}}^p\Big\|\big[(D_L\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\Big\|_{S_2}^p. \tag{42}$$

Inserting the definition

$$\Big\|\bar{\Sigma}_{T_c}^{(n)\frac{2-p}{2}}\Big\|_{S_{\frac{2p}{2-p}}}^p = \Big(\sum_{i=r+1}^d\big(\sigma_i^2(X^{(n)})+\epsilon^{(n)2}\big)^{\frac{2p(2-p)}{(2-p)4}}\Big)^{\frac{2-p}{2}} = \Big(\sum_{i=r+1}^d\big(\sigma_i^2(X^{(n)})+\epsilon^{(n)2}\big)^{\frac{p}{2}}\Big)^{\frac{2-p}{2}}$$

allows us to rewrite the first factor, while the second factor can be bounded by

$$\begin{aligned}\big\|\big[(D_L\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\big\|_{S_2}^p &\leqslant \big\|(D_L\Omega)_{\mathrm{mat}}\big\|_{S_2}^p \leqslant \|\Omega_{\mathrm{mat}}\|_{S_2}^p = \|(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})^*\widetilde{W}^{(n)\frac{1}{2}}\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2}^p\\ &= \|\widetilde{W}^{(n)\frac{1}{2}}\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2}^p = \|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^p,\end{aligned}$$

as the matrix $D_L\in M_{d_1d_2\times d_1d_2}$ from (41) fulfills $\|D_L\|_{S_\infty}\leqslant 1$ since its entries are bounded by 1; we also recall the definition (40) of Ω and that $\widetilde{V}^{(n)}$ and $\widetilde{U}^{(n)}$ are unitary.

The term $\Big\|\big[(D_R\Omega)_{\mathrm{mat}}\big]_{T_c,T_c}\bar{\Sigma}_{d_2,T_c}^{(n)\frac{2-p}{2}}\Big\|_{S_p}^p$ in the bound of $\big\|U_{T_c}^{(n)*}\eta^{(n+1)}V_{T_c}^{(n)}\big\|_{S_p}^p$ can be estimated analogously. Combining this with (38), we obtain

$$\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^{2p} \leqslant 2^{-p}\Big(2\Big(\sum_{i=r+1}^d\big(\sigma_i^2(X^{(n)})+\epsilon^{(n)2}\big)^{\frac{p}{2}}\Big)^{\frac{2-p}{2}}\Big)^2\|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^{2p},$$

concluding the proof of statement (i).

(ii) Using the strong Schatten-p null space property (18) of order $2r$ and that $\eta^{(n+1)}\in\mathcal{N}(\Phi)$, we estimate

$$\begin{aligned}\|\eta^{(n+1)}\|_{S_2}^{2p} &= \big(\|\eta_{2r}^{(n+1)}\|_{S_2}^2 + \|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_2}^2\big)^p \leqslant \Big(\frac{\gamma_{2r}^{2/p}+\gamma_{2r}^{2/p-1}}{(2r)^{2/p-1}}\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^2\Big)^p\\ &\leqslant \frac{\gamma_{2r}^{2-p}(\gamma_{2r}+1)^p}{(2r)^{2-p}}\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^{2p} \leqslant 2^p\frac{\gamma_{2r}^{2-p}}{2^{2-p}r^{2-p}}\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^{2p},\end{aligned}$$

where we use in the second inequality a version of Stechkin's lemma (Kabanava et al., 2016, Lemma 3.1), which leads to the estimate

$$\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_2}^2 \leqslant \frac{\|\eta_{2r}^{(n+1)}\|_{S_2}^{2-p}}{(2r)^{2-p}}\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^p \leqslant \frac{\gamma_{2r}^{2/p-1}}{(2r)^{2/p-1}}\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^2.$$

Combining the estimate for $\|\eta^{(n+1)}\|_{S_2}^{2p}$ with statement (i), this results in

$$\|\eta^{(n+1)}\|_{S_2}^{2p} \leqslant 2^p \frac{\gamma_{2r}^{2-p}}{r^{2-p}}\Big(\sum_{i=r+1}^{d}\big(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\big)^{\frac{p}{2}}\Big)^{2-p}\|\eta_{\mathrm{vec}}^{(n+1)}\|_{\ell_2(\widetilde{W}^{(n)})}^{2p},$$

which shows statement (ii).

(iii) For the third statement, we use the strong Schatten-p NSP (18) to see that

$$\|\eta^{(n+1)}\|_{S_p}^p = \|\eta_{2r}^{(n+1)}\|_{S_p}^p + \|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^p \leqslant (1+\gamma_{2r})\|\eta^{(n+1)} - \eta_{2r}^{(n+1)}\|_{S_p}^p,$$

and combine this with statement (i). ∎

Lemma 21 *Let $(X^{(n)})_n$ be the output sequence of Algorithm 1 with parameters Φ, Y, r and $0 < p \leqslant 1$, and $\widetilde{W}^{(n)}$ be the harmonic mean weight matrix matrix (10) for $n \in \mathbb{N}$. Let $X_0 \in M_{d_1\times d_2}$ be a rank-r matrix such that $\Phi(X_0) = Y$ with condition number $\kappa := \frac{\sigma_1(X_0)}{\sigma_r(X_0)}$.*

(i) If (24) is fulfilled for iteration n, then $\eta^{(n+1)} = X^{(n)} - X_0$ fulfills

$$\Big\|\eta_{\mathrm{vec}}^{(n+1)}\Big\|_{\ell_2(\widetilde{W}^{(n)})}^{2p} \leqslant \frac{4^p r^{p/2}\sigma_r(X_0)^{p(p-1)}}{(1-\zeta)^{2p}}\kappa^p\frac{\|\eta^{(n)}\|_{S_\infty}^{2p-p^2}}{(\epsilon^{(n)})^{2p-p^2}}\|\eta^{(n+1)}\|_{S_2}^p.$$

(ii) Under the same assumption as for (i), it holds that

$$\Big\|\eta_{\mathrm{vec}}^{(n+1)}\Big\|_{\ell_2(\widetilde{W}^{(n)})}^{2p} \leqslant \frac{7^p r^{p/2}\max(r,d-r)^{p/2}\sigma_r(X_0)^{p(p-1)}}{(1-\zeta)^{2p}}\kappa^p\frac{\|\eta^{(n)}\|_{S_\infty}^{2p-p^2}}{(\epsilon^{(n)})^{2p-p^2}}\|\eta^{(n+1)}\|_{S_\infty}^p.$$

Proof (i) Recall that $X^{(n+1)} = \operatorname*{arg\,min}_{\Phi(X)=Y}\|X_{\mathrm{vec}}\|_{\ell_2(\widetilde{W}^{(n)})}^2$ is the minimizer of the weighted least squares problem with weight matrix $\widetilde{W}^{(n)}$. As $\eta^{(n+1)} = X^{(n+1)} - X_0$ is in the null space of the measurement map Φ, it follows from Lemma 16 that

$$0 = \langle\widetilde{W}^{(n)}X_{\mathrm{vec}}^{(n+1)}, \eta_{\mathrm{vec}}^{(n+1)}\rangle = \langle\widetilde{W}^{(n)}(\eta^{(n+1)} + X_0)_{\mathrm{vec}}, \eta_{\mathrm{vec}}^{(n+1)}\rangle,$$

which is equivalent to

$$\Big\|\eta_{\mathrm{vec}}^{(n+1)}\Big\|_{\ell_2(\widetilde{W}^{(n)})}^2 = \langle\widetilde{W}^{(n)}\eta_{\mathrm{vec}}^{(n+1)}, \eta_{\mathrm{vec}}^{(n+1)}\rangle = -\langle\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}, \eta_{\mathrm{vec}}^{(n+1)}\rangle.$$

Using Hölder's inequality, we can therefore estimate

$$\begin{aligned}\Big\|\eta_{\mathrm{vec}}^{(n+1)}\Big\|_{\ell_2(\widetilde{W}^{(n)})}^2 &= -\langle\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}, \eta_{\mathrm{vec}}^{(n+1)}\rangle_{\ell_2} = -\langle\big[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\big]_{\mathrm{mat}}, \eta^{(n+1)}\rangle_F \\ &\leqslant \big\|\big[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\big]_{\mathrm{mat}}\big\|_{S_2}\|\eta^{(n+1)}\|_{S_2}.\end{aligned} \tag{43}$$

To bound the first factor, we first rewrite the action of $\widetilde{W}^{(n)}$ on X_0 in the matrix space as

$$\begin{aligned}\Big[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\Big]_{\mathrm{mat}} &= 2[(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})\big(\bar{\Sigma}_{d_1}^{(n)2-p}\oplus\bar{\Sigma}_{d_2}^{(n)2-p}\big)^{-1}(\widetilde{V}^{(n)}\otimes\widetilde{U}^{(n)})^*(X_0)_{\mathrm{vec}}]_{\mathrm{mat}} = \\ &= \widetilde{U}^{(n)}\big(H^{(n)}\circ(\widetilde{U}^{(n)*}X_0\widetilde{V}^{(n)})\big)\widetilde{V}^{(n)*},\end{aligned}$$

using (39) and Lemma 20 about the action of inverses of Kronecker sums, with the notation that $H^{(n)} \in M_{d_1\times d_2}$ such that

$$H_{ij}^{(n)} = 2\Big[\mathbb{1}_{\{i\leqslant d\}}(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2})^{\frac{2-p}{2}} + \mathbb{1}_{\{j\leqslant d\}}(\sigma_j^2(X^{(n)}) + \epsilon^{(n)2})^{\frac{2-p}{2}}\Big]^{-1}$$

for $i \in [d_1]$, $j \in [d_2]$, where $\mathbb{1}_{\{i\leqslant d\}} = 1$ if $i \leqslant d$ and $\mathbb{1}_{\{i\leqslant d\}} = 0$ otherwise. This enables us to estimate

$$\begin{aligned}\Big\|\big[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\big]_{\mathrm{mat}}\Big\|_{S_2}^2 &= \Big\|\widetilde{U}^{(n)}\big(H^{(n)}\circ(\widetilde{U}^{(n)*}X_0\widetilde{V}^{(n)})\big)\widetilde{V}^{(n)*}\Big\|_{S_2}^2 = \Big\|H^{(n)}\circ(\widetilde{U}^{(n)*}X_0\widetilde{V}^{(n)})\Big\|_{S_2}^2 \\ &= \left\|H^{(n)}\circ\begin{pmatrix}U_T^{(n)*}X_0V_T^{(n)} & U_T^{(n)*}X_0V_{T_c}^{(n)}\\ U_{T_c}^{(n)*}X_0V_T^{(n)} & U_{T_c}^{(n)*}X_0V_{T_c}^{(n)}\end{pmatrix}\right\|_{S_2}^2 \\ &= \Big\|H_{T,T}^{(n)}\circ(U_T^{(n)*}X_0V_T^{(n)})\Big\|_{S_2}^2 + \Big\|H_{T,T_c}^{(n)}\circ(U_T^{(n)*}X_0V_{T_c}^{(n)})\Big\|_{S_2}^2 \\ &+ \Big\|H_{T_c,T}^{(n)}\circ(U_{T_c}^{(n)*}X_0V_T^{(n)})\Big\|_{S_2}^2 + \Big\|H_{T_c,T_c}^{(n)}\circ(U_{T_c}^{(n)*}X_0V_{T_c}^{(n)})\Big\|_{S_2}^2,\end{aligned} \tag{44}$$

using the notation from the proof of Lemma 20. To bound the first summand, we calculate

$$\begin{aligned}&\Big\|H_{T,T}^{(n)}\circ(U_T^{(n)*}X_0V_T^{(n)})\Big\|_{S_2} \leqslant \Big\|H_{T,T}^{(n)}\circ(U_T^{(n)*}X^{(n)}V_T^{(n)})\Big\|_{S_2} + \Big\|H_{T,T}^{(n)}\circ(-U_T^{(n)*}\eta^{(n)}V_T^{(n)})\Big\|_{S_2} \\ &\leqslant \Big\|H_{T,T}^{(n)}\circ\Sigma_T^{(n)}\Big\|_{S_2} + \Big\|H_{T,T}^{(n)}\circ(U_T^{(n)*}\eta^{(n)}V_T^{(n)})\Big\|_{S_2} \\ &\leqslant \Big(\sum_{i=1}^{r}\frac{\sigma_i^2(X^{(n)})}{\big(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\big)^{2-p}}\Big)^{1/2} + \max_{i,j=1}^{r}|H_{i,j}^{(n)}|\|U_T^{(n)*}\eta^{(n)}V_T^{(n)}\|_{S_2} \\ &\leqslant \sqrt{r}\sigma_r^{p-1}(X^{(n)}) + (\sigma_r^2(X^{(n)}) + \epsilon^{(n)2})^{\frac{p-2}{2}}\|U_T^{(n)*}\eta^{(n)}V_T^{(n)}\|_{S_2} \\ &\leqslant \sqrt{r}\sigma_r^{p-1}(X^{(n)}) + \sigma_r^{p-2}(X^{(n)})\sqrt{r}\|\eta^{(n)}\|_{S_\infty} = \sqrt{r}\sigma_r^{p-2}(X^{(n)})\big[\sigma_r(X^{(n)}) + \|\eta^{(n)}\|_{S_\infty}\big],\end{aligned}$$

denoting $\Sigma_T^{(n)} = \mathrm{diag}(\sigma_i(X^{(n)}))_{i=1}^r$ and that the matrices $U_T^{(n)}$ and $V_T^{(n)}$ contain the first r left resp. right singular vectors of $X^{(n)}$ in the second inequality, together with the estimates $\|X\|_{S_1} \leqslant \sqrt{r}\|X\|_{S_2} \leqslant r\|X\|_{S_\infty}$ for $(r\times r)$-matrices X.

With the notations $s_r^0 := \sigma_r(X_0)$ and $s_1^0 := \sigma_1(X_0)$, we note that

$$\sigma_r(X^{(n)}) \geqslant s_r^0(1-\zeta),$$

as the assumption (24) implies that

$$s_r^0 = \sigma_r(X_0) = \sigma_r(X^{(n)} - \eta^{(n)}) \leqslant \sigma_r(X^{(n)}) + \sigma_1(\eta^{(n)}) \leqslant \sigma_r(X^{(n)}) + \zeta s_r^0,$$

using Bernstein (2009, Proposition 9.6.8) in the first inequality.

Therefore, we can bound the first summand of (44) such that

$$\left\| H_{T,T}^{(n)} \circ (U_T^{(n)*} X_0 V_T^{(n)}) \right\|_{S_2} \leqslant \sqrt{r}(s_r^0(1-\zeta))^{p-2}[s_r^0(1-\zeta)+\zeta s_r^0] = \sqrt{r}(s_r^0)^{p-1}(1-\zeta)^{p-2}. \quad (45)$$

For the second summand in the estimate of $\left\| \left[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\right]_{\mathrm{mat}} \right\|_{S_2}^2$, similar arguments and again assumption (24) are used to compute

$$\begin{aligned}
&\left\| H_{T,T_c}^{(n)} \circ (U_T^{(n)*} X_0 V_{T_c}^{(n)}) \right\|_{S_2} \leqslant \left\| H_{T,T_c}^{(n)} \circ \overbrace{(U_T^{(n)*} X^{(n)} V_{T_c}^{(n)})}^{=0} \right\|_{S_2} + \\
&\left\| H_{T,T_c}^{(n)} \circ (U_T^{(n)*} \eta^{(n)} V_{T_c}^{(n)}) \right\|_{S_2} \leqslant \max_{\substack{i\in[r] \\ j\in\{r+1,\dots,d_2\}}} |H_{i,j}^{(n)}| \|U_T^{(n)*} \eta^{(n)} V_{T_c}^{(n)}\|_{S_2} \\
&\leqslant \frac{2\|U_T^{(n)*} \eta^{(n)} V_{T_c}^{(n)}\|_F}{\left[(\sigma_r(X^{(n)})^2 + \epsilon^{(n)2})^{\frac{2-p}{2}}\right]} \leqslant 2\sigma_r(X^{(n)})^{p-2} \|U_T^{(n)*} \eta^{(n)} V_{T_c}^{(n)}\|_{S_2} \\
&\leqslant 2\sqrt{r}(s_r^0(1-\zeta))^{p-2}\|\eta^{(n)}\|_{S_\infty} \leqslant 2\zeta\sqrt{r}(s_r^0)^{p-1}(1-\zeta)^{p-2}.
\end{aligned} \quad (46)$$

From exactly the same arguments it follows that also

$$\left\| H_{T_c,T}^{(n)} \circ (U_{T_c}^{(n)*} X_0 V_T^{(n)}) \right\|_{S_2} \leqslant 2\zeta\sqrt{r}(s_r^0)^{p-1}(1-\zeta)^{p-2}. \quad (47)$$

It remains to bound the last summand $\left\| H_{T_c,T_c}^{(n)} \circ (U_{T_c}^{(n)*} X_0 V_{T_c}^{(n)}) \right\|_{S_2}^2$. We see that

$$\begin{aligned}
&\left\| H_{T_c,T_c}^{(n)} \circ (U_{T_c}^{(n)*} X_0 V_{T_c}^{(n)}) \right\|_{S_2} \leqslant \max_{\substack{i\in\{r+1,\dots,d_1\} \\ j\in\{r+1,\dots,d_2\}}} |H_{i,j}^{(n)}| \|U_{T_c}^{(n)*} X_0 V_{T_c}^{(n)}\|_{S_2} \\
&\leqslant (\epsilon^{(n)})^{p-2} \|U_{T_c}^{(n)*} X_0 V_{T_c}^{(n)}\|_{S_2} \leqslant (\epsilon^{(n)})^{p-2} \|U_{T_c}^{(n)*} U_T^0\|_{S_\infty} \|S^0\|_{S_2} \|V_T^{0*} V_{T_c}^{(n)}\|_{S_\infty} \\
&\leqslant (\epsilon^{(n)})^{p-2} \frac{\sqrt{2}\|\eta^{(n)}\|_{S_\infty}}{(1-\zeta)s_r^0} \sqrt{r} s_1^0 \frac{\sqrt{2}\|\eta^{(n)}\|_{S_\infty}}{(1-\zeta)s_r^0} = 2\sqrt{r}\|\eta^{(n)}\|_{S_\infty}^2 (\epsilon^{(n)})^{p-2}(1-\zeta)^{-2}(s_r^0)^{-1}\frac{s_1^0}{s_r^0},
\end{aligned} \quad (48)$$

where Hölder's inequality for Schatten norms was used in the third inequality. In the fourth inequality, Wedin's singular value perturbation bound of Lemma 19 is used with the choice $Z = X_0$, $\bar{Z} = X^{(n)}$, $\alpha = s_r^0$ and $\delta = (1-\zeta)s_r^0$, and finally $\epsilon^{(n)} \leqslant \zeta s_r^0$ in the last inequality, which is implied by the rule (14) for $\epsilon^{(n)}$ together with assumption (24).

Summarizing the estimates (45)–(48), we conclude that

$$\begin{aligned}
&\left\| \left[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\right]_{\mathrm{mat}} \right\|_{S_2}^2 \leqslant \frac{r(s_r^0)^{2p-2}}{(1-\zeta)^{4-2p}} \left[1 + 8\zeta^2 + 4\frac{\|\eta^{(n)}\|_{S_\infty}^4}{(1-\zeta)^{2p}} (\epsilon^{(n)})^{2p-4} (s_r^0)^{-2p} \left(\frac{s_1^0}{s_r^0}\right)^2 \right] \\
&= \frac{r(s_r^0)^{2p-2}}{(1-\zeta)^4} \left[(1+8\zeta^2)(1-\zeta)^{2p} + 4\frac{\|\eta^{(n)}\|_{S_\infty}^{4-2p}}{(\epsilon^{(n)})^{4-2p}} \frac{\|\eta^{(n)}\|_{S_\infty}^{2p}}{(s_r^0)^{2p}} \left(\frac{s_1^0}{s_r^0}\right)^2 \right] \\
&\leqslant \frac{r(s_r^0)^{2p-2}}{(1-\zeta)^4} \left[9 + 4\frac{\|\eta^{(n)}\|_{S_\infty}^{4-2p}}{(\epsilon^{(n)})^{4-2p}} \zeta^{2p}\kappa^2 \right] \leqslant \frac{13r(s_r^0)^{2p-2}}{(1-\zeta)^4} \left[\frac{\|\eta^{(n)}\|_{S_\infty}^{4-2p}}{(\epsilon^{(n)})^{4-2p}} \kappa^2 \right],
\end{aligned}$$

as $0 < \zeta < 1$, $\epsilon^{(n)} \leqslant \sigma_{r+1}(X^{(n)}) = \|X_{T_c}^{(n)}\|_{S_\infty} \leqslant \|\eta^{(n)}\|_{S_\infty}$ and using the assumption (24) in the second inequality. This concludes the proof of Lemma 21(i) together with inequality (43) as $13^{p/2} \leqslant 16^{p/2} = 4^p$.

(ii) For the second statement of Lemma 21, we proceed similarly as before, but note that by Hölder's inequality, also

$$\left\| \eta_{\mathrm{vec}}^{(n+1)} \right\|_{\ell_2(\widetilde{W}^{(n)})}^2 \leqslant \left\| \left[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\right]_{\mathrm{mat}} \right\|_{S_1} \|\eta^{(n+1)}\|_{S_\infty}, \quad (49)$$

cf. (43). Furthermore

$$\begin{aligned}
\left\| \left[\widetilde{W}^{(n)}(X_0)_{\mathrm{vec}}\right]_{\mathrm{mat}} \right\|_{S_1} &\leqslant \left\| H_{T,T}^{(n)} \circ (U_T^{(n)*} X_0 V_T^{(n)}) \right\|_{S_1} + \left\| H_{T,T_c}^{(n)} \circ (U_T^{(n)*} X_0 V_{T_c}^{(n)}) \right\|_{S_1} \\
&+ \left\| H_{T_c,T}^{(n)} \circ (U_{T_c}^{(n)*} X_0 V_T^{(n)}) \right\|_{S_1} + \left\| H_{T_c,T_c}^{(n)} \circ (U_{T_c}^{(n)*} X_0 V_{T_c}^{(n)}) \right\|_{S_1}.
\end{aligned} \quad (50)$$

The four Schatten-1 norms can then be estimated by $\max(r,(d-r))^{1/2}$ times the corresponding Schatten-2 norms. Using then again inequalities (45)–(48), we conclude the proof of (ii). ∎

We proceed now to the proof of Theorem 11.

Proof First we note that

$$\left(\sum_{i=r+1}^{d} \left(\sigma_i^2(X^{(n)}) + \epsilon^{(n)2}\right)^{\frac{p}{2}} \right)^{2-p} \leqslant 2^{p-\frac{p^2}{2}} (d-r)^{2-p} \sigma_{r+1}(X^{(n)})^{p(2-p)} \quad (51)$$

as $\epsilon^{(n)} \leqslant \sigma_{r+1}(X^{(n+1)})$ due to the choice of $\epsilon^{(n)}$ in (14). We proceed by induction over $n \geqslant \bar{n}$. Theorem 20(ii) and Theorem 21(ii) imply together with (51) that for $n = \bar{n}$,

$$\begin{aligned}
\|\eta^{(n+1)}\|_{S_\infty}^p &\leqslant \frac{\|\eta^{(n+1)}\|_{S_2}^{2p}}{\|\eta^{(n+1)}\|_{S_\infty}^p} \leqslant 2^p \gamma_{2r}^{2-p} 2^{p-\frac{p^2}{2}} \left(\frac{d-r}{r}\right)^{2-p/2} \frac{7^p r^p (s_r^0)^{p(p-1)}}{(1-\zeta)^{2p}} \kappa^p \|\eta^{(n)}\|_{S_\infty}^{2p-p^2} \\
&\leqslant 2^{5p} \gamma_{2r}^{2-p} \left(\frac{d-r}{r}\right)^{2-p/2} \frac{r^p (s_r^0)^{p(p-1)}}{(1-\zeta)^{2p}} \kappa^p \|\eta^{(n)}\|_{S_\infty}^{p(2-p)}
\end{aligned} \quad (52)$$

as $\sigma_{r+1}(X^{(n)}) = \epsilon^{(n)}$ by assumption for $n = \bar{n}$.

Similarly, by Lemma 20(iii), Lemma 21(ii) and (51), the error in the Schatten-p quasi-norm fulfills

$$\|\eta^{(n+1)}\|_{S_p}^{2p} \leqslant (1+\gamma_{2r})^2 2^{2+2p}\big(d-r\big)^{2-p}\frac{r^{p/2}(s_r^0)^{p(p-1)}}{(1-\zeta)^{2p}}\kappa^p\|\eta^{(n)}\|_{S_\infty}^{p(2-p)}\|\eta^{(n+1)}\|_{S_2}^p \tag{53}$$

for $n=\bar{n}$. Using the strong Schatten-p null space property of order $2r$ for the operator Φ, we see with the arguments of the proof of Lemma 20(ii) that

$$\|\eta^{(n)}\|_{S_\infty}^p \leqslant \|\eta^{(n)}\|_{S_2}^p \leqslant \frac{2^{p-1}\gamma_{2r}^{1-p/2}}{r^{1-p/2}}\|\eta^{(n)}\|_{S_p}^p$$

and also $\|\eta^{(n+1)}\|_{S_2}^p \leqslant \frac{2^{p-1}\gamma_{2r}^{1-p/2}}{r^{1-p/2}}\|\eta^{(n+1)}\|_{S_p}^p$. Inserting this in (53) and dividing by $\|\eta^{(n+1)}\|_{S_p}^p$, we obtain

$$\|\eta^{(n+1)}\|_{S_p}^p \leqslant 2^{4p}(1+\gamma_{2r})^2\gamma_{2r}^{2-p}\Big(\frac{d-r}{r}\Big)^{2-p}\frac{r^{p/2}(s_r^0)^{p(p-1)}}{(1-\zeta)^{2p}}\kappa^p\|\eta^{(n)}\|_{S_\infty}^{p(1-p)}\|\eta^{(n)}\|_{S_p}^p.$$

Under the assumption that (25) holds, it follows from this and (52) that

$$\|\eta^{(n+1)}\|_{S_\infty}^p \leqslant \|\eta^{(n)}\|_{S_\infty}^p \text{ and } \|\eta^{(n+1)}\|_{S_p}^p \leqslant \|\eta^{(n)}\|_{S_p}^p \tag{54}$$

for $n=\bar{n}$, which also entails the statement of Theorem 11 for this iteration.

Let now $n'>\bar{n}$ such that (54) is true for all n with $n'>n\geqslant\bar{n}$.

If $\sigma_{r+1}(X^{(n')})\leqslant\epsilon^{(n'-1)}$, then $\epsilon^{(n')}=\sigma_{r+1}(X^{(n')})$ and the arguments from above show (54) also in the case $n=n'$.

Otherwise $\sigma_{r+1}(X^{(n')})>\epsilon^{(n'-1)}$ and there exists $n'>n''\geqslant\bar{n}$ such that $\epsilon^{(n')}=\epsilon^{(n'')}=\sigma_{r+1}(X^{(n'')})$. Then

$$\|\eta^{(n'+1)}\|_{S_\infty}^p \leqslant 14^p\frac{\gamma_{2r}^{2-p}}{r^{2-p}}\Big[\sum_{i=r+1}^d\Big(\frac{\sigma_i^2(X^{(n')})}{\epsilon^{(n'')2}}+1\Big)^{\frac{p}{2}}\Big]^{2-p}\frac{r^{p/2}\max(r,d-r)^{p/2}}{(s_r^0)^{p(1-p)}(1-\zeta)^{2p}}\kappa^p\|\eta^{(n')}\|_{S_\infty}^{p(2-p)}$$

and we compute

$$\begin{aligned}
\Big[\sum_{i=r+1}^d\Big(\frac{\sigma_i^2(X^{(n')})}{\epsilon^{(n'')2}}+1\Big)^{\frac{p}{2}}\Big]^{2-p} &\leqslant \Big[\sum_{i=r+1}^d\frac{\sigma_i^p(X^{(n')})}{\epsilon^{(n'')p}}+(d-r)\Big]^{2-p}\\
&\leqslant \Big[\frac{\|\eta^{(n')}\|_{S_p}^p}{\epsilon^{(n'')p}}+(d-r)\Big]^{2-p} \leqslant \Big[\frac{\|\eta^{(n'')}\|_{S_p}^p}{\epsilon^{(n'')p}}+(d-r)\Big]^{2-p}\\
&\leqslant \Big[\frac{2(1+\gamma_{2r})\|X_{T_c}^{(n'')}\|_{S_p}^p}{(1-\gamma_{2r})\epsilon^{(n'')p}}+(d-r)\Big]^{2-p} \leqslant \Big(\frac{3+\gamma_{2r}}{1-\gamma_{2r}}\Big)^{2-p}(d-r)^{2-p},
\end{aligned}$$

using that X_0 is a matrix of rank at most r in the second inequality, the inductive hypothesis in the third inequality and an analogue of (61) for a Schatten-p quasi-norm on the left hand side (cf. Kabanava et al. (2016, Lemma 3.2) for the corresponding result for $p=1$) in the last inequality. The latter argument uses the assumption on the null space property. This shows that

$$\|\eta^{(n'+1)}\|_{S_\infty}^p \leqslant \mu\|\eta^{(n')}\|_{S_\infty}^{p(2-p)}$$

for

$$\widetilde{\mu} := 2^{4p}\gamma_{2r}^{2-p}\Big(\frac{(3+\gamma_{2r})(d-r)}{(1-\gamma_{2r})r}\Big)^{2-p}\frac{r^{p/2}(s_r^0)^{p(p-1)}}{(1-\zeta)^{2p}}\kappa^p\max\Big(2^p(d-r)^{\frac{p}{2}},(1+\gamma_{2r})^2\Big),$$

and $\|\eta^{(n'+1)}\|_{S_\infty}^p\leqslant\|\eta^{(n')}\|_{S_\infty}^p$ under the assumption (25) of Theorem 11, as $\widetilde{\mu}\leqslant\mu$ with μ as in (26). Indeed, it holds that $\widetilde{\mu}\leqslant\mu$ since

$$\max\Big(2^p(d-r)^{\frac{p}{2}},(1+\gamma_{2r})^2\Big)\Big(\frac{d-r}{r}\Big)^{2-p}r^{p/2} \leqslant 2^p(1+\gamma_{2r})^2\Big(\frac{d-r}{r}\Big)^{2-p/2}r^p.$$

The same argument shows that $\|\eta^{(n'+1)}\|_{S_p}^p\leqslant\|\eta^{(n')}\|_{S_p}^p$, which finishes the proof. ■

Remark 22 *We note that the weight matrices of the previous IRLS approaches* `IRLS-col` *and* `IRLS-row` *Fornasier et al. (2011); Mohan and Fazel (2012) at iteration n could be expressed in our notation as*

$$\mathbf{I}_{d_2}\otimes W_L^{(n)} := \mathbf{I}_{d_2}\otimes U^{(n)}(\bar{\Sigma}_{d_1}^{(n)})^{p-2}U^{(n)*}$$

and

$$W_R^{(n)}\otimes\mathbf{I}_{d_1} := V^{(n)}(\bar{\Sigma}_{d_2}^{(n)})^{p-2}V^{(n)*}\otimes\mathbf{I}_{d_1},$$

respectively, cf. Section 2.2, if $X^{(n)}=U^{(n)}\Sigma^{(n)}V^{(n)*}=U_T^{(n)}\Sigma_T^{(n)}V_T^{(n)*}+U_{T_c}^{(n)}\Sigma_{T_c}^{(n)}V_{T_c}^{(n)*}$ is the SVD of the iterate $X^{(n)}$ with $U_T^{(n)}$ and $V_T^{(n)}$ containing the r first left- and right singular vectors.

Now let

$$T^{(n)} := \big\{U_T^{(n)}Z_1^*+Z_2V_T^{(n)*} : Z_1\in M_{d_1\times r}, Z_2\in M_{d_2\times r}\big\}$$

be the tangent space of the smooth manifold of rank-r matrices at the best rank-r approximation $U_T^{(n)}\Sigma_T^{(n)}V_T^{(n)*}$ of $X^{(n)}$, or, put differently, the direct sum of the row and column spaces of $U_T^{(n)}\Sigma_T^{(n)}V_T^{(n)*}$.

The fact that left- or right-sided weight matrices do not lead to algorithms with superlinear convergence rates for $p<1$ can be explained by noting that there are always parts of the space $T^{(n)}$ that are equipped with too large weights if $X^{(n)}=U^{(n)}\Sigma^{(n)}V^{(n)*}$ is already approximately low-rank. In particular, proceeding as in (44), we obtain for $\mathbf{I}_{d_2}\otimes W_L^{(n)}$

$$\begin{aligned}
\Big\|\big[\mathbf{I}_{d_2}\otimes W_L^{(n)}(X_0)_{\mathrm{vec}}\big]_{\mathrm{mat}}\Big\|_{S_2}^2 &= \Big\|(\bar{\Sigma}_T^{(n)})^{p-2}U_T^{(n)*}X_0V_T^{(n)}\Big\|_{S_2}^2+\Big\|(\bar{\Sigma}_T^{(n)})^{p-2}U_T^{(n)*}X_0V_{T_c}^{(n)}\Big\|_{S_2}^2\\
&\quad+\Big\|\big(\bar{\Sigma}_{T_c}{}^{(n)}\big)^{p-2}U_{T_c}^{(n)*}X_0V_T^{(n)}\Big\|_{S_2}^2+\Big\|(\bar{\Sigma}_{T_c}^{(n)})^{p-2}U_{T_c}^{(n)*}X_0V_{T_c}^{(n)}\Big\|_{S_2}^2
\end{aligned}$$

if $\bar{\Sigma}_T^{(n)}$ denotes the diagonal matrix with the first r non-zero entries of $\bar{\Sigma}_{d_1}^{(n)}$ and $\bar{\Sigma}_{T_c}^{(n)}$ the one of the remaining entries.

Here, the third of the four summands would become too large for $p<1$ to allow for a superlinear convergence when the last $d-r$ singular values of $X^{(n)}$ approach zero. An analogous argument can be used for the right-sided weight matrix $W_R^{(n)}\otimes\mathbf{I}_{d_1}$ and, notably, also for arithmetic mean weight matrices $W_{(arith)}^{(n)}=\mathbf{I}_{d_2}\otimes W_L^{(n)}+W_R^{(n)}\otimes\mathbf{I}_{d_1}$, cf. Section 2.3.

Acknowledgments

The two authors acknowledge the support and hospitality of the Hausdorff Research Institute for Mathematics (HIM) during the early stage of this work within the HIM Trimester Program "Mathematics of Signal Processing". C.K. is supported by the German Research Foundation (DFG) in the context of the Emmy Noether Junior Research Group "Randomized Sensing and Quantization of Signals and Images" (KR 4512/1-1) and the ERC Starting Grant "High-Dimensional Sparse Optimal Control" (HDSPCONTR - 306274). J.S. is supported by the DFG through the D-A-CH project no. I1669-N26 and through the international research training group IGDK 1754 "Optimization and Numerical Analysis for Partial Differential Equations with Nonsmooth Structures". The authors thank Ke Wei for providing code of his implementations. They also thank Massimo Fornasier for helpful discussions.

Appendix A. Kronecker and Hadamard products

For two matrices $A = (a_{ij})_{i\in[d_1],j\in[d_3]} \in \mathbb{C}^{d_1\times d_3}$ and $B \in \mathbb{C}^{d_2\times d_4}$, we call the matrix representation of their tensor product with respect to the standard bases the *Kronecker product* $A\otimes B \in \mathbb{C}^{d_1\cdot d_2\times d_3\cdot d_4}$. By its definition, $A\otimes B$ is a block matrix of $d_2 \times d_4$ blocks whose block of index $(i, j) \in [d_1] \times [d_3]$ is the matrix $a_{ij}B \in \mathbb{R}^{d_2\times d_4}$. This implies, e.g., for $A \in \mathbb{C}^{d_1\times d_3}$ with $d_1 = 2$ and $d_3 = 3$ that

$$A \otimes B = \left[\begin{array}{ccc} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{array}\right] \otimes B = \left[\begin{array}{c|c|c} a_{11}B & a_{12}B & a_{13}B \\ \hline a_{21}B & a_{22}B & a_{23}B \end{array}\right].$$

The Kronecker product is useful for the elegant formulation of matrix equations involving left and right matrix multiplications with the variable X, as

$$AXB^* = Y \qquad \text{if and only if} \qquad (B\otimes A)X_{\mathrm{vec}} = Y_{\mathrm{vec}}.$$

We define the *Hadamard product* $A \circ B \in \mathbb{C}^{d_1\times d_2}$ of two matrices $A \in \mathbb{C}^{d_1\times d_2}$ and $B \in \mathbb{C}^{d_1\times d_2}$ as their entry-wise product

$$(A \circ B)_{i,j} = A_{i,j}B_{i,j}$$

with $i \in [d_1]$ and $j \in [d_2]$. The Hadamard product is also known as *Schur product* in the literature.

Furthermore, if $d_1 = d_3$ and $d_2 = d_4$, we define the *Kronecker sum* $A \oplus B \in \mathbb{C}^{d_1d_2\times d_1d_2}$ of two matrices $A \in \mathbb{C}^{d_1\times d_1}$ and $B \in \mathbb{C}^{d_2\times d_2}$ as the matrix

$$A \oplus B = (\mathbf{I}_{d_2} \otimes A) + (B \otimes \mathbf{I}_{d_1}). \tag{55}$$

Note that equations of the form $AX + XB^* = Y$ can be rewritten as

$$(A \oplus B)X_{\mathrm{vec}} = Y_{\mathrm{vec}},$$

using again the vectorizations of X and Y. An explicit formula that expresses the inverse $(A \oplus B)^{-1}$ of the Kronecker sum $A \oplus B$ is provided by the following lemma.

Lemma 23 (Jameson (1968)) *Let $A \in H_{d_1\times d_1}$ and $B \in H_{d_2\times d_2}$, where one of the matrices is positive definite and the other positive semidefinite. If we denote the singular vectors of A by $u_i \in \mathbb{C}^{d_1}$, $i \in [d_1]$, its singular values by σ_i, $i \in [d_1]$ and the singular vectors resp. values of B by $v_j \in \mathbb{C}^{d_2}$ resp. μ_j, $j \in [d_2]$, then*

$$(A \oplus B)^{-1} = \sum_{i=1}^{d_1}\sum_{j=1}^{d_2} \frac{v_jv_j^* \otimes u_iu_i^*}{\sigma_i + \mu_j}. \tag{56}$$

Furthermore, the action of $(A \oplus B)^{-1}$ on the matrix space $M_{d_1\times d_2}$ can be written as

$$\big[(A \oplus B)^{-1}Z_{\mathrm{vec}}\big]_{\mathrm{mat}} = U\big(H \circ (U^*ZV)\big)V^*. \tag{57}$$

for $Z \in M_{d_1\times d_2}$, $U = [u_1, \dots, u_{d_1}]$, and $V = [v_1, \dots, v_{d_2}]$ and the matrix $H \in M_{d_1\times d_2}$ with the entries $H_{i,j} = (\sigma_i + \mu_j)^{-1}$, $i \in [d_1]$, $j \in [d_2]$.

Appendix B. Proofs of preliminary statements in Section 6

B.1 Proof of Lemma 14: Main part

First, we define the function

$$f^p_{X,\epsilon}(Z) = \mathcal{J}_p(X, \epsilon, Z) = \begin{cases} \frac{p}{2}\|X_{\mathrm{vec}}\|^2_{\ell_2(\widetilde{W}(Z))} + \frac{\epsilon^2 p}{2}\sum_{i=1}^{d}\sigma_i(Z) + \frac{2-p}{2}\sum_{i=1}^{d}\sigma_i(Z)^{\frac{p}{(p-2)}} & \text{if } \operatorname{rank}(Z) = d, \\ +\infty & \text{if } \operatorname{rank}(Z) < d, \end{cases}$$

for $X \in M_{d_1\times d_2}$, $\epsilon > 0$ fixed and with $Z \in M_{d_1\times d_2}$ as its only argument. We note that the set of minimizers of $f^p_{X,\epsilon}(Z)$ does not contain an instance Z with rank smaller than d as the value of $f^p_{X,\epsilon}(Z)$ is infinite at such points and, therefore, it is sufficient to search for minimizers on the set $\Omega = \{Z \in M_{d_1\times d_2} | \operatorname{rank}(Z) = d\}$ of matrices with rank d. We observe that the set Ω is an open set and that we have that

(a) $f^p_{X,\epsilon}(Z)$ is lower semi-continuous, which means that any sequence $(Z^k)_{k\in\mathbb{N}}$ with $Z^k \overset{k\to\infty}{\longrightarrow} Z$ fulfills $\liminf_{k\to\infty} f^p_{X,\epsilon}(Z^k) \geqslant f^p_{X,\epsilon}(Z)$,

(b) $f^p_{X,\epsilon}(Z) \geqslant \alpha$ for all $Z \in M_{d_1\times d_2}$ for some constant α,

(c) $f^p_{X,\epsilon}(Z)$ is coercive, i.e., for any sequence $(Z^k)_{k\in\mathbb{N}}$ with $\|Z^k\|_F \overset{k\to\infty}{\longrightarrow} \infty$, we have $f^p_{X,\epsilon}(Z^k) \overset{k\to\infty}{\longrightarrow} \infty$.

Property (a) is true as $f^p_{X,\epsilon}(Z))|_\Omega$ is a concatenation of an indicator function of an open set, which is lower semi-continuous and a sum of continuous functions on Ω. Property (b) is obviously true for the choice $\alpha = 0$.

To justify point (c), we note that $f^p_{X,\epsilon}(Z) > \frac{\epsilon^2 p}{2}\sum_{i=1}^{d}\sigma_i(Z) = \frac{\epsilon^2 p}{2}\|Z\|_{S_1} \geqslant \frac{\epsilon^2 p}{2}\|Z\|_F$ and therefore, coercivity is clear from its definition. As a consequence from (a) and (c), it is

also true that the level sets $L_C = \left\{ Z \in M_{d_1 \times d_2} | f^p_{X,\epsilon}(Z) \leqslant C \right\}$ are closed and bounded and, therefore, compact.

Via the direct method of the calculus of variations, we conclude from the properties (a)–(c) that $f^p_{X,\epsilon}(Z)$ has at least one global minimizer belonging to the set of critical points of $f^p_{X,\epsilon}(Z)$ (Dacorogna, 1989, Theorem 1).

To characterize the set of critical points of $f^p_{X,\epsilon}(Z)$, its derivative with respect to Z is calculated explicitly and equated with zero in Subsection B.2. The solution of the resulting equation reveals that $Z_{\text{opt}} = \sum_{i=1}^d (\sigma_i^2(X) + \epsilon^2)^{\frac{p-2}{2}} u_i v_i^* =: \sum_{i=1}^d \tilde{\sigma}_i u_i v_i^*$ is the only critical point and consequently the unique global minimizer of $f^p_{X,\epsilon}(Z)$. We define the matrices $W^L_{\text{opt}} := \sum_{i=1}^d \tilde{\sigma}_i u_i u_i^*$ and $W^R_{\text{opt}} := \sum_{i=1}^d \tilde{\sigma}_i v_i v_i^*$, and note that $\widetilde{W}(Z_{\text{opt}}) = 2\big((W^R_{\text{opt}})^{-1} \oplus (W^L_{\text{opt}})^{-1}\big)^{-1}$ with Definition 13. To verify the second part of the theorem, we simply plug the optimal solution Z_{opt} into the functional $\mathcal{J}_p$ and compute using (56) that

$$\begin{aligned}
\mathcal{J}_p(X, \epsilon, Z_{\text{opt}}) &= \frac{p}{2}\|X_{\text{vec}}\|^2_{\ell_2(\widetilde{W}(Z_{\text{opt}}))} + \frac{\epsilon^2 p}{2}\sum_{i=1}^d \tilde{\sigma}_i + \frac{2-p}{2}\sum_{i=1}^d \tilde{\sigma}_i^{\frac{p}{p-2}} \\
&= \frac{p}{2}\sum_{i=1}^d \left[\sigma_i^2(X)(u_i^* \otimes v_i^*) 2 \left(\sum_{k=1}^{d_2}\sum_{j=1}^{d_1} \frac{u_k u_k^* \otimes v_j v_j^*}{\tilde{\sigma}_k^{-1} + \tilde{\sigma}_j^{-1}} \right)(u_i \otimes v_i) \right]_{ii} + \frac{\epsilon^2 p}{2}\sum_{i=1}^d \tilde{\sigma}_i + \frac{2-p}{2}\sum_{i=1}^d \tilde{\sigma}_i^{\frac{p}{p-2}} \\
&= \frac{p}{2}\sum_{i=1}^d (\sigma_i^2(X) + \epsilon^2)\tilde{\sigma}_i + \frac{2-p}{2}\sum_{i=1}^d \tilde{\sigma}_i^{\frac{p}{p-2}} \\
&= \frac{p}{2}\sum_{i=1}^d (\sigma_i^2(X) + \epsilon^2)(\sigma_i^2(X) + \epsilon^2)^{\frac{p-2}{2}} + \frac{2-p}{2}\sum_{i=1}^d (\sigma_i^2(X) + \epsilon^2)^{\frac{p}{2}} \\
&= \sum_{i=1}^d (\sigma_i^2(X) + \epsilon^2)^{\frac{p}{2}}.
\end{aligned}$$

B.2 Proof of Lemma 14: Critical points of $f^p_{X,\epsilon}$

Let us without loss of generality consider the case $d = d_1 = d_2$ and define

$$\Omega = \{ Z \in M_{d \times d} \text{ s.t. } \operatorname{rank}(Z) = d \}.$$

As already mentioned in (27), the harmonic mean matrix $\widetilde{W}(Z)$ can then be rewritten as

$$\widetilde{W}(Z) = 2\widetilde{W}_1\big(\widetilde{W}_1 + \widetilde{W}_2\big)^{-1}\widetilde{W}_2 = 2(\widetilde{W}_1^{-1} + \widetilde{W}_2^{-1})^{-1}$$

for $Z \in \Omega$ with the definitions $\widetilde{W}_1 := \mathbf{I}_d \otimes (ZZ^*)^{\frac{1}{2}}$ and $\widetilde{W}_2 = (Z^*Z)^{\frac{1}{2}} \otimes \mathbf{I}_d$. For $Z \in \Omega$, we reformulate the auxiliary functional such that

$$\begin{aligned}
f^p_{X,\epsilon}(Z) = \mathcal{J}^p(X, \epsilon, Z) &= \frac{p}{2}\|X_{\text{vec}}\|^2_{\ell_2(\widetilde{W}(Z))} + \frac{\epsilon^2 p}{2}\sum_{i=1}^d \sigma_i(Z) + \frac{2-p}{2}\sum_{i=1}^d \sigma_i(Z)^{\frac{p}{(p-2)}} \\
&= \frac{p}{2}\|X_{\text{vec}}\|^2_{\ell_2(\widetilde{W}(Z))} + \frac{\epsilon^2 p}{2}\|(Z^*Z)^{1/2}\|_F^2 + \frac{2-p}{2}\|(Z^*Z)^{\frac{p}{2(p-2)}}\|_F^2.
\end{aligned}$$

To identify the set of critical points of $f^p_{X,\epsilon}(Z)$ located in Ω, we compute its derivative with respect to Z using the derivative rules (7), (12), (13), (15), (16), (18), (20) in Chapter 8.2

and Theorem 3 in Chapter 8.4 of (Magnus and Neudecker, 1999) in the following. Using the notation of Magnus and Neudecker (1999), we calculate

$$\begin{aligned}
\partial f^p_{X,\epsilon}(Z) = &-\frac{p}{2}\operatorname{tr}\left(X^*_{\text{vec}}\widetilde{W}\partial\widetilde{W}^{-1}\widetilde{W}X_{\text{vec}}\right) + \frac{p\epsilon^2}{4}\left(\operatorname{tr}\left(Z(Z^*Z)^{-\frac{1}{2}}\partial Z^*\right) + \operatorname{tr}((Z^*Z)^{-\frac{1}{2}}Z^*\partial Z)\right) \\
&-\frac{p}{4}\left(\operatorname{tr}\left(Z(Z^*Z)^{\frac{4-p}{2(p-2)}}\partial Z^*\right) + \operatorname{tr}((Z^*Z)^{\frac{4-p}{2(p-2)}}Z^*\partial Z)\right)
\end{aligned}$$

where

$$\begin{aligned}
\partial\widetilde{W}^{-1} = \frac{1}{2}\partial\left[(ZZ^*)^{-\frac{1}{2}} \oplus (Z^*Z)^{-\frac{1}{2}}\right] = &-\frac{1}{4}\left[\left((Z^*Z)^{-\frac{3}{2}}Z^*\partial Z + \partial Z^* Z(Z^*Z)^{-\frac{3}{2}}\right) \otimes \mathbf{I}_{d_1}\right] \\
&-\frac{1}{4}\left[\mathbf{I}_{d_2} \otimes \left(\partial Z(ZZ^*)^{-\frac{3}{2}}Z^* + (ZZ^*)^{-\frac{3}{2}}Z\partial Z^*\right)\right].
\end{aligned} \tag{58}$$

We can reformulate the first term as follows using the cyclicity of the trace,

$$\begin{aligned}
-\frac{p}{2}\operatorname{tr}\left(X^*_{\text{vec}}\widetilde{W}\partial\widetilde{W}^{-1}\widetilde{W}X_{\text{vec}}\right) = \frac{p}{8}\Big[&\operatorname{tr}\left((\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(Z^*Z)^{-\frac{3}{2}}Z^*\partial Z\right) \\
&+ \operatorname{tr}\left(Z(Z^*Z)^{-\frac{3}{2}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}\partial Z^*\right) \\
&+ \operatorname{tr}\left(Z^*(ZZ^*)^{-\frac{3}{2}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}\partial Z\right) \\
&+ \operatorname{tr}\left((\widetilde{W}X_{\text{vec}})_{\text{mat}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(ZZ^*)^{-\frac{3}{2}}Z\partial Z^*\right)\Big].
\end{aligned}$$

To determine the critical points of $f^p_{X,\epsilon}(Z)$, we summarize the calculations above, rearrange the terms and equate the derivative with zero, such that

$$\begin{aligned}
\partial f^p_{X,\epsilon}(Z) = &\frac{p}{8}\operatorname{tr}\Big(\Big[(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(Z^*Z)^{-\frac{3}{2}}Z^* + Z^*(ZZ^*)^{-\frac{3}{2}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}} \\
&\quad +2\epsilon^2(Z^*Z)^{-\frac{1}{2}}Z^* - 2(Z^*Z)^{\frac{4-p}{2(p-2)}}Z^*\Big]\partial Z\Big) \\
&+\frac{p}{8}\operatorname{tr}\Big(\Big[Z(Z^*Z)^{-\frac{3}{2}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(\widetilde{W}X_{\text{vec}})_{\text{mat}} + (\widetilde{W}X_{\text{vec}})_{\text{mat}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(ZZ^*)^{-\frac{3}{2}}Z \\
&\quad +2\epsilon^2 Z(Z^*Z)^{-\frac{1}{2}} - 2Z(Z^*Z)^{\frac{4-p}{2(p-2)}}\Big]\partial Z^*\Big) \\
=: &\frac{p}{8}\operatorname{tr}(A\partial Z) + \frac{p}{8}\operatorname{tr}(A^*\partial Z^*) = \frac{p}{8}\operatorname{tr}((A \oplus A)\partial Z) = 0,
\end{aligned}$$

where

$$\begin{aligned}
A = \Big[&(\widetilde{W}X_{\text{vec}})^*_{\text{mat}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(Z^*Z)^{-\frac{3}{2}}Z^* + Z^*(ZZ^*)^{-\frac{3}{2}}(\widetilde{W}X_{\text{vec}})_{\text{mat}}(\widetilde{W}X_{\text{vec}})^*_{\text{mat}} \\
&+2\epsilon^2(Z^*Z)^{-\frac{1}{2}}Z^* - 2(Z^*Z)^{\frac{4-p}{2(p-2)}}Z^*\Big].
\end{aligned} \tag{59}$$

and hence an easy calculation as in (Duchi) gives

$$\frac{\partial f^p_{X,\epsilon}(Z)}{\partial Z} = \frac{\frac{p}{8}\operatorname{tr}((A \oplus A)\partial Z)}{\partial Z} = \frac{p}{8}(A \oplus A) = 0.$$

Now we have to find Z such that $A \oplus A = 0$. This implies that all eigenvalues of $A \oplus A = A \otimes \mathbf{I}_d + \mathbf{I}_d \otimes A$ are equal to zero. The eigenvalues of the Kronecker sum of two

matrices A_1 and A_2 with eigenvalues λ_s and μ_t with $s,t \in [d]$ are the sum of the eigenvalues $\lambda_s + \mu_t$. As in our case $A = A_1 = A_2$ this means that all eigenvalues of A itself have to be zero. This is only possible if A is the zero matrix.

Let $Z = U\Sigma V^* \in M_{d\times d}$ with $U, V \in \mathcal{U}_d$ and $\Sigma \in M_{d\times d}$, where $\Sigma = \operatorname{diag}(\sigma)$ is a diagonal matrix with *ascending* entries. We define the matrix $H = H_{i,j} = \frac{2}{\sigma_i^{-1}+\sigma_j^{-1}}$ for $i = 1,\dots,d, j = 1,\dots,d$ corresponding to the result of reshaping the diagonal of $2(\Sigma \oplus \Sigma)$ into a $d \times d$-matrix. Using (57), we can express $(\widetilde{W}X_{\mathrm{vec}})_{\mathrm{mat}} = U\big(H \circ (U^*XV)\big)V^*$ and denote $B := H \circ (U^*XV)$.

Plugging the decomposition $Z = U\Sigma V^*$ into (59), we can therefore calculate

$$\begin{aligned}
\mathrm{A} = 0 \Leftrightarrow\ & (UBV^*)^*(UBV^*)(V\Sigma^2V^*)^{-3/2}(U\Sigma V^*)^* + (U\Sigma V^*)^*(U\Sigma^2U^*)^*)^{-3/2}(UBV^*)(UBV^*)^* \\
& + 2\epsilon^2(V\Sigma^2V^*)^{-1/2}(U\Sigma V^*)^* - 2(V\Sigma^2V^*)^{\frac{4-p}{2(p-2)}}(U\Sigma V^*)^* = 0 \\
\Leftrightarrow\ & VB^*B\Sigma^{-2}U^* + V\Sigma^{-2}BB^*U^* + 2\epsilon^2V\mathbf{I}_dU^* - 2V\Sigma^{\frac{2}{p-2}}U^* = 0 \\
\Leftrightarrow\ & B^*B\Sigma^{-2} + \Sigma^{-2}BB^* + 2\epsilon^2\mathbf{I}_d - 2\Sigma^{\frac{2}{p-2}} = 0.
\end{aligned} \tag{60}$$

We now note that $2\epsilon^2\mathbf{I}_d - 2\Sigma^{\frac{2}{p-2}}$ is diagonal and therefore, $B^*B\Sigma^{-2} + \Sigma^{-2}BB^*$ is diagonal as well. Moreover, observe that $B^*B + \Sigma^{-2}BB^*\Sigma^2$ is again a diagonal matrix and has a symmetric first summand B^*B. As the sum or difference of symmetric matrices is again symmetric also the second summand $\Sigma^{-2}BB^*\Sigma^2$ has to be symmetric, i.e., $\Sigma^{-2}BB^*\Sigma^2 = (\Sigma^{-2}BB^*\Sigma^2)^* = \Sigma^2BB^*\Sigma^{-2}$. We conclude that it has to hold that $BB^*\Sigma^4 = \Sigma^4BB^*$ and hence Σ^4 and BB^* commute.

This is only possible if either Σ is a multiple of the identity or if BB^* is diagonal. Assuming the first case, (60) would imply that also BB^* and B^*B have to be a multiple of the identity. Therefore, this first case, where Σ is a multiple of the identity is a special case of the second possible scenario, where BB^* is diagonal. Hence, it suffices to further consider the more general second case. (Considerations for B^*B can be carried out analogously.)

Diagonality of BB^* only occurs if B is either orthonormal or diagonal. Assuming orthonormality would lead to contradictions with the equations in (60). Hence $B = H \circ (U^*XV)$ can only be diagonal.

Let now be $X = \bar{U}\bar{S}\bar{V}^*$ the singular value decomposition of X. As H has no zero entries due to the full rank of W, this implies the diagonality of $U^*\bar{U}\bar{S}\bar{V}^*V$. Consequently, U and V can only be chosen such that $P = [U^*\bar{U}]_{d\times d}$ and $P^* = [\bar{V}^*V]_{d\times d}$ for a permutation matrix $P \in \mathcal{U}_d$. The reshuffled indexing corresponding to P is denoted by $p(i) \in [d]$ for $i \in [d]$. Bearing in mind that $H_{ii} = \sigma_i$ for $i \in [d]$, we obtain

$$\begin{aligned}
& (H \circ (P\bar{S}P^*))^*(H \circ (P\bar{S}P^*))\Sigma^{-2} + \Sigma^{-2}(H \circ (P\bar{S}P^*))(H \circ (P\bar{S}P^*))^* + 2\epsilon^2\mathbf{I}_d - 2\Sigma^{\frac{2}{p-2}} = 0 \\
\Leftrightarrow\ & 2\bar{s}_{p(i)}^2 + 2\epsilon^2 = 2\sigma_i^{\frac{2}{p-2}} \text{ for all } i \in [d] \\
\Leftrightarrow\ & \sigma_i = (\bar{s}_{p(i)}^2 + \epsilon^2)^{\frac{p-2}{2}} \text{ for all } i \in [d].
\end{aligned}$$

As the diagonal of Σ was assumed to have ascending entries and the diagonal of $\bar{S}$ has descending entries, the permutation matrix P has to be equal to the identity matrix. From $P = \mathbf{I}_d$, it follows that $U = \bar{U}$ and $V = \bar{V}$ and hence $\Sigma = (\bar{S}^2 + \epsilon^2\mathbf{I}_d)^{\frac{p-2}{2}}$.

We summarize our calculations by stating that

$$Z_{\mathrm{opt}} = \bar{U}\Sigma\bar{V}^* = \bar{U}(\bar{S}^2 + \epsilon^2\mathbf{I}_d)^{\frac{p-2}{2}}\bar{V}^*$$

is the only critical point of $f^p_{X,\epsilon}$ on the domain Ω.

The results extend for the case $d_1 \neq d_2$, where the definition of $\widetilde{W}(Z)$ is adapted by introducing the Moore-Penrose pseudo inverse of $(ZZ^*)^{1/2}$

$$\widetilde{W}(Z) = 2\widetilde{W}_1\big(\widetilde{W}_1 + \widetilde{W}_2\big)^{-1}\widetilde{W}_2 = 2(\widetilde{W}_1^{+} + \widetilde{W}_2^{-1})^{-1}.$$

The corresponding derivative rule as pointed out in Theorem 5 in Chapter 8.4 of Magnus and Neudecker (1999) can be used for the calculation in (58).

B.3 Proof of Lemma 16

The equality of the optimization problems (32) can easily be seen by the fact that only the first summand of $\mathcal{J}_p(X,\epsilon,Z)$ depends on X. Now, it is important to show first that $\widetilde{W}(Z) = 2([(Z^*Z)^{\frac{1}{2}}]^+ \oplus [(ZZ^*)^{\frac{1}{2}}]^+)^{-1}$ is positive definite as minimizing $\mathcal{J}_p(X,\epsilon,Z)$ then reduces to minimizing a quadratic form. Let $Z = \sum_{i=1}^d \sigma_i u_i v_i^*$, where u_i, v_i for $i \in [d]$ are the left and right singular vector respectively and σ_i for $i \in [d]$ are the singular values of Z. Since $Z^*Z = \sum_{i=1}^d \sigma_i^2 v_i v_i^* \geq 0$, also the generalized inverse root fulfills $[(ZZ^*)^{\frac{1}{2}}]^+ \geq 0$ and for $ZZ^* = \sum_{i=1}^d \sigma_i^2 u_i u_i^* \geq 0$, it follows that $[(ZZ^*)^{\frac{1}{2}}]^+ \geq 0$. We stress that at least one of the matrices $(ZZ^*)^{\frac{1}{2}}$ and $(Z^*Z)^{\frac{1}{2}}$ is positive definite and hence also $\widetilde{W}(Z) \succ 0$. With the fact that $\widetilde{W}(Z) \succ 0$, the statement can be proven analogously to the results in (Fornasier et al., 2011, Lemma 5.1).

B.4 Proof of Lemma 17

(a) With the minimization property that defines $X^{(n+1)}$ in (29), the inequality $\epsilon^{(n+1)} \leqslant \epsilon^{(n)}$, and the minimization property that defines $Z^{(n+1)}$ in (28) and Lemma 14, the monotonicity follows from

$$\begin{aligned}
\mathcal{J}_p(X^{(n)},\epsilon^{(n)},Z^{(n)}) &\geqslant \mathcal{J}_p(X^{(n+1)},\epsilon^{(n)},Z^{(n)}) \geqslant \mathcal{J}_p(X^{(n+1)},\epsilon^{(n+1)},Z^{(n)}) \\
&\geqslant \mathcal{J}_p(X^{(n+1)},\epsilon^{(n+1)},Z^{(n+1)}).
\end{aligned}$$

(b) Using Theorem 14 and the monotonicity property of (a) for all $n \in \mathbb{N}$, we see that

$$\|X^{(n)}\|_{S_p}^p \leqslant g^p_{\epsilon^{(n)}}(X^{(n)}) = \mathcal{J}_p(X^{(n)},\epsilon^{(n)},Z^{(n)}) \leqslant \mathcal{J}_p(X^{(1)},\epsilon^{(0)},Z^{(0)}).$$

(c) The proof follows analogously to (Fornasier et al., 2011, Proposition 6.1) where only the technical calculation to bound $\sigma_1^p((\widetilde{W}^{(n)})^{-1})$ requires to take into account that the spectrum of a Kronecker sum $A \oplus B$ consists of the pairwise sum of the spectra of A and B (Bernstein, 2009, Proposition 7.2.3).

B.5 Proof of Lemma 18

The first statement $\widetilde{W}(X^{(n)},\epsilon^{(n)}) = \widetilde{W}^{(n)}$ is clear from the definition of $\widetilde{W}(X,\epsilon)$ and (10). To show the necessity of (35), let $X \in M_{d_1\times d_2}$ be a critical point of (34). Without loss

of generality, let us assume that $d_1 \leqslant d_2$. In this case, a short calculation shows that $g_\epsilon^p(X) = \operatorname{tr}\left[(XX^* + \epsilon^2 \mathbf{I}_{d_1})^{p/2}\right]$. It follows from the matrix derivative rules of Magnus and Neudecker (1999, (7),(15),(18),(20) of Chapter 8.2) that

$$\nabla g_\epsilon^p(X) = p(XX^* + \epsilon^2 \mathbf{I}_{d_1})^{\frac{p-2}{2}} X = p \sum_{i=1}^{d} (\sigma_i^2 + \epsilon^2)^{\frac{p-2}{2}} \sigma_i u_i v_i^*,$$

using the singular value decomposition $X = \sum_{i=1}^{d} \sigma_i u_i v_i^*$ in the last equality. Using the Kronecker sum inversion formula (56), we see that $\nabla g_\epsilon^p(X) = p\left[\widetilde{W}(X,\epsilon) X_{\mathrm{vec}}\right]_{\mathrm{mat}}$. The proof can be continued analogously to (Daubechies et al., 2010, Lemma 5.2).

Appendix C. Proof of Theorem 9

For statement (i) of the convergence result of Algorithm 1, we use the following *reverse triangle inequalities* implied by the strong Schatten-p NSP: Let $X, X' \in M_{d_1 \times d_2}$ such that $\Phi(X - X') = 0$. Then

$$\|X' - X\|_F^p \leqslant \frac{2^p \gamma_r^{1-p/2}}{r^{1-p/2}} \frac{1}{1-\gamma_r} \left(\|X'\|_{S_p}^p - \|X\|_{S_p}^p + 2\beta_r(X)_{S_p} \right), \tag{61}$$

where $\beta_r(X)_{S_p}$ is defined in (22). This inequality can be proven using an adaptation of the proof of the corresponding result for ℓ_p-minimization in (Gao et al., 2015, Theorem 13) and the generalization of Mirksy's singular value inequality to concave functions (Audenaert, 2014; Foucart, 2018). Furthermore, the proof of the similar statement in (Kabanava et al., 2016, Theorem 12) can be adapted to show (61).

The further part of the proof of (i) as well as (ii) follow analogously to (Fornasier et al., 2011, Theorem 6.11) and (Daubechies et al., 2010, Theorem 5.3) using the preliminary results deduced in Section 6.

Statement (iii) is a direct consequence of Theorem 11, which is proven in Section 6.3.

References

A. Ahmed and J. Romberg. Compressive multiplexing of correlated signals. *IEEE Trans. Inf. Theory*, 61(1):479–498, 2015.

K. M. R. Audenaert. A generalisation of Mirsky's singular value inequalities. preprint, arXiv:1410.4941 [math.FA], 2014.

D. S. Bernstein. *Matrix Mathematics: Theory, Facts, and Formulas (Second Edition)*. Princeton University Press, 2009.

S. Bhojanapalli, B. Neyshabur, and N. Srebro. Global optimality of local search for low rank matrix recovery. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3873–3881, 2016.

J. D. Blanchard, J. Tanner, and K. Wei. CGIHT: conjugate gradient iterative hard thresholding for compressed sensing and matrix completion. *Inf. Inference*, 4(4):289–327, 2015.

E. J. Candès and Y. Plan. Tight oracle inequalities for low-rank matrix recovery from a minimal number of noisy random measurements. *IEEE Trans. Inf. Theory*, 57(4):2342–2359, April 2011.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Found. Comput. Math.*, 9(6):717–772, 2009.

E. J. Candès, Y. Eldar, T. Strohmer, and V. Voroninski. Phase retrieval via matrix completion. *SIAM J. Imag. Sci.*, 6(1):199–225, 2013.

E. J. Candès, T. Strohmer, and V. Voroninski. PhaseLift: Exact and Stable Signal Recovery from Magnitude Measurements via Convex Programming. *Commun. Pure Appl. Math.*, 66(8):1241–1274, 2013.

E. J. Candès, X. Li, and M. Soltanolkotabi. Phase retrieval via Wirtinger flow: Theory and algorithms. *IEEE Trans. Inf. Theory*, 61(4):1985–2007, 2015.

R. Chartrand. Exact reconstructions of sparse signals via nonconvex minimization. *IEEE Signal Process. Lett.*, 14:707–710, 2007.

J. A. Chavez-Dominguez and D. Kutzarova. Stability of low-rank matrix recovery and its connections to Banach space geometry. *J. Math. Anal. Appl.*, 427(1):320–335, 2015.

B. Dacorogna. *Direct Methods in the Calculus of Variations*. Springer, New York, 1989.

I. Daubechies, R. DeVore, M. Fornasier, and C.S. Güntürk. Iteratively reweighted least squares minimization for sparse recovery. *Commun. Pure Appl. Math.*, 63:1–38, 2010.

M. A. Davenport and J. Romberg. An overview of low-rank matrix recovery from incomplete observations. *IEEE J. Sel. Topics Signal Process.*, 10:608–622, 06 2016.

D. L. Donoho, M. Gavish, and A. Montanari. The phase transition of matrix recovery from Gaussian measurements matches the minimax MSE of matrix denoising. *Proc. Nat. Acad. Sci. U.S.A.*, 110(21):8405–8410, 2013.

J. Duchi. Properties of the Trace and Matrix Derivatives. Available electronically at `https://web.stanford.edu/~jduchi/projects/matrix_prop.pdf`.

Y.C. Eldar, D. Needell, and Y. Plan. Uniqueness conditions for low-rank matrix recovery. *Appl. Comput. Harmon. Anal.*, 33(2):309–314, 2012.

M. Fazel. *Matrix rank minimization with applications*. Ph.D. Thesis, Electrical Engineering Department, Stanford University, 2002.

M. Fornasier, H. Rauhut, and R. Ward. Low-rank matrix recovery via iteratively reweighted least squares minimization. *SIAM J. Optim.*, 21(4):1614–1640, 2011. code from `https://github.com/rward314/IRLSM`].

M. Fornasier, S. Peter, H. Rauhut, and S. Worm. Conjugate gradient acceleration of iteratively re-weighted least squares methods. *Comput. Optim. Appl.*, 65(1):205–259, 2016.

S. Foucart. Concave Mirsky Inequality and Low-Rank Recovery. *SIAM J. Matrix Anal. Appl.*, 39(1):99–103, 2018.

S. Foucart and H. Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Birkhäuser/Springer, New York, 2013.

Y. Gao, J. Peng, S. Yue, and Y. Zhao. On the null space property of ℓ_q-minimization for $0 < q \leqslant 1$ in compressed sensing. *J. Funct. Spaces*, 2015:4203–4215, 2015.

R. Ge, J. D. Lee, and T. Ma. Matrix completion has no spurious local minimum. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2973–2981, 2016.

I. Gohberg, S. Goldberg, and N. Krupnik. *Traces and determinants of linear operators*, volume 116 of *Operator Theory: Advances and Applications*. Birkhäuser, Basel, 2000.

D. Goldberg, D. Nichols, B. M. Oki, and D. Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35(12):61–70, 1992.

M. Grant and S. Boyd. CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`, March 2014.

D. Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Trans. Inf. Theory*, 57(3):1548–1566, 2011.

D. Gross, Y.-K. Liu, S. T. Flammia, S. Becker, and J. Eisert. Quantum state tomography via compressed sensing. *Phys. Rev. Lett.*, 105:150401, 2010.

D. Gross, F. Krahmer, and R. Kueng. A partial derandomization of phaselift using spherical designs. *J. Fourier Anal. Appl.*, 21(2):229–266, 2015.

J. P. Haldar and D. Hernando. Rank-constrained solutions to linear matrix equations using powerfactorization. *IEEE Signal Process. Lett.*, 16(7):584–587, July 2009. [using `AltMin` (Alternating Minimization) algorithm].

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Rev.*, 53 (2):217–288, 2011.

P. Jain, Raghu M., and I. S. Dhillon. Guaranteed rank minimization via singular value projection. In *Advances in Neural Information Processing Systems (NIPS)*, pages 937–945, 2010.

P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Proc. ACM Symp. Theory Comput. (STOC)*, pages 665–674, Palo Alto, CA, USA, June 2013.

A. Jameson. Solution of the equation ax + xb = c by inversion of an m x m or n x n matrix. *SIAM J. Appl. Math.*, 16(5):1020–1023, 1968.

M. Kabanava, R. Kueng, H. Rauhut, and U. Terstiege. Stable low-rank matrix recovery via null space properties. *Inf. Inference*, 5(4):405–441, 2016.

F. J. Király, L. Theran, and R. Tomioka. The Algebraic Combinatorial Approach for Low-Rank Matrix Completion. *J. Mach. Learn. Res.*, 16:1391–1436, 2015.

A. Kyrillidis and V. Cevher. Matrix recipes for hard thresholding methods. *J. Math. Imaging Vision*, 48(2):235–265, 2014. [using `Matrix ALPS II` ('Matrix ALgrebraic PursuitS II') algorithm, code from `http://akyrillidis.github.io/projects/`].

C. Kümmerle and J. Sigl. Harmonic Mean Iteratively Reweighted Least Squares for low-rank matrix recovery. In *12th International Conference on Sampling Theory and Applications (SampTA)*, pages 489–493, 2017.

Z. Liu and L. Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM J. Matrix Anal. Appl.*, 31(3):1235–1256, 2010.

Z. Liu, A. Hansson, and L. Vandenberghe. Nuclear norm system identification with missing inputs and outputs. *Systems Control Lett.*, 62(8):605–612, 2013.

J.R. Magnus and H. Neudecker. *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley Series in Probability and Statistics. Wiley, 1999.

B. Mishra, G. Meyer, F. Bach, and R. Sepulchre. Low-rank optimization with trace norm penalty. *SIAM J. Optim.*, 23(4):2124–2149, 2013.

K. Mohan and M. Fazel. Iterative reweighted algorithms for matrix rank minimization. *J. Mach. Learn. Res.*, 13(1):3441–3473, 2012. [using `IRLS-MF` ('IRLS-p') algorithm, code from `https://faculty.washington.edu/mfazel/`].

S. Oymak, K. Mohan, M. Fazel, and B. Hassibi. A simplified approach to recovery conditions for low rank matrices. In *Proceedings of the IEEE International Symposium on Information Theory (ISIT)*, pages 2318–2322, 2011.

D. Park, A. Kyrillidis, C. Caramanis, and S. Sanghavi. Finding Low-rank Solutions to Matrix Problems, Efficiently and Provably. preprint, arXiv:1606.03168 [math.OC], [using `BFGD` ('Bi-Factored Gradient Descent') algorithm, code from `http://akyrillidis.github.io/projects/`], 2016.

D.L. Pimentel-Alarcón, N. Boston, and R. D. Nowak. A Characterization of Deterministic Sampling Patterns for Low-Rank Matrix Completion. preprint, arXiv:1503.02596v3 [stat.ML], October 2016.

B. Recht, M. Fazel, and P. A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Rev.*, 52(3):471–501, 2010.

B. Recht, W. Xu, and B. Hassibi. Null space conditions and thresholds for rank minimization. *Math. Program.*, 127(1):175–202, 2011.

É. Schost and P.-J. Spaenlehauer. A quadratically convergent algorithm for structured low-rank approximation. *Found. Comput. Math.*, 16(2):457–492, 2016.

N. Srebro, J. Rennie, and T. S. Jaakkola. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1329–1336, 2005.

M. Stewart. Perturbation of the SVD in the presence of small singular values. *Linear Algebra Appl.*, 419(1):53–77, 2006.

R. Sun and Z. Q. Luo. Guaranteed matrix completion via non-convex factorization. *IEEE Trans. Inf. Theory*, 62(11):6535–6579, 2016.

J. Tanner and K. Wei. Normalized Iterative Hard Thresholding for Matrix Completion. *SIAM J. Sci. Comput.*, 35(5):S104–S125, 2013.

J. Tanner and K. Wei. Low rank matrix completion by alternating steepest descent methods. *Appl. Comput. Harmon. Anal.*, 40(2):417–429, 2016. [using `ASD` ('Alternating Steepest Descent') algorithm, code from `https://www.math.ucdavis.edu/~kewei/publications.html`].

S. Tu, R. Boczar, M. Simchowitz, M. Soltanolkotabi, and B. Recht. Low-rank Solutions of Linear Matrix Equations via Procrustes Flow. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48, pages 964–973, 2016.

B. Vandereycken. Low-rank matrix completion by Riemannian optimization. *SIAM J. Optim.*, 23(2):1214–1236, 2013. [using `Riemann_Opt` ('Riemannian Optimization') algorithm, code from `http://www.unige.ch/math/vandereycken/matrix_completion.html`].

P.-Å. Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 12(1):99–111, 1972.

K. Wei, J.-F. Cai, T. F. Chan, and S. Leung. Guarantees of Riemannian Optimization for Low Rank Matrix Recovery. *SIAM J. Matrix Anal. Appl.*, 37(3):1198–1222, 2016.

Z. Wen, W. Yin, and Y. Zhang. Solving a low-rank factorization model for matrix completion by a nonlinear successive over-relaxation algorithm. *Math. Program. Comput.*, 4(4):333–361, 2012.

Q. Zheng and J. Lafferty. A convergent gradient descent algorithm for rank minimization and semidefinite programming from random linear measurements. In *Advances in Neural Information Processing Systems (NIPS)*, pages 109–117, 2015.

Journal of Machine Learning Research 19 (2018) 1-49 Submitted 5/17; Published 9/18

On Generalized Bellman Equations and Temporal-Difference Learning

Huizhen Yu JANEY.HZYU@GMAIL.COM
Reinforcement Learning and Artificial Intelligence Group
Department of Computing Science, University of Alberta
Edmonton, AB, T6G 2E8, Canada

A. Rupam Mahmood RUPAM@KINDRED.AI
Kindred Inc.
243 College St
Toronto, ON M5T 1R5, Canada

Richard S. Sutton RSUTTON@UALBERTA.CA
Reinforcement Learning and Artificial Intelligence Group
Department of Computing Science, University of Alberta
Edmonton, AB, T6G 2E8, Canada

Editor: Csaba Szepesvári

Abstract

We consider off-policy temporal-difference (TD) learning in discounted Markov decision processes, where the goal is to evaluate a policy in a model-free way by using observations of a state process generated without executing the policy. To curb the high variance issue in off-policy TD learning, we propose a new scheme of setting the λ-parameters of TD, based on generalized Bellman equations. Our scheme is to set λ according to the eligibility trace iterates calculated in TD, thereby easily keeping these traces in a desired bounded range. Compared with prior work, this scheme is more direct and flexible, and allows much larger λ values for off-policy TD learning with bounded traces. As to its soundness, using Markov chain theory, we prove the ergodicity of the joint state-trace process under nonrestrictive conditions, and we show that associated with our scheme is a generalized Bellman equation (for the policy to be evaluated) that depends on both the evolution of λ and the unique invariant probability measure of the state-trace process. These results not only lead immediately to a characterization of the convergence behavior of least-squares based implementation of our scheme, but also prepare the ground for further analysis of gradient-based implementations.

Keywords: Markov decision process, approximate policy evaluation, generalized Bellman equation, reinforcement learning, temporal-difference method, Markov chain, randomized stopping time

1. Introduction

We consider discounted Markov decision processes (MDPs) and off-policy temporal-difference (TD) learning methods for approximate policy evaluation with linear function approximation. The goal is to evaluate a policy in a model-free way by using observations of a state process generated without executing the policy. Off-policy learning is an important part of the reinforcement learning methodology (Sutton and Barto, 1998) and has been studied in the areas of operations research and machine learning. (For an incomplete list of references, see e.g., Glynn and Iglehart, 1989; Precup et al., 2000, 2001; Randhawa and Juneja, 2004; Sutton et al., 2008, 2009; Maei, 2011; Yu, 2012; Dann et al., 2014; Geist and Scherrer, 2014; Mahadevan et al., 2014; Mahmood et al., 2014; Liu et al., 2015; Sutton et al., 2016; Dai et al., 2018.) Available TD algorithms, however, tend to have very high variances due to the use of importance sampling, an issue that limits their applicability in practice. The purpose of this paper is to introduce a new TD learning scheme that can help address this problem.

Our work is motivated by the recently proposed Retrace algorithm (Munos et al., 2016) and ABQ algorithm (Mahmood et al., 2017), and by the Tree-Backup algorithm (Precup et al., 2000) that existed earlier. These algorithms, as explained by Mahmood et al. (2017), all try to use the λ-parameters of TD to curb the high variance issue in off-policy learning. In particular, they all choose the values of λ according to the current state or state-action pair in such a way that guarantees the boundedness of the eligibility traces in TD learning, which can help reduce significantly the variance of the TD iterates. A limitation of these algorithms, however, is that they tend to be over-conservative and restrict λ to small values, whereas small λ can result in large approximation bias in TD solutions.

In this paper, we propose a new scheme of setting the λ-parameters of TD, based on generalized Bellman equations. Our scheme is to set λ according to the eligibility trace iterates calculated in TD, thereby easily keeping those traces in a desired bounded range. Compared with the schemes used in the previous work just mentioned, this is a direct way to bound the traces in TD, and it is also more flexible and allows much larger λ values for off-policy learning.

Regarding generalized Bellman equations, in our context, they will correspond to a family of dynamic programming equations for the policy to be evaluated. These equations all have the true value function as their unique solution, and their associated operators have contraction properties, like the standard Bellman operator. We will refer to the associated operators as generalized Bellman operators or Bellman operators for short. Some authors have considered, at least conceptually, the use of an even broader class of equations for policy evaluation. For example, Ueno et al. (2011) have considered treating the policy evaluation problem as a parameter estimation problem in the statistical framework of estimating equations, and in their framework, any equation that has the true value function as the unique solution can be used to estimate the value function. The family of generalized Bellman equations we consider has a more specific structure. They generalize multistep Bellman equations, and they are associated with randomized stopping times and arise from the strong Markov property (see Section 3.1 for details).

Generalized Bellman equations and operators are powerful tools. In classic MDP theory they have been used in some intricate optimality analyses (e.g., Schäl and Sudderth, 1987). Their computational use, however, seems to emerge primarily in the field of reinforcement learning. Through the λ-parameters and eligibility traces, TD learning is naturally connected with, not a single Bellman operator, but a family of Bellman operators, with different choices of λ or different rules of calculating the eligibility trace iterates corresponding

to different Bellman operators. Early efforts that use this aspect to broaden the scope of TD algorithms and to analyze such algorithms include Sutton's work (1995) on learning at multiple timescales and Tsitsiklis' work on generalized TD algorithms in the tabular case (see the book by Bertsekas and Tsitsiklis, 1996, Chap. 5.3). In the context of off-policy learning, there are more recent approaches that try to utilize this connection of TD with generalized Bellman operators to make TD learning more efficient (Precup et al., 2000; Yu and Bertsekas, 2012; Munos et al., 2016; Mahmood et al., 2017). This is also our aim, in proposing the new scheme of setting the λ-parameters.

Our analyses of the new TD learning scheme will focus on its theoretical side. Using Markov chain theory, we prove the ergodicity of the joint state and trace process under nonrestrictive conditions (see Theorem 2.1), and we show that associated with our scheme is a generalized Bellman equation (for the policy to be evaluated) that depends on both the evolution of λ and the unique invariant probability measure of the state-trace process (see Theorem 3.2 and Corollary 3.1). These results not only lead immediately to a characterization of the convergence behavior of least-squares based implementation of our scheme (see Corollary 2.1 and Remark 3.2), but also prepare the ground for further analysis of gradient-based implementations. (The latter analysis has been carried out recently by Yu (2017); see Remark 3.3.)

In addition to the theoretical study, we also present the results from a preliminary numerical study that compares several ways of setting λ for the least-squares based off-policy algorithm. The results demonstrate the advantages of the proposed new scheme with its greater flexibility.

We remark that although we shall focus exclusively on policy evaluation in this paper, approximate policy evaluation methods are highly pertinent to finding near-optimal policies in MDPs. They can be applied in approximate policy iteration, in policy-gradient algorithms for gradient estimation or in direct policy search (see e.g., Konda, 2002; Mannor et al., 2003). In addition to solving MDPs, they can also be used in artificial intelligence and robotics applications as a means to generate experience-based world models (see e.g., Sutton, 2009). It is, however, beyond the scope of this paper to discuss these applications of our results.

The rest of the paper is organized as follows. In Section 2, after a brief background introduction, we present our scheme of TD learning with bounded traces, and we establish the ergodicity of the joint state-trace process. In Section 3, we first discuss generalized Bellman operators associated with randomized stopping times, and we then derive the generalized Bellman equation associated with our scheme. In Section 4, we present the experimental results on the least-squares based implementation of our scheme. Appendices A-B include a proof for generalized Bellman operators and materials about approximation properties of TD solutions that are too long to include in the main text.

2. Off-Policy TD Learning with Bounded Traces

We describe the off-policy policy evaluation problem and the algorithmic form of TD learning in Section 2.1. We then present our scheme of history-dependent λ in Section 2.2, and analyze the properties of the resulting eligibility trace iterates and the convergence of the corresponding least-squares based algorithm in Section 2.3.

2.1 Preliminaries

The off-policy learning problem we consider in this paper concerns two Markov chains on a finite state space $\mathcal{S} = \{1, \ldots, N\}$. The first chain has transition matrix P, and the second P^o. Whatever physical mechanisms that induce the two chains shall be denoted by π and π^o, and referred to as the target policy and behavior policy, respectively. The second Markov chain we can observe; however, it is the system performance of the first Markov chain that we want to evaluate.

Specifically, we consider a one-stage reward function $r_\pi : \mathcal{S} \to \Re$ and an associated discounted total reward criterion with state-dependent discount factors $\gamma(s) \in [0, 1], s \in \mathcal{S}$. Let Γ denote the $N \times N$ diagonal matrix with diagonal entries $\gamma(s)$. We assume that P and P^o satisfy the following conditions:

Condition 2.1 (Conditions on the target and behavior policies)

(i) *P is such that the inverse $(I - P\Gamma)^{-1}$ exists, and*

(ii) *P^o is such that for all $s, s' \in \mathcal{S}$, $P^o_{ss'} = 0 \Rightarrow P_{ss'} = 0$, and moreover, P^o is irreducible.*

The performance of π is defined as the expected discounted total rewards for each initial state $s \in \mathcal{S}$:

$$v_\pi(s) := \mathbb{E}^\pi_s \left[r_\pi(S_0) + \textstyle\sum_{t=1}^\infty \gamma(S_1)\gamma(S_2) \cdots \gamma(S_t) \cdot r_\pi(S_t) \right], \tag{2.1}$$

where the notation $\mathbb{E}^\pi_s$ means that the expectation is taken with respect to (w.r.t.) the Markov chain $\{S_t\}$ starting from $S_0 = s$ and induced by π (i.e., with transition matrix P). The function v_π is well-defined under Condition 2.1(i). It is called the *value function* of π, and by standard MDP theory (see e.g., Puterman, 1994), we can write it in matrix/vector notation as

$$v_\pi = r_\pi + P\Gamma\, v_\pi, \qquad \text{i.e.,} \quad v_\pi = (I - P\Gamma)^{-1} r_\pi.$$

The first equation above is known as the Bellman equation (or dynamic programming equation) for a stationary policy (cf. Footnote 2).

We compute an approximation of v_π of the form $v(s) = \phi(s)^\top \theta$, $s \in \mathcal{S}$, where $\theta \in \Re^n$ is a parameter vector and $\phi(s)$ is an n-dimensional feature representation for each state s (here $\phi(s), \theta$ are column vectors and the symbol $^\top$ stands for transpose). Data available for this computation are:

(i) a realization of the Markov chain $\{S_t\}$ with transition matrix P^o generated by π^o, and

(ii) rewards $R_t = r(S_t, S_{t+1})$ associated with state transitions, where the function r relates to $r_\pi(s)$ as $r_\pi(s) = \mathbb{E}^\pi_s[r(s, S_1)]$ for all $s \in \mathcal{S}$.[1]

To find a suitable parameter θ for the approximation $\phi(s)^\top\theta$, we use the off-policy TD learning scheme. Define $\rho(s, s') = P_{ss'}/P^o_{ss'}$ (the importance sampling ratio),[2] and write

$$\rho_t = \rho(S_t, S_{t+1}), \qquad \gamma_t = \gamma(S_t).$$

1. One can add to R_t a zero-mean finite-variance noise term. This makes little difference to our analyses, so we have left it out for notational simplicity.
2. Our problem formulation entails both value function and state-action value function estimation for a stationary policy in the standard MDP context. In these applications, it is the state-action space of the MDP that corresponds to the state space $\mathcal{S}$ here. In particular, for value function estimation, S_t here

Given an initial $e_0 \in \Re^n$, for each $t \geq 1$, the eligibility trace vector $e_t \in \Re^n$ and the scalar temporal-difference term $\delta_t(v)$ for any approximate value function $v : \mathcal{S} \to \Re$ are calculated according to

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + \phi(S_t), \tag{2.2}$$

$$\delta_t(v) = \rho_t \big(R_t + \gamma_{t+1} v(S_{t+1}) - v(S_t)\big). \tag{2.3}$$

Here $\lambda_t \in [0, 1], t \geq 1$, are important parameters in TD learning, the choice of which we shall elaborate on shortly.

There exist a number of TD algorithms that use e_t and δ_t to generate a sequence of parameters θ_t for approximate value functions. One such algorithm is LSTD (Boyan, 1999; Yu, 2012), which obtains θ_t by solving the linear equation for $\theta \in \Re^n$,

$$\tfrac{1}{t} \textstyle\sum_{k=0}^{t-1} e_k \delta_k(v) = 0, \quad v = \Phi\theta \tag{2.4}$$

(if it admits a solution), where Φ is a matrix with row vectors $\phi(s)^\top, s \in \mathcal{S}$. LSTD updates the equation (2.4) iteratively by incorporating one by one the observation of (S_t, S_{t+1}, R_t) at each state transition. We will discuss primarily this algorithm in the paper, as its behavior can be characterized directly using our subsequent analyses of the joint state-trace process.

As mentioned earlier, our analyses will also provide bases for analyzing other gradient-based TD algorithms (e.g., Sutton et al., 2008, 2009; Maei, 2011; Mahadevan et al., 2014) by using stochastic approximation theory (Kushner and Yin, 2003; Borkar, 2008; Karmakar and Bhatnagar, 2018). Because of the complexity of this subject, however, we will not delve into it in the present paper, and we refer the reader to the recent work (Yu, 2017) for details.

2.2 Our Scheme of History-dependent λ

We now come to the choices of λ_t in the trace iterates (2.2). For TD with function approximation, one often lets λ_t be a constant or a function of S_t (Sutton, 1988; Tsitsiklis and Van Roy, 1997; Sutton and Barto, 1998). If neither the behavior policy nor the λ_t's are further constrained, $\{e_t\}$ can have unbounded variances and is also unbounded in many natural situations (see e.g., Yu, 2012, Section 3.1), and this makes off-policy TD learning challenging.[3] If we let the behavior policy to be close enough to the target policy so that $P^o \approx P$, then variance can be reduced, but it is not a satisfactory solution, for the applicability of off-policy learning would be seriously limited.

Without restricting the behavior policy, as mentioned earlier, the two recent papers (Munos et al., 2016; Mahmood et al., 2017), as well as the closely related early work by Precup et al. (2000), exploit state-dependent λ's to control variance. Their choices of λ_t are such that $\lambda_t \gamma_t \rho_{t-1} < 1$ for all t, so that the trace iterates e_t are made bounded, which can help reduce the variance of the iterates.

Motivated by this prior work, our proposal is to set λ_t according to e_{t-1} directly, so that we can keep e_t in a desired range straightforwardly and at the same time, allow a much larger range of values for the λ-parameters. As a simple example, if we use λ_t to scale the vector $\gamma_t \rho_{t-1} e_{t-1}$ to be within a ball with some given radius, then we keep e_t always bounded.

In the rest of this paper, we shall focus on analyzing the iteration (2.2) with a particular choice of λ_t of the kind just mentioned. We want to be more general than the preceding simple example. However, since the dependence on the trace e_{t-1} would make λ_t dependent on the entire past history $(S_0, \ldots, S_{t-1})$, we also want to retain certain Markovian properties that are very useful for convergence analysis. This leads us to consider λ_t *being a certain function of the previous trace and past states.* More specifically, we will let λ_t be a function of the previous trace e_{t-1} and a certain memory state that is a summary of the states observed so far. The formulation is as follows.

2.2.1 Formulation and Examples

We denote the memory state at time t by y_t. For simplicity, we assume that y_t can only take values from a finite set $\mathcal{M}$, and its evolution is Markovian: $y_t = g(y_{t-1}, S_t)$ for some given function g. The joint process $\{(S_t, y_t)\}$ is then a simple finite-state Markov chain. Each y_t is a function of the history $(S_0, \ldots, S_t)$ and y_0. We further require, besides the irreducibility of $\{S_t\}$ (cf. Condition 2.1(ii)), that

Condition 2.2 (Evolution of memory states) *Under the behavior policy π^o, the Markov chain $\{(S_t, y_t)\}$ on $\mathcal{S} \times \mathcal{M}$ has a single recurrent class.*

This recurrence condition is nonrestrictive: If the Markov chain has multiple recurrent classes, each recurrent class can be treated separately by using the same arguments we present in this paper. However, we remark that the finiteness assumption on $\mathcal{M}$ is a simplification. We choose to work with finite $\mathcal{M}$ mainly for the reason that with the traces lying in a continuous space, to study the joint state and trace process, we need to resort to properties of Markov chains on infinite spaces. With an infinite $\mathcal{M}$, we would need to introduce more technical conditions that are not essential to our analysis and can obscure our main arguments.

We thus let y_t and λ_t evolve as

$$y_t = g(y_{t-1}, S_t), \qquad \lambda_t = \lambda(y_t, e_{t-1}) \tag{2.5}$$

where $\lambda : \mathcal{M} \times \Re^n \to [0, 1]$. We require the function λ to satisfy two conditions.

Condition 2.3 (Conditions for $\lambda(\cdot)$) *For some norm $\|\cdot\|$ on $\Re^n$, the following hold for each memory state $y \in \mathcal{M}$:*

(i) *For any $e, e' \in \Re^n$, $\|\lambda(y, e)\, e - \lambda(y, e')\, e'\| \leq \|e - e'\|$.*

(ii) *For some constant C_y, $\|\gamma(s')\rho(s, s') \cdot \lambda(y, e)\, e\| \leq C_y$ for all $e \in \Re^n$ and all possible state transitions (s, s') that can lead to the memory state y.*

corresponds to the pair of previous action and current state in the MDP, whereas for state-value function estimation, S_t here corresponds to the current state-action pair in the MDP. The ratio $\rho(s, s') = P_{ss'}/P^o_{ss'}$ then comes out as the ratio of action probabilities under π and π^o, the same as what appears in most of the off-policy learning literature. For the details of these correspondences, see (Yu, 2012, Examples 2.1, 2.2). The third application is in a simulation context where P^o corresponds to a simulated system and both P^o, P are known so that the ratio $\rho(s, s')$ is available. Such simulations are useful, for example, in studying system performance under perturbations, and in speeding up the computation when assessing the impacts of events that are rare under the dynamics P.

3. However, asymptotic convergence can still be ensured for several algorithms (Yu, 2012, 2015, 2016b), thanks partly to a powerful law of large numbers for stationary processes.

In the above, the second condition is to restrict $\{e_t\}$ in a desired range (as it makes $\|e_t\| \leq \max_{y\in\mathcal{M}} C_y + \max_{s\in\mathcal{S}} \|\phi(s)\|$). The first condition is about the continuity of the function $\lambda(y,e)\,e$ in the trace variable e for each memory state y, and it plays a key role in the subsequent analysis, where we will use this condition to ensure that the traces e_t and the states (S_t, y_t) jointly form a Markov chain with appealing properties. We shall defer a further discussion on the technical roles of these conditions to the end of Section 2.3 (cf. Remark 2.2).

Let us give a few simple examples of choosing λ that satisfy Condition 2.3. We will later use these examples in our experimental study (Section 4).

Example 2.1 We consider again the simple scaling example mentioned earlier and describe it using the terminologies just introduced. In this example, we let $y_t = (S_{t-1}, S_t)$. For each $y = (s, s')$, we define the function $\lambda(y,\cdot)$ so that when multiplied with $\lambda(y,e)$, the vector $\gamma(s')\rho(s,s')\,e$ is scaled down whenever its length exceeds a given threshold $C_{ss'}$:

$$\lambda\big(y,e\big) = \begin{cases} 1 & \text{if } \ \gamma(s')\rho(s,s')\|e\|_2 \leq C_{ss'}; \\ \frac{C_{ss'}}{\gamma(s')\rho(s,s')\|e\|_2} & \text{otherwise.} \end{cases} \tag{2.6}$$

Condition 2.3(i) is satisfied because for $y = (s,s')$ with $\gamma(s')\rho(s,s') = 0$, $\lambda\big(y,e\big)e = e$, whereas for $y = (s,s')$ with $\gamma(s')\rho(s,s') \neq 0$, $\lambda\big(y,e\big)e$ is simply the Euclidean projection of e onto the ball (centered at the origin) with radius $C_{ss'}/(\gamma(s')\rho(s,s'))$ and is therefore Lipschitz continuous in e with modulus 1 w.r.t. $\|\cdot\|_2$. Corresponding to (2.6), the update rule (2.2) of e_t becomes

$$e_t = \begin{cases} \gamma_t\,\rho_{t-1}\,e_{t-1} + \phi(S_t) & \text{if } \ \gamma_t\rho_{t-1}\|e_{t-1}\|_2 \leq C_{S_{t-1}S_t}; \\ C_{S_{t-1}S_t}\cdot\frac{e_{t-1}}{\|e_{t-1}\|_2} + \phi(S_t) & \text{otherwise.} \end{cases} \tag{2.7}$$

Note that this scheme of setting λ encourages the use of large λ_t: $\lambda_t = 1$ will be chosen whenever possible. A variation of the scheme is to multiply the right-hand side (r.h.s.) of (2.6) by another factor $\beta_{ss'} \in [0,1]$, so that λ_t can be at most $\beta_{S_{t-1}S_t}$. In particular, one such variation is to simply multiply the r.h.s. of (2.6) by a constant $\beta \in (0,1)$ so that $\lambda_t \leq \beta < 1$ for all t. ■

Example 2.2 The Retrace algorithm (Munos et al., 2016) modifies the trace updates in off-policy TD learning by truncating the importance sampling ratios by 1. In particular, for the off-policy TD(λ) algorithm with a constant $\lambda = \beta \in (0,1]$, Retrace modifies the trace updates to be

$$e_t = \beta\,\gamma_t\cdot\min\{1,\rho_{t-1}\}\cdot e_{t-1} + \phi(S_t). \tag{2.8}$$

As pointed out by Mahmood et al. (2017), to retain the original interpretation of λ as a bootstrapping parameter in TD learning, we can rewrite the above update rule of Retrace equivalently as

$$e_t = \lambda_t\,\gamma_t\,\rho_{t-1}\,e_{t-1} + \phi(S_t) \qquad \text{for } \ \lambda_t = \beta\cdot\tfrac{\min\{1,\rho_{t-1}\}}{\rho_{t-1}} \ \ \text{(with } 0/0 = 0). \tag{2.9}$$

Each λ_t here is a function of (S_{t-1}, S_t) only and does not depend on e_{t-1}, so this choice of λ-parameters automatically satisfies Condition 2.3(i) with the memory states being $y_t = (S_{t-1}, S_t)$. When the discount factors $\gamma(s)$ are all strictly less than 1, $\|e_t\|$ for all t are bounded by a deterministic constant that depends on the initial e_0. Then for each initial e_0, Retrace's choice of λ coincides with a choice in our framework, since the C-parameters in Condition 2.3(ii) can be made vacuously large so that the condition is satisfied by all the traces e_t that could be encountered by Retrace. Thus in this case our framework for choosing λ effectively encompasses the particular choice used by Retrace.

One can make variations on Retrace's trace update rule. For example, instead of truncating each importance sampling ratio $\rho(s,s')$ by 1, one can truncate it by a constant $K_{ss'} \geq 1$, and then use a scaling scheme similar to Example 2.1 to bound the traces. The simplest such variation is to choose two memory-independent positive constants K and C, and replace the definition of λ_t in (2.9) by the following: with $\tilde{\lambda}_t = \frac{\min\{K,\rho_{t-1}\}}{\rho_{t-1}}$ (where we treat $0/0 = 0$),

$$\lambda_t = \begin{cases} \beta\,\tilde{\lambda}_t & \text{if } \ \tilde{\lambda}_t\gamma_t\,\rho_{t-1}\,\|e_{t-1}\|_2 \leq C; \\ \beta\,\tilde{\lambda}_t\cdot\frac{C}{\tilde{\lambda}_t\gamma_t\,\rho_{t-1}\,\|e_{t-1}\|_2} & \text{otherwise.} \end{cases} \tag{2.10}$$

Correspondingly, instead of (2.8), the update rule of e_t becomes

$$e_t = \begin{cases} \beta\,\gamma_t\cdot\min\{K,\rho_{t-1}\}\cdot e_{t-1} + \phi(S_t) & \text{if } \ \gamma_t\cdot\min\{K,\rho_{t-1}\}\cdot\|e_{t-1}\|_2 \leq C; \\ \beta\,C\cdot\frac{e_{t-1}}{\|e_{t-1}\|_2} + \phi(S_t) & \text{otherwise.} \end{cases} \tag{2.11}$$

These variations of Retrace are similar to Example 2.1 and satisfy Condition 2.3.[4] ■

2.2.2 Comparison with Previous Work

For policy evaluation, the Retrace algorithm (Munos et al., 2016) and the ABQ algorithm (Mahmood et al., 2017) are very similar (ABQ was actually developed independently of Retrace before the Munos et al. (2016) paper was published, although the ABQ paper itself was released much later). Both Retrace and ABQ include the Tree-Backup algorithm (Precup et al., 2000) as a special case. They can use additional parameters to select λ from a range of values, whereas Tree-Backup specifies λ, implicitly, in a particular way (which has the advantage of requiring no knowledge of the behavior policy) and does not have the freedom in choosing λ. Because of the relations between these algorithms, when comparing our method to them, we will compare it with Retrace only. In the experimental study given later in Section 4 on the performance of LSTD for various ways of setting λ, we will compare our scheme of choosing λ with that of Retrace for $\beta = 1$, which lets Retrace use the largest λ that it can take.

We see in Example 2.2 that the eligibility trace update rule of Retrace can be written in two equivalent forms, (2.8) and (2.9). The second form (2.9) has the advantage that the λ-parameters involved are shown explicitly. In TD learning, the λ-parameters directly affect the associated Bellman operators and can be meaningfully interpreted as stopping probabilities (see Section 3), whereas the importance sampling ratio terms in the eligibility

4. To see this, let the memory states be $y_t = (S_{t-1}, S_t)$. For each $y = (s,s')$, let $\lambda(y,e)$ be defined according to (2.10), and let C_y in Condition 2.3(ii) be $C_{ss'} = \frac{\rho(s,s')}{\min\{K,\,\rho(s,s')\}}\cdot C$ (treat $0/0 = 0$). Then note that $\|\lambda(y,e)e - \lambda(y,e')e'\|_2 \leq \beta\min\big\{\frac{K}{\rho(s,s')},\,1\big\}\cdot\|e - e'\|_2 \leq \beta\|e - e'\|_2$.

trace iterates are essentially unchanged, for they have to be there in order to correct for the discrepancy between the behavior and target policies. For this reason, we prefer (2.9) to (2.8) and prefer thinking in terms of the selection of λ-parameters to that of what occurs *apparently* to those importance sampling ratio terms in the trace updates.

As mentioned in Example 2.2, the Munos et al. (2016) paper does not make the connection between (2.8) and (2.9). Mahmood et al. (2017) recognized the role of the λ-parameters and made explicit use of it to derive the ABQ algorithm. However, in the ABQ paper, the discussion and the presentation of the algorithm still emphasize the apparent changes in those importance sampling ratio terms in the trace iterates. This is an unsatisfactory point in that paper that we hope we have clarified with our present work.

We mentioned in the introduction that Retrace, ABQ and Tree-Backup are too conservative and tend to use too small λ values. Let us now make this statement more precise and also explain the reason behind.

These algorithms tend to behave effectively like TD(λ) with small constant λ, despite that they can have $\lambda_t = 1$ at some time steps t. This is due to the nature of TD learning with time-varying λ, which is very different from that of TD with constant λ. For time-varying λ, a large λ_t at one time step need not mean that we are using the information of the cumulative rewards over a long time horizon to estimate the value at the state S_t encountered at time t. Because the next λ_{t+1} could be very small or even zero, forcing a TD algorithm to "bootstrap" immediately. When large λ_t's are interleaved with small ones, we are effectively in the situation of TD with small λ. This could occur to our proposed scheme as well if, for example, in Example 2.1 the thresholds $C_{ss'}$ are set too small. When we use larger thresholds, we allow larger λ. By comparison, Retrace, ABQ, and Tree-Backup constrain the state-dependent λ-parameters to be small enough so that all the products $\lambda_t\gamma_t\rho_{t-1} < 1$, and this makes them prone to the small-λ issue just mentioned. (See the experiments in Section 4.2 for demonstrations.)

While we consider Retrace for approximate policy evaluation, the Munos et al. (2016) paper actually focuses primarily on finding an optimal policy for an MDP, in the tabular case, and it has demonstrated good empirical performance of Retrace and Tree-Backup for that purpose. Despite this, its results are not adequate yet to establish asymptotic optimality of these algorithms in the online optimistic policy iteration setting (personal communication with Munos), and it is still an open theoretical question whether online TD algorithms can solve an MDP like the Q-learning algorithm (Watkins, 1989; Tsitsiklis, 1994), when positive λ (small or not) and rapidly changing target policies are involved.

We also mention that for policy evaluation, Munos et al. (2016, Section 3.1) have also conceived the use of generalized Bellman operators, although they did not relate these operators explicitly to history-dependent λ's and did not study corresponding algorithms in this general case.

2.3 Ergodicity Result

The properties of the joint state-trace process $\{(S_t, y_t, e_t)\}$ are important for understanding and characterizing the behavior of our proposed TD learning scheme. We study them in this subsection. Most importantly, we shall establish the ergodicity of the state-trace process. The result will be useful in convergence analysis of several associated TD algorithms (Yu,

2017), although in this paper we discuss only the LSTD algorithm. In the next section we will also use the ergodicity result when we relate the LSTD equation (2.4) to a generalized Bellman equation for the target policy in order to interpret the LSTD solutions.

We note that to obtain the results in this subsection, we will follow similar lines of argument used in (Yu, 2012) for analyzing off-policy LSTD with constant λ. However, because λ is now history-dependent, some proof steps in (Yu, 2012) no longer apply. We shall explain this in more detail after we prove the main result of this subsection.

As another side note, one can introduce nonnegative coefficients $i(y)$ for memory states y to weight the state features (similarly to the use of "interest" weights in the ETD algorithm (Sutton et al., 2016)) and update e_t according to

$$e_t = \lambda_t \gamma_t \rho_{t-1} e_{t-1} + i(y_t)\,\phi(S_t). \tag{2.12}$$

The results given below apply to this update rule as well.

Let us start with two basic properties of $\{(S_t, y_t, e_t)\}$ that follow directly from our choice of the λ function:

(i) By Condition 2.3(i), for each y, $\lambda(y, e)e$ is a continuous function of e, and thus e_t depends continuously on e_{t-1}. This, together with the finiteness of $\mathcal{S} \times \mathcal{M}$, ensures that $\{(S_t, y_t, e_t)\}$ is a weak Feller Markov chain.[5]

(ii) Then, by a property of weak Feller Markov chains (Meyn and Tweedie, 2009, Theorem 12.1.2(ii)), the boundedness of $\{e_t\}$ ensured by Condition 2.3(ii) implies that $\{(S_t, y_t, e_t)\}$ has at least one invariant probability measure.

The third property, given in the lemma below, concerns the behavior of $\{e_t\}$ for different initial e_0. It is an important implication of Condition 2.3(i); actually, it is our purpose of introducing the condition 2.3(i) in the first place. In the lemma, $\overset{a.s.}{\to}$ stands for "converges almost surely to."

Lemma 2.1 *Let $\{e_t\}$ and $\{\hat{e}_t\}$ be generated by the iteration (2.2) and (2.5), using the same trajectory of states $\{S_t\}$ and initial y_0, but with different initial e_0 and $\hat{e}_0$, respectively. Then under Conditions 2.1(i) and 2.3(i), $e_t - \hat{e}_t \overset{a.s.}{\to} 0$.*

Proof The proof is similar to that of (Yu, 2012, Lemma 3.2). Let $\Delta_t = \|e_t - \hat{e}_t\|$, and let $\mathcal{F}_t$ denote the σ-algebra generated by $S_k, k \le t$. Note that under our assumption, in the generation of the two trace sequences $\{e_t\}$ and $\{\hat{e}_t\}$, the states $\{S_t\}$ and the memory states $\{y_t\}$ are the same, but the λ-parameters are different. Let us denote them by $\{\lambda_t\}$ and $\{\hat{\lambda}_t\}$ for the two trace sequences, respectively. Then by (2.2), $e_t - \hat{e}_t = \gamma_t \rho_{t-1}(\lambda_t e_{t-1} - \hat{\lambda}_t \hat{e}_{t-1})$, and by Condition 2.3(i), $\|\lambda_t e_{t-1} - \hat{\lambda}_t \hat{e}_{t-1}\| \le \|e_{t-1} - \hat{e}_{t-1}\|$. Hence $\|e_t - \hat{e}_t\| \le \gamma_t \rho_{t-1} \|e_{t-1} - \hat{e}_{t-1}\|$, so $\mathbb{E}\big[\Delta_t \big| \mathcal{F}_{t-1}\big] \le \mathbb{E}\big[\gamma_t \rho_{t-1} \big| \mathcal{F}_{t-1}\big] \cdot \Delta_{t-1} \le \Delta_{t-1}$. This shows $\{(\Delta_t, \mathcal{F}_t)\}$ is a nonnegative supermartingale. By the supermartingale convergence theorem (Dudley, 2002, Theorem 10.5.7 and Lemma 4.3.3), $\{\Delta_t\}$ converges a.s. to a nonnegative random variable Δ_∞ with $\mathbb{E}[\Delta_\infty] \le \liminf_{t\to\infty} \mathbb{E}[\Delta_t]$. From the inequality $\|e_t - \hat{e}_t\| \le \gamma_t \rho_{t-1} \|e_{t-1} - \hat{e}_{t-1}\|$ for all t, we have $\Delta_t \le \Delta_0 \cdot \prod_{k=1}^{t} \gamma_k \rho_{k-1}$, from which a direct calculation shows $\mathbb{E}\big[\Delta_t\big] \le \Delta_0 \cdot \mathbf{1}^\top (P\Gamma)^t \mathbf{1}$

5. This means that for any bounded continuous function f on $\mathcal{S} \times \mathcal{M} \times \Re^n$ (endowed with the usual topology), with $X_t = (S_t, y_t, e_t)$, $\mathbb{E}\big[f(X_1) \mid X_0 = x\big]$ is a continuous function of x (Meyn and Tweedie, 2009, Prop. 6.1.1).

where $\mathbf{1}$ denotes the n-dimensional vector of all 1's. As $t \to \infty$, $(P\Gamma)^t$ converges to the zero matrix under Condition 2.1(i). Therefore, $\liminf_{t\to\infty} \mathbb{E}[\Delta_t] = 0$ and consequently, we must have $\Delta_\infty = 0$ a.s., i.e., $\Delta_t \overset{a.s.}{\to} 0$. ■

We use Lemma 2.1 and ergodicity properties of weak Feller Markov chains (Meyn, 1989) to prove the ergodicity theorem below. A direct application to LSTD will be discussed immediately after the theorem, before we give its proof.

To state the result, we need some terminology and notation. For $\{(S_t, y_t, e_t)\}$ starting from the initial condition $x = (s, y, e)$, we write $\mathbf{P}_x$ for its probability distribution, and we write "$\mathbf{P}_x$-a.s." for "almost surely with respect to $\mathbf{P}_x$." The *occupation probability measures* are denoted by $\{\mu_{x,t}\}$, and they are random probability measures on $\mathcal{S} \times \mathcal{M} \times \Re^n$ given by

$$\mu_{x,t}(D) := \tfrac{1}{t}\textstyle\sum_{k=0}^{t-1} \mathbb{1}\big((S_k, y_k, e_k) \in D\big) \qquad \forall \text{ Borel sets } D \subset \mathcal{S} \times \mathcal{M} \times \Re^n,$$

where $\mathbb{1}(\cdot)$ is the indicator function. We are interested in the asymptotic convergence of these occupation probability measures in the sense of *weak convergence*: for probability measures $\{\mu_t\}$ and μ on a metric space, $\{\mu_t\}$ converges weakly to μ if $\int f d\mu_t \to \int f d\mu$ as $t \to \infty$, for every bounded continuous function f.

We shall also consider the Markov chain $\{(S_t, S_{t+1}, y_t, e_t)\}$, whose occupation probability measures are defined likewise. This Markov chain is essentially the same as $\{(S_t, y_t, e_t)\}$, but it is more convenient for applying our ergodicity result to TD algorithms because the temporal-difference term $\delta_t(v)$ involves (S_t, S_{t+1}, e_t). Regarding invariant probability measures of the two Markov chains, obviously, if ζ is an invariant probability measure of $\{(S_t, y_t, e_t)\}$, then an invariable probability measure of $\{(S_t, S_{t+1}, y_t, e_t)\}$ is the probability measure ζ_1 composed from the marginal ζ and the conditional distribution of S_1 given (S_0, y_0, e_0) specified by P^o; i.e.,

$$\zeta_1(D) = \int \textstyle\sum_{s' \in \mathcal{S}} P^o_{ss'}\, \mathbb{1}\big((s, s', y, e) \in D\big)\, \zeta\big(d(s, y, e)\big) \qquad \forall \text{ Borel sets } D \subset \mathcal{S}^2 \times \mathcal{M} \times \Re^n. \tag{2.13}$$

(In the above, we used the notation $\int f(x)\, \zeta(dx)$ to write the integral of f w.r.t. ζ, and the notation $\zeta\big(d(s, y, e)\big)$ is the same as $\zeta(dx)$ with $x = (s, y, e)$.)

Theorem 2.1 *Let Conditions 2.1-2.3 hold. Then $\{(S_t, y_t, e_t)\}$ is a weak Feller Markov chain and has a unique invariant probability measure ζ. For each initial condition $x := (s, y, e)$ of (S_0, y_0, e_0), the occupation probability measures $\{\mu_{x,t}\}$ converge weakly to ζ, $\mathbf{P}_x$-a.s.*

Likewise, the same holds for $\{(S_t, S_{t+1}, y_t, e_t)\}$, whose unique invariant probability measure is as given in (2.13).

If the initial distribution of (S_0, y_0, e_0) is ζ, the state-trace process $\{(S_t, y_t, e_t)\}$ is stationary. Let $\mathbb{E}_\zeta$ denote expectation w.r.t. this stationary process. We now state a corollary of the above theorem for LSTD, before we prove the theorem.

Consider the sequence of equations in v, $\frac{1}{t}\sum_{k=0}^{t-1} e_k\, \delta_k(v) = 0$, appeared in (2.4) for LSTD. From the definition (2.3) of $\delta_t(v)$,

$$\delta_t(v) = \rho_t \left(R_t + \gamma_{t+1} v(S_{t+1}) - v(S_t)\right),$$

we see that for fixed v, every $e_k\, \delta_k(v)$ can be expressed as $f(S_k, S_{k+1}, e_k)$ for a continuous function f. Since the traces and hence the entire process lie in a bounded set under Condition 2.3(ii), the weak convergence of the occupation probabilities measures of $\{(S_t, S_{t+1}, y_t, e_t)\}$ shown by Theorem 2.1 implies that this sequence of equations has an asymptotic limit that can be expressed in terms of the stationary state-trace process as follows.

Corollary 2.1 *Let Conditions 2.1-2.3 hold. Then for each initial condition of (S_0, y_0, e_0), almost surely, the sequence of linear equations in v, $\frac{1}{t}\sum_{k=0}^{t-1} e_k\, \delta_k(v) = 0$, tends asymptotically to $\mathbb{E}_\zeta[e_0\, \delta_0(v)] = 0$ (also a linear equation in v), in the sense that the random coefficients in the former equations converge to the corresponding coefficients in the latter equation as $t \to \infty$.*

In the rest of this section we prove Theorem 2.1. Broadly speaking, the line of argument is as follows: We first prove the weak convergence of occupation probability measures to the same invariant probability measure, for each initial condition. This will in turn imply the uniqueness of the invariant probability measure.

After the proof we will first comment in Remark 2.1 on the differences between our proof and that of a similar result in the previous work (Yu, 2012). We will then comment in Remark 2.2 about the technical roles of Condition 2.3 (which concerns the choice of the function $\lambda(\cdot)$) and whether some part of that condition can be relaxed.

Proof of Theorem 2.1 As we discussed before Lemma 2.1, under Conditions 2.3, $\{(S_t, y_t, e_t)\}$ is weak Feller and has at least one invariant probability measure ζ. Then, by (Meyn, 1989, Prop. 4.1), there exists a set $D \subset \mathcal{S} \times \mathcal{M} \times \Re^n$ with ζ-measure 1 such that for each initial condition $x = (s, y, e) \in D$, the occupation probability measures $\{\mu_{x,t}\}$ converge weakly, $\mathbf{P}_x$-a.s., to an invariant probability measure μ_x that depends only on the initial condition x. To prove the theorem using this result, we need to show that (i) all these $\{\mu_x \mid x \in D\}$ are the same invariant probability measure, and (ii) for all $x \notin D$, $\{\mu_{x,t}\}$ has the same weak convergence property.

To this end, we first consider an arbitrary pair (s, y_s) in the recurrent class of $\{(S_t, y_t)\}$ (cf. Condition 2.2). Let us show that for all initial conditions $x \in \{(s, y_s, e) \mid e \in \Re^n\}$, $\{\mu_{x,t}\}$ converges weakly to the same invariant probability measure, almost surely.

Since the finite-state Markov chain $\{(S_t, y_t)\}$ has a single recurrent class (Condition 2.2) and its evolution is not affected by $\{e_t\}$, the marginal of ζ on $\mathcal{S} \times \mathcal{M}$ coincides with the unique invariant probability distribution of $\{(S_t, y_t)\}$. So the fact that $\zeta(D) = 1$ and (s, y_s) is a recurrent state of $\{(S_t, y_t)\}$ implies that there exists some $\hat{e}$ with $(s, y_s, \hat{e}) \in D$. For the initial condition $\hat{x} = (s, y_s, \hat{e})$, by the result of (Meyn, 1989) mentioned earlier, $\{\mu_{\hat{x},t}\}$ converges weakly to $\mu_{\hat{x}}$, almost surely.

Now consider $x = (s, y_s, e)$ for an arbitrary $e \in \Re^n$. Generate iterates $\{\hat{e}_t\}$ and $\{e_t\}$ according to (2.2), using the same trajectory $\{(S_t, y_t)\}$ with $(S_0, y_0) = (s, y_s)$, but with $\hat{e}_0 = \hat{e}$ and $e_0 = e$. By Lemma 2.1, $\hat{e}_t - e_t \overset{a.s.}{\to} 0$. Therefore, except on a null set of sample

paths, it holds for all bounded Lipschitz continuous functions f on $\mathcal{S}\times\mathcal{M}\times\Re^n$ that[6]

$$\left|\int f d\mu_{\hat{x},t} - \int f d\mu_{x,t}\right| = \left|\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,\hat{e}_k) - \tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)\right| \to 0. \tag{2.14}$$

By the a.s. weak convergence of $\mu_{\hat{x},t}$ to $\mu_{\hat{x}}$ proved earlier, except on a null set, $\int f d\mu_{\hat{x},t} \to \int f d\mu_{\hat{x}}$ for all such functions f. Combining this with (2.14) yields that almost surely, $\int f d\mu_{x,t} \to \int f d\mu_{\hat{x}}$ for all such f. By (Dudley, 2002, Theorem 11.3.3), this implies that almost surely, $\mu_{x,t} \to \mu_{\hat{x}}$ weakly.

Thus we have proved that for all initial conditions $x=(s,y_s,e), e\in\Re^n$, $\{\mu_{x,t}\}$ converges weakly, almost surely, to the same invariant probability measure $\mu_{\hat{x}}$. Denote $\mu=\mu_{\hat{x}}$. Let us now show that for any initial condition x, $\{\mu_{x,t}\}$ also converges to μ, $\mathbf{P}_x$-a.s.

Consider $\{(S_t,y_t,e_t)\}$ with an arbitrary initial condition $\bar{x}=(\bar{s},\bar{y},\bar{e})$. Let $\tau=\min\{t \mid (S_t,y_t)=(s,y_s)\}$ (the pair (s,y_s) is as in the proof above). Note that $\tau<\infty$ a.s., because (s,y_s) is a recurrent state of $\{(S_t,y_t)\}$. Define $(\tilde{S}_k,\tilde{y}_k)=(S_{\tau+k},y_{\tau+k})$, $\tilde{e}_k=e_{\tau+k}$ for $k\geq 0$.

By the strong Markov property (see e.g. Nummelin, 1984, Theorem 3.3), $\{(\tilde{S}_k,\tilde{y}_k)\}_{k\geq 0}$ has the same probability distribution as the Markov chain $\{(S_t,y_t)\}$ that starts from $(S_0,y_0)=(s,y_s)$. Therefore, by the preceding proof, $\mathbf{P}_{\bar{x}}$-almost surely, for all bounded continuous functions f on $\mathcal{S}\times\mathcal{M}\times\Re^n$,

$$\lim_{m\to\infty}\tfrac{1}{m}\sum_{k=0}^{m-1} f(\tilde{S}_k,\tilde{y}_k,\tilde{e}_k) = \int f d\mu. \tag{2.15}$$

Denote $a\wedge b=\min\{a,b\}$. Using (2.15) and the fact $\tau<\infty$ a.s., we have that $\mathbf{P}_{\bar{x}}$-almost surely,

$$\begin{aligned}\lim_{t\to\infty}\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k) &= \lim_{t\to\infty}\left(\tfrac{1}{t}\sum_{k=0}^{t\wedge(\tau-1)} f(S_k,y_k,e_k) + \tfrac{1}{t}\sum_{k=\tau}^{t-1} f(S_k,y_k,e_k)\right)\\ &= \lim_{t\to\infty}\tfrac{1}{t}\sum_{k=0}^{t-\tau-1} f(S_{\tau+k},y_{\tau+k},e_{\tau+k})\\ &= \lim_{m\to\infty}\tfrac{1}{m}\sum_{k=0}^{m-1} f(\tilde{S}_k,\tilde{y}_k,\tilde{e}_k) = \int f d\mu.\end{aligned}$$

This proves that $\{\mu_{x,t}\}$ converges weakly to μ almost surely, for each initial condition x.

It now follows that μ must be the unique invariant probability measure of $\{(S_t,y_t,e_t)\}$. To see this, suppose ζ is another invariant probability measure. For any bounded continuous function f, by stationarity, $\mathbb{E}_\zeta[\frac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)] = \int f d\zeta$ for all $t\geq 1$. On the other hand, the preceding proof has established that for all initial conditions x,

$$\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k) = \int f d\mu_{x,t} \to \int f d\mu, \quad \mathbf{P}_x\text{-a.s.},$$

which implies that if ζ is the initial distribution of (S_0,y_0,e_0), then $\frac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)\to\int f d\mu$, $\mathbf{P}_\zeta$-a.s. We thus have

$$\begin{aligned}\int f d\zeta = \mathbb{E}_\zeta\left[\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)\right] &= \lim_{t\to\infty}\mathbb{E}_\zeta\left[\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)\right]\\ &= \mathbb{E}_\zeta\left[\lim_{t\to\infty}\tfrac{1}{t}\sum_{k=0}^{t-1} f(S_k,y_k,e_k)\right] = \int f d\mu,\end{aligned}$$

6. Here we are using the same $(S_k,y_k), k\leq t$ in the occupation probability measures $\mu_{\hat{x},t}$ and $\mu_{x,t}$. This is valid because the e_t's do not affect the evolution of $\{(S_t,y_t)\}$ and are functions of these states and the given initial e_0. If we call the $\mu_{x,t}$ here $\tilde{\mu}_{x,t}$ instead and define $\mu_{x,t}$ using another independent copy of $\{(S_t,y_t)\}$, then since the two sequences of occupation probability measures will have the same probability distribution, $\{\mu_{x,t}\}$ will have the same weak convergence property as $\{\tilde{\mu}_{x,t}\}$.

where the third equality follows from the bounded convergence theorem. This shows $\int f d\zeta = \int f d\mu$ for all bounded continuous functions f, and hence $\zeta=\mu$ by (Dudley, 2002, Prop. 11.3.2), proving the uniqueness of the invariant probability measure.

The conclusions for the Markov chain $\{(S_t,S_{t+1},y_t,e_t)\}$ follow from the same arguments given above, if we replace S_t with (S_t,S_{t+1}) and replace the set $\mathcal{S}$ with the set of possible state transitions. (We could have proved the assertions for $\{(S_t,S_{t+1},y_t,e_t)\}$ first and then deduced as their implications the assertions for $\{(S_t,y_t,e_t)\}$. We treated the latter first, as it makes the notation in the proof simpler.) ■

Remark 2.1 (About the proof) Theorem 2.1 is similar to (Yu, 2012, Theorem 3.2) for off-policy LSTD with constant λ (the analysis given in (Yu, 2012) also applies to state-dependent λ). Some of the techniques used to prove the two theorems are also similar. The main difference to (Yu, 2012) is that in the proof here we used an argument based on the strong Markov property to extend the weak convergence property of $\{\mu_{x,t}\}$ for a subset of initial conditions $x\in\{(s,y_s,e)\mid e\in\Re^n\}$ to all initial conditions, whereas in (Yu, 2012) this step was proved using a result on the convergence-in-mean of LSTD iterates established first. The latter approach would not work here due to the dependence of λ_t on the history. Indeed, due to this dependence, the proof of the convergence-in-mean of LSTD given in (Yu, 2012) does not carry over to our case, even though that convergence does hold as a consequence of Theorem 2.1, in view of the boundedness of traces by construction. Compared with the proof of the ergodicity result in (Yu, 2012), the proof we gave here is more direct and therefore better.

Regarding possible alternative proofs of Theorem 2.1, let us also mention that if we prove first the uniqueness of the invariant probability measure, then, since $\{(S_t,y_t,e_t)\}_{t\geq 1}$ lie in a bounded set, the weak convergence of occupation probability measures will follow immediately from (Meyn, 1989, Prop. 4.2). However, because the evolution of the λ_t's depends on both states and traces, it does not seem easy to us to prove directly the uniqueness part first. ■

Remark 2.2 (About the conditions on the function $\lambda(\cdot)$) Our proof of Theorem 2.1 relied on Lemma 2.1 and the two properties discussed preceding that lemma, namely, that $\{(S_t,y_t,e_t)\}$ is a weak Feller Markov chain and has at least one invariant probability measure. As long as these hold when we weaken or change the conditions on the function $\lambda(\cdot)$, the proof and the conclusions of the theorem will remain applicable.

We introduced Condition 2.3(ii) to bound the traces for algorithmic concerns. For the ergodicity of the state-trace process, Condition 2.3(ii) is unimportant—in fact, it can be removed from the conditions of Theorem 2.1. The reason is that we used this condition before Lemma 2.1 to quickly infer that $\{(S_t,y_t,e_t)\}$ has at least one invariant probability measure, but this is still true without Condition 2.3(ii), in view of (Meyn and Tweedie, 2009, Theorem 12.1.2(ii)) and the fact that under Condition 2.1(i), $\{e_t\}$ is bounded in probability (the proof of this fact is straightforward and similar to the proof of (Yu, 2012, Lemma 3.1) or (Yu, 2015, Prop. A.1)).

Condition 2.3(i) is actually two conditions combined into one. The first is the continuity of $\lambda(y,e)e$ in e for each y, which was used to ensure that the state-trace process is a weak

Feller Markov chain. To be more general, instead of letting the evolutions of the traces and memory states be governed by the functions λ and g, one may consider letting them be governed by stochastic kernels. Then by placing a suitable continuity condition on the stochastic kernel λ, one can ensure that the state-trace process has the desired weak Feller Markov property.

The second condition packed into Condition 2.3(i) is that for each y, $\lambda(y,e)e$ is a Lipschitz continuous function of e with modulus 1. This condition is somewhat restrictive, and one may consider instead allowing the function to have Lipschitz modulus greater than 1. However, additional conditions are then needed to ensure that Lemma 2.1 holds. (If this lemma does not hold, then the state-trace process may not be ergodic and one will need a different approach than the one we took to characterize the sample path properties of the state-trace process.)

From an algorithmic perspective, if it is desirable to choose even larger λ_t's or to have greater flexibility in choosing these λ-parameters, some of the generalizations just mentioned can be considered. For example, Condition 2.3(ii) can be replaced and stochastic kernels can be introduced to allow for occasionally large traces e_t, so that instead of having the traces bounded, one only make their variances bounded in a desired range. ■

3. Generalized Bellman Equations

In this section, we continue the analysis started in Section 2.3. Recall that Corollary 2.1 established that the asymptotic limit of the linear equations (2.4) for LSTD is the linear equation (in v):

$$\mathbb{E}_\zeta[\, e_0\, \delta_0(v)] = 0.$$

Our goal now is to relate this equation to a generalized Bellman equation for the target policy π. This will then allow us to interpret solutions of (2.4) computed by LSTD as solutions of approximate versions of that generalized Bellman equation.

To this end, we will first give a general description of randomized stopping times and associated Bellman operators (Section 3.1). We will then use these notions to derive the particular Bellman operators that correspond to our choices of the λ-parameters and appear in the linear equations for LSTD (Section 3.2). We will also discuss a composite scheme of choosing the λ-parameters as a direct application and extension of our results.

To simplify notation in subsequent derivations, we shall use the following shorthand notation: For $k \le m$, denote $S_k^m = (S_k, S_{k+1}, \dots S_m)$,

$$\rho_k^m = \textstyle\prod_{i=k}^m \rho_i, \qquad \lambda_k^m = \prod_{i=k}^m \lambda_i, \qquad \gamma_k^m = \prod_{i=k}^m \gamma_i. \tag{3.1}$$

Also, we shall treat $\rho_k^m = \lambda_k^m = \gamma_k^m = 1$ if $k > m$.

3.1 Randomized Stopping Times and Associated Bellman Operators

Consider the Markov chain $\{S_t\}$ induced by the target policy π. Let Condition 2.1(i) hold. Recall that for the value function v_π, we have that for each state $s \in \mathcal{S}$,

$$v_\pi(s) = \mathbb{E}_s^\pi\big[\textstyle\sum_{t=0}^\infty \gamma_1^t\, r_\pi(S_t)\big] \quad \text{(by definition)}$$

and

$$v_\pi(s) = r_\pi(s) + \mathbb{E}_s^\pi[\gamma_1 v_\pi(S_1)].$$

The second equation is the standard one-step Bellman equation.

To write generalized Bellman equations for π, we shall make use of *randomized stopping times* for $\{S_t\}$, a notion that generalizes naturally stopping times for $\{S_t\}$ in that whether to stop at time t depends not only on the past states S_0^t but also on certain random outcomes. A simple example is to toss a coin at each time and stop as soon as the coin lands on heads, regardless of the history S_0^t. (The corresponding Bellman equation is the one associated with TD(λ) for a constant λ; cf. Example 3.1.) Of interest here is the general case where the stopping decision does depend on the entire history.

To define a randomized stopping time formally, first, the probability space of $\{S_t\}$ is enlarged to take into account whatever randomization scheme that is used to make the stopping decision. (The enlargement will be problem-dependent, as the next subsection will demonstrate.) Then, on the enlarged space, a randomized stopping time τ for $\{S_t\}$ is a stopping time[7] relative to some increasing sequence of σ-algebras $\mathcal{F}_0 \subset \mathcal{F}_1 \subset \cdots$, where the sequence $\{\mathcal{F}_t\}$ is such that

(i) for all $t \ge 0$, $\mathcal{F}_t \supset \sigma(S_0^t)$ (the σ-algebra generated by S_0^t), and

(ii) relative to $\{\mathcal{F}_t\}$, $\{S_t\}$ remains to be a Markov chain with transition probability P, i.e., for all $s \in \mathcal{S}$, $\text{Prob}(S_{t+1} = s \mid \mathcal{F}_t) = P_{S_t s}$.

See (Nummelin, 1984, Chap. 3.3); in particular, see Prop. 3.6 in p. 31-32 therein for several equivalent definitions of randomized stopping times.

Note that if $\mathcal{F}_t = \sigma(S_0^t)$ for all t, then the history of states S_0^t fully determines whether $\tau \le t$ and τ reduces to a stopping time for the Markov chain $\{S_t\}$. The properties (i)-(ii) in the above definition encapsulate our earlier intuitive discussion about making stopping decisions, namely, stopping decisions are made based on the history S_0^t and additional random outcomes that do not affect the evolution of the Markov chain.

Like stopping times, the strong Markov property also holds for randomized stopping times for a Markov chain. This is an important basic property. It says that in the event $\tau < \infty$, conditioned on the σ-algebra $\mathcal{F}_\tau$ associated with the stopping time τ relative to $\{\mathcal{F}_t\}$ (which is the σ-algebra generated by the events that "happen before τ"), the conditional distribution of $(S_\tau, S_{\tau+1}, \dots)$ is the same as the probability distribution of a Markov chain $(S_0, S_1, \dots)$ with initial state $S_0 = S_\tau$ (Nummelin, 1984, Theorem 3.3).

The above abstract definition of a randomized stopping time allows us to write Bellman equations in general forms without worrying about the details of the enlarged space, which are not important at this point. For notational simplicity, when there is no confusion, we shall still write $\mathbf{P}^\pi$ for the probability measure on the enlarged probability space and use $\mathbb{E}^\pi$ and $\mathbb{E}_s^\pi$ to denote the expectation and conditional expectation given $S_0 = s$, respectively, w.r.t. $\mathbf{P}^\pi$.

If τ is a randomized stopping time for $\{S_t\}$, the strong Markov property (Nummelin, 1984, Theorem 3.3) allows us to express v_π in terms of $v_\pi(S_\tau)$ and the total discounted

7. A random time τ is called a stopping time relative to a sequence $\{\mathcal{F}_t\}$ of increasing σ-algebras if the event $\{\tau = t\} \in \mathcal{F}_t$ for every t.

rewards R^τ prior to stopping:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_s^\pi\left[\sum_{t=0}^{\tau-1}\gamma_1^t\, r_\pi(S_t) + \sum_{t=\tau}^{\infty}\gamma_1^\tau\cdot\gamma_{\tau+1}^t\, r_\pi(S_t)\right] \\ &= \mathbb{E}_s^\pi\big[R^\tau + \gamma_1^\tau\, v_\pi(S_\tau)\big], \end{aligned} \tag{3.2}$$

where $R^\tau = \sum_{t=0}^{\tau-1}\gamma_1^t\, r_\pi(S_t)$ for $\tau \in \{0,1,2,\ldots\}\cup\{+\infty\}$.[8] We can also write the Bellman equation (3.2) in terms of $\{S_t\}$ only, by taking expectation over τ:

$$\begin{aligned} v_\pi(s) &= \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty}\Big(\mathbb{1}(\tau>t)\cdot\gamma_1^t\, r_\pi(S_t) + \mathbb{1}(\tau=t)\cdot\gamma_1^t\, v_\pi(S_t)\Big)\right], \\ &= \mathbb{E}_s^\pi\left[\sum_{t=0}^{\infty}\Big(q_t^+(S_0^t)\cdot\gamma_1^t\, r_\pi(S_t) + q_t(S_0^t)\cdot\gamma_1^t\, v_\pi(S_t)\Big)\right], \end{aligned} \tag{3.3}$$

where

$$q_t^+(S_0^t) = \mathbf{P}^\pi(\tau > t \mid S_0^t), \qquad q_t(S_0^t) = \mathbf{P}^\pi(\tau = t \mid S_0^t). \tag{3.4}$$

The r.h.s. of (3.2) or (3.3) defines a generalized Bellman operator $T:\Re^N\to\Re^N$ associated with τ, which has several equivalent expressions; e.g.,

$$(Tv)(s) = \mathbb{E}_s^\pi\big[R^\tau + \gamma_1^\tau\, v(S_\tau)\big] = \mathbb{E}_s^\pi\Big[\sum_{t=0}^{\infty}\Big(q_t^+(S_0^t)\cdot\gamma_1^t\, r_\pi(S_t) + q_t(S_0^t)\cdot\gamma_1^t\, v(S_t)\Big)\Big], \quad s\in\mathcal{S}.$$

Depending on the context, one expression can be more convenient to use than the other. For example, the first expression is convenient for defining T through the associated τ and for deducing the contraction property of T, whereas expressions like the second will be of interest when we want to know more explicitly the particular T for our TD learning scheme and its dependence on the λ-parameters.

In common with one-step Bellman operator, the generalized Bellman operator T is affine and involves a substochastic matrix. If $\tau\geq 1$ a.s., then the value function v_π is the unique fixed point of T, i.e., the unique solution of $v = Tv$, and T is a sup-norm contraction. In fact, this can be shown for slightly more general τ:

Theorem 3.1 *Let Condition 2.1(i) hold, and let the randomized stopping time τ be such that $\mathbf{P}^\pi(\tau\geq 1 \mid S_0 = s) > 0$ for all states $s\in\mathcal{S}$. Then v_π is the unique fixed point of the generalized Bellman operator T associated with τ, and T is a contraction w.r.t. a weighted sup-norm on $\Re^N$.*

8. We explain the derivations in this footnote. In the case $\tau = 0$, $R^0 = 0$. In the case $\tau=\infty$, by Condition 2.1(i), $R^\infty = \sum_{t=0}^{\infty}\gamma_1^t\, r_\pi(S_t)$ is almost surely well-defined, while the second term $\gamma_1^\tau\, v_\pi(S_\tau)$ in (3.2) is 0 because $\gamma_1^\infty := \prod_{k=1}^{\infty}\gamma_k = 0$ a.s., under Condition 2.1(i). Equation (3.2) is derived as follows: By the strong Markov property (Nummelin, 1984, Theorem 3.3), on $\{\tau<\infty\}$,

$$\mathbb{E}^\pi\left[\sum_{t=\tau}^{\infty}\gamma_1^\tau\cdot\gamma_{\tau+1}^t\, r_\pi(S_t)\mid\mathcal{F}_\tau\right] = \gamma_1^\tau\cdot\mathbb{E}_{S_\tau}^\pi\left[\sum_{t=0}^{\infty}\gamma_1^t\, r_\pi(S_t)\right] = \gamma_1^\tau v_\pi(S_\tau).$$

Then, since the term $\mathbb{E}_s^\pi\big[\sum_{t=\tau}^{\infty}\gamma_1^\tau\cdot\gamma_{\tau+1}^t\, r_\pi(S_t)\big] = \mathbb{E}_s^\pi\big[\mathbb{1}(\tau<\infty)\cdot\sum_{t=\tau}^{\infty}\gamma_1^\tau\cdot\gamma_{\tau+1}^t\, r_\pi(S_t)\big]$, we use the property of the conditional expectation given $\mathcal{F}_\tau$ and the fact $\mathcal{F}_\tau\supset\sigma(S_0)$ to rewrite this term as

$$\mathbb{E}_s^\pi\big[\mathbb{1}(\tau<\infty)\cdot\mathbb{E}^\pi\big[\textstyle\sum_{t=\tau}^{\infty}\gamma_1^\tau\cdot\gamma_{\tau+1}^t\, r_\pi(S_t)\mid\mathcal{F}_\tau\big]\big] = \mathbb{E}_s^\pi[\mathbb{1}(\tau<\infty)\cdot\gamma_1^\tau v_\pi(S_\tau)] = \mathbb{E}_s^\pi[\gamma_1^\tau v_\pi(S_\tau)],$$

where in the last equality we also used the fact $\gamma_1^\infty = 0$ a.s. This gives (3.2).

We prove this theorem in Appendix A. The proof amounts to showing that if a state process evolves according to the substochastic matrix $\tilde{P}$ involved in the affine operator T, then all the states in $\mathcal{S}$ are transient (equivalently, the spectral radius of $\tilde{P}$ is less than 1 and $I-\tilde{P}$ is invertible (Puterman, 1994, Appendix A.4)). From this the conclusions of the theorem follow as a basic fact from nonnegative matrix theory (Seneta, 2006, Theorem 1.1), and one specific choice of the weights of the sup-norm in the theorem is simply the expected time for the process to leave $\mathcal{S}$ from each initial state (see e.g., the proof of (Bertsekas and Tsitsiklis, 1996, Prop. 2.2)).

For TD algorithms that do not use history-dependent λ, the random times τ and the corresponding Bellman operators T have simple descriptions:

Example 3.1 (TD with constant or state-dependent λ) Depending on the choice of λ, TD(λ) algorithms are associated with different randomized stopping times τ. In the case of constant λ, starting from time 1, we stop the system with probability $1-\lambda$ if it has not stopped yet; i.e.,

$$\tau\geq 1 \quad\text{and}\quad \mathbf{P}^\pi(\tau = t \mid \tau > t-1, S_0^t) = 1-\lambda, \quad \forall\, t\geq 1.$$

In particular, we always stop at $t=1$ if $\lambda = 0$, and we never stop if $\lambda = 1$. Similarly, for state-dependent λ where $\lambda_t = \lambda(S_t)$, a function of the current state, the preceding stopping probability is replaced by $1-\lambda(S_t)$: $\mathbf{P}^\pi(\tau = t \mid \tau > t-1, S_0^t) = 1-\lambda(S_t)$ for $t\geq 1$. In these cases, by taking expectations over τ, the corresponding Bellman operators can be expressed solely in terms of λ and the model parameters for the target policy. ■

3.2 Bellman Equation for the Proposed TD Learning Scheme

With the terminology of randomized stopping times, we are now ready to write down the generalized Bellman equation associated with the TD learning scheme proposed in Section 2.2. It corresponds to a particular randomized stopping time. We shall first describe this random time, from which a generalized Bellman equation follows as seen in the preceding subsection. That this is indeed the Bellman equation for our TD learning scheme will then be proved.

Consider the Markov chain $\{S_t\}$ *under the target policy* π. We define a randomized stopping time τ for $\{S_t\}$:

- Let $y_t, \lambda_t, e_t, t\geq 1$, evolve according to (2.5) and (2.2):

$$y_t = g(y_{t-1}, S_t), \qquad \lambda_t = \lambda(y_t, e_{t-1}), \qquad e_t = \lambda_t\,\gamma_t\,\rho_{t-1}\, e_{t-1} + \phi(S_t), \quad t\geq 1.$$

- Let the initial (S_0, y_0, e_0) be distributed according to ζ, the unique invariant probability measure in Theorem 2.1 for the state-trace process induced by the behavior policy.
- At time $t\geq 1$, we stop the system with probability $1-\lambda_t$ if it has not yet been stopped. Let τ be the time when the system stops ($\tau=\infty$ if the system never stops).

To make the dependence on the initial distribution ζ explicit, we write $\mathbf{P}_\zeta^\pi$ for the probability measure of this process.

Note that by definition λ_t *and* $\lambda_1^t = \prod_{k=1}^t \lambda_k$ *are functions of the initial* (y_0, e_0) *and states* S_0^t. From how the random time τ is defined, we have for all $t \geq 1$,

$$\mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t, y_0, e_0) = \lambda_1^t =: h_t^+(y_0, e_0, S_0^t), \tag{3.5}$$

$$\mathbf{P}_\zeta^\pi(\tau = t \mid S_0^t, y_0, e_0) = \lambda_1^{t-1}(1-\lambda_t) =: h_t(y_0, e_0, S_0^t), \tag{3.6}$$

and hence

$$q_t^+(S_0^t) := \mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t) = \int h_t^+(y, e, S_0^t)\, \zeta\big(d(y,e) \mid S_0\big), \tag{3.7}$$

$$q_t(S_0^t) := \mathbf{P}_\zeta^\pi(\tau = t \mid S_0^t) = \int h_t(y, e, S_0^t)\, \zeta\big(d(y,e) \mid S_0\big), \tag{3.8}$$

where $\zeta(d(y,e) \mid s)$ is the conditional distribution of (y_0, e_0) given $S_0 = s$, w.r.t. the initial distribution ζ. As before, we can write the generalized Bellman operator T associated with τ in several equivalent forms. Let $\mathbb{E}_\zeta^\pi$ denote expectation under $\mathbf{P}_\zeta^\pi$. Similarly to the derivation of (3.3), we can rewrite (3.2) in this case by taking expectation over τ conditioned on (S_0^t, y_0, e_0) to derive that for all $v : \mathcal{S} \to \Re, s \in \mathcal{S}$,

$$(Tv)(s) = \mathbb{E}_\zeta^\pi\Big[\textstyle\sum_{t=0}^\infty \lambda_1^t \gamma_1^t\, r_\pi(S_t) + \sum_{t=1}^\infty \lambda_1^{t-1}(1-\lambda_t)\gamma_1^t\, v(S_t) \mid S_0 = s\Big]. \tag{3.9}$$

Or express T in the form of (3.3) by further integrating over (y_0, e_0) and using (3.7)-(3.8):

$$(Tv)(s) = \mathbb{E}_\zeta^\pi\Big[\textstyle\sum_{t=0}^\infty \Big(q_t^+(S_0^t)\cdot\gamma_1^t\, r_\pi(S_t) + q_t(S_0^t)\cdot\gamma_1^t\, v(S_t)\Big) \,\Big|\, S_0 = s\Big], \tag{3.10}$$

for all $v : \mathcal{S} \to \Re, s \in \mathcal{S}$, where in the case $t = 0$, $q_0^+(S_0) = 1$ and $q_0(S_0) = 0$ since $\tau > 0$ by construction.

It will be useful later to express $TV - V$ in terms of temporal differences. From (3.9), by writing $\lambda_1^{t-1}(1-\lambda_t)\gamma_1^t\, v(S_t) = \lambda_1^{t-1}\gamma_1^t\, v(S_t) - \lambda_1^t\gamma_1^t\, v(S_t)$ and rearranging terms, we have for all $v : \mathcal{S} \to \Re, s \in \mathcal{S}$,

$$\begin{aligned}(Tv)(s) - v(s) &= \mathbb{E}_\zeta^\pi\Big[\textstyle\sum_{t=0}^\infty \lambda_1^t\gamma_1^t\, r_\pi(S_t) + \sum_{t=0}^\infty \lambda_1^t\gamma_1^{t+1}\, v(S_{t+1}) - \sum_{t=0}^\infty \lambda_1^t\gamma_1^t\, v(S_t) \mid S_0 = s\Big] \\ &= \mathbb{E}_\zeta^\pi\Big[\textstyle\sum_{t=0}^\infty \lambda_1^t\gamma_1^t\cdot\Big(r_\pi(S_t) + \gamma_{t+1}\, v(S_{t+1}) - v(S_t)\Big) \,\Big|\, S_0 = s\Big]. \end{aligned} \tag{3.11}$$

In a similar way, from (3.10), we can write[9]

$$(Tv)(s) - v(s) = \mathbb{E}_\zeta^\pi\Big[\textstyle\sum_{t=0}^\infty q_t^+(S_0^t)\cdot\gamma_1^t\cdot\Big(r_\pi(S_t) + \gamma_{t+1}\, v(S_{t+1}) - v(S_t)\Big) \,\Big|\, S_0 = s\Big].$$

Remark 3.1 Comparing the two expressions (3.9) and (3.10) of T, we remark that the expression (3.9) reflects the role of the λ_t's in determining the stopping time, whereas the expression (3.10), which has eliminated the auxiliary variables y_t and e_t, shows more clearly the dependence of the stopping time on the entire history S_0^t. It can also be seen, from the initial distribution ζ, the dependence of λ_t on the traces and the dependence of the traces on the function $\rho(\cdot)$ (which describes importance sampling ratios), that both the behavior policy and the choice of the feature representation assert a significant role in determining the Bellman operator T for the target policy. This is in contrast with off-policy TD learning that uses a constant λ, where the behavior policy and the approximation subspace affect only how one approximates the Bellman equation underlying TD, not the Bellman equation itself, which is solely determined by λ (cf. Example 3.1).

Furthermore, note that as the invariant distribution of the state-trace process, ζ is associated with the dynamic behavior of the states *and traces* under the behavior policy. Generally, there is no explicit expression of ζ in terms of P^o and the parameters in the λ function. As a result, in general we cannot express the operator T in terms of these parameters in the learning scheme. This is different from the case of TD(λ) where λ is a function of the present state only. ■

9. Since τ is a randomized stopping time for the Markov chain $\{S_t\}$, we have $\mathbf{P}_\zeta^\pi(\tau > t \mid S_0^{t+1}) = \mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t)$, so $\mathbf{P}_\zeta^\pi(\tau > t \mid S_0^t) - \mathbf{P}_\zeta^\pi(\tau = t+1 \mid S_0^{t+1}) = \mathbf{P}_\zeta^\pi(\tau > t+1 \mid S_0^{t+1})$, i.e., $q_t^+(S_0^t) - q_{t+1}(S_0^{t+1}) = q_{t+1}^+(S_0^{t+1})$. Thus we can write the term $q_t(S_0^t)$ in (3.10) for $t \geq 1$ as $q_{t-1}^+(S_0^{t-1}) - q_t^+(S_0^t)$, and the expression for $(Tv - v)(s)$ then follows by rearranging terms.

We now proceed to show how the Bellman equation $v = Tv$ given above relates to the off-policy TD learning scheme in Section 2.2. Some notation is needed. Denote by $\zeta_\mathcal{S}$ the invariant probability measure of the Markov chain $\{S_t\}$ *induced by the behavior policy*; note that it coincides with the marginal of ζ on $\mathcal{S}$. For two functions v_1, v_2 on $\mathcal{S}$, we write $v_1 \perp_{\zeta_\mathcal{S}} v_2$ if $\sum_{s\in\mathcal{S}} \zeta_\mathcal{S}(s)\, v_1(s)\, v_2(s) = 0$. If $\mathcal{L}$ is a linear subspace of functions on $\mathcal{S}$ and $v \perp_{\zeta_\mathcal{S}} v'$ for all $v' \in \mathcal{L}$, we write $v \perp_{\zeta_\mathcal{S}} \mathcal{L}$. Recall that ϕ is a function that maps each state s to an n-dimensional feature vector. Denote by $\mathcal{L}_\phi$ the subspace spanned by the n component functions of ϕ, which is the space of approximate value functions for our TD learning scheme. Recall also that $\mathbb{E}_\zeta$ denotes expectation w.r.t. the *stationary* state-trace process $\{(S_t, y_t, e_t)\}$ under the behavior policy (cf. Theorem 2.1).

Theorem 3.2 *Let Conditions 2.1-2.3 hold. Then as a linear equation in* v, $\mathbb{E}_\zeta\big[e_0\, \delta_0(v)\big] = 0$ *is equivalently* $Tv - v \perp_{\zeta_\mathcal{S}} \mathcal{L}_\phi$, *where* T *is the generalized Bellman operator for* π *given in (3.9) or (3.10).*

Remark 3.2 (On LSTD) Note that

$$Tv - v \perp_{\zeta_\mathcal{S}} \mathcal{L}_\phi, \quad v \in \mathcal{L}_\phi$$

is a projected version of the generalized Bellman equation $Tv - v = 0$ (projecting the left-hand side onto the approximation subspace $\mathcal{L}_\phi$ w.r.t. the $\zeta_\mathcal{S}$-weighted Euclidean norm). Theorem 3.2 and Corollary 2.1 together show that this is what LSTD solves in the limit.

Note also that although the generalized Bellman operator T is a contraction (Theorem 3.1), the composition of projection with T is in general not a contraction (cf. Example B.1 in Appendix B). Thus we cannot use contraction-based arguments to analyze approximation properties. For that purpose, we use the oblique projection viewpoint of Scherrer (2010). Specifically, if the preceding projected Bellman equation admits a unique solution $\bar{v}$, then $\bar{v}$ can be viewed as an oblique projection of v_π (Scherrer, 2010) and the approximation error $\bar{v} - v_\pi$ can be characterized as in (Yu and Bertsekas, 2010) by using the oblique projection viewpoint. The details of these are given in Appendix B. ■

Remark 3.3 (On gradient-based TD) While Theorem 3.2 is about the LSTD algorithm, it also helps prepare the ground for analyzing gradient-based algorithms similar to

those discussed in (Maei, 2011; Mahadevan et al., 2014). Like LSTD, these algorithms aim to solve the same projected generalized Bellman equation as characterized by Theorem 3.2 (cf. Remark 3.2). Their average dynamics, which is important for analyzing their convergence using the mean ODE approach from stochastic approximation theory (Kushner and Yin, 2003), can be studied based on the ergodicity result of Theorem 2.1, in essentially the same way as we did in Section 2.3 for the LSTD algorithm. For details of the convergence analysis of these gradient-based TD algorithms, see the recent work (Yu, 2017). ■

In the rest of this subsection, we give a corollary to Theorem 3.2, deferring the proofs of both the theorem and the corollary to the next subsection. The corollary concerns a composite scheme of setting λ, which is slightly more general than what Section 2.2 described. It results in a Bellman operator that is a composition of the components of other Bellman operators, and it can be useful in practice for variance control. Let us describe the scheme first, before explaining our motivation for it.

Partition the state space into m nonempty disjoint sets: $\mathcal{S} = \cup_{i=1}^m \mathcal{S}_i$. Associate each set $\mathcal{S}_i$ with a possibly different scheme of setting λ that is of the type described in Section 2.2, and denote its memory states by $y_t^{(i)}$ and λ-function by $\lambda^{(i)}(\cdot,\cdot)$. Keep m trace vectors $e_t^{(1)},\ldots,e_t^{(m)}$, one for each set, and update them according to

$$e_t^{(i)} = \lambda_t^{(i)}\,\gamma_t\,\rho_{t-1}\,e_{t-1}^{(i)} + \phi(S_t)\,\mathbb{1}(S_t\in\mathcal{S}_i), \qquad 1\le i\le m, \tag{3.12}$$

where $\lambda_t^{(i)} = \lambda^{(i)}\big(y_t^{(i)}, e_{t-1}^{(i)}\big)$. We then have m ergodic state-trace processes that share the same state variables, $\big\{\big(S_t, y_t^{(i)}, e_t^{(i)}\big)\big\}$, $i=1,2,\ldots,m$. Each process has a unique invariant probability measure $\zeta^{(i)}$ (Theorem 2.1) and an associated randomized stopping time $\tau^{(i)}$ and generalized Bellman operator $T^{(i)}$, as discussed in this subsection. Define now an operator T by concatenating the component mappings of $T^{(i)}$ for $\mathcal{S}_i$ as follows: for all $v\in\Re^N$ and $s\in\mathcal{S}$,

$$(Tv)(s) := (T^{(i)}v)(s) \quad \text{if } s\in\mathcal{S}_i. \tag{3.13}$$

Consider an LSTD algorithm that defines the trace e_t to be the sum of the m trace vectors,

$$e_t = \textstyle\sum_{i=1}^m e_t^{(i)}, \tag{3.14}$$

and uses the traces to form the linear equation as before,

$$\tfrac{1}{t}\textstyle\sum_{k=0}^{t-1} e_k\,\delta_k(v) = 0, \quad v=\Phi\theta.$$

Note that $\frac{1}{t}\sum_{k=0}^{t-1} e_k\delta_k(v)=0$ is the same as $\sum_{i=1}^m \frac{1}{t}\sum_{k=0}^{t-1} e_k^{(i)}\delta_k(v)=0$. By Corollary 2.1, as a linear equation in v, it tends asymptotically (as $t\to\infty$) to the linear equation $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}}\big[e_0^{(i)}\delta_0(v)\big]=0$.

Corollary 3.1 *Let Condition 2.1 hold. Consider the composite scheme of setting λ discussed above, and let Conditions 2.2-2.3 hold for each of the m schemes involved. Let LSTD calculate traces according to (3.12) and (3.14). Then the limiting linear equation (in v) associated with LSTD, $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}}\big[e_0^{(i)}\delta_0(v)\big]=0$, is equivalently $Tv - v \perp_{\zeta_\mathcal{S}} \mathcal{L}_\phi$, where T is the generalized Bellman operator for π given by (3.13) and has the same fixed point and contraction properties as stated in Theorem 3.1.*

The use of composite schemes will be demonstrated by experiments in Section 4.2.2. Here let us explain informally our motivation for such schemes.

Remark 3.4 (About composite schemes of setting λ) Our motivation for using the composite schemes is revealed by the equation (3.13). Typically each $T^{(i)}$ is designed to be simple to implement in TD learning. For example, if we ignore for now the bounding of traces introduced in Section 2.2 and just consider TD(λ) with constant λ, $T^{(i)}$ can be the Bellman operator $T^{(\lambda)}$ for TD(λ) with some constant λ. A simple, extreme example is to partition the state space into two sets, and associate one with $T^{(\lambda)}, \lambda=1$, and the other with $T^{(\lambda)}, \lambda=0$. Using the combination (3.13) of the two operators in TD then means that for the first set of states whose $\lambda=1$, we want to estimate their values by using the information about the total rewards received when starting from those states, whereas for the second set of states whose $\lambda=0$, we only use the information about their one-stage rewards and how these states relate to the "neighboring" states in the transition graph. While this way of using different kinds of information for different states is natural and useful for TD-based policy evaluation, it cannot be realized by keeping a single trace sequence as before and only letting λ_t evolve with states or histories. Indeed, in that case, as discussed in Section 2.2.2, interleaving large and small λ_t's would make the algorithm behave effectively like TD with small λ over the entire state space.

In the context of the more complex scheme of setting λ discussed in this paper, our motivation and reasons for considering composite schemes are the same. Each $T^{(i)}$ can be designed to be simple to implement, such as in the simple scaling example in Section 2.2. The parameters in the ith scheme can be chosen so that they encourage the use of large λ_t's throughout time or dictate the use of only small λ_t's. By combining component mappings of $T^{(i)}$ through (3.13), composite schemes allow us to use cumulative rewards and transition structures at different timescales for different states. This provides additional flexibility in managing the bias-variance trade-off when estimating the value function (see Figure 9 and Figure 11 in Section 4.2.2 for a demonstration).

Finally, we mention that for off-policy LSTD(λ) with constant λ, composite schemes were proposed in (Yu and Bertsekas, 2012) and analyzed in (Yu, 2012, Proposition 4.5, Section 4.3). Our Corollary 3.1 extends that result. The convergence analysis of the gradient-based algorithms for the composite schemes is given in (Yu, 2017). ■

3.3 Proofs of Theorem 3.2 and Corollary 3.1

We divide the proof of Theorem 3.2 into two steps. The first step deals with an expression for the trace vector, given in the following lemma. It is more subtle than the other step in the proof, which involves mostly calculations.

We start by extending the stationary state-trace process $\{(S_t,y_t,e_t)\}_{t\ge0}$ to $t=-1,-2,\ldots$, and work with a double-ended stationary process $\{(S_t,y_t,e_t)\}_{-\infty<t<\infty}$ (by Kolmogorov's existence theorem (Dudley, 2002, Theorem 12.1.2), such a process exists). Note that as before this is a Markov chain whose transition probability is defined by the behavior policy π^o together with the update rules (2.2) and (2.5) for e_t, y_t and λ_t, and the marginal distribution of each (S_t,y_t,e_t) is ζ. We keep using the notation $\mathbf{P}_\zeta$ and $\mathbb{E}_\zeta$ for this double-ended stationary Markov chain.

Recall the shorthand notation (3.1) introduced at the beginning of Section 3: For $k \le m$, $\rho_k^m = \prod_{i=k}^m \rho_i$, $\lambda_k^m = \prod_{i=k}^m \lambda_i$, $\gamma_k^m = \prod_{i=k}^m \gamma_i$, and in addition, $\lambda_1^0 = \gamma_1^0 = \rho_0^{-1} = 1$ by convention.

Lemma 3.1 *$\mathbf{P}_\zeta$-almost surely, $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$ is well-defined and finite, and*

$$e_0 = \phi(S_0) + \textstyle\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1}\, \phi(S_{-t}). \tag{3.15}$$

Proof First, we show $\mathbb{E}_\zeta\big[\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}\big] < \infty$. Indeed,

$$\mathbb{E}_\zeta\big[\textstyle\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}\big] = \sum_{t=1}^\infty \mathbb{E}_\zeta\big[\gamma_{1-t}^0 \rho_{-t}^{-1}\big] = \sum_{t=1}^\infty \zeta_{\mathcal{S}}^\top (P\Gamma)^t \mathbf{1} < \infty,$$

where the first equality follows from the monotone convergence theorem, the second equality from Condition 2.1(ii) and a direct calculation, and the last inequality follows from Condition 2.1(i) (since $(I - P\Gamma)^{-1} = \sum_{t=0}^\infty (P\Gamma)^t$). This implies $\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1} < \infty$, $\mathbf{P}_\zeta$-a.s., so $\gamma_{1-t}^0 \rho_{-t}^{-1} \to 0$ as $t \to \infty$, $\mathbf{P}_\zeta$-a.s. Since $\lambda_{1-t}^0 \le 1$ for all t, it also implies that

$$\mathbb{E}_\zeta\big[\textstyle\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1}\, \|\phi(S_{-t})\|\big] \le \max_{s\in\mathcal{S}} \|\phi(s)\| \cdot \mathbb{E}_\zeta\big[\textstyle\sum_{t=1}^\infty \gamma_{1-t}^0 \rho_{-t}^{-1}\big] < \infty. \tag{3.16}$$

It then follows from a theorem on integration (Rudin, 1966, Theorem 1.38, p. 28-29) that $\mathbf{P}_\zeta$-almost surely, the infinite series $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$ converges to a finite limit.

We now prove the expression for e_0. By unfolding the iteration (2.2) for e_t backwards in time, we have for all $m \ge 1$,

$$e_0 = \phi(S_0) + \textstyle\sum_{t=1}^{m-1} \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1}\, \phi(S_{-t}) + \lambda_{1-m}^0 \gamma_{1-m}^0 \rho_{-m}^{-1}\, e_{-m}. \tag{3.17}$$

Let $m \to \infty$ in the r.h.s. of (3.17). For the last term, the trace e_{-m} lies in a bounded set by Condition 2.3(ii), $\lambda_{1-m}^0 \le 1$, and as we just showed, $\gamma_{1-m}^0 \rho_{-m}^{-1} \to 0$, $\mathbf{P}_\zeta$-a.s. So the last term converges to zero $\mathbf{P}_\zeta$-a.s. Also as we just showed, the second term converges $\mathbf{P}_\zeta$-almost surely to $\sum_{t=1}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1} \phi(S_{-t})$. The expression (3.15) for e_0 then follows. ■

Proof of Theorem 3.2 Treating $\lambda_1^0 = \gamma_1^0 = \rho_0^{-1} = 1$, we write the expression of e_0 given in Lemma 3.1 as $e_0 = \sum_{t=0}^\infty \lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1}\, \phi(S_{-t})$, $\mathbf{P}_\zeta$-a.s. We use this expression to calculate first $\mathbb{E}_\zeta\big[e_0 \cdot \rho_0 f(S_0^1)\big]$ for an arbitrary function f on $\mathcal{S} \times \mathcal{S}$. (Note that f is bounded and measurable, since $\mathcal{S}$ is finite.) We have

$$\begin{aligned}
\mathbb{E}_\zeta\big[e_0 \cdot \rho_0 f(S_0^1)\big] &= \textstyle\sum_{t=0}^\infty \mathbb{E}_\zeta\Big[\lambda_{1-t}^0 \gamma_{1-t}^0 \rho_{-t}^{-1}\, \phi(S_{-t}) \cdot \rho_0 f(S_0^1)\Big] \\
&= \textstyle\sum_{t=0}^\infty \mathbb{E}_\zeta\Big[\lambda_1^t \gamma_1^t \rho_0^{t-1} \phi(S_0) \cdot \rho_t f(S_t^{t+1})\Big] \\
&= \textstyle\sum_{t=0}^\infty \mathbb{E}_\zeta\Big[\phi(S_0) \cdot \mathbb{E}_\zeta\left[\lambda_1^t \gamma_1^t \rho_0^t\, f(S_t^{t+1}) \mid S_0, y_0, e_0\right]\Big]
\end{aligned} \tag{3.18}$$

where we used the stationarity of the double-ended state-trace process to derive the second equality, and we changed the order of expectation and summation in the first equality. This change is justified by the dominated convergence theorem (cf. (3.16)), and so are similar interchanges of expectation and summation that will appear in the rest of this proof.

To proceed with the calculation, we relate the expectations in the summation in (3.18) to expectations w.r.t. the process with probability measure $\mathbf{P}_\zeta^\pi$ introduced in Section 3.2

(which we recall is induced by the target policy π and involves the randomized stopping time τ). Let $\tilde{\mathbb{E}}_\zeta^\pi$ denote expectation w.r.t. the marginal of $\mathbf{P}_\zeta^\pi$ on the space of $\{(S_t, y_t, e_t)\}_{t\ge 0}$. From the change of measure performed through ρ_0^t, we have

$$\mathbb{E}_\zeta\left[\lambda_1^t \gamma_1^t \rho_0^t\, f(S_t^{t+1}) \mid S_0, y_0, e_0\right] = \tilde{\mathbb{E}}_\zeta^\pi\left[\lambda_1^t \gamma_1^t\, f(S_t^{t+1}) \mid S_0, y_0, e_0\right], \quad t \ge 0. \tag{3.19}$$

Combining this with (3.18) and using the fact that ζ is the marginal distribution of (S_0, y_0, e_0) in both processes, we obtain

$$\begin{aligned}
\mathbb{E}_\zeta\big[e_0 \cdot \rho_0 f(S_0^1)\big] &= \textstyle\sum_{t=0}^\infty \tilde{\mathbb{E}}_\zeta^\pi\Big[\phi(S_0) \cdot \tilde{\mathbb{E}}_\zeta^\pi\left[\lambda_1^t \gamma_1^t\, f(S_t^{t+1}) \mid S_0, y_0, e_0\right]\Big] \\
&= \tilde{\mathbb{E}}_\zeta^\pi\Big[\phi(S_0) \cdot \textstyle\sum_{t=0}^\infty \tilde{\mathbb{E}}_\zeta^\pi\left[\lambda_1^t \gamma_1^t\, f(S_t^{t+1}) \mid S_0\right]\Big].
\end{aligned} \tag{3.20}$$

We now use (3.20) to calculate $\mathbb{E}_\zeta\big[e_0\, \delta_0(v)\big]$ for a given function v. Recall from (2.3) that $\delta_0(v) = \rho_0 \cdot \big(r(S_0^1) + \gamma_1 v(S_1) - v(S_0)\big)$, so we let $f(S_t^{t+1}) = r(S_t^{t+1}) + \gamma_{t+1} v(S_{t+1}) - v(S_t)$ in (3.20). Since $\mathbb{E}_\zeta^\pi[r(S_t^{t+1}) \mid S_0^t] = r_\pi(S_t)$, we have

$$\begin{aligned}
\textstyle\sum_{t=0}^\infty \tilde{\mathbb{E}}_\zeta^\pi\left[\lambda_1^t \gamma_1^t\, f(S_t^{t+1}) \mid S_0\right] &= \textstyle\sum_{t=0}^\infty \tilde{\mathbb{E}}_\zeta^\pi\Big[\lambda_1^t \gamma_1^t \Big(r_\pi(S_t) + \gamma_{t+1} v(S_{t+1}) - v(S_t)\Big) \,\Big|\, S_0\Big] \\
&= (Tv - v)(S_0),
\end{aligned}$$

where the last equality follows from the expression (3.11) for $TV - V$. Therefore, by (3.20),

$$\mathbb{E}_\zeta\big[e_0\, \delta_0(v)\big] = \textstyle\sum_{s\in\mathcal{S}} \zeta_{\mathcal{S}}(s)\, \phi(s) \cdot (Tv - v)(s), \tag{3.21}$$

and this shows that $\mathbb{E}_\zeta\big[e_0\, \delta_0(v)\big] = 0$ is equivalent to $Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_\phi$. ■

We now prove Corollary 3.1.

Proof of Corollary 3.1 We apply Theorem 3.2 to each state-trace process $\big\{\big(S_t, y_t^{(i)}, e_t^{(i)}\big)\big\}$ for $i = 1, 2, \ldots, m$. Specifically, by (3.21) and the definition (3.12) of $e_t^{(i)}$,

$$\mathbb{E}_{\zeta^{(i)}}\big[e_0^{(i)} \delta_0(v)\big] = \textstyle\sum_{s\in\mathcal{S}} \zeta_{\mathcal{S}}(s) \cdot \phi(s) \mathbb{1}(s \in \mathcal{S}_i) \cdot \big(T^{(i)} v - v\big)(s).$$

Hence

$$\begin{aligned}
\textstyle\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}}\big[e_0^{(i)} \delta_0(v)\big] &= \textstyle\sum_{s\in\mathcal{S}} \zeta_{\mathcal{S}}(s)\, \phi(s) \cdot \Big[\sum_{i=1}^m \mathbb{1}(s \in \mathcal{S}_i) \cdot \big(T^{(i)} v - v\big)(s)\Big] \\
&= \textstyle\sum_{s\in\mathcal{S}} \zeta_{\mathcal{S}}(s)\, \phi(s) \cdot (Tv - v)(s),
\end{aligned}$$

where the last equality follows from the definition (3.13) of the operator T. This shows that the linear equation in v, $\sum_{i=1}^m \mathbb{E}_{\zeta^{(i)}}\big[e_0^{(i)} \delta_0(v)\big] = 0$, is equivalently $Tv - v \perp_{\zeta_{\mathcal{S}}} \mathcal{L}_\phi$.

We now prove that T has v_π as its unique fixed point and is a contraction with respect to a weighted sup-norm—in other words, Theorem 3.1 applies to T. For this, it suffices to show that T satisfies the conditions of Theorem 3.1, namely, T is a generalized Bellman operator associated with a randomized stopping time τ that satisfies $\mathbf{P}^\pi(\tau \ge 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$. We can define such a random time τ from the randomized stopping times $\tau^{(i)}$ associated with the Bellman operators $T^{(i)}$. In particular, by enlarging

the probability space if necessary, we can regard $\tau^{(i)}$, $i = 1, 2, \ldots, m$, as being defined on the same probability space.[10] We then let $\tau = \tau^{(i)}$ if $S_0 \in \mathcal{S}_i$. With this definition, we have $\mathbf{P}^\pi(\tau \geq 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$ (since $\tau^{(i)} \geq 1$ a.s. for all i). For each set $\mathcal{S}_i$, by (3.2), the component mappings of the generalized Bellman operator T_τ associated with τ are given by

$$(T_\tau v)(s) = \mathbb{E}_s^\pi\big[R^\tau + \gamma_1^\tau\, v(S_\tau)\big] = \mathbb{E}_s^\pi\big[R^{\tau^{(i)}} + \gamma_1^{\tau^{(i)}}\, v(S_{\tau^{(i)}})\big] = (T^{(i)} v)(s), \quad s \in \mathcal{S}_i.$$

So $T_\tau = T$ by the definition (3.13) of T; i.e., T is the Bellman operator associated with the randomized stopping time τ. ■

4. Numerical Study

In this section, we first use a toy problem to illustrate the behavior of traces calculated by off-policy LSTD(λ) for constant λ and for λ that evolves according to a simple special case of our proposed scheme described in Example 2.1. We then compare the behavior of LSTD for various choices of λ, on the toy problem and on the Mountain Car problem.

4.1 Behavior of Traces

The toy problem we use in this study has 21 states, arranged as shown in Figure 1 (left). One state is located at the centre, and the rest of the states split evenly into four groups, indicated by the four loops in the figure. The topology of the transition graph is the same for the target and behavior policies. We have drawn the transition graph only for the northeast group in Figure 1 (left); the states in each of the other three groups are arranged in the same manner and have the same transition structure. Given this symmetry, to specify the transition matrices P and P^o for the target and behavior policies π and π^o respectively, it suffices to specify the submatrices for the central state and one of the groups. If we label the central state as state 1 and the states in the northeast group clockwise as states 2-6, the submatrices of P and P^o for these states are given, respectively, by

$$\pi: \begin{pmatrix} 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0.8 & 0 & 0 \\ 0 & 0 & 0.2 & 0 & 0.8 & 0 \\ 0 & 0 & 0 & 0.2 & 0 & 0.8 \\ 0.8 & 0 & 0 & 0 & 0.2 & 0 \end{pmatrix}, \quad \pi^o: \begin{pmatrix} 0 & 0.25 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 & 0 & 0.5 \\ 0.5 & 0 & 0 & 0 & 0.5 & 0 \end{pmatrix}.$$

Intuitively speaking, from the central state, the system enters one group of states by moving diagonally in one of the four directions with equal probability, and after spending some time in that group, eventually returns to the central state and the process repeats. The behavior policy on average spends more time wandering inside each group than the target policy, while the target policy tends to traverse clockwise through the group more quickly.

10. That we can do so is clear from the definition of each $\tau^{(i)}$ as described at the beginning of Section 3.2 and from the fact that for each i, the initial distribution $\zeta^{(i)}$ on $(S_0, y_0^{(i)}, e_0^{(i)})$ has the same marginal on $\mathcal{S}$, which is $\zeta_{\mathcal{S}}$.

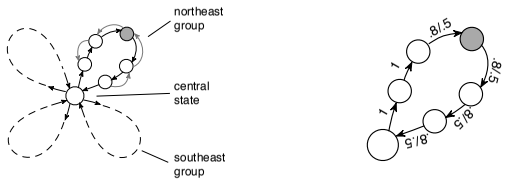


Figure 1: The transition graph of a toy problem (left) and a cycle pattern in it (right). The numbers appearing in the right graph indicate the importance sampling ratios for the state transitions represented by each directed edge. From such cycle patterns one can infer whether the trace sequence is unbounded almost surely.

All the rewards are zero except for the middle state in each group—for the northeast group, this is the shaded state in Figure 1 (left). For the two northern (southern) groups, their middle states have reward 1 (-1). The discount factor is $\gamma = 0.9$ for all states. As to features, we aggregate states into 5 groups: the 4 groups mentioned earlier and the central state forming its own group, and we let each state have 5 binary features indicating its membership.

We now discuss and illustrate the behavior of traces in this toy problem. For comparison, we first do this for the off-policy TD(λ) with a constant λ. It can help explain the challenges in off-policy TD learning and our motivation for proposing the new scheme of setting λ.

4.1.1 Traces for Constant λ

In this experiment we let $\lambda = 1$ and consider the trace iterates $\{e_t\}$ calculated by TD(1). In general, by identifying certain cycle patterns in the transition graph, one can infer whether $\{e_t\}$ will be unbounded over time almost surely (Yu, 2012, Section 3.1). Figure 1 (right) shows such a cycle of states in the transition graph of the toy problem. It consists of the central state and the northeast group of states. Labeled on each edge of the cycle is the importance sampling ratio for that state transition. Traversing through the cycle once from any starting state, and multiplying together the importance sampling ratios of each edge and the discount factors of the destination states, we get $\left(\frac{0.8}{0.5}\right)^4 \cdot \gamma^6 = \left(\frac{0.8}{0.5}\right)^4 \cdot 0.9^6 > 1$. From this one can infer that $\{e_t\}$ calculated by off-policy TD(1) will be unbounded in this problem (cf. Yu, 2012, Prop. 3.1).

We plotted in the upper left graph of Figure 2 the Euclidean norm $\|e_t\|$ of the traces over 8×10^5 iterations for TD(1). One can see the recurring spikes and the exceptionally large values of some of these spikes in the plot. This is consistent with the unboundedness of $\{e_t\}$ just discussed.

The unboundedness of $\{e_t\}$ tells us that the invariant probability measure ζ of the state-trace process $\{(S_t, e_t)\}$ has an unbounded support. Despite this unboundedness, $\{e_t\}$ is bounded in probability (Yu, 2012, Lemma 3.4) and under the invariant distribution ζ, $\mathbb{E}_\zeta\big[\|e_0\|\big] < \infty$ (Yu, 2012, Prop. 3.2). The latter property implies that under the invariant distribution, the probability of $\|e_0\| > x$ decreases as $o(1/x)$ for large x. Since the empirical

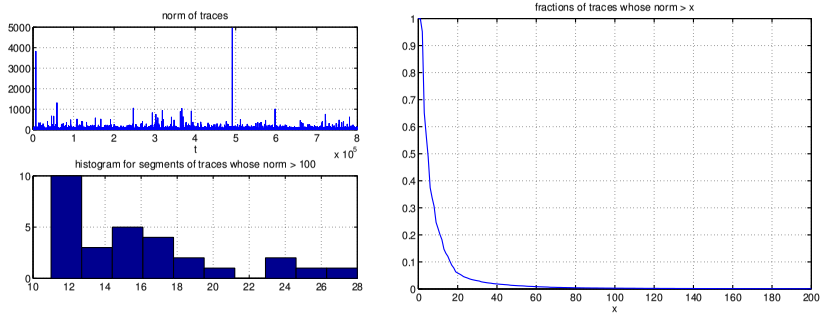


Figure 2: Statistics of traces for TD(1) in a toy problem (see the text in Section 4.1.1 for detailed explanations).

distribution of the state-trace process converges to ζ almost surely (Yu, 2012, Theorem 3.2), during a run of many iterations, we expect to see the fraction of traces with $\|e_t\| > x$ drop in a similar way as x increases.

The simulation result shown in the right part of Figure 2 agrees with the preceding discussion. Plotted in the graph are fractions of traces with $\|e_t\| > x$ during 8×10^5 iterations (the vertical axis indicates the fraction, and the horizontal axis indicates x). It can be seen that despite the recurring spikes in $\|e_t\|$ during the entire run, the fraction of traces with large magnitude x drops sharply with the increase in x.

While only a small fraction of traces have exceptionally large magnitude, they can occur in consecutive iterations. This is illustrated by the histogram in the lower left part of Figure 2. The histogram concerns the excursions of the trajectory $\{e_t\}$ outside of the ball $\{e \in \Re^n \mid \|e\| \leq 100\}$. The horizontal axis indicates the lengths of the excursions (where the length is the number of iterations an excursion contains), and the vertical axis indicates how many excursions of length x occurred during the 8×10^5 iterations of the experimental run. We plotted the histogram for lengths $x > 10$. It can be seen that one can have large traces during many consecutive iterations. Such behavior, although tolerable by LSTD, is especially detrimental to TD algorithms and can disrupt their learning. This is our main motivation for suggesting the use of λ-parameters to bound the traces directly.

4.1.2 Traces with Evolving λ

We now proceed to illustrate the behavior of traces and LSTD for λ that evolves according to our proposed scheme. Specifically, for this demonstration, we will use the simple scaling example given by (2.6)-(2.7) in Example 2.1, with all the thresholds $C_{ss'}$ being the same constant C. That is, the update rule for e_t used in this experiment is

$$e_t = \begin{cases} \gamma_t \rho_{t-1} e_{t-1} + \phi(S_t) & \text{if } \gamma_t \rho_{t-1} \|e_{t-1}\|_2 \leq C; \\ C \cdot \frac{e_{t-1}}{\|e_{t-1}\|_2} + \phi(S_t) & \text{otherwise.} \end{cases} \tag{4.1}$$

We first simulate the state-trace process to illustrate the ergodicity of this process stated by Theorem 2.1. We will shortly study the performance of LSTD for different values of C in Section 4.2.1. The results of these two experiments are shown in Figure 3 and Figures 4-5, respectively, and the details are as follows.

According to the ergodicity result of Theorem 2.1, no matter from which initial state and trace pair (S_0, e_0) we generate a trajectory $\{(S_t, e_t)\}_{0 \leq t \leq \bar{t}}$ according to the behavior policy, the empirical distribution of state and trace pairs in this trajectory should converge, as $\bar{t} \to \infty$, to the same distribution on $\mathcal{S} \times \Re^n$, which is the marginal of the invariant probability measure ζ on that space. Since $\mathcal{S}$ is discrete, we can verify this fact by examining the empirical conditional distribution of the trace given the state. In other words, for each state s, we examine the empirical distribution of the trace for the sub-trajectory (S_{t_k}, e_{t_k}), $k = 1, 2, \ldots$, where $S_{t_k} = s$ and it is the kth visit to state s by the trajectory. We check if this empirical distribution converges to the same one as we increase the length $\bar{t}$ of the trajectory and as we vary the initial condition (S_0, e_0).

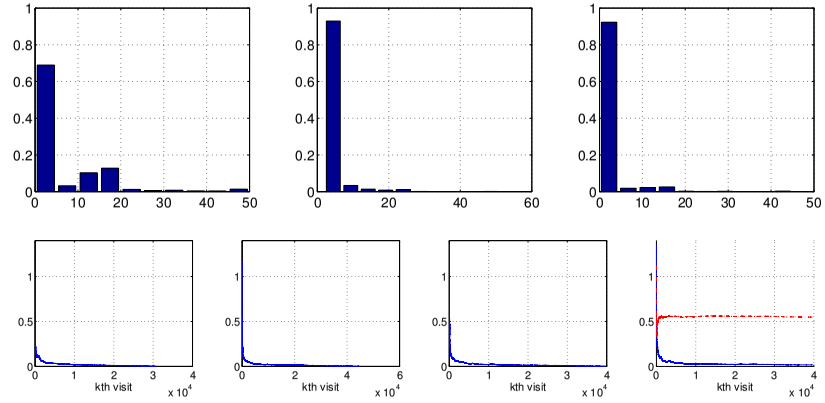


Figure 3: Demonstration of convergence of empirical conditional distributions on the trace space. (See the text in Section 4.1.2 for detailed explanations.)

To give a sense of what these limiting distributions over the trace space look like, we set the parameter $C = 50$ and generated a long trajectory with 8×10^5 iterations. In the top row of Figure 3, we plotted three normalized histograms of the first trace component for the sub-trajectories associated with three states of the toy problem, respectively: the central state (left), the middle state of the northeast group (middle), and the first state of the southeast group (right).

To check whether the empirical conditional distributions on the trace space converge along the sub-trajectory $\{(S_{t_k}, e_{t_k})\}$ for a given state, we compare the characteristic functions f_k of these distributions with the characteristic function f of the empirical conditional distribution obtained at the end of the sub-trajectory during the experimental run. In par-

ticular, we evaluate all these (complex-valued) characteristic functions at 500 points, which are chosen randomly according to the multivariate normal distribution on $\Re^5$ with mean 0 and covariance matrix $200^2 I$. We take the maximal difference between f_k and f at these 500 points as an indicator of the deviation between the two corresponding distributions.[11] In the bottom row of Figure 3, the first three plots show the difference curves obtained in the way just described, for three different states, respectively. These three states are the same ones mentioned earlier in the description of the top row of Figure 3. The horizontal axis of these plots indicates k, the number of visits to the corresponding state. As can be seen, the difference curves all tend to zero as k increases, which is consistent with the predicted convergence of the empirical conditional distributions on the trace space.

So far we compared the empirical distributions along the same trajectory. Next we compare them against the one obtained at the end of another trajectory that starts from a different initial condition. The difference curve for one state (the first state in the southeast group) is plotted in the last graph in the bottom row of Figure 3, and it is the lower curve in that graph. As can be seen, the curve tends to zero, suggesting that the limiting distribution of these empirical distributions does not change if we vary the initial condition, which is consistent with Theorem 2.1.

For comparison, we also plotted the difference curve when these same empirical conditional distributions are compared against the empirical conditional distribution obtained from the same trajectory but for a different state (specifically, the middle state of the northeast group). This is the upper curve in the last graph in the bottom row of Figure 3. It clearly indicates that for the two states, the associated limiting conditional distributions on the trace space are different. It also shows that the characteristic function approach we adopted in this experiment can effectively distinguish between two different distributions (cf. Footnote 11).

4.2 LSTD with Evolving λ

We now present experiments on the LSTD algorithm.

4.2.1 A Toy Problem

Let us first continue with the toy problem of the previous subsection and show how the LSTD algorithm performs in this problem as we vary the parameter C in the λ function for bounding the traces. We ran LSTD for $C = 10, 20, \ldots, 100$, using the same trajectory, for 3×10^5 iterations, and we computed the (Euclidean) distance of these LSTD solutions to the asymptotic TD(1) solution (in the space of the θ-parameters), normalized by the norm of the latter. We then repeat this calculation 10 times, each time with an independently generated trajectory. Plotted in Figure 4 (left) against the values of C are the means and standard deviations of the normalized distances of LSTD solutions thus obtained.

11. Recall that a tight sequence $\{p_k\}$ of probability distributions on $\Re^m$ converges to a probability distribution p if and only if the characteristic functions of p_k converge pointwise to the characteristic function of p (Dudley, 2002, Lemma 9.5.5). Recall also that we are dealing with convergence in distribution here, which is much weaker than convergence in total variation, so we cannot use total variation as a metric on the distribution space in this case.

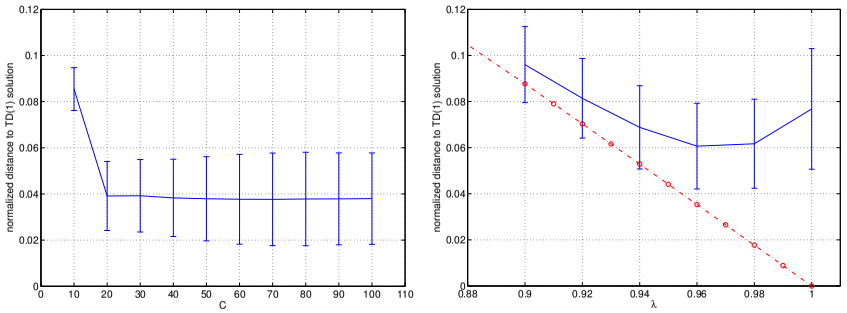


Figure 4: Compare LSTD solutions with evolving λ (left) and with constant λ (right). For constant λ, the red dot-dash curve in the right plot shows the quality of the asymptotic TD(λ) solutions, and LSTD(λ) would approach this curve in the limit, but due to variance issues, it can require an impractically large number of iterations to exhibit this convergent behavior. LSTD with evolving λ outperforms LSTD with constant λ in this case and effectively archives the quality of TD(λ) solutions for large constant λ.

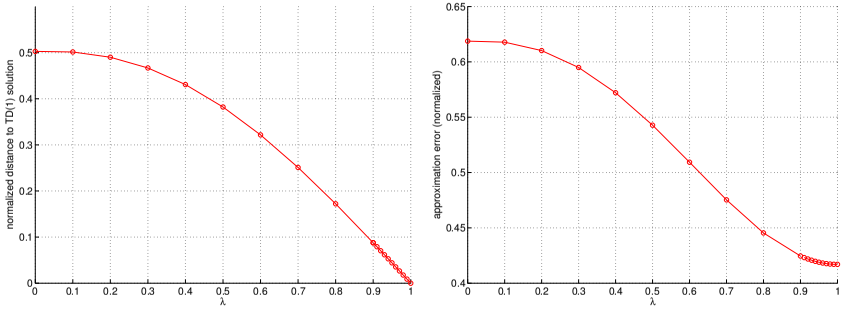


Figure 5: Approximation quality of asymptotic TD(λ) solutions with constant λ for the toy problem. Larger λ yields better approximations.

For comparison, we did the same for LSTD with a large constant λ: $\lambda = 0.9, 0.92, \ldots, 1$. Figure 4 (right) shows the result, where the dash-dot curve indicates the normalized distance of the asymptotic TD(λ) solution—the solution that LSTD(λ) would obtain in the limit. It can be seen that the performance of LSTD deteriorates as λ gets close to 1. We think that this is because due to the high variance issue, the convergence of LSTD with a large constant λ is too slow and requires far more iterations than the 3×10^5 iterations performed. In comparison, LSTD with evolving λ behaves better: it works effectively for $C \geq 20$, and

the approximation quality it achieved with such C is comparable to that of asymptotic TD(λ) solutions for a large constant λ around 0.96.

Figure 5 shows the quality of the asymptotic solutions of TD(λ) with constant λ, for the full range of λ values. Plotted in the left graph is the normalized distance of the TD(λ) solution to the TD(1) solution. Plotted in the right graph is the normalized approximation error of the corresponding approximate value function, where the error is measured by the weighted Euclidean norm with weights specified by $\zeta_{\mathcal{S}}$ (the invariant distribution on $\mathcal{S}$ under the behavior policy), and the normalization is over the weighted norm $\|v_\pi\|_{\zeta_{\mathcal{S}}}$ of the true value function v_π of the target policy. It can be seen that using $\lambda > 0.9$ provides considerably better approximations for this problem than using small λ.

We also found that for this problem, if we set λ according to the Retrace algorithm with $\beta = 1$ (cf. (2.8) in Example 2.2), then the performance of LSTD is comparable to TD(λ) with a small λ around 0.5. (Specifically, for Retrace, the normalized distance to the TD(1) solution is 0.37, and the normalized approximation error is 0.54, which are comparable to the numbers for TD(0.5), as Figure 5 shows.) This is not surprising, because, as we discussed earlier in Section 2.2.2, in keeping $\lambda_t \rho_{t-1} \leq 1$ always, Retrace and ABQ can be too "conservative," resulting in an overall effect that is like using a small λ, even though λ_t may appear to be large at times. Recall also that this can happen to our proposed scheme too. In the present experiment, for instance, this can happen when C is small; in particular, the case $C = 0$ reduces to LSTD(0).

4.2.2 Mountain Car Problem

In this subsection we demonstrate LSTD with evolving λ on a problem adapted from the well-known Mountain Car problem (Sutton and Barto, 1998). The details of this adaptation, including the target and behavior policies involved, can be found in the report (Yu, 2016a, Section 5.1, p. 23-26); most of these details are not crucial for our experiments, so to avoid distraction, we only describe briefly the experimental setup here.

In Mountain Car, the goal is to drive an underpowered car to reach the top of a steep hill, from the bottom of a valley. A state consists of the position and velocity of the car, whose values lie in the intervals $[-1.2, 0.5]$, $[-0.07, 0.07]$, respectively. The position 0.5 corresponds to the desired hill top destination, while the position $-\pi/6$ (≈ -0.52) lies at the bottom of a valley that is between the destination and a second hill peaked at -1.2 in the opposite direction (see the illustration in Figure 6). Except for the destination state, each state has three available actions: $\{\texttt{back}, \texttt{coast}, \texttt{forward}\}$, and the rewards depend only on the action taken and are -1.5, 0 and -1 for the three actions, respectively. The dynamics is as given in (Sutton and Barto, 1998). We consider undiscounted expected total rewards, so the discount factor is 1 except at the destination state, where the discount factor is 0 and from where the car enters a rewardless termination state permanently.

The target policy π is a simple but reasonably well-behaved policy. On either slopes between the two hills, it tries to increase its energy (kinetic plus gravitational potential energy) by accelerating in the direction of its current motion. If this brings it up to the opposite hill (position < -1), it coasts; otherwise, if its velocity drops to near zero, it goes forward or backward with equal probability. Figure 6 visualizes the total costs $-v_\pi$ of the

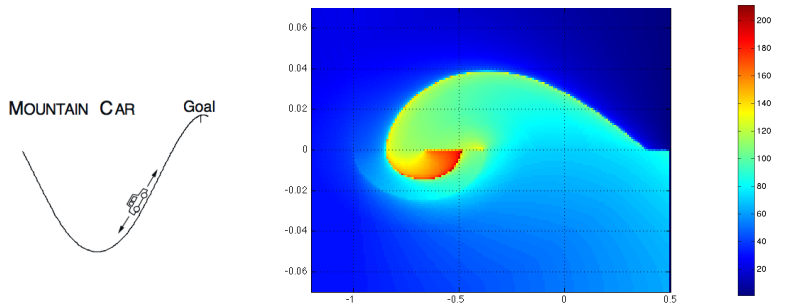


Figure 6: Left: Illustration of the Mountain Car problem. Right: Costs $-v_\pi$ for this problem are estimated and visualized as a color image, with the coloring scheme indicated by the colorbar. (The horizontal and vertical axes of the image correspond to position and velocity, respectively, for the 2-dimensional state space of this problem.)

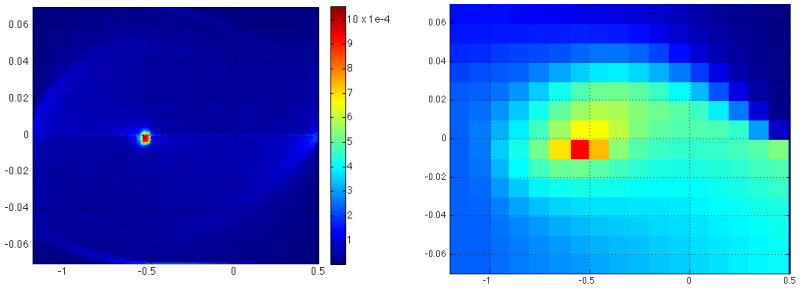


Figure 7: Left: Visualization of weights on states induced by the behavior policy. Right: The color image visualizes the approximation of $-v_\pi$ obtained from a discretized model for the Mountain Car problem, where the coloring scheme is the same as that shown in Figure 6 (right). The quality of this approximation is close to that of TD(0).

target policy,[12] where the horizontal (vertical) axis indicates position (velocity) and the

12. The values of v_π shown in Figure 6 are estimated by simulating the target policy for each starting state in a set of 171 × 141 points evenly spaced in the position-velocity space. In particular, the position (velocity) interval is evenly divided into subintervals of length 0.01 (0.001), and for each stating state, the target policy is simulated 600 times.

colorbar on the right shows the value corresponding to each color. The discontinuity of the function in certain regions can be seen in this figure.

The behavior policy π^o is an artificial policy that takes a random action (chosen with equal probability from the three actions) 90% of the time, and explores the state space by jumping to some random state 10% of the time. It also restarts when it is at the destination: with equal probability, it either restarts near the bottom of the valley or restarts from a random point sampled uniformly from the state space.[13]

We now explain how we will measure approximation qualities. Since the Mountain Car problem has a continuous state space, it is actually not covered by our analysis, which is for finite-state problems. Although we can treat it as essentially a finite-state problem (since the simulation is done with finite precision in computers), the number of states would still be too large to calculate the weights $\zeta_{\mathcal{S}}$. So, to measure weighted approximation errors $\|v - v_\pi\|$ for approximate value functions v produced by various LSTD algorithms in the subsequent experiments, we will compare v and v_π at a grid of points in the state space and calculate a weighted Euclidean distance between the function values at these grid points, using a set of weights precalculated by simulating the behavior policy.[14] The image in Figure 7 (left) visualizes these weights.

We use tile-coding (Sutton and Barto, 1998) to generate 145 binary features for our experiments.[15] The approximate value functions obtained with LSTD algorithms are thus piecewise constant. For comparison, we also build a discrete approximate model by state aggregation. The discretization is done at a resolution comparable to our tile-coding scheme, and the dynamics and rewards of this model are calculated based on data collected under the behavior policy. The solution of the discrete approximate model is shown in Figure 7 (right) (the coloring scheme for this and the subsequent images are the same as shown in Figure 6). It is similar to the approximate value function calculated by LSTD(0), which is shown in Figure 8 (first image, top row). As will be seen shortly, with positive λ, the approximation quality of LSTD improves. Thus the discrete model approximation approach is not as effective as the TD method in this case.

We now report the results of our experiments on the Mountain Car problem.

First Experiment: In this experiment, we compare three ways of setting λ: (i) Retrace with $\beta = 1$ (cf. Example 2.2); (ii) our simple scaling scheme with parameter C used in the previous experiments (cf. (4.1) and Example 2.1); and (iii) a composite scheme of the type discussed at the end of Section 3.2, which partitions the state space into two sets[16]

13. The behavior policy is exactly the same as described in (Yu, 2016a, p. 24-25) except for the possibility of restarting near the bottom of the valley whenever the destination is reached.
14. Specifically, we chose a grid of 171×141 points evenly spaced in the position-velocity space. We ran the behavior policy for 8×10^5 *effective* iterations, where an iteration is considered to be *ineffective* if the behavior policy takes an action, e.g., the restart action, that is impossible for the target policy. A visit to a state at an effective iteration was counted as a visit to the nearest grid point. At the end of the run, visits to a boundary point $(-1.2, 0)$ were disregarded as they were due to boundary effects in the dynamics of this problem, and the final counts were normalized to produce a set of weights on the grid points that sum to 1.
15. Two tilings are used: the first (second) comprises of 64 (81) uneven-sized rectangles that cover the state space. Together they produce a total of 145 binary features. The details of the coding scheme are as described in (Yu, 2016a, p. 28).
16. The first set consists of those states (position, velocity) with either position ≤ -0.9 or velocity ≥ 0.04. The rest of the states belong to the second set.

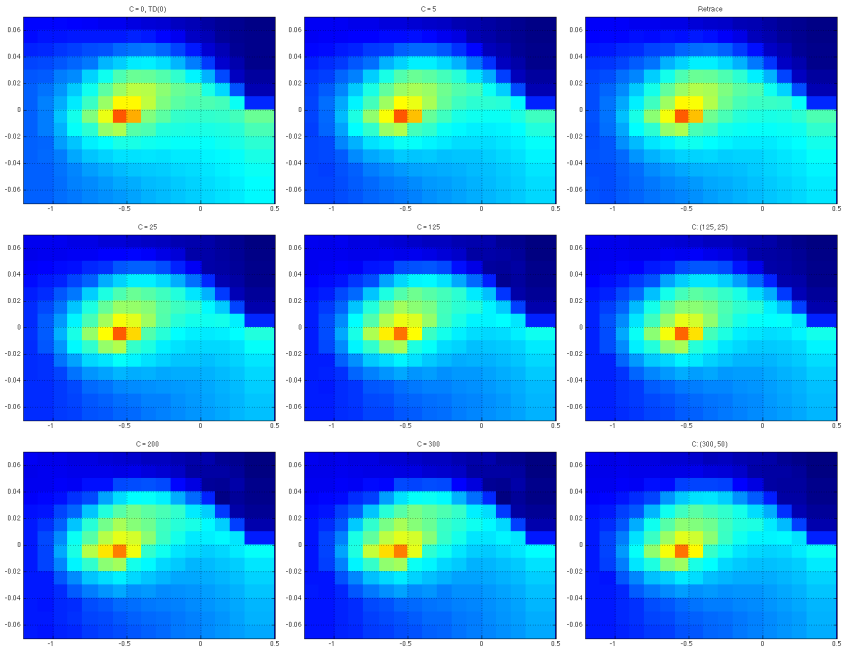


Figure 8: Visualized in the color images are approximations of $-v_\pi$ obtained by LSTD with different schemes of setting λ, where the coloring scheme is as that shown in Figure 6 (right). The choices of λ for each image, from left to right and top to bottom, are as follows. Top row: $C = 0$ (equivalent to LSTD(0)), $C = 5$, Retrace. Middle row: $C = 25$, $C = 125$, C: $(125, 25)$. Bottom row: $C = 200$, $C = 300$, C: $(300, 50)$.

and applies the simple scaling scheme with parameters C_1, C_2 for the first and second set, respectively. When referring to this composite scheme in the figures, we will use the designation C: (C_1, C_2).

We ran LSTD with different ways of setting λ just mentioned, on the same state trajectory generated by the behavior policy, for 6×10^5 effective iterations (cf. Footnote 14). Some of the approximate value functions obtained at the end of the run are visualized as images in Figure 8. It can be seen that the result of Retrace is similar to that of the simple scaling scheme with a small C, and as we increase C, the approximation from the scaling scheme improves.

To compare more precisely the approximation errors and see how they change over time for each algorithm, we did 10 independent runs, each of which consists of 6×10^5 effective

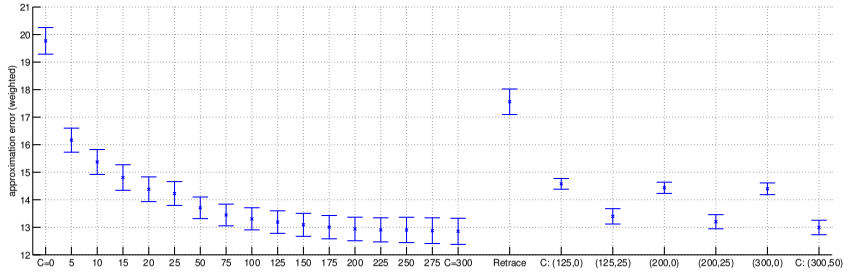


Figure 9: Compare the approximation error of LSTD for different schemes of setting λ.

iterations. The results are shown in Figures 9-11. Plotted in Figure 9 for each algorithm are the mean and standard deviation of the approximation errors for the 10 approximate value functions obtained by that algorithm at the end of the 10 runs. We can see from this figure the improvement in approximation quality as C increases. We can also see that the result of Retrace is in between those of $C = 0$ and $C = 5$, which is consistent with what the images in the top row of Figure 8 tell us.

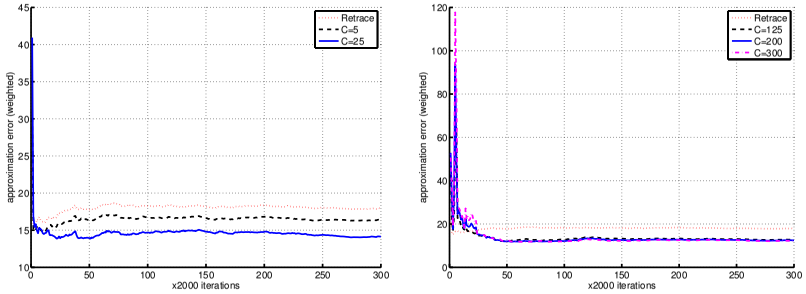


Figure 10: Compare the temporal behavior of LSTD for different schemes of setting λ.

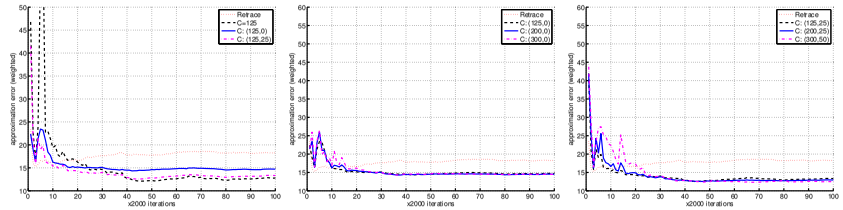


Figure 11: Compare the temporal behavior of LSTD for different schemes of setting λ.

Plotted in Figures 10-11 are the approximation errors calculated per 2000 effective iterations for each algorithm, during one of the 10 experimental runs. Figure 10 (left) shows how Retrace compares with the simple scaling with $C = 5$ and $C = 25$. It can be seen that the latter two achieved better approximation quality than Retrace without increases in variance. Figure 10 (right) shows that for larger values of C, variances also became larger initially; however, after about 5×10^4 iterations, these schemes overtook Retrace, yielding better approximations.

Figure 11 shows how the composite scheme of setting λ performs. Comparing the plots in this figure with the right plot in Figure 10, it can be seen that the composite schemes helped in reducing variances, and in about 2×10^4 iterations the schemes $C\colon (125, 0)$ and $C\colon (125, 25)$ overtook Retrace and yielded better approximations. Together with Figure 9, Figure 11 shows clearly the bias-variance trade-off of using composite schemes in this problem.

Second Experiment: Similarly to the previous experiment, we now compare our proposed method with several other ways of setting λ for the LSTD algorithm as well as with a constrained variant of LSTD: (i) Retrace with $\beta = 1$ as before; (ii) constant λ; (iii) constrained LSTD with constant λ; and (iv) the simple scaling scheme with parameter C. For constant $\lambda \in [0, 1]$, the constrained LSTD(λ) used in this experiment evolves the trace vectors as off-policy LSTD(λ) does, but it forms and solves the linear equation $\frac{1}{t}\sum_{k=0}^{t-1}[e_k]_{50} \cdot \delta_k(v) = 0, v = \Phi\theta$ instead, where the function $[\cdot]_{50}$ truncates each component of the trace vector to be within the interval $[-50, 50]$. (Such an algorithm follows naturally from the ergodicity of the state-trace process and the approximation of an unbounded integrable function by a bounded one. For a detailed discussion, see (Yu, 2016b, Section 3.2) or (Yu, 2017, Section 3.3).)

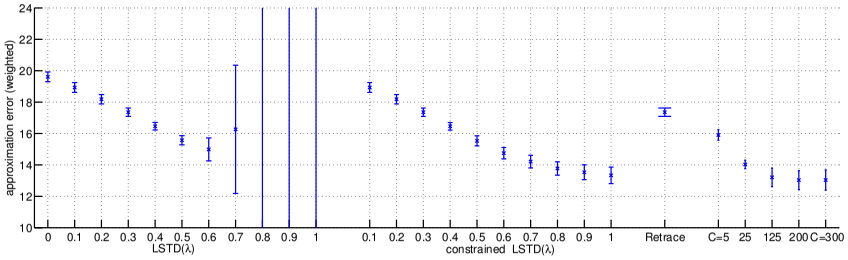


Figure 12: Compare the approximation errors of several LSTD algorithms.

Figure 12 is similar to Figure 9 and shows, for each algorithm, the mean and standard deviation of the approximation errors of 10 approximate value functions obtained at the end of 10 independent experimental runs (each of which consists of 6×10^5 effective iterations). The horizontal axis indicates the algorithms and their parameters. As can be seen from this figure, for constant small λ, LSTD(λ) performed well in this problem, and LSTD(0.3) and Retrace are comparable. For constant $\lambda > 0.7$, LSTD(λ) failed to give sensible results, and LSTD(0.7) started to show this unreliable behavior. This behavior of LSTD(λ) is related to what we observed in Figure 4(right) in the small toy problem, and it is, we think, due to

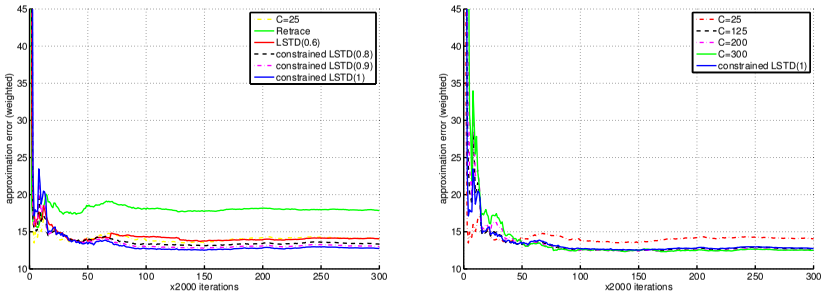


Figure 13: Compare the temporal behavior of several LSTD algorithms.

the high variance issue, which becomes more severe as λ gets larger. Constrained LSTD(λ) is much more reliable, and it did consistently well for all values of λ tested. LSTD with evolving λ also did well, and with $C > 125$, it achieved a slightly better approximation quality than constrained LSTD(1).

Figure 13 compares the temporal behavior of several algorithms during one experimental run. Plotted are the approximation errors calculated per 2000 iterations for each algorithm in the comparison. It can be seen that in this problem constrained LSTD(λ) did not suffer from large variances even with large λ values, and compared with constrained LSTD(1), the behavior of LSTD with evolving λ was also reasonable for large C.

Third Experiment: In this experiment we first compare the simple scaling scheme and Retrace, where both schemes now use an additional parameter $\beta \in [0, 1]$, as discussed in Examples 2.1-2.2. Recall that when $\beta = 1$, they reduce to the schemes that we already compared in the previous experiments. For both schemes, as β becomes smaller, we expect the approximation quality to drop but the variance to get smaller.

Plotted in Figure 14 for $C = 125, C = 200$ and Retrace are the results obtained from a single experimental run consisting of 3×10^5 effective iterations. As before the approximation errors were calculated per 2000 iterations. The results do show the expected bias-variance trade-off effects of the parameter β, during the initial period of the experimental run, although the effects on Retrace turned out to be smaller and hard to discern at the scale of the plot.

Next we test some of the variations on Retrace discussed in Example 2.2. Specifically, we consider (2.10)-(2.11) with parameters $\beta = 0.9$, $K \in \{1.5, 2.0, 2.5, 3.0\}$ and $C \in \{50, 125\}$. Plotted in Figure 15 are the results from one experimental run of 3×10^5 effective iterations. For comparison, the plots also show the behavior of Retrace and the simple scaling scheme with the same values of C during that run (these algorithms used the same $\beta = 0.9$). As expected and can be seen from the figure, the approximation quality improves with K and C. While the variances also tend to increase during the initial part of the run, the variants that truncate the importance sampling ratios by $K = 1.5$ performed comparably to Retrace initially, soon overtook Retrace and achieved better approximation quality.

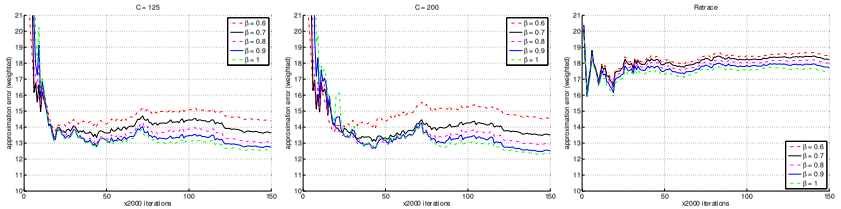


Figure 14: Compare the approximation error and temporal behavior of LSTD for different schemes of setting λ. From left to right: $C = 125$, $C = 200$, Retrace.

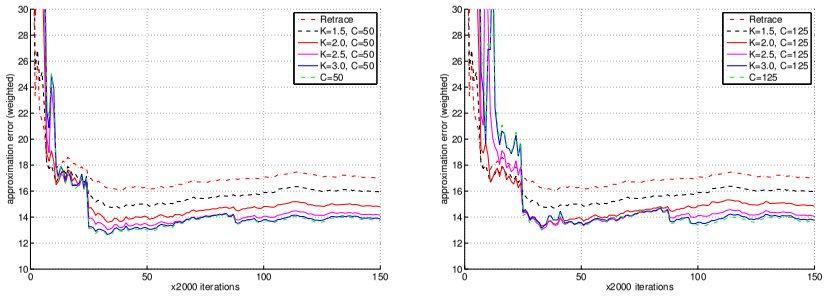


Figure 15: Compare the approximation error and temporal behavior for some variations on Retrace.

5. Conclusion

We developed in this paper a new scheme of setting the λ-parameters for off-policy TD learning, using the ideas of randomized stopping times and generalized Bellman equations for MDPs. Like the two recently proposed algorithms Retrace (Munos et al., 2016) and ABQ (Mahmood et al., 2017), our scheme keeps the traces bounded to reduce variances, but it is much more general and flexible. To study its theoretical properties, we analyzed the resulting state-trace process and established convergence and solution properties for the associated LSTD algorithm, and these results have prepared the ground for convergence analysis of the gradient-based implementation of our proposed scheme (Yu, 2017). In addition we did a preliminary numerical study. It showed that with the proposed scheme LSTD can outperform several existing off-policy LSTD algorithms. It also demonstrated that in order to achieve better bias-variance trade-offs in off-policy learning, it is helpful to have more flexibility in choosing the λ-parameters and to allow for large λ values. Future research is to conduct a more extensive numerical study of both least-squares based and gradient-based algorithms, with more versatile ways of using the memory states and λ-parameters, in off-policy learning applications.

Acknowledgments

An abridged version of this article appeared first at *The 30th Canadian Conference on Artificial Intelligence (CAI)*, May 2017. We thank several anonymous reviewers for CAI and JMLR for their helpful feedback. We also thank Dr. Ajin Joseph and Dr. Martha Steenstrup for reading parts of our manuscript and providing comments that helped improve our presentation. This research was supported by a grant from Alberta Innovates—Technology Futures.

Appendix A. Proof of Theorem 3.1

We prove Theorem 3.1 in this appendix. Recall from Section 3.1 that the generalized Bellman operator T associated with a randomized stopping time τ satisfies that $v_\pi = Tv_\pi$. By (3.2), the substochastic matrix $\tilde{P}$ in this affine operator T is given by

$$\tilde{P}_{ss'} = \mathbb{E}_s^\pi\big[\gamma_1^\tau \, \mathbb{1}(S_\tau = s')\big], \qquad s, s' \in \mathcal{S}. \tag{A.1}$$

We can extend $\tilde{P}$ to a transition matrix $\tilde{P}^e$ by adding an additional absorbing state Δ to the system, so that $\tilde{P}^e_{\Delta\Delta} = 1$ and

$$\tilde{P}^e_{s\Delta} = 1 - \textstyle\sum_{s' \in \mathcal{S}} \tilde{P}_{ss'} = 1 - \mathbb{E}_s^\pi\big[\gamma_1^\tau\big], \qquad s \in \mathcal{S}.$$

Both conclusions of Theorem 3.1 will follow immediately if we show that $I - \tilde{P}$ is invertible. Indeed, if $I - \tilde{P}$ is invertible, then $v = Tv$ has a unique solution, which must be v_π since v_π is always a solution of this equation. In addition, if $I - \tilde{P}$ is invertible, then since $\tilde{P}$ is a substochastic matrix, the spectral radius of $\tilde{P}$ must be less than 1. Consider adding to $\tilde{P}$ a small enough perturbation ϵM, where M is the matrix of all ones and ϵ is a sufficiently small positive number so that the spectral radius of $\tilde{P} + \epsilon M$ is less than 1. Applying (Seneta, 2006, Theorem 1.1) to the nonnegative primitive matrix $\tilde{P} + \epsilon M$, we have that $(\tilde{P} + \epsilon M)w < w$ where w is a positive eigenvector of the matrix $\tilde{P} + \epsilon M$ corresponding to a positive eigenvalue that is strictly less than 1. Consequently $\tilde{P}w \le (\tilde{P} + \epsilon M)w < w$, implying that $\tilde{P}$ is a linear contraction w.r.t. a weighted sup-norm (with weights w), which is the second conclusion of the theorem.

Hence, to prove Theorem 3.1, it suffices to show that the inverse $(I - \tilde{P})^{-1}$ exists, which is equivalent to that for the Markov chain on $\mathcal{S} \cup \{\Delta\}$ with transition matrix $\tilde{P}^e$, all the states in $\mathcal{S}$ are transient (see e.g., Puterman 1994, Appendix A.4).

We prove this by contradiction. Suppose it is not true, and let $\tilde{\mathcal{S}} \subset \mathcal{S}$ be a recurrent class of the Markov chain. Then for all $s \in \tilde{\mathcal{S}}$, $\tilde{P}^e_{ss'} = 0$ for $s' \notin \tilde{\mathcal{S}}$ (i.e., the submatrix of $\tilde{P}$ corresponding to $\tilde{\mathcal{S}}$ is a transition matrix). In particular, $\tilde{P}^e_{s\Delta} = 0$, so

$$\mathbb{E}_s^\pi\big[\gamma_1^\tau\big] = 1 - \tilde{P}^e_{s\Delta} = 1, \qquad \forall\, s \in \tilde{\mathcal{S}}, \tag{A.2}$$

implying that given $S_0 \in \tilde{\mathcal{S}}$, $\gamma_1^\tau = 1$ a.s. (since $\gamma_1^\tau \in [0, 1]$). Then by (A.1),

$$\tilde{P}_{ss'} = \mathbb{E}_s^\pi\big[\mathbb{1}(S_\tau = s')\big], \qquad \forall\, s, s' \in \tilde{\mathcal{S}}. \tag{A.3}$$

Observe from (A.2) that given $S_0 \in \tilde{\mathcal{S}}$, the event $\{\tau = \infty\}$ has zero probability. This is because under Condition 2.1(i), $\gamma_1^t \overset{a.s.}{\to} 0$ (as $t \to \infty$) and consequently $\gamma_1^\infty := \prod_{t=1}^\infty \gamma_t = 0$ a.s. Indeed, the existence of the inverse $(I - P\Gamma)^{-1} = \sum_{t=0}^\infty (P\Gamma)^t$ under Condition 2.1(i) implies $(P\Gamma)^t \to 0$ as $t \to \infty$. Since the sth entry of $(P\Gamma)^t \mathbf{1}$ equals $\mathbb{E}_s^\pi[\gamma_1^t]$, we have $\mathbb{E}_s^\pi[\gamma_1^t] \to 0$ as $t \to \infty$. As the nonnegative sequence $\{\gamma_1^t\}_{t \ge 1}$ is nonincreasing (since each $\gamma_t \in [0, 1]$), this implies, by Fatou's lemma (Dudley, 2002, Lemma 4.3.3), that $\gamma_1^t \overset{a.s.}{\to} 0$. Thus if $S_0 \in \tilde{\mathcal{S}}$, τ is almost surely finite.

We now consider a Markov chain $\{S_t\}$ with transition matrix P and $S_0 \in \tilde{\mathcal{S}}$. We will extract from it a Markov chain $\{\tilde{S}_k\}_{k \ge 0}$ with transition matrix $\tilde{P}^e$ on the recurrent class $\tilde{\mathcal{S}}$, by employing multiple stopping times. We will show $\gamma_1^\infty = 1$ a.s., contrary to the fact $\gamma_1^\infty = 0$ a.s. just discussed.

For ease of explanation, let us imagine that there is a device that if we give it a sequence of states $S_0, S_1, \ldots$ generated according to P, it will output the stopping decision at a random time τ that is exactly the randomized stopping time τ associated with the operator T. (Because τ is a randomized stopping time for $\{S_t\}$, a device that correctly implements τ does not affect the evolution of $\{S_t\}$, so we can give $\{S_t\}$ to the device as inputs.)

Let $\{S_t\}$ start from some state $S_0 = s \in \tilde{\mathcal{S}}$. We use the device just mentioned to generate the first randomized stopping time τ_1. We then reset the device so that it now "sees" the time-shifted process $\{S_{\tau_1 + t'} \mid t' \ge 0\}$ with the initial state being S_{τ_1}. We wait till the device makes another stopping decision, and we designate that time by τ_2. We repeat this procedure as soon as the device makes yet another stopping decision. This gives us a nondecreasing sequence $0 \le \tau_1 \le \tau_2 \le \cdots$.

Since $S_0 = s \in \tilde{\mathcal{S}}$, by what we proved earlier, τ_1 is almost surely finite. Let $\tilde{S}_1 = S_{\tau_1}$ and $\tilde{S}_0 = S_0$. The transition from $\tilde{S}_0$ to $\tilde{S}_1$ is according to the transition matrix $\tilde{P}^e$ by construction (cf. (A.3)). Since $\tilde{\mathcal{S}}$ is a recurrent class for $\tilde{P}^e$, we must have $\tilde{S}_1 \in \tilde{\mathcal{S}}$ almost surely. Then, repeating the same argument and using induction, we have that almost surely, for all $k \ge 1$, τ_k is defined and finite and $\tilde{S}_k := S_{\tau_k} \in \tilde{\mathcal{S}}$. Thus we obtain an infinite sequence $\{\tilde{S}_k\}$, which is a recurrent Markov chain on $\tilde{\mathcal{S}}$ with its transition matrix given by the corresponding submatrix of $\tilde{P}^e$.

Now in view of (A.2), almost surely,

$$\gamma_1^{\tau_1} = 1, \quad \gamma_{\tau_1+1}^{\tau_2} = 1, \quad \cdots, \quad \gamma_{\tau_k+1}^{\tau_{k+1}} = 1, \quad \cdots \tag{A.4}$$

(recall that if $\tau_k + 1 > \tau_{k+1}$, $\gamma_{\tau_k+1}^{\tau_{k+1}} = 1$ by definition). Consider first a simpler case of the randomized stopping time τ that defines T: for any initial state S_0, $\tau \ge 1$ a.s. Then, $\{\tau_k\}$ is strictly increasing, and by multiplying the variables in (A.4) together, we have $\gamma_1^\infty = 1$ a.s. For the general case of τ assumed in the theorem, $\tau = 0$ is possible, but $\mathbf{P}^\pi(\tau \ge 1 \mid S_0 = s) > 0$ for all states $s \in \mathcal{S}$. This means that the event of τ_k being the same for all k greater than some (random) $\bar{k}$ has probability zero. So $\{\tau_k\}$ must converge to $+\infty$ almost surely, and we again obtain, by multiplying the variables in (A.4) together, that $\gamma_1^\infty = 1$ a.s. This contradicts the fact proved earlier, namely, that for any initial state S_0, $\gamma_1^\infty = 0$ a.s. So the assumption of a recurrent class $\tilde{\mathcal{S}} \subset \mathcal{S}$ for $\tilde{P}^e$ must be false. This proves Theorem 3.1.

Appendix B. Oblique Projection Viewpoint and Error Bound for TD

In this appendix we first explain Scherrer's interpretation of TD solutions as oblique projections (Scherrer, 2010), and we then give approximation error bounds for TD similar to those

given by Yu and Bertsekas (2010), which do not rely on contraction properties. We will explain these properties of TD in the context of generalized Bellman operators discussed in this paper. Although this was not the framework used in (Scherrer, 2010; Yu and Bertsekas, 2010) and our setup here is more general than the one discussed in those previous papers, the arguments and reasoning are essentially the same.

B.1 Solutions of TD as Oblique Projections of the Value Function

Let us start with the projected Bellman equation associated with TD/LSTD that we noted in Remark 3.2:

$$Tv - v \perp_{\zeta_\mathcal{S}} \mathcal{L}_\phi, \quad v \in \mathcal{L}_\phi,$$

where T is a generalized Bellman operator with v_π as its unique fixed point, and $\mathcal{L}_\phi$ is the approximation subspace. We can write this equation equivalently as

$$v = \Pi_{\zeta_\mathcal{S}} T v, \tag{B.1}$$

where $\Pi_{\zeta_\mathcal{S}}$ denotes the projection onto the approximation subspace $\mathcal{L}_\phi$ with respect to the $\zeta_\mathcal{S}$-weighted Euclidean norm. In the subsequent derivations, we will not use the fact that $\zeta_\mathcal{S}$ is the invariant probability measure induced by the behavior policy on $\mathcal{S}$, so the analyses we give in this appendix apply to any weighted Euclidean norm $\|\cdot\|_{\zeta_\mathcal{S}}$.

Scherrer (2010) first realized that the solution of the projected Bellman equation (B.1) can be viewed as an *oblique projection of the value function* v_π on the approximation subspace $\mathcal{L}_\phi$. This viewpoint provides an intuitive geometric interpretation of the TD solution and explains conceptually the source of its approximation bias. Analytically, this view also gives tight bounds on the approximation bias, as we will elaborate later in Section B.2.

An oblique projection is defined by two nonorthogonal subspaces of equal dimensions: it is the projection onto the first subspace orthogonally to the second (Saad, 2003), as illustrated in Figure 16. More precisely, for any two n-dimensional subspaces $\mathcal{L}_1, \mathcal{L}_2$ of $\Re^N$ such that no vector in $\mathcal{L}_2$ is orthogonal to $\mathcal{L}_1$, there is an associated *oblique projection operator* $\Pi_{\mathcal{L}_1\mathcal{L}_2} : \Re^N \to \mathcal{L}_1$ defined by

$$\Pi_{\mathcal{L}_1\mathcal{L}_2}\, x \in \mathcal{L}_1, \qquad x - \Pi_{\mathcal{L}_1\mathcal{L}_2}\, x \perp \mathcal{L}_2, \qquad \forall\, x \in \Re^N. \tag{B.2}$$

If the two subspaces are the same: $\mathcal{L}_1 = \mathcal{L}_2$ or if x lies in $\mathcal{L}_1$, then the oblique projection $\Pi_{\mathcal{L}_1\mathcal{L}_2} x$ is the same as $\Pi_{\mathcal{L}_1} x$, the orthogonal projection of x onto $\mathcal{L}_1$. In general this need not be the case and $\Pi_{\mathcal{L}_1\mathcal{L}_2}\, x \neq \Pi_{\mathcal{L}_1} x$ typically (cf. Figure 16). A matrix representation of the projection operator $\Pi_{\mathcal{L}_1\mathcal{L}_2}$ is given by

$$\Pi_{\mathcal{L}_1\mathcal{L}_2} = \Phi_1 \big(\Phi_2^\top \Phi_1\big)^{-1} \Phi_2^\top, \tag{B.3}$$

where Φ_1 and Φ_2 are $N \times n$ matrices whose columns form a basis of $\mathcal{L}_1$ and $\mathcal{L}_2$, respectively (see Saad 2003, Chap. 1.12). For comparison, a matrix representation of the orthogonal projection operator $\Pi_{\mathcal{L}_1}$ is $\Pi_{\mathcal{L}_1} = \Phi_1 \big(\Phi_1^\top \Phi_1\big)^{-1} \Phi_1^\top$. (Below we will use the same notation for a projection operator and its matrix representations.)

Back to the projected Bellman equation (B.1), let us assume it has a unique solution v_{TD} and express v_{TD} in terms of v_π. We have $v_{\text{TD}} = \Pi_{\zeta_\mathcal{S}} T v_{\text{TD}}$. By Theorem 3.1, v_π is the

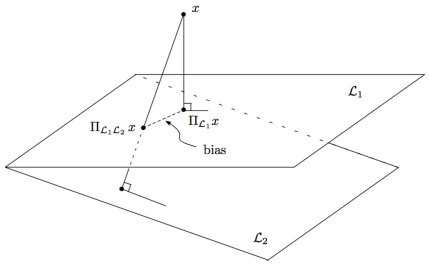


Figure 16: Oblique projection of x onto the subspace $\mathcal{L}_1$ orthogonally to the subspace $\mathcal{L}_2$.

unique solution of $v = Tv$. Recall that the generalized Bellman operator T is affine and can be expressed as $Tv = \tilde{r}_\pi + \tilde{P}_\pi v$ for some vector $\tilde{r}_\pi$ and substochastic matrix $\tilde{P}_\pi$. It follows that $\tilde{r}_\pi = (I - \tilde{P}_\pi)\, v_\pi$ and

$$\big(I - \Pi_{\zeta_\mathcal{S}} \tilde{P}_\pi\big)\, v_{\text{TD}} = \Pi_{\zeta_\mathcal{S}} \tilde{r}_\pi = \Pi_{\zeta_\mathcal{S}} \big(I - \tilde{P}_\pi\big)\, v_\pi. \tag{B.4}$$

Let the columns of Φ form a basis of the approximation subspace $\mathcal{L}_\phi$, and let D be the diagonal matrix with $\zeta_\mathcal{S}$ as its diagonal elements. Then a matrix representation of the projection operator $\Pi_{\zeta_\mathcal{S}}$ is $\Pi_{\zeta_\mathcal{S}} = \Phi \big(\Phi^\top D\, \Phi\big)^{-1} \Phi^\top D$. Using this representation of $\Pi_{\zeta_\mathcal{S}}$ and the fact $v_{\text{TD}} \in \mathcal{L}_\phi$, we obtain from (B.4) an expression of v_{TD} in terms of v_π:

$$v_{\text{TD}} = \Phi \big(\Phi^\top D (I - \tilde{P}_\pi)\, \Phi\big)^{-1} \Phi^\top D (I - \tilde{P}_\pi)\, v_\pi. \tag{B.5}$$

(Here the invertibility of the matrix $\Phi^\top D(I - \tilde{P}_\pi)\Phi$ is equivalent to our assumption that v_{TD} is the unique solution of (B.1).)

Let us compare the expression (B.5) with (B.3). We see that if the geometry on $\Re^N$ is determined by the usual Euclidean norm $\|\cdot\|_2$, then v_{TD} is an oblique projection of v_π for the two subspaces $\mathcal{L}_1 = \mathcal{L}_\phi$ and $\mathcal{L}_2 = \text{column-space}\big((I - \tilde{P}_\pi)^\top D \Phi\big)$.

Alternatively, consider the case where the geometry on $\Re^N$ is determined by some weighted Euclidean norm $\|\cdot\|_\xi$ with weights ξ (for example, $\xi = \zeta_\mathcal{S}$, which is one of the cases of interest in off-policy learning). In this case we can first scale each coordinate by the square root $\sqrt{\xi(s)}$ of its weight to reduce the case to that of the norm $\|\cdot\|_2$. Specifically, let $\perp_\xi$ denote orthogonality with respect to $\|\cdot\|_\xi$, and let Ξ denote the diagonal matrix that has ξ as its diagonal elements. For the linear mapping $h : v \mapsto \Xi^{1/2} v$, we have

$$\|v\|_\xi = \|h(v)\|_2, \qquad \text{and} \quad v_1 \perp_\xi v_2 \quad \Leftrightarrow \quad h(v_1) \perp h(v_2). \tag{B.6}$$

The second relation means that $\bar{v}$ is an oblique projection of v for two subspaces $\mathcal{L}_1, \mathcal{L}_2$ with respect to $\|\cdot\|_\xi$ (i.e., $\bar{v} \in \mathcal{L}_1$ and $v - \bar{v} \perp_\xi \mathcal{L}_2$), if and only if $h(\bar{v})$ is an oblique projection of $h(v)$ for the two subspaces $h(\mathcal{L}_1), h(\mathcal{L}_2)$ with respect to $\|\cdot\|_2$:

$$h(\bar{v}) \in h(\mathcal{L}_1), \qquad h(v) - h(\bar{v}) \perp h(\mathcal{L}_2).$$

With these facts in mind, we rewrite (B.5) equivalently as follows:

$$\Xi^{1/2}v_{\text{TD}} = \Xi^{1/2}\Phi \cdot \left(\Phi^\top D(I-\tilde{P}_\pi)\,\Xi^{-1}\Xi^{1/2}\cdot \Xi^{1/2}\Phi\right)^{-1}\cdot \Phi^\top D(I-\tilde{P}_\pi)\,\Xi^{-1}\Xi^{1/2}\cdot\Xi^{1/2}v_\pi$$
$$\implies \quad h(v_{\text{TD}}) = h(\Phi)\cdot\left(h(\Phi_2)^\top\cdot h(\Phi)\right)^{-1}h(\Phi_2)^\top\cdot h(v_\pi), \tag{B.7}$$

where

$$\Phi_2 = \Xi^{-1}(I-\tilde{P}_\pi)^\top D\Phi \tag{B.8}$$

and h applied to a matrix denotes the result of applying h to each column of that matrix. Comparing (B.7) with (B.3), we see that $h(v_{\text{TD}})$ is an oblique projection of $h(v_\pi)$ with respect to $\|\cdot\|_2$ for the two subspaces $h(\mathcal{L}_1)$, $h(\mathcal{L}_2)$, where

$$\mathcal{L}_1 = \mathcal{L}_\phi = \text{column-space}(\Phi), \qquad \mathcal{L}_2 = \text{column-space}(\Phi_2). \tag{B.9}$$

So based on the discussion earlier, with respect to the weighted Euclidean norm $\|\cdot\|_\xi$ on $\Re^N$, v_{TD} is an oblique projection of v_π for the two subspaces $\mathcal{L}_1, \mathcal{L}_2$.

Note that by (B.8), the second subspace $\mathcal{L}_2$ defining the above oblique projection is the image of the approximation subspace $\mathcal{L}_\phi$ under the linear transformation $\Xi^{-1}(I-\tilde{P}_\pi)^\top D$. Thus $\mathcal{L}_2$ *depends on the dynamics induced by the target policy as well as the generalized Bellman operator* T *that we choose*. Relating the oblique projection interpretation of v_{TD} to Figure 16, we can see where the approximation bias of TD, $v_{\text{TD}} - \Pi_\xi v_\pi$, comes from.

B.2 Approximation Error Bound

We now consider the approximation error of v_{TD} and use the oblique projection viewpoint to derive a sharp bound on the approximation bias $v_{\text{TD}} - \Pi_\xi v_\pi$. Before proceeding, however, let us first remind the reader that unless the norm $\|\cdot\|_{\zeta_\mathcal{S}}$ for the projection operator $\Pi_{\zeta_\mathcal{S}}$ is purposefully chosen, the composition of $\Pi_{\zeta_\mathcal{S}}$ with a generalized Bellman operator T is usually not a contraction, and thus error bounds for projected generalized Bellman equations usually cannot be obtained with contraction-based arguments. This is the case even for on-policy learning, as the following example shows.

Example B.1 (Non-contractive $\Pi_{\zeta_\mathcal{S}}T$) If the target policy π induces an irreducible Markov chain with invariant probability measure $\zeta_\mathcal{S}$, and if T is the Bellman operator for TD(λ) with a constant λ, then, as Tsitsiklis and Van Roy (1997) showed, $\Pi_{\zeta_\mathcal{S}}T$ is a contraction operator w.r.t. the weighted Euclidean norm $\|\cdot\|_{\zeta_\mathcal{S}}$. Consequently, the matrix $\Phi^\top D(\tilde{P}_\pi - I)\,\Phi$ associated with the TD(λ) algorithm is negative definite (Tsitsiklis and Van Roy, 1997). The derivation of the contraction property of $\Pi_{\zeta_\mathcal{S}}T$ in this case relies critically on the inequality $\zeta_\mathcal{S}^\top\tilde{P}_\pi < \zeta_\mathcal{S}^\top$. This inequality generally does not hold for the substochastic matrix $\tilde{P}_\pi$ in the generalized Bellman operator T, when λ is not constant. So, for non-constant λ, we can no longer expect $\Pi_{\zeta_\mathcal{S}}T$ to be a contraction or the matrix $\Phi^\top D(\tilde{P}_\pi - I)\,\Phi$ to be negative definite.

As an example, consider a simple two-state problem in which the system under the target policy π moves from one state to another in a cycle. Let $\pi^o = \pi$, let the discount factor γ be a constant, and let λ be a function of states with $\lambda(1) = 0$, $\lambda(2) = 1$. Then $\zeta_\mathcal{S}^\top = (0.5, 0.5)$ and $\tilde{P}_\pi = \begin{pmatrix}\gamma^2 & 0\\ \gamma & 0\end{pmatrix}$. For γ near 1, e.g., $\gamma = 0.95$, and for Φ as given below,

we can calculate the $\zeta_\mathcal{S}$-weighted norm of $\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi$ and the matrix associated with TD(λ):

$$\Phi = \begin{pmatrix}3 & 1\\ 1 & 1\end{pmatrix}, \qquad \left\|\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi\right\|_{\zeta_\mathcal{S}} = \left\|\tilde{P}_\pi\right\|_{\zeta_\mathcal{S}} \approx 1.31 > 1,$$
$$\Phi^\top D(\tilde{P}_\pi - I)\,\Phi = \begin{pmatrix}0.4862 & -0.1713\\ 0.7787 & -0.0738\end{pmatrix}.$$

The latter matrix is not negative definite; in fact, its eigenvalues have positive real parts, so it is not even a Hurwitz matrix (TD(λ) can diverge in this case). In the above, we have $\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi = \tilde{P}_\pi$, and while $\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi$ is not a contraction w.r.t. $\|\cdot\|_{\zeta_\mathcal{S}}$, it is still a contraction w.r.t. some matrix norm. If we now let $\Phi = (3,1)^\top$ instead, then the spectral radius of the matrix $\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi$ comes out as $\sigma(\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi) \approx 1.10 > 1$, so $\Pi_{\zeta_\mathcal{S}}\tilde{P}_\pi$ (and hence $\Pi_{\zeta_\mathcal{S}}T$) cannot be a contraction w.r.t. any matrix norm. ■

We now proceed to bound the bias term $v_{\text{TD}} - \Pi_\xi v_\pi$ relative to $\|v_\pi - \Pi_\xi v_\pi\|_\xi$, the distance between v_π and the approximation subspace measured with respect to $\|\cdot\|_\xi$. It is more transparent to derive the bound for the case of a general oblique projection operator $\Pi_{\mathcal{L}_1\mathcal{L}_2}$ with respect to the usual Euclidean norm $\|\cdot\|_2$, so let us do that first and then use the linear transformation $h(\cdot)$ to translate the result to TD, as we did earlier in the preceding subsection.

We bound the bias $\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x - \Pi_{\mathcal{L}_1}x\|_2$ relative to $\|x - \Pi_{\mathcal{L}_1}x\|_2$, by calculating

$$\kappa := \sup_{x\in\Re^N}\frac{\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x - \Pi_{\mathcal{L}_1}x\|_2}{\|x-\Pi_{\mathcal{L}_1}x\|_2} = \sup_{x\in\Re^N}\frac{\|\Pi_{\mathcal{L}_1\mathcal{L}_2}(x-\Pi_{\mathcal{L}_1}x)\|_2}{\|x-\Pi_{\mathcal{L}_1}x\|_2} \tag{B.10}$$

(where we treat $0/0 = 0$). This constant κ depends on the two subspaces $\mathcal{L}_1$, $\mathcal{L}_2$, and reflects the "angle" between them. It has several equivalent expressions, e.g.,

$$\kappa = \sup_{x\perp\mathcal{L}_1,\,\|x\|_2=1}\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x\|_2 = \|\Pi_{\mathcal{L}_1\mathcal{L}_2}(I-\Pi_{\mathcal{L}_1})\|_2, \tag{B.11}$$

or with $\sigma(F)$ denoting the spectral radius of a square matrix F,

$$\kappa = \sqrt{\sigma\left(\Pi_{\mathcal{L}_1\mathcal{L}_2}(I-\Pi_{\mathcal{L}_1})\cdot(I-\Pi_{\mathcal{L}_1})^\top\Pi_{\mathcal{L}_1\mathcal{L}_2}^\top\right)} = \sqrt{\sigma\left(\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1}\right)}. \tag{B.12}$$

(After the definition of κ, each expression of κ in (B.10)-(B.12) follows from the preceding one; in particular, for the last expression in (B.12), we used the fact that $\Pi_{\mathcal{L}_1}^\top = \Pi_{\mathcal{L}_1}, \Pi_{\mathcal{L}_1\mathcal{L}_2}^\top = \Pi_{\mathcal{L}_2\mathcal{L}_1}$, and $\Pi_{\mathcal{L}_1}\Pi_{\mathcal{L}_2\mathcal{L}_1} = \Pi_{\mathcal{L}_1}$.)

We can express κ in terms of the spectral radius of an $n\times n$ matrix, similarly to what was done in (Yu and Bertsekas, 2010). In particular, we take the last expression of κ in (B.12) and rewrite the symmetric matrix in that expression using the matrix representations of the projection operators as follows:

$$\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1} = \Phi_1(\Phi_2^\top\Phi_1)^{-1}\Phi_2^\top\cdot\Phi_2(\Phi_1^\top\Phi_2)^{-1}\Phi_1^\top - \Phi_1(\Phi_1^\top\Phi_1)^{-1}\Phi_1^\top.$$

By a result in matrix theory (Horn and Johnson, 1985, Theorem 1.3.20), for any $N\times n$ matrix F_1 and $n\times N$ matrix F_2, $\sigma(F_1F_2) = \sigma(F_2F_1)$. Applying this result to the preceding

expression with $F_1 = \Phi_1$, we have that $\sigma\big(\Pi_{\mathcal{L}_1\mathcal{L}_2}\Pi_{\mathcal{L}_2\mathcal{L}_1} - \Pi_{\mathcal{L}_1}\big)$ is equal to the spectral radius of the matrix

$$(\Phi_2^\top\Phi_1)^{-1}(\Phi_2^\top\Phi_2)(\Phi_1^\top\Phi_2)^{-1}(\Phi_1^\top\Phi_1) - I.$$

Combing this with (B.12), we obtain that

$$\kappa^2 = \sigma(F) - 1, \qquad \text{where} \quad F = (\Phi_2^\top\Phi_1)^{-1}(\Phi_2^\top\Phi_2)(\Phi_1^\top\Phi_2)^{-1}(\Phi_1^\top\Phi_1). \tag{B.13}$$

Thus, for the above κ, the bound below holds for all $x \in \Re^N$ and with equality attained at some x:

$$\|\Pi_{\mathcal{L}_1\mathcal{L}_2}x - \Pi_{\mathcal{L}_1}x\|_2 \le \kappa\,\|x - \Pi_{\mathcal{L}_1}x\|_2. \tag{B.14}$$

We now translate the result (B.13)-(B.14) to our TD context. We want to bound the relative bias $\big\|v_{\text{TD}} - \Pi_\xi v_\pi\big\|_\xi / \big\|v_\pi - \Pi_\xi v_\pi\big\|_\xi$. As discussed earlier in Section B.1, we can replace $\|\cdot\|_\xi$ with $\|\cdot\|_2$ by using the linear transformation $h(\cdot)$ to scale the coordinates. In particular, by the two relations given in (B.6),

$$\frac{\big\|v_{\text{TD}} - \Pi_\xi v_\pi\big\|_\xi}{\big\|v_\pi - \Pi_\xi v_\pi\big\|_\xi} = \frac{\big\|h(v_{\text{TD}}) - \Pi\,h(v_\pi)\big\|_2}{\big\|h(v_\pi) - \Pi\,h(v_\pi)\big\|_2}, \tag{B.15}$$

where Π on the r.h.s. stands for the orthogonal projection onto $h(\mathcal{L}_\phi)$ with respect to $\|\cdot\|_2$. As shown by (B.7), $h(v_{\text{TD}})$ is an oblique projection of $h(v_\pi)$ for the two subspaces $h(\mathcal{L}_\phi)$ and $h(\mathcal{L}_2)$, where $\mathcal{L}_2$ is given by (B.8)-(B.9). According to (B.13), the constant κ for this oblique projection is $\sqrt{\sigma(F) - 1}$ where, if we take $\Phi_1 = \Phi$ and Φ_2 as defined by (B.8), F is now given by the expression in (B.13) with $h(\Phi_1), h(\Phi_2)$ in place of Φ_1, Φ_2, respectively. Thus, by (B.14) and (B.15) we obtain that

$$\big\|v_{\text{TD}} - \Pi_\xi v_\pi\big\|_\xi \le \kappa\,\big\|v_\pi - \Pi_\xi v_\pi\big\|_\xi \qquad \text{for} \quad \kappa = \sqrt{\sigma(F) - 1}, \tag{B.16}$$

where F is an $n \times n$ matrix given by

$$F = \big[h(\Phi_2)^\top h(\Phi_1)\big]^{-1} \cdot \big[h(\Phi_2)^\top h(\Phi_2)\big] \cdot \big[h(\Phi_1)^\top h(\Phi_2)\big]^{-1} \cdot \big[h(\Phi_1)^\top h(\Phi_1)\big] \tag{B.17}$$

for

$$\Phi_1 = \Phi, \qquad \Phi_2 = \Xi^{-1}(I - \tilde{P}_\pi)^\top D\Phi, \tag{B.18}$$

or more explicitly, after substituting the expressions of $h(\Phi_1)$ and $h(\Phi_2)$ in the formula of F and removing h, we have

$$F = \big(\Psi^\top\Phi\big)^{-1}\big(\Psi^\top\Xi^{-1}\Psi\big)\big(\Phi^\top\Psi\big)^{-1}\big(\Phi^\top\Xi\Phi\big), \quad \text{where} \quad \Psi = (I - \tilde{P}_\pi)^\top D\Phi.$$

Note that by the definition of κ, the bound (B.16) is a worst-case bound that depends only on the two subspaces involved in the oblique projection operator. In other words, given the approximation subspace $\mathcal{L}_\phi$, the dynamics described by $\tilde{P}_\pi$, the projection norm $\|\cdot\|_{\zeta_\mathcal{S}}$ and the norm $\|\cdot\|_\xi$ for measuring the approximation quality, the bound (B.16) is attained by a worst-case choice of the rewards $\tilde{r}_\pi$ for the target policy. In this sense, the bound (B.16) is tight.

References

D. P. Bertsekas and J. N. Tsitsiklis. *Neuro-Dynamic Programming.* Athena Scientific, Belmont, MA, 1996.

V. S. Borkar. *Stochastic Approximation: A Dynamic Viewpoint.* Cambridge University Press, Cambridge, 2008.

J. A. Boyan. Least-squares temporal difference learning. In *The 16th Int. Conf. Machine Learning (ICML)*, 1999.

B. Dai, A. Shaw, L. Li, L. Xiao, N. He, Z. Liu, J. Chen, and L. Song. SBEED: Convergent reinforcement learning with nonlinear function approximation. In *The 35th Int. Conf. Machine Learning (ICML)*, 2018.

C. Dann, G. Neumann, and J. Peters. Policy evaluation with temporal differences: A survey and comparison. *Journal of Machine Learning Res.*, 15:809–883, 2014.

R. M. Dudley. *Real Analysis and Probability.* Cambridge University Press, Cambridge, 2002.

M. Geist and B. Scherrer. Off-policy learning with eligibility traces: A survey. *Journal of Machine Learning Res.*, 15:289–333, 2014.

P. W. Glynn and D. L. Iglehart. Importance sampling for stochastic simulations. *Management Science*, 35:1367–1392, 1989.

R. A. Horn and C. R. Johnson. *Matrix Analysis.* Cambridge University Press, Cambridge, UK, 1985.

P. Karmakar and S. Bhatnagar. Two timescale stochastic approximation with controlled Markov noise and off-policy temporal difference learning. *Math. Oper. Res.*, 43(1):130–151, 2018.

V. R. Konda. *Actor-Critic Algorithms.* PhD thesis, MIT, 2002.

H. J. Kushner and G. G. Yin. *Stochastic Approximation and Recursive Algorithms and Applications.* Springer-Verlag, New York, 2nd edition, 2003.

B. Liu, J. Liu, M. Ghavamzadeh, S. Mahadevan, and M. Petrik. Finite-sample analysis of proximal gradient TD algorithms. In *The 31st Conf. Uncertainty in Artificial Intelligence (UAI)*, 2015.

H. R. Maei. *Gradient Temporal-Difference Learning Algorithms.* PhD thesis, University of Alberta, 2011.

S. Mahadevan, B. Liu, P. Thomas, W. Dabney, S. Giguere, N. Jacek, I. Gemp, and J. Liu. Proximal reinforcement learning: A new theory of sequential decision making in primal-dual spaces, 2014. arXiv:1405.6757.

A. R. Mahmood, H. van Hasselt, and R. S. Sutton. Weighted importance sampling for off-policy learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS) 27*, 2014.

A. R. Mahmood, H. Yu, and R. S. Sutton. Multi-step off-policy learning without importance-sampling ratios, 2017. arXiv:1702.03006.

S. Mannor, R. Rubinstein, and Y. Gat. The cross entropy method for fast policy search. In *The 20th Int. Conf. Machine Learning (ICML)*, 2003.

S. Meyn. Ergodic theorems for discrete time stochastic systems using a stochastic Lyapunov function. *SIAM J. Control Optim.*, 27:1409–1439, 1989.

S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge University Press, Cambridge, 2nd edition, 2009.

R. Munos, T. Stepleton, A. Harutyunyan, and M. G. Bellemare. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NIPS) 29*, 2016.

E. Nummelin. *General Irreducible Markov Chains and Non-Negative Operators*. Cambridge University Press, Cambridge, 1984.

D. Precup, R. S. Sutton, and S. Singh. Eligibility traces for off-policy policy evaluation. In *The 17th Int. Conf. Machine Learning (ICML)*, 2000.

D. Precup, R. S. Sutton, and S. Dasgupta. Off-policy temporal-difference learning with function approximation. In *The 18th Int. Conf. Machine Learning (ICML)*, 2001.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, New York, 1994.

R. S. Randhawa and S. Juneja. Combining importance sampling and temporal difference control variates to simulate Markov chains. *ACM Trans. Modeling and Computer Simulation*, 14(1):1–30, 2004.

W. Rudin. *Real and Complex Analysis*. McGraw-Hill, New York, 1966.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, Philadelphia, 2nd edition, 2003.

M. Schäl and W. Sudderth. Stationary policies and Markov policies in Borel dynamic programming. *Probability Theory and Related Fields*, 74:91–111, 1987.

B. Scherrer. Should one compute the temporal difference fix point or minimize the Bellman residual? The unified oblique projection view. In *The 27th Int. Conf. Machine Learning (ICML)*, 2010.

E. Seneta. *Non-negative Matrices and Markov Chains*. Springer, New York, 2006.

R. S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44, 1988.

R. S. Sutton. TD models: Modeling the world at a mixture of time scales. In *The 12th Int. Conf. Machine Learning (ICML)*, 1995.

R. S. Sutton. The grand challenge of predictive empirical abstract knowledge. In *IJCAI Workshop on Grand Challenges for Reasoning from Experiences*, 2009.

R. S. Sutton and A. G. Barto. *Reinforcement Learning*. MIT Press, Cambridge, MA, 1998.

R. S. Sutton, C. Szepesvári, and H. Maei. A convergent $O(n)$ algorithm for off-policy temporal-difference learning with linear function approximation. In *Advances in Neural Information Processing Systems (NIPS) 21*, 2008.

R. S. Sutton, H. R. Maei, D. Precup, S. Bhatnagar, D. Silver, C. Szepesvári, and E. Wiewiora. Fast gradient-descent methods for temporal-difference learning with linear function approximation. In *The 26th Int. Conf. Machine Learning (ICML)*, 2009.

R. S. Sutton, A. R. Mahmood, and M. White. An emphatic approach to the problem of off-policy temporal-difference learning. *Journal of Machine Learning Res.*, 17(73):1–29, 2016.

J. N. Tsitsiklis. Asynchronous stochastic approximation and Q-learning. *Machine Learning*, 16:185–202, 1994.

J. N. Tsitsiklis and B. Van Roy. An analysis of temporal-difference learning with function approximation. *IEEE Trans. Autom. Control*, 42(5):674–690, 1997.

T. Ueno, S. Maeda, M. Kawanabe, and S. Ishii. Generalized TD learning. *Journal of Machine Learning Res.*, 12:1977–2020, 2011.

C. J. C. H. Watkins. *Learning from Delayed Rewards*. PhD thesis, Cambridge Univ., England, 1989.

H. Yu. Least squares temporal difference methods: An analysis under general conditions. *SIAM J. Control Optim.*, 50:3310–3343, 2012.

H. Yu. On convergence of emphatic temporal-difference learning. In *The 28th Ann. Conf. Learning Theory (COLT)*, 2015. A longer version at arXiv:1506.02582.

H. Yu. Some simulation results for emphatic temporal-difference learning algorithms, 2016a. arxiv:1605.02099.

H. Yu. Weak convergence properties of constrained emphatic temporal-difference learning with constant and slowly diminishing stepsize. *Journal of Machine Learning Res.*, 17 (220):1–58, 2016b.

H. Yu. On convergence of some gradient-based temporal-differences algorithms for off-policy learning, 2017. arxiv:1712.09652.

H. Yu and D. P. Bertsekas. Error bounds for approximations from projected linear equations. *Math. Oper. Res.*, 35(2):306–329, 2010.

H. Yu and D. P. Bertsekas. Weighted Bellman equations and their applications in approximate dynamic programming. LIDS Technical Report 2876, MIT, 2012.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 9/17; Revised 8/18; Published 9/18

Design and Analysis of the NIPS 2016 Review Process

Nihar B. Shah* NIHARS@CS.CMU.EDU
Machine Learning Department and Computer Science Department
Carnegie Mellon University
Pittsburgh, PA 15213, USA

Behzad Tabibian* ME@BTABIBIAN.COM
Max Planck Institute for Intelligent Systems, and
Max Planck Institute for Software Systems
Tübingen, Germany

Krikamol Muandet KRIKAMOL@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems
Tübingen, Germany

Isabelle Guyon GUYON@CHALEARN.ORG
Universite Paris-Saclay, France, and
ChaLearn, California

Ulrike von Luxburg LUXBURG@INFORMATIK.UNI-TUEBINGEN.DE
University of Tübingen, and
Max Planck Institute for Intelligent Systems,
Tübingen, Germany

Editor: Neil Lawrence

Abstract

Neural Information Processing Systems (NIPS) is a top-tier annual conference in machine learning. The 2016 edition of the conference comprised more than 2,400 paper submissions, 3,000 reviewers, and 8,000 attendees. This represents a growth of nearly 40% in terms of submissions, 96% in terms of reviewers, and over 100% in terms of attendees as compared to the previous year. The massive scale as well as rapid growth of the conference calls for a thorough quality assessment of the peer-review process and novel means of improvement. In this paper, we analyze several aspects of the data collected during the review process, including an experiment investigating the efficacy of collecting ordinal rankings from reviewers. We make a number of key observations, provide suggestions that may be useful for subsequent conferences, and discuss open problems towards the goal of improving peer review.

Keywords: Peer review, post hoc analysis, NIPS, consistency, ordinal

∗. Authors contributed equally.

. Nihar Shah, Behzad Tabibian and Krikamol Muandet performed most of the data analysis reported in this paper. Behzad Tabibian and Krikamol Muandet were also the workflow team of NIPS 2016 and were responsible for all the programs, scripts and CMT-related issues during the review process. Isabelle Guyon and Ulrike von Luxburg were the program chairs of NIPS 2016.

1. Introduction

The review process for NIPS 2016 involved 2,425 papers submitted by 5,756 authors, 100 area chairs, and 3,242 active reviewers submitting 13,674 reviews in total. Designing a review process as fair as possible at this scale was a challenge. In order to scale, all parts of the process have to be as decentralized as possible. Just to get a feeling, if the two program chairs were supposed to take final decisions just for the 5% most challenging submissions, which means that they would have to read and decide on 150 papers — this is the scale of a whole conference such as COLT. Furthermore, the complexity of the logistics and software to manage the review process is rather high already. A controlled experiment (Lawrence and Cortes, 2014) from NIPS 2014 has shown that there is a high disagreement in the reviews. Hence the primary goal must be to keep bias and variance of the decisions as small as possible.

In this paper, we present an analysis of many aspects of the data collected throughout the review phase of the NIPS 2016 conference, performed subsequent to the completion of the review process. Our goal in this analysis is to examine various aspects of the data collected from the peer-review process to check for any systematic issues. Before delving into the details, we note the following limitations of this analysis:

- There is no ground truth ranking of the papers or knowledge of the set of papers which should ideally have been accepted.
- The analysis is post hoc, unlike the controlled experiment from NIPS 2014 (Lawrence and Cortes, 2014).
- The analysis primarily evaluates the ratings and rankings provided by reviewers, and does not study the textual comments provided by the reviewers.

The analysis is used to obtain insights into the peer-review process, usable suggestions for subsequent conferences, and important open problems towards improving peer-review in academia.

Here is a summary of our findings:

(i) there are very few positive bids by reviewers and area chairs (Section 3.1);

(ii) graph-theoretic techniques can be used to ensure a good reviewer assignment (Section 3.2);

(iii) there is significant miscalibration with respect to the rating scale (Section 3.3);

(iv) review scores provided by invited and volunteer reviewers have comparable biases and variance, and junior reviewers report a lower confidence (Section 3.4);

(v) there is little change in reviewer scores after rebuttals (Section 3.5);

(vi) there is no observable bias towards any research area in accepted papers (Section 3.6);

(vii) there is lower disagreement among reviewers in NIPS 2016 as compared to NIPS 2015 (Section 3.7);

(viii) a significant fraction of scores provided by the reviewers are tied, and ordinal rankings can ameliorate this issue (Section 3.8);

(ix) there are some inconsistencies in the reviews and these can be identified in an automated manner using ordinal rankings (Section 3.9).

We describe the review procedure followed at NIPS 2016 in Section 2. We present an elaborate description of the analysis and the results in Section 3. Alongside each analysis, we present a set of key observations, action items for future conferences, and some open problems that arise out of the analysis. We conclude the paper with a discussion in Section 4.

2. Review procedure

In this section, we present an overview of the design of the review process at NIPS 2016.

2.1 Selecting area chairs and reviewers

Area Chairs (ACs) are the backbone of the NIPS reviewing process. Their role is similar to that of an associate editor for a journal. Each AC typically handles 20-30 submissions, so with an estimated number of submissions between 2000 and 3000, we needed to recruit about 100 area chairs. As it is impossible to intimately know all the diverse research areas covered by NIPS, we came up with the following procedure. We asked the NIPS Board and all the ACs of NIPS from the past two years to nominate potential ACs for this year. In this manner, we covered the entire variety of NIPS topics and obtained qualified suggestions. We obtained around 350 suggestions. We asked the NIPS Board to go through the list of suggested ACs and vote in favor of suggested ACs. We also accounted for the distribution of subject areas of submitted papers of the previous year's NIPS conference. Combining all these inputs, we compiled a final list of ACs: by the end of January we had recruited exactly 100 ACs. In a subsequent step, we formed "buddy pairs" among the ACs. Based on the ACs preferences, each AC got assigned a buddy AC. We revisit the role of buddy pairs in more detail later.

The process of **recruiting reviewers** is time consuming, it essentially went on from January until the submission deadline at end of May. A significant departure from the review processes of NIPS from earlier years, this time we had two kinds of reviewers, "invited senior reviewers" (Pool 1) and "volunteer reviewers" (Pool 2):

- **Pool 1, invited senior reviewers:** We asked all ACs to suggest at least 30 reviewers who have completed their PhDs (however, this requirement was not strictly observed by all ACs). We then also asked all confirmed reviewers to "clone themselves" by inviting at least one researcher with a similar research background and with at least as good a qualification as themselves.

- **Pool 2, volunteer author-reviewers:** The rapid growth in the number of submissions at NIPS poses the formidable challenge of accordingly scaling the number of reviewers. An obvious means to achieve this objective is to ask authors to become reviewers as well. This idea has been used in the past, for example, to evaluate NSF grant proposals (Mervis, 2014) or to allocate telescope time (Merrifield and Saari, 2009). In order to implement this idea, without constraining unwilling authors, we requested authors to volunteer during the submission process by naming at least one author per paper as volunteer reviewers.

The area chairs were aware of the respective pools to which each of their reviewers belonged. The number of reviewers that we eventually ended up with writing reviews are as follows:

Number of reviewers	Senior researchers / faculty	Junior researchers / postdocs	PhD students	Not Specified	Total
Pool 1: Invited	1236	566	255	7	2064
Pool 2: Volunteer	143	206	827	2	1178

The total number of reviews in each category are as follows:

Number of reviews	Senior researchers / faculty	Junior researchers / postdocs	PhD students	Not Specified	Total
Pool 1: Invited	5759	2559	888	38	9244
Pool 2: Volunteer	576	795	3050	9	4430

2.2 Assignment of papers to reviewers and area chairs

The assignment of papers to area chairs was made in the following manner. Prior to the review process, the ACs and reviewers were allowed to see the list of submitted papers and "bid" whether they were interested or disinterested in handling/reviewing any paper. For any paper, an AC/reviewer could either indicate "Not Willing" or "In-a-pinch" – which we count as negative bids, or indicate "Willing" or "Eager" – which we count as positive bids, or choose to not bid for that paper. The Toronto paper matching system or TPMS was then employed to compute an affinity score for every AC (and reviewer) with every submitted paper based on the content of the paper and the academic profile of the AC or reviewer. In addition, every AC/reviewer as well as the submitter of every paper was asked to select a set of most relevant subject areas, and these subject areas were also employed to compute a similarity between every AC/reviewer and every paper.

Based on the similarity scores and bids, an overall similarity score is computed for every {paper, AC} and every {paper, reviewer} pair: score $= 2^b(s_{\text{affinity}} + s_{\text{subject}})/2$, where $s_{\text{affinity}} \in [0, 1]$ is the affinity score obtained from TPMS, $s_{\text{subject}} \in [0, 1]$ is the score obtained by comparing the subject areas of the paper and the subject areas selected by the AC or reviewer, and $b \in [-1, 1]$ is the bidding score provided by the AC or reviewer. Based on these overall similarity scores, a preliminary paper assignment to ACs was then produced in an automated manner using the TPMS assignment algorithm (Charlin and Zemel, 2013). The ACs were given a provision to decline handling certain papers for various reasons such as conflicts of interest. These papers were re-assigned manually by the program chairs.

The AC of each paper was responsible to first assign one senior, highly qualified reviewer manually. Two more invited reviewers from pool 1 and three volunteer reviewers from pool 2 were then assigned automatically to each paper using the same procedure as described above.

The ACs were asked to verify whether each of their assigned papers had at least 3 highly competent reviewers; the ACs could manually change reviewer assignments to ensure that this is the case. During the decision process, additional emergency reviewers were invited to provide complementary reviews if some of the reviewers had not turned in their reviews or if no consensus was reached among the selected reviewers.

2.3 Review criteria and scores

We completely changed the scoring method this year. In previous years, NIPS papers were rated using a single score between 1 and 10. A single score alone did not allow reviewers to give a differentiated quantitative appreciation on various aspects of paper quality. Furthermore, the role of the ACs was implicitly to combine the decisions of the reviewers (late integration) rather than combining the reviews to make the final decision (early integration). Introducing multiple scores allowed us to better separate the roles: the reviewers were in charge of evaluating the papers; the ACs were in charge of making decisions based on all the evaluations. Furthermore the multiple specialized scores allowed the ACs to guide reviewers to focus discussions on "facts" rather than "opinion" in the discussion phase. We asked reviewers to provide a separate score for each of the following four criteria:

- Technical quality,
- Novelty/originality,
- Potential impact or usefulness,
- Clarity and presentation.

The scores were on a scale of 1 to 5, with the following rubric provided to the reviewers:

5 = Award level (1/1000 submissions),

4 = Oral level (top 3% submissions),

3 = Poster level (top 30% submissions),

2 = Sub-standard for NIPS,

1 = Low or very low.

The scoring guidelines also reflect the hierarchy of the papers: the conference selects the top few papers for awards, the next best accepted papers are presented as oral presentations, and the remaining accepted papers are presented as posters at the conference. The scores provided by reviewers had to be complemented by justifications in designated text boxes. We also asked the reviewers to flag "fatal flaws" in the papers they reviewed. For each paper, we also asked the reviewers to declare their overall "level of confidence":

3 = Expert (read the paper in detail, know the area, quite certain of opinion),

2 = Confident (read it all, understood it all reasonably well),

1 = Less confident (might not have understood significant parts).

2.4 Discussions and rebuttals

Once most reviews were in, authors had the opportunity to look at the reviews and write a rebuttal. One section of the rebuttal was revealed to all the reviewers of the paper, and a second section was private and visible only to the ACs. Some reviews were still missing at

this point, but it would not have helped to delay the rebuttal deadline as the missing reviews trickled in only slowly. Subsequently, ACs and reviewers engaged in discussions about the pros and cons of the submitted papers. To support the ACs, we sent individual reports to all area chairs to flag papers whose reviews were of too low confidence, too high variance or where reviews were still missing. In many cases, area chairs recruited additional emergency reviewers to increase the overall quality of the decisions.

2.5 Decision procedure

The decision procedure involved making an acceptance or rejection decision for each paper, and furthermore, to select a subset of (the best) accepted papers for oral presentation.

We introduced a decentralized decision process based on pairs of ACs ("buddy pairs"). Each AC got assigned one buddy AC. Each pair of buddy ACs was responsible for all papers in their joint bag and made the accept/reject decisions jointly, following guidelines given by the program chairs. Difficult cases were taken to the program chairs, which included cases involving conflicts of interest and plagiarism. In order to harmonize decisions across buddy pairs, all area chairs had access to various statistics and histograms over the set of their papers and the set of all submitted papers. To decide which accepted paper would get an oral presentation, each buddy pair was asked to champion one or two papers from their joint bag as a candidate for an oral presentation. The final selection was then made by the program chairs, with the goals of exhibiting the diversity of NIPS papers and exposing the community with novel and thought-provoking ideas. In the end, 568 papers got accepted to the conference, and 45 of these papers were selected for oral presentations.

Like previous years, we adopted a "double blind" review policy. That is, the author(s) of each paper did not get to know the identity of the reviewers and vice versa throughout the review process. ACs got to know the identity of the reviewers and the author(s) for the papers under their responsibility. During the discussion phase, reviewers who reviewed the same papers got to know each other's identity. Lastly, PCs and program managers had access to all information about the submissions, the ACs, the reviewers, and the authors.

2.6 Experimental ordinal reviews

In the main NIPS 2016 review process, we elicited only cardinal scores from the reviewers – one score in 1 to 5 for each of four criteria. Subsequent to the review process, we then requested each reviewer to also provide a total ranking of the papers that they reviewed. We received rankings from a total of 2189 reviewers. Note that the collection of ordinal data was performed subsequent to the normal review submission but before release of the final decisions. The ordinal data was not used as a part of the decision procedure in the conference.

3. Detailed analysis

In this section, we present details of our analyses of the review data and the associated results. Each subsection contains one analysis and concludes with a summary that highlights the key observations, concrete action items for future conferences, and open problems that arise from the analysis.

The results are computed for a snapshot of reviews at the end of the review process when the acceptance decisions were made. This choice does not affect our results since there was very little change in the scores provided by reviewers across different time instants. All t-tests conducted correspond to two-sample t-tests with unequal variances. All mentions of p-values correspond to two-sided tail probabilities. All mentions of statistical significance correspond to a p-value threshold of 0.01 (we also provide the exact p-values alongside). Multiple testing is accounted for using the Bonferroni correction. The effect sizes refer to Cohen's d. Wherever applicable, the error bars in the figures represent 95% confidence intervals.

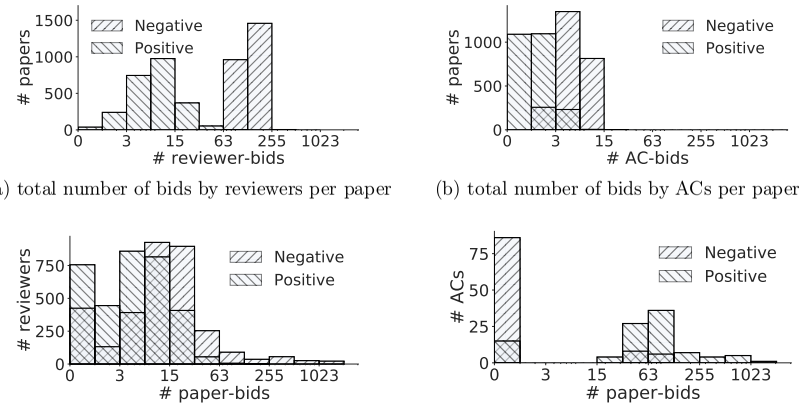
Wherever applicable, we also perform our analyses on a subset of the submitted papers which we term as the top 2k papers. The top 2k papers comprise all of the 568 accepted papers, and an equal number (568) of the rejected papers. The 568 rejected papers are chosen as those with the maximum mean score (where the mean for any paper is taken across all reviewers and all reviewers).

3.1 Reviewer and AC bids

A large number of conferences in computer science ask area chairs and/or reviewers to bid which papers they would like or not like to review, in order to obtain a better understanding of the expertise and the preferences of reviewers. Such an improved understanding is desirable as it leads to a more informed assignment of reviewers to papers, thereby improving the overall quality of the review process.

Figure 1 depicts the distribution of number of bids on papers submitted by area chairs and reviewers in NIPS 2016. Panels (a) and (b) of the figure depict the distribution of counts per paper for reviewers and area chairs respectively; panels (c) and (d) depict the distribution per area chairs and reviewers. The "not willing" and "in-a-pinch" bids were considered negative bids, whereas "willing" and "eager" bids were considered positive bids. From the data, we observe that there are very few positive bids, but a considerably higher number of negative bids.

The distribution of number of bids by reviewers is skewed by few reviewers who bid (positive and negative) on too many papers: 27% of reviewers make 90% of all bids, and 50% of reviewers make 90% of all positive bids. Moreover, there are 148 reviewers with no (positive or negative) bids and 1201 reviewers with at most 2 positive bids. In comparison, NIPS 2016 assigned at least 3 papers to most reviewers (and many other conferences also do likewise). We thus observe that a large number of reviewers do not even provide positive bids amounting to the number of papers they would review. As a consequence of the low number of bids by reviewers, we are left with 278 papers with at most 2 positive bids and 816 papers with at most 5 positive bids. In contrast, NIPS 2016 assigned 6 reviewers to most papers. There is thus a significant fraction of papers with fewer positive bids than the number of requisite reviewers. Finally there are 1090 papers with no positive bids by any AC.



(a) total number of bids by reviewers per paper

(b) total number of bids by ACs per paper

(c) number of bids per reviewer

(d) number of bids per AC

Figure 1: Histogram of number of positive and negative bids (x-axis; on a logarithmic scale) per entity (counts on y-axis) for various entities. The first column in each histogram represents number of entities with 0 bids. For example, the first column of panel (c) depicts that 756 reviewers made zero positive bids and 425 reviewers made zero negative bids.

Summary 1: Reviewer and AC bids

Key observations:

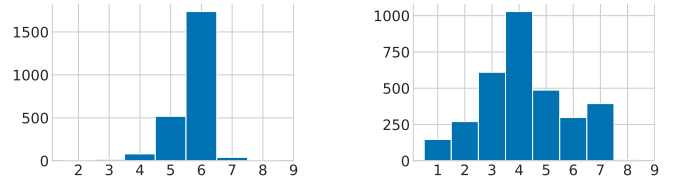
- There are very few positive bids by reviewers, with 278 papers receiving at most 2 positive bids and 816 papers receiving at most 5 positive bids.
- From the reviewers' side, the bids are highly skewed: 50% of reviewers make 90% of all positive bids, 148 reviewers make no (positive or negative) bids, and 1201 reviewers make at most 2 positive bids.
- There are 1090 papers with no positive bids by any AC.

Action items:

- When a reviewer or AC logs into the system, show unbid papers on top.
- Inform reviewers of the procedure employed to use their bids for assigning papers. Make reviewers aware of the benefits of bidding, such as receiving more relevant papers to read and serving the community by improving the review process.

Open problems:

- How to incentivize more (positive) bids so that the organizers understand preferences better for accurate reviewer assignment?
- Design a principled means of combining bids, paper content-reviewer profile similarity, and subject similarity.



(a) number of reviewers per paper (b) number of papers per reviewer

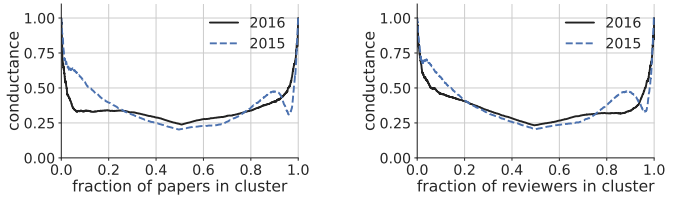
Figure 2: Histogram of number of reviews.

3.2 Reviewer assignment

Figure 2 depicts the histograms of the number of reviewers assigned per paper, and the number of papers handled by each reviewer.

In order to ensure that the information about each paper "spreads" across the entire system, it is important that there is no set of reviewers or papers that has only a small overlap with the remaining reviewers and papers (Olfati-Saber et al., 2007; Shah et al., 2016a). To analyze whether this was the case, we considered two graphs. We built a *reviewer graph* that has reviewers as vertices, and an edge between any two reviewers if there exists at least one paper that has been reviewed by both of them. Analogously we built a *paper graph*, where vertices represent papers, and we connect two papers by an edge if there exists a reviewer who has reviewed both papers. Note that the graph structure is in part dictated by a constraint on the maximum number of papers per reviewer as well as the specified number of reviewers per paper.

Our objective is to examine the structure of the graphs and determine if there were any separated communities of nodes. In order to do so, we employ a method based on spectral clustering. Formally, denote any graph as $G = (V, E)$ where V is set of nodes, and E is the set of (undirected) edges between nodes, and let $|V|$ denote the number of nodes in the graph. We can denote graph connectivity by its associated adjacency matrix A which is a $(|V| \times |V|)$ matrix; we have $A_{ij} = 1$ if there is an edge between nodes i and j and $A_{ij} = 0$ otherwise. With this notation, a quantity known as the "conductance" Φ of any set of nodes



(a) paper graph (b) reviewer graph

Figure 3: Conductance value as function of varying cluster size. The x-axes in these plots represent the normalized cluster size $k/|V|$.

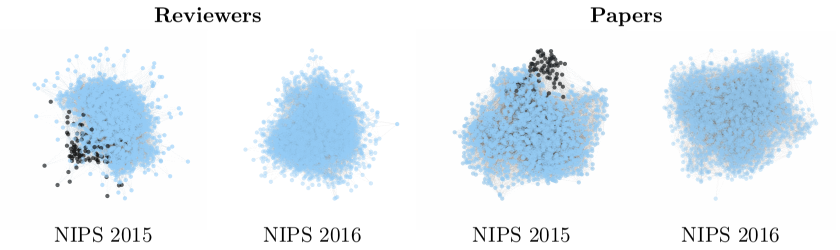


Figure 4: Graphs depicting connectivity of reviewers and that of papers for NIPS 2015 and NIPS 2016. The nodes in black (dark) show set of nodes identified by the local minima in the conductance plots (Figure 3) for NIPS 2015, and the remaining nodes are plotted in blue (light).

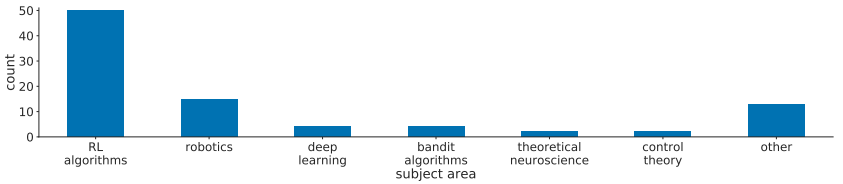


Figure 5: Histogram of subject areas in the identified cluster (from Figure 4) of reviewers in NIPS 2015 which is not well connected with the set of remaining reviewers.

$S \subset V$ is then defined as:

$$\Phi(S) = \frac{\sum_{i \in S, j \notin S} A_{ij}}{\max\{|S|, |V \backslash S|\}},$$

where $V \backslash S$ is the complement of set S. A lower value of the conductance indicates that the nodes in the cut are less connected to the remaining graph. Next, with a minor abuse of notation, the conductance of a graph as function of cluster sizes is defined as:

$$\Phi(k) = \min_{S \in V, |S| = k} \Phi(S),$$

for every $k \in \{1, \ldots, |V| - 1\}$. The plot of k versus $\Phi(k)$ is called a Network Community Profile or NCP plot (Leskovec et al., 2008). The NCP plot measures the quality of the least connected community (lowest conductance) in a large network, as a function of the size of the community. Although computing the function $\Phi(k)$ exactly may be computationally hard, an approximate value can be computed using a simple "second left eigenvector" procedure (Section 2.3 of Benson et al., 2015). A well connected graph would have a smooth plot of $\Phi(k)$ with a minima at around $k = |V|/2$.

Figure 3 shows the NCP plot for an increasing number of papers (respectively reviewers) in the paper graph (respectively reviewer graph). For reference we also plot the same curve

for graphs associated with NIPS 2015 conference. Both plots for NIPS 2015 have local minima at around $k = 0.96|V|$, indicating that there is a densely connected community of reviewers and papers that are not well connected with the rest of the graph. In contrast, the plot associated with NIPS 2016 decreases smoothly and reaches its global minimum when half of the nodes are in one cluster and the other half in another cluster, indicating an absence of such a fragmentation.

In Figure 4, we plot the graph of reviewers and papers using the algorithm of Fruchterman and Reingold (1991). In these figures we identify the set of nodes that are identified using the aforementioned NCP method; these nodes are colored black (dark) in the figure in contrast to the blue (light) color of the remaining nodes. We can see from the Figure 4 that these nodes are on the periphery of the network with lower connectivity compared to the rest of the graph.

We further examine the cluster of reviewers in NIPS 2015 which is not well connected with the rest. In Figure 5, we plot the decomposition of this set in terms of the primary subject areas indicated by the reviewers. Our analysis reveals that a bulk of this cluster comprises a single subject area—reinforcement learning. Conversely, 50 out of 78 reviewers who identified their primary subject area as reinforcement learning lie in this cluster. All in all, graph connectivity issues of this form can lead to increased noise or bias in the overall decisions. Our main message for future conferences is to employ such methods of graph analysis in order to catch issues of this form *at a global level* (not just local to individual ACs) before the reviews are assigned.

Summary 2: Reviewer assignment

Key observations:

- A cluster of papers and reviewers primarily in the reinforcement learning area are not well connected to the remaining papers and reviewers in the NIPS 2015 reviewer assignments. We did not find any such separated cluster in NIPS 2016.

Action items:

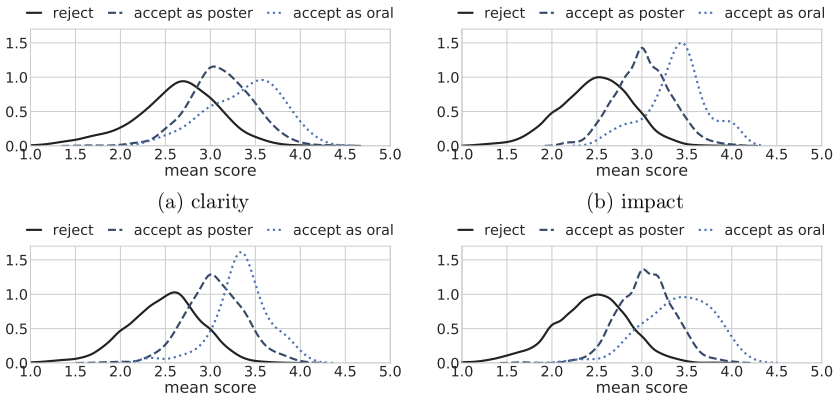
- Use graph-theoretic techniques to check global structure of graph for reviewer assignment.

Open problems:

- Design principled graph-theoretic techniques, tailored specifically to the nuances of peer-review graphs, to verify soundness of reviewer assignments.

3.3 Review-score distribution and mismatches in calibration

Recall from Section 2.3 that in the review process, for each criterion, the reviewers were asked to provide a score on a scale of 1 to 5. Specifically, they were asked to provide a score of 5 for submissions they considered as being in the top 0.1%, a score of 4 for submissions that they deemed to be in the top 3%, and a score of 3 for submissions they deemed to be in the top 30%. In this section, we compare the actual empirical distribution of reviewer scores with the distribution prescribed in the guidelines to reviewers.



(a) clarity (b) impact

(c) novelty (d) quality

Figure 6: Distribution of the mean value (across reviewers) of the score per paper for different criteria, separated according to the final decisions.

We begin by computing the distribution of the mean value (across reviewers) of the score per paper for different criteria, separated according to the final decisions. We plot these distributions in Figure 6 for each of the four criteria of clarity, impact, novelty, and quality separately.

At first glance, these histograms and numbers look quite reasonable. However, what was surprising to us was the percentage of papers that received any particular score – see Table 1. Even though the reviewers were asked to give a paper a score of 3 (poster level) or higher only if they think the paper lies in the top 30% of all papers, nearly 60% of the scores were 3 or higher. Similar effects occurred for scores 4 and 5.

One possible explanation for this phenomenon is that there were a large number of high-quality submissions to NIPS 2016. Such an improvement in quality has obvious upsides

	1 (low or very low)	2 (sub-standard)	3 (poster level: top 30%)	4 (oral level: top 3%)	5 (award level: top 0.1%)
Impact	6.6 %	36.4 %	45.9 %	10.7 %	0.4 %
Quality	6.7 %	38.3 %	45.0 %	9.6 %	0.4 %
Novelty	6.4 %	35.0 %	48.4 %	9.8 %	0.4 %
Clarity	7.1 %	28.1 %	48.9 %	14.7 %	1.2 %

Table 1: Distribution of the reviews according to the provided scores for each of the four criteria. The column headings indicate the guidelines that were provided to the reviewers. Observe that the percentage of reviews providing scores of 3, 4 or 5 is considerably higher than the requested values.

such as uplifting the overall experience of the conference. The downside is that the burden on selecting the accepted papers among all those good submissions is with the area chairs, who now still had to reduce the 60% good papers to 23% accepted papers. A second possible explanation is that the reviewers were not calibrated that well with respect to the paper quality. A third possible explanation is that the elicitation was in terms of a score in the set $\{1, 2, 3, 4, 5\}$ which represents a linear scale with equal spacing, whereas the text instructions expected reviewers to rate on a non-linear scale. This mismatch could be a source of bias in the elicited ratings. Using a linear rating scale when the actual elicitation is non-linear is a common practice in many conferences, and it will be useful to perform a similar analysis on the data from these conferences. In either case, we understand that this obviously led to the frustration of many authors, whose papers received good scores but were rejected.

In addition to scores for the four criteria, the reviewer could also indicate whether the paper had a "fatal flaw". We observe that 32% of all papers were flagged to have a "fatal flaw" by at least one reviewer.

Summary 3: Review-score distribution and mismatches in calibration

Key observations:

- The fraction of reviews with high ratings is significantly higher than what was asked from the reviewers. For instance, nearly 60% of scores are 3 or higher even though reviewers were asked of scores of 3 or higher only when they thought the paper was in the top 30% of submissions.

Action items:

- If eliciting ratings, do not use numbered scales (that is, do not use "1", "2", ...). Alternatively, one may employ other means of elicitation such as rankings.
- When making reviews visible to authors, show the percentile with respect to the data instead of absolute scores, e.g., provide feedback of the form "your paper is in the top 40% of all submitted papers in terms of novelty..."
- Include an expert in elicitation, survey methodology or user interface design to help to design what and how to ask (O'Hagan et al., 2006).

Open problems:

- Since each reviewer reviews only a small subset of the submitted papers, how to calibrate the reviews?
- What is the best interface for eliciting reviewer responses?
- What is the best way to present the review results to authors in order to provide most useful feedback and minimizing distress?

3.4 Different types of reviewers

In this section, we compare the reviews provided by the volunteer (pool 2) reviewers to those provided by the invited (pool 1) reviewers. The inclusion of volunteer reviewers has two important benefits: (a) It increases the transparency of the review process. (b) Volunteer

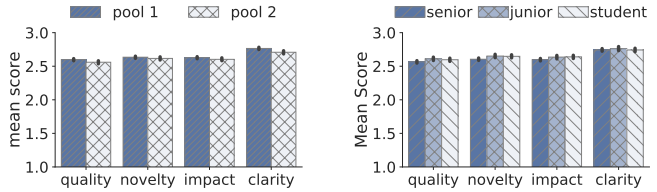
reviewers may be new today but in 2 years down the line they will gain experience and become useful to accommodate the massive growth of the conference. Given these benefits of including volunteer reviewers, this analysis looks for any systematic differences between the review scores provided by the two pools of reviewers.

Mean scores.

Junior reviewers are often perceived to be more critical than senior reviewers (Tomiyama, 2007; Toor, 2009). As Tomiyama (2007) notes, *"You submit your manuscript and then just pray it doesn't get sent to a junior faculty member – young faculty are merciless!"* In this section, we examine this hypothesis in the NIPS 2016 reviews. In Figure 7, we plot the mean score provided by each group of reviewers for each individual criterion. We apply a t-test on observed scores and compute the effect size to examine if there is a statistically significant difference in the underlying means of the scores provided by different categories of reviewers. For Pool 1 vs Pool 2, this analysis shows only clarity to have a statistically significant difference between the two pools after accounting for multiple testing. Specifically, the p-values (before accounting for multiple testing) and effect sizes for the four criteria are: novelty p=0.2143, d= 0.0264, quality p=0.0061, d= 0.0581, impact p=0.0961, d= 0.0353, and clarity p=1.91×10^{-04}, d= 0.0788. Sample sizes for Pool 1 and Pool 2 reviews are 9244 and 4430 respectively.

A similar analysis between senior researchers (e.g., faculty), junior researchers (e.g., postdocs), and PhD students reveals no significant difference between these categories. The remainder of this paragraph details the p-values and effect sizes. The p-values (before accounting for multiple testing) and effect sizes for senior researcher vs. junior researchers for the four criteria are: quality p=0.0071, d= −0.0662, novelty p=0.0037, d= −0.0704, impact p=0.0199, d= −0.0569, and clarity p=0.3064, d= −0.0253; for junior researcher vs. students: quality p=0.4662, d= 0.0164, novelty p=0.8247, d= 0.0049, impact p=0.8733,d= −0.0036, and clarity p=0.3529, d= 0.0209; for senior researcher vs. students: quality p=0.0440, d= −0.0454, novelty p=0.0499, d= −0.0629, impact p=0.0076, d= −0.0601 and clarity p=0.9968, d= 0.00009. The sample sizes for senior, junior and student reviews are: 6335, 3938, and 3354 respectively. This analysis excludes 47 reviews by reviewers who did not identify themselves as any of the above categories.

Self-reported confidence. We next study the difference in the self-reported confidence among different groups of reviewers. The mean value of reported confidence is plotted in



(a) invited and volunteer reviewers (b) senior, junior, and student reviewers

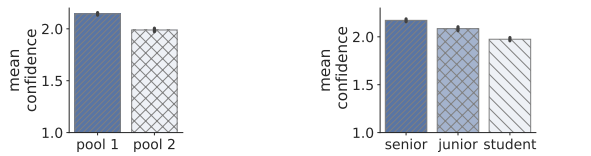
Figure 7: Mean of scores provided for different criteria grouped by different reviewer types.

Figure 8. In this case, we see a statistically significant correlation between seniority and self-reported confidence. Following are p-values (before accounting for multiple testing) and corresponding effect sizes: senior vs. junior researcher: p=4.1683×10^{-11}, d= 0.1604, senior researcher vs. PhD student: p=3.308×10^{-57}, d= 0.3577 and junior researcher vs. PhD student: p=8.074×10^{-15}, d= 0.1758. We observe a similar difference in confidence score and effect size between pool 1 and pool 2 reviewers: p=3.9679×10^{-44}, d= 0.2943.

Consistency. We now study the consistency within reviewers of pool 1 (invited), and within reviewers of pool 2 (volunteer). The consistency captures the amount of variance or disagreements in the reviews provided by that pool. As noted by Ragone et al. (2013), *"the disagreement among reviewers is a useful metric to check and monitor during the review process. Having a high disagreement means, in some way, that the judgment of the involved peers is not sufficient to state the value of the contribution itself. This metric can be useful to improve the quality of the review process..."*

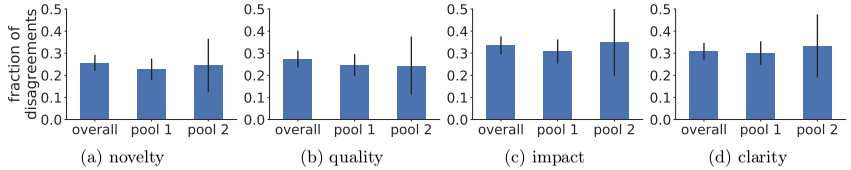
Concretely, consider any pair of reviewers within a given pool, any pair of papers that is reviewed by both the reviewers, and any criterion. We say that this pair of reviewers agrees on this pair of papers (for this criterion) if both reviewers rate the same paper higher than the other; we say that this pair disagrees if the paper rated higher by one reviewer is rated lower by the other. Ties are discarded. We count the total number of such agreements and disagreements within each of the two pools.

Figure 9 plots the fraction of disagreements within each of the two pools for the cardinal scores. At this aggregate level, we do not see enough difference to conclusively rate any one pool's intra-pool agreement above the other (note that the sample size for pool 2 is small, as listed below). Specifically, for the Pearson's chi-squared test and effect sizes of pool 1 vs. pool 2, the results for the four criteria (before accounting for multiple testing) are: novelty



(a) invited and volunteer reviewers (b) senior, junior, and student reviewers

Figure 8: Self-reported confidence grouped by different reviewer types.



(a) novelty (b) quality (c) impact (d) clarity

Figure 9: Proportions of inter-reviewer disagreements on each criterion.

p=0.9269 d= -0.0426, quality p=0.8648, d= 0.0039, impact p=0.7296, d= -0.0936, and clarity p=0.8029, d= -0.0709. The total sample sizes for the three categories of overall, pool 1 and pool 2 respectively across the four criteria are: novelty 554, 282 and 49; quality 523, 285 and 41; impact 513, 276 and 37; and clarity 572, 286 and 42. Section 3.8 presents similar consistency results for the two pools in the ordinal rankings. (We also attempted to run this analysis restricted to the top 2k papers, but this restriction results in a very low sample complexity and hence underpowered tests.)

Participation in discussions. One fact that caught our attention was the amount of participation in the discussion by the different reviewer groups: senior reviewers take much more active roles in the discussions than junior researchers. Please see Section 3.5.1 for details, where we provide a more detailed study of the discussion phase.

Summary 4: Different types of reviewers

Key observations:

- We find no evidence of a critical bias of junior reviewers (except for a small difference in the "clarity" criterion).
- Self-reported confidence correlates with seniority.
- Volunteer reviewers yield benefits of scalability and transparency, with no observable biases and a similar inter-reviewer agreement as the invited pool. These reviewers can soon be an asset in dealing with the rapid growth of conferences such as NIPS.

Action items:

- Continue to include volunteer reviewers with an appropriate moderation of their reviews.

Open problems:

- How do we make most effective use of volunteer reviewers in a manner that authors can trust, which reduces randomness in the peer-review process, and trains junior reviewers effectively?

3.5 Rebuttals and discussions

This section is devoted to the analysis of the rebuttal stage and the participation of reviewers in discussions. We begin with some summary statistics. The authors of 2188 papers submitted a rebuttal. There were a total of 12154 reviews that came in before the rebuttals started, and with some more reviews received after the rebuttal round, the total number of {reviewer, paper} pairs eventually ended up being 13674. Out of the 12154 reviews that were submitted before the rebuttals, the scores of only 1193 changed subsequently. These changed review scores were distributed among 886 papers.

There were 842 papers for which no reviewer participated in the discussions, 339 papers for which exactly one reviewer participated, and 436, 376, 218, 135 and 49 papers for which 2, 3, 4, 5 and 6 reviewers participated respectively. There were a total of 5255 discussion posts, and 4180 of the 13674 {reviewer, paper} pairs participated in the discussions.

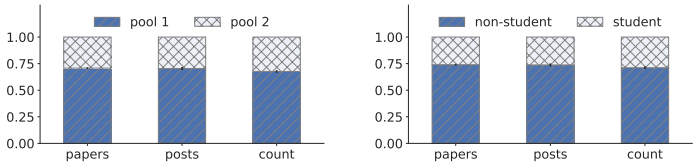
3.5.1 Who participates in discussions?

We compare the amount of participation of various groups of reviewers in the discussion phase of the review process.

Pool 1 (invited) versus pool 2 (volunteer) reviewers. We compare the participation of the reviewers in two pools in the discussions as follows, and plot the results in Figure 10(a). In order to set a baseline, we first compute the total number of {pool 1 reviewer, paper} pairs and the total number of {pool 2 reviewer, paper} pairs – these counts are computed irrespective of whether the reviewer participated in the discussions or not. We plot the proportions of these counts as the "count" bar in the figure. Next we compute the total number of posts made by pool 1 reviewers and that made by pool 2 reviewers – the resulting proportions are plotted as the "posts" bar in the figure. Finally, we compute the number of {pool 1 reviewer, paper} pairs in which that reviewer put at least one post in the discussion for that paper, and the number of {pool 2 reviewer, paper} pairs in which that reviewer put at least one post in the discussion for that paper. We plot the two proportions in the "papers" bar. The total sample sizes for the categories of counts, posts and papers are 13674, 5255 and 4180 respectively.

We tested whether the mean number of posts per {reviewer, paper} pair is identical for the two pools of reviewers. For the null hypothesis that the means are identical for the two pools of reviewers, the t-test yielded $p = 1.36 \times 10^{-4}$. We also conducted this analysis for the restriction of papers to the top 2k, and for this subset, the t-test yielded $p = 9.458 \times 10^{-4}$. We see a statistically significantly higher participation by the pool 1 reviewers as compared to the pool 2 reviewers in the discussions. However, the absolute amount of participation by either group is moderate at best, and the effect sizes are small with d= 0.0704 and d= 0.0894 for analysis of all papers and top 2k papers respectively.

Student versus non-student reviewers. We calculated the above three sets of quantities for student and non-student reviewers. Figure 10(b) depicts the results. We tested whether the mean number of posts per {reviewer, paper} pair for the student reviewers is identical to the non-student reviewers. For the null hypothesis that the means are identical, the t-test yielded $p = 3.016 \times 10^{-4}$. When restricted to the top 2k papers, the t-test yielded $p = 8.932 \times 10^{-4}$. We see a statistically significantly higher participation by the non-student reviewers as compared to the student reviewers in the discussions. However, the total amount



(a) invited and volunteer reviewers (b) student and non-student reviewers

Figure 10: Proportions of contributions from different types of reviewers in discussions ("posts" and "papers") and the total number of such reviewers ("count").

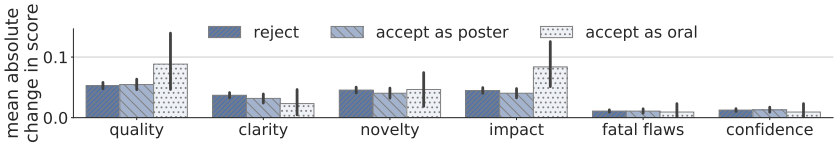


Figure 11: Mean absolute value of the change in the scores from before the rebuttal round to the end of the discussion phase.

of participation by either group is not too large, and the effect sizes are small with d= 0.0695 and d= 0.0929 respectively.

3.5.2 How do discussions change the scores?

A total of 1193 out of 12154 reviews that were submitted before rebuttals changed subsequently. These changed reviews were distributed among 886 papers. As a result, the amount of change in review scores is quite small. Figure 11 depicts the score change – in absolute value – averaged across all reviewers and all papers. While the allowed range of the scores is 1 to 5, the change in mean score is less than 0.1.

From the point of view of reviewers, we see a significant correlation between participation in the discussions and the final decisions. Specifically, for each paper we computed the mean of the scores given by all reviewers who participated in the discussions and the mean of the scores given by all reviewers who did not participate (when there was at least one reviewer of each type). We discarded this paper if both types of reviewers provided an identical mean score. If the participating reviewers gave a higher mean score than the non-participating reviewers and if the paper was accepted, we counted it as an agreement of the final decision with the participating reviewers. If the participating reviewers gave a lower mean score than the non-participating reviewers and if the paper was not accepted, then also we count it as an agreement of the final decision with the participating reviewers. Otherwise, we counted the paper as having a disagreement between the final decisions and the participating reviewers. From the data, we observe a statistically significant agreement of the final decisions and the participating reviewers with $p = 1.6 \times 10^{-6}$ with d= 0.13. We continue to observe a statistically significant correlation when this analysis is performed restricted to pool 1 reviewers ($p = 7.7 \times 10^{-4}$) or to pool 2 reviewers ($p = 1.3 \times 10^{-4}$) alone. Of course, we cannot tell the causality from this correlation, as to whether the discussions actually influenced the decisions or not.

All in all, we observe that only a small fraction of the reviews change scores following the rebuttals. Moreover the magnitude of this change in scores is very small. This observation suggests that this rebuttal process may not be very useful. That said, there are various qualitative aspects that are not accommodated in this quantitative aggregate statistic. First, it may be possible that more reviews changed with respect to the text comments but the reviewers just did not bother to change the scores – we are unable to check this property since there is no snapshot of the text comments before the rebuttal. Second, there are a reasonable number of discussion posts, however, we do not know what fraction of these posts where reviewers shifted from their earlier opinion. Third, the final decisions are correlated positively

with the reviewers who participated in discussions. Taking these factors into account, we think that the present rebuttal system should be put under the microscope regarding its value for the time and effort of such a large number of people. It may also be worth trying alternative systems of recourse for authors, such as a formal appeals process, that help to put more focus on the actual borderline cases.

Summary 5: Rebuttals and discussions

Key observations:

- There is little change in scores post-rebuttal and a moderate amount of discussion.
- Invited and non-student reviewers participate marginally more in the discussions.
- Final decisions correlated with scores given by reviewers who participated in discussions, even when stratified by individual pools.

Action items:

- Force every reviewer to change or confirm their scores after the end of the discussion session.

Open problems:

- How to incentivize reviewer participation in rebuttals/discussions?
- How to de-bias reviewers from their initial opinion?
- Compare the amount of discussion and changes in scores with that in open review processes (particularly when open reviews are used for conferences of this scale).
- Compare the efficiency of the rebuttal process with a post-decision appeal procedure to catch only cases that deserve discussion (i.e., possible mistakes).

3.6 Distribution across subject areas

Figure 12 plots scatter plot of the number of submitted papers and the number of accepted papers per (primary) subject area. Of course the proportions are not identical, but the plots do not show any systematic bias either towards or against any particular areas. A chi-square test of homogeneity of the two distributions failed to detect any significant difference between the two distributions: p=0.6029, $\chi^2(dof = 62, \#samples = 2425) = 57.51$.

Summary 6: Distribution across subject areas

Key observations:

- No observable bias across subject areas in terms of final acceptances.

Action items:

- Test for systematic biases for/against any subject area before announcing decisions.

Open problems:

- How to assimilate different, subjective opinions of reviewers across subject areas.

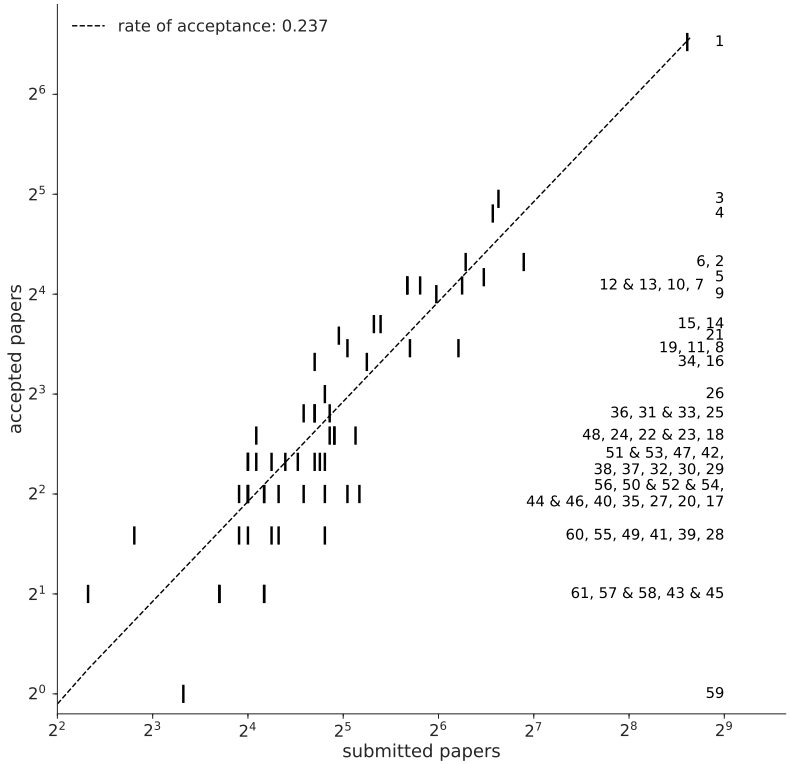


Figure 12: Number of accepted vs submitted papers per (primary) subject area. The indices of the subject areas are provided on the right of the corresponding points and their names are provided in Appendix A. For subject areas with same number of accepted papers, the labels associated with each subject area are listed (from left to right) in ascending order of number of submitted papers; if the number of submitted papers is also identical then the indices are grouped with an "&" sign. Both axes are on a logarithmic scale. The plot excludes subject area 62 since it had no accepted papers.

3.7 Quantifying the randomness

Quantifying the extent to which the outcome of a peer-review process is different from a random selection of papers is one of the most pressing questions for the scientific community (Somerville, 2016). In this section, we conduct two analyses to quantify the randomness in the review scores in NIPS 2016.

3.7.1 Messy middle model

The NIPS 2014 experiment (Lawrence and Cortes, 2014) led to the proposal of an interesting "messy middle" model (Price, 2014). The messy middle model postulates that the best and the worst papers are clear accepts and clear rejects respectively, whereas the papers in the middle suffer from random decisions that are independent of the content of the papers. The messy middle model is obviously a stylized model, but it nevertheless suggests an interesting investigation into the randomness in the reviews and decisions of the papers that lie in the middle. In this section, we describe such an investigation using the NIPS 2016 data.

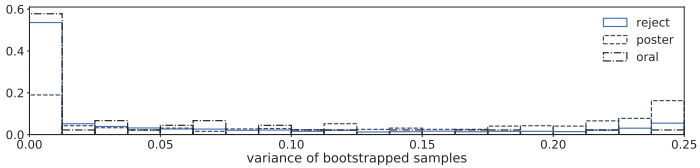
The messy middle model assumes random judgments for the middle papers. If the messy middle model were correct then for any pair of papers in the middle, and any pair of common reviewers, the probability of an agreement on the relative ranking of the two papers must be identical to the probability of disagreement. With this model in mind, we restrict attention to the papers in the middle, and then measure how far the agreements of the reviewers are from equiprobable agreements and disagreements. An analysis of this quantity for various notions of the "middle" papers yields insight into the messiness in the reviews for papers in the middle.

Procedure: We now describe the procedure employed for the analysis. Here we let n denote the total number of papers submitted to the conference and β denote the fraction of papers accepted to the conference (we have $n = 2425$ and $\beta = 0.237$ in NIPS 2016). The procedure is associated to two parameters: μ is the minimum number of samples required and α is a threshold of messiness. We choose $\mu = 100$ and $\alpha = 0.01$ in our subsequent analysis, noting that the overall conclusions are robust to these choices.

1. Rank order all papers with respect to their mean scores. Denote this ordering as Θ.
2. For every $t \in [0, 1]$ and $b \in [0, 1]$ (up to some granularity), do the following.
   - 2.1 Initialize variables $n_{\text{agree}}[t, b] = n_{\text{disagree}}[t, b] = 0$.
   - 2.2 Consider the set of papers obtained by removing the top t fraction of papers and bottom b fraction of papers from Θ. Denote this (unordered) set of "middle papers" as M.
   - 2.3 If $(\beta - t)n < \mu$ or $((1 - \beta) - b)n < \mu$ then continue to the next values of (t, b) in Step 2.
   - 2.4 Consider any pair of reviewers and any pair of papers in M that is reviewed by both the reviewers. We say that this pair of reviewers agrees on this pair of papers if both reviewers provide a higher mean score (mean computed across all criteria) to the same paper as compared to the other paper. We say that this pair disagrees if the paper rated higher by one reviewer (in terms of the mean score across the criteria) is rated lower by the other reviewer. Ties are discarded. We count the total number of such agreements (denoted as $n_{\text{agree}}[t, b]$) and disagreements (denoted as $n_{\text{disagree}}[t, b]$) within each of the two pools.
3. Find the largest value of $(1-t-b)$ such that we have $(n_{\text{agree}}[t, b]+n_{\text{disagree}}[t, b]) \geq \mu$ and $\frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]} < 0.5+\alpha$. This largest value of $(1-t-b)$ is defined as the size of the messy middle.

Conference	Size of messy middle
NIPS 2015	45%
NIPS 2016	30%

(a) Size of the messy middle windows.



(b) Histogram of the variance of acceptance decisions (according to mean scores) of the papers in a bootstrapped analysis.

Figure 13: Amount of randomness in the reviews.

Let us spend a moment interpreting some steps of the procedure. Step 2.3 and the μ-condition in Step 3 ensure that there are a sufficient number of samples for any computation on the messy middle region. Specifically, the conditions $(\beta - t)n < \mu$ and $((1 - \beta) - b)n < \mu$ ensure existence of a sufficient number of papers above and below the acceptance threshold. Under this constraint, Step 3 then finds the largest window of papers in the middle such that the fraction of reviewer-agreements is at most $(0.5 + \alpha)$. If the objective is to minimize inter-reviewer disagreement (Cole et al., 1981; Whitehurst, 1984; Lindsey, 1988), then a *smaller* size of the window is a desirable property.

We can now use this analysis to compare messy middle window sizes for two or more conferences. When making such a comparison, we make one adjustment. In the last step (Step 3), we consider only those values of (t, b) such that $n_{\text{agree}}[t, b] + n_{\text{disagree}}[t, b] \geq \mu$ for both datasets. We compare the sizes of the messy middle considering only these values.

Results: We used this procedure to compute the size of the messy middle in NIPS 2016 and also in NIPS 2015. The granularity we used is 1/20, that is, $t, b \in \{0, 1/20, 2/20, \ldots, 1\}$. NIPS 2015 had a marginally higher average number of reviews per paper as compared to NIPS 2016. We set $\mu = 100$ and $\alpha = 0.01$ (note that the conclusions drawn below are robust to these choices). The results of the analysis are tabulated in Figure 13a.

In the NIPS 2016 data, we observe that the size of the messy middle is 30%. Specifically, if we remove the bottom 70% of papers (and none of the top papers) then we see that the inter-reviewer agreements are near-random, but farther from random otherwise. On the other hand, we observe that the size of the messy middle is 45% in the NIPS 2015 data, which occurs when removing 15% of the top papers and 40% of the bottom papers.

Such an analysis is useful in comparing the noise in the review data across conferences. It can particularly be useful to evaluate the effects of any changes made in the peer-review process. The ease of doing this post hoc analysis, without necessitating any controlled experiment, is a significant benefit to this approach of analysis. In order to enable comparisons of the size of the messy middle of NIPS 2016 with other conferences, we provide the values of

$\frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]}$ and $(n_{\text{agree}}[t,b] + n_{\text{disagree}}[t,b])$ for the NIPS 2016 data for all values of (t, b) in Appendix B.

It is important to note that this post hoc analysis is not strictly comparable to the NIPS 2014 controlled experiment because we do not have access to a true ranking or a counterfactual. That said, since such an analysis can easily be performed post hoc using the data from reviews and does not require any special arrangement in the review process, it would be useful to see how these results compare to the data from other conferences.

3.7.2 A bootstrapped analysis

In this section, we conduct an analysis to measure the randomness in the reviews in the NIPS 2016 data compared to that of random selection. In our analysis, we first conduct 1000 iterations of the following procedure. For each paper, we consider the set of reviewers who reviewed this paper. We then choose the same number of reviewers uniformly at random with replacement from the set of original reviewers for this paper. We then take the mean of the scores across all criteria and across all the sampled reviewers for that paper. Next we rank order all papers in terms of these mean scores and choose the top 23.7% of the papers as "accepted" in this iteration and the others as rejected.

Our analysis focuses on the variance of the acceptance decisions for each paper. At the end of all iterations, for each paper, we compute the fraction of iterations in which the paper was accepted. Letting $\beta_i \in [0, 1]$ denote this fraction for any paper i, the variance in the acceptance decisions for this paper equals $\beta_i(1 - \beta_i)$. We plot a histogram of the computed variances (for every paper) in Figure 13b. For comparison, note that in an ideal world, the variance of the decisions for each paper would be zero. Observe that a large fraction of rejected papers as well as a large fraction of papers that were accepted as oral presentations have a near-zero variance. On the other hand, a notable fraction of papers accepted as posters as well as those rejected have a variance close to its largest possible value of $\frac{1}{4}$.

We conclude with a clarifying comment. In the NIPS 2016 data, the messy middle analysis outputs the top 30% of the papers as most noisy, while the bootstrap analysis shows a very low variance on the top (oral) papers. This may appear as a contradiction, but it is not. Notice that the messy middle analysis primarily focuses on the reviewers' opinions on the pairwise-relative values of reviews. On the other hand, the bootstrapped analysis focuses on values of reviews in relation to all other papers. Thus we have that there is a significant amount of disagreement at the top regarding which paper is better within the top set of papers, but a significant agreement that most of these papers are good enough for acceptance in an absolute sense.

Summary 7: Quantifying the randomness

Key observations:

- A notable subset of papers incurs "messy middle" randomness. The messy middle region is smaller in NIPS 2016 as compared to NIPS 2015.
- A bootstrapped analysis shows a significant variance in reviewer scores for a notable fraction of papers that are accepted as posters. A large fraction of papers accepted for oral presentations or rejected have near-zero variance.

Action items:

- Measure and compare post hoc goodness (using the analyses in this paper or through other methods) of various review processes in order to choose a good review process in a data-dependent manner.

Open problems:

- Principled design of statistical tests for post hoc comparison of goodness of different review processes.

3.8 Ordinal data collection

The data collected from the reviewers in the NIPS 2016 review process comprises cardinal ratings (in addition to the free-form text-based reviews) where reviewers score each paper on four criteria on a scale of 1 to 5. A second form of data collection that is popular in many applications, although not as much in conference reviews, is ordinal or comparative ranked data. The ordinal data collection procedure that we consider asks each reviewer to provide a total ordering of all papers that the reviewer reviewed.

There are various tradeoffs between collecting cardinal ratings and ordinal rankings. In the context of paper reviews, cardinal ratings make reviewers read each individual paper more carefully (and not make snap judgments), and can elicit more than a just one bit of information. On the other hand, ordinal rankings allow for nuanced comparative feedback, help avoid ties, and are free of various biases and calibration issues that otherwise arise in cardinal scores (Harzing et al., 2009; Krosnick and Alwin, 1988; Russell and Gray, 1994; Rankin and Grube, 1980; Cambre et al., 2018). We refer the reader to the papers by Barnett (2003); Stewart et al. (2005); Shah et al. (2016a,b); Heckel et al. (2016) and references therein for more details on ordinal data collection and processing. In the present paper, we present three sets of analyses with the ordinal rankings collected from reviewers.

3.8.1 Tie breaks

An ordinal ranking of the papers provided by a reviewer ensures that there are no ties in the reviewer's evaluations. On the other hand, asking cardinal scores can result in scores that are tied, thereby preventing an opportunity for the AC to discern a difference between the two papers from the provided scores.

In order to evaluate the prevalence of ties under cardinal scores, we performed the following computation. For every {`paper`, `paper`, `reviewer`} triplet such that the reviewer

reviewed both papers, and for any chosen criterion (i.e., quality, novelty, impact, and clarity), we computed whether the reviewer provided the same score to both papers or not. We totaled such ties and non-ties across all such triplets.

Figure 14 depicts the proportion of ties computed across all submitted papers. The total sample size is 26106. Observe that a significant fraction – exceeding 30% for each of the four criteria – of pairs of reviewer scores are tied. When only the top 2k papers were used in the calculation, the fraction of ties in each criterion further increases by approximately 10% to 15% of the respective value in the setting of all papers. In conclusion, these results reveal a significant proportion of ties in the cardinal scoring scheme. The use of ordinal rankings, on the other hand, does not suffer from such a drawback.

3.8.2 Consistency of ordinal ranking data

While there is substantial literature on benefits of collecting data in an ordinal ranking form, several past works also recommend verifying if the application setting under consideration is appropriate for ordinal rankings. For instance, Russell and Gray (1994) state the benefits of ranking for settings "where the items are highly discriminable"; Peng et al. (1997) ask respondents to rank 18 values in order of importance but observe unstable and inconsistent results; Harzing et al. (2009) argue that ranking generally requires a higher level of attention than rating and that asking respondents to rank more than a handful of statements puts a very high demand on their cognitive abilities. Accordingly, this section is devoted to performing sanity checks on the ordinal ranking data obtained in NIPS 2016. We do so by comparing certain measures of consistency of the ordinal data with the cardinal ratings for the four criteria.

Agreements within ordinal rankings. For every pair of papers that have two reviewers in common, we compute whether these two reviewers agree on the relative ordinal ranking of the two papers or if they disagree. In more detail, we say that a pair of reviewers agrees on a pair of papers if both reviewers rank the same paper higher than the other in their respective ordinal rankings; we say that this pair disagrees if the paper ranked higher by one reviewer is ranked lower by the other. Figure 15a depicts the proportion of disagreements for the ordinal rankings in the entire set of papers, as well as broken down by the type of reviewer. First,

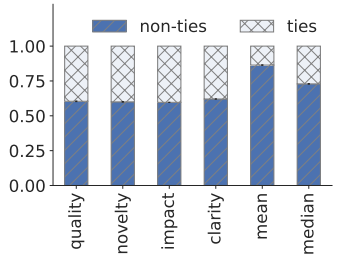


Figure 14: Proportion of ties in reviewer scores. The bars titled "mean" and "median" represent the mean and median scores across all four criteria.

observe that the ordinal rankings have a comparable level of consistency as that observed in the cardinal scores in Figure 9. Second, we observe no statistically significant difference between the two pools: p=0.9849 for Pearson's chi-squared test and effect size d= 0.0018. The sample sizes are 696, 348 and 56 for all reviewers, pool 1 and pool 2 respectively.

Agreement of ordinal rankings with cardinal ratings. Let us now evaluate how well the overall ordinal rankings associate with the cardinal scores given for the individual criteria. For every pair of papers that have a common reviewer, we compare whether the relative ordering of the cardinal scores for a given criterion agree with the ordinal ranking given by the reviewer for the pair of papers. We report the proportion of disagreements in Figure 15b. We observe the high amount of agreement of the ordinal rankings with the cardinal scores – for instance, the median cardinal score agrees in about 90% of cases with the overall ordinal rankings provided by the reviewers.

Agreement of ordinal rankings with final decisions. We finally compute the amount of agreement between the ordinal rankings provided by the reviewers and the final decisions of acceptance. We consider all {paper, paper, reviewer} triplets where the reviewer reviewed both papers, and one of these papers was eventually accepted and the other was rejected. For every such triplet, we evaluate whether the reviewer had ranked the accepted paper higher than the rejected paper ("agreement") or vice versa ("disagreement"). We report the proportion of agreements and disagreements in Figure 15c. We see that there are roughly five agreements for every disagreement.

When restricted to the top 2k papers, we observe that the disagreements of ordinal rankings with final decisions increase to 27-28% in all three categories (overall, pool 1 and pool 2) from 16-17% in the case of all papers. Note that the experiments on inter-reviewer agreements do not permit an effective analysis when restricted to top 2k papers as the sample size reduces quadratically (that is, reduces to a fraction $.47^2 \approx .2$ of the sample size with all papers).

3.8.3 Detecting anomalies

Ordinal rankings can be used to detect anomalies in reviews. We discuss this aspect in the Section 3.9.

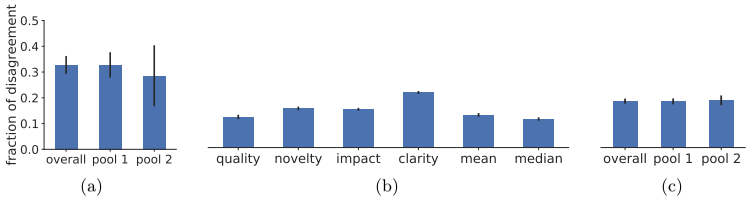


Figure 15: Fraction of disagreements (a) within ordinal rankings between different pairs of reviewer types; (b) between ordinal rankings and cardinal ratings ("mean" and "median" refer to the mean and median of the cardinal scores for the four criteria); and (c) between ordinal rankings and final acceptance decisions.

Summary 8: Ordinal data collection

Key observations:

- Ordinal rankings are a viable option for collecting reviewer opinions.
- There are a large number of ties in ratings provided by reviewers: there are more than 30% ties in each criterion and even greater fraction of ties in the top 2k papers.
- Ordinal rankings can be used to check inconsistencies in the reviews.

Action items:

- Use a hybrid collection method which elicits and combines cardinal ratings and ordinal rankings in a clever manner to avail benefits of both these types of data.

Open problems:

- Perform controlled experiments in order to quantify the benefits and possible problems with ordinal rankings.
- Design algorithms to efficiently combine cardinal ratings for criteria and ordinal overall rankings to provide useful guidelines to area chairs for their decisions.

3.9 Checking inconsistencies

In this section, we propose an automated technique to help reduce some human errors and inconsistencies in the review process. In particular, we propose to automatically check for inconsistencies in the review ratings provided by the reviewers. On finding any such inconsistency, we propose to then have the area chairs either manually investigate this inconsistency or to manually or automatically contact the reviewer requesting an explanation. In what follows, we propose two notions of inconsistencies in regards to the NIPS 2016 review process and quantify their presence in the NIPS 2016 review data.

Anomalies in criteria ratings. We investigate whether any reviewer indicated that paper "A" is strictly better than paper "B" in all four criteria, but rank paper "A" lower than paper "B" in the ordinal ranking. We find that there are 55 such pairs of reviews provided by 44 distinct reviewers. If we restrict attention to the top 2k papers, we find that that there are 10 such pairs of reviews provided by 10 distinct reviewers.[1]

Anomalies in fatal flaws. We now investigate if there are cases when a reviewer indicated a fatal flaw in a paper, but that reviewer ranked it above another paper that did not have a fatal flaw according to the reviewer. We found 349 such cases across 176 such reviewers. The proportion of such cases is similar among volunteer and invited reviewers. Among the top 2k papers, there are 55 such pairs across 33 reviewers.

One may think that the number of such cases is large because ordinal survey was done after the review process, so people may not have remembered the papers well or may not have done a thorough job as they knew it would not count towards the reviews. However,

1. Note that the total number of pairs of papers reduces more than 4-fold when moving from the set of all papers to the top 2k set.

the ordinal data actually is quite consistent with the cardinal data (Section 3.8.2). Hence we do not think such a large discrepancy with fatal flaws can be explained solely due to such a delay-related noise.

Two possible explanations for such anomalies are as follows. Either the reviewer may not have done an adequate job of the review, or the set of provided criteria are grossly inadequate to express reviewers' opinions. In either case, we suggest automatically checking for such inconsistencies (irrespective of whether ordinal or cardinal final ratings are used) during the review process, and contacting the respective reviewers to understand their reasoning.[2] We hope that such a checkpoint will be useful in improving the overall quality of the review process.

Summary 9: Checking inconsistencies

Key observations:

- 55 cases (across 44 reviewers) of a reviewer rating a paper higher than another for all criteria but inverting the relative ranking of the two papers in the overall ordering.
- 349 cases where a reviewer indicated a fatal flaw in a paper but ranked it higher than another paper without any indicated fatal flaw.

Action items:

- Check for inconsistencies in the reviews and contact respective reviewers.

Open problems:

- What other inconsistencies can be checked in an automated manner?

4. Discussion and conclusions

NIPS has historically been the terrain of much experimentation to improve the review process and this paper is our contribution to advance the state of the art in review process design. It is an on-going debate to which extent the decision process should be automated and what means could be used to automate it. We provide some elements to fuel this discussion. In this paper, we reported a post hoc analysis of the NIPS 2016 review process. Our analysis yielded useful insights into the peer-review process, suggested action items for future conferences, and resulted in several open problems towards improving the academic peer-review process, as enumerated throughout this paper.

Our tools include several means of detecting potential artifacts or biases, and statistical tests to validate hypotheses made: Comparing the distribution of topics in submitted papers and accepted papers; creating a graph of proximity of reviewers (according to commonly reviewed papers) and papers (according to common reviewers) to detect potential disconnected communities; test to compare two pools of reviewers; quantifying the noise in the review scores. We also observed that the histogram of scores obtained included a significantly larger

2. This analysis was performed after completion of the review process, and hence reviewers were not contacted for these inconsistencies.

fraction of papers than the guidelines suggested. This observation suggests a more careful design of the elicitation interface and the type of feedback provided to authors.

Selection biases that arise when recruiting reviewers and ACs in a review process of this scale are difficult to deal with. Some designs in the selection of reviewers lend themselves more to bias than others. In NIPS2016, we made some design choices of the review process with the intention of reducing these biases. For instance, the recruitment of volunteer author-reviewers helped increase the diversity of the reviewer pool. They were less prone to selection bias compared to selecting reviewers by invitation only, primarily based on AC recommendations. With respect to reducing bias across AC decisions, we introduced the "AC buddy system" in which pairs of ACs had to make decisions jointly about all their papers. This method scales well with the increase in number of papers, but is sub-optimal to calibrate well decisions since buddy pairs form disjoint decision units (no paper overlap between buddy pairs). However, decision processes based on a conference between several or all ACs, as done in earlier editions of the conference, are also not perfect because decisions are sometimes dominated by self-confident and/or opinionated ACs. Although the evidence we gathered from our analyses did not reveal any "obvious" bias, it does not mean that there is none. We hope that some designs of our review process will shed some lights on ways of improving bias-immune or bias-avoidance procedures for future conferences.

A major challenge facing the NIPS conference is that of scaling the review process with the rapid growth of the conference. To this end, we introduced the idea of inviting *volunteer* author reviewers. Training junior reviewers today will ensure a much larger and stronger reviewer pool in a few years from now. Recruiting more reviewers (between 4 and 6 per paper) ensured that each paper had a better chance to get a few competent reviews. We gave a strong role to the ACs who arbitrated between good and bad reviews and made the final decision. Some of the ACs systematically disregarded volunteer reviews, judging that they could not be trusted. Additionally, next to many PhD students, this brought a considerable amount of senior reviewers in the system as well. Our analysis did not reveal any systematic bias or additional variance in the invited reviewer pool. However, more senior reviewers seem to put more effort into providing detailed reviews, and participating to rebuttals and discussions. Hence we need to find means of encouraging and possibly educating more junior reviewers to participate in these aspects. As a means of self-assessment and encouragement, reviewers could receive statistics about review length, amount of agreement between reviewers, and participation to rebuttals and discussions, as well as figures concerning their own participation.

We evaluated how rebuttals and discussions change the scores. Although this concerns only a minority of papers, we believe that ACs have a key role in arbitrating decisions when there is a controversy and that this is not easy to monitor merely with scores. Since scores do not seem to be consistently updated by reviewers after rebuttal/discussions, maybe the review process should include a score confirmation to make sure that absence of change in score is not due to negligence. Mixing ordinal and cardinal scores may reduce the problems of reviewer calibration, tie breaking, and identifying anomalies possibly due to human error.

All in all, it is important to realize that in a review process of this scale, there is not a single person who really controls what is going on at all levels. Program chairs spend a lot of time on quality control, but definitely cannot control the decisions on all individual papers or the quality of individual reviewers. In the end, we have to trust the area chairs and

reviewers: the better reviews *all of us* provide, the better the outcome of the review process. We as a community must also continue to strive improving the peer-review process itself, via experiments, analysis, and open discussions. This topic in itself is a fertile ground for future research with many useful open problems including those enumerated throughout the paper.

Acknowledgments

We thank the two anonymous reviewers for their valuable suggestions in terms of improving both the content and the presentation of the paper. We thank the action editor Neil Lawrence for the prompt and efficient handling of the paper. This work would not have been possible without the support of the NIPS foundation and the entire committee of the NIPS 2016 conference. We thank the program chairs and program managers of the NIPS 2015 conference for their help during the organization. We thank Baiyu Chen for preliminary experiments conducted on ordinal data.

Isabelle Guyon acknowledges funding from the Paris-Saclay scientific foundation. Ulrike von Luxburg has been supported by the Deutsche Forschungsgemeinschaft (Institutional Strategy of the University of Tübingen, ZUK 63). Krikamol Muandet acknowledges fundings from the Faculty of Science, Mahidol University and the Thailand Research Fund (TRF). The work of Nihar B. Shah was supported in parts by NSF grants CRII-CCF-1755656 and CCF-1763734.

References

W. Barnett. The modern theory of consumer behavior: Ordinal or cardinal? *Quarterly Journal of Austrian Economics*, 6(1):41–65, 2003.

A. R. Benson, D. F. Gleich, and J. Leskovec. Tensor spectral clustering for partitioning higher-order network structures. In *SIAM International Conference on Data Mining*, pages 118–126. SIAM, 2015.

Julia Cambre, Scott Klemmer, and Chinmay Kulkarni. Juxtapeer: Comparative peer review yields higher quality feedback and promotes deeper reflection. In *CHI*, 2018.

Laurent Charlin and Richard Zemel. The toronto paper matching system: an automated paper-reviewer assignment system. In *ICML*, 2013.

Stephen Cole, Jonathan R Cole, and Gary A Simon. Chance and consensus in peer review. *Science*, 214:20, 1981.

T. Fruchterman and E. Reingold. Graph drawing by force-directed placement. *Software: Practice and experience*, 21(11):1129–1164, 1991.

Anne-Wil Harzing, Joyce Baldueza, Wilhelm Barner-Rasmussen, Cordula Barzantny, Anne Canabal, Anabella Davila, Alvaro Espejo, Rita Ferreira, Axele Giroud, Kathrin Koester, et al. Rating versus ranking: What is the best way to reduce response and language bias in cross-national research? *International Business Review*, 18(4), 2009.

Reinhard Heckel, Nihar B Shah, Kannan Ramchandran, and Martin J Wainwright. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arXiv*

preprint arXiv:1606.08842, 2016.

Jon A Krosnick and Duane F Alwin. A test of the form-resistant correlation hypothesis: Ratings, rankings, and the measurement of values. *Public Opinion Quarterly*, 52(4): 526–538, 1988.

N. Lawrence and C. Cortes. The NIPS Experiment. `http://inverseprobability.com/2014/12/16/the-nips-experiment`, 2014. [Online; accessed 3-June-2017].

J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, pages 695–704. ACM, 2008.

Duncan Lindsey. Assessing precision in the manuscript review process: A little better than a dice roll. *Scientometrics*, 14(1-2):75–82, 1988.

M. Merrifield and D. Saari. Telescope time without tears: a distributed approach to peer review. *Astronomy & Geophysics*, 50(4):4.16–4.20, 2009.

J. Mervis. Want a grant? First review someone else's proposal. *Sciencemag News*, July 2014. URL `http://www.sciencemag.org/news/2014/07/want-grant-first-review-someone-elses-proposal`.

A O'Hagan, C Buck, A Daneshkhah, J Eiser, P Garthwaite, D Jenkinson, J Oakley, and T Rakow. *Uncertain judgements: eliciting experts' probabilities*. John Wiley & Sons, 2006.

R. Olfati-Saber, J. A. Fax, and R. Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

Kaiping Peng, Richard E Nisbett, and Nancy YC Wong. Validity problems comparing values across cultures and possible solutions. *Psychological methods*, 2(4):329, 1997.

E. Price. The NIPS experiment. `http://blog.mrtz.org/2014/12/15/the-nips-experiment.html`, 2014. [Online; accessed 3-June-2017].

Azzurra Ragone, Katsiaryna Mirylenka, Fabio Casati, and Maurizio Marchese. On peer review in computer science: Analysis of its effectiveness and suggestions for improvement. *Scientometrics*, 97(2):317–356, 2013.

William L Rankin and Joel W Grube. A comparison of ranking and rating procedures for value system measurement. *European Journal of Social Psychology*, 10(3):233–246, 1980.

Philip A Russell and Colin D Gray. Ranking or rating? some data and their implications for the measurement of evaluative response. *British journal of Psychology*, 85(1):79–92, 1994.

N. B Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M. J Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. *Journal of Machine Learning Research*, 17:1–47, 2016a.

N. B. Shah, S. Balakrishnan, A. Guntuboyina, and M. J. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. *IEEE Transactions on Information Theory*, 2016b.

A. Somerville. A Bayesian analysis of peer reviewing. *Significance*, 13(1):32–37, 2016.

N. Stewart, G. Brown, and N. Chater. Absolute identification by relative judgment. *Psychological review*, 112(4):881, 2005.

A. Janet Tomiyama. Getting Involved in the Peer Review Process. Psychological Science Agenda, American Psychological Association. `http://www.apa.org/science/about/psa/2007/06/student-council.aspx`, 2007. [Online; accessed 27-January-2018].

Rachel Toor. Reading Like a Graduate Student. The Chronicle of Higher Education. `https://www.chronicle.com/article/Reading-Like-a-Graduate/47922`, 2009. [Online; accessed 27-January-2018].

Grover J Whitehurst. Interrater agreement for journal manuscript reviews. *American Psychologist*, 39(1):22, 1984.

APPENDIX

In the appendix we present some additional details about the experiments.

Appendix A. Subject areas

Here are the subject areas associated to the subject area indices in Figure 12.

1. Deep learning/Neural networks
2. (Application) Computer Vision
3. Learning theory
4. Convex opt. and big data
5. Sparsity and feature selection
6. Clustering
7. Reinforcement learning
8. Large scale learning
9. Graphical models
10. Bandit algorithms
11. Matrix factorization
12. Online learning
13. (Other) Optimization
14. (Other) Neuroscience
15. Kernel methods
16. Gaussian process
17. Multitask/Transfer learning
18. Component Analysis (ICA, PCA, ...)
19. Combinatorial optimization
20. Time series analysis
21. (Other) Probabilistic Models and Methods
22. (Other) Applications
23. (Other) Machine Learning Topics
24. (Cognitive/Neuro) Theoretical Neuroscience
25. (Other) Unsupervised Learning Methods
26. MCMC
27. Semi-supervised
28. (Other) Classification
29. (Application) Natural Language and Text
30. (Application) Object and Pattern Recognition
31. (Cognitive/Neuro) Neural Coding
32. Causality
33. Bayesian nonparametrics
34. Variational inference
35. Similarity and Distance Learning
36. (Other) Statistics
37. Spectral methods
38. Active Learning
39. Graph-based Learning
40. (Other) Bayesian Inference
41. (Application) Collab. Filtering / Recommender Systems
42. Information Theory
43. (Application) Signal and Speech Processing
44. (Application) Social Networks
45. (Other) Robotics and Control
46. Nonlin. dim. reduction
47. Model selection and structure learning
48. Ensemble methods and Boosting
49. Stochastic methods
50. (Other) Cognitive Science
51. Structured prediction
52. Ranking and Preference Learning
53. Game Theory and Econometrics
54. (Application) Privacy, Anonymity, Security
55. (Cognitive/Neuro) Perception
56. (Application) Bioinfo. and Systems Bio.
57. Regularization and Large Margin Methods
58. (Other) Regression
59. (Application) Information Retrieval
60. (Application) Web App. and Internet
61. (Cognitive/Neuro) Reinforcement Learning
62. (Cognitive/Neuro) Language

Appendix B. Messy middle details

In Figure 16 we provide the values of the fraction of agreements $r := \frac{n_{\text{agree}}[t,b]}{n_{\text{agree}}[t,b]+n_{\text{disagree}}[t,b]}$ at the top of the corresponding cell and number of pairs $m := (n_{\text{agree}}[t,b] + n_{\text{disagree}}[t,b])$ for every value of (t,b) at the bottom of the corresponding cell. Note that the values are computed for all values of (t,b) ignoring the sample size restriction imposed by Step 2.3 of the procedure outlined in Section 3.7.1. Each cell in the table is color-coded by the size of the 95% confidence interval (on a log-scale) computed as $(2 \times 1.96)\sqrt{\frac{r(1-r)}{m}}$.

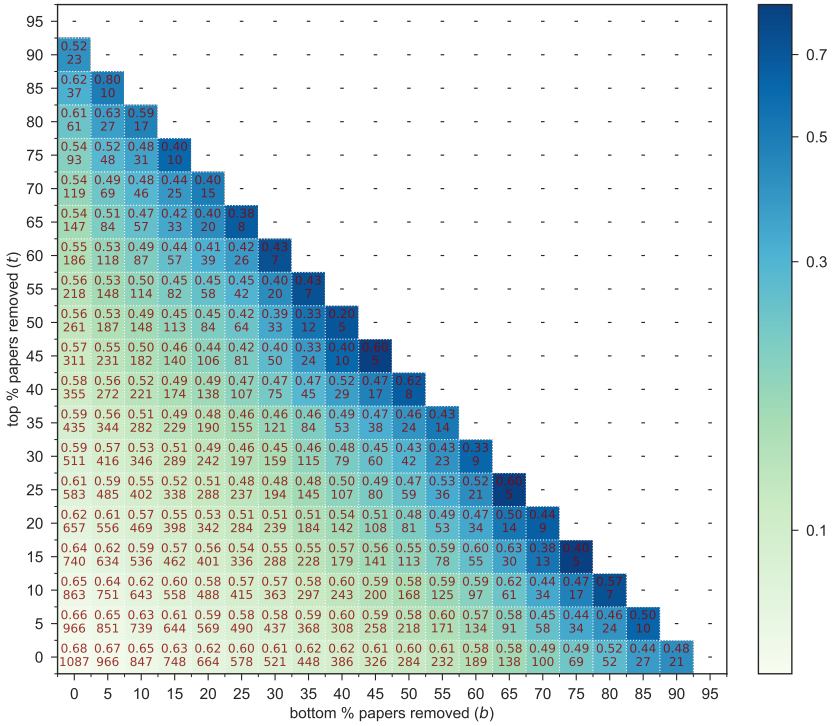


Figure 16: The inter-reviewer agreement ratios in the messy middle. For each value of t and b, we report two numbers: The agreement ratio $r := n_{\text{agree}}/(n_{\text{agree}} + n_{\text{disagree}})$ and the number of overlapping paper-reviewer pairs $m := (n_{\text{agree}} + n_{\text{disagree}})$. Each cell is color-coded by the size of the 95% confidence interval (on a log scale).

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 01/17; Revised 4/18; Published 09/12

Emergence of Invariance and Disentanglement in Deep Representations

Alessandro Achille ACHILLE@CS.UCLA.EDU
Department of Computer Science
University of California
Los Angeles, CA 90095, USA

Stefano Soatto SOATTO@CS.UCLA.EDU
Department of Computer Science
University of California
Los Angeles, CA 90095, USA

Editor: Yoshua Bengio

Abstract

Using established principles from Statistics and Information Theory, we show that invariance to nuisance factors in a deep neural network is equivalent to information minimality of the learned representation, and that stacking layers and injecting noise during training naturally bias the network towards learning invariant representations. We then decompose the cross-entropy loss used during training and highlight the presence of an inherent overfitting term. We propose regularizing the loss by bounding such a term in two equivalent ways: One with a Kullbach-Leibler term, which relates to a PAC-Bayes perspective; the other using the information in the weights as a measure of complexity of a learned model, yielding a novel Information Bottleneck for the weights. Finally, we show that invariance and independence of the components of the representation learned by the network are bounded above and below by the information in the weights, and therefore are implicitly optimized during training. The theory enables us to quantify and predict sharp phase transitions between underfitting and overfitting of random labels when using our regularized loss, which we verify in experiments, and sheds light on the relation between the geometry of the loss function, invariance properties of the learned representation, and generalization error.

Keywords: Representation learning; PAC-Bayes; information bottleneck; flat minima; generalization; invariance; independence;

1. Introduction

Efforts to understand the empirical success of deep learning have followed two main lines: Representation learning and optimization. In optimization, a deep network is treated as a black-box family of functions for which we want to find parameters (*weights*) that yield good generalization. Aside from the difficulties due to the non-convexity of the loss function, the fact that deep networks are heavily over-parametrized presents a theoretical challenge: The bias-variance trade-off suggests they may severely overfit; yet, even without explicit regularization, they perform remarkably well in practice. Recent work suggests that this is related to properties of the loss landscape and to the implicit regularization performed by stochastic gradient descent (SGD), but the overall picture is still hazy (Zhang et al., 2017).

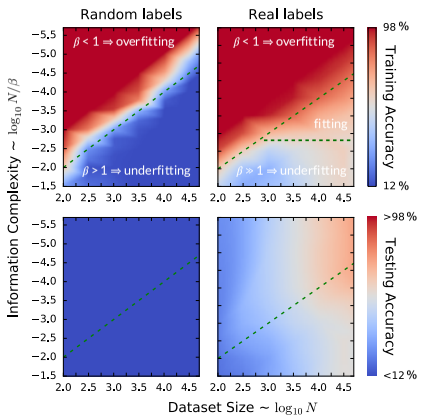


Figure 1: **(Left)** The AlexNet model of Zhang et al. (2017) achieves high accuracy (red) even when trained with random labels on CIFAR-10. Using the IB Lagrangian to limit information in the weights leads to a sharp transition to underfitting (blue) predicted by the theory (dashed line). To overfit, the network needs to memorize the dataset, and the information needed grows linearly. **(Right)** For real labels, the information sufficient to fit the data without overfitting saturates to a value that depends on the dataset, but somewhat independent of the number of samples. Test accuracy shows a uniform blue plot for random labels, while for real labels it increases with the number of training samples, and is higher near the critical regularizer value $\beta = 1$.

Representation learning, on the other hand, focuses on the properties of the representation learned by the layers of the network (the *activations*) while remaining largely agnostic to the particular optimization process used. In fact, the effectiveness of deep learning is often ascribed to the ability of deep networks to learn representations that are insensitive (invariant) to nuisances such as translations, rotations, occlusions, and also "disentangled," that is, separating factors in the high-dimensional space of data (Bengio, 2009). Careful engineering of the architecture plays an important role in achieving insensitivity to simple geometric nuisance transformations, like translations and small deformations; however, more complex and dataset-specific nuisances still need to be learned. This poses a riddle: *If neither the architecture nor the loss function explicitly enforce invariance and disentangling, how can these properties emerge consistently in deep networks trained by simple generic optimization?*

In this work, we address these questions by establishing information theoretic connections between these concepts. In particular, we show that: (a) a sufficient representation of the data is invariant if and only if it is minimal, *i.e.*, it contains the smallest amount of information, although may not have small dimension; (b) the information in the representation, along with its total correlation (a measure of disentanglement) are tightly bounded by the information that the weights contain about the dataset; (c) the information in the weights, which is related to overfitting (Hinton and Van Camp, 1993), flat minima (Hochreiter and Schmidhuber, 1997), and a PAC-Bayes upper-bound on the test error (Section 6),

can be controlled by implicit or explicit regularization. Moreover, we show that adding noise during the training is a simple and natural way of biasing the network towards invariant representations.

Finally, we perform several experiments with realistic architectures and datasets to validate the assumptions underlying our claims. In particular, we show that using the information in the weights to measure the complexity of a deep neural network (DNN), rather than the number of its parameters, leads to a sharp and theoretically predicted transition between overfitting and underfitting regimes for random labels, shedding light on the questions of Zhang et al. (2017).

1.1 Related work

The Information Bottleneck (IB) was introduced by Tishby et al. (1999) as a generalization of minimal sufficient statistics that allows trading off fidelity (sufficiency) and complexity of a representation. In particular, the IB Lagrangian reduces finding a minimal sufficient representation to a variational optimization problem. Later, Tishby and Zaslavsky (2015) and Shwartz-Ziv and Tishby (2017) advocated using the IB between the test data and the activations of a deep neural network, to study the sufficiency and minimality of the resulting representation. In parallel developments, the IB Lagrangian was used as a regularized loss function for learning representation, leading to new information theoretic regularizers (Achille and Soatto, 2018; Alemi et al., 2017a; Alemi et al., 2017b).

In this paper, we introduce an IB Lagrangian between the weights of a network and the training data, as opposed to the traditional one between the activations and the test datum. We show that the former can be seen both as a generalization of Variational Inference, related to Hinton and Van Camp (1993), and as a special case of the more general PAC-Bayes framework (McAllester, 2013), that can be used to compute high-probability upper-bounds on the test error of the network. One of our main contributions is then to show that, due to a particular duality induced by the architecture of deep networks, minimality of the weights (a function of the training dataset) and of the learned representation (a function of the test input) are connected: in particular we show that networks regularized either explicitly, or implicitly by SGD, are biased toward learning invariant and disentangled representations. The theory we develop could be used to explain the phenomena described in small-scale experiments in Shwartz-Ziv and Tishby (2017), whereby the initial fast convergence of SGD is related to sufficiency of the representation, while the later asymptotic phase is related to compression of the activations: While SGD is seemingly agnostic to the property of the learned representation, we show that it does minimize the information in the weights, from which the compression of the activations follows as a corollary of our bounds. Practical implementation of this theory on real large scale problems is made possible by advances in Stochastic Gradient Variational Bayes (Kingma and Welling, 2014; Kingma et al., 2015).

Representations learned by deep networks are observed to be insensitive to complex nuisance transformations of the data. To a certain extent, this can be attributed to the architecture. For instance, the use of convolutional layers and max-pooling can be shown to yield insensitivity to local group transformations (Bruna and Mallat, 2011; Anselmi et al., 2016; Soatto and Chiuso, 2016). But for more complex, dataset-specific, and in particular non-local, non-group transformations, such insensitivity must be acquired as

part of the learning process, rather than being coded in the architecture. We show that a sufficient representation is maximally insensitive to nuisances if and only if it is minimal, allowing us to prove that a regularized network is naturally biased toward learning invariant representations of the data.

Efforts to develop a theoretical framework for representation learning include Tishby and Zaslavsky (2015) and Shwartz-Ziv and Tishby (2017), who consider representations as stochastic functions that approximate minimal sufficient statistics, different from Bruna and Mallat (2011) who construct representations as (deterministic) operators that are invertible in the limit, while exhibiting reduced sensitivity ("stability") to small perturbations of the data. Some of the deterministic constructions are based on the assumption that the underlying data is spatially stationary, and therefore work best on textures and other visual data that are not subject to occlusions and scaling nuisances. Anselmi et al. (2016) develop a theory of invariance to locally compact groups, and aim to construct maximal ("distinctive") invariants, like Sundaramoorthi et al. (2009) that, however, assume nuisances to be infinite-dimensional groups (Grenander, 1993). These efforts are limited by the assumption that nuisances have a group structure. Such assumptions were relaxed by Soatto and Chiuso (2016) who advocate seeking for *sufficient* invariants, rather than *maximal* ones. We further advance this approach, but unlike prior work on sufficient dimensionality reduction, we do not seek to minimize the dimension of the representation, but rather its information content, as prescribed by our theory. Recent advances in Deep Learning provide us with computationally viable methods to train high-dimensional models and predict and quantify observed phenomena such as convergence to flat minima and transitions from overfitting to underfitting random labels, thus bringing the theory to fruition. Other theoretical efforts focus on complexity considerations, and explain the success of deep networks by ways of statistical or computational efficiency (Lee et al., 2017; Bengio, 2009; LeCun, 2012). "Disentanglement" is an often-cited property of deep networks (Bengio, 2009), but seldom formalized and studied analytically, although Ver Steeg and Galstyan (2015) has suggested studying it using the Total Correlation of the representation, also known as multi-variate mutual information, which we also use.

We connect invariance properties of the representation to the geometry of the optimization residual, and to the phenomenon of *flat minima* (Dinh et al., 2017).

Following (McAllester, 2013), we have also explored relations between our theory and the PAC-Bayes framework (Dziugaite and Roy, 2017). As we show, our theory can also be derived in the PAC-Bayes framework, without resorting to information quantities and the Information Bottleneck, thus providing both an independent and alternative derivation, and a theoretically rigorous way to upper-bound the optimal loss function. The use of PAC-Bayes theory to study the generalization properties of deep networks has been championed by Dziugaite and Roy (2017), who point out that minima that are flat in the sense of having a large volume, toward which stochastic gradient descent algorithms are implicitly or explicitly biased (Chaudhari and Soatto, 2018), naturally relates to the PAC-Bayes loss for the choice of a normal prior and posterior on the weights. This has been leveraged by Dziugaite and Roy (2017) to compute non-vacuous PAC-Bayes error bounds, even for deep networks.

2. Preliminaries

A training set $\mathcal{D} = \{\mathbf{x}, \mathbf{y}\}$, where $\mathbf{x} = \{x^{(i)}\}_{i=1}^{N}$ and $\mathbf{y} = \{y^{(i)}\}_{i=1}^{N}$), is a collection of N randomly sampled data points $x_i^{(i)}$ and their associated (usually discrete) labels. The samples are assumed to come from an unknown, possibly complex, distribution $p_\theta(x, y)$, parametrized by a parameter θ. Following a Bayesian approach, we also consider θ to be a random variable, sampled from some unknown prior distribution $p(\theta)$, but this requirement is not necessary (see Section 6). A test datum x is also a random variable. Given a test sample, our goal is to infer the random variable y, which is therefore referred to as our *task*.

We will make frequent use of the following standard information theoretic quantities (Cover and Thomas, 2012): Shannon entropy $H(x) = \mathbb{E}_p[-\log p(x)]$, conditional entropy $H(x|y) := \mathbb{E}_{\bar{y}}[H(x|y = \bar{y})] = H(x, y) - H(y)$, (conditional) mutual information $I(x; y|z) = H(x|z) - H(x|y, z)$, Kullbach-Leibler (KL) divergence $KL(p(x)||q(x)) = \mathbb{E}_p[\log p/q]$, cross-entropy $H_{p,q}(x) = \mathbb{E}_p[-\log q(x)]$, and total correlation $TC(z)$, which is also known as multi-variate mutual information and defined as

$$\mathrm{TC}(z) = \mathrm{KL}(\, p(z) \,\|\, \textstyle\prod_i p(z_i)\,),$$

where $p(z_i)$ are the marginal distributions of the components of z. Recall that the KL divergence between two distributions is always non-negative and zero if and only if they are equal. In particular $\mathrm{TC}(z)$ is zero if and only if the components of z are independent, in which case we say that z is *disentangled*. We often use of the following identity:

$$I(z; x) = \mathbb{E}_{x\sim p(x)}\, \mathrm{KL}(\, p(z|x) \,\|\, p(z)\,).$$

We say that x, z, y form a Markov chain, indicated with $x \to z \to y$, if $p(y|x, z) = p(y|z)$. The Data Processing Inequality (DPI) for a Markov chain $x \to z \to y$ ensures that $I(x; z) \geq I(x; y)$: If z is a (deterministic or stochastic) function of x, it cannot contain more information about y than x itself (we cannot create new information by simply applying a function to the data we already have).

2.1 General definitions and the Information Bottleneck Lagrangian

We say that z is a *representation* of x if z is a stochastic function of x, or equivalently if the distribution of z is fully described by the conditional $p(z|x)$. In particular we have the Markov chain $y \to x \to z$. We say that a representation z of x is **sufficient** for y if $y \perp\!\!\!\perp x \mid z$, or equivalently if $I(z; y) = I(x; y)$; it is **minimal** when $I(x; z)$ is smallest among sufficient representations. To study the trade-off between sufficiency and minimality, Tishby et al. (1999) introduces the Information Bottleneck Lagrangian

$$\mathcal{L}(p(z|x)) = H(y|z) + \beta\, I(z; x), \tag{1}$$

where β trades off sufficiency (first term) and minimality (second term); in the limit $\beta \to 0$, the IB Lagrangian is minimized when z is minimal and sufficient. It does not impose any restriction on disentanglement nor invariance, which we introduce next.

2.2 Nuisances for a task

A **nuisance** is any random variable that affects the observed data x, but is not informative to the task we are trying to solve. More formally, a random variable n is a nuisance for the task y if $y \perp\!\!\!\perp n$, or equivalently $I(y; n) = 0$. Similarly, we say that the representation z is **invariant** to the nuisance n if $z \perp\!\!\!\perp n$, or $I(z; n) = 0$. When z is not strictly invariant but it minimizes $I(z; n)$ among all sufficient representations, we say that the representation z is **maximally insensitive** to n.

One typical example of nuisance is a group G, such as translation or rotation, acting on the data. In this case, a deterministic representation f is invariant to the nuisances if and only if for all $g \in G$ we have $f(g \cdot x) = f(x)$. Our definition however is more general in that it is not restricted to deterministic functions, nor to group nuisances. An important consequence of this generality is that the observed data x can always be written as a deterministic function of the task y and of all nuisances n affecting the data, as explained by the following proposition.

Proposition 2.1 (Task-nuisance decomposition, Appendix C.1) *Given a joint distribution $p(x, y)$, where y is a discrete random variable, we can always find a random variable n independent of y such that $x = f(y, n)$, for some deterministic function f.*

3. Properties of optimal representations

To simplify the inference process, instead of working directly with the observed high dimensional data x, we want to use a representation z that captures and exposes only the information relevant for the task y. Ideally, such a representation should be (a) **sufficient** for the task y, *i.e.* $I(y; z) = I(y; x)$, so that information about y is not lost; among all sufficient representations, it should be (b) **minimal**, *i.e.* $I(z; x)$ is minimized, so that it retains as little about x as possible, simplifying the role of the classifier; finally, it should be (c) **invariant** to the effect of nuisances $I(z; n) = 0$, so that the final classifier will not overfit to spurious correlations present in the training dataset between nuisances n and labels y. Such a representation, if it exists, would not be unique, since any bijective mapping preserves all these properties. We can use this to our advantage and further aim to make the representation (d) maximally **disentangled**, *i.e.,* choose the one(s) for which $\mathrm{TC}(z)$ is minimal. This simplifies the classifier rule, since no information will be present in the higher-order correlations between the components of z.

Inferring a representation that satisfies all these properties may seem daunting. However, in this section we show that we only need to enforce (a) sufficiency and (b) minimality, from which invariance and disentanglement follow naturally thanks to the stacking of noisy layers of computation in deep networks. We will then show that sufficiency and minimality of the learned representation can also be promoted easily through implicit or explicit regularization during the training process.

Proposition 3.1 (Invariance and minimality, Appendix C.2) *Let n be a nuisance for the task y and let z be a sufficient representation of the input x. Suppose that z depends on n only through x (*i.e., *$n \to x \to z$). Then,*

$$I(z; n) \leq I(z; x) - I(x; y).$$

Moreover, there is a nuisance n such that equality holds up to a (generally small) residual ϵ

$$I(z;n) = I(z;x) - I(x;y) - \epsilon,$$

where $\epsilon := I(z;y|n) - I(x;y)$. In particular $0 \leq \epsilon \leq H(y|x)$, and $\epsilon = 0$ whenever y is a deterministic function of x. Under these conditions, a sufficient statistic z is invariant (maximally insensitive) to nuisances if and only if it is minimal.

Remark 3.2 Since $\epsilon \leq H(y|x)$, and usually $H(y|x) = 0$ or at least $H(y|x) \ll I(x;z)$, we can generally ignore the extra term.

An important consequence of this proposition is that we can construct invariants by simply reducing the amount of information z contains about x, while retaining the minimum amount $I(z;x)$ that we need for the task y. This provides the network a way to automatically learn invariance to complex nuisances, which is complementary to the invariance imposed by the architecture. Specifically, one way of enforcing minimality explicitly, and hence invariance, is through the IB Lagrangian.

Corollary 3.3 (Invariants from the Information Bottleneck) *Minimizing the IB Lagrangian*

$$\mathcal{L}(p(z|x)) = H(y|z) + \beta\, I(z;x),$$

in the limit $\beta \to 0$, yields a sufficient invariant representation z of the test datum x for the task y.

Remarkably, the IB Lagrangian can be seen as the standard cross-entropy loss, plus a regularizer $I(z;x)$ that promotes invariance. This fact, without proof, is implicitly used in Achille and Soatto (2018), who also provide an efficient algorithm to perform the optimization. Alemi et al. (2017a) also propose a related algorithm and empirically show improved resistance to adversarial nuisances. In addition to modifying the cost function, invariance can also be fostered by choice of architecture:

Corollary 3.4 (Bottlenecks promote invariance) *Suppose we have the Markov chain of layers*

$$x \to z_1 \to z_2,$$

and suppose that there is a communication or computation bottleneck between z_1 and z_2 such that $I(z_1;z_2) < I(z_1;x)$. Then, if z_2 is still sufficient, it is more invariant to nuisances than z_1. More precisely, for all nuisances n we have $I(z_2;n) \leq I(z_1;z_2) - I(x;y)$.

Such a bottleneck can happen for example because $\dim(z_2) < \dim(z_1)$, *e.g.*, after a pooling layer, or because the channel between z_1 and z_2 is noisy, *e.g.*, because of dropout.

Proposition 3.5 (Stacking increases invariance) *Assume that we have the Markov chain of layers*

$$x \to z_1 \to z_2 \to \ldots \to z_L,$$

and that the last layer z_L is sufficient of x for y. Then z_L is more insensitive to nuisances than all the preceding layers.

Notice, however, that the above corollary does not simply imply that the more layers the merrier, as it assumes that one has successfully trained the network (z_L is sufficient), which becomes increasingly difficult as the size grows. Also note that in some architectures, such as ResNets (He et al., 2016), the layers do not necessarily form a Markov chain because of skip connections; however, their "blocks" still do.

Proposition 3.6 (Actionable Information) *When $z = f(x)$ is a deterministic invariant, if it minimizes the IB Lagrangian it also maximizes Actionable Information (Soatto, 2013), which is $\mathcal{H}(x) := H(f(x))$.*

Although Soatto (2013) addressed maximal invariants, we only consider sufficient invariants, as advocated by (Soatto and Chiuso, 2016).

Information in the weights

Thus far we have discussed properties of representations in generality, regardless of how they are implemented or learned. Given a source of data (for example randomly generated, or from a fixed dataset), and given a (stochastic) training algorithm, the output weight w of the training process can be thought as a random variable (that depends on the stochasticity of the initialization, training steps and of the data). We can therefore talk about the information that the weights contain about the dataset $\mathcal{D}$ and the training procedure, which we denote by $I(w;\mathcal{D})$.

Two extreme cases consist of the trivial settings where we use the weights to memorize the dataset (the most extreme form of overfitting), or where the weights are constant, or pure noise (sampled from a process that is independent of the data). In between, the amount of information the weights contain about the training turns out to be an important quantity both in training deep networks, as well as in establishing properties of the resulting representation, as we discuss in the next section.

Note that in general we do not need to compute and optimize the quantity of information in the weights. Instead, we show that we can *control* it, for instance by injecting noise in the weights, drawn from a chosen distribution, in an amount that can be modulated between zero (thus in theory allowing full information about the training set to be stored in the weights) to an amount large enough that no information is left. We will leverage this property in the next sections to perform regularization.

4. Learning minimal weights

In this section, we let $p_\theta(x,y)$ be an (unknown) distribution from which we randomly sample a dataset $\mathcal{D}$. The parameter θ of the distribution is also assumed to be a random variable with an (unknown) prior distribution $p(\theta)$. For example p_θ can be a fairly general generative model for natural images, and θ can be the parameters of the model that generated our dataset. We then consider a deep neural network that implements a map $x \mapsto f_w(x) := q(\cdot\,|x,w)$ from an input x to a class distribution $q(y|x,w)$.[1] In full generality, and following a Bayesian approach, we let the weights w of the network be sampled from a parametrized

1. We use p to denote the real (and unknown) data distribution, while q denotes approximate distributions that are optimized during training.

distribution $q(w|\mathcal{D})$,whose parameters are optimized during training.[2] The network is then trained in order to minimize the expected cross-entropy loss[3]

$$H_{p,q}(\mathbf{y}|\mathbf{x},w) = \mathbb{E}_{\mathcal{D}=(\mathbf{x},\mathbf{y})}\mathbb{E}_{w\sim q(w|\mathcal{D})}\sum_{i=1}^{N} -\log q(y^{(i)}|x^{(i)},w),$$

in order for $q(y|x,w)$ to approximate $p_\theta(y|x)$.

One of the main problems in optimizing a DNN is that the cross-entropy loss in notoriously prone to overfitting. In fact, one can easily minimize it even for completely random labels (see Zhang et al. (2017), and Figure 1). The fact that, somehow, such highly over-parametrized functions manage to generalize when trained on real labels has puzzled theoreticians and prompted some to wonder whether this may be inconsistent with the intuitive interpretation of the bias-variance trade-off theorem, whereby unregularized complex models should overfit wildly. However, as we show next, there is no inconsistency if one measures complexity by the information content, and not the dimensionality, of the weights.

To gain some insights about the possible causes of over-fitting, we can use the following decomposition of the cross-entropy loss (we refer to Appendix C for the proof and the precise definition of each term):

$$H_{p,q}(\mathbf{y}|\mathbf{x},w) = \underbrace{H(\mathbf{y}|\mathbf{x},\theta)}_{\text{intrinsic error}} + \underbrace{I(\theta;\mathbf{y}|\mathbf{x},w)}_{\text{sufficiency}} + \mathbb{E}_{\mathbf{x},w}\underbrace{\mathrm{KL}(\, p(\mathbf{y}|\mathbf{x},w) \,\|\, q(\mathbf{y}|\mathbf{x},w)\,)}_{\text{efficiency}} - \underbrace{I(\mathbf{y};w|\mathbf{x},\theta)}_{\text{overfitting}}. \tag{2}$$

The first term of the right-hand side of (8) relates to the intrinsic error that we would commit in predicting the labels even if we knew the underlying data distribution p_θ; the second term measures how much information that the dataset has about the parameter θ is captured by the weights, the third term relates to the efficiency of the model and the class of functions f_w with respect to which the loss is optimized. The last, and only negative, term relates to how much information about the labels, but uninformative of the underlying data distribution, is memorized in the weights. Unfortunately, without implicit or explicit regularization, the network can minimize the cross-entropy loss (LHS), by just maximizing the last term of eq. (8), *i.e.,* by memorizing the dataset, which yields poor generalization.

To prevent the network from doing this, we can neutralize the effect of the negative term by adding it back to the loss function, leading to a regularized loss $L = H_{p,q}(\mathbf{y}|\mathbf{x},w) + I(\mathbf{y};w|\mathbf{x},\theta)$. However, computing, or even approximating, the value of $I(\mathbf{y},w|\mathbf{x},\theta)$ is at least as difficult as fitting the model itself.

We can, however, add an upper bound to $I(\mathbf{y};w|\mathbf{x},\theta)$ to obtain the desired result. In particular, we explore two alternate paths that lead to equivalent conclusions under different premises and assumptions: In one case, we use a PAC-Bayes upper-bound, which is $\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, p(w)\,)$ where $p(w)$ is an arbitrary prior. In the other, we use the IB Lagrangian

2. Note that, while the two are somewhat related, here by $q(w|\mathcal{D})$ we denote the output distribution of the weights after training with our choice algorithm on the dataset $\mathcal{D}$, and not the Bayesian posterior of the weights given the dataset, which would be denoted $p(w|\mathcal{D})$. When $q(w|\mathcal{D})$ is a Dirac delta at a point, we recover the standard loss function for a MAP estimate of the weights.
3. Note that for generality here we treat the dataset $\mathcal{D}$ as a random variable. In practice, when a single dataset is given, the expectation w.r.t. the dataset can be ignored.

and upper-bound it with the information in the weights $I(w;\mathcal{D})$. We discuss this latter approach now, and look at the PAC-Bayes approach in Section 6.

Notice that to successfully learn the distribution p_θ, we only need to memorize in w the information about the latent parameters θ, that is we need $I(\mathcal{D};w) = I(\mathcal{D};\theta) \le H(\theta)$, which is bounded above by a constant. On the other hand, to overfit, the term $I(\mathbf{y};w|\mathbf{x},\theta) \le I(\mathcal{D};w|\theta)$ needs to grow linearly with the number of training samples N. We can exploit this fact to prevent overfitting by adding a Lagrange multiplier β to make the amount of information a constant with respect to N, leading to the regularized loss function

$$\mathcal{L}(q(w|\mathcal{D})) = H_{p,q}(\mathbf{y}|\mathbf{x},w) + \beta I(w;\mathcal{D}), \tag{3}$$

which, remarkably, has the same general form of an IB Lagrangian, and in particular is similar to (1), but now interpreted as a function of the weights w rather than the activations z. This use of the IB Lagrangian is, to the best of our knowledge, novel, as the role of the Information Bottleneck has thus far been confined to characterizing the activations of the network, and not as a learning criterion. Equation (3) can be seen as a generalization of other suggestions in the literature:

IB Lagrangian, Variational Learning and Dropout. Minimizing the information stored at the weights $I(w;\mathcal{D})$ was proposed as far back as Hinton and Van Camp (1993) as a way of simplifying neural networks, but no efficient algorithm to perform the optimization was known at the time. For the particular choice $\beta = 1$, the IB Lagrangian reduces to the variational lower-bound (VLBO) of the marginal log-likelihood $p(\mathbf{y}|\mathbf{x})$. Therefore, minimizing eq. (3) can also be seen as a generalization of variational learning. A particular case of this was studied by Kingma et al. (2015), who first showed that a generalization of Dropout, called Variational Dropout, could be used in conjunction with the *reparametrization trick* Kingma and Welling (2014) to minimize the loss efficiently.

Information in the weights as a measure of complexity. Just as Hinton and Van Camp (1993) suggested, we also advocate using the information regularizer $I(w;\mathcal{D})$ as a measure of the effective complexity of a network, rather than the number of parameters $\dim(w)$, which is merely an upper bound on the complexity. As we show in experiments, this allows us to recover a version of the bias-variance trade-off where networks with lower information complexity underfit the data, and networks with higher complexity overfit. In contrast, there is no clear relationship between number of parameters and overfitting (Zhang et al., 2017). Moreover, for random labels the information complexity allows us to precisely predict the overfitting and underfitting behavior of the network (Section 7).

4.1 Computable upper-bound to the loss

Unfortunately, computing $I(w,\mathcal{D}) = \mathbb{E}_{\mathcal{D}}\,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, q(w)\,)$ is still too complicated, since it requires us to know the marginal $q(w)$ over all possible datasets and trainings of the network. To avoid computing this term, we can use the more general upper-bound

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}\,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, q(w)\,) &\le \mathbb{E}_{\mathcal{D}}\,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, q(w)\,) + \mathrm{KL}(\, q(w) \,\|\, p(w)\,) \\ &= \mathbb{E}_{\mathcal{D}}\,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, p(w)\,),\end{aligned}$$

where $p(w)$ is any fixed distribution of the weights. Once we instantiate the training set, we have a single sample of $\mathcal{D}$, so the expectation over $\mathcal{D}$ becomes trivial. This gives us the following upper bound to the optimal loss function

$$\mathcal{L}(q(w|\mathcal{D})) = H_{p,q}(\mathbf{y}|\mathbf{x}, w) + \beta \,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, p(w)\,) \tag{4}$$

Generally, we want to pick $p(w)$ in order to give the sharpest upper-bound, and to be a fully factorized distribution, *i.e.*, a distribution with independent components, in order to make the computation of the KL term easier. The sharpest upper-bound to $\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, q(w)\,)$ that can be obtained using a factorized distribution p is obtained when $p(w) := \tilde{q}(w) = \prod_i q(w_i)$ where $q(w_i)$ denotes the marginal distributions of the components of $q(w)$. Notice that. once a training procedure is fixed, this may be approximated by training multiple times and approximating each marginal weight distribution. With this choice of prior, our final loss function becomes

$$\mathcal{L}(q(w|\mathcal{D})) = H_{p,q}(\mathbf{y}|\mathbf{x}, w) + \beta \,\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \tilde{q}(w)\,) \tag{5}$$

for some fixed distribution $\tilde{q}$ that approximates the real marginal distribution $q(w)$. The IB Lagrangian for the weights in eq. (3) can be seen as a generally intractable special case of eq. (5) that gives the sharpest upper-bound to our desired loss in this family of losses.

In the following, to keep the notation uncluttered, we will denote our upper bound $\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \tilde{q}(w)\,)$ to the mutual information $I(w;\mathcal{D})$ simply by $\tilde{I}(w;\mathcal{D})$, where

$$\tilde{I}(w;\mathcal{D}) := \mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \tilde{q}(w)\,) = \mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \textstyle\prod_i q(w_i)\,).$$

4.2 Bounding the information in the weights of a network

To derive precise and empirically verifiable statements about $\tilde{I}(w;\mathcal{D})$, we need a setting where this can be expressed analytically and optimized efficiently on standard architectures. To this end, following Kingma et al. (2015), we make the following modeling choices.

Modeling assumptions. Let w denote the vector containing all the parameters (weights) in the network, and let W^k denote the weight matrix at layer k. We assume an improper log-uniform prior on w, that is $\tilde{q}(w_i) = c/|w_i|$. Notice that this is the only scale-invariant prior (Kingma et al., 2015), and closely matches the real marginal distributions of the weights in a trained network (Achille and Soatto, 2018); we parametrize the weight distribution $q(w_i|\mathcal{D})$ during training as

$$w_i|\mathcal{D} \sim \epsilon_i \hat{w}_i,$$

where $\hat{w}_i$ is a learned mean, and $\epsilon_i \sim \log \mathcal{N}(-\alpha_i/2, \alpha_i)$ is i.i.d. multiplicative log-normal noise with mean 1 and variance $\exp(\alpha_i) - 1$.[4] Note that while Kingma et al. (2015) uses this parametrization as a local approximation of the Bayesian posterior for a given (log-uniform) prior, we rather *define* the distribution of the weights w after training on the dataset $\mathcal{D}$ to be $q(w|\mathcal{D})$.

4. For a log-normal $\log \mathcal{N}(\mu, \sigma^2)$ mean and variance are respectively $\exp(\mu + \sigma^2/2)$ and $[\exp(\sigma^2) - 1]\exp(2\mu + \sigma^2)$.

Proposition 4.1 (Information in the weights, Theorem C.4) *Under the previous modeling assumptions, the upper-bound to the information that the weights contain about the dataset is*

$$I(w;\mathcal{D}) \le \tilde{I}(w;\mathcal{D}) = -\frac{1}{2} \sum_{i=1}^{\dim(w)} \log \alpha_i + C,$$

where the constant C is arbitrary due to the improper prior.

Remark 4.2 (On the constant C) *To simplify the exposition, since the optimization is unaffected by any additive constant, in the following we abuse the notation and, under the modeling assumptions stated above, we rather* define $\tilde{I}(w;\mathcal{D}) := -\frac{1}{2}\sum_{i=1}^{\dim(w)} \log \alpha_i$. *Nekludov et al. (2017) also suggest a principled way of dealing with the arbitrary constant by using a proper log-uniform prior.*

Note that computing and optimizing this upper-bound to the information in the weights is relatively simple and efficient using the reparametrization trick of Kingma et al. (2015).

4.3 Flat minima have low information

Thus far we have suggested that adding the explicit information regularizer $I(w;\mathcal{D})$ prevents the network from memorizing the dataset and thus avoid overfitting, which we also confirm empirically in Section 7. However, real networks are not commonly trained with this regularizer, thus seemingly undermining the theory. However, even when not explicitly present, the term $I(w;\mathcal{D})$ is implicit in the use of SGD. In particular, Chaudhari and Soatto (2018) show that, under certain conditions, SGD introduces an entropic bias of a very similar form to the information in the weights described thus far, where the amount of information can be controlled by the learning rate and the size of mini-batches.

Additional indirect empirical evidence is provided by the fact that some variants of SGD (Chaudhari et al., 2017) bias the optimization toward "flat minima", that are local minima whose Hessian has mostly small eigenvalues. These minima can be interpreted exactly as having low information $I(w;\mathcal{D})$, as suggested early on by Hochreiter and Schmidhuber (1997): Intuitively, since the loss landscape is locally flat, the weights may be stored at lower precision without incurring in excessive inference error. As a consequence of previous claims, we can then see flat minima as having better generalization properties and, as we will see in Section 5, the associated representation of the data is more insensitive to nuisances and more disentangled. For completeness, here we derive a more precise relationship between flatness (measured by the nuclear norm of the loss Hessian), and the information content based on our model.

Proposition 4.3 (Flat minima have low information, Appendix C.5) *Let $\hat{w}$ be a local minimum of the cross-entropy loss $H_{p,q}(\mathbf{y}|\mathbf{x}, w)$, and let $\mathcal{H}$ be the Hessian at that point. Then, for the optimal choice of the posterior $w|\mathcal{D} = \epsilon \odot \hat{w}$ centered at $\hat{w}$ that optimizes the IB Lagrangian, we have*

$$I(w;\mathcal{D}) \le \tilde{I}(w;\mathcal{D}) \le \frac{1}{2} K[\log \|\hat{w}\|_2^2 + \log \|\mathcal{H}\|_* - K \log(K^2 \beta/2)]$$

where $K = \dim(w)$ and $\|\cdot\|_*$ denotes the nuclear norm.

Notice that a converse inequality, that is, low information implies flatness, needs not hold, so there is no contradiction with the results of Dinh et al. (2017). Also note that for $\tilde{I}(w;\mathcal{D})$ to be invariant to reparametrization one has to consider the constant C, which we have ignored (Remark 4.2). The connection between flatness and overfitting has also been studied by Neyshabur et al. (2017), including the effect of the number of parameters in the model.

In the next section, we prove one of our main results, that networks with low information in the weights realize invariant and disentangled representations. Therefore, invariance and disentanglement emerge naturally when training a network with implicit (SGD) or explicit (IB Lagrangian) regularization, and are related to flat minima.

5. Duality of the Bottleneck

The following proposition gives the fundamental link in our model between information in the weights, and hence flatness of the local minima, minimality of the representation, and disentanglement.

Proposition 5.1 (Appendix C.6) *Let $z = Wx$, and assume as before $W = \epsilon \odot \hat{W}$, with $\epsilon_{i,j} \sim \log\mathcal{N}(-\alpha_i/2, \alpha_i)$. Further assume that the marginals of $p(z)$ and $p(z|x)$ are both approximately Gaussian (which is reasonable for large $\dim(x)$ by the Central Limit Theorem). Then,*

$$I(z;x) + \mathrm{TC}(z) = -\frac{1}{2}\sum_{i=1}^{\dim(z)} \mathbb{E}_x \log \frac{\tilde{\alpha}_i \hat{W}_i^2 \cdot x^2}{\hat{W}_i \cdot \mathrm{Cov}(x)\hat{W}_i + \tilde{\alpha}_i \hat{W}_i^2 \cdot \mathbb{E}(x^2)}, \tag{6}$$

where W_i denotes the i-th row of the matrix W, and $\tilde{\alpha}_i$ is the noise variance $\tilde{\alpha}_i = \exp(\alpha_i) - 1$. In particular, $I(z;x) + \mathrm{TC}(z)$ is a monotone decreasing function of the weight variances α_i.

The above identity is difficult to apply in practice, but with some additional hypotheses, we can derive a cleaner uniform tight bound on $I(z;x) + \mathrm{TC}(z)$.

Proposition 5.2 (Uniform bound for one layer, Appendix C.7) *Let $z = Wx$, where $W = \epsilon \odot \hat{W}$, where $\epsilon_{i,j} \sim \log\mathcal{N}(-\alpha/2, \alpha)$; assume that the components of x are uncorrelated, and that their kurtosis is uniformly bounded.*[5] *Then, there is a strictly increasing function $g(\alpha)$ s.t. we have the uniform bound*

$$g(\alpha) \leq \frac{I(x;z) + TC(z)}{\dim(z)} \leq g(\alpha) + c,$$

where $c = O(1/\dim(x)) \leq 1$, $g(\alpha) = -\log(1 - e^{-\alpha})/2$ and α is related to $\tilde{I}(w;\mathcal{D})$ by $\alpha = \exp\{-I(W;\mathcal{D})/\dim(W)\}$. In particular, $I(x;z) + TC(z)$ is tightly bounded by $\tilde{I}(W;\mathcal{D})$ and increases strictly with it.

5. This is a technical hypothesis, always satisfied if the components x_i are IID, (sub-)Gaussian, or with uniformly bounded support.

The above theorems tells us that whenever we decrease the information in the weights, either by explicit regularization, or by implicit regularization (*e.g.*, using SGD), we automatically improve the minimality, and hence, by Proposition 3.1, the invariance, and the disentanglement of the learner representation. In particular, we obtain as a corollary that SGD is biased toward learning invariant and disentangled representations of the data. Using the Markov property of the layers, we can easily extend this bound to multiple layers:

Corollary 5.3 (Multi-layer case, Appendix C.8) *Let W^k for $k = 1, ..., L$ be weight matrices, with $W^k = \epsilon^k \odot \hat{W}^k$ and $\epsilon^k_{i,j} = \log\mathcal{N}(-\alpha^k/2, \alpha^k)$, and let $z_{i+1} = \phi(W^k z_k)$, where $z_0 = x$ and ϕ is any nonlinearity. Then,*

$$I(z_L;x) \leq \min_{k<L} \Big\{ \dim(z_k)\big[g(\alpha^k) + 1\big] \Big\}$$

where $\alpha^k = \exp\big\{-I(W^k;\mathcal{D})/\dim(W^k)\big\}$.

Remark 5.4 (Tightness) While the bound in Proposition 5.2 is tight, the bound in the multilayer case needs not be. This is to be expected: Reducing the information in the weights creates a bottleneck, but we do not know how much information about x will actually go through this bottleneck. Often, the final layers will let most of the information through, while initial layers will drop the most.

Remark 5.5 (Training-test transfer) We note that we did not make any (explicit) assumption about the test set having the same distribution of the training set. Instead, we make the less restrictive assumption of sufficiency: If the test distribution is entirely different from the training one – one may not be able to achieve sufficiency. This prompts interesting questions about measuring the distance between *tasks* (as opposed to just distance between distributions), which will be studied in future work.

6. Connection with PAC-Bayes bounds

In this section we show that using a PAC-Bayes bound, we arrive at the same regularized loss function eq. (5) we obtained using the Information Bottleneck, without the need of any approximation. By Theorem 2 of McAllester (2013), we have that for any fixed $\lambda > 1/2$, prior $p(w)$, and any weight distribution $q(w|\mathcal{D})$, the test error $L^{\text{test}}(q(w|\mathcal{D}))$ that the network commits using the weight distribution $q(w|\mathcal{D})$ is upper-bounded in expectation by

$$\mathbb{E}_{\mathcal{D}}[L^{\text{test}}(q(w|\mathcal{D}))] \leq \frac{1}{N(1-\frac{1}{2\lambda})}\Big(H_{p,q}(\mathbf{y}|\mathbf{x},w) + \lambda L_{\max}\mathbb{E}_{\mathcal{D}}[\mathrm{KL}(\,q(w|\mathcal{D})\,\|\,p(w)\,)]\Big), \tag{7}$$

where L_{max} is the maximum per-sample loss function, which for a classification problem we can assume to be upper-bounded, for example by clipping the cross-entropy loss at chance level. Notice that right hand side coincides, modulo a multiplicative constant, with eq. (4) that we derived as an approximation of the IB Lagrangian for the weights (eq. (3)).

Now, recall that since we have

$$\begin{aligned}\mathbb{E}_{\mathcal{D}}[\mathrm{KL}(\,q(w|\mathcal{D})\,\|\,q(w)\,)] &= \mathbb{E}_{\mathcal{D}}[\mathrm{KL}(\,q(w|\mathcal{D})\,\|\,p(w)\,)] - \mathrm{KL}(\,q(w)\,\|\,p(w)\,)\\ &\leq \mathbb{E}_{\mathcal{D}}[\mathrm{KL}(\,q(w|\mathcal{D})\,\|\,p(w)\,)],\end{aligned}$$

the sharpest PAC-Bayes upper-bound to the test error is obtained when $p(w) = q(w)$, in which case eq. (7) reduces (modulo a multiplicative constant) to the IB Lagrangian of the weights. That is, the IB Lagrangian for the weights can be considered as a special case of PAC-Bayes giving the sharpest bound.

Unfortunately, as we noticed in Section 4, the joint marginal $q(w)$ of the weights is not tractable. To circumvent the problem, we can instead consider that the sharpest PAC-Bayes upper-bound that can be obtained using a tractable factorized prior $p(w)$, which is obtained exactly when $p(w) = \tilde{q}(w) = \prod_i q(w_i)$ is the product of the marginals, leading again to our practical loss eq. (5).

On a last note, recall that under our modeling assumptions the marginal $\tilde{q}(w)$ is assumed to be an improper log-uniform distribution. While this has the advantage of being a non-informative prior that closely matches the real marginal of the weights of the network, it also has the disadvantage that it is only defined modulo an additive constant, therefore making the bound on the test error vacuous under our model.

The PAC-Bayes bounds has also been used by Dziugaite and Roy (2017) to study the generalization property of deep neural networks and their connection with the optimization algorithm. They use a Gaussian prior and posterior, leading to a non-vacuous generalization bound.

7. Empirical validation

7.1 Transition from overfitting to underfitting

As pointed out by Zhang et al. (2017), when a standard convolutional neural network (CNN) is trained on CIFAR-10 to fit random labels, the network is able to (over)fit them perfectly. This is easily explained in our framework: It means that the network is complex enough to memorize all the labels but, as we show here, it has to pay a steep price in terms of information complexity of the weights (Figure 2) in order to do so. On the other hand, when the information in the weights is bounded using and information regularizer, overfitting is prevented in a theoretically predictable way.

In particular, in the case of completely random labels, we have $I(\mathbf{y}; w|\mathbf{x}, \theta) = I(\mathbf{y}; w) \leq I(w; \mathcal{D})$, where the first equality holds since $\mathbf{y}$ is by construction random, and therefore independent of $\mathbf{x}$ and θ. In this case, the inequality used to derive eq. (3) is an equality, and the IBL is an optimal regularizer, and, regardless of the dataset size N, for $\beta > 1$ it should completely prevent memorization, while for $\beta < 1$ overfitting is possible. To see this, notice that since the labels are random, to decrease the classification error by $\log |\mathcal{Y}|$, where $|\mathcal{Y}|$ is the number of possible classes, we need to memorize a new label. But to do so, we need to store more information in the weights of the network, therefore increasing the second term $I(w; \mathcal{D})$ by a corresponding quantity. This trade-off is always favorable when $\beta < 1$, but it is not when $\beta > 1$. Therefore, the theoretically the optimal solution to *eq.* (1) is to memorize all the labels in the first case, and not memorize anything in the latter.

As discussed, for real neural networks we cannot directly minimize *eq.* (1), and we need to use a computable upper bound to $I(w; \mathcal{D})$ instead (Section 4.2). Even so, the empirical behavior of the network, shown in Figure 1, closely follows this prediction, and for various sizes of the dataset clearly shows a phase transition between overfitting and underfitting near the critical value $\beta = 1$. Notice instead that for real labels the situation is different:

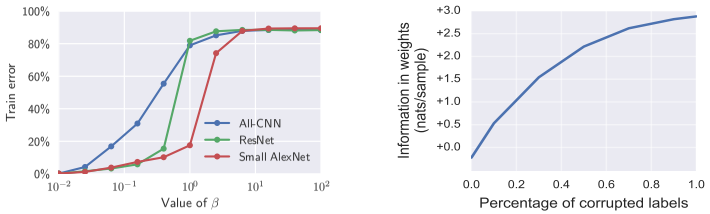


Figure 2: **(Left)** Plot of the training error on CIFAR-10 with random labels as a function of the parameter β for different models (see the appendix for details). As expected, all models show a sharp phase transition from complete overfitting to underfitting before the critical value $\beta = 1$. **(Right)** We measure the quantity of information in the weights necessary to overfit as we vary the percentage of corrupted labels under the same settings of Figure 1. To fit increasingly random labels, the network needs to memorize more information in the weights; the increase needed to fit entirely random labels is about the same magnitude as the size of a label (2.30 nats/sample).

The model is still able to overfit when $\beta < 1$, but importantly there is a large interval of $\beta > 1$ where the model can fit the data *without* overfitting to it. Indeed, as soon as $\beta N \propto I(w; \mathcal{D})$ is larger than the constant $H(\theta)$, the model trained on real data fits real labels without excessive overfitting (Figure 1).

Notice that, based on this reasoning, we expect the presence of a phase transition between an overfitting and an underfitting regime at the critical value $\beta = 1$ to be largely independent on the network architecture: To verify this, we train different architectures on a subset of 10000 samples from CIFAR-10 with random labels. As we can see on the left plot of Figure 2, even very different architectures show a phase transition at a similar value of β. We also notice that in the experiment ResNets has a sharp transition close to the critical β.

In the right plot of Figure 2 we measure the quantity information in the weights for different levels of corruption of the labels. To do this, we fix $\beta < 1$ so that the network is able to overfit, and for various level of corruption we train until convergence, and then compute $I(w; \mathcal{D})$ for the trained model. As expected, increasing the randomness of the labels increases the quantity of information we need to fit the dataset. For completely random labels, $I(w; \mathcal{D})$ increases by ~ 3 nats/sample, which the same order of magnitude as the quantity required to memorize a 10-class labels (2.30 nats/sample), as shown in Figure 2.

7.2 Bias-variance trade-off

The Bias-Variance trade-off is sometimes informally stated as saying that low-complexity models tend to underfit the data, while excessively complex models may instead overfit, so that one should select an adequate intermediate complexity. This is apparently at odds with the common practice in Deep Learning, where increasing the depth or the number of weights of the network, and hence increasing the "complexity" of the model measured by

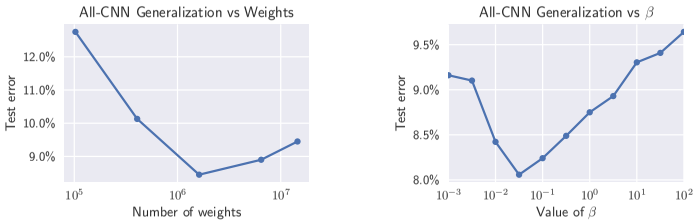


Figure 3: Plots of the test error obtained training the All-CNN architecture on CIFAR-10 (no data augmentation). **(Left)** Test error as we increase the number of weights in the network using weight decay but without any additional explicit regularization. Notice that increasing the number of weights the generalization error plateaus rather than increasing. **(Right)** Changing the value of β, which controls the amount of information in the weights, we obtain the characteristic curve of the bias-variance trade-off. This suggests that the quantity of information in the weights correlates well with generalization.

the number of parameters, does not seem to induce overfitting. Consequently, a number of alternative measures of complexity have been proposed that capture the intuitive bias-variance trade-off curve, such as different norms of the weights (Neyshabur et al., 2015).

From the discussion above, we have seen that the quantity of information in the weights, or alternatively its computable upperbound $\tilde{I}(w;\mathcal{D})$, also provides a natural choice to measure model complexity in relation to overfitting. In particular, we have already seen that models need to store increasingly more information to fit increasingly random labels (Figure 2). In Figure 3 we show that by controlling $\tilde{I}(w;\mathcal{D})$, which can be done easily by modulating β, we recover the right trend for the bias-variance tradeoff, whereas models with too little information tend to underfit, while models memorizing too much information tend to overfit.

7.3 Nuisance invariance

Corollary 5.3 shows that by decreasing the information in the weights $I(w;\mathcal{D})$, which can be done for example using eq. (3), the learned representation will be increasingly minimal, and therefore insensitive to nuisance factors n, as measured by $I(z;n)$. Here, we adapt a technique from the GAN literature Sønderby et al. (2017) that allows us to explicitly measure $I(z;n)$ and validate this effect, provided we can sample from the nuisance distribution $p(n)$ and from $p(x|n)$; that is, if given a nuisance n we can generate data x affected by that nuisance. Recall that by definition we have

$$\begin{aligned} I(z;n) &= \mathbb{E}_{n\sim p(n)}\,\mathrm{KL}(\,p(z|n)\,\|\,p(z)\,) \\ &= \mathbb{E}_{n\sim p(n)}\mathbb{E}_{z\sim p(z|n)}\log[p(z|n)/p(z)]. \end{aligned}$$

To approximate the expectations via sampling we need a way to approximate the likelihood ratio $\log p(\mathbf{z}|\mathbf{n})/p(\mathbf{z})$. This can be done as follows: Let $D(z;n)$ be a binary discriminator that given the representation z and the nuisance n tries to decide whether z is sampled

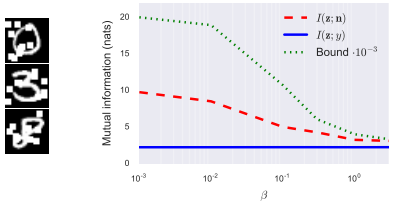


Figure 4: **(Left)** A few training samples generated adding nuisance clutter n to the MNIST dataset. **(Right)** Reducing the information in the weights makes the representation z learned by the digit classifier increasingly invariant to nuisances ($I(n;z)$ decreases), while sufficiency is retained ($I(z;y) = I(x;y)$ is constant). As expected, $I(z;n)$ is smaller but has a similar behavior to the theoretical bound in Theorem 5.3.

from the posterior distribution $p(z|n)$ or from the prior $p(z)$. Since by hypothesis we can generate samples from both distributions, we can generate data to train this discriminator. Intuitively, if the discriminator is not able to classify, it means that z is insensitive to changes of n. Precisely, since the optimal discriminator is

$$D^*(z;n) = \frac{p(z)}{p(z)+p(z|n)},$$

if we assume that D is close to the optimal discriminator D^*, we have

$$\log\frac{p(z|n)}{p(z)} = \log\frac{1-D^*(z;n)}{D^*(z;n)} \simeq \log\frac{1-D(z;n)}{D(z;n)}.$$

therefore we can use D to estimate the log-likelihood ratio, and so also the mutual information $I(z;n)$. Notice however that this comes with no guarantees on the quality of the approximation.

To test this algorithm, we add random occlusion nuisances to MNIST digits (Figure 4). In this case, the nuisance n is the occlusion pattern, while the observed data x is the occluded digit. For various values of β, we train a classifier on this data in order to learn a representation z, and, for each representation obtained this way, we train a discriminator as described above and we compute the resulting approximation of $I(z;n)$. The results in Figure 4 show that decreasing the information in the weights makes the representation increasingly more insensitive to n.

8. Discussion and conclusion

In this work, we have presented bounds, some of which are tight, that connect the amount of information in the weights, the amount of information in the activations, the invariance property of the network, and the geometry of the residual loss. These results leverage the structure of deep networks, in particular the multiplicative action of the weights, and the Markov property of the layers. This leads to the surprising result that reducing information

stored in the weights about the past (dataset) results in desirable properties of the learned internal representation of the test datum (future).

Our notion of representation is intrinsically stochastic. This simplifies the computation as well as the derivation of information-based relations. However, note that even if we start with a deterministic representation w, Proposition 4.3 gives us a way of converting it to a stochastic representation whose quality depends on the flatness of the minimum. Our theory uses, but does not depend on, the Information Bottleneck Principle, which dates back to over two decades ago, and can be re-derived in a different frameworks, for instance PAC-Bayes, which yield the same results and additional bounds on the test error.

This work focuses on the inference and learning of optimal representations, that seek to get the most out of the data we have for a specific task. This does not guarantee a good outcome since, due to the Data Processing Inequality, the representation can be easier to use but ultimately no more informative than the data themselves. An orthogonal but equally interesting issue is how to get the most informative data possible, which is the subject of active learning, experiment design, and perceptual exploration. Our work does not address transfer learning, where a representation trained to be optimal for a task is instead used for a different task, which will be subject of future investigations.

Acknowledgments

Supported by ONR N00014-17-1-2072, ARO W911NF-17-1-0304, AFOSR FA9550-15-1-0229 and FA8650-11-1-7156. We wish to thank our reviewers and David McAllester, Kevin Murphy, Alessandro Chiuso for the many insightful comments and suggestions.

References

Alessandro Achille and Stefano Soatto. Information dropout: Learning optimal representations through noisy computation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, PP(99):1–1, 2018.

Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017a.

Alexander A. Alemi, Ben Poole, Ian Fischer, Joshua V. Dillon, Rif A. Saurous, and Kevin Murphy. Fixing a Broken ELBO. *ArXiv e-prints*, November 2017b.

Fabio Anselmi, Lorenzo Rosasco, and Tomaso Poggio. On invariance and selectivity in representation learning. *Information and Inference*, 5(2):134–158, 2016.

Zhaojun Bai, Gark Fahey, and Gene Golub. Some large-scale matrix computation problems. *Journal of Computational and Applied Mathematics*, 74(1-2):71–89, 1996.

Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends in Machine Learning*, 2(1):1–127, 2009.

Sterling K. Berberian. Borel spaces, April 1988.

Joan Bruna and Stéphane Mallat. Classification with scattering operators. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1561–1566, 2011.

Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*, 2015.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017.

Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *Proc. Uncertainty in Artificial Intelligence (UAI)*, 2017.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Ulf Grenander. *General Pattern Theory*. Oxford University Press, 1993.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

Geoffrey E Hinton and Drew Van Camp. Keeping the neural networks simple by minimizing the description length of the weights. In *Proceedings of the 6th annual conference on Computational learning theory*, pages 5–13. ACM, 1993.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

Diederik P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In *Advances in Neural Information Processing Systems 28*, pages 2575–2583, 2015.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *Technical report, University of Toronto*, 2009.

Yann LeCun. Learning invariant feature hierarchies. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 496–505, 2012.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Holden Lee, Rong Ge, Andrej Risteski, Tengyu Ma, and Sanjeev Arora. On the ability of neural nets to express distributions. In *Proceedings of Machine Learning Research*, volume 65, pages 1–26, 2017.

David McAllester. A pac-bayesian tutorial with a dropout bound. *arXiv preprint arXiv:1307.2118*, 2013.

Dmitry Molchanov, Arsenii Ashukha, and Dmitry Vetrov. Variational dropout sparsifies deep neural networks. In *Proceedings of the 34 th International Conference on Machine Learning (ICML)*, 2017.

Kirill Neklyudov, Dmitry Molchanov, Arsenii Ashukha, and Dmitry P Vetrov. Structured bayesian pruning via log-normal multiplicative noise. In *Advances in Neural Information Processing Systems 30*, pages 6775–6784. Curran Associates, Inc., 2017.

Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *Advances in Neural Information Processing Systems*, pages 2422–2430, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pages 5949–5958, 2017.

Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Ravid Shwartz-Ziv and Naftali Tishby. Opening the black box of deep neural networks via information. *arXiv preprint arXiv:1703.00810*, 2017.

Stefano Soatto. Actionable information in vision. In *Machine learning for computer vision*. Springer, 2013.

Stefano Soatto and Alessandro Chiuso. Visual representations: Defining properties and deep approximations. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.

Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

Ganesh Sundaramoorthi, Peter Petersen, V. S. Varadarajan, and Stefano Soatto. On the set of images modulo viewpoint and contrast changes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *Information Theory Workshop (ITW), 2015 IEEE*, pages 1–5. IEEE, 2015.

Naftali Tishby, Fernando C Pereira, and William Bialek. The information bottleneck method. In *The 37th annual Allerton Conference on Communication, Control, and Computing*, pages 368–377, 1999.

Greg Ver Steeg and Aram Galstyan. Maximally informative hierarchical representations of high-dimensional data. In *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics*, 2015.

Shuo Yang, Ping Luo, Chen-Change Loy, and Xiaoou Tang. From facial parts responses to face detection: A deep learning approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3676–3684, 2015.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Appendix A. Details of the experiments

A.1 Random labels

We use a similar experimental setup as Zhang et al. (2017). In particular, we train a small version of AlexNet on a 28×28 central crop of CIFAR-10 with completely random labels. The dataset is normalized using the global channel-wise mean and variance, but no additional data augmentation is performed. The exact structure of the network is in Table 1. As common in practice we use batch normalization before all the ReLU nonlinearities, except for the first layer. Optimization of the IB Lagrangian loss function is performed similarly to Kingma et al. (2015) and Molchanov et al. (2017). We found that constraining the variance α_i of the weights to be the same for all weights in the same filter helps stabilizing the training process. We train with learning rates $\eta \in \{0.02, 0.005\}$ and select the best performing network of the two. Generally, we found that a higher learning rate is needed to overfit when the number of training samples N is small, while a lower learning rate is needed for larger N. We train with SGD with momentum 0.9 for 360 epochs reducing the learning rate by a factor of 10 every 140 epochs. We use a large batch-size of 500 to minimize the noise coming from SGD. No weight decay or other regularization methods are used.

The final plot is obtained by triangulating the convex envelope of the data points, and by interpolating their value on the resulting simplexes. Outside of the convex envelope (where the accuracy is mostly constant), the value was obtained by inpainting.

To measure the information content of the weights as the percentage of corrupted labels varies, we fix $\beta = 0.1$, $N = 30000$ and $\eta = 0.005$ and train the network on different corruption levels with the same settings as before.

To test the phase transition on multiple architectures, we train the Small AlexNet, the AllCNN network and a ResNet (see Table 1). For all architectures, we train with $N = 10000$ random labels, $\eta = 0.05$ and different values of β log-uniformly spaced in $[10^{-2}, 10^{2}]$.

A.2 Bias-variance trade-off

For this experiment we train the AllCNN architecture (Table 1) on the CIFAR-10 dataset with ZCA whitening Krizhevsky and Hinton (2009) and without any additional data augmentation. First, we train a standard network and change the number of filters (we multiplying the number of filters of all layers by the same constant) and train with $\eta = 0.05$, batch size 128, weight decay 0.001. Then, we use the standard number of layers and train instead with the IBL loss function with different values of β.

A.3 Nuisance invariance

The cluttered MNIST dataset is generated by adding ten 4×4 squares uniformly at random on the digits of the MNIST dataset (LeCun et al., 1998). For each level of β, we train the classifier in Table 1 on this dataset. The weights of all layers, excluding the first and last one, are threated as a random variable with multiplicative Gaussian noise (Appendix B) and optimized using the local reparameterization trick of Kingma et al. (2015). We use the last convolutional layer before classification as representation z.

The discriminator network used to estimate the log-likelihood ratio is constructed as follows: the inputs are the nuisance pattern n, which is a 28×28×1 image containing 10 random

Input 32x32
conv 64 ReLU
MaxPool 2x2
conv 64 + BN ReLU
MaxPool 2x2
FC 3136x384 + BN ReLU
FC 384x192 + BN ReLU
FC 192x10
softmax

Input 28x28
conv 96 + BN + ReLU conv 96 + BN + ReLU
conv 192 s2 + BN + ReLU
conv 192 + BN + ReLU conv 192 + BN + ReLU
conv 192 s2 + BN + ReLU
conv 192 + BN + ReLU conv 192 + BN + ReLU
conv 1x1x10 Average pooling 7x7
softmax

Input 28x28
conv 64
block 64 s1
block 128 s2
block 256 s3
block 512 s3
Average pooling 4x4 linear 10
softmax

Table 1: **(Left)** The Small AlexNet model used in the random label experiment, adapted from Zhang et al. (2017). All convolutions have a 5×5 kernel. The use of batch normalization makes the training procedure more stable, but did not significantly change the results of the experiments. **(Center)** All Convolutional Network (Springenberg et al., 2014) used as a classifier in the experiments. All convolutions but the last one use a 3×3 kernel, "s2" denotes a convolution with stride 2. The final representation we use are the activations of the last "conv 192" layer. **(Right)** The ResNet architecture (He et al., 2016) on which we test the phase transition. Each block with `f` filters and stride s is structured as follows: `BN -> ReLU -> conv f stride s -> BN -> ReLU -> conv f` with a skip connection between first ReLU and the output.

occluding squares, and the 7×7×192 representation z obtained from the classifier. First we preprocess n using the following network: `conv 48` → `conv 48` → `conv 96 s2` → `conv 96` → `conv 96` → `conv 96 s2`, where each conv block is a 3×3 convolution followed by batch normalization and ReLU. Then, we concatenate the 7×7×96 result with z along the feature maps, and the final discriminator output is obtained by applying the following network: `conv 192` → `conv 192` → `conv 1×1×192` → `conv 1×1×1` → `AvgPooling 7×7` → `sigmoid`.

A.4 Visualizing the representation

Even when we cannot generate data affected by nuisances like in the previous section, we can still visualize the information content of z to learn what nuisances are discarded in the representation. To this end, given a representation z, we want to learn to sample from a distribution $q(\hat{x}|z)$ of images that are maximally likely to have z as their representation. Formally, this means that we want a distribution $q(\hat{x}|z)$ that maximizes the amortized maximum a posteriori estimate of z:

$$\mathbb{E}_z \mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(\hat{x}|z)] = \mathbb{E}_z \underbrace{\mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(z|\hat{x})]}_{\text{Reconstruction error}} + \underbrace{\mathbb{E}_{\hat{x}\sim q(\hat{x})}[\log p(\hat{x})]}_{\text{Distance from prior}} + C.$$

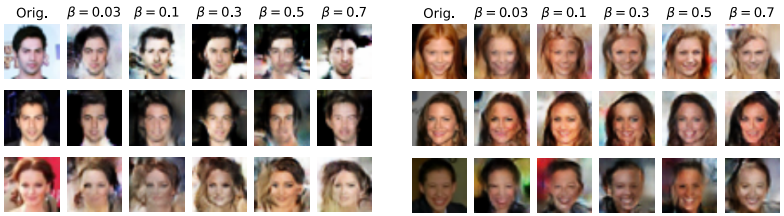


Figure 5: For different values of β, we show the image $\hat{x}$ reconstructed from a representation $z \sim p(z|x)$ of the original image x in the first column. For small β, z contains more information regarding x, thus the reconstructed image $\hat{x}$ is close to x, background included. Increasing β decreases the information in the weighs, thus the representation z becomes more invariant to nuisances: Reconstructed image matches important details in x that are preserved in z (*i.e.*, hair color, sex, expression), but background, hair style, and other nuisances are generated anew.

Unfortunately, the term $p(\hat{x})$ in the expression is difficult to estimate. However, Sønderby et al. (2017) notice that the modified gain function

$$\mathbb{E}_z\mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(\hat{x}|z)] \,+\, H(p(\hat{x})) \;\;=\;\; \mathbb{E}_z\mathbb{E}_{\hat{x}\sim q(\hat{x}|z)}[\log p(z|\hat{x})] \,-\, \mathrm{KL}(\, q(\hat{x}) \,\|\, p(\hat{x})\,) \,+\, C,$$

differs from the amortizes MAP only by a term $H(p(\hat{\mathbf{x}}))$, which has the positive effect of improving the exploration of the reconstruction, and contains the term $\mathrm{KL}(\, q(\hat{x}) \,\|\, p(\hat{x})\,)$, which can be estimated easily using the discriminator network of a GAN Sønderby et al. (2017). To maximize this gain, we can simply train a GAN with an additional reconstruction loss $-\log p(z|\hat{x})$.

To test this algorithm, we train a representation z to classify the 40 binary attributes in the CelebA face dataset (Yang et al., 2015), and then use the above loss function to train a GAN network to reconstruct an input image $\hat{x}$ from the representation z. The results in Figure 5 show that, as expected, increasing the value of β, and therefore reducing $I(w;\mathcal{D})$, generates samples that have increasingly more random backgrounds and hair style (nuisances), while retaining facial features. In other words, the representation z is increasingly insensitive to nuisances affecting the data, while information pertaining the task is retained in the reconstruction $\hat{x}$.

More precisely, we first train a classifier on the images from the CelebA datasets resized to 32×32, where the task is to recover the 40 binary attributes associated to each image. The classifier network is the same as the one in Table 1 with the following modifications: we use Exponential Linear Units (Clevert et al., 2015) for the activations, instead of ReLU, since invertible activations generally perform better when training a GAN, and we divide by two the number of output filters in all layers to reduce the training time. A sigmoid nonlinearity is applied to the final 40-way output of the network.

To generate the image $\hat{x}$ given the 8×8×96 representation z computed by the classifier, we use a similar structure to DCGAN (Radford et al., 2016), namely $z \rightarrow$ `conv 256` $\rightarrow$ `ConvT 256s2` $\rightarrow$ `ConvT 128s2` $\rightarrow$ `conv 3` $\rightarrow$ `tanh`, where `ConvT 256s2` denotes a transpose convolution with 256 feature maps and stride 2. All convolutions have a batch

normalization layer before the activations. Finally, the discriminator network is given by $\hat{x}$ $\rightarrow$ `conv 64s2` $\rightarrow$ `conv 128s2` $\rightarrow$ `ConvT 256s2` $\rightarrow$ `conv 1` $\rightarrow$ `sigmoid`. Here, all convolutions use batch normalization followed by Leacky ReLU activations.

In this experiment, we use Gaussian multiplicative noise which is slightly more stable during training (Appendix B). To stabilize the training of the GAN, we found useful to (1) scale down the "reconstruction error" term in the loss function and (2) slowly increase the weight of the reconstruction error up to the desired value during training.

Appendix B. Gaussian multiplicative noise

In developing the theory, we chose to use log-normal multiplicative noise for the weights: The main benefit is that with this choice the information in the weights $I(w;\mathcal{D})$ can be expressed in closed form, up to an arbitrary constant C which does not matter during the optimization process (but see also Neklyudov et al. (2017) for a principled approach to this problem that uses a proper log-uniform prior). Another possibility, suggested by Kingma et al. (2015) is to use Gaussian multiplicative noise with mean 1. Unfortunately, there is no analytical expression for $I(w;\mathcal{D})$ when using Gaussian noise, but $I(w;\mathcal{D})$ can still be approximated numerically with high precision (Molchanov et al., 2017), and it makes the training process slightly more stable. The theory holds with minimal changes also in this case, and we use this choice in some experiments.

Appendix C. Proofs of theorems

Lemma C.1 (Task-nuisance decomposition) *Given a joint distribution $p(x,y)$, where y a discrete random variable, we can always find a random variable n independent of y such that $x = f(y,n)$, for some deterministic function f.*

Proof Fix $n \sim \mathrm{Uniform}(0,1)$ to be the uniform distribution on $[0,1]$. We claim that, for a fixed value of y, there is a function $\Phi_y(n)$ such that $x|y = \Phi_{y*}(n)$, where $(\cdot)_*$ denotes the push-forward map of measures. Given the claim, let $\Phi(y,n) = (y, \Phi_y(n))$. Since y is a discrete random variable, $\Phi(y,n)$ is easily seen to be a measurable function and by construction $(x,y) \sim \Phi_*(y,n)$. To see the claim, notice that, since there exists a measurable isomorphism between $\mathbb{R}^n$ and $\mathbb{R}$ (Theorem 3.1.1 of Berberian (1988)), we can assume without loss of generality that $x \in \mathbb{R}$. In this case, by definition, we can take $\Phi_y(n) = F_y^{-1}(n)$ where $F_y(t) = \mathbb{P}[x < t \,|\, y]$ is the cumulative distribution function of $p(x|y)$. ■

Proposition C.2 (Invariance and minimality) *Let n be a nuisance for the task y and let z be a sufficient representation of the input x. Suppose that z depends on n only through x* (i.e., $n \rightarrow x \rightarrow z$)*. Then,*

$$I(z;n) \leq I(z;x) - I(x;y).$$

Moreover, there exists a nuisance n such that equality holds up to a (generally small) residual ϵ

$$I(z;n) = I(z;x) - I(x;y) - \epsilon,$$

where $\epsilon := I(z;y|n) - I(x;y)$. *In particular* $0 \leq \epsilon \leq H(y|x)$, *and* $\epsilon = 0$ *whenever* y *is a deterministic function of* x. *Under these conditions, a sufficient statistic* z *is invariant (maximally insensitive) to nuisances if and only if it is minimal.*

Proof By hypothesis, we have the Markov chain $(y, n) \to x \to z$; therefore, by the DPI, we have $I(z;y,n) \leq I(z;x)$. The first term can be rewritten using the chain rule as $I(z;y,n) = I(z;n) + I(z;y|n)$, giving us

$$I(z;n) \leq I(z;x) - I(z;y|n).$$

Now, since y and n are independent, $I(z;y|n) \geq I(z;y)$. In fact,

$$\begin{aligned} I(z;y|n) &= H(y|n) - H(y|z,n) \\ &= H(y) - H(y|z,n) \\ &\geq H(y) - H(y|z) = I(y;z). \end{aligned}$$

Substituting in the inequality above, and using the fact that z is sufficient, we finally obtain

$$I(z;n) \leq I(z;x) - I(z;y) = I(z;x) - I(x;y).$$

Moreover, let n be as in Lemma 2.1. Then, since x is a deterministic function of y and n, we have

$$I(z;x) = I(z;n,y) = I(z;n) + I(z;y|n),$$

and therefore

$$I(z;n) = I(z;x) - I(z;y|n) = I(z;x) - I(x;y) - \epsilon.$$

with ϵ defined as above. Using the sufficiency of z, the previous inequality for $I(z;y|n)$, the DPI, we get the chain of inequalities

$$\begin{aligned} \epsilon = I(z;y|n) - I(x;z) &\leq I(x;y|n) - I(x;y) \\ &\leq H(y|n) - H(y|n,z) - H(y) + H(y|x) \\ &\leq H(y) - H(y|n,z) - H(y) + H(y|x) \\ &= H(y|x) - H(y|n,z) \\ &\leq H(y|x) \end{aligned}$$

from which we obtain the desired bounds for ϵ. ∎

While the proof of the following theorem is quite simple, some clarifications on the notation are in order: We assume, following a Bayesian perspective, that the data is generated by some generative model $p(\mathbf{x}, \mathbf{y}|\theta)$, where the parameters θ of the model are sampled from some (unknown) prior $p(\theta)$. Given the parameters θ, the training dataset $\mathcal{D} = (\mathbf{x}, \mathbf{y}) \sim p(x, y|\theta)$ is composed of i.i.d. samples from the unknown distribution $p(x, y|\theta)$. The output of the training algorithm on the dataset $\mathcal{D}$ is a (generally simple, *e.g.*, normal or log-normal) distribution $q(w|\mathbf{x}, \mathbf{y})$ over the weights. Putting everything together, we have a well-defined joint distribution $p(\mathbf{x}, \mathbf{y}, \theta, w) = p(\theta)p(\mathbf{x}, \mathbf{y}|\theta)q(w|\mathbf{x}, \mathbf{y})$.

Given the weights w, the network then defines an inference distribution $q(\mathbf{y}|\mathbf{x}, w)$, which we know and can compute explicitly. Another distribution, which instead we do not know, is $p(\mathbf{y}|\mathbf{x}, w)$, which is obtained from $p(\mathbf{x}, \mathbf{y}, \theta, w)$ and express the optimal inference we could perform on the labels $\mathbf{y}$ using the information contained in the weights. In a well trained network, we want the distribution approximated by the network to match the optimal distribution $q(\mathbf{y}|\mathbf{x}, w) = p(\mathbf{y}|\mathbf{x}, w)$.

Finally, recall that the conditional entropy is defined as

$$H_p(y|z) := \mathbb{E}_{y,z\sim p(y,z)}[-\log p(y|z)],$$

where z can be one random variable or a tuple of random variables. When not specified, it is assumed that the cross-entropy is computed with respect to unknown underlying data distribution $p(\mathbf{x}, \mathbf{y}, w, \theta)$. Similarly, the conditional cross-entropy is defined as

$$\begin{aligned} H_{p,q}(y|z) :=& \mathbb{E}_{y,z\sim p(y,z)}[-\log q(y|z)] \\ =& \mathbb{E}_{y,z\sim p(y,z)}[-\log p(y|z)] + \mathbb{E}_{y,z\sim p(y,z)}[\log \frac{p(y|z)}{q(y|z)}] \\ =& H_p(y|z) + \mathbb{E}_{z\sim p(z)} \mathrm{KL}(\, p(y|z) \,\|\, q(y|z) \,). \end{aligned}$$

Proposition C.3 (Information Decomposition) *Let* $\mathcal{D} = (\mathbf{x}, \mathbf{y})$ *denote the training dataset, then for any training procedure, we have*

$$H_{p,q}(\mathbf{y}|\mathbf{x}, w) = H(\mathbf{y}|\mathbf{x}, \theta) + I(\theta; \mathbf{y}|\mathbf{x}, w) + \mathbb{E}_{\mathbf{x},w} \mathrm{KL}(\, p(\mathbf{y}|\mathbf{x}, w) \,\|\, q(\mathbf{y}|\mathbf{x}, w) \,) - I(\mathbf{y}; w|\mathbf{x}, \theta). \quad (8)$$

Proof Recall that cross-entropy can be written as

$$H_{p,q}(\mathbf{y}|\mathbf{x}, w) = H_p(\mathbf{y}|\mathbf{x}, w) + \mathbb{E}_{\mathbf{x},w} \mathrm{KL}(\, p(\mathbf{y}|\mathbf{x}, w) \,\|\, q(\mathbf{y}|\mathbf{x}, w) \,),$$

so we only have to prove that

$$H_p(\mathbf{y}|\mathbf{x}, w) = H_p(\mathbf{y}|\mathbf{x}, \theta) + I(\mathbf{y}; \theta|\mathbf{x}, w) - I(\mathbf{y}; w|\mathbf{x}, \theta),$$

which is easily done using the following identities:

$$\begin{aligned} I(\mathbf{y}; \theta|\mathbf{x}, w) &= H_p(\theta, \mathbf{y}|w) - H_p(\mathbf{y}|\theta, \mathbf{x}, w), \\ I(\mathbf{y}; w|\mathbf{x}, \theta) &= H_p(\mathbf{y}|\mathbf{x}, \theta) - H_p(\mathbf{y}|\mathbf{x}, \theta, w). \end{aligned}$$

∎

Proposition C.4 (Information in the weights) *Under the previous modeling assumptions, the upper-bound to the information that the weights contain about the dataset is*

$$I(w;\mathcal{D}) \leq \tilde{I}(w;\mathcal{D}) = -\frac{1}{2} \sum_{i=1}^{\dim(w)} \log \alpha_i + C,$$

where the constant C *is arbitrary due to the improper prior.*

Proof Recall that we defined the upperbound $\tilde{I}(\mathbf{w};\mathcal{D})$ as

$$\tilde{I}(\mathbf{w};\mathcal{D}) = \mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \tilde{q}(w)\,),$$

where $\tilde{q}(w)$ is a factorized log-uniform prior. Since the KL divergence is reparametrization invariant, we have:

$$\begin{aligned}\mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, \tilde{q}(w)\,) &= \mathrm{KL}(\, \log \mathcal{N}(\mu,\alpha) \,\|\, \log \mathrm{Uniform}\,) \\ &= \mathrm{KL}(\, \mathcal{N}(\mu,\alpha) \,\|\, \mathrm{Uniform}\,) \\ &= H(\mathcal{N}(\mu,\alpha)) + \mathrm{const} \\ &= -\sum_{i=1}^{\dim(w)} \frac{1}{2}\log(\alpha_i) + \mathrm{const},\end{aligned}$$

where we have used the formula for the entropy of a Gaussian and the fact that the KL divergence of a distribution from the uniform prior is the entropy of the distribution modulo an arbitrary constant. ■

Proposition C.5 (Flat minima have low information) *Let $\hat{w}$ be a local minimum of the cross-entropy loss $H_{p,q}(\mathbf{y}|\mathbf{x},w)$, and let $\mathcal{H}$ be the Hessian at that point. Then, for the optimal choice of the posterior $w|\mathcal{D} = \epsilon \odot \hat{w}$ centered at $\hat{w}$ that optimizes the IB Lagrangian, we have*

$$I(w;\mathcal{D}) \le \tilde{I}(w;\mathcal{D}) \le \frac{1}{2}K[\log \|w\|_2^2 + \log \|H\|_* - K\log(K^2\beta/2)]$$

where $K = \dim(w)$ and $\|\cdot\|_*$ denotes the nuclear norm.

Proof First, we switch to a logarithmic parametrization of the weights, and let $h := \log|w|$ (we can ignore the sign of the weights since it is locally constant). In this parametrization, we can approximate the IB Lagrangian to second order as

$$\mathcal{L} = \mathbb{E}_{h\sim p(h|\mathcal{D})}[H_0 + [(h-h_0)\odot w]^T \mathcal{H}[(h-h_0)\odot w] - \frac{\beta}{2}\sum_i \log\alpha_i$$

where $H_0 = H(\mathbf{y}|\mathbf{x},\hat{w})$. Now, notice that since $q(w|\mathcal{D})$ is a log-normal distribution, we have $q(h|\mathcal{D}) \sim \mathcal{N}(h_0,\alpha)$.[6] Therefore, can compute the expectation exactly as

$$\mathcal{L} = H_0 + \sum_{i=1}^{\dim(w)} \alpha_i w_i^2 \mathcal{H}_{ii} - \frac{\beta}{2}\sum_i \log\alpha_i.$$

Optimizing w.r.t. α_i we get

$$\alpha_i = \frac{\beta}{2w_i^2\mathcal{H}_{ii}},$$

6. Note that for simplicity we have ignored the offset $\alpha/2$ in the mean of the log-normal distribution.

and plugging it back in the expression for $\tilde{I}(w;\mathcal{D})$ that we obtained in the previous proposition, we have

$$\tilde{I}(w;\mathcal{D}) = -\frac{1}{2}\sum_i \log\alpha_i = \frac{1}{2}\sum_i \log(w_i^2) + \log(\mathcal{H}_{ii}) - \log(\beta/2).$$

Finally, by Jensen's inequality, we have

$$\begin{aligned}\tilde{I}(w;\mathcal{D}) &\le \frac{1}{2}K[\log(\sum_i w_i^2) + \log(\sum_i \mathcal{H}_{ii}) - \log(K^2\beta/2)] \\ &= \frac{1}{2}K[\log(\|w\|_2^2) + \log(\|H\|_*) - \log(K^2\beta/2)],\end{aligned}$$

as we wanted. ■

Proposition C.6 *Let $z = Wx$, and assume as before $W = \epsilon\odot\hat{W}$, with $\epsilon_{i,j} \sim \log\mathcal{N}(-\alpha_i/2,\alpha_i)$. Further assume that the marginals of $q(z)$ and $q(z|x)$ are both approximately Gaussian (which is reasonable for large $\dim(x)$ by the Central Limit Theorem). Then,*

$$I(z;x) + \mathrm{TC}(z) = -\frac{1}{2}\sum_{i=1}^{\dim(z)} \mathbb{E}_x \log \frac{\tilde{\alpha}_i \hat{W}_i^2 \cdot x^2}{\hat{W}_i \cdot \mathrm{Cov}(x)\hat{W}_i + \tilde{\alpha}_i \hat{W}_i^2 \cdot \mathbb{E}(x^2)},$$

where W_i denotes the i-th row of the matrix W, and $\tilde{\alpha}_i$ is the noise variance $\tilde{\alpha}_i = \exp(\alpha_i)-1$. In particular, $I(z;x)+\mathrm{TC}(z)$ is a monotone decreasing function of the weight variances α_i.

Proof First, we consider the case in which $\dim(z) = 1$, and so $w := W$ is a single row vector. By hypothesis, $q(z)$ is approximately Gaussian, with mean and variance

$$\begin{aligned}\mu_1 &:= \mathbb{E}[z] = \mathbb{E}[\sum_i \epsilon_i\hat{w}_i x_i] = \sum_i \hat{w}_i \mathbb{E}[x_i] = \hat{w}\cdot\mathbb{E}[x] \\ \sigma_1^2 &:= \mathrm{var}[z] = \mathbb{E}[(\sum_i \epsilon_i\hat{w}_i x_i)^2] - (\mathbb{E}[\sum_i \epsilon_i\hat{w}_i x_i])^2, \\ &= \mathbb{E}[\sum_{i,j} \epsilon_i\epsilon_j\hat{w}_i\hat{w}_j x_i x_j] - \sum_{i,j}\hat{w}_i\hat{w}_j\mathbb{E}[x_i]\mathbb{E}[x_j] \\ &= \tilde{\alpha}\sum_i \hat{w}_i^2\mathbb{E}[x_i]^2 + \sum_{i,j}\hat{w}_i\hat{w}_j\,(\mathbb{E}[x_i x_j] - \mathbb{E}[x_i]\mathbb{E}[x_j]) \\ &= \tilde{\alpha}\hat{w}^2\cdot\mathbb{E}[x^2] + \hat{w}\cdot\mathrm{Cov}(x)\hat{w}.\end{aligned}$$

A similar computation gives us mean and variance of $q(z|x)$:

$$\begin{aligned}\mu_0 &:= \mathbb{E}[z|x] = \hat{w}\cdot x, \\ \sigma_0^2 &:= \mathrm{var}[z|x] = \tilde{\alpha}\hat{w}^2\cdot x^2.\end{aligned}$$

Since we are assuming $\dim(z) = 1$, we trivially have $\mathrm{TC}(z) = 0$, so we are only left with $I(z;x)$ which is given by

$$\begin{aligned} I(z;x) &= \mathbb{E}_x \,\mathrm{KL}(\, q(z|x) \,\|\, q(z)\,) \\ &= \mathbb{E}_x \,\mathrm{KL}(\, \mathcal{N}(\mu_0, \sigma_0^2) \,\|\, \mathcal{N}(\mu_1, \sigma_1^2)\,) \\ &= \frac{1}{2}\mathbb{E}_x \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2 + (\hat{w}\cdot x - \hat{w}\cdot \mathbb{E}[x])^2}{\sigma_1^2} - 1 - \log\frac{\sigma_0^2}{\sigma_1^2} \\ &= -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{\hat{w}\cdot \mathrm{Cov}(x)\hat{w} + \tilde{\alpha}\hat{w}^2 \cdot \mathbb{E}[x^2]}. \end{aligned}$$

Now, for the general case of $\dim(z) \geq 1$, notice that

$$\begin{aligned} I(\mathbf{z};\mathbf{x}) + \mathrm{TC}(\mathbf{z}) &= \mathbb{E}_x \,\mathrm{KL}(\, \prod_k q(z_i|\mathbf{x}) \,\|\, \prod_k q(z_i)\,) \\ &= \sum_{i=1}^{\dim(z)} \mathbb{E}_x \,\mathrm{KL}(\, q(z_i|\mathbf{x}) \,\|\, q(z_i)\,), \end{aligned}$$

where $q(z_i)$ is the marginal of the k-th component of z. We can then use the previous result for each component separately, and sum everything to get the desired identity. ■

Proposition C.7 (Uniform bound for one layer) *Let $z = Wx$, where $W = \epsilon \odot \hat{W}$, where $\epsilon_{i,j} \sim \log\mathcal{N}(-\alpha/2, \alpha)$; assume that the components of x are uncorrelated, and that their kurtosis is uniformly bounded. Then, there is a strictly increasing function $g(\alpha)$ s.t. we have the uniform bound*

$$g(\alpha) \leq \frac{I(x;z) + TC(z)}{\dim(z)} \leq g(\alpha) + c,$$

where $c = O(1/\dim(x)) \leq 1$, $g(\alpha) = \log(1 - e^{-\alpha})/2$ and α is related to $I(w;\mathcal{D})$ by $\alpha = \exp\{-I(W;\mathcal{D})/\dim(W)\}$. In particular, $I(x;z) + TC(z)$ is tightly bounded by $I(W;\mathcal{D})$ and increases strictly with it.

Proof To simplify the notation we do the case $\dim z = 1$, the general case being identical. Let $w := W$ be the only row of W. First notice that, since x is uncorrelated, we have

$$\hat{w}\cdot \mathrm{Cov}(x)\hat{w} = \sum_i w_i^2(\mathbb{E}[x_i^2] - \mathbb{E}[x_i]^2) \leq w^2 \cdot \mathbb{E}[x^2]$$

Therefore,

$$\begin{aligned} I(x;z) =& -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{\hat{w}\cdot \mathrm{Cov}(x)\hat{w} + \tilde{\alpha}\hat{w}^2 \cdot \mathbb{E}[x^2]} \\ \leq& -\frac{1}{2}\mathbb{E}_x \log \frac{\tilde{\alpha}\hat{w}^2 \cdot x^2}{(1+\tilde{\alpha})\hat{w}^2 \cdot \mathbb{E}[x^2]} \\ =& \frac{1}{2}\log(1+\tilde{\alpha}^{-1}) \\ & -\frac{1}{2}\mathbb{E}_x \log\left[1 + \frac{\hat{w}^2\cdot(x^2 - \mathbb{E}[x^2])}{\hat{w}^2\cdot\mathbb{E}[x^2]}\right]. \end{aligned}$$

To conclude, we want to approximate the expectation of the logarithm using a Taylor expansion, but we first need to check that the variance of the term inside the logarithm is low, which is where we need the bound on the kurtosis. In fact, since the kurtosis is bounded, there is some constant C such that for all i

$$\frac{\mathbb{E}(x_i^2 - \mathbb{E}[x_i^2])^2}{\mathbb{E}[x_i^2]^2} \leq C.$$

Now,

$$\begin{aligned} \mathrm{var}\,\frac{\hat{w}^2\cdot(x^2 - \mathbb{E}[x^2])}{\hat{w}^2\cdot\mathbb{E}[x^2]} &= \frac{\sum_i \hat{w}_i^4 \mathbb{E}(x^2 - \mathbb{E}[x^2])^2}{\sum_{i,j}\hat{w}_i^2\hat{w}_j^2\mathbb{E}[x_i^2]\mathbb{E}[x_j^2]} \\ &\leq C\frac{\sum_i \hat{w}_i^4\mathbb{E}[x_i^2]^2}{\sum_{i,j}\hat{w}_i^2\hat{w}_j^2\mathbb{E}[x_i^2]\mathbb{E}[x_j^2]} \\ &= O(1/\dim(x)). \end{aligned}$$

Therefore, we can conclude

$$I(x;z) \leq \frac{1}{2}\log(1+\tilde{\alpha}^{-1}) + O(1/\dim(x)).$$

■

Corollary C.8 (Multi-layer case) *Let W^k for $k = 1, ..., L$ be weight matrices, with $W^k = \epsilon^k \odot \hat{W}^k$ and $\epsilon^k_{i,j} = \log\mathcal{N}(-\alpha^k/2, \alpha^k)$, and let $z_{i+1} = \phi(W^k z_k)$, where $z_0 = x$ and ϕ is any nonlinearity. Then,*

$$I(z_L;x) \leq \min_{k<L}\Big\{\dim(z_k)\big[g(\alpha^k) + 1\big]\Big\}$$

where $\alpha^k = \exp\big\{-I(W^k;\mathcal{D})/\dim(W^k)\big\}$.

Proof Since we have the Markov chain $x \to z_1 \to \ldots \to z_L$, by the Data Processing Inequality we have $I(z_L;x) \leq \min\{I(z_L;z_{L-1}), I(z_{L-1};x)\}$. Iterating this inequality, we have

$$I(z_L;x) \leq \min_{k<L} I(z_{k+1}, z_k).$$

Now, notice that $I(z_{k+1};z_k) \leq I(\phi(W^k z_k);z_k) \leq I(W^k z_k;z_k)$, since applying a deterministic function can only decrease the information. But $I(W^k z_k;z_k)$ is exactly the quantity we bounded in Corollary 5.2, leading us to the desired inequality. ■

Appendix D. Q&A

It is well-known that overfitting relates to the "effective number of degrees of freedom" that can be measured in a number of ways (Friedman et al., 2001,

Chapter 7). Why should we use the information in the weights? The information in the weights is indeed one particular choice of measure of complexity. One nice aspect is that it plays a central role in many different frameworks (minimum description length, variational inference, PAC-Bayes), and correlates well with the performance of a real network.

How do you compute the nuclear norm of the Hessian? It sounds expensive! We do not need to. What we show is that, if the optimization algorithm happens to find flat minima (nuclear norm being a proxy), then it automatically limits the information in the weights - which promotes good generalization. However, if one wanted to approximate the trace of the Hessian, it could be done in linear time (Bai et al., 1996, Prop 4.1).

A nuisance n should convey no information on y *given* x, so why imposing $I(y;n) = 0$ rather than $I(y;n|x) = 0$? While $I(y;n|x) = 0$ may seem intuitively the right condition, it is actually too weak. Suppose for example that y is a deterministic function of x (*i.e.*, the labels are perfectly determined by the data, as often is the case). Then, we would have $I(y;n|x) = 0$ for any n, which would imply that everything is a nuisance for the task, which of course is not intuitively the case.

Why is the dataset a random variable, if we only have one realization of it? The theory is almost identical in both the case of a fixed dataset and a randomly sampled dataset. We consider the case of a randomly sampled dataset since it is simpler and at the same time more general. Some expressions simplify slightly and are easier to interpret, but in the end a fixed training set is given either way.

The use of $KL(q(w|D)||\tilde{q}(w))$ where $q(w|D)$ is a "posterior" defined by a learning algorithm that returns w and $\tilde{q}(w)$ is a log-uniform prior is the basic PAC-Bayes bound, that, however, gives an vacuous generalization bound due to the improper prior. A generalization bound could be stated in terms of the length of a finite interval approximation of log-uniform prior. Indeed, this is the case. However, in the limit of the interval length going to infinity, the KL divergence would still be infinite, and the optimization would be slightly more complex. As simpler option to have a generalization-bound would be to use Gaussian prior and posterior Dziugaite and Roy (2017). However, computing a good PAC-Bayes upper-bound is outside the scope of the paper, and the use of a non-informative, scale invariant prior matches the empirical behavior of networks and simplifies the theoretical analysis.

Of course a minimal representation should be invariant to nuisance variation since that's what it means to be minimal for the task. While the result may be intuitive to some, compression and invariance are not the same thing, and we are unaware of an existing proof of the claim other than for special tasks like clustering, and for small perturbations.

How do you compute the information in the weights, since you only have a sample (one set of weights that the network converged to for a given dataset)? And how do you optimize it? That looks hard! We do not need to compute the information in the weights, since we can control it. Even if we do not do so explicitly, optimizing cross-entropy with SGD yields the right solution (the solution that minimizes

the IBL for the weights). We only have one sample of the weights if we anneal the learning rate to zero, but otherwise SGD produces a posterior distribution of the weights, even if we do not impose additional stochasticity. In many cases we do, for instance using Information Dropout (Achille and Soatto (2018)) or its simpler version, Dropout. A special case of the theory can be re-derived assuming an instantiated dataset and a set of weights, with the same results.

The use of $I(w;\mathcal{D})$ does not seem to upper bound the PAC-Bayes bound. The direction of inequality is the opposite. The IB Lagrangian for the weights is a particular case that gives the sharpest PAC-Bayes bound. Choosing $q(w)$ requires some attention: Once we fix a training procedure, we can (in theory) compute/approximate the marginal distribution $q(w)$ over the stochasticity of the data (if any) as well as the stochasticity of the training procedure. This marginal can then be used in the PAC-Bayes bound: This gives the sharpest bound (McAllester, 2013) and is also equivalent to an IBL since the term $\mathbb{E}_{\mathcal{D}} \mathrm{KL}(\, q(w|\mathcal{D}) \,\|\, q(w)\,)$ measures the mutual information between the training procedure and the weights. Since in general it is not possible to explicitly compute this bound, in practice we use less tight a bound based on a factorized approximation of the marginal. This opportunistic choice later turns out to play a role in disentanglement.

Journal of Machine Learning Research 19 (2018) 1-49 Submitted 11/17; Revised 7/18; Published 8/18

Covariances, Robustness, and Variational Bayes

Ryan Giordano RGIORDANO@BERKELEY.EDU
Department of Statistics, UC Berkeley
367 Evans Hall, UC Berkeley
Berkeley, CA 94720

Tamara Broderick TBRODERICK@CSAIL.MIT.EDU
Department of EECS, MIT
77 Massachusetts Ave., 38-401
Cambridge, MA 02139

Michael I. Jordan JORDAN@CS.BERKELEY.EDU
Department of Statistics and EECS, UC Berkeley
367 Evans Hall, UC Berkeley
Berkeley, CA 94720

Editor: Mohammad Emtiyaz Khan

Abstract

Mean-field Variational Bayes (MFVB) is an approximate Bayesian posterior inference technique that is increasingly popular due to its fast runtimes on large-scale data sets. However, even when MFVB provides accurate posterior means for certain parameters, it often mis-estimates variances and covariances. Furthermore, prior robustness measures have remained undeveloped for MFVB. By deriving a simple formula for the effect of infinitesimal model perturbations on MFVB posterior means, we provide both improved covariance estimates and local robustness measures for MFVB, thus greatly expanding the practical usefulness of MFVB posterior approximations. The estimates for MFVB posterior covariances rely on a result from the classical Bayesian robustness literature that relates derivatives of posterior expectations to posterior covariances and includes the Laplace approximation as a special case. Our key condition is that the MFVB approximation provides good estimates of a select subset of posterior means—an assumption that has been shown to hold in many practical settings. In our experiments, we demonstrate that our methods are simple, general, and fast, providing accurate posterior uncertainty estimates and robustness measures with runtimes that can be an order of magnitude faster than MCMC.

Keywords: Variational Bayes; Bayesian robustness; Mean field approximation; Linear response theory; Laplace approximation; Automatic differentiation

1. Introduction

Most Bayesian posteriors cannot be calculated analytically, so in practice we turn to approximations. Variational Bayes (VB) casts posterior approximation as an optimization problem in which the objective to be minimized is the divergence, among a sub-class of tractable distributions, from the exact posterior. For example, one widely-used and relatively simple flavor of VB is "mean field variational Bayes" (MFVB), which employs Kullback-Leibler (KL) divergence and a factorizing exponential family approximation for the tractable sub-class of posteriors (Wainwright and Jordan, 2008). MFVB has been increasingly popular as an alternative to Markov Chain Monte Carlo (MCMC) in part due to its fast runtimes on large-scale data sets. Although MFVB does not come with any general accuracy guarantees (except asymptotic ones in special cases (Westling and McCormick, 2015; Wang and Blei, 2017)), MFVB produces posterior mean estimates of certain parameters that are accurate enough to be useful in a number of real-world applications (Blei et al., 2016). Despite this ability to produce useful point estimates for large-scale data sets, MFVB is limited as an inferential tool; in particular, MFVB typically underestimates marginal variances (MacKay, 2003; Wang and Titterington, 2004; Turner and Sahani, 2011). Moreover, to the best of our knowledge, techniques for assessing Bayesian robustness have not yet been developed for MFVB. It is these inferential issues that are the focus of the current paper.

Unlike the optimization approach of VB, an MCMC posterior estimate is an empirical distribution formed with posterior draws. MCMC draws lend themselves naturally to the approximate calculation of posterior moments, such as those required for covariances. In contrast, VB approximations lend themselves naturally to sensitivity analysis, since we can analytically differentiate the optima with respect to perturbations. However, as has long been known in the Bayesian robustness literature, the contrast between derivatives and moments is not so stark since, under mild regularity conditions that allow the exchange of integration and differentiation, there is a direct correspondence between derivatives and covariance (Gustafson, 1996b; Basu et al., 1996; Efron, 2015, Section 2.2 below).

Thus, in order to calculate local sensitivity to model hyperparameters, the Bayesian robustness literature re-casts derivatives with respect to hyperparameters as posterior covariances that can be calculated with MCMC. In order to provide covariance estimates for MFVB, we turn this idea on its head and use the sensitivity of MFVB posterior expectations to estimate their covariances. These sensitivity-based covariance estimates are referred to as "linear response" estimates in the statistical mechanics literature (Opper and Saad, 2001), so we refer to them here as *linear response variational Bayes* (LRVB) covariances. Additionally, we derive straightforward MFVB versions of hyperparameter sensitivity measures from the Bayesian robustness literature. Under the assumption that the posterior means of interest are well-estimated by MFVB for all the perturbations of interest, we establish that LRVB provides a good estimate of local sensitivities. In our experiments, we compare LRVB estimates to MCMC, MFVB, and Laplace posterior approximations. We find that the LRVB covariances, unlike the MFVB and Laplace approximations, match the MCMC approximations closely while still being computed over an order of magnitude more quickly than MCMC.

In Section 2 we first discuss the general relationship between Bayesian sensitivity and posterior covariance and then define local robustness and sensitivity. Next, in Section 3, we introduce VB and derive the linear system for the MFVB local sensitivity estimates. In Section 4, we show how to use the MFVB local sensitivity results to estimate covariances and calculate canonical Bayesian hyperparameter sensitivity measures. Finally, in Section 5, we demonstrate the speed and effectiveness of our methods with simple simulated data, an application of automatic differentiation variational inference (ADVI), and a large-scale industry data set.

2. Bayesian Covariances and Sensitivity

2.1 Local Sensitivity and Robustness

Denote an unknown model parameter by the vector $\theta \in \mathbb{R}^K$, assume a dominating measure for θ on $\mathbb{R}^K$ given by λ, and denote observed data by x. Suppose that we have a vector-valued hyperparameter $\alpha \in \mathcal{A} \subseteq \mathbb{R}^D$ that parameterizes some aspects of our model. For example, α might represent prior parameters, in which case we would write the prior density with respect to λ as $p(\theta|\alpha)$, or it might parameterize a class of likelihoods, in which case we could write the likelihood as $p(x|\theta, \alpha)$. Without loss of generality, we will include α in the definition of both the prior and likelihood. For the moment, let $p_\alpha(\theta)$ denote the posterior density of θ given x and α, as given by Bayes' Theorem (this definition of $p_\alpha(\theta)$ will be a special case of the more general Definition 2 below):

$$p_\alpha(\theta) := p(\theta|x, \alpha) = \frac{p(x|\theta, \alpha)\, p(\theta|\alpha)}{\int p(x|\theta', \alpha)\, p(\theta'|\alpha)\, \lambda(d\theta')} = \frac{p(x|\theta, \alpha)\, p(\theta|\alpha)}{p(x|\alpha)}.$$

We will assume that we are interested in a posterior expectation of some function $g(\theta)$ (e.g., a parameter mean, a posterior predictive value, or squared loss): $\mathbb{E}_{p_\alpha}[g(\theta)]$. In the current work, we will quantify the uncertainty of $g(\theta)$ by the posterior variance, $\mathrm{Var}_{p_\alpha}(g(\theta))$. Other measures of central tendency (e.g., posterior medians) or uncertainty (e.g., posterior quantiles) may also be good choices but are beyond the scope of the current work.

Note the dependence of $\mathbb{E}_{p_\alpha}[g(\theta)]$ on both the likelihood and prior, and hence on α, through Bayes' Theorem. The choice of a prior and choice of a likelihood are made by the modeler and are almost invariably a simplified representation of the real world. The choices are therefore to some extent subjective, and so one hopes that the salient aspects of the posterior would not vary under reasonable variation in either choice. Consider the prior, for example. The process of prior elicitation may be prohibitively time-consuming; two practitioners may have irreconcilable subjective prior beliefs, or the model may be so complex and high-dimensional that humans cannot reasonably express their prior beliefs as formal distributions. All of these circumstances might give rise to a range of reasonable prior choices. A posterior quantity is "robust" to the prior to the extent that it does not change much when calculated under these different prior choices.

Quantifying the sensitivity of the posterior to variation in the likelihood and prior is one of the central concerns of the field of robust Bayes (Berger et al., 2000). (We will not discuss the other central concern, which is the selection of priors and likelihoods that lead to robust estimators.) Suppose that we have determined that the hyperparameter α belongs to some open set $\mathcal{A}$, perhaps after expert prior elicitation. Ideally, we would calculate the extrema of $\mathbb{E}_{p_\alpha}[g(\theta)]$ as α ranges over all of $\mathcal{A}$. These extrema are a measure of *global robustness*, and their calculation is intractable or difficult except in special cases (Moreno, 2000; Huber, 2011, Chapter 15). A more practical alternative is to examine how much $\mathbb{E}_{p_\alpha}[g(\theta)]$ changes locally in response to small perturbations in the value of α near some tentative guess, $\alpha_0 \in \mathcal{A}$. To this end we define the *local sensitivity at* α_0 (Gustafson, 2000).

Definition 1 *The local sensitivity of* $\mathbb{E}_{p_\alpha}[g(\theta)]$ *to hyperparameter* α *at* α_0 *is given by*

$$\mathbf{S}_{\alpha_0} := \left.\frac{d\mathbb{E}_{p_\alpha}[g(\theta)]}{d\alpha}\right|_{\alpha_0}. \tag{1}$$

$\mathbf{S}_{\alpha_0}$, the local sensitivity, can be considered a measure of *local robustness* (Gustafson, 2000). Throughout the paper we will distinguish between sensitivity, which comprises objectively defined quantities such as $\mathbf{S}_{\alpha_0}$, and robustness, which we treat as a more subjective concept that may be informed by the sensitivity as well as other considerations. For example, even if one knows $\mathbf{S}_{\alpha_0}$ precisely, how much posterior change is too much change and how much prior variation is reasonable remain decisions to be made by the modeler. For a more in-depth discussion of how we use the terms sensitivity and robustness, see Appendix C.

The quantity $\mathbf{S}_{\alpha_0}$ can be interpreted as measuring sensitivity to hyperparameters within a small region near $\alpha = \alpha_0$ where the posterior dependence on α is approximately linear. Then local sensitivity provides an approximation to global sensitivity in the sense that, to first order,

$$\mathbb{E}_{p_\alpha}[g(\theta)] \approx \mathbb{E}_{p_{\alpha_0}}[g(\theta)] + \mathbf{S}_{\alpha_0}^\intercal(\alpha - \alpha_0).$$

Generally, the dependence of $\mathbb{E}_{p_\alpha}[g(\theta)]$ on α is not given in any closed form that is easy to differentiate. However, as we will now see, the derivative $\mathbf{S}_{\alpha_0}$ is equal, under mild regularity conditions, to a particular posterior covariance that can easily be estimated with MCMC draws.

2.2 Covariances and Sensitivity

We will first state a general result relating sensitivity and covariance and then apply it to our specific cases of interest as they arise throughout the paper, beginning with the calculation of $\mathbf{S}_{\alpha_0}$ from Section 2.1. Consider a general base density $p_0(\theta)$ defined relative to λ and define $\rho(\theta,\alpha)$ to be a λ-measurable log perturbation function that depends on $\alpha \in \mathcal{A} \subseteq \mathbb{R}^D$. We will require the following mild technical assumption:

Assumption 1 *For all* $\alpha \in \mathcal{A}$, $\rho(\theta,\alpha)$ *is continuously differentiable with respect to* α, *and, for a given* λ*-measurable* $g(\theta)$ *there exist* λ*-integrable functions* $f_0(\theta)$ *and* $f_1(\theta)$ *such that* $|p_0(\theta)\exp(\rho(\theta,\alpha))\,g(\theta)| < f_0(\theta)$ *and* $|p_0(\theta)\exp(\rho(\theta,\alpha))| < f_1(\theta)$.

Under Assumption 1 we can normalize the log-perturbed quantity $p_0(\theta)\exp(\rho(\theta,\alpha))$ to get a density in θ with respect to λ.

Definition 2 *Denote by* $p_\alpha(\theta)$ *the normalized posterior given* α*:*

$$p_\alpha(\theta) := \frac{p_0(\theta)\exp(\rho(\theta,\alpha))}{\int p_0(\theta')\exp(\rho(\theta',\alpha))\,\lambda(d\theta')}. \tag{2}$$

For example, $p_\alpha(\theta)$ defined in Section 2.1 is equivalent to taking $p_0(\theta) = p(\theta|x,\alpha_0)$ and $\rho(\theta,\alpha) = \log p(x|\theta,\alpha) + \log p(\theta|\alpha) - \log p(x|\theta,\alpha_0) - \log p(\theta|\alpha_0)$.

For a λ-measurable function $g(\theta)$, consider differentiating the expectation $\mathbb{E}_{p_\alpha}[g(\theta)]$ with respect to α:

$$\frac{d\mathbb{E}_{p_\alpha}[g(\theta)]}{d\alpha^\intercal} := \frac{d}{d\alpha}\int p_\alpha(\theta)\,g(\theta)\,\lambda(d\theta). \tag{3}$$

When evaluated at some $\alpha_0 \in \mathcal{A}$, this derivative measures the local sensitivity of $\mathbb{E}_{p_\alpha}[g(\theta)]$ to the index α at α_0. Define $\mathcal{A}_0 \subseteq \mathcal{A}$ to be an open ball containing α_0. Under Assumption 1 we assume without loss of generality that $\rho(\theta,\alpha_0) \equiv 0$ so that $p_0(\theta) = p_{\alpha_0}(\theta)$; if $\rho(\theta,\alpha_0)$ is non-zero, we can simply incorporate it into the definition of $p_0(\theta)$. Then, under Assumption 1, the derivative in Eq. (3) is equivalent to a particular posterior covariance.

Theorem 1 *Under Assumption 1 ,*

$$\left.\frac{d\mathbb{E}_{p_\alpha}[g(\theta)]}{d\alpha^\intercal}\right|_{\alpha_0} = \mathrm{Cov}_{p_0}\left(g(\theta), \left.\frac{\partial\rho(\theta,\alpha)}{\partial\alpha}\right|_{\alpha_0}\right). \tag{4}$$

Theorem 1 is a straightforward consequence of the Lebesgue dominated convergence theorem; see Appendix A for a detailed proof. Versions of Theorem 1 have appeared many times before; e.g., Diaconis and Freedman (1986); Basu et al. (1996); Gustafson (1996b); Pérez et al. (2006) have contributed variants of this result to the robustness literature.

By using MCMC draws from $p_0(\theta)$ to calculate the covariance on the right-hand side of Eq. (4), one can form an estimate of $d\mathbb{E}_{p_\alpha}[g(\theta)]/d\alpha^\intercal$ at $\alpha = \alpha_0$. One might also approach the problem of calculating $d\mathbb{E}_{p_\alpha}[g(\theta)]/d\alpha^\intercal$ using importance sampling as follows (Owen, 2013, Chapter 9). First, an importance sampling estimate of the dependence of $\mathbb{E}_{p_\alpha}[g(\theta)]$ on α can be constructed with weights that depend on α. Then, differentiating the weights with respect to α provides a sample-based estimate of $d\mathbb{E}_{p_\alpha}[g(\theta)]/d\alpha^\intercal$. We show in Appendix B that this importance sampling approach is equivalent to using MCMC samples to estimate the covariance in Theorem 1.

An immediate corollary of Theorem 1 allows us to calculate $\mathbf{S}_{\alpha_0}$ as a covariance.

Corollary 1 *Suppose that Assumption 1 holds for some* $\alpha_0 \in \mathcal{A}$, *some* $g(\theta)$, *and for*

$$\rho(\theta,\alpha) = \log p(x|\theta,\alpha) + \log p(\theta|\alpha) - \log p(x|\theta,\alpha_0) - \log p(\theta|\alpha_0).$$

Then Theorem 1 implies that

$$\mathbf{S}_{\alpha_0} = \mathrm{Cov}_{p_0}\left(g(\theta), \left.\frac{\partial\rho(\theta,\alpha)}{\partial\alpha}\right|_{\alpha_0}\right). \tag{5}$$

Corollary 1 can be found in Basu et al. (1996), in which a version of Corollary 1 is stated in the proof of their Theorem 1, as well as in Pérez et al. (2006) and Efron (2015). Note that the definition of $\rho(\theta,\alpha)$ does not contain any normalizing constants and so can typically be easily calculated. Given N_s MCMC draws $\{\theta_n\}_{n=1}^{N_s}$ from a chain that we assume to have reached equilibrium at the stationary distribution $p_0(\theta)$, one can calculate an estimate of $\mathbf{S}_{\alpha_0}$ using the sample covariance version of Eq. (4):

$$\hat{\mathbf{S}}_{\alpha_0} := \frac{1}{N_s}\sum_{n=1}^{N_s} g(\theta_n)\left.\frac{\partial\rho(\theta_n,\alpha)}{\partial\alpha}\right|_{\alpha_0} - \left(\frac{1}{N_s}\sum_{n=1}^{N_s} g(\theta_n)\right)\left(\frac{1}{N_s}\sum_{n=1}^{N_s}\left.\frac{\partial\rho(\theta_n,\alpha)}{\partial\alpha}\right|_{\alpha_0}\right) \tag{6}$$

for $\theta_n \sim p_0(\theta)$, where $n = 1, ..., N_s$.

3. Variational Bayesian Covariances and Sensitivity

3.1 Variational Bayes

We briefly review variational Bayes and state our key assumptions about its accuracy. We wish to find an approximate distribution, in some class $\mathcal{Q}$ of tractable distributions, selected to minimize the Kullback-Leibler divergence (KL divergence) between $q \in \mathcal{Q}$ and the exact log-perturbed posterior p_α. We assume that distributions in $\mathcal{Q}$ are parameterized by a finite-dimensional parameter η in some feasible set $\Omega_\eta \subseteq \mathbb{R}^{K_\eta}$.

Definition 3 *The approximating variational family is given by*

$$\mathcal{Q} \quad := \quad \{q : q = q(\theta;\eta) \text{ for } \eta \in \Omega_\eta\}. \tag{7}$$

Given $\mathcal{Q}$, we define the optimal $q \in \mathcal{Q}$, which we call $q_\alpha(\theta)$, as the distribution that minimizes the KL divergence $KL(q(\theta;\eta)\,||p_\alpha(\theta))$ from $p_\alpha(\theta)$. We denote the corresponding optimal variational parameters as η^*.

Definition 4 *The variational approximation $q_\alpha(\theta)$ to $p_\alpha(\theta)$ is defined by*

$$q_\alpha(\theta) := q(\theta;\eta^*) \quad := \quad \textit{argmin}_{q\in\mathcal{Q}}\{KL(q(\theta;\eta)\,||p_\alpha(\theta))\}, \tag{8}$$

where

$$KL(q(\theta;\eta)\,||p_\alpha(\theta)) = \mathbb{E}_{q(\theta;\eta)}[\log q(\theta;\eta) - \log p_\alpha(\theta)].$$

In the KL divergence, the (generally intractable) normalizing constant for $p_\alpha(\theta)$ does not depend on $q(\theta)$ and so can be neglected when optimizing. In order for the KL divergence to be well defined, we assume that both $p_0(\theta)$ and $q(\theta)$ are given with respect to the same base measure, λ, and that the support of $q(\theta)$ is contained in the support of $p_\alpha(\theta)$. We will require some additional mild regularity conditions in Section 3.2 below.

A common choice for the approximating family $\mathcal{Q}$ in Eq. (7) is the "mean field family" (Wainwright and Jordan, 2008; Blei et al., 2016),

$$\mathcal{Q}_{mf} := \left\{ q(\theta) : q(\theta) = \prod_k q(\theta_k;\eta_k) \right\}, \tag{9}$$

where k indexes a partition of the full vector θ and of the parameter vector η. That is, $\mathcal{Q}_{mf}$ approximates the posterior $p_\alpha(\theta)$ as a distribution that factorizes across sub-components of θ. This approximation is commonly referred to as "MFVB," for "mean field variational Bayes." Note that, in general, each function $q(\theta_k;\eta_k)$ in the product is different. For notational convenience we write $q(\theta_k;\eta_k)$ instead of $q_k(\theta_k;\eta_k)$ when the arguments make it clear which function we are referring to, much as the same symbol p is used to refer to many different probability distributions without additional indexing.

One may additionally assume that the components $q(\theta_k;\eta_k)$ are in a convenient exponential family. Although the exponential family assumption does not in general follow from a factorizing assumption, for compactness we will refer to both the factorization and the exponential family assumption as MFVB.

In an MFVB approximation, Ω_η could be a stacked vector of the natural parameters of the exponential families, or the moment parameterization, or perhaps a transformation of these parameters into an unconstrained space (e.g., the entries of the log-Cholesky decomposition of a positive definite information matrix). For more concrete examples, see Section 5. Although all of our experiments and much of our motivating intuition will use MFVB, our results extend to other choices of $\mathcal{Q}$ that satisfy the necessary assumptions.

3.2 Variational Bayes sensitivity

Just as MCMC approximations lend themselves to moment calculations, the variational form of VB approximations lends itself to sensitivity calculations. In this section we derive the sensitivity of VB posterior means to generic perturbations—a VB analogue of Theorem 1. In Section 4 we will choose particular perturbations to calculate VB prior sensitivity and, through Theorem 1, posterior covariances.

In Definition 4, the variational approximation is a function of α through the optimal parameters $\eta^*(\alpha)$, i.e., $q_\alpha(\theta) = q(\theta, \eta^*(\alpha))$. In turn, the posterior expectation $\mathbb{E}_{q_\alpha}[g(\theta)]$ is also a function of α, and its derivative at α_0—the local sensitivity of the variational approximation to α—has a closed form under the following mild technical conditions. As with p_0, define $q_0 := q_{\alpha_0}$, and define $\eta_0^* := \eta^*(\alpha_0)$.

All the following assumptions are intended to hold for a given $p_\alpha(\theta)$, approximating class $\mathcal{Q}$, λ-measurable function $g(\theta)$, and to hold for all $\alpha \in \mathcal{A}_0$ and all η in an open neighborhood of η_0^*.

Assumption 2 *The KL divergence at $KL(q(\theta;\eta)\,||p_0(\theta))$ and expected log perturbation $\mathbb{E}_{q(\theta;\eta)}[\rho(\theta,\alpha)]$ are twice continuously differentiable in η and α.*

Assumption 3 *There exists a strict local minimum, $\eta^*(\alpha)$, of $KL(q(\theta;\eta)\,||p_\alpha(\theta))$ in Eq. (8) such that $\eta^*(\alpha)$ is interior to Ω_η.*

Assumption 4 *The expectation $\mathbb{E}_{q(\theta;\eta)}[g(\theta)]$ is a continuously differentiable function of η.*

We define the following quantities for notational convenience.

Definition 5 *Define the following derivatives of variational expectations evaluated at the optimal parameters:*

$$\mathbf{H}_{\eta\eta} := \left.\frac{\partial^2 KL(q(\theta;\eta)\,||p_0(\theta))}{\partial\eta\partial\eta^\intercal}\right|_{\eta=\eta_0^*} \quad \mathbf{f}_{\alpha\eta} := \left.\frac{\partial^2 \mathbb{E}_{q(\theta;\eta)}[\rho(\theta,\alpha)]}{\partial\alpha\partial\eta^\intercal}\right|_{\eta=\eta_0^*,\alpha=\alpha_0} \quad \mathbf{g}_\eta := \left.\frac{\partial \mathbb{E}_{q(\theta;\eta)}[g(\theta)]}{\partial\eta^\intercal}\right|_{\eta=\eta_0^*}.$$

Since $g(\theta)$, α, and η are all vectors, the quantities $\mathbf{H}_{\eta\eta}$, $\mathbf{f}_{\alpha\eta}$, and $\mathbf{g}_\eta$ are matrices. We are now ready to state a VB analogue of Theorem 1.

Theorem 2 *Consider a variational approximation $q_\alpha(\theta)$ to $p_\alpha(\theta)$ as given in Definition 4 and a λ-measurable function $g(\theta)$. Then, under Assumptions 1–4 , using the definitions given in Definition 5, we have*

$$\left.\frac{d\mathbb{E}_{q_\alpha}[g(\theta)]}{d\alpha^\intercal}\right|_{\alpha_0} \quad = \quad \mathbf{g}_\eta \mathbf{H}_{\eta\eta}^{-1} \mathbf{f}_{\alpha\eta}^\intercal. \tag{10}$$

A proof of Theorem 2 is given in Appendix D. As with Theorem 1, by choosing the appropriate $\rho(\theta,\alpha)$ and evaluating $\mathbf{f}_{\alpha\eta}$, we can use Theorem 2 to calculate the exact sensitivity of VB solutions to any arbitrary local perturbations that satisfy the regularity conditions. Assumptions 1–4 are typically not hard to verify. For an example, see Appendix E, where we establish Assumptions 1–4 for a multivariate normal target distribution and a mean-field approximation.

Eq. (10) is formally similar to frequentist sensitivity estimates. For example, the pioneering paper of Cook (1986) contains a formula for assessing the curvature of a marginal likelihood surface (Cook, 1986, Equation 15) that, like our Theorem 2, represents the sensitivity as a linear system involving the Hessian of an objective function at its optimum. The geometric interpretation of local robustness suggested by Cook (1986) has been extended to Bayesian settings (see, for example, Zhu et al. (2007, 2011)). In addition to generality, one attractive aspect of their geometric approach is its invariance to parameterization. Investigating geometric interpretations of the present work may be an interesting avenue for future research.

3.3 Approximating with Variational Bayes

Recall that we are ultimately interested in $\mathbb{E}_{p_\alpha}[g(\theta)]$. Variational approximations and their sensitivity measures will be useful to the extent that both the variational means and sensitivities are close to the exact means and sensitivities. We formalize these desiderata as follows.

Condition 1 *Under Assumptions 1–4 and the quantities defined therein, we additionally have, for all* $\alpha \in \mathcal{A}$,

$$\mathbb{E}_{q_\alpha}\left[g\left(\theta\right)\right] \approx \mathbb{E}_{p_\alpha}\left[g\left(\theta\right)\right] \quad \textit{and} \tag{11}$$

$$\left.\frac{d\mathbb{E}_{q_\alpha}\left[g\left(\theta\right)\right]}{d\alpha^\intercal}\right|_{\alpha_0} \approx \left.\frac{d\mathbb{E}_{p_\alpha}\left[g\left(\theta\right)\right]}{d\alpha^\intercal}\right|_{\alpha_0} \tag{12}$$

We will not attempt to be precise about what we mean by the "approximately equal" sign, since we are not aware of any practical tools for evaluating quantitatively whether Condition 1 holds other than running both VB and MCMC (or some other slow but accurate posterior approximation) and comparing the results. However, VB has been useful in practice to the extent that Condition 1 holds true for at least some parameters of interest. We provide some intuition for when Condition 1 might hold in Section 5.1, and will evaluate Condition 1 in each of our experiments below by comparing the VB and MCMC posterior approximate means and sensitivities.

Since Condition 1 holds only for a particular choice of $g\left(\theta\right)$, it is weaker than the assumption that q_α is close to p_α in KL divergence, or even that all the posterior means are accurately estimated. For example, as discussed in Appendix B of Giordano et al. (2015) and in Section 10.1.2 of Bishop (2006), a mean-field approximation to a multivariate normal posterior produces inaccurate covariances and may have an arbitrarily bad KL divergence from p_α, but Condition 1 holds exactly for the location parameters. We discuss the multivariate normal example further in Section 4.1 and Section 5.1 below.

4. Calculation and Uses of Sensitivity

In this section, we discuss two applications of Theorem 1 and Theorem 2: calculating improved covariance estimates and prior sensitivity measures for MFVB. Throughout this section, we will assume that we can apply Theorem 1 and Theorem 2 unless stated otherwise.

4.1 Covariances for Variational Bayes

Consider the mean field approximating family, $\mathcal{Q}_{mf}$, from Section 3.1 and a fixed exact posterior $p_0\left(\theta\right)$. It is well known that the resulting marginal variances also tend to be under-estimated even when parameters means are well-estimated (see, e.g., (MacKay, 2003; Wang and Titterington, 2004; Turner and Sahani, 2011; Bishop, 2006, Chapter 10)). Even more obviously, any $q \in \mathcal{Q}_{mf}$ yields zero as its estimate of the covariance between sub-components of θ that are in different factors of the mean field approximating family. It is therefore unreasonable to expect that $\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right)$. However, if Condition 1 holds, we may expect the sensitivity of MFVB means to certain perturbations to be accurate by Condition 1, and, by Theorem 1, we expect the corresponding covariances to be accurately estimated by the MFVB sensitivity. In particular, by taking $\rho\left(\theta, \alpha\right) = \alpha^\intercal g\left(\theta\right)$ and $\alpha_0 = 0$, we have by Condition 1 that

$$\left.\frac{d\mathbb{E}_{q_\alpha}\left[g\left(\theta\right)\right]}{d\alpha^\intercal}\right|_{\alpha=0} \approx \left.\frac{d\mathbb{E}_{p_\alpha}\left[g\left(\theta\right)\right]}{d\alpha^\intercal}\right|_{\alpha=0} = \mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right). \tag{13}$$

We can consequently use Theorem 2 to provide an estimate of $\mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right)$ that may be superior to $\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$. With this motivation in mind, we make the following definition.

Definition 6 *The linear response variational Bayes (LRVB) approximation,* $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$*, is given by*

$$\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right) := \mathbf{g}_\eta \mathbf{H}_{\eta\eta}^{-1} \mathbf{g}_\eta^\intercal. \tag{14}$$

Corollary 2 *For a given* $p_0\left(\theta\right)$*, class* $\mathcal{Q}$*, and function* $g\left(\theta\right)$*, when Assumptions 1–4 and Condition 1 hold for* $\rho\left(\theta, \alpha\right) = \alpha^\intercal g\left(\theta\right)$ *and* $\alpha_0 = 0$*, then*

$$\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right).$$

The strict optimality of η_0^* in Assumption 3 guarantees that $\mathbf{H}_{\eta\eta}$ will be positive definite and symmetric, and, as desired, the covariance estimate $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$ will be positive semidefinite and symmetric. Since the optimal value of every component of $\mathbb{E}_{q_\alpha}\left[g\left(\theta\right)\right]$ may be affected by the log perturbation $\alpha^\intercal g\left(\theta\right)$, $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$ can estimate non-zero covariances between elements of $g\left(\theta\right)$ even when they have been partitioned into separate factors of the mean field approximation.

Note that $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$ and $\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$ differ only when there are at least some moments of p_0 that q_0 fails to accurately estimate. In particular, if q_α provided a good approximation to p_α for both the first and second moments of $g\left(\theta\right)$, then we would have $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$ since, for q_0 and p_0,

$$\begin{gathered}\mathbb{E}_{q_0}\left[g\left(\theta\right)\right] \approx \mathbb{E}_{p_0}\left[g\left(\theta\right)\right] \text{ and} \\ \mathbb{E}_{q_0}\left[g\left(\theta\right) g\left(\theta\right)^\intercal\right] \approx \mathbb{E}_{p_0}\left[g\left(\theta\right) g\left(\theta\right)^\intercal\right] \Rightarrow \\ \mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right),\end{gathered}$$

and, for q_α and p_α,

$$\begin{gathered}\mathbb{E}_{q_\alpha}\left[g\left(\theta\right)\right] \approx \mathbb{E}_{p_\alpha}\left[g\left(\theta\right)\right] \Rightarrow \\ \mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right).\end{gathered}$$

Putting these two approximate equalities together, we see that, when the first and second moments of q_α approximately match those of p_α,

$$\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right) \approx \mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right).$$

However, in general, $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right) \neq \mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$. In this sense, any discrepancy between $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$ and $\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$ indicates an inadequacy of the variational approximation for at least the second moments of $g\left(\theta\right)$.

Let us consider a simple concrete illustrative example which will demonstrate both how $\mathrm{Cov}_{q_0}\left(g\left(\theta\right)\right)$ can be a poor approximation to $\mathrm{Cov}_{p_0}\left(g\left(\theta\right)\right)$ and how $\mathrm{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)$ can improve the approximation for some moments but not others. Suppose that the exact posterior is a bivariate normal,

$$p_0\left(\theta\right) = \mathcal{N}\left(\theta | \mu, \mathbf{\Sigma}\right), \tag{15}$$

where $\theta = \left(\theta_1, \theta_2\right)^\intercal$, $\mu = \left(\mu_1, \mu_2\right)^\intercal$, $\mathbf{\Sigma}$ is invertible, and $\mathbf{\Lambda} := \mathbf{\Sigma}^{-1}$. One may think of μ and $\mathbf{\Sigma}$ as known functions of x via Bayes' theorem, for example, as given by a normal-normal conjugate model. Suppose we use the MFVB approximating family

$$\mathcal{Q}_{mf} = \left\{q\left(\theta\right) : q\left(\theta\right) = q\left(\theta_1\right) q\left(\theta_2\right)\right\}.$$

One can show (see Appendix E) that the optimal MFVB approximation to p_α in the family $\mathcal{Q}_{mf}$ is given by

$$\begin{aligned}q_0\left(\theta_1\right) &= \mathcal{N}\left(\theta_1 | \mu_1, \mathbf{\Lambda}_{11}^{-1}\right) \\ q_0\left(\theta_2\right) &= \mathcal{N}\left(\theta_2 | \mu_2, \mathbf{\Lambda}_{22}^{-1}\right).\end{aligned}$$

Note that the posterior mean of θ_1 is exactly estimated by the MFVB procedure:

$$\mathbb{E}_{q_0}\left[\theta_1\right] = \mu_1 = \mathbb{E}_{p_0}\left[\theta_1\right].$$

However, if $\mathbf{\Sigma}_{12} \neq 0$, then $\mathbf{\Lambda}_{11}^{-1} < \mathbf{\Sigma}_{11}$, and the variance of θ_1 is underestimated. It follows that the expectation of θ_1^2 is *not* correctly estimated by the MFVB procedure:

$$\mathbb{E}_{q_\alpha}\left[\theta_1^2\right] = \mu_1^2 + \mathbf{\Lambda}_{11}^{-1} < \mu_1^2 + \mathbf{\Sigma}_{11} = \mathbb{E}_{p_\alpha}\left[\theta_1^2\right].$$

An analogous statement holds for θ_2. Of course, the covariance is also mis-estimated if $\mathbf{\Sigma}_{12} \neq 0$ since, by construction of the MFVB approximation,

$$\mathrm{Cov}_{q_0}(\theta_1, \theta_2) = 0 \neq \mathbf{\Sigma}_{12} = \mathrm{Cov}_{p_0}(\theta_1, \theta_2).$$

Now let us take the log perturbation $\rho(\theta, \alpha) = \theta_1\alpha_1 + \theta_2\alpha_2$. For all α in a neighborhood of zero, the log-perturbed posterior given by Eq. (2) remains multivariate normal, so it remains the case that, as a function of α, $\mathbb{E}_{q_\alpha}[\theta_1] = \mathbb{E}_{p_\alpha}[\theta_1]$ and $\mathbb{E}_{q_\alpha}[\theta_2] = \mathbb{E}_{p_\alpha}[\theta_2]$. Again, see Appendix E for a detailed proof. Consequently, Condition 1 holds with equality (not approximate equality) when $g(\theta) = \theta$. However, since the second moments are not accurate (irrespective of α), Condition 1 does not hold exactly when $g(\theta) = (\theta_1^2, \theta_2^2)^\intercal$, nor when $g(\theta) = \theta_1\theta_2$. (Condition 1 may still hold approximately for second moments when $\mathbf{\Sigma}_{12}$ is small.) The fact that Condition 1 holds with equality for $g(\theta) = \theta$ allows us to use Theorem 1 and Theorem 2 to calculate $\mathrm{Cov}_{q_0}^{LR}(g(\theta)) = \mathrm{Cov}_{p_0}(g(\theta))$, even though $\mathbb{E}_{p_0}[\theta_1\theta_2]$ and $\mathbb{E}_{p_0}\left[(\theta_1^2, \theta_2^2)^\intercal\right]$ are mis-estimated.

In fact, when Condition 1 holds with equality for some θ_i, then the estimated covariance in Eq. (14) for all terms involving θ_i will be exact as well. Condition 1 holds with equality for the means of θ_i in the bivariate normal model above, and in fact holds for the general multivariate normal case, as described in Appendix E. Below, in Section 5, in addition to robustness measures, we will also report the accuracy of Eq. (14) for estimating posterior covariances. We find that, for most parameters of interest, particularly location parameters, $\mathrm{Cov}_{q_0}^{LR}(g(\theta))$ provides a good approximation to $\mathrm{Cov}_{p_0}(g(\theta))$.

4.2 Linear Response Covariances in Previous Literature

The application of sensitivity measures to VB problems for the purpose of improving covariance estimates has a long history under the name "linear response methods." These methods originated in the statistical physics literature (Tanaka, 2000; Opper and Saad, 2001) and have been applied to various statistical and machine learning problems (Kappen and Rodriguez, 1998; Tanaka, 1998; Welling and Teh, 2004; Opper and Winther, 2004). The current paper, which builds on this line of work and on our earlier work (Giordano et al., 2015), represents a simplification and generalization of classical linear response methods and serves to elucidate the relationship between these methods and the local robustness literature. In particular, while Giordano et al. (2015) focused on moment-parameterized exponential families, we derive linear-response covariances for generic variational approximations and connect the linear-response methodology to the Bayesian robustness literature.

A very reasonable approach to address the inadequacy of MFVB covariances is simply to increase the expressiveness of the model class $\mathcal{Q}$—although, as noted by Turner and Sahani (2011), increased expressiveness does not necessarily lead to better posterior moment estimates. This approach is taken by much of the recent VB literature (e.g., Tran et al., 2015a,b; Ranganath et al., 2016; Rezende and Mohamed, 2015; Liu and Wang, 2016). Though this research direction remains lively and promising, the use of a more complex class $\mathcal{Q}$ sometimes sacrifices the speed and simplicity that made VB attractive in the first place, and often without the relatively well-understood convergence guarantees of MCMC. We also stress that the current work is not necessarily at odds with the approach of increasing expressiveness. Sensitivity methods can be a supplement to any VB approximation for which our estimators, which require solving a linear system involving the Hessian of the KL divergence, are tractable.

4.3 The Laplace Approximation and Linear Response Covariances

In this section, we briefly compare linear response covariances to the Laplace approximation (Gelman et al., 2014, Chapter 13). The Laplace approximation to $p_0(\theta)$ is formed by first finding the "maximum *a posteriori*" (MAP) estimate,

$$\hat{\theta}_{Lap} := \operatorname*{argmax}_\theta p_0(\theta), \tag{16}$$

and then forming the multivariate normal posterior approximation

$$\mathbf{H}_{Lap} := -\left.\frac{\partial^2 p_0(\theta)}{\partial\theta\partial\theta^\intercal}\right|_{\hat{\theta}_{Lap}} \tag{17}$$

$$\mathrm{Cov}_{q_{Lap}}^{Lap}(\theta) := \mathbf{H}_{Lap}^{-1}$$

$$q_{Lap}(\theta) := \mathcal{N}\left(\theta|\hat{\theta}_{Lap}, \mathrm{Cov}_{q_{Lap}}^{Lap}(\theta)\right). \tag{18}$$

Since both LRVB and the Laplace approximation require the solution of an optimization problem (Eq. (8) and Eq. (16) respectively) and the estimation of covariances via an inverse Hessian of the optimization objective (Eq. (14) and Eq. (17) respectively), it will be instructive to compare the two approaches.

Following Neal and Hinton (1998), we can, in fact, view the MAP estimator as a special variational approximation, where we define

$$\mathcal{Q}_{Lap} := \Big\{ q(\theta;\theta_0) : \int q(\theta;\theta_0)\log p_0(\theta)\,\lambda(d\theta) = \log p_0(\theta_0) \text{ and } \int q(\theta;\theta_0)\log q(\theta;\theta_0)\,\lambda(d\theta) = Constant \Big\},$$

where the $Constant$ term is constant in θ_0. That is, $\mathcal{Q}_{Lap}$ consists of "point masses" at θ_0 with constant entropy. Generally such point masses may not be defined as densities with respect to λ, and the KL divergence in Eq. (8) may not be formally defined for $q \in \mathcal{Q}_{Lap}$. However, if $\mathcal{Q}_{Lap}$ can be approximated arbitrarily well by well-defined densities (e.g., normal distributions with variance fixed at an arbitrarily small number), then we can use $\mathcal{Q}_{Lap}$ as a heuristic tool for understanding the MAP estimator.

Since $\mathcal{Q}_{Lap}$ contains only point masses, the covariance of the variational approximation is the zero matrix: $\mathrm{Cov}_{q_{Lap}}(\theta) = 0$. Thus, as when one uses the mean field assumption, $\mathrm{Cov}_{q_{Lap}}(\theta)$ underestimates the marginal variances and magnitudes of the covariances of $\mathrm{Cov}_{p_0}(\theta)$. Of course, the standard Laplace approximation uses $\mathrm{Cov}_{q_{Lap}}^{Lap}(\theta)$, not $\mathrm{Cov}_{q_{Lap}}(\theta)$, to approximate $\mathrm{Cov}_{p_0}(\theta)$. In fact, $\mathrm{Cov}_{q_{Lap}}^{Lap}(\theta)$ is equivalent to a linear response covariance matrix calculated for the approximating family $\mathcal{Q}_{Lap}$:

$$KL(q(\theta;\theta_0)\,||p_0(\theta)) = -\log p_0(\theta_0) - Constant \Rightarrow$$

$$\hat{\theta}_{Lap} = \operatorname*{argmax}_\theta p_0(\theta) = \operatorname*{argmin}_{\theta_0} KL(q(\theta;\theta_0)\,||p_0(\theta)) = \theta_0^*$$

$$\mathbf{H}_{Lap} = -\left.\frac{\partial^2 p_0(\theta)}{\partial\theta\partial\theta^\intercal}\right|_{\hat{\theta}_{Lap}} = -\left.\frac{\partial^2 KL(q(\theta;\theta_0)\,||p_0(\theta))}{\partial\theta_0\partial\theta_0^\intercal}\right|_{\theta_0^*} = \mathbf{H}_{\eta\eta}.$$

So $\hat{\theta}_{Lap} = \theta_0^*$, $\mathbf{H}_{Lap} = \mathbf{H}_{\eta\eta}$, and $\mathrm{Cov}_{q_{Lap}}^{Lap}(\theta) = \mathrm{Cov}_{q_0}^{LR}(\theta)$ for the approximating family $\mathcal{Q}_{Lap}$.

From this perspective, the accuracy of the Laplace approximation depends precisely on the extent to which Condition 1 holds for the family of point masses $\mathcal{Q}_{Lap}$. Typically, VB approximations use a $\mathcal{Q}$ that is more expressive than $\mathcal{Q}_{Lap}$, and we might expect Condition 1 to be more likely to apply for a more expressive family. It follows that we might expect the LRVB covariance estimate $\mathrm{Cov}_{q_0}^{LR}$ for general $\mathcal{Q}$ to be more accurate than the Laplace covariance approximation $\mathrm{Cov}_{q_{Lap}}^{Lap}$. We demonstrate the validity of this intuition in the experiments of Section 5.

4.4 Local Prior Sensitivity for MFVB

We now turn to estimating prior sensitivities for MFVB estimates—the variational analogues of $\mathbf{S}_{\alpha_0}$ in Definition 1. First, we define the variational local sensitivity.

Definition 7 *The local sensitivity of* $\mathbb{E}_{q_\alpha}[g(\theta)]$ *to prior parameter* α *at* α_0 *is given by*

$$\mathbf{S}_{\alpha_0}^q := \left.\frac{d\mathbb{E}_{q_\alpha}[g(\theta)]}{d\alpha}\right|_{\alpha_0}.$$

Corollary 3 *Suppose that Assumptions 1–4 and Condition 1 hold for some $\alpha_0 \in \mathcal{A}$ and for*

$$\rho(\theta, \alpha) = \log p(x|\theta, \alpha) + \log p(\theta|\alpha) - \log p(x|\theta, \alpha_0) - \log p(\theta|\alpha).$$

Then $\mathbf{S}^q_{\alpha_0} \approx \mathbf{S}_{\alpha_0}$.

Corollary 3 states that, as with the covariance approximations in Section 4.1, $\mathbf{S}^q_{\alpha_0}$ is a useful approximation to $\mathbf{S}_{\alpha_0}$ to the extent that Condition 1 holds—that is, to the extent that the MFVB means are good approximations to the exact means for the prior perturbations $\alpha \in \mathcal{A}_0$.

Under the $\rho(\theta, \alpha)$ given in Corollary 3, Theorem 2 gives the following formula for the variational local sensitivity:

$$\mathbf{S}^q_{\alpha_0} = \mathbf{g}_\eta \mathbf{H}_{\eta\eta}^{-1} \frac{\partial}{\partial \eta^\intercal} \mathbb{E}_{q(\theta;\eta)} \left[\frac{\partial \rho(\theta, \alpha)}{\partial \alpha}\bigg|_{\alpha_0} \right] \bigg|_{\eta_0^*}. \tag{19}$$

We now use Eq. (19) to reproduce MFVB versions of some standard robustness measures found in the existing literature. A simple case is when the prior $p(\theta|\alpha)$ is believed to be in a given parametric family, and we are simply interested in the effect of varying the parametric family's parameters (Basu et al., 1996; Giordano et al., 2016). For illustration, we first consider a simple example where $p(\theta|\alpha)$ is in the exponential family, with natural sufficient statistic θ and log normalizer $A(\alpha)$, and we take $g(\theta) = \theta$. In this case,

$$\begin{aligned}
\log p(\theta|\alpha) &= \alpha^\intercal \theta - A(\alpha) \\
\mathbf{f}_{\alpha\eta} &= \frac{\partial}{\partial \eta^\intercal} \mathbb{E}_{q(\theta;\eta)} \left[\frac{\partial}{\partial \alpha} \left(\alpha^\intercal \theta - A(\alpha) \right)\bigg|_{\alpha_0} \right] \bigg|_{\eta_0^*} \\
&= \left(\frac{\partial}{\partial \eta^\intercal} \mathbb{E}_{q(\theta;\eta)}[\theta] - \frac{\partial}{\partial \eta^\intercal} \frac{\partial A(\alpha)}{\partial \alpha}\bigg|_{\alpha_0} \right) \bigg|_{\eta_0^*} \\
&= \frac{\partial}{\partial \eta^\intercal} \mathbb{E}_{q(\theta;\eta)}[\theta] \bigg|_{\eta_0^*} \\
&= \mathbf{g}_\eta.
\end{aligned}$$

Note that when $\mathbf{f}_{\alpha\eta} = \mathbf{g}_\eta$, Eq. (19) is equivalent to Eq. (14). So we see that

$$\mathbf{S}^q_{\alpha_0} = \mathrm{Cov}^{LR}_{q_0}(\theta).$$

In this case, the sensitivity is simply the linear response covariance estimate of the covariance, $\mathrm{Cov}^{LR}_{q_0}(\theta)$. By the same reasoning, the exact posterior sensitivity is given by

$$\mathbf{S}_{\alpha_0} = \mathrm{Cov}_{p_0}(\theta).$$

Thus, $\mathbf{S}^q_{\alpha_0} \approx \mathbf{S}_{\alpha_0}$ to the extent that $\mathrm{Cov}^{LR}_{q_0}(\theta) \approx \mathrm{Cov}_{p_0}(\theta)$, which again holds to the extent that Condition 1 holds. Note that if we had used a mean field assumption and had tried to use the direct, uncorrected response covariance $\mathrm{Cov}_{q_0}(\theta)$ to try to evaluate $\mathbf{S}^q_{\alpha_0}$, we would have erroneously concluded that the prior on one component, θ_{k_1}, would not affect the posterior mean of some other component, θ_{k_2}, for $k_2 \neq k_1$.

Sometimes it is easy to evaluate the derivative of the log prior even when it is not easy to normalize it. As an example, we will show how to calculate the local sensitivity to the concentration parameter of an LKJ prior (Lewandowski et al., 2009) under an inverse Wishart variational approximation. The LKJ prior is defined as follows. Let $\mathbf{\Sigma}$ (as part of θ) be an unknown $K \times K$ covariance matrix. Define the $K \times K$ scale matrix $\mathbf{M}$ such that

$$\mathbf{M}_{ij} = \begin{cases} \sqrt{\mathbf{\Sigma}_{ij}} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Using $\mathbf{M}$, define the correlation matrix $\mathbf{R}$ as

$$\mathbf{R} = \mathbf{M}^{-1} \mathbf{\Sigma} \mathbf{M}^{-1}.$$

The LKJ prior on the covariance matrix $\mathbf{R}$ with concentration parameter $\alpha > 0$ is given by:

$$p_{\mathrm{LKJ}}(\mathbf{R}|\alpha) \propto |\mathbf{R}|^{\alpha - 1}.$$

The Stan manual recommends the use of p_{LKJ}, together with an independent prior on the diagonal entries of the scaling matrix $\mathbf{M}$, for the prior on a covariance matrix that appears in a hierarchical model (Stan Team, 2015, Chapter 9.13).

Suppose that we have chosen the variational approximation

$$q(\mathbf{\Sigma}) := \mathrm{InverseWishart}(\mathbf{\Sigma}|\mathbf{\Psi}, \nu),$$

where $\mathbf{\Psi}$ is a positive definite scale matrix and ν is the number of degrees of freedom. In this case, the variational parameters are $\eta = (\mathbf{\Psi}, \nu)$. We write η with the understanding that we have stacked only the upper-diagonal elements of $\mathbf{\Psi}$ since $\mathbf{\Psi}$ is constrained to be symmetric and η^* must be interior. As we show in Appendix G,

$$\mathbb{E}_q[\log p_{\mathrm{LKJ}}(\mathbf{R}|\alpha)] = (\alpha - 1) \left(\log|\mathbf{\Psi}| - \psi_K\left(\frac{\nu}{2}\right) - \sum_{k=1}^K \log\left(\frac{1}{2}\mathbf{\Psi}_{kk}\right) + K\psi\left(\frac{\nu - K + 1}{2}\right) \right) + Constant,$$

where $Constant$ contains terms that do not depend on α, and where ψ_K denotes the multivariate digamma function. Consequently, we can evaluate

$$\begin{aligned}
\mathbf{f}_{\alpha\eta} &= \frac{\partial}{\partial \eta^\intercal} \mathbb{E}_{q(\theta;\eta)} \left[\frac{\partial}{\partial \alpha} \log p(\mathbf{\Sigma}|\alpha) \right] \bigg|_{\eta=\eta_0^*, \alpha=\alpha_0} \\
&= \frac{\partial}{\partial \eta^\intercal} \left(\log|\mathbf{\Psi}| - \psi_K\left(\frac{n}{2}\right) - \sum_{k=1}^K \log\left(\frac{1}{2}\mathbf{\Psi}_{kk}\right) + K\psi\left(\frac{n - K + 1}{2}\right) \right) \bigg|_{\eta_0^*}.
\end{aligned} \tag{20}$$

This derivative has a closed form, but the bookkeeping required to represent an unconstrained parameterization of the matrix $\mathbf{\Psi}$ within η would be tedious. In practice, we evaluate terms like $\mathbf{f}_{\alpha\eta}$ using automatic differentiation tools (Baydin et al., 2018).

Finally, in cases where we cannot evaluate $\mathbb{E}_{q(\theta;\eta)}[\log p(\theta|\alpha)]$ in closed form as a function of η, we can use numerical techniques as described in Section 4.5. We thus view $\mathbf{S}^q_{\alpha_0}$ as the exact sensitivity to an approximate KL divergence.

4.5 Practical Considerations when Computing the Sensitivity of Variational Approximations

We briefly discuss practical issues in the computation of Eq. (10), which requires calculating the product $\mathbf{g}_\eta \mathbf{H}_{\eta\eta}^{-1}$ (or, equivalently, $\mathbf{H}_{\eta\eta}^{-1} \mathbf{g}_\eta^\intercal$ since $\mathbf{H}_{\eta\eta}$ is symmetric). Calculating $\mathbf{H}_{\eta\eta}$ and solving this linear system can be the most computationally intensive part of computing Eq. (10).

We first note that it can be difficult and time consuming in practice to manually derive and implement second-order derivatives. Even a small programming error can lead to large errors in Theorem 2. To ensure accuracy and save analyst time, we evaluated all the requisite derivatives using the Python `autograd` automatic differentiation library (Maclaurin et al., 2015) and the Stan math automatic differentiation library (Carpenter et al., 2015).

Note that the dimension of $\mathbf{H}_{\eta\eta}$ is as large as that of η, the parameters that specify the variational distribution $q(\theta; \eta)$. Many applications of MFVB employ many latent variables, the number of which may even scale with the amount of data—including several of the cases that we examine in Section 5. However, these applications typically have special structure that render $\mathbf{H}_{\eta\eta}$ sparse, allowing the practitioner to calculate $\mathbf{g}_\eta \mathbf{H}_{\eta\eta}^{-1}$

quickly. Consider, for example, a model with "global" parameters, θ_{glob}, that are shared by all the individual datapoint likelihoods, and "local" parameters, $\theta_{loc,n}$, associated with likelihood of a single datapoint indexed by n. By "global" and "local" we mean the likelihood and assumed variational distribution factorize as

$$p\left(x, \theta_{glob}, \theta_{loc,1}, ..., \theta_{loc,N}\right) = p\left(\theta_{glob}\right) \prod_{n=1}^{N} p\left(x|\theta_{loc,n}, \theta_{glob}\right) p\left(\theta_{loc,n}|\theta_{glob}\right) \tag{21}$$

$$q\left(\theta;\eta\right) = q\left(\theta_{glob};\eta_{glob}\right) \prod_{n=1}^{N} q\left(\theta_{loc,n};\eta_{n}\right) \text{ for all } q\left(\theta;\eta\right) \in \mathcal{Q}.$$

In this case, the second derivatives of the variational objective between the parameters for local variables vanish:

$$\text{for all } n \neq m, \frac{\partial^2 KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)}{\partial\eta_{loc,n}\partial\eta_{loc,m}^{\intercal}} = 0.$$

The model in Section 5.3 has such a global / local structure; see Section 5.3.2 for more details. Additional discussion, including the use of Schur complements to take advantage of sparsity in the log likelihood, can be found in Giordano et al. (2015).

When even calculating or instantiating $\mathbf{H}_{\eta\eta}$ is prohibitively time-consuming, one can use conjugate gradient algorithms to approximately compute $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_{\eta}^{\intercal}$ (Wright and Nocedal, 1999, Chapter 5). The advantage of conjugate gradient algorithms is that they approximate $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_{\eta}^{\intercal}$ using only the Hessian-vector product $\mathbf{H}_{\eta\eta}\mathbf{g}_{\eta}^{\intercal}$, which can be computed efficiently using automatic differentiation without ever forming the full Hessian $\mathbf{H}_{\eta\eta}$. See, for example, the `hessian_vector_product` method of the Python `autograd` package (Maclaurin et al., 2015). Note that a separate conjugate gradient problem must be solved for each column of $\mathbf{g}_{\eta}^{\intercal}$, so if the parameter of interest $g\left(\theta\right)$ is high-dimensional it may be faster to pay the price for computing and inverting the entire matrix $\mathbf{H}_{\eta\eta}$. See 5.3.2 for more discussion of a specific example.

In Theorem 2, we require η_0^* to be at a true local optimum. Otherwise the estimated sensitivities may not be reliable (e.g., the covariance implied by Eq. (14) may not be positive definite). We find that the classical MFVB coordinate ascent algorithms (Blei et al. (2016, Section 2.4)) and even quasi-second order methods, such as BFGS (e.g., Regier et al., 2015), may not actually find a local optimum unless run for a long time with very stringent convergence criteria. Consequently, we recommend fitting models using second-order Newton trust region methods. When the Hessian is slow to compute directly, as in Section 5, one can use the conjugate gradient trust region method of Wright and Nocedal (1999, Chapter 7), which takes advantage of fast automatic differentiation Hessian-vector products without forming or inverting the full Hessian.

5. Experiments

We now demonstrate the speed and effectiveness of linear response methods on a number of simulated and real data sets. We begin with simple simulated data to provide intuition for how linear response methods can improve estimates of covariance relative to MFVB and the Laplace approximation. We then develop linear response covariance estimates for ADVI and apply them to four real-world models and data sets taken from the Stan examples library (Stan Team, 2017). Finally, we calculate both linear response covariances and prior sensitivity measures for a large-scale industry data set. In each case, we compare linear response methods with ordinary MFVB, the Laplace approximation, and MCMC. We show that linear response methods provide the best approximation to MCMC while still retaining the speed of approximate methods. Code and instructions to reproduce the results of this section can be found in the git repository `rgiordan/CovariancesRobustnessVBPaper`.

5.1 Simple Expository Examples

In this section we provide a sequence of simple examples comparing MFVB and LRVB with Laplace approximations. These examples provide intuition for the covariance estimate $\mathrm{Cov}_{q_0}^{LR}\left(g\left(\theta\right)\right)$ and illustrate how

the sensitivity analysis motivating $\mathrm{Cov}_{q_0}^{LR}\left(g\left(\theta\right)\right)$ differs from the local posterior approximation motivating $\mathrm{Cov}_{q_{Lap}}^{Lap}\left(g\left(\theta\right)\right)$.

For each example, we will explicitly specify the target posterior $p_0\left(\theta\right)$ using a mixture of normals. This will allow us to define known target distributions with varying degrees of skewness, over-dispersion, or correlation and compare the truth with a variational approximation. Formally, for some fixed K_z, component indicators z_k, $k = 1, ..., K_z$, component probabilities π_k, locations μ_k, and covariances Σ_k, we set

$$p\left(z\right) = \prod_{k=1}^{K_z} \pi_k^{z_k}$$

$$p_0\left(\theta\right) = \sum_z p\left(z\right) p\left(\theta|z\right) = \sum_z p\left(z\right) \prod_{k=1}^{K_z} \mathcal{N}\left(\theta; m_k, \mathbf{\Sigma}_k\right)^{z_k}.$$

The values π, m and Σ will be chosen to achieve the desired shape for each example using up to $K_z = 3$ components. There will be no need to state the precise values of π, m, and Σ; rather, we will show plots of the target density and report the marginal means and variances, calculated by Monte Carlo.[1]

We will be interested in estimating the mean and variance of the first component, so we take $g\left(\theta\right) = \theta_1$. Consequently, in order to calculate $\mathrm{Cov}_{q_0}^{LR}(\theta_1)$, we will be considering the perturbation $\rho\left(\theta, \alpha\right) = \alpha\theta_1$ with scalar α and $\alpha_0 = 0$.

For the variational approximations, we will use a factorizing normal approximation:

$$\mathcal{Q}_{mf} = \left\{ q\left(\theta\right) : q\left(\theta\right) = \prod_{k=1}^{K} \mathcal{N}\left(\theta_k; \mu_k, \sigma_k^2\right) \right\}.$$

In terms of Eq. (7), we take $\eta = \left(\mu_1, ..., \mu_K, \log\sigma_1, ..., \log\sigma_K\right)^{\intercal}$. Thus $\mathbb{E}_{q(\theta;\eta)}\left[g\left(\theta\right)\right] = \mathbb{E}_{q(\theta;\eta)}\left[\theta_1\right] = \mu_1$. In the examples below, we will use multiple distinct components in the definition of $p_0\left(\theta\right)$, so that $p_0\left(\theta\right)$ is non-normal and $p_0\left(\theta\right) \notin \mathcal{Q}_{mf}$.

Since the expectation $\mathbb{E}_{q(\theta;\eta)}\left[\log p\left(\theta\right)\right]$ is intractable, we replace the exact KL divergence with a Monte Carlo approximation using the "re-parameterization trick" (Kingma and Welling, 2013; Rezende et al., 2014; Titsias and Lázaro-Gredilla, 2014). Let $\circ$ denote the Hadamard (component-wise) product. Let $\xi_m \overset{iid}{\sim} \mathcal{N}\left(0, I_K\right)$ for $m = 1, ..., M$. We define

$$\theta_m := \sigma \circ \xi_m + \mu$$

$$KL_{approx}\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right) := -\frac{1}{M}\sum_{m=1}^{M} \log p_0\left(\theta_m\right) - \sum_{k=1}^{K} \log\sigma_k,$$

which is a Monte Carlo estimate of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$. We found $M = 10000$ to be more than adequate for our present purposes of illustration. Note that we used the same draws ξ_m for both optimization and for the calculation of $\mathbf{H}_{\eta\eta}$ in order to ensure that the η_0^* at which $\mathbf{H}_{\eta\eta}$ was evaluated was in fact an optimum. This approach is similar to our treatment of ADVI; see Section 5.2 for a more detailed discussion.

5.1.1 MULTIVARIATE NORMAL TARGETS

If we take only a single component in the definition of $p_0\left(\theta\right)$ ($K_z = 1$), then $p_\alpha\left(\theta\right)$ is a multivariate normal distribution for all α, and the Laplace approximation $q_{Lap}\left(\theta\right)$ is equal to $p_\alpha\left(\theta\right)$ for all α. Furthermore, as discussed in Section 4.1 and Appendix E, the variational means $\mathbb{E}_{q_\alpha}\left[\theta\right] = \mu$ are exactly equal to the exact posterior mean $\mathbb{E}_{p_\alpha}\left[\theta\right] = m_1$ for all α (even though in general $\mathrm{Cov}_{q_0}\left(\theta\right) \neq \Sigma_1$). Consequently, for all α,

1. MFVB is often used to approximate the posterior when the Bayesian generative model for data x is a mixture model (e.g., Blei et al. (2003)). By contrast, we note for clarity that we are *not* using the mixture model as a generative model for x here. E.g., z is not one of the parameters composing θ, and we are not approximating the distribution of z in the variational distribution $q\left(\theta\right)$. Rather, we are using mixtures as a way of flexibly defining skewed and over-dispersed targets, $p\left(\theta\right)$.

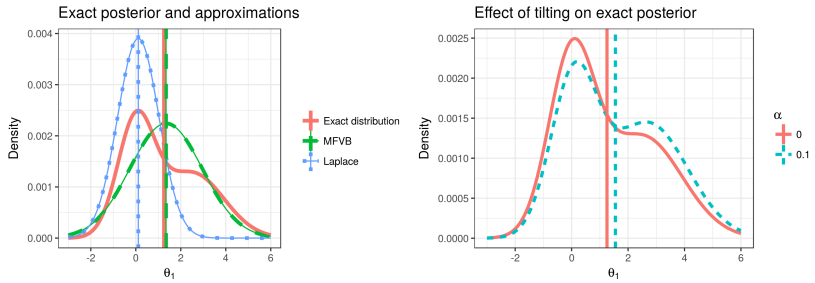
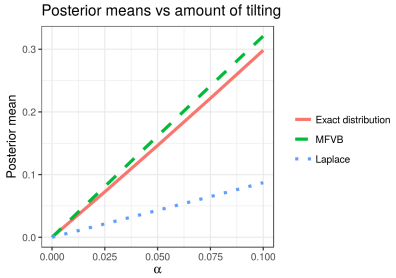


Figure 1: A univariate skewed distribution. Vertical lines show the location of the means.



Metric	Exact	LRVB	MFVB	Laplace
mean	1.253		1.345	0.111
var	2.872	3.245	2.599	0.849

Figure 2: Effect of tilting on a univariate skew distribution.

the variational approximation, the Laplace approximation, and the exact $p_0(\theta)$ all coincide in their estimates of $\mathbb{E}[\theta]$, and by, Corollary 2, $\Sigma = \mathrm{Cov}_{p_0}(\theta) = \mathrm{Cov}^{LR}_{q_0}(\theta) = \mathrm{Cov}^{Lap}_{q_{Lap}}(\theta)$. Of course, if Σ is not diagonal, $\mathrm{Cov}_{q_0}(\theta) \neq \Sigma$ because of the mean field assumption. Since this argument holds for the whole vector θ, it holds *a fortiori* for our quantity of interest, the first component $g(\theta) = \theta_1$.

In other words, the Laplace approximation will differ only from the LRVB approximation when $p_0(\theta)$ is not multivariate normal, a situation that we will now bring about by adding new components to the mixture; i.e., by increasing K_z.

5.1.2 A Univariate Skewed Distribution

If we add a second component ($K_z = 2$), then we can make $p_0(\theta)$ skewed, as shown (with the approximations) in Fig. 1. In this case, we expect $\mathbb{E}_{q_\alpha}[\theta_1]$ to be more accurate than the Laplace approximation $\mathbb{E}_{q_{Lap}}[\theta_1]$ because $\mathcal{Q}_{mf}$ is more expressive than $\mathcal{Q}_{Lap}$. This intuition is born out in the left panel of Fig. 1. Since $\hat{\theta}_{Lap}$ uses only information at the mode, it fails to take into account the mass to the right of the mode, and the Laplace approximation's mean is too far to the left. The MFVB approximation, in contrast, is quite accurate for the posterior mean of θ_1, even though it gets the overall shape of the distribution wrong.

This example also shows why, in general, one cannot naively form a "Laplace approximation" to the posterior centered at the variational mean rather than at the MAP. As shown in the left panel of Fig. 1, in this case the posterior distribution is actually convex at the MFVB mean. Consequently, a naive second-order approximation to the log posterior centered at the MFVB mean would imply a negative variance.

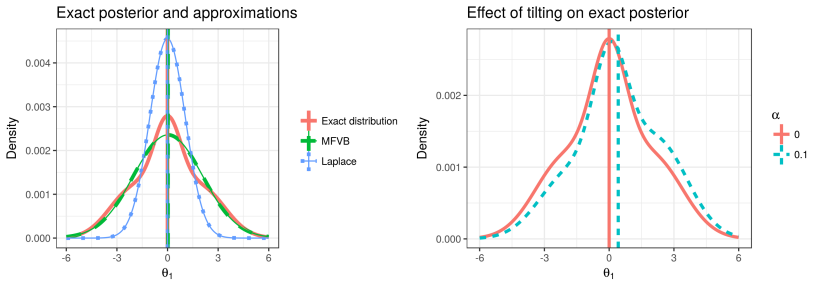


Figure 3: A univariate over-dispersed distribution. Vertical lines show the location of the means.

The perturbation $\rho(\theta, \alpha) = \alpha\theta_1$ is sometimes also described as a "tilting," and the right panel of Fig. 1 shows the effect of tilting on this posterior approximation. Tilting increases skew, but the MFVB approximation remains accurate, as shown in Fig. 2. Since local sensitivity of the expectation of θ_1 to α is the variance of θ_1 (see Eq. (13)), we have in Fig. 2 that:

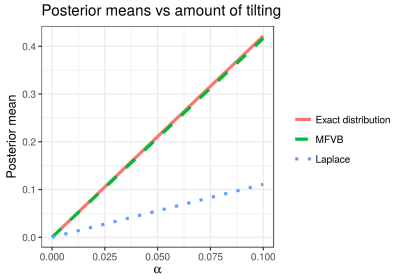
- The slope of the exact distribution's line is $\mathrm{Cov}_{p_0}(\theta_1)$;
- The slope of the MFVB line is the LRVB variance $\mathrm{Cov}^{LR}_{q_0}(\theta_1)$; and
- The slope of the Laplace line is $\mathrm{Cov}^{Lap}_{q_{Lap}}(\theta_1)$.

Since the MFVB and exact lines nearly coincide, we expect the LRVB variance estimate to be quite accurate for this example. Similarly, since the slope of the Laplace approximation line is lower, we expect the Laplace variance to underestimate the exact variance. This outcome, which can be seen visually in the left-hand panel of Fig. 2, is shown quantitatively in the corresponding table in the right-hand panel. The columns of the table contain information for the exact distribution and the three approximations. The first row, labeled "mean," shows $\mathbb{E}[\theta_1]$ and the second row, labeled "var," shows $\mathrm{Cov}(\theta_1)$. (The "LRVB" entry for the mean is blank because LRVB differs from MFVB only in covariance estimates.) We conclude that, in this case, Condition 1 holds for $\mathcal{Q}_{mf}$ but not for $\mathcal{Q}_{Lap}$.

5.1.3 A Univariate Over-dispersed Distribution

Having seen how MFVB can outperform the Laplace approximation for a univariate skewed distribution, we now apply that intuition to see why the linear response covariance can be superior to the Laplace approximation covariance for over-dispersed but symmetric distributions. Such a symmetric but over-dispersed distribution, formed with $K_z = 3$ components, is shown in Fig. 3 together with its approximations. By symmetry, both the MFVB and Laplace means are exactly correct (up to Monte Carlo error), as can be seen in the left panel of Fig. 3.

However, the right panel of Fig. 3 shows that symmetry is not maintained as the distribution is tilted. For $\alpha > 0$, the distribution becomes skewed to the right. Thus, by the intuition from the previous section, we expect the MFVB mean to be more accurate as the distribution is tilted and α increases from zero. In particular, we expect that the Laplace approximation's mean will not shift enough as α varies, i.e., that the Laplace approximation variance will be underestimated. Fig. 4 shows that this is indeed the case. The slopes in the left panel once again correspond to the estimated variances shown in the table, and, as expected the LRVB variance estimate is superior to the Laplace approximation variance.



Metric	Exact	LRVB	MFVB	Laplace
mean	-0.001		0.027	-0.000
var	4.218	4.153	4.161	1.107

Figure 4: Effect of tilting on a univariate over-dispersed distribution.

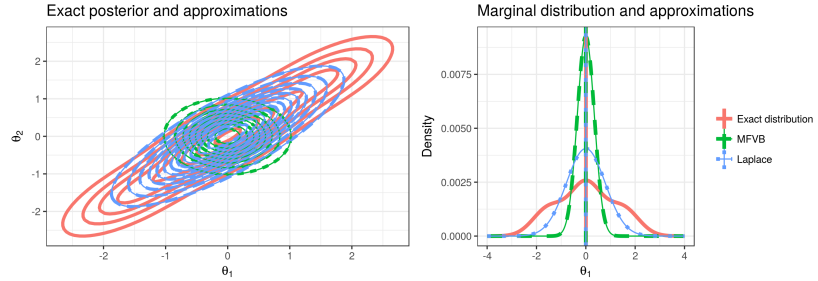
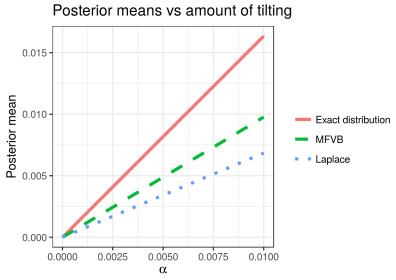


Figure 5: A bivariate over-dispersed distribution.

In this case, Condition 1 holds for $\mathcal{Q}_{mf}$. For the Laplace approximation, $\mathbb{E}_{q_{Lap}}[g(\theta)] = \mathbb{E}_{p_0}[g(\theta)]$ for $\alpha = 0$, so $\mathcal{Q}_{Lap}$ satisfies Eq. (11) of Condition 1 for α near zero, the derivatives of the two expectations with respect to α are quite different, so Eq. (12) of Condition 1 does not hold for $\mathcal{Q}_{Lap}$.

5.1.4 A BIVARIATE OVER-DISPERSED DISTRIBUTION

In the previous two examples the mean field approximation in $\mathcal{Q}$ did not matter, since the examples were one-dimensional. The only reason that the variational approximation was different from the exact $p_0(\theta)$ was the normal assumption in $\mathcal{Q}_{mf}$. Indeed, the tables in Fig. 2 and Fig. 4 show that the MFVB variance estimate is also reasonably close to the exact variance. In order to demonstrate why the LRVB variance can be better than both the Laplace approximation and the MFVB approximation, we turn to a bivariate, correlated, over-dispersed $p_0(\theta)$. For this we use $K_z = 3$ correlated normal distributions, shown in the left panel of Fig. 5. The right panel of Fig. 5 shows the marginal distribution of θ_1, in which the over-dispersion can be seen clearly. As Fig. 5 shows, unlike in the previous two examples, the mean field approximation causes $q_0(\theta)$ to dramatically underestimate the marginal variance of θ_1. Consequently, the MFVB means will also be under-responsive to the skew introduced by tilting with α. Though the Laplace approximation has a larger marginal variance, it remains unable to take skewness into account. Consequently, as seen in Fig. 6, the LRVB variance, while not exactly equal to the correct variance, is still an improvement over the Laplace covariance, and a marked improvement on the badly under-estimated MFVB variance.



Metric	Exact	LRVB	MFVB	Laplace
mean	0.005		-0.002	-0.000
var	1.635	0.976	0.241	0.684

Figure 6: Effect of tilting on a bivariate over-dispersed distribution.

One might say, in this case, that Condition 1 does not hold for either $\mathcal{Q}_{mf}$ or $\mathcal{Q}_{Lap}$, or, if it does, it is with a liberal interpretation of the "approximately equals" sign. However, the expressiveness of $\mathcal{Q}_{mf}$ allows LRVB to improve on the Laplace approximation, and the linear response allows it to improve over the MFVB approximation, and so LRVB gives the best of both worlds.

Thinking about problems in terms of these three simple models can provide intuition about when and whether Condition 1 might be expected to hold in a sense that is practically useful.

5.2 Automatic Differentiation Variational Inference (ADVI)

In this section we apply our methods to automatic differentiation variational inference (ADVI) (Kucukelbir et al., 2017). ADVI is a "black-box" variational approximation and optimization procedure that requires only that the user provide the log posterior, $\log p_0(\theta)$, up to a constant that does not depend on θ. To achieve this generality, ADVI employs:

- A factorizing normal variational approximation,[2]
- An unconstraining parameterization,
- The "re-parameterization trick," and
- Stochastic gradient descent.

ADVI uses a family employing the factorizing normal approximation

$$\mathcal{Q}_{ad} := \left\{ q(\theta) : q(\theta) = \prod_{k=1}^{K} \mathcal{N}(\theta_k | \mu_k, \exp(2\zeta_k)) \right\}.$$

That is, $\mathcal{Q}_{ad}$ is a fully factorizing normal family with means μ_k and log standard deviations ζ_k. Because we are including exponential family assumptions in the definition of MFVB (as described in Section 3.1), $\mathcal{Q}_{ad}$ is an instance of a mean-field family $\mathcal{Q}_{mf}$. In the notation of Eq. (7),

$$\eta = (\mu_1, ..., \mu_K, \zeta_1, ..., \zeta_K)^\intercal, \tag{22}$$

2. Kucukelbir et al. (2017) describe a non-factorizing version of ADVI, which is called "fullrank" ADVI in Stan. The factorizing version that we describe here is called "meanfield" ADVI in Stan. On the examples we describe, in the current Stan implementation, we found that fullrank ADVI provided much worse approximations to the MCMC posterior means than the meanfield version, and so we do not consider it further.

$\Omega_\eta = \mathbb{R}^{2K}$, λ is the Lebesgue measure, and the objective function Eq. (8) is

$$KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right) = -\int \mathcal{N}\left(\theta_k|\mu_k, \exp\left(2\zeta_k\right)\right)\log p_0\left(\theta\right)\lambda\left(d\theta\right) - \sum_{k=1}^{K}\zeta_k,$$

where we have used the form of the univariate normal entropy up to a constant.

The unconstraining parameterization is required because the use of a normal variational approximation dictates that the base measure on the parameters $\theta \in \mathbb{R}^K$ be supported on all of $\mathbb{R}^K$. Although many parameters of interest, such as covariance matrices, are not supported on $\mathbb{R}^K$, there typically exist differentiable maps from an unconstrained parameterization supported on $\mathbb{R}^K$ to the parameter of interest. Software packages such as Stan automatically provide such transforms for a broad set of parameter types. In our notation, we will take these constraining maps to be the function of interest, $g\left(\theta\right)$, and take θ to be unconstrained. Note that, under this convention, the prior $p\left(\theta|\alpha\right)$ must be a density in the unconstrained space. In practice (e.g., in the Stan software package), one usually specifies the prior density in the constrained space and converts it to a density $p\left(\theta|\alpha\right)$ in the unconstrained space using the determinant of the Jacobian of the constraining transform $g\left(\cdot\right)$.

The re-parameterization trick allows easy approximation of derivatives of the (generally intractable) objective $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$. By defining z_k using the change of variable

$$z_k := \left(\theta_k - \mu_k\right)/\exp\left(\zeta_k\right), \tag{23}$$

$KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$ can be re-written as an expectation with respect to a standard normal distribution. We write $\theta = \exp\left(\zeta\right)\circ z + \mu$ by using the component-wise Hadamard product $\circ$. Then

$$KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right) = -\mathbb{E}_z\left[\log p_0\left(\exp\left(\zeta\right)\circ z + \mu\right)\right] - \sum_{k=1}^{K}\zeta_k + Constant.$$

The expectation is still typically intractable, but it can be approximated using Monte Carlo and draws from a K-dimensional standard normal distribution. For a fixed number M of draws $z_1, ..., z_M$ from a standard K-dimensional normal, we can define the approximate KL divergence

$$\widehat{KL}\left(\eta\right) := -\frac{1}{M}\sum_{m=1}^{M}\log p_0\left(\exp\left(\zeta\right)\circ z_m + \mu\right) - \sum_{k=1}^{K}\zeta_k + Constant. \tag{24}$$

For any fixed M,

$$\mathbb{E}\left[\frac{\partial}{\partial\eta}\widehat{KL}\left(\eta\right)\right] = \frac{\partial}{\partial\eta}KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right),$$

so gradients of $\widehat{KL}\left(\eta\right)$ are unbiased for gradients of the exact KL divergence. Furthermore, for fixed draws $z_1, ..., z_M$, $\widehat{KL}\left(\eta\right)$ can be easily differentiated (using, again, the re-parameterization trick). Standard ADVI uses this fact to optimize $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$ using the unbiased gradient draws $\frac{\partial}{\partial\eta}\widehat{KL}\left(\eta\right)$ and a stochastic gradient optimization method, where the stochasticity comes from draws of the standard normal random variable z. Note that stochastic gradient methods typically use a new draw of z at every gradient step.

5.2.1 LINEAR RESPONSE FOR ADVI (LR-ADVI)

Since ADVI uses a factorizing normal approximation, the intuition from Section 5.1 may be expected to apply. In particular, we might expect that the ADVI means $\hat{\mu}$ might be a good approximation to $\mathbb{E}_{p_0}\left[\theta\right]$, that the ADVI variances $\exp\left(2\hat{\zeta}\right)$ would be under-estimates of the posterior variance $\mathrm{Cov}_{p_0}\left(\theta\right)$, so that using $\mathrm{Cov}_{q_0}^{LR}\left(\theta\right)$ could improve the approximations to the posterior variance. We refer to LRVB covariances calculated using an ADVI approximation as LR-ADVI.

To apply linear response to an ADVI approximation, we need to be able to approximate the Hessian of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$ and to be assured that we have found an optimal η_0^*. But, by using a stochastic gradient method, ADVI avoids ever actually calculating the expectation in $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$. Furthermore even if a stochastic gradient method finds an point that is close to the optimal value of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$ it may not be close to an optimum of $\widehat{KL}\left(\eta\right)$ for a particular finite M. Indeed, we found that, even for very large M, the optimum found by ADVI's stochastic gradient method is typically not close enough to an optimum of the approximate $\widehat{KL}\left(\eta\right)$ for sensitivity calculations to be useful. Sensitivity calculations are based on differentiating the fixed point equation given by the gradient being zero (see the proof in Appendix D), and do not apply at points for which the gradient is not zero either in theory nor in practice.

Consequently, in order to calculate the local sensitivity, we simply eschew the stochastic gradient method and directly optimize $\widehat{KL}\left(\eta\right)$ for a particular choice of M. (We will discuss shortly how to choose M.) We can then use $\widehat{KL}\left(\eta\right)$ in Eq. (10) rather than the exact KL divergence. Directly optimizing $\widehat{KL}\left(\eta\right)$ both frees us to use second-order optimization methods, which we found to converge more quickly to a high-quality optimum than first-order methods, and guarantees that we are evaluating the Hessian $\mathbf{H}_{\eta\eta}$ at an optimum of the objective function used to calculate Eq. (10).

As M approaches infinity, we expect the optimum of $\widehat{KL}\left(\eta\right)$ to approach the optimum of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$ by the standard frequentist theory of estimating equations (Keener, 2010, Chapter 9). In practice we must fix a particular finite M, with larger M providing better approximations of the true KL divergence but at increased computational cost. We can inform this tradeoff between accuracy and computation by considering the frequentist variability of η_0^* when randomly sampling M draws of the random variable z used to approximate the intractable integral in $\widehat{KL}\left(\eta\right)$. Denoting this frequentist variability by $\mathrm{Cov}_z\left(\eta_0^*\right)$, standard results (Keener, 2010, Chapter 9) give that

$$\mathrm{Cov}_z\left(\eta_0^*\right) \approx \mathbf{H}_{\eta\eta}^{-1}\mathrm{Cov}_z\left(\left.\frac{\partial}{\partial\eta}\widehat{KL}\left(\eta\right)\right|_{\eta_0^*}\right)\mathbf{H}_{\eta\eta}^{-1}. \tag{25}$$

A sufficiently large M will be one for which $\mathrm{Cov}_z\left(\eta_0^*\right)$ is adequately small. One notion of "adequately small" might be that the ADVI means found with $\widehat{KL}\left(\eta\right)$ are within some fraction of a posterior standard deviation of the optimum of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$. Having chosen a particular M, we can calculate the frequentist variability of μ^* using $\mathrm{Cov}_{q_0}^{LR}\left(g\left(\theta\right)\right)$ and estimate the posterior standard deviation using Eq. (14). If we find that each μ^* is probably within 0.5 standard deviations of the optimum of $KL\left(q\left(\theta;\eta\right)||p_0\left(\theta\right)\right)$, we can keep the results; otherwise, we increase M and try again. In the examples we consider here, we found that the relatively modest $M = 10$ satisfies this condition and provides sufficiently accurate results.

Finally, we note a minor departure from Eq. (14) when calculating $\mathrm{Cov}_{q_0}^{LR}\left(g\left(\theta\right)\right)$ from $\mathbf{H}_{\eta\eta}$. Recall that, in this case, we are taking $g\left(\cdot\right)$ to be ADVI's constraining transform, and that Eq. (14) requires the Jacobian, $\mathbf{g}_\eta$, of this transform. At the time of writing, the design of the Stan software package did not readily support automatic calculation of $\mathbf{g}_\eta$, though it did support rapid evaluation of $g\left(\theta\right)$ at particular values of θ. Consequently, we used linear response to estimate $\mathrm{Cov}_{q_0}^{LR}\left(\theta\right)$, drew a large number N_s of Monte Carlo draws from $\theta_n \sim \mathcal{N}\left(\mu, \mathrm{Cov}_{q_0}^{LR}\left(\theta\right)\right)$ for $n = 1, ..., N_s$, and then used these draws to form a Monte Carlo estimate of the sample covariance of $g\left(\theta\right)$. Noting that $\mathbb{E}_{q_\alpha}\left[\theta\right] = \mu$, and recalling the definition of η for ADVI in Eq. (22), by Eq. (14) we have

$$\mathrm{Cov}_{q_0}^{LR}\left(\theta\right) = \frac{\partial\mathbb{E}_{q_\alpha}\left[\theta\right]}{\partial\eta^\intercal}\mathbf{H}_{\eta\eta}^{-1}\frac{\partial\mathbb{E}_{q_\alpha}\left[\theta^\intercal\right]}{\partial\eta} = \begin{pmatrix} I_K & 0 \\ 0 & 0 \end{pmatrix}\mathbf{H}_{\eta\eta}^{-1}\begin{pmatrix} I_K & 0 \\ 0 & 0 \end{pmatrix},$$

which is the upper-left quarter of the matrix $\mathbf{H}_{\eta\eta}^{-1}$. In addition to obviating the need for $\mathbf{g}_\eta$, this approach also allowed us to take into account possible nonlinearities in $g\left(\cdot\right)$ at little additional computational cost.

5.2.2 RESULTS

We present results from four models taken from the Stan example set, namely the models `election88` ("Election model"), `sesame_street1` ("Sesame Street model"), `radon_vary_intercept_floor` ("Radon

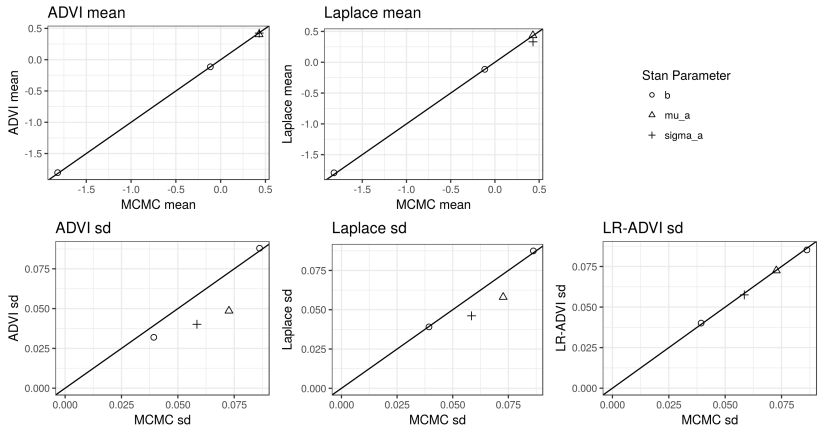


Figure 7: Election model

model"), and `cjs_cov_randeff` ("Ecology model"). We experimented with many models from the Stan examples and selected these four as representative of the type of model where LR-ADVI can be expected to provide a benefit—specifically, they are models of a moderate size. For very small models, MCMC runs quickly enough in Stan that fast approximations are not necessary, and for very large models (with thousands of parameters) the relative advantages of LR-ADVI and the Laplace approximation diminish due to the need to calculate $\mathbf{H}_{\eta\eta}$ or $\mathbf{H}_{Lap}$ using automatic differentiation.[3] The size of the data and size of the parameter space for our four chosen models are shown in Fig. 11. We also eliminated from consideration models where Stan's MCMC algorithm reported divergent transitions or where Stan's ADVI algorithm returned wildly inaccurate posterior mean estimates.

For brevity, we do not attempt to describe the models or data in any detail here; rather, we point to the relevant literature in their respective sections. The data and Stan implementations themselves can be found on the Stan website (Stan Team, 2017) as well as in Appendix F.

To assess the accuracy of each model, we report means and standard deviations for each of Stan's model parameters as calculated by Stan's MCMC and ADVI algorithms and a Laplace approximation, and we report the standard deviations as calculated by $\mathrm{Cov}_{q_0}^{LR}\left(g\left(\theta\right)\right)$. Recall that, in our notation, $g\left(\cdot\right)$ is the (generally nonlinear) map from the unconstrained latent ADVI parameters to the constrained space of the parameters of interest. The performance of ADVI and Laplace vary, and only LR-ADVI provides a consistently good approximation to the MCMC standard deviations. LR-ADVI was somewhat slower than a Laplace approximation or ADVI alone, but it was typically about five times faster than MCMC; see Section 5.2.7 for detailed timing results.

5.2.3 ELECTION MODEL ACCURACY

We begin with `election88`, which models binary responses in a 1988 poll using a Bernoulli hierarchical model with normally distributed random effects for state, ethnicity, and gender and a logit link. The model and data are described in detail in Gelman and Hill (2006, Chapter 14). Fig. 7 shows that both the Laplace approximation and ADVI do a reasonable job of matching to MCMC, though LR-ADVI is slightly more accurate for standard deviations.

3. We calculated $\mathbf{H}_{\eta\eta}$ using a custom branch of Stan's automatic differentiation software (Carpenter et al., 2015) that exposes Hessians and Hessian-vector products in the `Rstan modelfit` class. When this custom branch is merged with the main branch of Stan, it will be possible to implement LR-ADVI for generic Stan models.

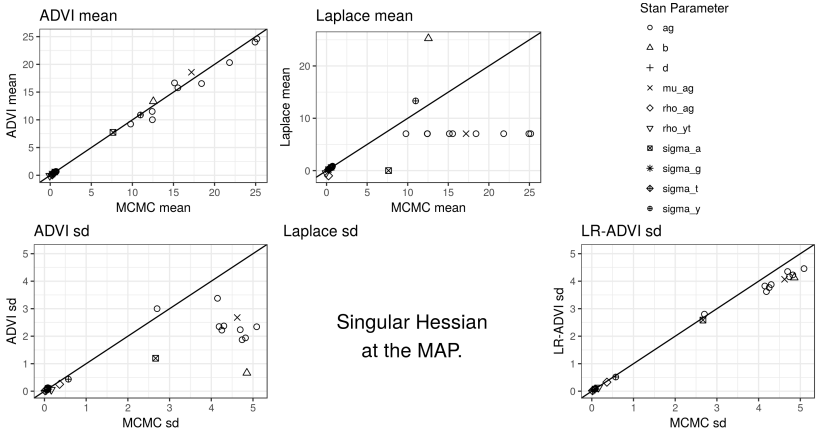


Figure 8: Sesame Street model

5.2.4 SESAME STREET MODEL ACCURACY

Next, we show results for `sesame_street1`, an analysis of a randomized controlled trial designed to estimate the causal effect of watching the television show Sesame Street on a letter-recognition test. To control for different conditions in the trials, a hierarchical model is used with correlated multivariate outcomes and unknown covariance structure. The model and data are described in detail in Gelman and Hill (2006, Chapter 23).

As can be seen in Fig. 8, the MAP under-estimates the variability of the random effects ag, and, in turn, under-estimates the variance parameter $sigma\_a$. Because the MAP estimate of $sigma\_a$ is close to zero, the log posterior has a very high curvature with respect to the parameter ag at the MAP, and the Hessian used for the Laplace approximation is numerically singular. ADVI, which integrates out the uncertainty in the random effects, provides reasonably good estimates of the posterior means but underestimates the posterior standard deviations due to the mean-field assumption. Only LR-ADVI provides accurate estimates of posterior uncertainty.

5.2.5 RADON MODEL ACCURACY

We now turn to `radon_vary_intercept_floor`, a hierarchical model of radon levels in Minnesota homes described in Gelman and Hill (2006, Chapters 16 and 21). This model is relatively simple, with univariate normal observations and unknown variances. Nevertheless, the Laplace approximation again produces a numerically singular covariance matrix. The ADVI means are reasonably accurate, but the standard deviations are not. Only LR-ADVI produces an accurate approximation to the MCMC posterior standard deviations.

5.2.6 ECOLOGY MODEL ACCURACY

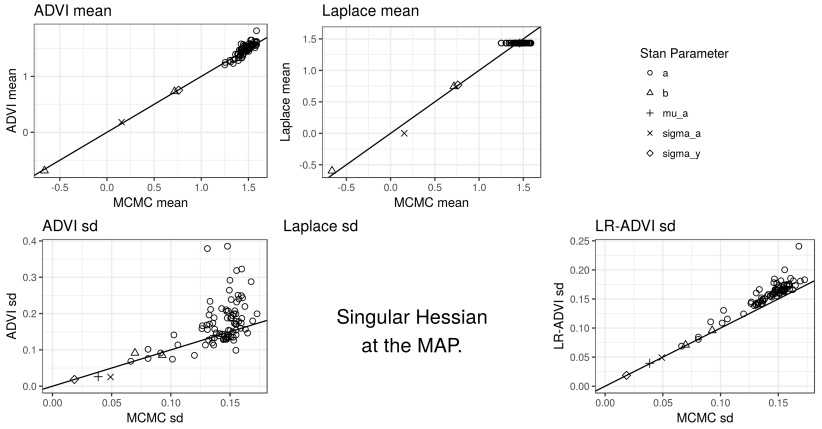


Figure 9: Radon model

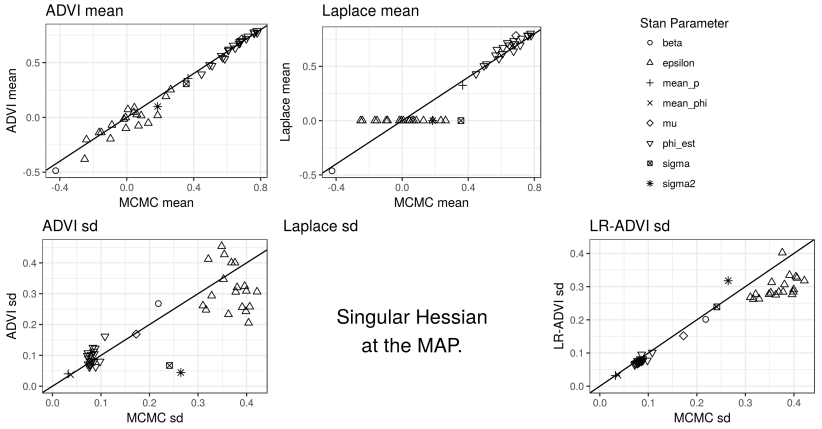


Figure 10: Ecology model

Finally, we consider a more complicated mark-recapture model from ecology known as the Cormack-Jolly-Seber (CJS) model. This model is described in detail in Kéry and Schaub (2011, Chapter 7), and discussion of the Stan implementation can be found in Stan Team (2015, Section 15.3).

The Laplace approximation is again degenerate, and the ADVI standard deviations again deviate considerably from MCMC. In this case, the ADVI means are also somewhat inaccurate, and some of the LR-ADVI standard deviations are mis-estimated in turn. However, LR-ADVI remains by far the most accurate method for approximating the MCMC standard errors.

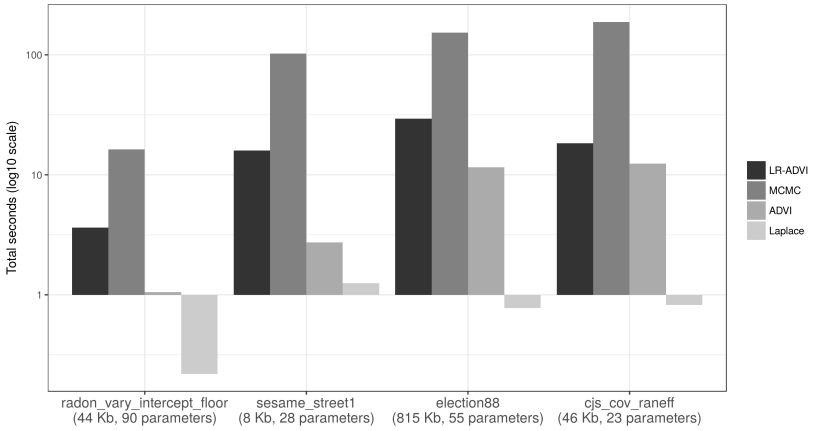


Figure 11: Comparision of timing in ADVI experiments

5.2.7 TIMING RESULTS

Detailed timing results for the ADVI experiments are shown in Fig. 11. Both the Laplace approximation and ADVI alone are faster than LR-ADVI, which in turn is about five times faster than MCMC. We achieved the best results optimizing $\widehat{KL}(\eta)$ by using the conjugate gradient Newton's trust region method (`trust-ncg` of `scipy.optimize`), but the optimization procedure still accounted for an appreciable proportion of the time needed for LR-ADVI.

5.3 Criteo Dataset

We now apply our methods to a real-world data set using a logistic regression with random effects, which is an example of a generalized linear mixed model (GLMM) (Agresti and Kateri, 2011, Chapter 13). This data and model have several advantages as an illustration of our methods: the data set is large, the model contains a large number of imprecisely-estimated latent variables (the unknown random effects), the model exhibits the sparsity of $\mathbf{H}_{\eta\eta}$ that is typical in many MFVB applications, and the results exhibit the same shortcomings of the Laplace approximation seen above. For this model, we will evaluate both posterior covariances and prior sensitivities.

5.3.1 DATA AND MODEL

We investigated a custom subsample of the 2014 Criteo Labs conversion logs data set (Criteo Labs, 2014), which contains an obfuscated sample of advertising data collected by Criteo over a period of two months. Each row of the data set corresponds to a single user click on an online advertisement. For each click, the data set records a binary outcome variable representing whether or not the user subsequently "converted" (i.e., performed a desired task, such as purchasing a product or signing up for a mailing list). Each row contains two timestamps (which we ignore), eight numerical covariates, and nine factor-valued covariates. Of the eight numerical covariates, three contain 30% or more missing data, so we discarded them. We then applied a per-covariate normalizing transform to the distinct values of those remaining. Among the factor-valued covariates, we retained only the one with the largest number of unique values and discarded the others.

These data-cleaning decisions were made for convenience. The goal of the present paper is to demonstrate our inference methods, not to draw conclusions about online advertising.

Although the meaning of the covariates has been obfuscated, for the purpose of discussion we will imagine that the single retained factor-valued covariate represents the identity of the advertiser, and the numeric covariates represent salient features of the user and/or the advertiser (e.g., how often the user has clicked or converted in the past, a machine learning rating for the advertisement quality, etc.). As such, it makes sense to model the probability of each row's binary outcome (whether or not the user converted) as a function of the five numeric covariates and the advertiser identity using a logistic GLMM. Specifically, we observe binary conversion outcomes, y_{it}, for click i on advertiser t, with probabilities given by observed numerical explanatory variables, x_{it}, each of which are vectors of length $K_x = 5$. Additionally, the outcomes within a given value of t are correlated through an unobserved random effect, u_t, which represents the "quality" of advertiser t, where the value of t for each observation is given by the factor-valued covariate. The random effects u_t are assumed to follow a normal distribution with unknown mean and variance. Formally,

$$\begin{aligned} y_{it}|p_{it} &\sim \text{Bernoulli}\left(p_{it}\right), \text{ for } t = 1, ..., T \text{ and } i = 1, ..., N_t \\ p_{it} &:= \frac{e^{\rho_{it}}}{1 + e^{\rho_{it}}} \quad \text{where} \quad \rho_{it} := x_{it}^T \beta + u_t \\ u_t|\mu, \tau &\sim \mathcal{N}\left(\mu, \tau^{-1}\right). \end{aligned}$$

Consequently, the unknown parameters are $\theta = (\beta^\intercal, \mu, \tau, u_1, ..., u_T)^\intercal$. We use the following priors:

$$\begin{aligned} \mu|\mu_0, \tau_\mu &\sim \mathcal{N}\left(\mu_0, \tau_\mu^{-1}\right) \\ \tau|\alpha_\tau, \beta_\tau &\sim \text{Gamma}\left(\alpha_\tau, \beta_\tau\right) \\ \beta|\beta_0, \tau_\beta, \gamma_\beta &\sim \mathcal{N}\left(\begin{pmatrix} \beta_0 \\ \vdots \\ \beta_0 \end{pmatrix}, \begin{pmatrix} \tau_\beta & \gamma_\beta & \gamma_\beta \\ \gamma_\beta & \ddots & \gamma_\beta \\ \gamma_\beta & \gamma_\beta & \tau_\beta \end{pmatrix}^{-1}\right). \end{aligned}$$

Note that we initially take $\gamma_\beta = 0$ so that the prior information matrix on β is diagonal. Nevertheless, by retaining γ_β as a hyperparameter we will be able to assess the sensitivity to the assumption of a diagonal prior in Section 5.3.6. The remaining prior values are given in Appendix H. It is reasonable to expect that a modeler would be interested both in the effect of the numerical covariates and in the quality of individual advertisers themselves, so we take the parameter of interest to be $g(\theta) = (\beta^\intercal, u_1, ..., u_T)^\intercal$.

To produce a data set small enough to be amenable to MCMC but large and sparse enough to demonstrate our methods, we subsampled the data still further. We randomly chose 5000 distinct advertisers to analyze, and then subsampled each selected advertiser to contain no more than 20 rows each. The resulting data set had $N = 61895$ total rows. If we had more observations per advertiser, the "random effects" u_t would have been estimated quite precisely, and the nonlinear nature of the problem would not have been important; these changes would thus have obscured the benefits of using MFVB versus the Laplace approximation. In typical internet data sets a large amount of data comes from advertisers with few observations each, so our subsample is representative of practically interesting problems.

5.3.2 INFERENCE AND TIMING

We estimated the expectation and covariance of $g(\theta)$ using four techniques: MCMC, the Laplace approximation, MFVB, and linear response (LRVB) methods. For MCMC, we used Stan (Stan Team, 2015), and to calculate the MFVB, Laplace, and LRVB estimates we used our own Python code using `numpy`, `scipy`, and `autograd` (Jones et al., 2001; Maclaurin et al., 2015). As described in Section 5.3.3, the MAP estimator did not estimate $\mathbb{E}_{p_0}[g(\theta)]$ very well, so we do not report standard deviations or sensitivity measures for the Laplace approximations. The summary of the computation time for all these methods is shown in Table 1, with details below.

Method	Seconds
MAP (optimum only)	12
VB (optimum only)	57
VB (including sensitivity for β)	104
VB (including sensitivity for β and u)	553
MCMC (Stan)	21066

Table 1: Timing results

For the MCMC estimates, we used Stan to draw 5000 MCMC draws (not including warm-up), which took 351 minutes. We estimated all the prior sensitivities of Section 5.3.6 using the Monte Carlo version of the covariance in Eq. (5).

For the MFVB approximation, we use the following mean field exponential family approximations:

$$\begin{aligned} q(\beta_k) &= \mathcal{N}\left(\beta_k; \eta_{\beta_k}\right), \text{ for } k = 1, ..., K_x \\ q(u_t) &= \mathcal{N}\left(u_t; \eta_{u_t}\right), \text{ for } t = 1, ..., T \\ q(\tau) &= \text{Gamma}\left(\tau; \eta_\tau\right) \\ q(\mu) &= \mathcal{N}\left(\mu; \eta_\mu\right) \\ q(\theta) &= q(\tau)\, q(\mu) \prod_{k=1}^{K_x} q(\beta_k) \prod_{t=1}^{T} q(u_t). \end{aligned}$$

With these choices, evaluating the variational objective requires the following intractable univariate variational expectation:

$$\mathbb{E}_{q(\theta;\eta)}\left[\log\left(1 - p_{it}\right)\right] = \mathbb{E}_{q(\theta;\eta)}\left[\log\left(1 - \frac{e^{\rho_{it}}}{1 + e^{\rho_{it}}}\right)\right].$$

We used the re-parameterization trick and four points of Gauss-Hermite quadrature to estimate this integral for each observation. See Appendix H for more details.

We optimized the variational objective using the conjugate gradient Newton's trust region method, `trust-ncg`, of `scipy.optimize`. One advantage of `trust-ncg` is that it performs second-order optimization but requires only Hessian-vector products, which can be computed quickly by `autograd` without constructing the full Hessian. The MFVB fit took 57 seconds, roughly 370 times faster than MCMC with Stan.

With variational parameters for each random effect u_t, $\mathbf{H}_{\eta\eta}$ is a 10014×10014 dimensional matrix. Consequently, evaluating $\mathbf{H}_{\eta\eta}$ directly as a dense matrix using `autograd` would have been prohibitively time-consuming. Fortunately, our model can be decomposed into global and local parameters, and the Hessian term $\mathbf{H}_{\eta\eta}$ in Theorem 2 is extremely sparse. In the notation of Section 4.5, take $\theta_{glob} = (\beta^\intercal, \mu, \tau)^\intercal$, take $\theta_{loc,t} = u_t$, and stack the variational parameters as $\eta = \left(\eta_{glob}^\intercal, \eta_{loc,1}, ..., \eta_{loc,T}\right)^\intercal$. The cross terms in $\mathbf{H}_{\eta\eta}$ between the local variables vanish:

$$\frac{\partial^2 KL\left(q(\theta;\eta)\,||p_\alpha(\theta)\right)}{\partial \eta_{loc,t_1} \partial \eta_{loc,t_2}} = 0 \text{ for all } t_1 \neq t_2.$$

Equivalently, note that the full likelihood in Appendix H, Eq. (31), has no cross terms between u_{t_1} and u_{t_2} for $t_1 \neq t_2$. As the dimension T of the data grows, so does the length of η. However, the dimension of η_{glob} remains constant, and $\mathbf{H}_{\eta\eta}$ remains easy to invert. We show an example of the sparsity pattern of the first few rows and columns of $\mathbf{H}_{\eta\eta}$ in Fig. 12 .

Taking advantage of this sparsity pattern, we used `autograd` to calculate the Hessian of the KL divergence one group at a time and assembled the results in a sparse matrix using the `scipy.sparse` Python

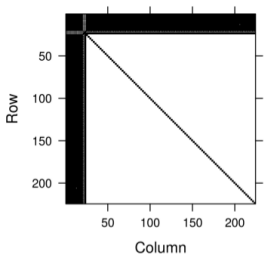


Figure 12: Sparsity pattern of top-left sub-matrix of $\mathbf{H}_{\eta\eta}$ for the logit GLMM model. The axis numbers represent indices within η, and black indicates non-zero entries of $\mathbf{H}_{\eta\eta}$.

package. Even so, calculating the entire sparse Hessian took 323 seconds, and solving the system $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$ using `scipy.sparse.linalg.spsolve` took an additional 173 seconds. These results show that the evaluation and inversion of $\mathbf{H}_{\eta\eta}$ was several times more costly than optimizing the variational objective itself. (Of course, the whole procedure remains much faster than running MCMC with Stan.)

We note, however, that instead of the direct approach to calculating $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$ one can use the conjugate gradient algorithm of `sp.sparse.linalg.cg` (Wright and Nocedal, 1999, Chapter 5) together with the fast Hessian-vector products of `autograd` to query one column at a time of $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$. On a typical column of $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$ in our experiment, calculating the conjugate gradient took only 9.4 seconds (corresponding to 81 Hessian-vector products in the conjugate gradient algorithm). Thus, for example, one could calculate the columns of $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$ corresponding to the expectations of the global variables β in only $9.4 \times K_x = 46.9$ seconds, which is much less time than it would take to compute the entire $\mathbf{H}_{\eta\eta}^{-1}\mathbf{g}_\eta^\intercal$ for both β and every random effect in u.

For the Laplace approximation, we calculated the MAP estimator and $\mathbf{H}_{Lap}$ using Python code similar to that used for the MFVB estimates. We observe that the MFVB approximation to posterior means would be expected to improve on the MAP estimator only in cases when there is both substantial uncertainty in some parameters and when this uncertainty, through nonlinear dependence between parameters, affects the values of posterior means. These circumstances obtain in the logistic GLMM model with sparse per-advertiser data since the random effects u_t will be quite uncertain and the other posterior means depend on them through the nonlinear logistic function.

5.3.3 POSTERIOR APPROXIMATION RESULTS

In this section, we assess the accuracy of the MFVB, Laplace, and LRVB methods as approximations to $\mathbb{E}_{p_0}[g(\theta)]$ and $\mathrm{Cov}_{p_0}(g(\theta))$. We take the MCMC estimates as ground truth. Although, as discussed in Section 5.3, we are principally interested in the parameters $g(\theta) = (\beta^\intercal, u_1, ..., u_T)^\intercal$, we will report the results for all parameters for completeness. For readability, the tables and graphs show results for a random selection of the components of the random effects u.

5.3.4 POSTERIOR MEANS

We begin by comparing the posterior means in Table 2, Fig. 13, and Fig. 14. We first note that, despite the long running time for MCMC, the β_1 and μ parameters did not mix well in the MCMC sample, as is reflected in the MCMC standard error and effective number of draws columns of Table 2. The x_{it} data corresponding to β_1 contained fewer distinct values than the other columns of x, which perhaps led to some co-linearity between β_1 and μ in the posterior. This co-linearity could have caused both poor MCMC mixing

Parameter	MCMC	MFVB	MAP	MCMC std. err.	Eff. # of MCMC draws
β_1	1.454	1.447	1.899	0.02067	33
β_2	0.031	0.033	0.198	0.00025	5000
β_3	0.110	0.110	0.103	0.00028	5000
β_4	-0.172	-0.173	-0.173	0.00016	5000
β_5	0.273	0.273	0.280	0.00042	5000
μ	2.041	2.041	3.701	0.04208	28
τ	0.892	0.823	827.724	0.00051	1232
u_{1431}	1.752	1.757	3.700	0.00937	5000
u_{4150}	1.217	1.240	3.699	0.01022	5000
u_{4575}	2.427	2.413	3.702	0.00936	5000
u_{4685}	3.650	3.633	3.706	0.00862	5000

Table 2: Results for the estimation of the posterior means

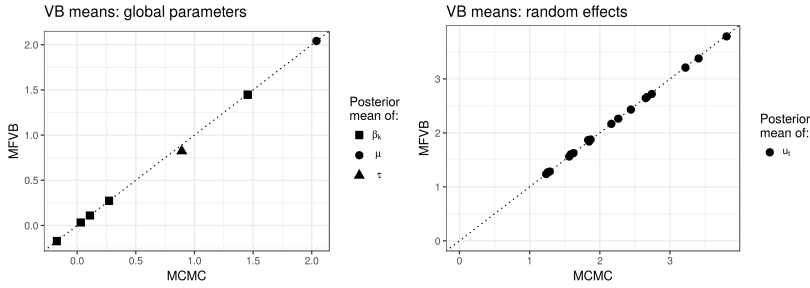


Figure 13: Comparison of MCMC and MFVB means

and, perhaps, excessive measured prior sensitivity, as discussed below in Section 5.3.6. Although we will report the results for both β_1 and μ without further comment, the reader should bear in mind that the MCMC "ground truth" for these two parameters is somewhat suspect.

The results in Table 2 and Fig. 13 show that MFVB does an excellent job of approximating the posterior means in this particular case, even for the random effects u and the related parameters μ and τ. In contrast, the MAP estimator does reasonably well only for certain components of β and does extremely poorly for the random effects parameters. As can be seen in Fig. 14, the MAP estimate dramatically overestimates the information τ of the random effect distribution (that is, it underestimates the variance). As a consequence, it estimates all the random effects to have essentially the same value, leading to mis-estimation of some location parameters, including both μ and some components of β. Since the MAP estimator performed so poorly at estimating the random effect means, we will not consider it any further.

5.3.5 POSTERIOR COVARIANCES

We now assess the accuracy of our estimates of $\mathrm{Cov}_{p_0}(g(\theta))$. The results for the marginal standard deviations are shown in Table 3 and Fig. 15. We refer to the standard deviations of $\mathrm{Cov}_{q_0}(g(\theta))$ as the "uncorrected MFVB" estimate, and of $\mathrm{Cov}_{q_0}^{LR}(g(\theta))$ as the "LRVB" estimate. The uncorrected MFVB variance estimates of β are particularly inaccurate, but the LRVB variances match the exact posterior closely.

In Fig. 16, we compare the off-diagonal elements of $\mathrm{Cov}_{q_0}^{LR}(g(\theta))$ and $\mathrm{Cov}_{p_0}(g(\theta))$. These covariances are zero, by definition, in the uncorrected MFVB estimates $\mathrm{Cov}_{q_0}(g(\theta))$. The left panel of Fig. 16 shows the

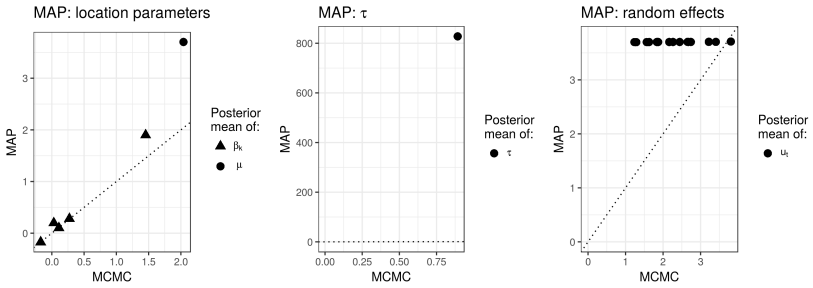


Figure 14: Comparison of MCMC and Laplace means

Parameter	MCMC	LRVB	Uncorrected MFVB
β_1	0.118	0.103	0.005
β_2	0.018	0.018	0.004
β_3	0.020	0.020	0.004
β_4	0.012	0.012	0.004
β_5	0.029	0.030	0.004
μ	0.223	0.192	0.016
τ	0.018	0.033	0.016
u_{1431}	0.663	0.649	0.605
u_{4150}	0.723	0.707	0.662
u_{4575}	0.662	0.649	0.615
u_{4685}	0.610	0.607	0.579

Table 3: Standard deviation results

estimated covariances between the global parameters and all other parameters, including the random effects, and the right panel shows only the covariances amongst the random effects. The LRVB covariances are quite accurate, particularly when we recall that the MCMC draws of μ may be inaccurate due to poor mixing.

5.3.6 PARAMETRIC SENSITIVITY RESULTS

Finally, we compare the MFVB prior sensitivity measures of Section 4.4 to the covariance-based MCMC sensitivity measures of Section 2.1. Since sensitivity is of practical interest only when it is of comparable order to the posterior uncertainty, we report sensitivities normalized by the appropriate standard deviation. That is, we report $\hat{\mathbf{S}}_{\alpha_0} / \sqrt{\text{diag}\left(\hat{\text{Cov}}_{p_0}\left(g\left(\theta\right)\right)\right)}$, and $\mathbf{S}^q_{\alpha_0} / \sqrt{\text{diag}\left(\text{Cov}^{LR}_{q_0}\left(g\left(\theta\right)\right)\right)}$, etc., where $\text{diag}\left(\cdot\right)$ denotes the diagonal vector of a matrix, and the division is element-wise. Note that we use the sensitivity-based variance estimates $\text{Cov}^{LR}_{q_0}$, not the uncorrected MFVB estimates Cov_{q_0}, to normalize the variational sensitivities. We refer to a sensitivity divided by a standard deviation as a "normalized" sensitivity.

The comparison between the MCMC and MFVB sensitivity measures is shown in Fig. 17. The MFVB and MCMC sensitivities correspond very closely, though the MFVB means appear to be slightly more sensitive to the prior parameters than the MCMC means. This close correspondence should not be surprising. As shown in Section 5.3.3, the MFVB and MCMC posterior means match quite closely. If we assume, reasonably, that they continue to match to first order in a neighborhood of our original prior parameters, then Condition 1 will hold and we would expect $\hat{\mathbf{S}}_{\alpha_0} \approx \mathbf{S}^q_{\alpha_0}$.

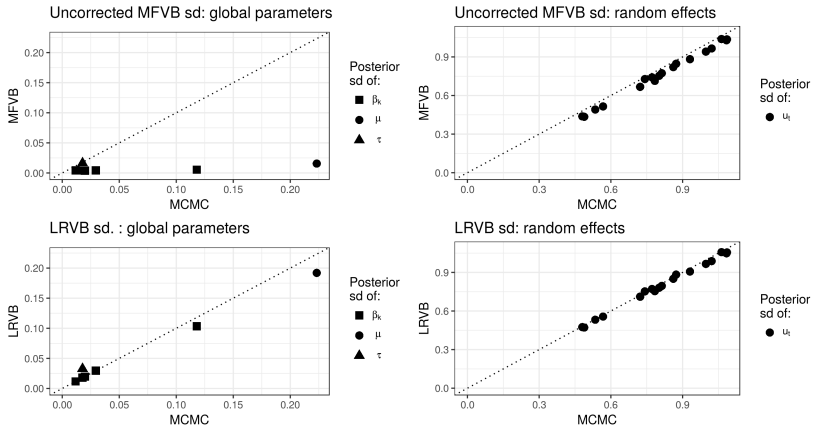


Figure 15: Comparison of MCMC, MFVB, and LRVB standard deviations

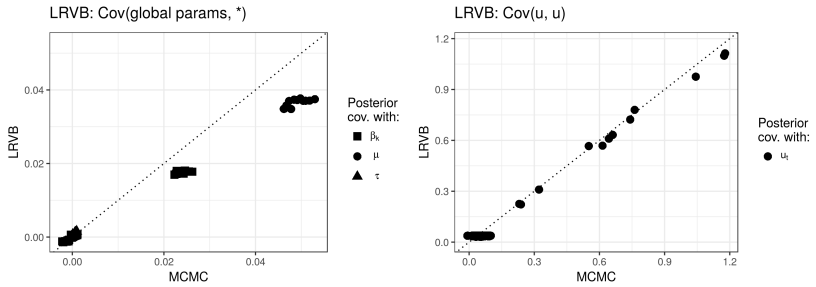


Figure 16: Comparison of MCMC and LRVB off-diagonal covariances

Table 4 shows the detailed MFVB normalized sensitivity results. Each entry is the sensitivity of the MFVB mean of the row's parameter to the column's prior parameter. One can see that several parameters are quite sensitive to the information parameter prior τ_μ. In particular, $\mathbb{E}_{p_\alpha}\left[\mu\right]$ and $\mathbb{E}_{p_\alpha}\left[\beta_1\right]$ are expected to change approximately -0.39 and -0.35 standard deviations, respectively, for every unit change in τ_μ. This size of change could be practically significant (assuming that such a change in τ_μ is subjectively plausible). To investigate this sensitivity further, we re-fit the MFVB model at a range of values of the prior parameter τ_μ, assessing the accuracy of the linear approximation to the sensitivity. The results are shown in Fig. 18. Even for very large changes in τ_μ—resulting in changes to $\mathbb{E}_{p_\alpha}\left[\mu\right]$ and $\mathbb{E}_{p_\alpha}\left[\beta_1\right]$ far in excess of two standard deviations—the linear approximation holds up reasonably well. Fig. 18 also shows a (randomly selected) random effect to be quite sensitive, though not to a practically important degree relative to its posterior standard deviation. The insensitivity of $\mathbb{E}_{p_\alpha}\left[\beta_2\right]$ is also confirmed. Of course, the accuracy of the linear approximation cannot be guaranteed to hold as well in general as it does in this particular case, and the quick

	β_0	τ_β	γ_β	μ_0	τ_μ	α_τ	$\beta\tau$
μ	0.0094	-0.1333	-0.0510	0.0019	-0.3920	0.0058	-0.0048
τ	0.0009	-0.0086	-0.0142	0.0003	-0.0575	0.0398	-0.0328
β_1	0.0089	-0.1464	-0.0095	0.0017	-0.3503	0.0022	-0.0018
β_2	0.0012	-0.0143	-0.0113	0.0003	-0.0516	0.0062	-0.0051
β_3	-0.0035	0.0627	-0.0081	-0.0006	0.1218	-0.0003	0.0002
β_4	0.0018	-0.0037	-0.0540	0.0004	-0.0835	0.0002	-0.0002
β_5	0.0002	0.0308	-0.0695	0.0002	-0.0383	0.0011	-0.0009
u_{1431}	0.0028	-0.0397	-0.0159	0.0006	-0.1169	0.0018	-0.0015
u_{4150}	0.0026	-0.0368	-0.0146	0.0005	-0.1083	0.0022	-0.0018
u_{4575}	0.0028	-0.0406	-0.0138	0.0006	-0.1153	0.0011	-0.0009
u_{4685}	0.0028	-0.0409	-0.0142	0.0006	-0.1163	0.0003	-0.0002

Table 4: MFVB normalized prior sensitivity results

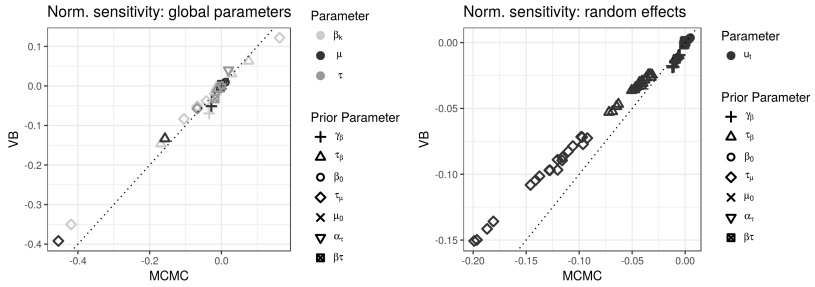


Figure 17: Comparison of MCMC and MFVB normalized parametric sensitivity results

and reliable evaluation of the linearity assumption without re-fitting the model remains interesting future work.

Since we started the MFVB optimization close to the new, perturbed optimum, each new MFVB fit took only 27.2 seconds on average. Re-estimating the MCMC posterior so many times would have been extremely time-consuming. (Note that importance sampling would be useless for prior parameter changes that moved the posterior so far from the original draws.) The considerable sensitivity of this model to a particular prior parameter, which is perhaps surprising on such a large data set, illustrates the value of having fast, general tools for discovering and evaluating prior sensitivity. Our framework provides just such a set of tools.

6. Conclusion

By calculating the sensitivity of MFVB posterior means to model perturbations, we are able to provide two important practical tools for MFVB posterior approximations: improved variance estimates and measures of prior robustness. When MFVB models are implemented in software that supports automatic differentiation, our methods are fast, scalable, and require little additional coding beyond the MFVB objective itself. In our experiments, we were able to calculate accurate posterior means, covariances, and prior sensitivity measures orders of magnitude more quickly than MCMC.

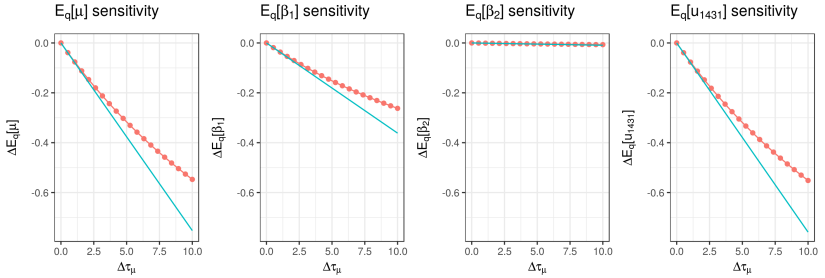


Figure 18: MFVB sensitivity as measured both by linear approximation (blue) and re-fitting (red)

Acknowledgments

We are grateful to the anonymous reviewers for their insightful comments and suggestions. Ryan Giordano's research was funded in part by the National Energy Research Scientific Computing Center, a DOE Office of Science User Facility supported by the Office of Science of the U.S. Department of Energy under Contract number DE-AC02- 05CH11231, and in part by the Gordon and Betty Moore Foundation through Grant GBMF3834 and by the Alfred P. Sloan Foundation through Grant 2013-10-27 to the University of California, Berkeley. Tamara Broderick's research was supported in part by an NSF CAREER Award, an ARO YIP Award, and a Google Faculty Research Award. This work was also supported by the DARPA program on Lifelong Learning Machines, the Office of Naval Research under contract/grant number N00014-17-1-2072, and the Army Research Office under grant number W911NF-17-1-0304.

Appendices

Appendix A. Proof of Theorem 1

In this section we prove Theorem 1.

Proof Under Assumption 1, we can exchange differentiation and integration in $\frac{\partial}{\partial \alpha^\intercal}\int p_0(\theta)\exp(\rho(\theta,\alpha))\,g(\theta)\,\lambda(d\theta)$ and $\frac{\partial}{\partial \alpha^\intercal}\int p_0(\theta)\exp(\rho(\theta,\alpha))\,\lambda(d\theta)$ by Fleming (1965, Chapter 5-11, Theorem 18), which ultimately depends on the Lebesgue dominated convergence theorem. By Assumption 1, $\mathbb{E}_{p_\alpha}[g(\theta)]$ is well-defined for $\alpha \in \mathcal{A}_0$ and

$$\frac{\partial p_0(\theta)\exp(\rho(\theta,\alpha))}{\partial \alpha} = p_0(\theta)\exp(\rho(\theta,\alpha))\frac{\partial \rho(\theta,\alpha)}{\partial \alpha} \quad \lambda\text{-almost everywhere.}$$

Armed with these facts, we can directly compute

$$\begin{aligned}
\left.\frac{d\mathbb{E}_{p_\alpha}[g(\theta)]}{d\alpha^\intercal}\right|_{\alpha_0} &= \left.\frac{d}{d\alpha^\intercal}\frac{\int g(\theta)\,p_0(\theta)\exp(\rho(\theta,\alpha))\,\lambda(d\theta)}{\int p_0(\theta)\exp(\rho(\theta,\alpha))\,\lambda(d\theta)}\right|_{\alpha_0} \\
&= \frac{\left.\frac{\partial}{\partial \alpha^\intercal}\int g(\theta)\,p_0(\theta)\exp(\rho(\theta,\alpha))\,\lambda(d\theta)\right|_{\alpha_0}}{\int p_0(\theta)\exp(\rho(\theta,\alpha_0))\,\lambda(d\theta)} - \mathbb{E}_{p_0}[g(\theta)]\frac{\left.\frac{\partial}{\partial \alpha^\intercal}\int p_0(\theta)\exp(\rho(\theta,\alpha))\,\lambda(d\theta)\right|_{\alpha_0}}{\int p_0(\theta)\exp(\rho(\theta,\alpha_0))\,\lambda(d\theta)} \\
&= \frac{\int g(\theta)\,p_0(\theta)\exp(\rho(\theta,\alpha))\left.\frac{\partial \rho(\theta,\alpha)}{\partial \alpha}\right|_{\alpha_0}\lambda(d\theta)}{\int p_0(\theta)\exp(\rho(\theta,\alpha_0))\,\lambda(d\theta)} - \mathbb{E}_{p_0}[g(\theta)]\,\mathbb{E}_{p_0}\left[\left.\frac{\partial \rho(\theta,\alpha)}{\partial \alpha}\right|_{\alpha_0}\right] \\
&= \mathrm{Cov}_{p_0}\left(g(\theta), \left.\frac{\partial \rho(\theta,\alpha)}{\partial \alpha}\right|_{\alpha_0}\right).
\end{aligned}$$

■

Appendix B. Comparison With MCMC Importance Sampling

In this section, we show that using importance sampling with MCMC samples to calculate the local sensitivity in Eq. (1) is precisely equivalent to using the same MCMC samples to estimate the covariance in Eq. (4) directly. For this section, will suppose that Assumption 1 holds. Further suppose, without loss of generality, we have samples θ_i drawn IID from $p_0(\theta)$:

$$\begin{aligned}
\theta_n &\overset{iid}{\sim} p_0(\theta), \text{ for } n = 1, ..., N_s \\
\mathbb{E}_{p_0}[g(\theta)] &\approx \frac{1}{N_s}\sum_{n=1}^{N_s} g(\theta_n).
\end{aligned}$$

Typically we cannot compute the dependence of the normalizing constant $\int p(\theta')\exp(\rho(\theta',\alpha))\,\lambda(d\theta')$ on α, so we use the following importance sampling estimate for $\mathbb{E}_{p_\alpha}[g(\theta)]$ (Owen, 2013, Chapter 9):

$$\begin{aligned}
w_n &= \exp(\rho(\theta_n,\alpha) - \rho(\theta_n,\alpha_0)) \\
\tilde{w}_n &:= \frac{w_n}{\sum_{n'=1}^{N_s} w_{n'}} \\
\mathbb{E}_{p_\alpha}[g(\theta)] &\approx \sum_{n=1}^{N_s}\tilde{w}_n g(\theta_n).
\end{aligned}$$

Note that $\tilde{w}_n|_{\alpha_0} = \frac{1}{N_s}$, so the importance sampling estimate recovers the ordinary sample mean at α_0. The derivatives of the weights are given by

$$\begin{aligned}
\frac{\partial w_n}{\partial \alpha} &= w_n\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha} \\
\frac{\partial \tilde{w}_n}{\partial \alpha} &= \frac{\frac{\partial w_n}{\partial \alpha}}{\sum_{n'=1}^{N_s} w_{n'}} - \frac{w_n\sum_{n'=1}^{N_s}\frac{\partial w_{n'}}{\partial \alpha}}{\left(\sum_{n'=1}^{N_s} w_{n'}\right)^2} \\
&= \frac{w_n}{\sum_{n'=1}^{N_s} w_{n'}}\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha} - \frac{w_n}{\sum_{n'=1}^{N_s} w_{n'}}\sum_{n'=1}^{N_s}\frac{w_n}{\sum_{n'=1}^{N_s} w_{n'}}\frac{\partial \rho(\theta_{n'},\alpha)}{\partial \alpha} \\
&= \tilde{w}_n\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha} - \tilde{w}_n\sum_{n'=1}^{N_s}\tilde{w}_{n'}\frac{\partial \rho(\theta_{n'},\alpha)}{\partial \alpha}.
\end{aligned}$$

It follows that

$$\begin{aligned}
\left.\frac{\partial}{\partial \alpha}\sum_{n=1}^{N_s}\tilde{w}_n g(\theta_n)\right|_{\alpha_0} &= \sum_{n=1}^{N_s}\left(\tilde{w}_n\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha} - \tilde{w}_n\sum_{n'=1}^{N_s}\tilde{w}_{n'}\frac{\partial \rho(\theta_{n'},\alpha)}{\partial \alpha}\right)\Bigg|_{\alpha_0} g(\theta_n) \\
&= \frac{1}{N_s}\sum_{n=1}^{N_s}\left.\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha}\right|_{\alpha_0} g(\theta_n) - \left[\frac{1}{N_s}\sum_{n=1}^{N_s}\left.\frac{\partial \rho(\theta_n,\alpha)}{\partial \alpha}\right|_{\alpha_0}\right]\left[\frac{1}{N_s}\sum_{n=1}^{N_s} g(\theta_n)\right],
\end{aligned}$$

which is precisely the sample version of the covariance in Theorem 1.

Appendix C. Our Use of the Terms "Sensitivity" and "Robustness"

In this section we clarify our usage of the terms "robustness" and "sensitivity." The quantity $\mathbf{S}_{\alpha_0}^\intercal(\alpha - \alpha_0)$ measures the *sensitivity* of $\mathbb{E}_{p_\alpha}[g(\theta)]$ to perturbations in the direction $\Delta\alpha$. Intuitively, as sensitivity increases, robustness decreases, and, in this sense, sensitivity and robustness are opposites of one another. However, we emphasize that sensitivity is a clearly defined, measurable quantity and that robustness is a subjective judgment informed by sensitivity, but also by many other less objective considerations.

Suppose we have calculated $\mathbf{S}_{\alpha_0}$ from Eq. (1) and found that it has a particular value. To determine whether our model is robust, we must additionally decide

1. How large of a change in the prior, $||\alpha - \alpha_0||$, is plausible, and
2. How large of a change in $\mathbb{E}_{p_\alpha}[g(\theta)]$ is important.

The set of plausible prior values necessarily remains a subjective decision.[4] Whether or not a particular change in $\mathbb{E}_{p_\alpha}[g(\theta)]$ is important depends on the ultimate use of the posterior mean. For example, the posterior standard deviation can be a guide: if the prior sensitivity is swamped by the posterior uncertainty then it can be neglected when reporting our subjective uncertainty about $g(\theta)$, and the model is robust. Similarly, even if the prior sensitivity is much larger than the posterior standard deviation but small enough that it would not affect any actionable decision made on the basis of the value of $\mathbb{E}_{p_\alpha}[g(\theta)]$, then the model is robust. Intermediate values remain a matter of judgment. An illustration of the relationship between sensitivity and robustness is shown in Fig. 19.

Finally, we note that if $\mathcal{A}$ is small enough that $\mathbb{E}_{p_\alpha}[g(\theta)]$ is roughly linear in α for $\alpha \in \mathcal{A}$, then calculating Eq. (1) for all $\alpha \in \mathcal{A}$ and finding the worst case can be thought of as a first-order approximation to a global robustness estimate. Depending on the problem at hand, this linearity assumption may not be plausible except

4. This decision can be cast in a formal decision theoretic framework based on a partial ordering of subjective beliefs (Insua and Criado, 2000).

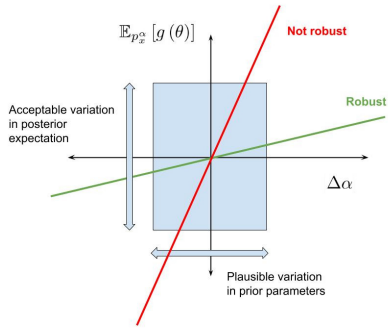


Figure 19: The relationship between robustness and sensitivity

for very small $\mathcal{A}$. This weakness is inherent to the local robustness approach. Nevertheless, even when the perturbations are valid only for a small $\mathcal{A}$, these easily-calculable measures may still provide valuable intuition about the potential modes of failure for a model.

If $g(\theta)$ is a scalar, it is natural to attempt to summarize the high-dimensional vector $\mathbf{S}_{\alpha_0}$ in a single easily reported number such as

$$\mathbf{S}_{\alpha_0}^{sup} := \sup_{\alpha:\|\alpha-\alpha_0\|\leq 1} \left|\mathbf{S}_{\alpha_0}^{\intercal}(\alpha-\alpha_0)\right|.$$

For example, the calculation of $\mathbf{S}_{\alpha_0}^{sup}$ is the principal ambition of Basu et al. (1996). The use of such summaries is also particularly common in work that considers function-valued perturbations (e.g., Gustafson, 1996b; Roos et al., 2015). (Function-valued perturbations can be connected to the finite-dimensional perturbations of the present work through the notion of the Gateaux derivative (Huber, 2011, Chapter 2.5), the elaboration of which we leave to future work.) Although the summary $\mathbf{S}_{\alpha_0}^{sup}$ has obvious merits, in the present work we emphasize the calculation only of $\mathbf{S}_{\alpha_0}$ in the belief that its interpretation is likely to vary from application to application and require some critical thought and subjective judgment. For example, the unit ball $\|\alpha-\alpha_0\|\leq 1$ (as in Basu et al. (1996)) may not make sense as a subjective description of the range of plausible variability of $p(\theta|\alpha)$. Consider, e.g.: why should the off-diagonal term of a Wishart prior plausibly vary as widely as the mean of some other parameter, when the two might not even have the same units? This problem is easily remedied by choosing an appropriate scaling of the parameters and thereby making the unit ball an appropriate range for the problem at hand, but the right scaling will vary from problem to problem and necessarily be a somewhat subjective choice, so we refrain from taking a stand on this decision. As another example, the worst-case function-valued perturbations of Gustafson (1996a,b) require a choice of a metric ball in function space whose meaning may not be intuitively obvious, may provide worst-case perturbations that depend on the data to a subjectively implausible degree, and may exhibit interesting but perhaps counter-intuitive asymptotic behavior for different norms and perturbation dimensions. Consequently, we do not attempt to prescribe a particular one-size-fits-all summary measure. The local sensitivity $\mathbf{S}_{\alpha_0}$ is a well-defined mathematical quantity. Its relationship to robustness must remain a matter of judgment.

Appendix D. Proof of Theorem 2

In this section we prove Theorem 2.

Proof For notational convenience, we will define

$$KL(\eta,\alpha) := KL(q(\theta;\eta)||p_\alpha(\theta)).$$

By Assumption 3, $\eta^*(\alpha)$ is both optimal and interior for all $\alpha\in\mathcal{A}_0$, and by Assumption 2, $KL(\eta,\alpha)$ is continuously differentiable in η. Therefore, the first-order conditions of the optimization problem in Eq. (8) give:

$$\left.\frac{\partial KL(\eta,\alpha)}{\partial\eta}\right|_{\eta=\eta^*(\alpha)} = 0 \text{ for all } \alpha\in\mathcal{A}_0. \tag{26}$$

$\left.\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\eta^\intercal}\right|_{\alpha_0}$ is positive definite by the strict optimality of η^* in Assumption 3, and $\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\alpha^\intercal}$ is continuous by Assumption 2. It follows that $\eta^*(\alpha)$ is a continuously differentiable function of α by application of the implicit function theorem to the first-order condition in Eq. (26) (Fleming, 1965, Chapter 4.6). So we can use the chain rule to take the total derivative of Eq. (26) with respect to α.

$$\begin{aligned}\frac{d}{d\alpha}\left(\left.\frac{\partial KL(\eta,\alpha)}{\partial\eta}\right|_{\eta=\eta^*(\alpha)}\right) &= 0 \text{ for all } \alpha\in\mathcal{A}_0 \Rightarrow\\ \left.\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\eta^\intercal}\right|_{\eta=\eta^*(\alpha)}\frac{d\eta^*(\alpha)}{d\alpha^\intercal} + \left.\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\alpha^\intercal}\right|_{\eta=\eta^*(\alpha)} &= 0 \text{ for all } \alpha\in\mathcal{A}_0.\end{aligned}$$

The strict optimality of $KL(\eta,\alpha)$ at $\eta^*(\alpha)$ in Assumption 3 requires that $\left.\frac{\partial^2 KL(\eta,t)}{\partial\eta\partial\eta^T}\right|_{\eta=\eta^*(\alpha)}$ be invertible. So we can evaluate at $\alpha=\alpha_0$ and solve to find that

$$\left.\frac{d\eta^*(\alpha)}{d\alpha^\intercal}\right|_{\alpha_0} = -\left(\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\eta^\intercal}\right)^{-1}\left.\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\alpha^\intercal}\right|_{\eta=\eta_0^*,\alpha=\alpha_0}.$$

$\mathbb{E}_{q_\alpha}[g(\theta)]$ is a continuously differentiable function of $\eta^*(\alpha)$ by Assumption 4. So by the chain rule and Assumption 2, we have that

$$\left.\frac{d\mathbb{E}_{q(\theta;\eta)}[g(\theta)]}{d\alpha^\intercal}\right|_{\alpha_0} = \left.\frac{\partial\mathbb{E}_{q(\theta;\eta)}[g(\theta)]}{\partial\eta}\frac{d\eta^*(\alpha)}{d\alpha^\intercal}\right|_{\eta=\eta_0^*,\alpha=\alpha_0}.$$

Finally, we observe that

$$\begin{aligned}KL(\eta,\alpha) &= \mathbb{E}_{q(\theta;\eta)}[\log q(\theta;\eta) - \log p(\theta) - \rho(\theta,\alpha)] + Constant \Rightarrow\\ \left.\frac{\partial^2 KL(\eta,\alpha)}{\partial\eta\partial\alpha^\intercal}\right|_{\eta=\eta_0^*,\alpha=\alpha_0} &= -\left.\frac{\partial^2\mathbb{E}_{q(\theta;\eta)}[\rho(\theta,\alpha)]}{\partial\eta\partial\alpha^\intercal}\right|_{\eta=\eta_0^*,\alpha=\alpha_0}.\end{aligned}$$

Here, the term $Constant$ contains quantities that do not depend on η. Plugging in gives the desired result. ■

Appendix E. Exactness of Multivariate Normal Posterior Means

In this section, we show that the MFVB estimate of the posterior means of a multivariate normal with known covariance is exact and that, as an immediate consequence, the linear response covariance recovers the exact posterior covariance, i.e., $\mathrm{Cov}_{q_0}^{LR}(\theta) = \mathrm{Cov}_{p_0}(\theta)$.

Suppose we are using MFVB to approximate a non-degenerate multivariate normal posterior, i.e.,

$$p_0(\theta) = \mathcal{N}(\theta;\mu,\mathbf{\Sigma})$$

for full-rank Σ. This posterior arises, for instance, given a multivariate normal likelihood $p(x|\mu) = \prod_{n=1:N} \mathcal{N}(x_n|\theta, \Sigma_x)$ with known covariance Σ_x and a conjugate multivariate normal prior on the unknown mean parameter $\theta \in \mathbb{R}^K$. Additionally, even when the likelihood is non-normal or the prior is not conjugate, the posterior may be closely approximated by a multivariate normal distribution when a Bayesian central limit theorem can be applied (Le Cam and Yang, 2012, Chapter 8).

We will consider an MFVB approximation to $p_0(\theta)$. Specifically, let the elements of the vector θ be given by scalars θ_k for $k = 1, ..., K$, and take the MFVB normal approximation with means m_k and variances v_k:

$$\mathcal{Q} = \left\{ q(\theta) : q(\theta) = \prod_{k=1}^{K} \mathcal{N}(\theta_k; m_k, v_k) \right\}.$$

In the notation of Eq. (9), we have $\eta_k = (m_k, v_k)^\intercal$ with $\Omega_\eta = \{\eta : v_k > 0, \forall k = 1, ..., K\}$. The optimal variational parameters are given by $\eta_k^* = (m_k^*, v_k^*)^\intercal$.

Lemma 1 *Let $p_0(\theta) = \mathcal{N}(\theta; \mu, \Sigma)$ for full-rank Σ and let $\mathcal{Q} = \left\{ q(\theta) : q(\theta) = \prod_{k=1}^{K} \mathcal{N}(\theta_k; m_k, v_k) \right\}$ be the mean field approximating family. Then there exists an $\eta^* = (m^*, v^*)$ that solves*

$$\eta^* = \underset{\eta : q(\theta;\eta) \in \mathcal{Q}}{\operatorname{argmin}} KL(q(\theta;\eta) \,||p_\alpha(\theta))$$

with $m^* = \mu$.

Proof Let $\operatorname{diag}(v)$ denote the $K \times K$ matrix with the vector v on the diagonal and zero elsewhere. Using the fact that the entropy of a univariate normal distribution with variance v is $\frac{1}{2}\log v$ plus a constant, the variational objective in Eq. (8) is given by

$$\begin{aligned}
KL(q(\theta;\eta) \,||p_\alpha(\theta)) &= \mathbb{E}_{q(\theta;\eta)}\left[\frac{1}{2}(\theta-\mu)^\intercal \Sigma^{-1}(\theta-\mu)\right] - \frac{1}{2}\sum_k \log v_k + Constant \\
&= \frac{1}{2}\operatorname{trace}\left(\Sigma^{-1}\mathbb{E}_{q(\theta;\eta)}\left[\theta\theta^\intercal\right]\right) - \mu^\intercal\Sigma^{-1}\mathbb{E}_{q(\theta;\eta)}[\theta] - \frac{1}{2}\sum_k \log v_k + Constant \\
&= \frac{1}{2}\operatorname{trace}\left(\Sigma^{-1}\left(mm^\intercal + \operatorname{diag}(v)\right)\right) - \mu^\intercal\Sigma^{-1}m - \frac{1}{2}\sum_k \log v_k + Constant \\
&= \frac{1}{2}\operatorname{trace}\left(\Sigma^{-1}\operatorname{diag}(v)\right) + \frac{1}{2}m^\intercal\Sigma^{-1}m - \mu^\intercal\Sigma^{-1}m - \frac{1}{2}\sum_k \log v_k + Constant.
\end{aligned} \tag{27}$$

The first-order condition for the optimal m^* is then

$$\begin{aligned}
\left.\frac{\partial KL(q(\theta;\eta) \,||p_0(\theta))}{\partial m}\right|_{m=m^*, v=v^*} = 0 \Rightarrow \\
\Sigma^{-1}m^* - \Sigma^{-1}\mu = 0 \Rightarrow \\
m^* = \mu.
\end{aligned}$$

The optimal variances follow similarly:

$$\begin{aligned}
\left.\frac{\partial KL(q(\theta;\eta) \,||p_0(\theta))}{\partial v_k}\right|_{m=m^*, v=v^*} = 0 \Rightarrow \\
\frac{1}{2}\left(\Sigma^{-1}\right)_{kk} - \frac{1}{2}\frac{1}{v_k^*} = 0 \Rightarrow \\
v_k^* = \frac{1}{\left(\Sigma^{-1}\right)_{kk}}.
\end{aligned}$$

Since $v_k^* > 0$, we have $\eta^* \in \Omega_\eta$.

Lemma 1 can be also be derived via the variational coordinate ascent updates (Bishop (2006, Section 10.1.2) and Giordano et al. (2015, Appendix B)). ■

Next, we show that Lemma 1 holds for all perturbations of the form $\rho(\theta, \alpha) = \alpha^\intercal\theta$ with $\alpha_0 = 0$ and that Assumptions 1–4 are satisfied for all finite α.

Lemma 2 *Under the conditions of Lemma 1, let $p_\alpha(\theta)$ be defined from Eq. (2) with $\rho(\theta, \alpha) = \alpha^\intercal\theta$ and $\alpha_0 = 0$. Take $g(\theta) = \theta$. Then, for all finite α, Assumptions 1–4 are satisfied, and Condition 1 is satisfied with equality.*

Proof Up to a constant that does not depend on θ, the log density of $p_\alpha(\theta)$ is

$$\begin{aligned}
\log p_\alpha(\theta) &= -\frac{1}{2}(\theta-\mu)^\intercal\Sigma^{-1}(\theta-\mu) + \alpha^\intercal\theta + Constant \\
&= -\frac{1}{2}\theta^\intercal\Sigma^{-1}\theta - \frac{1}{2}\mu^\intercal\Sigma^{-1}\mu + \left(\mu^\intercal\Sigma^{-1} + \alpha^\intercal\right)\theta + Constant.
\end{aligned}$$

Since θ is a natural sufficient statistic of the multivariate normal distribution and the corresponding natural parameter of $p_\alpha(\theta)$, $\Sigma^{-1}\mu + \alpha$, is interior when Σ is full-rank, $p_\alpha(\theta)$ is multivariate normal for any finite α. Assumption 1 follows immediately.

By inspection of Eq. (27), Assumption 2 is satisfied. Because Ω_η is an open set and Σ is positive definite, Assumption 3 is satisfied. Since $\mathbb{E}_{q(\theta;\eta)}[g(\theta)] = m$, Assumption 4 is satisfied. Finally, by Lemma 1, $\mathbb{E}_{q_\alpha}[\theta] = \mathbb{E}_{p_\alpha}[\theta]$, so Condition 1 is satisfied with equality. ■

It now follows immediately from Definition 6 that the linear response variational covariance exactly reproduces the exact posterior covariance for the multivariate normal distribution.

Corollary 4 *Under the conditions of Lemma 2,* $\operatorname{Cov}_{q_0}^{LR}(\theta) = \operatorname{Cov}_{p_0}(\theta)$.

Appendix F. ADVI Model Details

This section reports the Stan code for the models used in Section 5.2. For details on how to interpret the models as well as the unconstraining transforms, see the Stan manual (Stan Team, 2015). For the associated data, see the Stan example models wiki (Stan Team, 2017).

F.1 Election Model (`election88.stan`)

Listing 1: `election88.stan`

```
data {
  int<lower=0> N;
  int<lower=0> n_state;
  vector<lower=0,upper=1>[N] black;
  vector<lower=0,upper=1>[N] female;
  int<lower=1,upper=n_state> state[N];
  int<lower=0,upper=1> y[N];
}
parameters {
  vector[n_state] a;
  vector[2] b;
  real<lower=0,upper=100> sigma_a;
  real mu_a;
}
```

```
transformed parameters {
  vector[N] y_hat;

  for (i in 1:N)
    y_hat[i] <- b[1] * black[i] + b[2] * female[i] + a[state[i]];
}
model {
  mu_a ~ normal(0, 1);
  a ~ normal (mu_a, sigma_a);
  b ~ normal (0, 100);
  y ~ bernoulli_logit(y_hat);
}
```

F.2 Sesame Street Model (`sesame_street1`)

Listing 2: `sesame_street1.stan`

```
data {
  int<lower=0> J;
  int<lower=0> N;
  int<lower=1,upper=J> siteset[N];
  vector[2] yt[N];
  vector[N] z;
}
parameters {
  vector[2] ag[J];
  real b;
  real d;
  real<lower=-1,upper=1> rho_ag;
  real<lower=-1,upper=1> rho_yt;
  vector[2] mu_ag;
  real<lower=0,upper=100> sigma_a;
  real<lower=0,upper=100> sigma_g;
  real<lower=0,upper=100> sigma_t;
  real<lower=0,upper=100> sigma_y;
}
model {
  vector[J] a;
  vector[J] g;
  matrix[2,2] Sigma_ag;
  matrix[2,2] Sigma_yt;
  vector[2] yt_hat[N];

  //data level
  Sigma_yt[1,1] <- pow(sigma_y,2);
  Sigma_yt[2,2] <- pow(sigma_t,2);
  Sigma_yt[1,2] <- rho_yt*sigma_y*sigma_t;
  Sigma_yt[2,1] <- Sigma_yt[1,2];

  // group level
  Sigma_ag[1,1] <- pow(sigma_a,2);
  Sigma_ag[2,2] <- pow(sigma_g,2);
```



```
  Sigma_ag[1,2] <- rho_ag*sigma_a*sigma_g;
  Sigma_ag[2,1] <- Sigma_ag[1,2];

  for (j in 1:J) {
    a[j] <- ag[j,1];
    g[j] <- ag[j,2];
  }

  for (i in 1:N) {
    yt_hat[i,2] <- g[siteset[i]] + d * z[i];
    yt_hat[i,1] <- a[siteset[i]] + b * yt[i,2];
  }

  //data level
  sigma_y ~ uniform (0, 100);
  sigma_t ~ uniform (0, 100);
  rho_yt ~ uniform(-1, 1);
  d ~ normal (0, 31.6);
  b ~ normal (0, 31.6);

  //group level
  sigma_a ~ uniform (0, 100);
  sigma_g ~ uniform (0, 100);
  rho_ag ~ uniform(-1, 1);
  mu_ag ~ normal (0, 31.6);

  for (j in 1:J)
    ag[j] ~ multi_normal(mu_ag,Sigma_ag);

  //data model
  for (i in 1:N)
    yt[i] ~ multi_normal(yt_hat[i],Sigma_yt);

}
```

F.3 Radon Model (`radon_vary_intercept_floor`)

Listing 3: `radon_vary_intercept_floor.stan`

```
data {
  int<lower=0> J;
  int<lower=0> N;
  int<lower=1,upper=J> county[N];
  vector[N] u;
  vector[N] x;
  vector[N] y;
}
parameters {
  vector[J] a;
  vector[2] b;
  real mu_a;
  real<lower=0,upper=100> sigma_a;
```

```
  real<lower=0,upper=100> sigma_y;
}
transformed parameters {
  vector[N] y_hat;

  for (i in 1:N)
    y_hat[i] <- a[county[i]] + u[i] * b[1] + x[i] * b[2];
}
model {
  mu_a ~ normal(0, 1);
  a ~ normal(mu_a, sigma_a);
  b ~ normal(0, 1);
  y ~ normal(y_hat, sigma_y);
}
```

F.4 Ecology Model (`cjs_cov_randeff`)

Listing 4: `cjs_cov_randeff.stan`

```
// This models is derived from section 12.3 of "Stan Modeling Language
// User's Guide and Reference Manual"

functions {
  int first_capture(int[] y_i) {
    for (k in 1:size(y_i))
      if (y_i[k])
        return k;
    return 0;
  }

  int last_capture(int[] y_i) {
    for (k_rev in 0:(size(y_i) - 1)) {
      // Compoud declaration was enabled in Stan 2.13
      int k = size(y_i) - k_rev;
      //      int k;
      //      k = size(y_i) - k_rev;
      if (y_i[k])
        return k;
    }
    return 0;
  }

  matrix prob_uncaptured(int nind, int n_occasions,
                         matrix p, matrix phi) {
    matrix[nind, n_occasions] chi;

    for (i in 1:nind) {
      chi[i, n_occasions] = 1.0;
      for (t in 1:(n_occasions - 1)) {
        // Compoud declaration was enabled in Stan 2.13
        int t_curr = n_occasions - t;
        int t_next = t_curr + 1;
```



```
        /*
        int t_curr;
        int t_next;

        t_curr = n_occasions - t;
        t_next = t_curr + 1;
        */
        chi[i, t_curr] = (1 - phi[i, t_curr])
                         + phi[i, t_curr] * (1 - p[i, t_next - 1]) * chi[
    i, t_next];
      }
    }
    return chi;
  }
}

data {
  int<lower=0> nind;            // Number of individuals
  int<lower=2> n_occasions;     // Number of capture occasions
  int<lower=0,upper=1> y[nind, n_occasions];    // Capture-history
  vector[n_occasions - 1] x;    // Covariate
}

transformed data {
  int n_occ_minus_1 = n_occasions - 1;
  //  int n_occ_minus_1;
  int<lower=0,upper=n_occasions> first[nind];
  int<lower=0,upper=n_occasions> last[nind];
  vector<lower=0,upper=nind>[n_occasions] n_captured;

  //  n_occ_minus_1 = n_occasions - 1;
  for (i in 1:nind)
    first[i] = first_capture(y[i]);
  for (i in 1:nind)
    last[i] = last_capture(y[i]);
  n_captured = rep_vector(0, n_occasions);
  for (t in 1:n_occasions)
    for (i in 1:nind)
      if (y[i, t])
        n_captured[t] = n_captured[t] + 1;
}

parameters {
  real beta;                          // Slope parameter
  real<lower=0,upper=1> mean_phi;     // Mean survival
  real<lower=0,upper=1> mean_p;       // Mean recapture
  vector[n_occ_minus_1] epsilon;
  real<lower=0,upper=10> sigma;
  // In case a weakly informative prior is used
  //  real<lower=0> sigma;
}
```

```
84
85 transformed parameters {
86   matrix<lower=0,upper=1>[nind, n_occ_minus_1] phi;
87   matrix<lower=0,upper=1>[nind, n_occ_minus_1] p;
88   matrix<lower=0,upper=1>[nind, n_occasions] chi;
89   // Compoud declaration was enabled in Stan 2.13
90   real mu = logit(mean_phi);
91   //  real mu;
92
93   //  mu = logit(mean_phi);
94   // Constraints
95   for (i in 1:nind) {
96     for (t in 1:(first[i] - 1)) {
97       phi[i, t] = 0;
98       p[i, t] = 0;
99     }
100    for (t in first[i]:n_occ_minus_1) {
101      phi[i, t] = inv_logit(mu + beta * x[t] + epsilon[t]);
102      p[i, t] = mean_p;
103    }
104  }
105
106  chi = prob_uncaptured(nind, n_occasions, p, phi);
107 }
108
109 model {
110   // Priors
111   // Uniform priors are implicitly defined.
112   //  mean_phi ~ uniform(0, 1);
113   //  mean_p ~ uniform(0, 1);
114   //  sigma ~ uniform(0, 10);
115   // In case a weakly informative prior is used
116   //  sigma ~ normal(5, 2.5);
117   beta ~ normal(0, 100);
118   epsilon ~ normal(0, sigma);
119
120   for (i in 1:nind) {
121     if (first[i] > 0) {
122       for (t in (first[i] + 1):last[i]) {
123         1 ~ bernoulli(phi[i, t - 1]);
124         y[i, t] ~ bernoulli(p[i, t - 1]);
125       }
126       1 ~ bernoulli(chi[i, last[i]]);
127     }
128   }
129 }
130
131 generated quantities {
132   real<lower=0> sigma2;
133   vector<lower=0,upper=1>[n_occ_minus_1] phi_est;
134
```



```
135   sigma2 = square(sigma);
136   // inv_logit was vectorized in Stan 2.13
137   phi_est = inv_logit(mu + beta * x + epsilon); // Yearly survival
138   /*
139   for (t in 1:n_occ_minus_1)
140     phi_est[t] = inv_logit(mu + beta * x[t] + epsilon[t]);
141   */
142 }
```

Appendix G. LKJ Priors for Covariance Matrices in Mean Field Variational Inference

In this section we briefly derive closed-form expressions for using an LKJ prior with a Wishart variational approximation.

Proposition 3 *Let $\mathbf{\Sigma}$ be a $K \times K$ positive definite covariance matrix. Define the $K \times K$ matrix $\mathbf{M}$ such that*

$$\mathbf{M}_{ij} = \begin{cases} \sqrt{\mathbf{\Sigma}_{ij}} & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

Define the correlation matrix R as

$$\mathbf{R} = \mathbf{M}^{-1}\mathbf{\Sigma}\mathbf{M}^{-1}.$$

Define the LKJ prior on $\mathbf{R}$ with concentration parameter ξ (Lewandowski et al., 2009):

$$p_{\text{LKJ}}\left(\mathbf{R}|\xi\right) \propto |\mathbf{R}|^{\xi-1}.$$

Let $q\left(\mathbf{\Sigma}|\mathbf{V}^{-1}, \nu\right)$ be an inverse Wishart distribution with matrix parameter $\mathbf{V}^{-1}$ and ν degrees of freedom. Then

$$\mathbb{E}_q\left[\log|\mathbf{R}|\right] = \log\left|\mathbf{V}^{-1}\right| - \psi_K\left(\frac{\nu}{2}\right) - \sum_{k=1}^{K}\log\left(\left(\mathbf{V}^{-1}\right)_{kk}\right) + K\psi\left(\frac{\nu - K + 1}{2}\right) + Constant$$

$$\mathbb{E}_q\left[\log p_{\text{LKJ}}\left(\mathbf{R}|\xi\right)\right] = (\xi - 1)\,\mathbb{E}_q\left[\log|\mathbf{R}|\right] + Constant,$$

where $Constant$ does not depend on $\mathbf{V}$ or ν. Here, ψ_K is the multivariate digamma function.

Proof First note that

$$\begin{aligned} \log|\mathbf{\Sigma}| &= 2\log|\mathbf{M}| + \log|\mathbf{R}| \\ &= 2\sum_{k=1}^{K}\log\sqrt{\mathbf{\Sigma}_{kk}} + \log|\mathbf{R}| \\ &= \sum_{k=1}^{K}\log\mathbf{\Sigma}_{kk} + \log|\mathbf{R}| \Rightarrow \\ \log|\mathbf{R}| &= \log|\mathbf{\Sigma}| - \sum_{k=1}^{K}\log\mathbf{\Sigma}_{kk}. \end{aligned} \tag{28}$$

By Eq. B.81 in (Bishop, 2006), a property of the inverse Wishart distribution is the following relation.

$$E_q\left[\log|\mathbf{\Sigma}|\right] = \log\left|\mathbf{V}^{-1}\right| - \psi_K\left(\frac{\nu}{2}\right) - K\log 2, \tag{29}$$

where ψ_K is the multivariate digamma function. By the marginalization property of the inverse Wishart distribution,

$$\begin{aligned}\boldsymbol{\Sigma}_{kk} &\sim \text{InverseWishart}\left(\left(\mathbf{V}^{-1}\right)_{kk}, \nu - K + 1\right) \Rightarrow \\ E_q\left[\log \boldsymbol{\Sigma}_{kk}\right] &= \log\left(\left(\mathbf{V}^{-1}\right)_{kk}\right) - \psi\left(\frac{\nu - K + 1}{2}\right) - \log 2. \end{aligned} \tag{30}$$

Plugging Eq. (29) and Eq. (30) into Eq. (28) gives the desired result. ■

Appendix H. Logistic GLMM Model Details

In this section we include extra details about the model and analysis of Section 5. We will continue to use the notation defined therein. We use $Constant$ to denote any constants that do not depend on the prior parameters, parameters, or data. The log likelihood is

$$\begin{aligned}\log p\left(y_{it}|u_t, \beta\right) &= y_{it}\log\left(\frac{p_{it}}{1-p_{it}}\right) + \log\left(1-p_{it}\right) \\ &= y_{it}\rho + \log\left(1-p_{it}\right) + Constant \\ \log p\left(u|\mu,\tau\right) &= -\frac{1}{2}\tau\sum_{t=1}^{T}\left(u_t-\mu\right)^2 - \frac{1}{2}T\log\tau \\ &= -\frac{1}{2}\tau\sum_{t=1}^{T}\left(u_t^2 - \mu u_t + \mu^2\right) - \frac{1}{2}T\log\tau + Constant \\ \log p\left(\mu,\tau,\beta\right) &= -\frac{1}{2}\sigma_\mu^{-2}\left(\mu^2 + 2\mu\mu_0\right) + \\ &\quad \left(1-\alpha_\tau\right)\tau + \beta_\tau\log\tau + \\ &\quad -\frac{1}{2}\left(\text{trace}\left(\boldsymbol{\Sigma}_\beta^{-1}\beta\beta^T\right) + 2\text{trace}\left(\boldsymbol{\Sigma}_\beta^{-1}\beta_0\beta^T\right)\right). \end{aligned} \tag{31}$$

The prior parameters were taken to be

$$\begin{aligned}\mu_0 &= 0.000 \\ \sigma_\mu^{-2} &= 0.010 \\ \beta_0 &= 0.000 \\ \sigma_\beta^{-2} &= 0.100 \\ \alpha_\tau &= 3.000 \\ \beta_\tau &= 3.000. \end{aligned}$$

Under the variational approximation, ρ_{it} is normally distributed given x_{it}, with

$$\begin{aligned}\rho_{it} &= x_{it}^T\beta + u_t \\ \mathbb{E}_q\left[\rho_{it}\right] &= x_{it}^T\mathbb{E}_q\left[\beta\right] + \mathbb{E}_q\left[u_t\right] \\ \text{Var}_q\left(\rho_{it}\right) &= \mathbb{E}_q\left[\beta^T x_{it}x_{it}^T\beta\right] - \mathbb{E}_q\left[\beta\right]^T x_{it}x_{it}^T\mathbb{E}_q\left[\beta\right] + \text{Var}_q\left(u_t\right) \\ &= \mathbb{E}_q\left[\text{tr}\left(\beta^T x_{it}x_{it}^T\beta\right)\right] - \text{tr}\left(\mathbb{E}_q\left[\beta\right]^T x_{it}x_{it}^T\mathbb{E}_q\left[\beta\right]\right) + \text{Var}_q\left(u_t\right) \\ &= \text{tr}\left(x_{it}x_{it}^T\left(\mathbb{E}_q\left[\beta\beta^T\right] - \mathbb{E}_q\left[\beta\right]\mathbb{E}_q\left[\beta\right]^T\right)\right) + \text{Var}_q\left(u_t\right). \end{aligned}$$

We can thus use $n_{MC} = 4$ points of Gauss-Hermite quadrature to numerically estimate $\mathbb{E}_q\left[\log\left(1 - \frac{e^\rho}{1+e^\rho}\right)\right]$:

$$\begin{aligned}\rho_{it,s} &:= \sqrt{\text{Var}_q\left(\rho_{it}\right)}z_s + \mathbb{E}_q\left[\rho_{it}\right] \\ \mathbb{E}_q\left[\log\left(1 - \frac{e^{\rho_{it}}}{1+e^{\rho_{it}}}\right)\right] &\approx \frac{1}{n_{MC}}\sum_{s=1}^{n_{MC}}\log\left(1 - \frac{e^{\rho_{it,s}}}{1+e^{\rho_{it,s}}}\right) \end{aligned}$$

We found that increasing the number of points used for the quadrature did not measurably change any of the results. The integration points and weights were calculated using the `numpy.polynomial.hermite` module in Python (Jones et al., 2001).

References

A. Agresti and M. Kateri. *Categorical Data Analysis*. Springer, 2011.

S. Basu, S. Rao Jammalamadaka, and W. Liu. Local posterior robustness with parametric priors: Maximum and average sensitivity. In *Maximum Entropy and Bayesian Methods*, pages 97–106. Springer, 1996.

A. Baydin, B. Pearlmutter, A. Radul, and J. Siskind. Automatic differentiation in machine learning: a survey. *Journal of Machine Learning Research*, 18:1–43, 2018.

J. O. Berger, D. R. Insua, and F. Ruggeri. Robust Bayesian analysis. In D. R. Insua and F. Ruggeri, editors, *Robust Bayesian Analysis*, volume 152. Springer Science & Business Media, 2000.

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006. Chapter 10.

D. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3: 993–1022, 2003.

D. Blei, A. Kucukelbir, and J. D. McAuliffe. Variational inference: A review for statisticians. *arXiv preprint arXiv:1601.00670*, 2016.

B. Carpenter, M. D. Hoffman, M. Brubaker, D. Lee, P. Li, and M. Betancourt. The Stan math library: Reverse-mode automatic differentiation in C++. *arXiv preprint arXiv:1509.07164*, 2015.

R. D. Cook. Assessment of local influence. *Journal of the Royal Statistical Society: Series B*, 28(2):133–169, 1986.

Criteo Labs. Criteo conversion logs dataset, 2014. URL `http://criteolabs.wpengine.com/downloads/2014-conversion-logs-dataset/`. Downloaded on July 27th, 2017.

P. Diaconis and D. Freedman. On the consistency of Bayes estimates. *The Annals of Statistics*, 14(1):1–26, 1986.

B. Efron. Frequentist accuracy of Bayesian estimates. *Journal of the Royal Statistical Society: Series B*, 77 (3):617–646, 2015.

W. H. Fleming. *Functions of Several Variables*. Addison-Wesley Publishing Company, Inc., 1965.

A. Gelman and J. Hill. *Data analysis using regression and multilevel/hierarchical models*. Cambridge university press, 2006.

A. Gelman, J. B. Carlin, H. S. Stern, and D. B. Rubin. *Bayesian Data Analysis*, volume 2. Chapman & Hall/CRC, 2014.

R. J. Giordano, T. Broderick, and M. I. Jordan. Linear response methods for accurate covariance estimates from mean field variational Bayes. In *Advances in Neural Information Processing Systems*, pages 1441–1449, 2015.

R. J. Giordano, T. Broderick, R. Meager, J. Huggins, and M. I. Jordan. Fast robustness quantification with variational Bayes. *arXiv preprint arXiv:1606.07153*, 2016.

P. Gustafson. Local sensitivity of inferences to prior marginals. *Journal of the American Statistical Association*, 91(434):774–781, 1996a.

P. Gustafson. Local sensitivity of posterior expectations. *The Annals of Statistics*, 24(1):174–195, 1996b.

P. Gustafson. Local robustness in Bayesian analysis. In D. R. Insua and F. Ruggeri, editors, *Robust Bayesian Analysis*, volume 152. Springer Science & Business Media, 2000.

P. J. Huber. *Robust Statistics*. Springer, 2011.

D. R. Insua and R. Criado. Topics on the foundations of robust Bayesian analysis. In D. R. Insua and F. Ruggeri, editors, *Robust Bayesian Analysis*, volume 152. Springer Science & Business Media, 2000.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

H. J. Kappen and F. B. Rodriguez. Efficient learning in Boltzmann machines using linear response theory. *Neural Computation*, 10(5):1137–1156, 1998.

R. W. Keener. *Theoretical Statistics: Topics for a Core Course*. Springer Science & Business Media, 2010.

M. Kéry and M. Schaub. *Bayesian population analysis using WinBUGS: a hierarchical perspective*. Academic Press, 2011.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

A. Kucukelbir, D. Tran, R. Ranganath, A. Gelman, and D. Blei. Automatic differentiation variational inference. *The Journal of Machine Learning Research*, 18(1):430–474, 2017.

L. Le Cam and G. L. Yang. *Asymptotics in Statistics: Some Basic Concepts*. Springer Science & Business Media, 2012.

D. Lewandowski, D. Kurowicka, and H. Joe. Generating random correlation matrices based on vines and extended onion method. *Journal of Multivariate Analysis*, 100(9):1989–2001, 2009.

Q. Liu and D. Wang. Stein variational gradient descent: A general purpose Bayesian inference algorithm. In *Advances In Neural Information Processing Systems*, pages 2370–2378, 2016.

D. J. C. MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003. Chapter 33.

D. Maclaurin, D. Duvenaud, and R. P. Adams. Autograd: Effortless gradients in numpy. In *International Conference on Machine Learning 2015 AutoML Workshop*, 2015.

E. Moreno. Global Bayesian robustness for some classes of prior distributions. In D. R. Insua and F. Ruggeri, editors, *Robust Bayesian Analysis*, volume 152. Springer Science & Business Media, 2000.

R. M. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Springer, 1998.

M. Opper and D. Saad. *Advanced Mean Field Methods: Theory and Practice*. MIT press, 2001.

M. Opper and O. Winther. Variational linear response. In *Advances in Neural Information Processing Systems*, pages 1157–1164, 2004.

A. B. Owen. *Monte Carlo Theory, Methods and Examples*. 2013. URL `http://statweb.stanford.edu/~owen/mc/`. Accessed November 23rd, 2016.

C. J. Pérez, J. Martín, and M. J. Rufo. MCMC-based local parametric sensitivity estimations. *Computational Statistics & Data Analysis*, 51(2):823–835, 2006.

R. Ranganath, D. Tran, and D. Blei. Hierarchical variational models. In *International Conference on Machine Learning*, pages 324–333, 2016.

J. Regier, A. Miller, J. McAuliffe, R. Adams, M. Hoffman, D. Lang, D. Schlegel, and M. Prabhat. Celeste: Variational inference for a generative model of astronomical images. In *International Conference on Machine Learning*, pages 2095–2103, 2015.

D. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. *arXiv preprint arXiv:1401.4082*, 2014.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

M. Roos, T. G. Martins, L. Held, and H. Rue. Sensitivity analysis for Bayesian hierarchical models. *Bayesian Analysis*, 10(2):321–349, 2015.

Stan Team. *Stan Modeling Language Users Guide and Reference Manual, Version 2.8.0*, 2015. URL `http://mc-stan.org/`.

Stan Team. Stan example models wiki, 2017. URL `https://github.com/stan-dev/example-models/wiki`. Referenced on May 19th, 2017.

T. Tanaka. Mean-field theory of Boltzmann machine learning. *Physical Review E*, 58(2):2302, 1998.

T. Tanaka. Information geometry of mean-field approximation. *Neural Computation*, 12(8):1951–1968, 2000.

Michalis Titsias and Miguel Lázaro-Gredilla. Doubly stochastic variational Bayes for non-conjugate inference. In *International Conference on Machine Learning*, pages 1971–1979, 2014.

D. Tran, D. Blei, and E. M. Airoldi. Copula variational inference. In *Advances in Neural Information Processing Systems*, pages 3564–3572, 2015a.

D. Tran, R. Ranganath, and D. Blei. The variational Gaussian process. *arXiv preprint arXiv:1511.06499*, 2015b.

R. E. Turner and M. Sahani. Two problems with variational expectation maximisation for time-series models. In D. Barber, A. T. Cemgil, and S. Chiappa, editors, *Bayesian Time Series Models*. 2011.

M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1-2):1–305, 2008.

B. Wang and M. Titterington. Inadequacy of interval estimates corresponding to variational Bayesian approximations. In *Workshop on Artificial Intelligence and Statistics*, pages 373–380, 2004.

Y. Wang and D. M. Blei. Frequentist consistency of variational Bayes. *arXiv preprint arXiv:1705.03439*, 2017.

M. Welling and Y. W. Teh. Linear response algorithms for approximate inference in graphical models. *Neural Computation*, 16(1):197–221, 2004.

T. Westling and T. H. McCormick. Establishing consistency and improving uncertainty estimates of variational inference through m-estimation. *arXiv preprint arXiv:1510.08151*, 2015.

S. Wright and J. Nocedal. Numerical optimization. *Springer Science*, 35:67–68, 1999.

H. Zhu, J. G. Ibrahim, S. Lee, and H. Zhang. Perturbation selection and influence measures in local influence analysis. *The Annals of Statistics*, 35(6):2565–2588, 2007.

H. Zhu, J. G. Ibrahim, and N. Tang. Bayesian influence analysis: A geometric approach. *Biometrika*, 98(2): 307–323, 2011.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 11/17; Revised 8/18; Published 9/18

Accelerating Cross-Validation in Multinomial Logistic Regression with ℓ_1-Regularization

Tomoyuki Obuchi OBUCHI@C.TITECH.AC.JP

Yoshiyuki Kabashima KABA@C.TITECH.AC.JP

Department of Mathematical and Computing Science
Tokyo Institute of Technology
2-12-1, Ookayama, Meguro-ku, Tokyo, Japan

Editor: Manfred Opper

Abstract

We develop an approximate formula for evaluating a cross-validation estimator of predictive likelihood for multinomial logistic regression regularized by an ℓ_1-norm. This allows us to avoid repeated optimizations required for literally conducting cross-validation; hence, the computational time can be significantly reduced. The formula is derived through a perturbative approach employing the largeness of the data size and the model dimensionality. An extension to the elastic net regularization is also addressed. The usefulness of the approximate formula is demonstrated on simulated data and the ISOLET dataset from the UCI machine learning repository. MATLAB and python codes implementing the approximate formula are distributed in (Obuchi, 2017; Takahashi and Obuchi, 2017).

Keywords: classification, multinomial logistic regression, cross-validation, linear perturbation, self-averaging approximation

1. Introduction

Multinomial classification is a ubiquitous task. There are several ways to treat this task, such as the naive Bayesian methods, neural networks, decision trees, and hierarchical classification schemes (Trevor et al., 2009). Among them, in this paper, we focus on multinomial logistic regression (MLR), which is simple but powerful enough to be used in many present day applications.

Let us denote each feature vector by $\boldsymbol{x}_\mu \in \mathbb{R}^N$ and its class by $y_\mu \in \{1, \cdots, L\}$, where $\mu = 1 \cdots, M$ denotes the index of given data. The MLR uses a linear structural model with parameters $\{\boldsymbol{w}_a \in \mathbb{R}^N\}_{a=1}^L$ and computes a class-a bias as an overlap:

$$u_{\mu a} = \boldsymbol{x}_\mu^\top \boldsymbol{w}_a. \tag{1}$$

A probability such that the feature vector $\boldsymbol{x}_\mu$ belongs to the class a is computed through a softmax function ϕ as:

$$\phi\left(a \middle| \{u_{\mu b}\}_{b=1}^L\right) = \frac{e^{u_{\mu a}}}{\sum_{b=1}^L e^{u_{\mu b}}}. \tag{2}$$

These define the MLR.

The maximum likelihood estimation is usually employed to train the MLR, though the learning result tends to be inefficient when the data size is not sufficiently larger than the model dimensionality or noises in relevant levels are present. A common technique to overcome this difficulty is to introduce a penalty or regularization. In this paper, we use an ℓ_1-regularization, which induces a sparse classifier as a learning result and is accepted to be effective. Given M data points $D^M \equiv \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^M$, the ℓ_1-regularized estimator is defined by the following optimization problem:

$$\{\hat{\boldsymbol{w}}_a(\lambda)\}_a = \underset{\{\boldsymbol{w}_a\}_a}{\arg\min} \left\{\mathcal{H}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda\right)\right\}, \tag{3}$$

$$\mathcal{H}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda\right) \equiv \sum_{\mu=1}^M q_\mu\left(\{\boldsymbol{w}_a\}_{a=1}^L\right) + \lambda \sum_{a=1}^L ||\boldsymbol{w}_a||_1, \tag{4}$$

$$q_\mu\left(\{\boldsymbol{w}_a\}_{a=1}^L\right) = -\ln \phi\left(y_\mu \middle| \left\{u_{\mu a} = \boldsymbol{x}_\mu^\top \boldsymbol{w}_a\right\}_{a=1}^L\right), \tag{5}$$

where we denote the negative log-likelihood as q_μ and define a regularized cost function or Hamiltonian $\mathcal{H}$.

The introduction of regularization causes another problem of model selection or hyperparameter estimation with respect to λ. A versatile framework providing a reasonable estimate is cross-validation (CV), but it has a disadvantage in terms of the computational cost. The literal CV requires repeated optimizations which can be a serious computational burden when the data size and the model dimensionality are large. The purpose of this paper is to resolve this problem by inventing an efficient approximation of CV.

Our technique is based on a perturbative expansion employing the largeness of the data size and the model dimensionality. Similar techniques were also developed for the Bayesian learning of simple perceptron and committee machine (Opper and Winther, 1996, 1997), for Gaussian process and support vector machine (Opper and Winther, 2000a,b; Vapnik and Chapelle, 2000), for linear regression with the ℓ_1-regularization (Obuchi and Kabashima, 2016; Rad and Maleki, 2018; Wang et al., 2018) and with the two-dimensional total variation (Obuchi et al., 2017). Actually, this perturbative approach is fairly general and can be applied to a wide class of generalized linear models with simple convex regularizations. For example in the present MLR case, it is easy to extend our result to the case where both the ℓ_1- and ℓ_2-regularizations exist (elastic net, Zou and Hastie, 2005), which is used in a common implementation (Friedman et al., 2010). The derivation of our approximate formula below is, however, conducted on the case of the ℓ_1-regularization only, for simplicity. The extension to the elastic net case is stated after the derivation.

The rest of the paper is organized as follows. In sec. 2, we state our formulation and how to derive the approximate formula. In sec. 3, we compare our approximation result with that of the literally conducted CV on simulated data and on the ISOLET dataset from UCI machine learning repository (Lichman, 2013). The accuracy and the computational time of our approximate formula are reported in comparison with the literal CV. The limitation of a simplified version of the approximation is also examined. The last section is devoted to the conclusion.

2. Formulation

In the maximum likelihood estimation framework, it is natural to employ a predictive likelihood as a criterion for model selection (Bjornstad, 1990; Ando and Tsay, 2010). We require a good estimator of the predictive likelihood, and the CV provides a simple realization of it. Particularly in this paper, we consider an estimator based on the leave-one-out (LOO) CV. The LOO solution is described by

$$\{\hat{\boldsymbol{w}}_a^{\backslash\mu}(\lambda)\}_a = \underset{\{\boldsymbol{w}_a\}_a}{\arg\min} \left\{ \mathcal{H}^{\backslash\mu}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda \right) \right\}, \tag{6}$$

$$\mathcal{H}^{\backslash\mu}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda \right) \equiv \mathcal{H}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda \right) - q_\mu\left(\{\boldsymbol{w}_a\}_{a=1}^L \right). \tag{7}$$

Denoting the overlap of $\boldsymbol{x}_\mu$ with the LOO solution as $\hat{u}_{\mu a}^{\backslash\mu} = \boldsymbol{x}_\mu^\top \hat{\boldsymbol{w}}_a^{\backslash\mu}$, as well as that with the full solution $\hat{u}_{\mu a} = \boldsymbol{x}_\mu^\top \hat{\boldsymbol{w}}_a$, we can define the LOO estimator (LOOE) of the predictive negative log-likelihood as:

$$\epsilon_{\mathrm{LOO}}(\lambda) = \frac{1}{M}\sum_{\mu=1}^M q_\mu\left(\left\{ \hat{\boldsymbol{w}}_a^{\backslash\mu} \right\}_{a=1}^L \right) = -\frac{1}{M}\sum_{\mu=1}^M \ln \phi\left(y_\mu \middle| \{\hat{u}_{\mu a}^{\backslash\mu}\}_{a=1}^L \right). \tag{8}$$

In the following, the predictive negative log-likelihood is simply called prediction error. The minimum of the LOOE determines the optimal value of λ though its evaluation requires us to solve eq. (6) M times, which is computationally demanding.

2.0.1. Notations

Here, we fix the notations for a better flow of the derivation shown below. By summarizing the class index, we introduce a vector notation of the overlap as $\boldsymbol{u}_\mu = (u_{\mu a})_a \in \mathbb{R}^L$ and an extended vector representation of the weight vectors $\{\boldsymbol{w}_a\}_a$ as $\boldsymbol{W} = (\boldsymbol{w}_a)_a \in \mathbb{R}^{LN}$. The mth component of $\boldsymbol{W}$ can thus be decomposed into two parts as $m = (m_c, m_f)$ where $m_c \in \{1, \cdots, L\}$ denotes the class index and $m_f \in \{1, \cdots, N\}$ represents the component index of the feature vector. Namely we write $W_m = w_{m_c m_f}$. Correspondingly, we leverage a matrix $X^\mu \in \mathbb{R}^{L \times LN}$ to define a repetition representation of the feature vector $\boldsymbol{x}_\mu$: Each component is defined as:

$$X_{am}^\mu \equiv \delta_{am_c} x_{\mu m_f}. \tag{9}$$

This yields simple and convenient relations:

$$\boldsymbol{u}_\mu = X^\mu \boldsymbol{W},\ X^\mu = \left(\frac{\partial \boldsymbol{u}_\mu}{\partial \boldsymbol{W}} \right)^\top. \tag{10}$$

Further, the class-a probability of μth data at the full solution $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{w}}_a)_a$ is denoted by:

$$p_{a|\mu} = \phi(a|\{\hat{u}_{\mu b}\}_b) = \frac{e^{\hat{u}_{\mu a}}}{\sum_{b=1}^L e^{\hat{u}_{\mu b}}} \tag{11}$$

These notations express the gradient and the Hessian of q_μ at the full solution as:

$$\nabla q_\mu(\hat{\boldsymbol{W}}) \equiv \left. \frac{\partial q_\mu}{\partial \boldsymbol{W}} \right|_{\boldsymbol{W}=\hat{\boldsymbol{W}}} = \frac{\partial \boldsymbol{u}_\mu}{\partial \boldsymbol{W}} \left. \frac{\partial}{\partial \boldsymbol{u}_\mu} q_\mu \right|_{\boldsymbol{u}_\mu = \hat{\boldsymbol{u}}_\mu} = (X^\mu)^\top \boldsymbol{b}^\mu, \tag{12}$$

$$\partial^2 q_\mu(\hat{\boldsymbol{W}}) \equiv \left. \frac{\partial^2 q_\mu}{\partial \boldsymbol{W} \partial \boldsymbol{W}'} \right|_{\boldsymbol{W}=\boldsymbol{W}'=\hat{\boldsymbol{W}}}$$
$$= \frac{\partial \boldsymbol{u}_\mu}{\partial \boldsymbol{W}} \left(\left. \frac{\partial^2 q_\mu}{\partial \boldsymbol{u}_\mu \partial \boldsymbol{u}'_\mu} \right|_{\boldsymbol{u}_\mu = \boldsymbol{u}'_\mu = \hat{\boldsymbol{u}}_\mu} \right) \left(\frac{\partial \boldsymbol{u}'_\mu}{\partial \boldsymbol{W}'} \right)^\top = (X^\mu)^\top F^\mu X^\mu, \tag{13}$$

where

$$\boldsymbol{b}^\mu \equiv (p_{1|\mu} - \delta_{1 y_\mu}, p_{2|\mu} - \delta_{2 y_\mu}, \cdots, p_{L|\mu} - \delta_{L y_\mu})^\top, \tag{14}$$

$$F_{ab}^\mu \equiv \delta_{ab} p_{a|\mu} - p_{a|\mu} p_{b|\mu}. \tag{15}$$

In addition, we denote the cost function Hessians at the respective solutions as:

$$G \equiv \partial^2 \mathcal{H}(\hat{\boldsymbol{W}}) = \sum_\mu \left(\partial^2 q_\mu(\hat{\boldsymbol{W}}) \right), \tag{16}$$

$$G^{\backslash\mu} \equiv \partial^2 \mathcal{H}^{\backslash\mu}(\hat{\boldsymbol{W}}^{\backslash\mu}) = \sum_{\nu(\neq\mu)} \left(\partial^2 q_\nu(\hat{\boldsymbol{W}}^{\backslash\mu}) \right). \tag{17}$$

Finally, we introduce the symbol $A(\boldsymbol{W}) \equiv \{m | W_m \neq 0\}$ representing the index set of the active components of $\boldsymbol{W}$ and $\hat{A} \equiv A(\hat{\boldsymbol{W}})$. Given $\hat{\boldsymbol{W}}$, we denote the active components of a vector $\boldsymbol{Y} \in \mathbb{R}^{LN}$ by the subscript as $\boldsymbol{Y}_{\hat{A}}$. A similar notation is used for any matrix and the symbol $*$ is assumed to represent all of the components in the corresponding dimension.

2.1. Approximate formula

For a simple derivation, it is important to consider that the $\boldsymbol{w}$-dependence of ϕ appears only in the overlap $u = \boldsymbol{x}^\top \boldsymbol{w}$. Hence, it is sufficient to provide the relation between $\hat{u}_{\mu a}$ and $\hat{u}_{\mu a}^{\backslash\mu}$ in order to derive the approximate formula.

A crucial assumption to derive the formula is that the active set is "common" between the full and LOO solutions, $\hat{\boldsymbol{W}} = (\hat{\boldsymbol{w}}_a)_a$ and $\hat{\boldsymbol{W}}^{\backslash\mu} = (\hat{\boldsymbol{w}}_a^{\backslash\mu})_a$; namely $\hat{A} = \hat{A}^{\backslash\mu} \equiv A(\hat{\boldsymbol{W}}^{\backslash\mu})$. Although this assumption is literally not true, we numerically confirmed that this approximately holds. In other words, the change of the active set is small enough compared to the size of the active set itself when considering the LOO operation when N and M are large. Moreover, in a related problem of an ℓ_1-regularized linear regression, the so-called LASSO, it has been shown that the contribution of the active set change vanishes in a limit $N, M \to \infty$ keeping $\alpha = M/N = O(1)$ (Obuchi and Kabashima, 2016). It is expected that the same holds in the present problem. Hence, we adopt this assumption in the following definition. Note that this idea of the active set constantness can be found in preceding analyses of support vector machine (Opper and Winther, 2000b; Vapnik and Chapelle, 2000).

Once the active set $\hat{A}$ is assumed to be known and unchanged by the LOO operation, it is easy to determine the active components of the full and LOO solutions $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{W}}^{\backslash\mu}$.

The vanishing condition of the gradient of the cost function is the determining equation:

$$(\nabla\mathcal{H})_{\hat{A}} = 0 \Rightarrow \hat{\boldsymbol{W}}_{\hat{A}}, \tag{18}$$

$$\left(\nabla\mathcal{H}^{\backslash\mu}\right)_{\hat{A}} = (\nabla\mathcal{H})_{\hat{A}} - (\nabla q_\mu)_{\hat{A}} = 0 \Rightarrow \hat{\boldsymbol{W}}^{\backslash\mu}_{\hat{A}}. \tag{19}$$

The difference between the gradients is only ∇q_μ, and hence the difference between $\hat{\boldsymbol{W}}$ and $\hat{\boldsymbol{W}}^{\backslash\mu}$ is expected to be small. Denoting the difference as $\boldsymbol{d}^\mu = \hat{\boldsymbol{W}} - \hat{\boldsymbol{W}}^{\backslash\mu}$ and expanding eq. (19) with respect to $\boldsymbol{d}^\mu$ up to the first order, we obtain an equation determining $\boldsymbol{d}^\mu$:

$$\boldsymbol{d}^\mu_{\hat{A}} = -\left(G^{\backslash\mu}_{\hat{A}\hat{A}}\right)^{-1}\left(\nabla q_\mu(\hat{\boldsymbol{W}})\right)_{\hat{A}}. \tag{20}$$

Inserting this and eq. (12) into the definition $\boldsymbol{d}^\mu = \hat{\boldsymbol{W}} - \hat{\boldsymbol{W}}^{\backslash\mu}$ and multiplying X^μ from left, we obtain:

$$\hat{\boldsymbol{u}}^{\backslash\mu}_\mu \approx \hat{\boldsymbol{u}}_\mu + C^{\backslash\mu}_\mu \boldsymbol{b}^\mu, \tag{21}$$

$$C^{\backslash\mu}_\mu \equiv X^\mu_{*\hat{A}}\left(G^{\backslash\mu}_{\hat{A}\hat{A}}\right)^{-1}\left(X^\mu_{*\hat{A}}\right)^\top. \tag{22}$$

This equation implies that the matrix inversion operation is necessary for each μ, which still requires a significant computational cost. To avoid this, we employ an approximation and the Woodbury matrix inversion formula in conjunction with eqs. (13,16,17). The result is:

$$\left(G^{\backslash\mu}\right)^{-1} \equiv \left(\partial^2\mathcal{H}^{\backslash\mu}(\hat{\boldsymbol{W}}^{\backslash\mu})\right)^{-1} \approx \left(\partial^2\mathcal{H}^{\backslash\mu}(\hat{\boldsymbol{W}})\right)^{-1} = \left(G - (X^\mu)^\top F^\mu X^\mu\right)^{-1}$$
$$= G^{-1} - G^{-1}(X^\mu)^\top\left(-F^\mu + X^\mu G^{-1}(X^\mu)^\top\right)^{-1} X^\mu G^{-1}. \tag{23}$$

Inserting this into eq. (21) and simplifying several factors, we obtain:

$$\hat{\boldsymbol{u}}^{\backslash\mu}_\mu \approx \hat{\boldsymbol{u}}_\mu + C_\mu\left(I_L - F^\mu C_\mu\right)^{-1}\boldsymbol{b}^\mu, \tag{24}$$

where

$$C_\mu = X^\mu_{*\hat{A}}\left(G_{\hat{A}\hat{A}}\right)^{-1}\left(X^\mu_{*\hat{A}}\right)^\top. \tag{25}$$

Now, all of the variables on the righthand side of eq. (24) can be computed from the full solution $\hat{\boldsymbol{W}}$ only, which enables us to estimate the LOOE by leveraging a one-time optimization using all of the data D^M, while avoiding repeated optimizations.

We should mention the computational cost of this approximation: it is mainly scaled as $O(ML^2|\hat{A}| + ML|\hat{A}|^2 + |\hat{A}|^3)$. The first two terms come from the construction of $G_{\hat{A}\hat{A}}$ and C_μ, and the last one is derived from the inverse of G. If $|\hat{A}|$ is proportional to the feature dimensionality N, this computational cost is of the third order with respect to the system dimensionality N and M. This is admittedly not cheap and the computational cost for the k-fold literal CV with a moderate value of k becomes smaller than that for our approximation in a large dimensionality limit. We, however, stress that there actually exists a wide range of N and M values in which our approximation outperforms the literal

CV in terms of the computational time, as later demonstrated in sec. 3. Moreover, for treating much larger systems, we invent a further simplified approximation based on the above approximate formula. The computational cost of this simplified version is scaled only linearly with respect to the system parameters N and M. Its derivation is in sec. 2.2 and the precision comparison to the original approximation is in sec. 3.

Another sensitive issue is present in computing $\left(G_{\hat{A}\hat{A}}\right)^{-1}$. Occasionally the cost function Hessian G has zero eigenvalues and is not invertible. We handle this problem in the next subsection.

2.1.1. Handling zero modes

In the MLR, there is an intrinsic symmetry such that the model is invariant under the addition of any constant vector to the weight vectors of all classes:

$$\boldsymbol{w}_a \to \boldsymbol{w}_a + \boldsymbol{v}\ (\forall a). \tag{26}$$

In this sense, the weight vectors defining the same model are "degenerated" and our MLR is singular. For finite λ, this is not harmful because the regularization term resolves this singularity and selects an optimal one $\{\hat{\boldsymbol{w}}_a\}_a$ with the smallest value of $||\boldsymbol{w}_a||_1$ among the degenerated vectors. However, this does not mean that the associated Hessian is non-singular. The regularization term does not provide any direct contribution to the Hessian and as a result, the Hessian tends to have some zero modes. This prevents taking the inverse Hessian G^{-1} in eq. (25). How can we overcome this?

One possibility is to fix the weights of one certain class at constant values when solving the optimization problem (4). This is termed "gauge fixing" in physics, and one convenient gauge in the present problem will be the zero gauge in which the weights in a chosen class are fixed at zeros. This is actually found in some earlier implementations (Krishnapuram et al., 2005; Schmidt, 2010) and is preferable for our approximate formula because it removes the harmful zero modes of the Hessian from the beginning. However, some other implementations which are currently well accepted do not employ such gauge fixing (Friedman et al., 2010), and moreover even with gauge fixing very small eigenvalues sometimes accidentally emerge in the Hessian. Hence, for user convenience, we require another way of avoiding this problem.

Another possibility is to remove the zero modes by hand. By construction, the zero modes are associated to the model invariance. This implies that those zero modes are irrelevant and may be removed. In fact, we are only interested in the perturbations which truly change the model, and the modes which maintain the model unchanged are unnecessary. According to this consideration, we replace G^{-1} in eq. (25) with the zero-mode-removed inverse Hessian $\overline{G}^{-1}$. The computation of $\overline{G}^{-1}$ is straightforward: we perform the eigenvalue decomposition of $G_{\hat{A}\hat{A}}$ and obtain the eigenvalues $\{d_i\}_{i=1}^{|\hat{A}|}$ and eigenvectors $\{\boldsymbol{v}_i\}_{i=1}^{|\hat{A}|}$, which allows us to represent

$$G_{\hat{A}\hat{A}} = \sum_i d_i \boldsymbol{v}_i\boldsymbol{v}_i^\top = \sum_{i\in S^+} d_i\boldsymbol{v}_i\boldsymbol{v}_i^\top, \tag{27}$$

where S^+ denotes the index set of the modes with finite eigenvalues. Then, $\overline{G}^{-1}$ is defined as:

$$\overline{G}^{-1}_{\hat{A}\hat{A}} \equiv \sum_{i\in S^+} d_i^{-1} \boldsymbol{v}_i \boldsymbol{v}_i^\top. \tag{28}$$

Finally, we replace G^{-1} by $\overline{G}^{-1}$ in eq. (25), and obtain:

$$C_\mu = X^\mu_{*\hat{A}} \overline{G}^{-1}_{\hat{A}\hat{A}} \left(X^\mu_{*\hat{A}}\right)^\top. \tag{29}$$

By using this instead of eq. (25), the problem caused by the zero modes can be avoided.

2.1.2. Extension to the mixed regularization case

Let us briefly state how we can generalize the present result to the case of the mixed regularizations of the ℓ_1- and ℓ_2-terms (elastic net, Zou and Hastie, 2005). The problem to be solved can be defined as follows:

$$\{\hat{\boldsymbol{w}}_a(\lambda_1,\lambda_2)\}_a = \underset{\{\boldsymbol{w}_a\}_a}{\arg\min} \left\{ \sum_{\mu=1}^M q_\mu\left(\{\boldsymbol{w}_a\}_{a=1}^L\right) + \lambda_1 \sum_{a=1}^L ||\boldsymbol{w}_a||_1 + \frac{\lambda_2}{2}\sum_{a=1}^L ||\boldsymbol{w}_a||_2^2 \right\}. \tag{30}$$

where $||\cdot||_2$ denotes the ℓ_2 norm. Following the derivation in sec. 2.1, we realize that the derivation is essentially the same, and the difference only appears in the cost function Hessian:

$$G_{\mathrm{mxd}} = \sum_\mu \left(\partial^2 q_\mu(\hat{\boldsymbol{W}})\right) + \lambda_2 I_{NL}, \tag{31}$$

where I_K is the identity matrix of size K. As a result, we can compute the LOO solution by leveraging the same equation as eq. (24) by replacing the definition of C_μ, eq. (25), with:

$$C_\mu = X^\mu_{*\hat{A}} \left((G_{\mathrm{mxd}})_{\hat{A}\hat{A}}\right)^{-1} \left(X^\mu_{*\hat{A}}\right)^\top. \tag{32}$$

Thanks to the ℓ_2 term, the zero mode removal is not needed since the eigenvalues are lifted up by λ_2.

2.1.3. Binomial case

The binomial case $L = 2$ is particularly interesting in several applications and thus we write down the specific formula for this case.

In the binomial case, it is fairly common to express the class y as a binary $y = 0, 1$ and to use the following logit function:

$$\phi_{\mathrm{logit}}(y_\mu|u_\mu) = \frac{\delta_{y_\mu 1} + \delta_{y_\mu 0} e^{-u_\mu}}{1 + e^{-u_\mu}}, \tag{33}$$

where

$$u_\mu = \boldsymbol{x}_\mu^\top \boldsymbol{w}. \tag{34}$$

If we identify $y = 0$ in this case as $y = 1$ in the two-class MLR case, this is nothing but the two-class MLR with a zero gauge $\boldsymbol{w}_1 = \boldsymbol{0}$. Hence, there is no harmful zero mode in the Hessian and we can straightforwardly apply our approximate formula. The explicit form in this case is:

$$\hat{u}_\mu^{\backslash\mu} \approx \hat{u}_\mu + \frac{c^\mu}{1 - \frac{\partial^2 q_\mu}{\partial u_\mu^2} c^\mu} \frac{\partial q_\mu}{\partial u_\mu}, \tag{35}$$

where $q_\mu = -\ln \phi_{\mathrm{logit}}(y_\mu|u_\mu)$ and

$$\frac{\partial q_\mu}{\partial u_\mu} = \delta_{y_\mu 0} - \frac{e^{-u_\mu}}{1+e^{-u_\mu}}, \tag{36}$$

$$\frac{\partial^2 q_\mu}{\partial u_\mu^2} = \frac{e^{-u_\mu}}{(1+e^{-u_\mu})^2}, \tag{37}$$

$$G_{\hat{A}\hat{A}} = \sum_{\mu=1}^M \frac{\partial^2 q_\mu}{\partial u_\mu^2} \left(\boldsymbol{x}_\mu \boldsymbol{x}_\mu^\top\right)_{\hat{A}\hat{A}}, \tag{38}$$

$$c^\mu = \left(\boldsymbol{x}_\mu^\top\right)_{\hat{A}} \left(G_{\hat{A}\hat{A}}\right)^{-1} (\boldsymbol{x}_\mu)_{\hat{A}}, \tag{39}$$

and $\hat{A} = \{i|\hat{w}_i \neq 0\}$ is the active set of the full solution, as before.

Note that this approximation can be easily generalized to arbitrary differentiable output functions by replacing the logit function ϕ_{logit}. Readers are thus encouraged to implement approximate CVs in a variety of different problems.

2.2. Further simplified approximation

As mentioned above, the computational cost of our approximation is $O(ML^2|\hat{A}|+ML|\hat{A}|^2+|\hat{A}|^3)$ and should be reduced for treating larger systems. For this, we derive a further simplified approximation based on the invented approximate formula above. We call this a self-averaging (SA) approximation according to physics terminology.

The basic idea for simplifying our approximate formula is to assume that correlations between W_m and W_n are sufficiently weak. The meaning of "correlation" is not evident here, but as seen in sec. A the Hessian G can be connected to a (rescaled) covariance χ between W_m and W_n in a statistical mechanical formulation introducing a probability distribution of $\boldsymbol{W}$. Our weak correlation assumption requires that the correlation between different feature components is negligibly small; $\chi_{mn}(\equiv (1/\beta)\mathrm{cov}(W_m, W_n)) = \chi_{(m_f,m_c),(n_f,n_c)} = \delta_{m_f n_f}(\chi_{m_f})_{m_c n_c}$, where $m_c, n_c (= 1, \cdots, L)$ are the class indices and $m_f, n_f (= 1, \cdots, N)$ are the feature component indices defined thus far, and β is the rescaling factor. In this way, the Hessian is assumed to be expressed in a rather restricted form:

$$\left(G^{\backslash\mu}\right)^{-1}_{mn} \approx \begin{cases} \left(\chi_{m_f}\right)_{m_c n_c} \delta_{m_f n_f}, & (m,n \in \hat{A}) \\ 0, & (\text{otherwise}) \end{cases}, \tag{40}$$

Namely, the SA Hessian is allowed to take finite values if and only if the two indices share the same feature vector component. The dependence on the data index μ is also assumed

to be negligible, implying that strong heterogeneity among feature vectors is assumed to be absent.

To proceed with the computation, we require a closed equation to determine the $L \times L$ matrix χ_i for $i = 1, \cdots, N$. Its derivation is rather technical and is deferred to sec. A. The result is:

$$(\chi_i)_{\hat{A}_i \hat{A}_i} = \left(\lambda_2 I_{|\hat{A}_i|} + \sigma_x^2 \sum_{\nu=1}^{M} \left(\left(I_L + F^\nu C_{\mathrm{SA}} \right)^{-1} F^\nu \right)_{\hat{A}_i \hat{A}_i} \right)^{-1}, \tag{41}$$

where $\sigma_x^2 = \sum_\mu \sum_i x_{\mu i}^2 / (NM)$ and $\hat{A}_i = \{a | \hat{w}_{ai} \neq 0\}$ is the set of active class variables at the feature component i; the other components of χ_i related to inactive variables are zeros. The SA approximation of $C_\mu^{\backslash \mu}$, $C_{\mathrm{SA}} \in \mathbb{R}^{L \times L}$, is defined by:

$$C_{\mathrm{SA}} = \sigma_x^2 \sum_{i=1}^{N} \chi_i. \tag{42}$$

Using the solution of eqs. (41,42), the approximate formula is now simply expressed as:

$$\hat{\boldsymbol{u}}_\mu^{\backslash \mu} \approx \hat{\boldsymbol{u}}_\mu + C_{\mathrm{SA}} \boldsymbol{b}^\mu. \tag{43}$$

Note that there is no factor like $(I_L - F^\mu C_\mu)^{-1}$ in contrast to eq. (24), because we directly approximate $C_\mu^{\backslash \mu}$ in eq. (21).

When solving eqs. (41,42), the inverse at the right-hand side of eq. (41) becomes occasionally ill-defined again due to the presence of zero modes. In such cases, we should remove the zero modes as eq. (28). Putting $R = \lambda_2 I_L + \sigma_x^2 \sum_{\mu=1}^{M} \left(\left(I_L + F^\mu C_{\mathrm{SA}} \right)^{-1} F^\mu \right)$ and performing the eigenvalue decomposition, we define its zero-mode-removed inverse $\overline{R}^{-1}$ as:

$$R_{\hat{A}_i \hat{A}_i} = \sum_j d_j \boldsymbol{v}_j \boldsymbol{v}_j^\top = \sum_{j \in S^+} d_j \boldsymbol{v}_j \boldsymbol{v}_j^\top \Rightarrow \overline{R}^{-1}_{\hat{A}_i \hat{A}_i} = \sum_{j \in S^+} d_j^{-1} \boldsymbol{v}_j \boldsymbol{v}_j^\top, \tag{44}$$

where S^+ is the index set of the modes with finite eigenvalues. This requires a $O(L^3)$ computational cost at a maximum. Leveraging this approach, a naive way to solve eqs. (41,42) is a recursive substitution. If this converges in a constant time, irrespectively of the system parameters N, M and L, then the computational cost of the SA approximation is scaled as $O(NL^3 + ML^3)$. This is linear in the feature dimensionality N and the data size M and hence, its advantage is significant.

2.3. Summary of procedures

Here, we summarize the two versions of the approximation derived thus far as algorithmic procedures. We call the first version, based on eq. (24), the approximate CV or ACV, and call the second one, using eq. (43), the self-averaging approximate CV or SAACV. The procedures of ACV and SAACV are given in Alg. 1 and Alg. 2, respectively; they are written for the case of the mixed regularization (30). Comments are added for

Algorithm 1 Approximate CV of the MLR

procedure ACV($\hat{\boldsymbol{W}}(\lambda_1, \lambda_2), D^M, \lambda_2$)
  Compute the active set $\hat{A}$ from $\hat{\boldsymbol{W}}$
  Compute $\{\hat{\boldsymbol{u}}_\mu, X^\mu, \boldsymbol{b}^\mu, F^\mu\}_\mu$ by eqs. (1,9), (14) and (15)
  $G_{\hat{A}\hat{A}} \leftarrow \sum_{\mu=1}^{M} (X^\mu)^\top F^\mu X^\mu + \lambda_2 I_{|\hat{A}|}$ ▷ $O(ML|\hat{A}|^2 + ML^2|\hat{A}|)$
  if λ_2 is large enough **then** ▷ $O(|\hat{A}|^3)$
    $\overline{G}^{-1}_{\hat{A}\hat{A}} = \left(G_{\hat{A}\hat{A}} \right)^{-1}$
  else
    Compute $\overline{G}^{-1}_{\hat{A}\hat{A}}$ by eq. (28)
  end if
  for $\mu = 1, \cdots, M$ **do** ▷ $O(ML|\hat{A}|^2 + ML^2|\hat{A}| + ML^3)$
    $C_\mu \leftarrow X^\mu_{*\hat{A}} \overline{G}^{-1}_{\hat{A}\hat{A}} \left(X^\mu_{*\hat{A}} \right)^\top$
    $\hat{\boldsymbol{u}}_\mu^{\backslash \mu} \leftarrow \hat{\boldsymbol{u}}_\mu + C_\mu \left(I_L - F^\mu C_\mu \right)^{-1} \boldsymbol{b}^\mu$
  end for
  Compute ϵ_{LOO} from $\{\boldsymbol{u}_\mu^{\backslash \mu}\}_\mu$ by eq. (8)
  return ϵ_{LOO}
end procedure

specifying the time consuming parts in the entire procedures. In Alg. 2, we describe an actual implementation for solving C_{SA} by recursion, which is not fully specified in sec. 2.2. The symbol $||\cdot||_{\mathrm{F}}$ denotes the Frobenius norm and we set the threshold θ judging the convergence as $\theta = 10^{-6}$ in typical situations. We also set as 10^{-6} the threshold judging if λ_2 is large or not.

3. Numerical experiments

In this section, we examine the precision and actual computational time of ACV and SAACV in numerical experiments. Both simulated and actual datasets (from UCI machine learning repository, Lichman, 2013) are used.

For examination, we compute the errors also by literally conducting k-fold CV with some ks, and compare it to the result of our approximate formula. In principle, we should compare our approximate result with that of the LOO CV ($k = M$) because our formula approximates it. However for large M, the literal LOO CV requires huge computational burdens despite that the result is empirically not much different from that of the k-hold CV with moderate ks. Hence in some of the following experiments with large M, we use the 10-hold CV instead of the LOO CV. Further, to directly check the approximation accuracy, we also compute the normalized error difference defined as

$$\frac{\epsilon_{\mathrm{LOO}}^{\mathrm{approximate}} - \epsilon_{\mathrm{CV}}^{\mathrm{literal}}}{\epsilon_{\mathrm{CV}}^{\mathrm{literal}}}, \tag{45}$$

where $\epsilon_{\mathrm{CV}}^{\mathrm{literal}}$ denotes the literal CV estimator of the prediction error while $\epsilon_{\mathrm{LOO}}^{\mathrm{approximate}}$ is the approximated LOOE. Moreover, as a reference, we compute the negative log-likelihood of

Algorithm 2 Self-averaging approximate CV of the MLR

procedure SAACV($\hat{\boldsymbol{W}}(\lambda_1, \lambda_2), D^M, \lambda_2$)
  Compute the active sets $\{\hat{A}_i\}_{i=1}^N$ from $\hat{\boldsymbol{W}}$
  Compute $\{\boldsymbol{u}_\mu, X^\mu, \boldsymbol{b}_\mu, F^\mu\}_\mu$ by eqs. (1,9), (14) and (15)
  $t \leftarrow 0$ ▷ Start initialization
  for $i = 1, \cdots, N$ **do**
    $\left(\chi_i^{\setminus\mu}\right)^{(t)} \leftarrow 0$,
    $\left(\chi_i^{\setminus\mu}\right)^{(t)}_{\hat{A}_i\hat{A}_i} \leftarrow \sigma_x^{-2}$,
  end for
  $\Delta \leftarrow 100$ ▷ End initialization
  while $\Delta > \theta$ **do** ▷ Compute $C_{\rm SA}$ by recursion
    $C_{\rm SA}^{(t+1)} \leftarrow \sigma_x^2 \sum_{i=1}^N \left(\chi_i^{\setminus\mu}\right)^{(t)}$
    $R \leftarrow \sigma_x^2 \sum_{\mu=1}^M \left(I_L + F^\mu C_{\rm SA}^{(t+1)}\right)^{-1} F^\mu + \lambda_2 I_L$ ▷ $O(ML^3)$
    $\Delta \leftarrow 0$
    for $i = 1, \cdots, N$ **do** ▷ $O(NL^3)$
      if λ_2 is large enough
        $\overline{R}^{-1}_{\hat{A}_i\hat{A}_i} = \left(R_{\hat{A}_i\hat{A}_i}\right)^{-1}$
      else
        Compute $\overline{R}^{-1}_{\hat{A}_i\hat{A}_i}$ by eq. (44) from R
      end if then
      $\left(\chi_i^{\setminus\mu}\right)^{(t+1)}_{\hat{A}_i\hat{A}_i} \leftarrow \overline{R}^{-1}_{\hat{A}_i\hat{A}_i}$
      $\Delta \leftarrow \Delta + \left|\left|\left(\chi_i^{\setminus\mu}\right)^{(t+1)}_{\hat{A}_i\hat{A}_i} - \left(\chi_i^{\setminus\mu}\right)^{(t)}_{\hat{A}_i\hat{A}_i}\right|\right|_{\rm F}$
    end for
    $\Delta \leftarrow \Delta / N$
    $t \leftarrow t + 1$
  end while
  for $\mu = 1, \cdots, M$ **do**
    $\boldsymbol{u}_\mu^{\setminus\mu} \leftarrow \boldsymbol{u}_\mu + C_{\rm SA}^{(t)} \boldsymbol{b}^\mu$
  end for
  Compute $\epsilon_{\rm LOO}$ from $\{\boldsymbol{u}_\mu^{\setminus\mu}\}_\mu$ by eq. (8)
  return $\epsilon_{\rm LOO}$
end procedure

the full solution $\{\hat{\boldsymbol{w}}_a\}_{a=1}^L$ as:

$$\epsilon = \frac{1}{M} \sum_{\mu=1}^M q_\mu \left(\{\hat{\boldsymbol{w}}_a\}_{a=1}^L \right), \tag{46}$$

and call it the training error, hereafter. The training error is expected to be a monotonic increasing function with respect to λ, while the prediction one is supposed to be non-monotonic.

In all of the experiments, we used a single CPU of Intel(R) Xeon(R) E5-2630 v3 2.4GHz. To solve the optimization problems in eqs. (4,6), we employed *Glmnet* (Friedman et al., 2010) which is implemented as a *MEX* subroutine in MATLAB®. The two approximations were implemented as raw codes in MATLAB. This is not the most optimized approach, because as seen in Algs. 1,2 our approximate formula uses a number of *for* and *while* loops which are slow in MATLAB, and hence the comparison is not necessarily fair. However, even in this comparison there is a significant difference in the computational time between the literal CV and our approximations, as shown below.

In Glmnet, the corresponding optimization problem is parameterized as follows:

$$\{\hat{\boldsymbol{w}}_a(\tilde{\lambda}, \eta)\}_a = \mathop{\rm arg\,min}_{\{\boldsymbol{w}_a\}_a} \left\{ \frac{1}{M} \sum_{\mu=1}^M q_\mu \left(\{\boldsymbol{w}_a\}_{a=1}^L \right) + \tilde{\lambda} \left(\eta \sum_{a=1}^L ||\boldsymbol{w}_a||_1 + \frac{(1-\eta)}{2} \sum_{a=1}^L ||\boldsymbol{w}_a||_2^2 \right) \right\}. \tag{47}$$

In the following experiments, we present the results based on this parameterization. We basically prefer $\eta = 1$ in which the ℓ_2 term is absent, because the main contribution of the present paper is to overcome technical difficulties stemming from the ℓ_1 term. However, Glmnet or its employing algorithm occasionally loses its stability in some uncontrolled manner without the ℓ_2 term. Hence, in the following experiments we adaptively choose the value of η.[1]

A sensitive point which should be noted is the convergence problem of the algorithm for solving the present optimization problem. In Glmnet, a specialized version of coordinate descent methods is employed, and it requires a threshold δ to judge the algorithm convergence. Unless explicitly mentioned, we set this as $\delta = 10^{-8}$ being tighter than the default value. This is necessary since we treat problems of rather large sizes. A looser choice for δ rather strongly affects the literal CV result, while it does not change the full solution or the training error as much. As a result, our approximations employing only the full solution are rather robust against the choice of δ compared to the literal CV. This is also demonstrated below.

3.1. On simulated dataset

Let us start by testing with the simulated data. Suppose each "true" feature vector $\boldsymbol{w}_{0a}$ is independently identically drawn (i.i.d.) from the following Bernoulli-Gaussian prior:

$$\boldsymbol{w}_{0a} \sim \prod_{i=1}^N \{(1-\rho_0)\delta(w_{0ai}) + \rho_0 \mathcal{N}(0, 1/\rho_0)\}, \tag{48}$$

1. When employing our distributed codes implementing the approximate formula (Obuchi, 2017; Takahashi and Obuchi, 2017) in conjunction with Glmnet, the parameters λ_1 and λ_2 are read as $\lambda_1 = M\tilde{\lambda}\eta$ and $\lambda_2 = M\tilde{\lambda}(1-\eta)$.

where $\mathcal{N}(\mu, \sigma^2)$ denotes a Gaussian distribution whose mean and variance are μ and σ^2, respectively. The resultant feature vector $\boldsymbol{v}_a$ becomes $N\rho_0(\equiv K_0)$-sparse and its norm becomes $\sqrt{N}$ on average. Then, we choose a class y_μ from $\{1, \cdots, L\}$ uniformly and randomly, and generate an observed feature vector $\boldsymbol{x}_\mu$ by leveraging the following linear process:

$$\boldsymbol{x}_\mu = \frac{\boldsymbol{w}_{0y_\mu}}{\sqrt{N}} + \boldsymbol{\xi}, \tag{49}$$

where $\boldsymbol{\xi}$ is an observation noise each component of which is i.i.d. from a Gaussian $\mathcal{N}(0, \sigma_N^2)$.

For convenience, we introduce the ratio of the data size to the feature dimensionality, $\alpha = M/N$, and now obtain five parameters $\{N, L, \alpha, \rho_0, \sigma_\xi^2\}$ characterizing the experimental setup. It is rather heavy to obtain the dependence of all parameters and below, and hence we mainly focus on the dependence on L, σ_ξ^2, and N. Other parameters are set to be $\alpha = 2$ and $\rho_0 = 0.5$.

3.1.1. Result

Let us summarize the result on simulated data.

Fig. 1 shows the plots of the prediction and training errors against $\tilde{\lambda}$ for $L = 4, 8, 16$ at $N = 200$ and $\sigma_\xi^2 = 0.01$. This demonstrates that both approximations provide consistent results with the literal LOO CV, except at small $\tilde{\lambda}$s. This inconsistency at small $\tilde{\lambda}$s is considered to be due to a numerical instability occurring in the literal CV. Actually, for small $\tilde{\lambda}$s, we have observed that certain small changes in the data induce large differences in the literal CV result. This example demonstrates that our approximations provide robust curves even in such situations. Note that as L grows the number of estimated parameters $\{\boldsymbol{w}_a\}_{a=1}^L$ increases while the data size $M = \alpha N = 400$ is fixed, meaning that the problem becomes more and more underdetermined with the growth of L. Hence, Fig. 1 demonstrates that the developed approximations work irrespectively of how much the problem is underdetermined.

Fig. 2 exhibits the σ_ξ^2-dependence of the errors and the approximation results for $L = 8$ and $N = 200$. For the very weak noise case ($\sigma_\xi^2 = 0.001$, left), the difference between the predictive and training errors is negligible and hence all four curves are not discriminable. For the moderate ($\sigma_\xi^2 = 0.1$, middle) and large ($\sigma_\xi^2 = 1$, right) noise cases, the training curve is very different from the predictive ones. The approximation curves are again consistent with the literal LOO one.

Fig. 3 demonstrates how the approximation accuracy changes as the system size N grows. For small sizes $N = 50, 100$, a discriminable difference exists between the results of the approximations and the literal LOO CV, as well as the difference between the results of the two approximations. This is expected, because our derivation relies on the largeness of N and M. For large systems $N = 400, 800$, the difference among the two approximations and the literal CV is much smaller. Considering this example in conjunction with the middle panel of Fig. 1, we can recognize that our approximate formula becomes fairly precise for $N \geq 200$ in this parameter set. The normalized error difference corresponding to Fig. 3 is shown in Fig. 4. We can observe that the difference tends to be smaller as the system size increases, which is expected because the perturbation employed in our approximate formula is justified in the large N, M limit.

Finally, let us consider the actual computational time to evaluate $\{\hat{\boldsymbol{w}}_a\}_a$ and the approximate LOOEs, and observe its system size dependence. The left panel of Fig. 5 provides

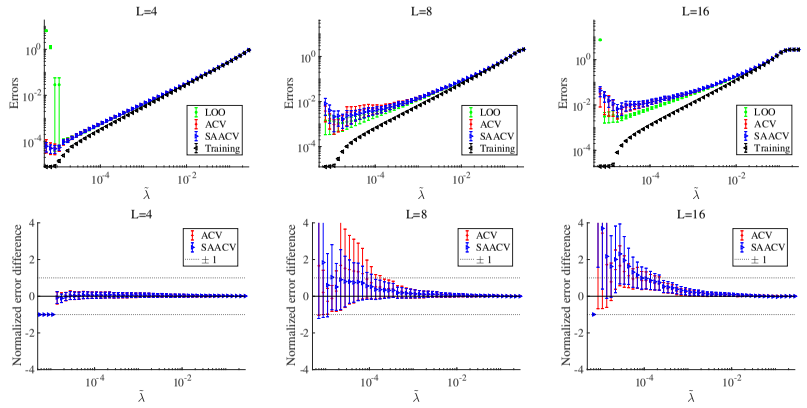


Figure 1: (Upper) Log-log plots of the errors against $\tilde{\lambda}$ for several values of the class number L. Other parameters are fixed at $N = 200$, $\sigma_\xi^2 = 0.01$, $\alpha = 2$ and $\rho_0 = 0.5$. The approximation results are consistent with the literal LOO CV results, except at small λs, which is presumably due to a numerical instability occurring in the literal CV at small λs. Here, $\eta = 0.9$. (Lower) The normalized error difference (45) plotted against $\tilde{\lambda}$. The parameters of each panel are them of the corresponding upper one. The horizontal dotted lines denote ± 1 and drawn for comparison. For small $\tilde{\lambda}$s the difference is not negligible, but the literal CV itself is not stable in that region and hence the error difference is not reliable.

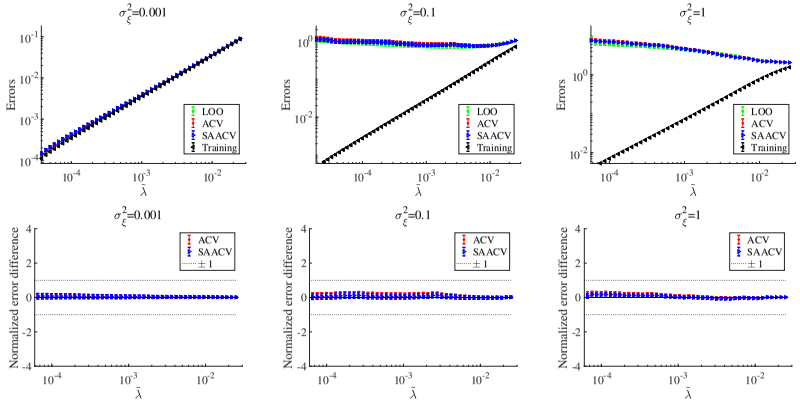


Figure 2: (Upper) Log-log plots of the errors against $\tilde{\lambda}$ for several noise strengths. Other parameters are fixed at $N = 200$, $L = 8$, $\alpha = 2$ and $\rho_0 = 0.5$. The approximation results are consistent with the literal LOO CV, irrespectively of the noise strength. The convergence threshold δ is set to be $\delta = 10^{-9}$ for the case $\sigma_\xi^2 = 1$. Here, $\eta = 1$. (Lower) The normalized error difference (45) plotted against $\tilde{\lambda}$. The parameters of each panel are them of the corresponding upper one. In the whole region the difference is negligibly small.

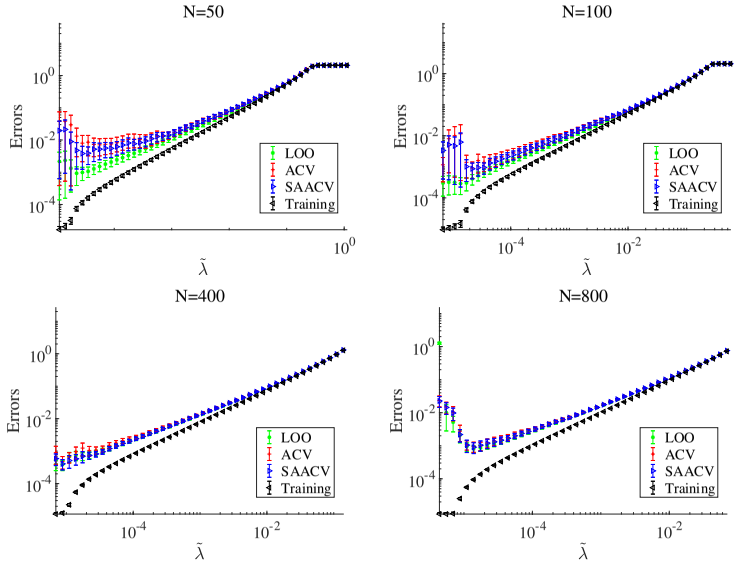


Figure 3: Log-log plots of the errors against $\tilde{\lambda}$ for several values of feature dimensionality N. Other parameters are fixed at $L = 8$, $\sigma_\xi^2 = 0.01$, $\alpha = 2$ and $\rho_0 = 0.5$. Here, $\eta = 0.9$.

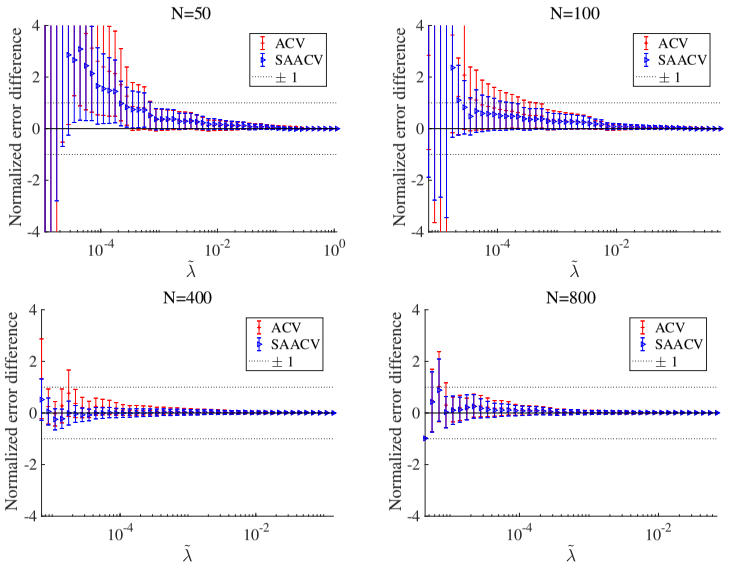


Figure 4: The plot of the normalized error difference corresponding to Fig. 3. The difference tends to be smaller as the system size increases.

the plot of the actual computational time against the system size. Here, the number of examined points of $\tilde{\lambda}$ to obtain a solution path is different from size to size, and hence the plotted time is given as the whole computational time to obtain the solution path divided by the number of $\tilde{\lambda}$s points. The left panel of Fig. 5 clearly displays the advantage and

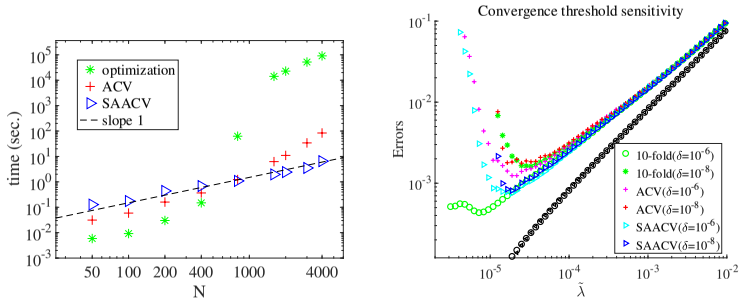


Figure 5: (Left) Actual computational time spent to find the solution of eq. (4) and that for ACV and SAACV, plotted against the feature dimensionality N in a double logarithmic scale. Note that the computational time for the k-fold CV is about k times larger than that for finding the solution of eq. (4), represented by the green asterisks. Parameters are fixed at $L = 8$, $\sigma_\xi^2 = 0.01$, $\alpha = 2$ and $\rho_0 = 0.5$. Here, $\eta = 1$. (Right) The errors are obtained for the two convergence thresholds $\delta = 10^{-6}$ and $\delta = 10^{-8}$. Error bars are omitted for visibility. For the tighter case $\delta = 10^{-8}$, the minimum value of $\tilde{\lambda}$ in the examined range is larger than that of the case $\delta = 10^{-6}$, though the systematic difference with the results of the literal LOO CV is already clear. The training errors of these two different δ, represented by black circles and left-pointing triangles, are completely overlapping. The system parameters are $N = 400$, $L = 8$, $\sigma_\xi^2 = 0.01$, $\alpha = 2$ and $\rho_0 = 0.5$. Here, $\eta = 1$.

disadvantage of the developed approximations. For small sizes, the computational time for optimization to obtain $\{\hat{\boldsymbol{w}}_a\}_a$ is shorter than the time to compute the approximate LOOEs, and hence the literal CV is better. However, for larger systems, the optimization cost increases rapidly and for $N \gtrsim 400$ the approximate CV is better. For $N \gtrsim 800$, the ACV cost exceeds that of SAACV. The SAACV cost behaves linearly as a function of N (see the black dashed line), and hence for larger systems of $N \gtrsim 800$ SAACV can be a very powerful tool. As a related issue, we mention the convergence problem of the algorithm. In the right panel of Fig. 5, we compare the errors at two different values of the convergence threshold δ. An important observation is that a significant difference exists in the literal CV results while other curves do not show a strong change. This implies that our approximate formula is rather robust and can be used with a rather loose convergence threshold or conversely, we can use the systematic deviation between the literal CV and our approximations as an

indicator to verify the tightness of the convergence threshold. This is beneficial, especially when treating large models, for which the convergence check is a common annoying task.

3.2. On real-world dataset

Next, we test the approximate formula on a real-world dataset. As shown above, our approximations become more precise if the model dimensionality and data size are large. Hence, we chose the ISOLET dataset which is a relatively large problem among classification tasks collected in the UCI machine learning repository (Lichman, 2013). The feature dimensionality, the data size, and the class number are $N = 617$, $M = 6238$, and $L = 26$, respectively. Here we apply the 10-fold CV, instead of the LOO CV because of the computational reason, and our approximations to this dataset. The result is given in Fig. 6. The

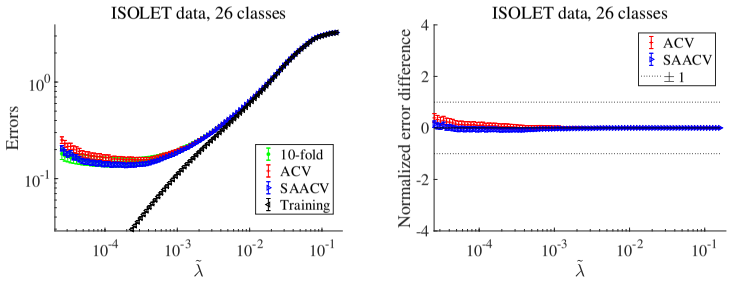


Figure 6: Approximate CV performance on the ISOLET data of $L = 26$ classes. The errors are shown in the left panel and the normalized error differences between the approximations and the 10-fold CV are in the right panel. At the estimated minimums of the prediction error, the accuracy rate for correctly classifying the test data is about 0.86 while the probability of recovering the training data is about 0.98, commonly among the literal CV and the two approximations. At the minimum value of $\tilde{\lambda}$, the leftmost point in the figure, the accuracy rates are different among the three different methods, and are 0.83, 0.78, and 0.81 for the literal CV, ACV and SAACV, respectively. Here, $\eta = 1$.

results of the approximations and of the 10-fold CV demonstrate a fairly good agreement, proving the actual effectiveness of the developed approximations. In an experiment, the actual computational time to obtain the result of the full simulation, of the 10-fold CV, of ACV, and of SAACV were 785, 7825, 5173, and 689 seconds, respectively. The system parameters $\{N, M, L\}$ are rather large in this problem and thus the advantage of ACV is not large, while the efficiency of SAACV stands out in such situations.

3.3. When does SAACV fail?

Two major factors neglected in SAACV are the correlations among feature components and the heterogeneity among feature vectors. If these factors are strong, the approximation accuracy of SAACV is expected to be degraded. In this section, we examine this point.

First, to test the impact of correlations in feature components, we add further constraints to the simulated data treated in sec. 3.1 and examine the approximation performance on the situation. Two cases are treated: the first is the case where the true feature vectors $\{\boldsymbol{w}_{0a}\}$ have common components among all the classes. The result of this case is shown in the left panels in Fig. 7. Here, the fraction of the common components to the non-zero components is $r_{\rm common} = 0.9$ and thus the overlap between feature vectors of different classes is rather large. The other is the case where the noise vector has strong correlations among the components. The result of this case is presented in the right panels in Fig. 7, in which the noise strength is $\sigma_\xi^2 = 1$ and the correlation coefficient of any pair of noise components is ${\rm Corr}(\xi_i, \xi_j) = 0.9$; hence the noise and the correlation are rather large. For both the cases, the performance of the approximate formula is fairly good, implying that SAACV is likely to perform well even when components of the feature vectors are correlated. Similar findings were actually obtained in the case of linear models (Obuchi and Kabashima, 2016). This is a preferable observation because it implies that the applicable limit of SAACV can be extended to a wider class of feature vectors than that is assumed in our present derivation in which the weakness of the correlations is assumed, as seen in sec. A. These also imply that there possibly exists another approximation formula taking into account the correlations but being similar to SAACV. A promising framework to derive such a formula might be the adaptive TAP method (Opper and Winther, 2001a,b, 2005). The adaptive TAP method itself requires a larger computational cost than that of SAACV but it is possible to reduce the computational cost up to the linear scaling with respect to N and M by employing an additional simplifying approximation (Kabashima and Vehkaperä, 2014; Çakmak and Opper, 2018). This is, however, rather technical and we leave it as a future work.

Second, to examine the effect of the heterogeneity among feature vectors, we introduce an amplifying factor Ω to control the norm of feature vectors. In particular, we multiply the factor Ω to the feature vectors of some chosen classes, as $\boldsymbol{x}_{\boldsymbol{\mu}} \to \Omega \boldsymbol{x}_{\boldsymbol{\mu}}$. Here, we use the simulated data identical to that for the center panels in Fig. 3 of the parameters $(N, L, \alpha, \rho_0, \sigma_\xi^2, \eta) = (200, 8, 2, 0.5, 0.1, 1)$, except that the amplifying factor $\Omega = 100$ is applied to the latter four classes $y = 5, 6, 7, 8$. The approximation performance on this dataset is shown in Fig. 8. We also examine the impact of the same heterogeneity on a real-world dataset in Fig. 9. Here, we treat the well-known MNIST data of handwritten digits (LeCun et al., 1998). For simplicity, we only use the data of two digits: 0 and 1. As a preprocessing, feature components with small variances are removed and only the $N = 350$ components of the largest variances are retained; the original size of the feature vector is $784 = 28 \times 28$ and thus almost the half of the components are discarded. Then, the usual standardization procedure is conducted. Further, we apply the amplifying factor $\Omega = 10$ to the class of 1 (right panels), while the case without the amplification (or $\Omega = 1$, left panels) is also examined for comparison. These two examples clearly show that ACV shows a consistency with the LOO CV behavior while SAACV does not, demonstrating that SAACV gives an inaccurate estimate of the CV error for datasets with strong heterogeneity. This kind of heterogeneity

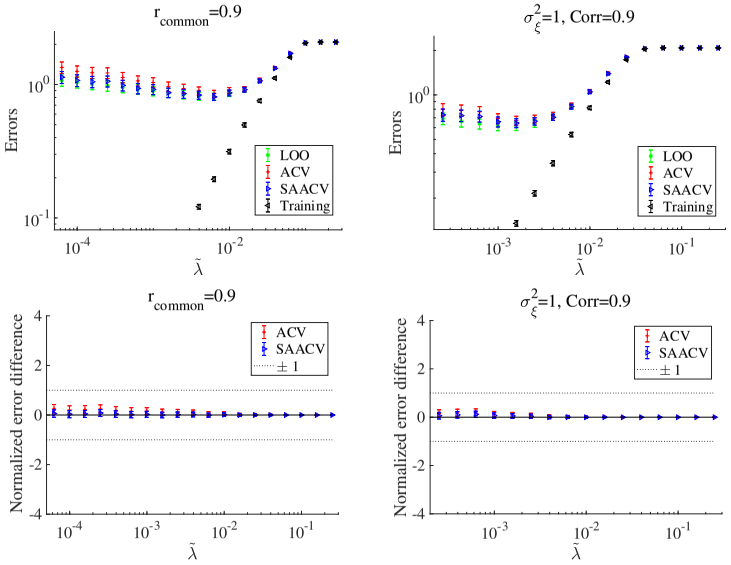


Figure 7: (Upper) Log-log plots of the errors against $\tilde{\lambda}$ for correlated feature vectors. The left panel is for the case with common components in true feature vectors while the right one is of the correlated noise case. Parameters $(N, L, \alpha, \rho_0, \eta) = (200, 8, 2, 0.5, 1)$ are common in both the cases, while the noise strengths and convergence thresholds are different: $(\sigma_\xi^2, \delta) = (0.1, 10^{-8})$ (left) and $(\sigma_\xi^2, \delta) = (1, 10^{-9})$ (right). (Lower) The normalized error difference (45) plotted against $\tilde{\lambda}$. The parameters of each panel are them of the corresponding upper one.

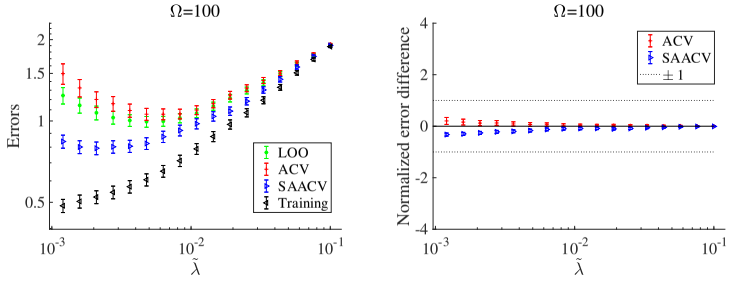


Figure 8: (Left) Log-log plots of the errors against $\tilde{\lambda}$ with strong heterogeneity in feature vectors. The same dataset as that of the center panels in Fig. 3 is used but the feature vectors for the classes $y_\mu = 5, 6, 7, 8$ are amplified as $\boldsymbol{x_\mu} \to \Omega \boldsymbol{x_\mu}$ by the factor $\Omega = 100$. The ACV result is consistent with the LOO CV one while that of SAACV is not. (Right) The normalized error difference corresponding to the left panel.

can naturally emerge in some applications: for example if we consider problems in medical statistics, a number of biological markers can give distinguishably large values for affected patients compared to unaffected ones, yielding larger values in norm for feature vectors of affected patients. This consideration suspects the efficiency of SAACV. We, however, stress that this kind of heterogeneity attributed to the belonging class can be absorbed by rescaling the weights as $\{\boldsymbol{w}_a\}_a \to \{\Omega_a^{-1}\boldsymbol{w}_a\}_a$, where Ω_a is chosen to homogenize the feature vector norm in different classes as $||\boldsymbol{x}_{y_\mu}\Omega_a||_2 \approx \text{const}$. For the ℓ_1 regularization case, this resultantly leads to the regularization coefficients which take different values adaptively to the belonging class as

$$\lambda \sum_a ||\boldsymbol{w}_a||_1 \to \sum_a \lambda \Omega_a ||\boldsymbol{w}_a||_1 = \sum_a \lambda_a ||\boldsymbol{w}_a||_1. \tag{50}$$

For this problem with adaptive regularization coefficients, our approximation formula can be applied in the completely same manner, which can be convinced by seeing Algs. 1,2 where the value of the regularization coefficient is not required as the argument. The ℓ_2-norm can also be handled, though the codes should be extended to take into account the groupwise coefficients as arguments. We argue that this rescaling is a natural prescription to treat strong heterogeneity among different classes, and once employing this prescription the weak point of SAACV is naturally cured.

As a noteworthy remark, we point out that the basic idea of SAACV is closely related to Wahba's generalized cross-validation (GCV) for linear regression (Golub et al., 1979). In GCV, the heterogeneity in coefficient corresponding to $C_\mu^{\backslash\mu}$ in SAACV is also neglected, and hence it shares the same weak point as SAACV, when it is regarded as an approximation of the CV estimator to generalization errors. We stress that this kind of approximation

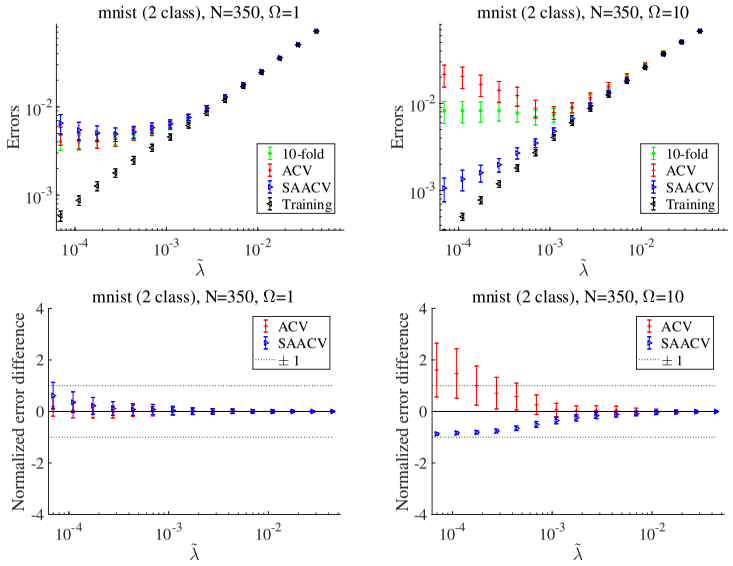


Figure 9: (Upper) Log-log plots of the errors against $\tilde{\lambda}$ of mnist handwritten data with two digits 0 and 1. The amplifying factor is not applied (or $\Omega = 1$) in the left panels while is applied ($\Omega = 10$) in the right ones. The strong heterogeneity among the classes affect the performance of SAACV. (Lower) The normalized error difference corresponding to the upper panels.

reducing the computational cost is again needed because the data size and the model dimensionality are increasing rapidly in recent years.

4. Conclusion

In this paper, we have developed an approximate formula for the CV estimator of the predictive likelihood of the multinomial logistic regression regularized by the ℓ_1-norm. An extension to the elastic net regularization has been also stated. We have demonstrated their advantages and disadvantages in numerical experiments using simulated and real-world datasets. Two versions of the approximation have been defined based on the developed formula. The first version, abbreviated as ACV, has a better performance, in terms of computational time, for middle size problems. It will eventually become worse than the literal k-fold CV with moderate ks as the problem size grows, because its computational time is scaled as a third-order polynomial of the feature dimensionality and data size, N and M, though such a tendency has not been observed in the investigated range of N. We have also defined the second version based on ACV, the computational time of which is just scaled linearly with respect to N and M. This second approximation is called SAACV, and it has been demonstrated that SAACV is slow for small size problems but has a great advantage for large size problems. Hence, we suggest leveraging the literal CV for small, ACV for middle, and SAACV for large size problems.

Our derivation is based on the perturbation which assumes that there is a small difference between the full and leave-one-out solutions. This assumption will not be satisfied for some specific cases. Even with this restriction, we expect the range of application of our formula is wide enough and we would like to encourage readers to leverage it in their own work. We have implemented MATLAB and python codes and they are available in (Obuchi, 2017; Takahashi and Obuchi, 2017).

The perturbative approach employed here is fairly general and can be applied to a wide class of generalized linear models with convex regularizations. The development of practical formulas for these cases will be of great assistance, given that we are living in the Big Data era.

Acknowledgments

This work was supported by JSPS KAKENHI Nos. 18K11463 (TO), 25120013 and 17H00764 (YK). TO is also supported by a Grant for Basic Science Research Projects from the Sumitomo Foundation. The authors are grateful to Takashi Takahashi for implementation of the approximation formula in python.

Appendix A. The SA approximation

Let us derive eq. (41) in the SA approximation. We work on a framework called a cavity method in statistical physics or belief propagation (BP) in computer science. We start from

defining the so-called Boltzmann distribution:

$$P\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda\right) = \frac{1}{Z\left(D^M, \lambda\right)} e^{-\beta \mathcal{H}\left(\{\boldsymbol{w}_a\}_{a=1}^L \middle| D^M, \lambda\right)}$$
$$= \frac{e^{-\beta \sum_a \left(\lambda_1 ||\boldsymbol{w}_a||_1 + \frac{\lambda_2}{2} ||\boldsymbol{w}_a||_2^2\right)}}{Z\left(D^M, \lambda\right)} \prod_{\mu=1}^M \phi^\beta \left(y_\mu | \{u_{\mu a}\}_a\right). \tag{51}$$

In the $\beta \to \infty$ limit, this distribution converges to a point-wise measure of the solution of eq. (4) and hence, it is useful for analyzing eq. (51). We note that the BP is usually applied to graphical models having sparse tree-like structures, but is also applicable to ones with densely connected structures. In such applications, the BP can be regarded as a systematic implementation of the Thouless-Anderson-Palmer (TAP) approach (Thouless et al., 1977) in statistical physics, which can yield a set of self-consistent equations of the first and second moments of variables in statistical models when the models are of densely connected types. This approach has been applied to many different models in machine learning, which continuously yields evidences of its effectiveness (Opper and Winther, 1996, 1997, 2000a,b). When applied to densely connected models, certain correlations between variables have to be neglected to make the computation tractable; for that reason this approach, or the associated algorithm derived from it, is recently called approximate message passing (AMP) (Kabashima, 2003; Donoho et al., 2009). Basically, the AMP assumes that "interactions" between the variables are weak: in the present problem this implies $(1/M)\sum_{\mu=1}^M x_{\mu_i} x_{\mu_j} - \left((1/M)\sum_{\mu=1}^M x_{\mu_i}\right)\left((1/M)\sum_{\mu=1}^M x_{\mu_j}\right) \approx 0$ $(i \neq j)$. This treatment can be justified if each feature vector $\boldsymbol{x}_\mu$ is i.i.d.. Rigorous proofs of this fact are available for linear models and their some variants (Bayati and Montanari, 2011; Barbier et al., 2017). We implicitly assume this in the following derivation.

In this appendix, we introduce a new vector representation summarizing class variables: $\boldsymbol{w}_i = (w_{ai})_a$. Note that this is different from the notation used in the main body of this paper, $\boldsymbol{w}_a = (w_{ai})_i$ in which the feature components are summarized.

By regarding $\boldsymbol{w}_i$ as a single variable node, the BP decomposes eq. (51) into two types of messages as follows:

$$\tilde{M}_{\mu\to i}(\boldsymbol{w}_i) = \int \prod_{j(\neq i)} d\boldsymbol{w}_j \ \phi^\beta(\boldsymbol{u}_\mu) \prod_{j(\neq i)} M_{j\to\mu}(\boldsymbol{w}_j), \tag{52}$$

$$M_{i\to\mu}(\boldsymbol{w}_i) = e^{-\beta\left(\lambda_1 ||\boldsymbol{w}_i||_1 + \frac{\lambda_2}{2}||\boldsymbol{w}_i||_2^2\right)} \prod_{\nu(\neq\mu)} \tilde{M}_{\nu\to i}(\boldsymbol{w}_i), \tag{53}$$

where $\boldsymbol{u}_\mu = (u_{\mu a})_a$. A crucial observation to assess eqs. (52,53) is that the argument of the potential function $\phi(\boldsymbol{u}_\mu)$ has a sum of an extensive number of random variables; the central limit theorem thus justifies treating it as a Gaussian variable with the appropriate mean and variance. Hence, according to eq. (52) where $\boldsymbol{w}_i$ is special, we can divide the extensive sum as follows:

$$u_{\mu a} = \sum_j x_{\mu j} w_{aj} = x_{\mu i} w_{ai} + \sum_{j(\neq i)} x_{\mu j} w_{aj} \approx x_{\mu i} w_{ai} + \sum_{j(\neq i)} x_{\mu j} \langle w_{aj} \rangle^{\setminus \mu} + t_a, \tag{54}$$

where the second term on the right-hand side represents the mean of $\sum_{j(\neq i)} x_{\mu j} w_{aj}$, the symbol $\langle\cdot\rangle^{\setminus\mu}$ denotes the average over the Boltzmann distribution without the μth potential function, and $\boldsymbol{t}^\mu = (t_a^\mu)_a$ denotes the zero-mean Gaussian variables whose covariance is set to be that of $\left(\sum_{j(\neq i)} x_{\mu j} w_{aj}\right)_a$. This expression allows us to replace the integration $\int \prod_{j(\neq i)} d\boldsymbol{w}_j$ by that over $\boldsymbol{t}^\mu$ in eq. (52). This significantly simplifies the computation and yields:

$$\tilde{M}_{\mu\to i}(\boldsymbol{w}_i) \approx \int d\boldsymbol{t} \ e^{\beta\left(-\frac{1}{2}\boldsymbol{t}^\top (C_\mu^{\setminus\mu})^{-1}\boldsymbol{t} - q_\mu(\boldsymbol{w}_i, \boldsymbol{t})\right)} \equiv \int d\boldsymbol{t} \ e^{\beta f^\mu(\boldsymbol{w}_i, \boldsymbol{t})} \tag{55}$$

where $q_\mu(\boldsymbol{w}_i, \boldsymbol{t})$ is the negative log-likelihood whose argument $u_{\mu a}$ is approximated by eq. (54) and $C_\mu^{\setminus\mu}$ is the rescaled covariance of $\sum_{j(\neq i)} x_{\mu j} w_{aj}$ defined as

$$\chi_{(ai)(bj)}^{\setminus\mu} \equiv \beta\left(\langle w_{ai} w_{bj}\rangle^{\setminus\mu} - \langle w_{ai}\rangle^{\setminus\mu}\langle w_{bj}\rangle^{\setminus\mu}\right),$$
$$\left(C_\mu^{\setminus\mu}\right)_{ab} \equiv \sum_{i,j} x_{\mu i} x_{\mu j} \chi_{(ai)(bj)}^{\setminus\mu}. \tag{56}$$

In the second equation we added the contribution from i for simplicity. It does not affect the following result because the ith term contribution is small enough. Let us focus on the limit $\beta \to \infty$. This limit allows us to use the saddle-point method, or Laplace's method, with respect to $\boldsymbol{t}^\mu$. The associated saddle-point equation is:

$$\hat{\boldsymbol{t}}^\mu = -C_\mu^{\setminus\mu} \boldsymbol{b}^\mu(\boldsymbol{w}_i, \hat{\boldsymbol{t}}^\mu), \tag{57}$$

where $\boldsymbol{b}^\mu(\boldsymbol{w}_i, \hat{\boldsymbol{t}}^\mu)$ is the gradient of q_μ defined at eq. (14) but the argument $u_{\mu a}$ is approximated by eq. (54). Now, let us expand the exponent $f^\mu(\boldsymbol{w}_i, \boldsymbol{t})$ in eq. (55) with respect to the dynamical variables $\boldsymbol{w}_i$ up to the second order. Putting $z_a = \sum_i x_{\mu i} w_{ai}$, we can define the derivatives as:

$$\frac{\partial \hat{\boldsymbol{t}}^\mu}{\partial z_a} = -(I_L + C_\mu^{\setminus\mu} F^\mu)^{-1} C_\mu^{\setminus\mu} F_{*a}^\mu, \tag{58}$$

$$\frac{\partial f^\mu(\boldsymbol{w}_i, \hat{\boldsymbol{t}}^\mu)}{\partial z_a} = -b_a^\mu(\boldsymbol{w}_i, \hat{\boldsymbol{t}}^\mu), \tag{59}$$

$$\frac{\partial^2 f^\mu(\boldsymbol{w}_i, \hat{\boldsymbol{t}}^\mu)}{\partial z_a \partial z_b} = -F_{ab}^\mu - \left(\frac{\partial \hat{\boldsymbol{t}}^\mu}{\partial z_a}\right)^\top F_{*b}^\mu = -\left(\left(I_L + F^\mu C_\mu^{\setminus\mu}\right)^{-1} F^\mu\right)_{ab}. \tag{60}$$

Hence,

$$\tilde{M}_{\mu\to i}(\boldsymbol{w}_i) \propto e^{\beta\left((\boldsymbol{h}_i^\mu)^\top \boldsymbol{w}_i - \frac{1}{2}\boldsymbol{w}_i^\top \Gamma_i^\mu \boldsymbol{w}_i\right)} \tag{61}$$

where

$$\boldsymbol{h}_i^\mu = -x_{\mu i}\boldsymbol{b}^\mu,$$
$$\Gamma_i^\mu = x_{\mu i}^2 (I_L + F^\mu C_\mu^{\setminus\mu})^{-1} F^\mu. \tag{62}$$

Note that this second order expansion is justified in the limit $\beta \to \infty$.

Collecting all the messages except for μ, we can construct the LOO marginal distribution of $\boldsymbol{w}_i$ as:

$$P^{\backslash\mu}(\boldsymbol{w}_i) \propto e^{-\beta\left(\lambda_1||\boldsymbol{w}_i||_1+\frac{\lambda_2}{2}||\boldsymbol{w}_i||_2^2\right)} \prod_{\nu(\neq\mu)} \tilde{M}_{\nu\to i}(\boldsymbol{w}_i)$$
$$\propto e^{\beta\left((\sum_{\nu(\neq\mu)} \boldsymbol{h}_i^\mu)^\top \boldsymbol{w}_i - \frac{1}{2}\boldsymbol{w}_i^\top \left(\lambda_2 I_L + \sum_{\nu(\neq\mu)} \Gamma_i^\mu\right)\boldsymbol{w}_i - \lambda_1||\boldsymbol{w}_i||_1\right)}. \tag{63}$$

Now, we can close the equation for the rescaled variance $\left(\chi_i^{\backslash\mu}\right)_{ab} \equiv \chi^{\backslash\mu}_{(ai),(bi)}$, because we can compute the variance of $\boldsymbol{w}_i$ from eq. (63). By considering the scaling, we can recognize that the variances vanish in the speed of $O(\beta^{-2})$ if one of the two components or both are inactive. The active-active components of the variance are scaled by $O(\beta^{-1})$ and remain in the rescaled variance. Focusing on the limit $\beta \to \infty$, we thus obtain:

$$\left(\chi_i^{\backslash\mu}\right)_{\hat{A}_i\hat{A}_i} = \left(\lambda_2 I_{|\hat{A}_i|} + \left(\sum_{\nu(\neq\mu)} \Gamma_i^\nu\right)_{\hat{A}_i\hat{A}_i}\right)^{-1} \approx \left(\lambda_2 I_{|\hat{A}_i|} + \left(\sum_{\nu} \Gamma_i^\nu\right)_{\hat{A}_i\hat{A}_i}\right)^{-1}. \tag{64}$$

At the last step, the μth term is added since its contribution is expected to be small enough in the summation. This manifests that the μ-dependence of $\chi^{\backslash\mu}$ can be neglected and we rewrite it as $\chi^{\backslash\mu} = \chi$ hereafter. By considering the meaning of the Hessian, it is easy to understand that $G^{\backslash\mu}$ is identified with $\left(\lambda_2 I_L + \sum_{\nu(\neq\mu)} \Gamma_i^\nu\right)$. This yields eq. (40).

By assuming the vanishing correlation between $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$ for $i \neq j$, we can write

$$\chi_{(ai)(bj)} \approx \delta_{ij} \begin{cases} (\chi_i)_{ab} & (a, b \in \hat{A}_i) \\ 0 & (\text{otherwise}) \end{cases}. \tag{65}$$

These leads to:

$$\left(C_\mu^{\backslash\mu}\right)_{ab} = \sum_{ij} x_{\mu i} x_{\mu j} \chi^{\backslash\mu}_{(ai)(bj)} \approx \sum_i x_{\mu i}^2 (\chi_i)_{ab} \approx \sigma_x^2 \sum_i (\chi_i)_{ab} \equiv (C_{\rm SA})_{ab} \tag{66}$$

The μ-dependence through $x_{\mu i}^2$ is neglected at the last step, because the sum $\sum_i$ would mask such a weak μ-dependence as long as strong heterogeneity in $\{x_{\mu i}\}_\mu$ is absent. Similarly, we may write the sum inside the parentheses of the righthand side of eq. (64) as:

$$\sum_\nu \Gamma_i^\nu \approx \sigma_x^2 \sum_\nu (I_L + F^\nu C_{\rm SA})^{-1} F^\nu. \tag{67}$$

Inserting eqs. (65-67) into eq. (64), we obtain eq. (41).

Careful readers may be concerned about the neglected μ-dependence of $\chi^{\backslash\mu}$, as well as that of $G^{\backslash\mu}$. If this can be neglected, may we replace $G^{\backslash\mu}$ with G from the beginning at eq. (21)? The answer is of course no. The reason is that the difference between $G^{\backslash\mu}$ and G is not negligible if they are "projected" onto X^μ as in eq. (21). If they are projected onto other directions perpendicular to X^μ, the difference is actually tiny and can be neglected, but for computing the factor $C_\mu^{\backslash\mu}$ we need to take into account this difference appropriately. This results in the additional factor $(I - F_\mu C_\mu)^{-1}$ in eq. (24). In the SA approximation, the

factor C is computed based on neglecting the difference between G and $G^{\backslash\mu}$. As a result we cannot discriminate the two factors C_μ and $C_\mu^{\backslash\mu}$. This consideration implies that our SA estimation of C, $C_{\rm SA}$, should be applied to $C_\mu^{\backslash\mu}$ in eq. (21) and should NOT be applied to C_μ in eq. (24), because the latter formula formally takes into account the difference in advance.

References

Tomohiro Ando and Ruey Tsay. Predictive likelihood for bayesian model selection and averaging. *International Journal of Forecasting*, 26(4):744–763, 2010.

Jean Barbier, Nicolas Macris, Mohamad Dia, and Florent Krzakala. Mutual information and optimality of approximate message-passing in random linear estimation. *arXiv preprint arXiv:1701.05823*, 2017.

Mohsen Bayati and Andrea Montanari. The dynamics of message passing on dense graphs, with applications to compressed sensing. *IEEE Transactions on Information Theory*, 57 (2):764–785, 2011.

Jan F Bjornstad. Predictive likelihood: a review. *Statistical Science*, pages 242–254, 1990.

Burak Çakmak and Manfred Opper. Expectation propagation for approximate inference: Free probability framework. *CoRR*, abs/1801.05411, 2018. URL `http://arxiv.org/abs/1801.05411`.

David L Donoho, Arian Maleki, and Andrea Montanari. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences*, 106(45):18914–18919, 2009.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

Gene H Golub, Michael Heath, and Grace Wahba. Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.

Yoshiyuki Kabashima. A CDMA multiuser detection algorithm on the basis of belief propagation. *Journal of Physics A: Mathematical and General*, 36(43):11111, 2003.

Yoshiyuki Kabashima and Mikko Vehkaperä. Signal recovery using expectation consistent approximation for linear observations. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 226–230. IEEE, 2014.

Balaji Krishnapuram, Lawrence Carin, Mario AT Figueiredo, and Alexander J Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):957–968, 2005.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

M. Lichman. UCI machine learning repository, 2013. URL `http://archive.ics.uci.edu/ml`.

Tomoyuki Obuchi. Matlab package of ACV on MLR. `https://github.com/T-Obuchi/AcceleratedCVonMLR_matlab`, 2017.

Tomoyuki Obuchi and Yoshiyuki Kabashima. Cross validation in lasso and its acceleration. *Journal of Statistical Mechanics: Theory and Experiment*, 2016(5):53304–53339, 2016.

Tomoyuki Obuchi, Shiro Ikeda, Kazunori Akiyama, and Yoshiyuki Kabashima. Accelerating cross-validation with total variation and its application to super-resolution imaging. *PloS one*, 12(12):e0188012, 2017.

Manfred Opper and Ole Winther. Mean field approach to bayes learning in feed-forward neural networks. *Physical review letters*, 76(11):1964, 1996.

Manfred Opper and Ole Winther. A mean field algorithm for bayes learning in large feed-forward neural networks. In *Advances in Neural Information Processing Systems*, pages 225–231, 1997.

Manfred Opper and Ole Winther. *Gaussian processes and SVM: Mean field results and leave-one-out*, pages 43–65. MIT, 10 2000a. ISBN 0262194481. Massachusetts Institute of Technology Press (MIT Press) Available on Google Books.

Manfred Opper and Ole Winther. Gaussian processes for classification: Mean-field algorithms. *Neural computation*, 12(11):2655–2684, 2000b.

Manfred Opper and Ole Winther. Adaptive and self-averaging thouless-anderson-palmer mean-field theory for probabilistic modeling. *Physical Review E*, 64(5):056131, 2001a.

Manfred Opper and Ole Winther. Tractable approximations for probabilistic models: The adaptive thouless-anderson-palmer mean field approach. *Physical Review Letters*, 86(17): 3695, 2001b.

Manfred Opper and Ole Winther. Expectation consistent approximate inference. *Journal of Machine Learning Research*, 6(Dec):2177–2204, 2005.

Kamiar Rahnama Rad and Arian Maleki. A scalable estimate of the extra-sample prediction error via approximate leave-one-out. *arXiv preprint arXiv:1801.10243*, 2018.

Mark Schmidt. Graphical model structure learning with l1-regularization. *University of British Columbia*, 2010.

Takashi Takahashi and Tomoyuki Obuchi. Python package of ACV on MLR. `https://github.com/T-Obuchi/AcceleratedCVonMLR_python`, 2017.

David J Thouless, Philip W Anderson, and Robert G Palmer. Solution of'solvable model of a spin glass'. *Philosophical Magazine*, 35(3):593–601, 1977.

Hastie Trevor, Tibshirani Robert, and Friedman Jerome. *The Elements of Statistical Learning; Data Mining, Inference, and Prediction*. Springer-Verlag New York, 2009. doi: 10.1007/978-0-387-84858-7.

Vladimir Vapnik and Olivier Chapelle. Bounds on error expectation for support vector machines. *Neural computation*, 12(9):2013–2036, 2000.

Shuaiwen Wang, Wenda Zhou, Haihao Lu, Arian Maleki, and Vahab Mirrokni. Approximate leave-one-out for fast parameter tuning in high dimensions. *arXiv preprint arXiv:1807.02694*, 2018.

Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 19 (2018) 1-40 Submitted 11/17; Revised 6/18; Published 9/18

Profile-Based Bandit with Unknown Profiles

Sylvain Lamprier sylvain.lamprier@lip6.fr
Sorbonne Universités, UPMC Paris 06, LIP6, CNRS UMR 7606

Thibault Gisselbrecht thibault.gisselbrecht@snips.ai
SNIPS, 18 rue Saint Marc, 75002 Paris

Patrick Gallinari patrick.gallinari@lip6.fr
Sorbonne Universités, UPMC Paris 06, LIP6, CNRS UMR 7606

Editor: Peter Auer

Abstract

Stochastic bandits have been widely studied since decades. A very large panel of settings have been introduced, some of them for the inclusion of some structure between actions. If actions are associated with feature vectors that underlie their usefulness, the discovery of a mapping parameter between such profiles and rewards can help the exploration process of the bandit strategies. This is the setting studied in this paper, but in our case the action profiles (constant feature vectors) are unknown beforehand. Instead, the agent is only given sample vectors, with mean centered on the true profiles, for a subset of actions at each step of the process. In this new bandit instance, policies have thus to deal with a doubled uncertainty, both on the profile estimators and the reward mapping parameters learned so far. We propose a new algorithm, called *SampLinUCB*, specifically designed for this case. Theoretical convergence guarantees are given for this strategy, according to various profile samples delivery scenarios. Finally, experiments are conducted on both artificial data and a task of focused data capture from online social networks. Obtained results demonstrate the relevance of the approach in various settings.

Keywords: Stochastic Linear Bandits, Profile-based Exploration, Upper Confidence Bounds

1. Introduction

Multi-armed bandits (MAB) correspond to online decision problems where, at each step of a sequential process, an agent has to choose an action - or arm - among a set of K actions, with the aim to maximize some cumulative reward function. In the so-called stochastic MAB setting, rewards collected for a given arm through time are assumed to be independently and identically distributed, following some hidden stationary distribution on every individual arm. The problem is therefore to deal with a tradeoff between exploitation - selecting actions according to some estimations about their usefulness - and exploration - selecting actions in order to increase the knowledge of their reward distribution. However, with classical stochastic bandit policies, the convergence towards the optimal arms can be slow when the number of actions becomes large.

On another hand, contextual bandits correspond to MAB settings where some side information can be leveraged to improve estimations of reward distributions. In these settings, a decision context is observed before selecting actions. This context can either correspond to a global decision feature vector or to specific feature vectors observed for each single action. Depending on the setting, these features can vary over time, which can help to predict reward fluctuations, or they can correspond to constant features on actions - that we call action profiles in the following - whose structure can be used to improve exploration policies over non-contextual approaches. In this paper we address the latter case, where we assume stationary distributions of rewards, but where distributions depend on constant profiles associated each arm. Reward distributions from the different arms are connected by a common unknown parameter to be learned (Filippi et al., 2010).

However, we introduce a new scenario where, for various possible reasons (technical, political, etc...), profile vectors are not available a priori. Instead, the agent gets sample vectors, centered on the true profiles, for a subset of actions at each decision step. This can happen in various situations where some restrictions limit our knowledge of the whole decision environment. For example in a focused data capture or technology intelligence scenario on social media, where an agent is asked to collect relevant information w.r.t. to a given need. Because of the extremely large number of accounts on media such as Twitter, the agent needs to focus on a subset of relevant users to follow at each time step (Gisselbrecht et al., 2015). However, given the strict restrictions set by the media, no knowledge about users is available beforehand. Profiles have therefore to be constructed from users activities, which are only observed for a small fraction of users at each step. As we will see later, the process which delivers profile samples can either be independent - e.g., an external process delivers activity samples for randomly chosen users at each step in the case of data capture from Twitter - or be included in the decision process - e.g., activity samples are only collected for followed users at the current time step.

To the best of our knowledge, this instance of contextual bandit has not been studied in the literature. Existing bandit approaches do not fit with this new setting. First, even if traditional algorithms such as `UCB` (Auer et al., 2002) could be applied, the information provided by the sample profile vectors would be entirely ignored. Our claim is that important benefits can arise from taking this available side-information into account. On the other hand, existing contextual bandit policies do not take into account uncertainty on context vectors, while we face here a bandit problem where uncertainty not only arises from regression parameters, as classically considered by contextual approaches, but also from the estimated profiles which serve as inputs for reward predictions at each step. The aim is to propose an approach able to leverage structural distributions of arms, based on noisy observations of their profiles, to improve over existing exploitation/exploration policies in bandit settings with partial side-information.

The contribution of this paper is threefold:

- We propose a new instance of the contextual bandit problem, based on constant contexts for each action, where action profiles are not known beforehand, but built from samples obtained at each iteration (3 cases are investigated regarding the sampling process);
- We design the *SampLinUCB* algorithm to solve this problem, for which we demonstrate some theoretical convergence guarantees;
- We experiment our proposal for both an artificial setting and a real-world task of focused data capture from Twitter, to empirically demonstrate the benefits of such a

profile-based approach with ony partial knowledge for exploitation/exploration problems.

The paper is organized as follows. In section 2, we present some background and related works. In section 3, we formalize the problem, propose our algorithm and derive the regret bound. Finally, section 4 reports our experiments.

2. Background: the Linear Stochastic Bandit

The multi-armed bandit problem, originally introduced in (Lai and Robbins, 1985) in its stationary form, has been widely studied in the literature. This learning problem aims at tackling the trade off between exploration and exploitation in decision processes where, at each step, an agent must choose an action - or arm - among a finite set of size K. After each decision, it receives a reward which quantifies the quality of the chosen action. The aim for the agent is to maximize the cumulative reward trough time, or equivalently to minimize the cumulative regret R_T at step T defined as:

$$R_T = \max_{i\in\{1,2,...,K\}} \sum_{t=1}^{T} r_{i,t} - \sum_{t=1}^{T} r_{i_t,t} \tag{1}$$

where i_t stands for the action selected at step t and $r_{i,t}$ the reward obtained by playing the action i at step t. This represents the amount of rewards that has been lost by selecting i_t at each step t, compared to what could be obtained by playing the optimal arm from the beginning to the end of the process. Note that, at each step t, only the reward for the chosen action $r_{i_t,t}$ is observed in practice, other ones remain unknown.

In the so-called stochastic case, one assume that rewards of an arm i are identically and independently sampled from a distribution with mean ν_i. Therefore, one usually rather consider the pseudo-regret of a policy, which introduces expectations of regret in the previous definition:

$$\hat{R}_T = T\nu_{i^*} - \sum_{t=1}^{T} \nu_{i_t} \tag{2}$$

where $i^\star$ stands as the arm with the best reward expectation.

One of the simplest and most straightforward algorithms to deal with the stochastic bandit problem is the well-known ϵ-greedy algorithm (Auer et al., 2002). This algorithm selects the arm with the best reward mean empirical estimation with probability $1-\epsilon$ and uniformly selects an arm among the whole set regardless their current estimations with probability ϵ. This guarantees to regularly reconsider estimations of all arms and therefore prevents from getting stuck on sub-optimal arms. However, the reward loss resulting from these blind selections prevents from ensuring a sub-linear upper bound of the pseudo-regret, unless setting an appropriate decay on ϵ. But this requires to know a lower bound on the difference of reward expectations between the best and the second best action (Auer et al., 2002).

Upper Confidence Bound algorithms (UCB) is another family of bandit approaches which define confident intervals for the reward expectations of each arm. Based on some concentration inequalities (Hoeffding, Bernstein, etc.), they propose optimistic policies which

consider possible deviations of the estimated mean of each arm. By using upper bounds of confidence intervals as selection scores, they ensure a cleaver balance between exploitation and exploration. Many extensions of the famous `UCB` algorithm proposed in (Auer et al., 2002) are known to guarantee a sub-linear bound of the pseudo-regret (see `UCBV` in (Audibert et al., 2009), `MOSS` in (Audibert and Bubeck, 2009) or `KL-UCB` in (Garivier, 2011)).

At last, Thompson sampling algorithms, originally proposed in (Thompson, 1933), develop a Bayesian approach to deal with uncertainty. By sampling from posterior distributions for the reward parameters, their exploration/exploitation mechanism is also proved to ensure a sub-linear regret (see (Kaufmann et al., 2012b) and (Agrawal and Goyal, 2012)).

The contextual bandit setting is an instance of the bandit problem where context vectors are observed before each decision step. Typically, contextual bandits assume a linear relation between context features and reward expectations. Formally, if we observe a context vector $x_{i,t} \in \mathbb{R}^d$ for each action $i \in \mathcal{K}$ at each time-step t, we consider the following assumption:

$$\exists \beta \in \mathbb{R}^d \text{ such that } r_{i,t} = x_{i,t}^\top \beta + \eta_{i,t} \tag{3}$$

where β is a mapping parameter between contexts and rewards, $\eta_{i,t}$ is a zero-mean conditionally R sub-Gaussian random noise, with constant $R > 0$ i.e: $\forall \lambda \in \mathbb{R} : \mathbb{E}[e^{\lambda \eta_{i,t}} | \mathcal{H}_{t-1}] \leq e^{\lambda^2 R^2/2}$, with $\mathcal{H}_{t-1} = \{(i_s, x_{i_s,s}, r_{i_s,s})\}_{s=1..t-1}$.

In this context, given a set $\mathcal{K}$ of K actions, any contextual bandit algorithm proceeds at each step $t \in \{1, 2, 3, \ldots, T\}$ as follows:

1. Observation of the context vector $x_{i,t} \in \mathbb{R}^d$ for each $i \in \{1, ..., K\}$;
2. According to the current estimate of β, selection of an action i_t and reception of the associated reward $r_{i_t,t}$;
3. Improvement of the selection policy by considering the new input $(i_t, x_{i_t,t}, r_{i_t,t})$ for the estimation of β.

Various contextual algorithms have been proposed in the literature. The first contextual bandit algorithm was introduced in (Auer, 2003). More recently the well-known `LinUCB` algorithm has been proposed for a task of personalized recommendation in (Li et al., 2010) and analyzed in (Chu et al., 2011). Both of these algorithms are UCB-like policies, each of them selecting the action whose upper bound of its reward confidence bound is the highest. Many other UCB approaches have been developed since then. In particular, algorithms such as `OFUL` or `ConfidenceBall` proposed in (Abbasi-Yadkori et al., 2011) and (Dani et al., 2008) have the advantage to enjoy a tighter regret upper bound (see also (Kaufmann et al., 2012a) and (Rusmevichientong and Tsitsiklis, 2010)). As in the stochastic bandit setting, Thompson sampling algorithms have also been designed for the contextual case, which also proved to be powerful, first empirically in (Chapelle and Li, 2011) and then theoretically in (Agrawal and Goyal, 2013) and (May et al., 2012).

In this paper, we consider a variant of the contextual bandit problem where contexts of actions are constant, which we call action profiles in the following. Hence, in our setting we assume that each action $i \in \mathcal{K}$ is associated with a profile vector $\mu_i \in \mathbb{R}^d$. The linear assumption of equation 3 becomes:

$$\exists \beta \in \mathbb{R}^d \textit{ such that } r_{i,t} = \mu_i^\top \beta + \eta_{i,t} \tag{4}$$

Thus, in this setting contexts cannot be used to anticipate some variations in the rewards expectations as it is traditionally the case in the literature about contextual bandit, but they can be leveraged to improve the exploration process, the use of a shared mapping parameter β allowing one to define areas of interest in the representation space of the actions. To illustrate this, the figure 1 represents the selection scores of a contextual algorithm (such as *OFUL* that we rely on in the following) at a given step for a simple case where $K = 4$ and $d = 2$. In this figure, green areas correspond to high scores, whereas red ones correspond to low scores areas. Color variations render the latent structure inferred by the model. In this setting, the algorithm would select the action 1, since its profile is located in the most promising area of the space. On the other hand, the action 3 is located in an area that is greatly less promising. The fact of using a common mapping parameter β allows one to perform a mutual learning, where observations on some actions inform on the usefulness of similar ones. This allows one to improve the exploration process by focusing more quickly on the useful areas: Imagine that a great number of actions are located in the red area of the figure. In that case, a classical bandit algorithm such as *UCB* would need to consider each of these actions several times to reveal their low reward expectation. On the other hand, a contextual bandit algorithm such as *OFUL* is able to avoid these actions really more quickly because of the proximity with other bad actions.

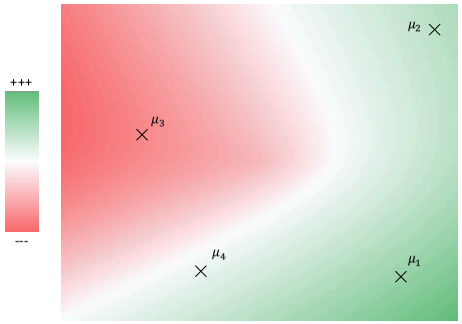


Figure 1: Illustration of `OFUL` scores for a profiles representation space.

This structured bandit setting has already been investigated in (Filippi et al., 2010). This work showed that great improvements could indeed be obtained by exploiting the structure of actions, since observations on some actions inform on the usefulness of similar ones. This comes down to a classical stochastic bandit where the pseudo-regret can be defined as follows:

$$\hat{R}_T = \sum_{t=1}^{\top} \mu_{i^\star}^\top \beta - \mu_{i_t}^\top \beta \tag{5}$$

where $\mu_i \in \mathbb{R}^d$ stands for the profile of the action $i \in \mathcal{K}$ and $\mu_{i^\star} = \underset{\mu_i, i=1..K}{\arg\max}\, \mu_i^\top \beta$ corresponds to the profile of the optimal action $i^\star$.

This is the setting which is studied in this paper. However, in our case the profile vectors are **unknown** to the agent beforehand, they have to be discovered iteratively during the process. Our problem differs from existing instances by the following two main aspects:

1. Action profiles are not directly available, one only get samples centered on them during the process;
2. At each step, one only get samples for a subset of actions.

In the following, we derive an UCB-based policy for this new setting.

3. Profile-Based Bandit with Unknown Profiles

In this section, we explore our new setting in which the set of profiles $\{\mu_1, .., \mu_K\}$ is not directly observed. Instead, at each iteration t, the agent is given a subset of actions $\mathcal{O}_t$ such that for every $i \in \mathcal{O}_t$, a sample $x_{i,t}$ of a random variable centered on μ_i is revealed. By assuming the same linear hypothesis as described in the previous section (formula 4), the relation of rewards with profiles can be re-written as follows for any time t, in order to introduce profile samples:

$$\begin{aligned} \forall s \leq t : r_{i,s} &= \mu_i^\top \beta + \eta_{i,s} \\ &= \hat{x}_{i,t}^\top \beta + (\mu_i - \hat{x}_{i,t})^\top \beta + \eta_{i,s} \\ &= \hat{x}_{i,t}^\top \beta + \epsilon_{i,t}^\top \beta + \eta_{i,s} \end{aligned} \tag{6}$$

where $\epsilon_{i,t} = \mu_i - \hat{x}_{i,t}$, $\hat{x}_{i,t} = \dfrac{1}{n_{i,t}} \sum\limits_{s \in T_{i,t}^{obs}} x_{i,s}$, with $T_{i,t}^{obs} = \{s \leq t, i \in \mathcal{O}_s\}$ and $n_{i,t} = |T_{i,t}^{obs}|$. In words, $n_{i,t}$ corresponds to the number of times a sample has been obtained for the action i until step t and $\hat{x}_{i,t}$ corresponds to the empirical mean of observed samples for i at time t. $\epsilon_{i,t}$ corresponds to the deviation of the estimator $\hat{x}_{i,t}$ from the true profile μ_i.

Compared to traditional contextual bandits, the uncertainty is double : as classically it arises from the β parameter estimator, but also from the profile estimators, since the algorithm must both estimate β and the profile vectors $\{\mu_1, .., \mu_K\}$ from observations. Figure 2[1] illustrates this new setting. Contrary to figure 1 where profiles are known, here we only get confidence areas for them, represented by circles centered on their corresponding empirical mean (represented by a blue cross). From the law of large numbers, the more observations for a given action we get, the lower the deviation between its true profile and its empirical estimator is. Therefore, the more knowledge we get about a given action, the smaller its confidence circle is. The best action is still the action 1, whose true profile (represented by a black cross) is in the greenest area. However, this information is unknown from the agent. A naive solution would be to directly use the empirical mean for each action in order to determine the selection scores. From the figure, this would lead to select the sub-optimal

1. Note that this is only an illustration of the general principle, in practice the surface of selection scores should also differ from figure 1, since β is estimated from biased inputs.

action 2, whose empirical mean is located in a greener area than the one of other actions. We propose to include the additional uncertainty in the selection scores by using the best location inside the confidence ellipsoid it is possible to reach for each action. This allows one to define an optimistic policy which would select the optimal action 1 in the example figure, whose confidence area contains the most promising profiles (i.e., includes the most green locations in the figure).

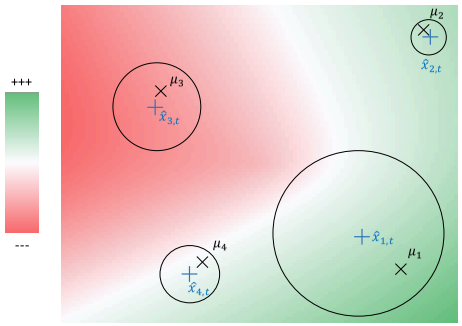


Figure 2: Illustration of the additional uncertainty arising from profile estimators.

In the following we propose to define an algorithm fitted for this setting. After deriving the algorithm in a generic context of samples delivery, we consider three different cases:

- **Case 1**: Every action delivers a profile sample at each step t (i.e., $\forall t$, $\mathcal{O}_t = \{1, ..., K\}$);
- **Case 2**: Each action i owns a probability p_i of sample delivery: at each step t, an action is included in $\mathcal{O}_t$ with probability p_i;
- **Case 3**: At each step t, only the action selected at the previous step delivers a sample (i.e., $\forall t$, $\mathcal{O}_t = i_{t-1}$).

The first case corresponds to the simplest case, where $\mathcal{O}_t$ is constant over time. The second case includes an additional difficulty since before any step, every action has not been observed the same number of times, which leads to different levels of uncertainty. The last case, probably the most interesting one for real-world applications, is the most difficult since decisions at each step not only affect knowledge about reward distributions but also profile estimations. For that case, it appears mandatory to take the uncertainty about profiles into account in the selection policy to guarantee the process to converge towards optimal actions.

After deriving confidence intervals for this new bandit instance, this section describes the proposed algorithm and analyzes its regret for the three settings listed above. Then, we consider the case where multiple actions can be performed at each step.

3.1. Confidence Intervals

In the following, we derive a series of propositions which will allow us to define a selection policy for our setting of profile-based bandit with unknown profiles. First, it is needed to define an estimator for the mapping parameter β, and the associated confidence ellipsoid. For that purpose, we rely on results from the theory of the self-normalized process (de la Peña et al., 2009). The next step is to define a way to mix it with the uncertainty on profiles to define an optimistic policy.

Proposition 1 *Let us consider that for any action i, all profile samples $x_{i,t} \in \mathbb{R}^d$ are iid from a distribution with mean $\mu_i \in \mathbb{R}^d$. Let us also assume that there exists a real number $L > 0$ such that $||x_{i,t}|| \leq L$ and a real number $S > 0$ such that $||\beta|| \leq S$. Then, for any $i \in \mathcal{K}$ and any step $s \leq t$, the random variable $\eta_{i,t,s} = \epsilon_{i,t}^\top \beta + \eta_{i,s}$ is conditionally sub-Gaussian with constant $R_{i,t} = \sqrt{R^2 + \frac{L^2 S^2}{n_{i,t}}}$.*

Proof Available in appendix A.1. ■

In this proposition, R is the constant of the sub-Gaussian random noise of the rewards (η_s from equation 6) and the notation $||x||$ stands as the norm of a vector x (i.e., $||x|| = \sqrt{x^T x}$). Since the noise $\eta_{i,t,s}$ is sub-Gaussian, it will be possible to apply the theory of self-normalized process for defining a confidence ellipsoid for β.

At step t, we can use the following set of observations to find an estimator for β: $\{(\hat{x}_{i_s,t}, r_{i_s,s})\}_{s=1..t-1}$ (i.e., at any decision step t, the reward $r_{i_s,s}$ observed at any previous step $s < t$ is associated with the profile of the selected action at step s, estimated knowing samples observed from step 1 to step t). The following notations are used in the remaining of the paper:

- $\eta'_{t-1} = (\eta_{i_s,s} + \epsilon_{i_s,t}^\top \beta)_{s=1..t-1}^\top$ the vector of noises of size $t-1$.
- $X_{t-1} = (\hat{x}_{i_s,t}^\top)_{s=1..t-1}$ the $(t-1) \times d$ matrix containing the empirical means of the selected actions, where the s-th row corresponds to the estimator at step t of the action selected at step s.
- $Y_{t-1} = (r_{i_s,s})_{s=1..t-1}^\top$ the rewards vector of size $t-1$.
- $A_{t-1} = diag(1/R_{i_s,t})_{s=1..t-1}$ the diagonal $(t-1) \times (t-1)$ matrix, where the s-th diagonal element equals $1/R_{i_s,t}$. Note that, for a specific action, the value of its corresponding coefficient increases with the number of observed samples for this action.

With these notations, the linear application from profiles to rewards can be written as:

$$Y_{t-1} = X_{t-1}\beta + \eta'_{t-1} \tag{7}$$

Proposition 2 *We note $\hat{\beta}_{t-1}$ the least square estimator of the parameter β at step t, according to the following l^2-regularized regression problem, where each element is weighted*

by the corresponding coefficient $1/R_{i_s,t}$*:*

$$\hat{\beta}_{t-1} = \arg\min_{\beta} \sum_{s=1}^{t-1} \frac{1}{R_{i_s,t}} (\beta^\top \hat{x}_{i_s,t} - r_{i_s,s})^2 + \lambda ||\beta||^2 \tag{8}$$

where $\lambda > 0$ *is the l2-regularization constant.*

We have:

$$\hat{\beta}_{t-1} = (X_{t-1}^\top A_{t-1} X_{t-1} + \lambda I)^{-1} X_{t-1}^\top A_{t-1} Y_{t-1} \tag{9}$$

Proof Let us rewrite the minimization problem such as:
$\hat{\beta}_{t-1} = \arg\min_{\beta} L$ with $L = (Y_{t-1} - X_{t-1}\beta)^\top A_{t-1}(Y_{t-1} - X_{t-1}\beta) + \lambda \beta^\top \beta$.
The gradient is given by:
$\nabla_\beta L = -2X_{t-1}^\top A_{t-1}(Y_{t-1} - X_{t-1}\beta) + 2\lambda\beta = 2(X_{t-1}^\top A_{t-1} X_{t-1} + \lambda I)\beta - 2X_{t-1}^\top A_{t-1} Y_{t-1}$
By canceling this gradient, we get the announced result. ■

This estimator of β uses empirical means of observed samples as inputs. Weighting each element according to the corresponding value $R_{i_s,t}$ allows one to consider the uncertainty associated with this approximation. It renders the confidence we have in the weighted input. Note that this coefficient tends towards a constant when the number of observed samples increases for the corresponding action. It allows one, according to the following proposition, to define a confidence ellipsoid for the estimator of β.

Proposition 3 *Let us define* $V_{t-1} = \lambda I + X_{t-1}^\top A_{t-1} X_{t-1} = \lambda I + \sum_{s=1}^{t-1} \frac{\hat{x}_{i_s,t} \hat{x}_{i_s,t}^\top}{R_{i_s,t}}$*. With the same assumptions as in proposition 1, for any* $0 < \delta < 1$*, with a probability at least equal to* $1 - \delta$*, the estimator* $\hat{\beta}_{t-1}$ *verifies for all* $t \geq 0$*:*

$$||\hat{\beta}_{t-1} - \beta||_{V_{t-1}} \leq \sqrt{2 \log\left(\frac{det(V_{t-1})^{1/2} det(\lambda I)^{-1/2}}{\delta}\right)} + \sqrt{\lambda} S = \alpha_{t-1} \tag{10}$$

where $||x||_V = \sqrt{x^\top V x}$ *is the V-norm of the vector* x*.*

Proof Available in appendix A.2. ■

This bound is very similar to the one defined in the `OFUL` algorithm (Abbasi-Yadkori et al., 2011) to build its confidence ellipsoid. However, a notable difference lies in the definition of the matrix V_t, in which weights in A_t are applied to cope with confidence differences between profile estimators. Without this weighting, no confidence ellipsoid could be found for β since no common bound could be defined for the various noises η_s' (see the proof of proposition 3 in appendix).

The following proposition can easily be deduced from the previous one to bound the expectation of reward with known profiles.

Proposition 4 *For every* $i \in \mathcal{K}$*, with probability greater than* $1 - \delta$*, we have for all* $t \geq 0$*:*

$$\beta^\top \mu_i \leq \hat{\beta}_{t-1}^\top \mu_i + \alpha_{t-1} ||\mu_i||_{V_{t-1}^{-1}} \tag{11}$$

Proof Available in appendix A.3 ■

This upper-bound for the expected reward contains two distinct terms: while the former corresponds to a classical exploitation term which estimates the expected reward with the current parameters, the latter corresponds to an exploration term since it takes into account the uncertainty on the reward parameter. If profiles were known, this could directly be used as a selection score for an UCB-like policy. However, in our setting, profiles are unknown. We have to consider confidence ellipsoids for the profiles of actions too. The following proposition defines confidence bounds for the profile estimators.

Proposition 5 *For every* $i \in \mathcal{K}$ *and any* $t > 0$*, with probability greater than* $1 - \delta/t^2$*, we have:*

$$||\hat{x}_{i,t} - \mu_i|| \leq \min(L \sqrt{\frac{2d}{n_{i,t}} \log\left(\frac{2dt^2}{\delta}\right)}, 2L) = \rho_{i,t,\delta} \tag{12}$$

Proof This inequality comes from the application of the Hoeffding's inequality to each dimension separately. The min operator comes from the base hypothesis $||x_{i,t}|| \leq L$, which can be more restrictive than the Hoeffding assumption. The proof is available in appendix A.4. ■

Contrary to the bound of the deviation of the mapping parameter β which holds simultaneously for all steps of the process, the one for the profile estimators is only valid for each step separately. To obtain a bound holding for every step simultaneously, which is important for the regret analysis (see section 3.3), we use the uniform bound principle. For a given action i, we have: $\mathbb{P}(\forall t, ||\hat{x}_{i,t} - \mu_i|| \leq \rho_{i,t,\delta}) = 1 - \mathbb{P}(\exists t, ||\hat{x}_{i,t} - \mu_i|| \geq \rho_{i,t,\delta}) \geq 1 - \sum_t \mathbb{P}(||\hat{x}_{i,t} - \mu_i|| \geq \rho_{i,t,\delta}) \geq 1 - \sum_t \delta/t^2$. This justifies the introduction of the t^2 term in the bound, which allows one to define a uniform probability over all steps since we have thereby: $\mathbb{P}(\forall t, ||\hat{x}_{i,t} - \mu_i|| \leq \rho_{i,t,\delta}) \geq 1 - \delta - \sum_{t=2}^{\infty} \delta/t^2 = 1 - \delta - \delta(\pi^2/6 - 1) \geq 1 - 2\delta$.

Now that we have defined probabilistic deviation bounds for the different estimators, we can use them conjointly to define the confidence interval of the reward expectation for the setting of unknown profiles, and thus to upper bound the expected reward for each action i.

Proposition 6 *For every* $i \in \mathcal{K}$ *and any* $t > 0$*, with probability greater than* $1 - \delta/t^2 - \delta$*, we have:*

$$\beta^\top \mu_i \leq \hat{\beta}_{t-1}^\top (\hat{x}_{i,t} + \overline{\epsilon}_{i,t}) + \alpha_{t-1} ||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \tag{13}$$

with: $\overline{\epsilon}_{i,t} = \frac{\rho_{i,t,\delta} \hat{\beta}_{t-1}}{||\hat{\beta}_{t-1}||}$ $\tilde{\epsilon}_{i,t} = \frac{\rho_{i,t,\delta} \hat{x}_{i,t}}{\sqrt{\lambda} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}}}$

Proof The proof is available in appendix A.5. ■

Compared to the bound given in proposition 4, we find the same two terms of exploitation and exploration. However in this case, profile vectors that are unknown are replaced by the estimator plus an additional term: $\bar{\epsilon}_{i,t}$ in the former part and $\tilde{\epsilon}_{i,t}$ in the latter one. These terms aim at coping with profile uncertainty, considering confidence ellipsoids for these profiles as defined in proposition 5. $\bar{\epsilon}_{i,t}$ is collinear with $\hat{\beta}_{t-1}$. It is used to translate the estimator $\hat{x}_{i,t}$ so that $\beta^\top \mu_i$ is upper-bounded. $\tilde{\epsilon}_{i,t}$ is collinear with $\hat{x}_{i,t}$. It is used to translate the estimator $\hat{x}_{i,t}$ so that the V_{t-1}^{-1}-norm $||\mu_i||_{V_{t-1}^{-1}}$ is upper-bounded. This bound enables us to derive an optimistic policy in the next section.

3.2. SampLinUCB

In this section, we detail our policy for the setting of unknown profiles, called *SampLinUCB*, which is directly derived from the bound proposed in proposition 6. Its process is detailed in algorithm 1. In words, it proceeds as follows:

1. Initialization of the shared variables V and b used to estimate the mapping parameter β in lines 1 and 2. The $d \times d$ matrix V is initialized with an identity matrix times the regularization parameter λ (the greater λ is, the more the parameter β will be constrained to have components close to zero). The vector b is initialized as a null vector of size d.

2. Initialization of the individual variables n_i, $\hat{x}_i$, R_i, N_i and S_i for every action in $\mathcal{K}$ (lines 3 to 6). The two latter are additional scalar variables which enable efficient updates for the shared variables after context observations. N_i counts the number of times an action has been selected from the beginning, S_i sums the rewards collected by selecting i from the beginning.

3. At each iteration t, for each action $i \in \mathcal{O}_t$, observation of the sample $x_{i,t}$ (line 11) and update of individual variables n_i, $\hat{x}_i$ and R_i for action i (line 12) and shared parameters V and b according to these new individual values for i (line 10 and 13). Since shared parameters are simple sums of elements, they can be simply updated by first removing old values (line 10) and then adding the new ones when updated (line 13). This is efficiently done without requiring an important memory load thanks to scalar variables N_i and S_i.

4. Computation of the selection score $s_{i,t}$ (line 21) for each action i according to equation 14 detailed below, and selection of the action associated with the highest selection score (line 23) (except in the early stages $\leq K$ where all actions are selected in turn to initialize their counts in line 16).

5. Collection of the associated reward (line 25) and update of variables N_i, S_i, V and b according to this new outcome (lines 26 to 28).

Algorithm 1: `SampLinUCB`

1 $V = \lambda I_{d\times d}$ (Identity matrix of size d);
2 $b = 0_d$ (Null vector of size d);
3 **for** $i \in \mathcal{K}$ **do**
4 $N_i = 0$; $S_i = 0$;
5 $n_i = 0$; $\hat{x}_i = 0_d$; $R_i = +\infty$;
6 **end**
7 **for** $t = 1..T$ **do**
8 Reception of $\mathcal{O}_t$;
9 **for** $i \in \mathcal{O}_t$ **do**
10 $V = V - N_i \frac{\hat{x}_i \hat{x}_i^\top}{R_i}$; $b = b - S_i \frac{\hat{x}_i}{R_i}$;
11 Observation of $x_{i,t}$;
12 $n_i = n_i + 1$; $\hat{x}_i = \frac{(n_i - 1)\hat{x}_i + x_{i,t}}{n_i}$; $R_i = \sqrt{R^2 + \frac{L^2 S^2}{n_i}}$;
13 $V = V + N_i \frac{\hat{x}_i \hat{x}_i^\top}{R_i}$; $b = b + S_i \frac{\hat{x}_i}{R_i}$;
14 **end**
15 **if** $t \leq K$ **then**
16 Selection of $i_t = t$;
17 **end**
18 **else**
19 $\hat{\beta} = V^{-1} b$;
20 **for** $i \in \mathcal{K}$ **do**
21 Computation of $s_{i,t}$ according to formula 14 ;
22 **end**
23 Selection of $i_t = \arg\max_{i \in \mathcal{K}} \; s_{i,t}$;
24 **end**
25 Reception of $r_{i_t,t}$;
26 $N_{i_t} = N_{i_t} + 1$;
27 $S_{i_t} = S_{i_t} + r_{i_t,t}$;
28 $V = V + \frac{\hat{x}_{i_t} \hat{x}_{i_t}^\top}{R_{i_t}}$; $b = b + r_{i_t,t} \frac{\hat{x}_{i_t}}{R_{i_t}}$;
29 **end**

The selection score $s_{i,t}$ used in our policy for each action i at any step t is directly derived from proposition 6:

$$s_{i,t} = (\hat{x}_{i,t} + \bar{\epsilon}_{i,t})^\top \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \tag{14}$$

For cases 2 and 3 of the profile delivery mechanism (see at the beginning of the section), there are actions i with $n_{i,t} = 0$ in the early steps of the process. No sample has ever been observed for these actions, which is problematic for the computation of $\rho_{i,t,\delta}$, and therefore

the computation of $\bar{\epsilon}_{i,t}$ and $\tilde{\epsilon}_{i,t}$. For the case 2, where we are not active on the process for observing contexts, this can be solved by simply ignoring actions until at least one sample of profile has been observed for them. For the case 3 however, samples are only obtained by selection. Thus, we need to force the observation of a sample for every action in the first steps. In that way, for actions with $n_{i,t} = 0$ at any step t, we arbitrarily set $s_{i,t} = +\infty$ in order to make the policy favor actions without any knowledge to initialize the process. Thus, in that case, algorithm 1 selects the K actions in turn in the K first steps of the process.

The selection score defined in formula 14 corresponds to the upper-bound of the expected reward for each action, as it is done in all UCB-based policies. Intuitively, it leads the algorithm to select actions whose profile estimator is either in an area with high potential, or is sufficiently uncertain to consider still likely that the action can be potentially useful. The goal is to quickly rule out bad actions, whose confidence ellipsoid does not include any potentially useful locations w.r.t. the current estimation of β. To better analyze the algorithm, we propose below a new formulation of the selection score.

Proposition 7 *The score $s_{i,t}$ from equation 14 can be re-written in the following way:*

$$s_{i,t} = \hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + \rho_{i,t,\delta} \left(||\hat{\beta}_{t-1}|| + \frac{\alpha_{t-1}}{\sqrt{\lambda}} \right) \tag{15}$$

Proof

$$\begin{aligned}
s_{i,t} &= (\hat{x}_{i,t} + \bar{\epsilon}_{i,t})^\top \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \\
&= \hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \frac{\rho_{i,t,\delta} \hat{\beta}_{t-1}^\top}{||\hat{\beta}_{t-1}||} \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t} + \frac{\rho_{i,t,\delta} \hat{x}_{i,t}}{\sqrt{\lambda} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}}}||_{V_{t-1}^{-1}} \\
&= \hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \rho_{i,t,\delta} ||\hat{\beta}_{t-1}|| + \alpha_{t-1} \left(1 + \frac{\rho_{i,t,\delta}}{\sqrt{\lambda} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}}} \right) ||\hat{x}_{i,t}||_{V_{t-1}^{-1}} \\
&= \hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + \rho_{i,t,\delta} \left(||\hat{\beta}_{t-1}|| + \frac{\alpha_{t-1}}{\sqrt{\lambda}} \right)
\end{aligned}$$

■

This new formulation of the selection score allows one to take a different look at the algorithm behavior. The first part of the score $\hat{x}_{i,t}^\top \hat{\beta}_{t-1} + \alpha_{t-1} ||\hat{x}_{i,t}||_{V_{t-1}^{-1}}$ is similar to a score that would use the classical `OFUL` algorithm (although with a different construction of V_t), with an exploitation term and a classical exploration term considering the uncertainty on the estimator of β. But it exhibits an additional part $\rho_{i,t,\delta} \left(||\hat{\beta}_{t-1}|| + \frac{\alpha_{t-1}}{\sqrt{\lambda}} \right)$ which is directly proportional to the coefficient $\rho_{i,t,\delta}$ and thus enables some exploration w.r.t. the uncertainty of the profile estimators. This highlights the premium granted to less observed actions. Note that for the case 1, this additional part is the same for every action. It therefore could be removed from the score since it does not permit to discriminate some action w.r.t any other one. However, this new exploration term is particularly useful for the

case 3, where observations of samples are directly connected to the selection policy, since it prevents from moving aside some optimal actions that have unluckily provided only not promising samples in the early steps.

To demonstrate that considering uncertainty on profiles is crucial in that case, let us consider a scenario where the optimal action $i^\star$ gets a null vector as the first profile sample. Then, in a setting where all profile samples are in $[0, L]^d$ and all rewards are in $[0, +\infty]$, it suffices that a sub-optimal action i gets a non-null vector as the first profile sample and a positive value as the first reward to lead to a linear regret from a given step. Indeed, since we only get samples with all components greater or equal than 0, i will never get a null vector as a profile estimator. On the other hand, while $i^\star$ is not selected, its profile estimator cannot change from the null vector. Thus, with a naive algorithm that would not include translations w.r.t. uncertainty of profiles, we would have $s_{i^\star,t} = \hat{x}_{i^\star,t}^\top \hat{\beta}_{t-1} + \alpha ||\hat{x}_{i^\star,t}||_{V_{t-1}^{-1}} = 0$ for all t until $i_t = i^\star$. Now, the least square estimator of β approximates observed reward values from estimated profiles. Since we have at least one non-null reward associated with a non-null profile estimator, β will always output a positive expected reward at least for one action. Thus, there is always an action i' with $s_{i',t} > 0$, which prevents from selecting the optimal action until the end of the process. This shows that a naive algorithm is not well fitted here, since it is likely to stay stuck on sub-optimal actions because of wrong knowledge about profiles. The point is now to show that the proposed additional term enables to solve this problem and ensures a sub-linear pseudo regret for our profile-based bandit algorithm with unknown profiles.

3.3. Regret

The following proposition establishes an upper bound for the cumulative pseudo-regret of the `SamplinUCB` algorithm proposed above. This is a generic bound for which no assumption is done on the process generating $\mathcal{O}_t$ at each step t.

Proposition 8 (Generic bound) *By choosing $\lambda \geq \max(1, L^2/\sqrt{R^2})$, with a probability greater than $1 - 3\delta$, the cumulative pseudo-regret of the algorithm `SampLinUCB` is upper-bounded by:*

$$\begin{aligned}
\hat{R}_T \leq\ & C + 4L \left(\sqrt{\frac{d}{\lambda} \log \left(\frac{1 + TL^2/\lambda}{\delta} \right)} + 2S \right) \sqrt{2d \log \left(\frac{2dT^2}{\delta} \right)} \sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} \\
& + 2 \left(\sqrt{d \log \left(\frac{1 + TL^2/\lambda}{\delta} \right)} + \sqrt{\lambda} S \right) \\
& \times \sqrt{Td \left(\sqrt{R^2 + L^2 S^2} \log \left(1 + \frac{TL^2}{\lambda d} \right) + \frac{4L^2}{\lambda} \log \left(\frac{2dT}{\delta} \right) \sum_{t=1}^{T} \frac{1}{n_{i_t,t}} \right)}
\end{aligned} \tag{16}$$

Proof Available in appendix A.6. ■

The study of dominant factors of the bound given above enables to obtain the following proposition for the three considered settings of the context delivery process, where we removed dependencies on L, λ, R and S to simplify the notations.

Proposition 9 (Bounds for the three profile delivery settings) *For each of the three considered settings for the profile delivery process, the upper bound of the cumulative pseudo-regret is*[2]*:*

- *For the case 1, with a probability greater than* $1-3\delta$*:*

$$\hat{R}_T = \mathcal{O}\left(d\log\left(\frac{T}{\delta}\right)\sqrt{T\log(T)}\right) \tag{17}$$

- *For the case 2, with a probability greater than* $(1-3\delta)(1-\delta)$*, and for* $T \geq 2\log(1/\delta)/p^2$*:*

$$\hat{R}_T = \mathcal{O}\left(d\log\left(\frac{T}{\delta}\right)\sqrt{T\frac{\log(T)}{p}}\right) \tag{18}$$

  where p *is the probability of profile delivery for any action at each step.*

- *For the case 3, with a probability greater than* $1-3\delta$*:*

$$\hat{R}_T = \mathcal{O}\left(d\log\left(\frac{T}{\delta}\right)\sqrt{TK\log(\frac{T}{K})}\right) \tag{19}$$

Proof The proofs for these three bounds are respectively given in appendix A.7.1, A.7.2 and A.7.3. ■

Thus, in every setting our `SamplinUCB` algorithm ensures a sub-linear upper bound for its cumulative pseudo-regret. The bound given for case 2 owns an additional dependency in p, the probability of context delivery for each action at each step. Obviously, the higher this probability is, the faster the uncertainty about profiles decreases. Note that this bound for case 2 is only valid from a given number of iterations inversely proportional to p^2, since it requires a minimal number of observations to hold. The bound for case 3 owns a dependency in the number of available actions K. This comes from the fact that only the selected action reveals its profile at each step, which re-introduces the need of considering each action a minimal number of times, as it is the case with traditional stationary approaches such as the classical `UCB` algorithm. However, as we show in our experiments below, the use of the structure of the actions, which enables some common learning of reward distributions, leads to greatly better results than existing stationary algorithms in various cases.

2. $\mathcal{O}$ renders the relation "dominated by", which means that $f = \mathcal{O}(g)$ implies that there exists a strictly positive constant C such that asymptotically we have: $|f| \leq C|g|$.

3.4. Extension to the multiple-plays setting

This short section extends our algorithm for the multiple-plays setting, where multiple actions are chosen at each step. Rather than only selecting a single action i_t at any step t, the algorithm has now to select a set $\mathcal{K}_t \subseteq \mathcal{K}$ of $k \geq 1$ actions for which it gets rewards. Algorithm 1 is therefore adapted to this setting, by simply selecting the k best actions at each step (those that get the best selection scores w.r.t. formula 14) rather than only the best one (line 23 of the algorithm). The aim is still to maximize the cumulative reward through time, where all rewards at any step t are simply summed to form the collected reward at step t (although other Lipschitz functions could have been considered for the collective reward construction from the k individual ones, such as proposed in Chen et al. (2013)).

Definition 1 *The cumulative pseudo-regret of our setting of bandit with multiple plays is defined as:*

$$\hat{R}_T = T\sum_{i\in\mathcal{K}^\star}\mu_i^\top\beta - \sum_{t=1}^{\top}\sum_{i\in\mathcal{K}_t}\mu_i^\top\beta \tag{20}$$

with $\mathcal{K}^\star$ *the set of* k *optimal actions, i.e. the* k *actions with the highest values* $\mu_i^\top\beta$*.*

Proposition 10 (Generic bound for the multiple-plays setting) *By choosing* $\lambda \geq \max(1, L^2/\sqrt{R^2})$*, with a probability greater than* $1-3\delta$*, the cumulative pseudo-regret for our* `SampLinUCB` *algorithm with multiple selections is upper bounded by:*

$$\begin{aligned}\hat{R}_T \leq\ & C + 4L\left(\sqrt{\frac{d}{\lambda}\log\left(\frac{1+TkL^2/\lambda}{\delta}\right)} + 2S\right)\sqrt{2d\log\left(\frac{2dT^2}{\delta}\right)}\sum_{t=1}^{T}\sum_{i\in\mathcal{K}_t}\frac{1}{\sqrt{n_{i,t}}} \\ & + 2\left(\sqrt{d\log\left(\frac{1+TkL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S\right) \\ & \times\sqrt{Td\left(\sqrt{R^2+L^2S^2}\log\left(1+\frac{TkL^2}{\lambda d}\right) + \frac{4L^2}{\lambda}\log\left(\frac{2dT}{\delta}\right)\sum_{t=1}^{T}\sum_{i\in\mathcal{K}_t}\frac{1}{n_{i,t}}\right)}\end{aligned} \tag{21}$$

Proof The proof is available in appendix A.8. ■

Equivalent bounds for the three cases of context delivery can be directly derived from this new generic bound by applying the same methods as in the previous section. This allows us to apply our algorithm for tasks where multiple actions can be triggered at each step, such as in the data capture task considered in our experiments in section 4.2.

4. Experiments

This section is divided in two parts. First, we propose a series of experiments on artificial data in order to observe the behavior of our approach in well-controlled environments. Then, we give results obtained on real-world data, for a task of data capture from social media.

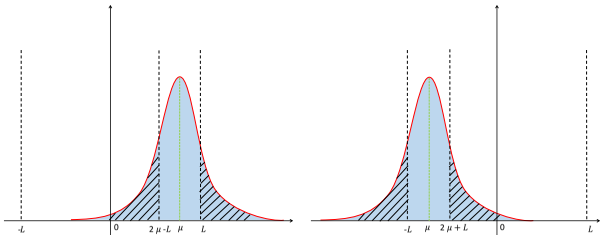
4.1. Artifical Data

4.1.1. Protocol

Data Generation: In order to assess the performances of the `SamplinUCB` algorithm, we propose to first experiment it in a context of simple selection ($k = 1$) on artificially generated data. For that purpose, we set the horizon T to 30000 iterations, the number of available actions K to 100 and the size of the profile space to $d = 5$ dimensions. Then, we sampled a mapping vector β randomly in $\left[-S/\sqrt{d}..S/\sqrt{d}\right]^d$, in order to fulfill the $||\beta|| \leq S = 1$ condition. For each arm i, we then sampled a random vector μ_i uniformly in $\left[-L/\sqrt{d}..L/\sqrt{d}\right]^d$ with $L = 1$. Finally, for each iteration $t \in \{1, ..., T\}$, we proceeded as follows to generate simulated data:

1. For each action $i \in \{1, ..., K\}$, we sampled a vector $x_{i,t}$ from the multivariate Gaussian $\mathcal{N}(\mu_i, \sigma^2 I)$. Note that, in order to assess the influence of profile samples variations on the performances of `SampLinUCB`, we tested different values for $\sigma \in \{0.5, 1.0, 2.0\}$. Moreover, in order to guarantee that $||x_{i,t}|| \leq L = 1$, while still getting sampled centered on μ_i, the Gaussian is truncated symmetrically around μ_i. This is illustrated by figure 3 for $d = 1$, where hatched areas correspond to excluded values. On the left is given the case with $\mu_i > 0$ and on the right the case with $\mu_i < 0$;

2. For each action $i \in \{1, ..., K\}$, we sampled a reward $r_{i,t}$ from a Gaussian with mean $\mu_i^\top \beta$ and variance $R^2 = 1$;



(a) Case $\mu > 0$ (b) Case $\mu < 0$

Figure 3: Profile Samples Generation Process: Truncated Gaussians

To emhasize the need of exploration on profiles, note that we set to null vectors the profile samples of the 100 first steps of each dataset. 100 datasets have been generated in such a way. The results given below correspond to averages on these artificial datasets.

Experimented Policies: We propose to compare `SampLinUCB` to the following bandit policies:

- `UCB`: the well-known `UCB` approach (Auer et al., 2002), which selects at each step the action with the best upper-confidence bound, estimated w.r.t. past rewards of actions, without any assumption about some latent structure of the actions;
- `UCBV`: the `UCBV` algorithm (Audibert et al., 2009) is an extension of `UCB`, where the variance of rewards for each action is included in the selection scores to lead the policy to ground their estimations in an higher number of observations for noisier actions.
- `UCB`$-\delta$: the `UCB`$-\delta$ algorithm (Abbasi-Yadkori et al., 2011) is a variant of `UCB` where optimism is ensured via a concentration inequality based on auto-normalised process (de la Peña et al., 2009), with a confidence level of $1 - \delta$. In our experiments, we set $\delta = 0.05$;
- `Thompson`: the `Thompson Sampling` algorithm (Thompson, 1933) introduces randomness in the exploration process by sampling reward expectations from their posterior at each time-step, following a Gaussian assumption of the rewards;
- `MOSS`: a variant of `UCB`, which usually obtains better results than the classical `UCB` but requires the knowledge about the horizon T (Audibert and Bubeck, 2009);

None of these approaches use any side information. Therefore, the only noise they have to deal with comes from the variance R^2 of the Gaussian distributions of rewards. For `SampLinUCB` an additional difficulty comes from the variations of the observed samples of profiles. The point is therefore to know whether these samples can be leveraged to exhibit some structure of the actions, that can benefit to stationary bandit tasks, despite such variations. Additionaly, the following two contextual baselines are considered in our experiments to analyze the performances of our approach:

- `LinUCB`: the very famous contextual approach that assumes a linear mapping between observed contexts and rewards (Li et al., 2010). In our case, observed profile samples correspond to the contexts that `LinUCB` takes into account in its selection policy. We consider this baseline in the interesting setting where contexts are only delivered for the selected arms (case 3 described above). In this setting, non selected arms deliver null context vectors for the next step;
- `MeanLinUCB`: this baseline corresponds to our approach but without the exploration term w.r.t. the profiles. Empirical means are considered as true profiles at each step of the process (this comes down to set $\rho_{i,t,\delta}$ to 0 for every arm and every step). As discussed above (see the last paragraph of section 3.2), such a baseline cannot guarantee a sub-linear regret since it can infinitely stay stuck on sub-optimal arms, but an empirical evaluation of its performances is useful to understand the benefits of the proposed approach.

To analyze the performances of `SampLinUCB`, we implement the three scenarios studied in previous sections. In the following, our approach is denoted `SampLinUCB_p=<p>`, where p

corresponds to the probability for any action to get a sample of its profile at every iteration. Different values for p are considered: $p \in \{0, 0.005, 0.01, 1\}$. Note that $p = 1$ corresponds to the case 1, while $p = 0$ refers to the case 3. In this latter instance, as considered in the previous sections, the samples delivery process is replaced by the ability to observe samples for the selected actions at each iteration. Note also that, for clarity and analysis purposes, in instances with $p > 0$ we do not observe samples for the selected actions (which exactly follows the cases studied in the previous section)[3]. In every instance, we set $\delta = 0.05$ for these experiments. Also, to avoid a too large exploration on profiles in the early steps, we multiplied each $\rho_{i,t,\delta}$ by a 0.01 coefficient, which still guarantees a sub-linear regret in the limit.

4.1.2. Results

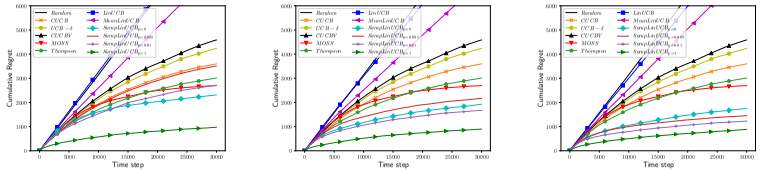
Figures 4(a), 4(b) and 4(c) report the evolution of the cumulative pseudo-regret through time for the tested policies, for σ values (variance of profile samples) respectively set to $\sigma = 2.0$, $\sigma = 1.0$ and $\sigma = 0.5$. Note that the curves of `UCB`, `UCB-`δ, `UCBV`, `Thompson` and `MOSS` are identical in every plot since their performances do not depend on the profile samples. We first notice from these plots that `UCB-`δ and `UCBV` do not provide good results on these data. It appears that these two policies over-explore during the whole bandit process. `Thompson` and `MOSS` obtain better results in average, but still far from the best contextual approach `SampLinUCB`$_{p=1}$. This confirms that using profiles of arms can be greatly advantegeous when there exist a linear correlation between these profiles and the associated rewards. In this setting (which corresponds to the case 1 studied above), the profiles are discovered step by step, but since we get a sample for every arm at each iteration, the estimators quickly converge towards the true profiles. This explains the very good results for this easy setting, and why there is nearly no differences in the results of `SampLinUCB`$_{p=1}$ for the three considered sample variances.

Let us now focus on the results provided by our `SampLinUCB` algorithm when only a subset of arms gets profile samples at each step of the process. As expected, the more the algorithm observes samples, the better it performs. However, we remark that `SampLinUCB`$_{p=0}$ obtains better results than `SampLinUCB`$_{p=0.005}$ for $\sigma = 2$ and $\sigma = 1$, and even better than `SampLinUCB`$_{p=0.01}$ when $\sigma = 2$ (while observing the same rate of samples as in this latter setting). This denotes a stronger robustness to the profile sample variance. By dynamically selecting the arms to observe, it is able to focus on the improvement of useful estimators rather than getting as many samples but for randomly selected arms (and potentially for arms that could be quickly discarded). In this interesting setting, `SampLinUCB` always outperforms non-contextual approaches for the studied sample variances, while we note a significant improvement of the results when the variance is low.

At last, we can note the very weak - near random - results obtained by `LinUCB`, which directly bases its strategy on the observed samples. More interesting are the weak results obtained by `MeanLinUCB`, which exhibits a linear regret. This emphasizes the crucial role of the profile exploration term of `SampLinUCB`: While `SampLinUCB`$_{p=0}$ is able to reconsider

3. Note that we could easily imagine tasks, which correspond to some mix of cases 2 and 3, where we both get samples from an external process and for the selected actions. For such cases, we can reasonably assume better results than those reported below for cases 2 and 3, since the process would benefit from both sample sources.

bad profiles observed in the early steps of the process, `MeanLinUCB` usually stays stuck on the first actions that provided a non-nul sample associated with a positive reward. If they are lucky, `LinUCB` and `MeanLinUCB` can exhibit good performances on some instances, but they are clearly not well fitted for the bandit setting considered in this paper.



(a) $\sigma = 2$ (b) $\sigma = 1$ (c) $\sigma = 0.5$

Figure 4: Cumulative pseudo-regret through time on artificial data with different settings for the profile samples delivery process (from very noisy samples on the left, to samples with low variance on the right).

To conclude, it appears from these first experiments that our `SampLinUCB` algorithm, while dealing with a doubled uncertainty (both on the mapping parameters and the profile estimators), is able to leverage the latent structure that it discovers through time from noisy samples of the profiles.

4.2. Real World Experiments: Social Data Capture

In this section we propose to apply our `SampLinUCB` algorithm to the task of dynamic data capture from Twitter introduced in (Gisselbrecht et al., 2015). According to a given information need, the aim is to collect relevant data from the streaming API proposed by Twitter. This API provides messages published by users on the social network in real-time. In this setting, each user account is associated with a stream of messages that can be monitored. However, for various reasons (notably w.r.t. constraints set by the social media), it is not possible to collect the whole activity of the social media. Only streams of messages published by a subset of k users can be monitored simultaneously ($k << K$). The aim is therefore to focus on users that are the most likely to publish messages that fit with the data need. The difficulty is that we do not know anything about the users beforehand, everything must be discovered during the capture process. We have thus to deal with an exploitation/exploration problem that suits well with the bandit setting studied in this paper.

Given a time period divided in T steps, the agent has to select, at each iteration $t \in \{1, ..., T\}$ of the process, a subset $\mathcal{K}_t$ of k user accounts to follow, among the whole set of possible users $\mathcal{K}$ ($\mathcal{K}_t \subseteq \mathcal{K}$). Given a relevance score $r_{i,t}$ assigned to the content posted by user $i \in \mathcal{K}_t$ during iteration t of the process (the set of tweets he posted during iteration t), the aim is to select at each iteration t the set of user accounts that maximize the sum of

collected scores:

$$\max_{(\mathcal{K}_t)_{t=1..T}} \sum_{t=1}^{T} \sum_{i \in \mathcal{K}_t} r_{i,t} \tag{22}$$

4.2.1. Rewards

In our experiments, we attempt to focus on users that have a great impact on some specified thematic. The *Follow Streaming* API of Twitter provides in real-time not only tweets posted by the followed users, but also all the re-tweets and replies to these users other users post on the network. Our reward function takes all of these messages into account to provide a reward score $r_{i,t}$ for each user $i \in \mathcal{K}_t$ after each capture period t:

$$r_{i,t} = tanh\left(\sum_{\omega \in \Omega_{i,t}} g_\gamma(\omega)\right) \tag{23}$$

where $\Omega_{i,t}$ contains the original messages from i, the re-tweets of messages from i and the replies to i during the period t, and g_γ is a function returning 1 if the content of the message as argument is judged as belonging to the desired thematic γ, 0 otherwise. To build this function g_γ, we trained a SVM topic classifier on the *20 Newsgroups* dataset (with TF bag of words representations of the texts, after stemming via the Porter Stemmer algorithm). We finally focus on 4 different topics γ: *Politics*, *Religion*, *Science* and *Sport*. Four different reward functions are therefore considered in the following (one for each topic).

4.2.2. Profiles

Following the setting of our profile based bandit, we assume that each user i of the social network is associated to an unknown vector μ_i corresponding to its profile. In these experiments, we assume that the profile of a user i corresponds to the mean of its content distribution: $\mu_i = \lim_{T\to\infty} \frac{1}{T} \sum_{t=1}^{T} x_{i,t}$, where $x_{i,t}$ is a given representation of the content posted by i during step $t-1$:

$$x_{i,t+1} = f(\sum_{\omega \in \Psi_{i,t}} \omega) \tag{24}$$

where $\Psi_{i,t} \subseteq \Omega_{i,t}$ contains all messages posted by i during step t and $\omega \in \mathbb{R}^m$ (with m the size of the vocabulary) is a TF bag of words representation of a message (after stemming via the Porter Stemmer algorithm). The function f aims at reducing the dimension of the representations, since the dimension d of the profile samples is the main factor of complexity in our algorithm, due to the required $d \times d$ matrix inversions. In order to reduce the dimension of profiles, we used a *Latent Dirichlet Allocation* method specifically designed for short texts (Hong and Davison, 2010), which aims at modeling texts as a mixture of topics. We set the number of topics to $d = 30$ and learned the LDA model on a preliminary 3-days random capture from Twitter.

4.2.3. Datasets

In order to be able to test different policies and simulate a real time decision process several times, we propose to conduct our experiments on offline datasets:

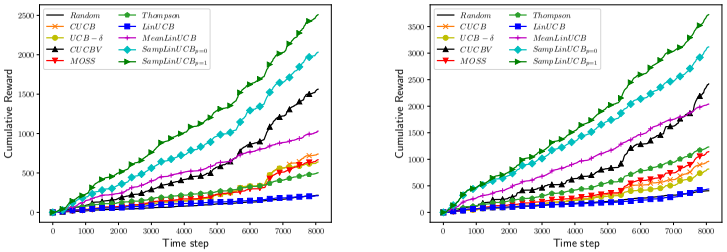
- *USElections*: dataset containing 3 587 961 messages produced by 5000 users during the ten days preceding the US presidential elections in 2012. The 5000 chosen accounts are the first ones who used either "Obama", "Romney" or "#USElections" from a preliminary random capture on Twitter.
- *OlympicGames*: dataset containing 15 010 322 messages produced by 5000 users in August 2016 during a period of three weeks covering the Olympic Games of Rio. The 5000 chosen accounts are the ones that were observed to use the most many hashtags "#Rio2016", "#Olympics", "#Olympics2016" or "#Olympicgames" within a period of preliminary random capture of three days before the Olympic Games.
- *Brexit*: dataset containing 2 118 235 messages produced by 5000 users during the first week of October 2016. The 5000 chosen accounts are the first ones who used "#Brexit" from a preliminary random capture from Twitter.

4.2.4. Results

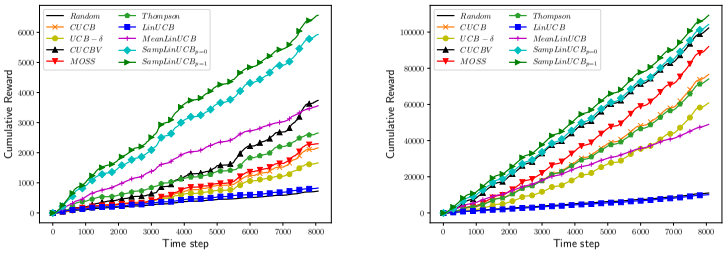
As done in (Gisselbrecht et al., 2015), we set k, the number of listened users at each time step, to 100, and the size of an iteration to 100 seconds. In these experiments, we assume $L = S = R = 1$ and we set $\delta = 0.05$ as done with artificial data.

Figures 5, 6 and 7 give the evolution of the cumulative reward through time for the datasets *USElections*, *OlympicGames* and *Brexit* respectively. In every case, we consider the four reward functions corresponding to the four topics *Politic*, *Religion*, *Science* and *Sport*. In order to lighten the plots, we only give in these figures the results of `SamplinUCB` for $p = 0$ and $p = 1$. In every plot, our algorithm is compared to the same baselines as described in section 4.1, where the policies are extended for the multiple-plays setting (as done in (Gisselbrecht et al., 2015)).

A first important observation from these plots is that in every setting, our algorithm `SampLinUCB` obtains better results than every other policy, even `CUCBV`, the extension of `UCBV` for the multiple-plays setting. Although `CUCBV` has demonstrated good performances for the task of social data capture (Gisselbrecht et al., 2015), where a high variance can be observed in the contents posted by users, the use of profiles associated to users of the networks enables an even more efficient exploration process. Globally, same manner as with artificial data, the performances of our approach increase with p, with a maximum reached when $p = 1$. Note however that the setting $p = 0$ (the case 3 studied above) is the most realistic one, since it does not use anything but the content collected by followed users at each step, which is the case in practice when collecting data from a social media such as Twitter. Interestingly, even for this setting the results obtained are always better than those of every compared approach. The improvement w.r.t. `CUCBV` is less significant for the `Sport` reward function for which greatly more rewards exist in the datasets (greatly more messages are categorized as sport), which allows non-contextual approaches to quickly



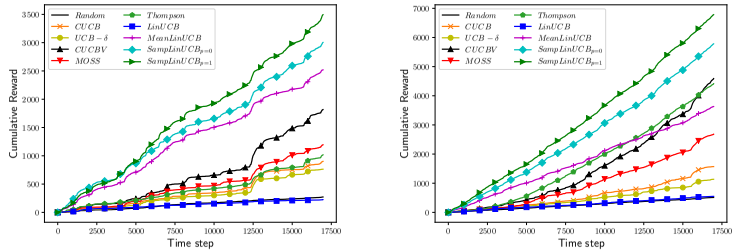
(a) Politic (b) Religion



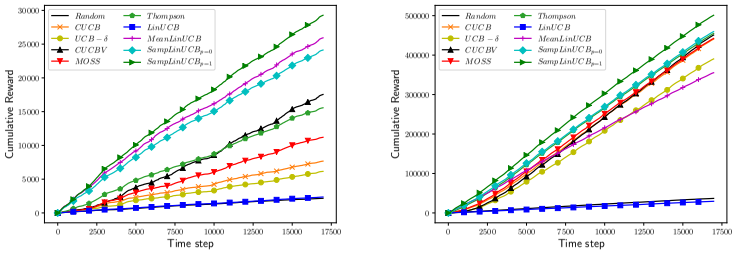
(c) Science (d) Sport

Figure 5: Evolution of the cumulative reward through time on the *USElections* dataset, according to the three considered reward functions *Politic*, *Religion*, *Science* and *Sport*.

collect knowledge about reward expectations of users. But for every other reward function `SampLinUCB`$_{p=0}$ always obtained results comprised between 1.5 and 3 times the ones obtained with the best non-contextual approach. Note also the crucial role of the exploration term for profile discovery ρ, since `MeanLinUCB`, which considers current empirical sample means as true profiles, always obtains greatly lower results than `SampLinUCB`$_{p=0}$ (except for the `Science` reward function on the `OlympicGames` and the `Brexit` datasets, where it benefits from good rewards and profile samples observed for some useful users in the initialization steps of the process). At last, as expected, `LinUCB`, which directly bases its selection policy on profile samples observed at the current step, obtains very bad results (near random). Since obtaining nul context vectors for every user not selected at the previous step, its selection mechanism very early focuses on a given set of users without ever reconsidering the others (except in the rare cases of context samples leading to negative reward expectations according to β). All these results highlight the interest of the proposed approach, based on confidence balls of the arm profiles, for tasks where contexts are only observed when the arms are selected .



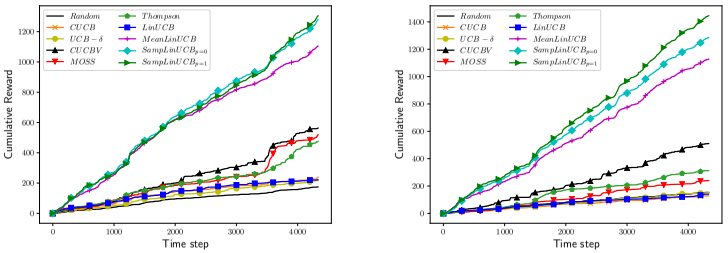
(a) Politic (b) Religion



(c) Science (d) Sport

Figure 6: Evolution of the cumulative reward through time on the *OlympicGames* dataset, according to the three considered reward functions *Politic*, *Religion*, *Science* and *Sport*.

Figures 8, 9 and 10 give the relative final cumulative rewards for different settings of the sample delivery process on the datasets *USElections*, *OlympicGames* et *Brexit* respectively (each score is normalized according to the score obtained when $p = 1$). Here we still observe that performances tend to decrease with p, for settings where $p > 0$. However it must be noticed that the setting $p = 0$ obtains results very close to other settings: it always obtains at least 80% of the final cumulative reward obtained when every user delivers a sample at each step of the process ($p = 1$). Better, in many cases `SampLinUCB`$_{p=0}$ succeeds in obtaining an higher final cumulative reward than $p = 0.01$ and $p = 0.02$. This is particularly true for the `Brexit` dataset where the dynamic selection of samples to be delivered appears very effective. On that dataset, `SampLinUCB`$_{p=0}$ even usually reaches the performances of `SampLinUCB`$_{p=0.05}$, while observing greatly less profile samples at each step (only 100 over 5000 at each iteration, which corresponds to the observation rate of the setting $p = 0.02$). While settings with $p > 0$ are greatly favored by the fact that they do not need to play an arm to get a sample of its profile, `SampLinUCB`$_{p=0}$ is not only active for the discovery of the mapping parameters, but also for the estimation of profiles. Its knowledge about profiles



(a) Politic (b) Religion

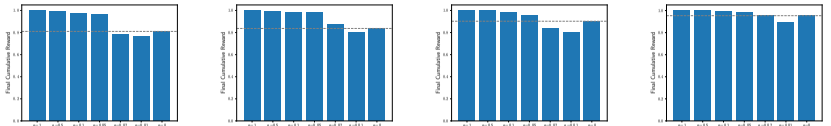
(c) Science (d) Sport

Figure 7: Evolution of the cumulative reward through time on the *Brexit* dataset, according to the three considered reward functions *Politic*, *Religion*, *Science* and *Sport*.

is directly connected to its selection strategy, with a selection score that favors promising actions with high uncertainty about their profile. This leads to an algorithm that efficiently deals with a trade-off between exploitation of good actions and exploration on both the mapping parameter and the profiles of actions.

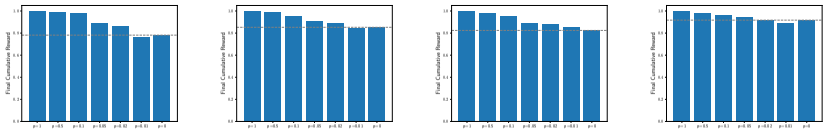
5. Conclusion

In this paper, we focused on structured stochastic bandits, where rewards depend on some constant profile associated with actions. More specifically, we introduced the case where the associated profiles are unknown beforehand, and must be discovered from samples delivered during the process. This setting implies a doubled uncertainty, both on profile estimators and on reward predictors, for which we designed a dedicated algorithm, named `SamplinUCB`, that seeks at leveraging the structure of the unknown profiles in its exploration process. Various settings for the profile samples delivery process have been considered, for which we gave theoretical convergence guarantees. Finally, experiments on both artificial data



(a) Politic (b) Religion (c) Science (d) Sport

Figure 8: Final normalized cumulative rewards for `SampLinUCB` on *USElections* with different p settings.



(a) Politic (b) Religion (c) Science (d) Sport

Figure 9: Final normalized cumulative rewards for `SampLinUCB` on *OlympicGames* with different p settings.

and a task of data capture from social networks demonstrate the very good behavior of the proposed approach. Ongoing works concern the inclusion of a non-stationary part in the selection strategy, where profiles may vary over time according to some evolving latent state of the actions.

Acknowledgments

This research work has been carried out in the framework of the Technological Research Institute SystemX, and therefore granted with public funds within the scope of the French Program "Investissements d'Avenir".

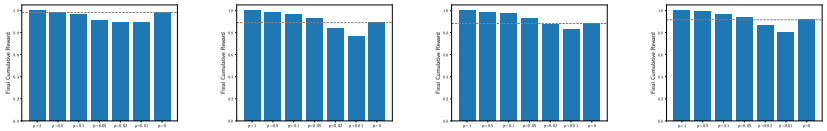
Appendix A. Appendix

A.1. Proof of proposition 1

The two following lemmas directly come from the definition of the sub-gaussian variables.

Lemma 1 *Let X be a random variable centered on 0. Then, X is said sub-gaussian with constant R if one of the two equivalent following conditions holds:*

- *Laplace Condition:* $\exists R > 0, \forall \lambda \in \mathbb{R}, \mathbb{E}[e^{\lambda X}] \leq e^{R^2\lambda^2/2}$



(a) Politic (b) Religion (c) Science (d) Sport

Figure 10: Final normalized cumulative rewards for `SampLinUCB` on *Brexit* with different p settings.

- *Sub-Gaussian Tail:* $\exists R > 0, \forall \gamma > 0, P(|X| \geq \gamma) \leq 2e^{-\gamma^2/(2R^2)}$

Lemma 2 *Let X_1 and X_2 be two sub-gaussian variables with respective constant R_1 and R_2. Let α_1 and α_2 be two real scalars. Then the variable $\alpha_1 X_1 + \alpha_2 X_2$ is sub-gaussian too, with constant $\sqrt{\alpha_1^2 R_1^2 + \alpha_2^2 R_2^2}$.*

The lemma 3 can be deduced from the application of the lemma 1 in the context of our specific problem of profile-bandit with unknown profiles, where the deviation of the profile estimators is a sub-gaussian variable.

Lemma 3 *Let us assume that for any i all samples $x_{i,t} \in \mathbb{R}^d$ observed for i at every step $t > 0$ are iid with mean $\mu_i \in \mathbb{R}^d$. Let us also assume that $||x_{i,t}|| \leq L$ for every i and t, and that $||\beta|| \leq S$. Then, for every i, and at each step t, $\beta^\top \epsilon_{i,t}$ is sub-gaussian with constant $\frac{LS}{\sqrt{n_{i,t}}}$ (with $\epsilon_{i,t} = \mu_i - \hat{x}_{i,t}$).*

Proof

By using the Cauchy-Schwarz inequality, for all i and at each step t we have: $|x_{i,t}^\top \beta| \leq ||\beta|| ||\hat{x}_{i,t}|| \leq LS$. Then, given that for all i, all samples $x_{i,t}$ are iid and $\mathbb{E}[x_{i,t}] = \mu_i$, we can apply the Hoeffding inequality to the random variable $\beta^\top \hat{x}_{i,t}$ with mean $\mu_i^\top \beta$:

$$\forall \gamma > 0, \mathbb{P}\left(|\beta^\top \hat{x}_{i,t} - \beta^\top \mu_i| > \gamma\right) = \mathbb{P}\left(|\beta^\top \epsilon_{i,t}| > \gamma\right) \leq 2e^{-\frac{n_{i,t}\gamma^2}{2S^2L^2}}$$

which allows us to say that $\epsilon_{i,t}^\top \beta$ is sub-gaussian with constant $\frac{LS}{\sqrt{n_{i,t}}}$. ■

We finally use the lemma 2 with the sum of $\beta^\top \epsilon_{i,t}$ and $\eta_{i,s}$ to prove the proposition 1, which establishes the random variable $\beta^\top \epsilon_{i,t} + \eta_{i,s}$ is conditionally sub-gaussian with constant $R_{i,t} = \sqrt{R^2 + \frac{L^2 S^2}{n_{i,t}}}$.

A.2. Proof of the proposition 3

To lighten notations, we removed the dependence on t in A and X. We have:

$$\begin{aligned}
\hat{\beta}_{t-1} &= \arg\min_\beta \sum_{s=1}^{t-1} \frac{1}{R_{i_s,t}} (\beta^\top \hat{x}_{i_s,t} - r_{i_s,s})^2 + \lambda ||\beta||^2 \\
&= (X^\top A X + \lambda I)^{-1} X^\top A Y \\
&= (X^\top A X + \lambda I)^{-1} X^\top A (X\beta + \eta') \\
&= (X^\top A X + \lambda I)^{-1} X^\top A \eta' + (X^\top A X + \lambda I)^{-1} (X^\top A X + \lambda I)\beta \\
&\quad - (X^\top A X + \lambda I)^{-1} \lambda I \beta \\
&= (X^\top A X + \lambda I)^{-1} X^\top A \eta' + \beta - \lambda (X^\top A X + \lambda I)^{-1} \beta
\end{aligned}$$

Then, the following main arguments of this proof come from the theory of auto-normalized process (de la Peña et al., 2009). By using a similar method to the one used in (Abbasi-Yadkori et al., 2011), we get:

$$||\hat{\beta}_{t-1} - \beta||_{V_{t-1}} \leq ||X^\top A \eta'||_{V_{t-1}^{-1}} + \lambda ||\beta||_{V_{t-1}^{-1}}$$

with $V_{t-1} = \lambda I + X^\top A X$, which is semi-definite positive since $\lambda > 0$. Since $||\beta|| \leq S$ and $||\beta||^2_{V_{t-1}^{-1}} \leq ||\beta||^2 / \lambda_{min}(V_{t-1}) \leq ||\beta||^2 / \lambda$, we get:

$$||\hat{\beta}_{t-1} - \beta||_{V_{t-1}} \leq ||X^\top A \eta'||_{V_{t-1}^{-1}} + \sqrt{\lambda} S$$

By using the proposition 1 of (Abbasi-Yadkori et al., 2011), and since we know from proposition 1 that $\frac{\eta_s'}{R_{i_s,t}}$ is sub-gaussian with constant 1, for any $\delta > 0$, with a probability of at least $1 - \delta$, for every $t \geq 0$ we have:

$$\begin{aligned}
||X^\top A \eta'||_{V_{t-1}^{-1}} &= ||\sum_{s=1}^{t-1} \frac{\eta_s'}{R_{i_s,t}} \hat{x}_{i_s,t}||_{V_{t-1}^{-1}} \\
&\leq \sqrt{2 \log\left(\frac{det(V_{t-1})^{1/2} det(\lambda I)^{-1/2}}{\delta}\right)} \\
&\leq \sqrt{d \log\left(\frac{1 + tL^2/\lambda}{\delta}\right)}
\end{aligned}$$

A.3. Proof of the proposition 4

Proof Let us assume that the inequality of the proposition 3 is valid. Therefore, we have for all $t > 0$ and every $i \in \mathcal{K}$:

$$\begin{aligned}\hat{\beta}_{t-1}^\top \mu_i + \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} - \beta^\top \mu_i =& (\hat{\beta}_{t-1} - \beta)^\top \mu_i + \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} \\ \geq & - ||\hat{\beta}_{t-1} - \beta||_{V_{t-1}}||\mu_i||_{V_{t-1}^{-1}} + \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} \\ \geq & - \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} + \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} \\ =& 0\end{aligned}$$

■

A.4. Proof of the proposition 5

With $||x_{i,t}|| \leq L$, we know that, for any $j \in [1..d]$, $|x_{i,t}^j| \leq L$. Thus, we can apply the Hoeffding inequality to each dimension of the profile estimators:

$$\forall \gamma > 0 : \ \mathbb{P}\left(|\hat{x}_{i,t}^j - \mu_i^j| > \gamma/\sqrt{d}\right) \leq 2e^{-\frac{n_{i,t}\gamma^2}{2L^2 d}}$$

Then, by using the fact that $||\hat{x}_{i,t} - \mu_i|| \leq \frac{1}{\sqrt{d}} \sum_{i=1}^{d} |\hat{x}_{i,t}^i - \mu_i^i|$ and the uniform bound property, we get:

$$\mathbb{P}\left(||\hat{x}_{i,t} - \mu_i|| \leq \gamma\right) \geq 1 - 2de^{-\frac{n_{i,t}\gamma^2}{2L^2 d}}$$

Thus, for every $i \in \{1, \ldots, K\}$ and every step $t > 0$, with a probability of at least $1 - \delta/t^2$:

$$||\hat{x}_{i,t} - \mu_i|| \leq L\sqrt{\frac{2d}{n_{i,t}} \log\left(\frac{2dt^2}{\delta}\right)}$$

This bound for the deviation of the profile estimator can be less restrictive than the base assumption which states that for any $i \in \mathcal{K}$ and $t \geq 0$, $||x_{i,t}|| \leq L$. From this assumption we indeed know that, $||\hat{x}_{i,t}|| \leq L$, $||\mu_i|| \leq L$ and thus $||\hat{x}_{i,t} - \mu_i|| \leq 2L$.

We therefore consider the following bound that holds for any $t \geq 0$ with a probability greater than $1 - \delta/t^2$:

$$||\hat{x}_{i,t} - \mu_i|| \leq \min(L\sqrt{\frac{2d}{n_{i,t}} \log\left(\frac{2dt^2}{\delta}\right)}, 2L) = \rho_{i,t,\delta}$$

A.5. Proof of the proposition 6

Proof Let us assume that the inequality of the proposition 5 is valid. Therefore, we have:

- $||\mu_i||_{V_{t-1}^{-1}} - ||\hat{x}_{i,t}||_{V_{t-1}^{-1}} \leq ||\mu_i - \hat{x}_{i,t}||_{V_{t-1}^{-1}} \leq ||\mu_i - \hat{x}_{i,t}||/\sqrt{\lambda} \leq \rho_{i,t,\delta}/\sqrt{\lambda}$. Thus: $||\mu_i||_{V_{t-1}^{-1}} \leq ||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + \rho_{i,t,\delta}/\sqrt{\lambda} = ||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}}$, with $\tilde{\epsilon}_{i,t} = \rho_{i,t,\delta}\hat{x}_{i,t}/(\sqrt{\lambda}||\hat{x}_{i,t}||_{V_{t-1}^{-1}})$.

- $|\hat{\beta}_{t-1}^\top(\hat{x}_{i,t} - \mu_i)| \leq ||\hat{\beta}_{t-1}||||(\hat{x}_{i,t} - \mu_i)|| \leq \hat{\beta}_{t-1}^\top \bar{\epsilon}_{i,t}$, with $\bar{\epsilon}_{i,t} = \rho_{i,t,\delta}\hat{\beta}_{t-1}/||\hat{\beta}_{t-1}||$.

By using these two results and the uniform bound property, we can proof the proposition:

$$\begin{aligned}&\hat{\beta}_{t-1}^\top(\hat{x}_{i,t} + \bar{\epsilon}_{i,t}) + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} - \beta^\top \mu_i \\ =& (\hat{\beta}_{t-1} - \beta)^\top \mu_i + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} - \hat{\beta}_{t-1}^\top(\mu_i - \hat{x}_{i,t}) + \hat{\beta}_{t-1}^\top \bar{\epsilon}_{i,t} \\ \geq & - ||\hat{\beta}_{t-1} - \beta||_{V_{t-1}}||\mu_i||_{V_{t-1}^{-1}} + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + \hat{\beta}_{t-1}^\top(\hat{x}_{i,t} - \mu_i + \bar{\epsilon}_{i,t}) \\ \geq & - \alpha_{t-1}||\mu_i||_{V_{t-1}^{-1}} + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + \hat{\beta}_{t-1}^\top(\hat{x}_{i,t} - \mu_i + \bar{\epsilon}_{i,t}) \\ \geq & - \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + \hat{\beta}_{t-1}^\top(\hat{x}_{i,t} - \mu_i + \bar{\epsilon}_{i,t}) \\ \geq & 0\end{aligned}$$

■

A.6. Proof of the proposition 8

Lemma 4 *For every $i \in \mathcal{K}$ and $t > 0$, with a probability of at least $1 - \delta/t^2 - \delta$, we have:*

$$\begin{aligned}&\hat{\beta}_{t-1}^\top(\hat{x}_{i,t} + \bar{\epsilon}_{i,t}) + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} - \beta^\top \mu_i \\ & \leq 2\alpha_t||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 4\sqrt{d}(\alpha_{t-1}/\sqrt{\lambda} + S)\rho_{i,t,\delta}\end{aligned}$$

Proof

As for proposition 6, we assume that the inequality of proposition 5 holds. Then, noting that $||\bar{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \leq ||\bar{\epsilon}_{i,t}||/\sqrt{\lambda} = \rho_{i,t,\delta}/\sqrt{\lambda}$ and $||\tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} = \rho_{i,t,\delta}/\sqrt{\lambda}$, we have:

$$\begin{aligned}&\hat{\beta}_{t-1}^\top(\hat{x}_{i,t} + \bar{\epsilon}_{i,t}) + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} - \beta^\top \mu_i \\ =& (\hat{\beta}_{t-1} - \beta)^\top \mu_i + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} - \hat{\beta}_{t-1}^\top(\mu_i - \hat{x}_{i,t}) + \hat{\beta}_{t-1}^\top \bar{\epsilon}_{i,t} \\ \leq & ||\hat{\beta}_{t-1} - \beta||_{V_{t-1}}||\mu_i||_{V_{t-1}^{-1}} + \alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + \hat{\beta}_{t-1}^\top(\hat{x}_{i,t} - \mu_i + \bar{\epsilon}_{i,t}) \\ \leq & 2\alpha_{t-1}||\hat{x}_{i,t} + \tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + 2||\hat{\beta}_{t-1}||_{V_{t-1}}||\bar{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \\ \leq & 2\alpha_{t-1}||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 2\alpha_{t-1}||\tilde{\epsilon}_{i,t}||_{V_{t-1}^{-1}} + 2(\alpha_{t-1} + S\sqrt{\lambda})||\bar{\epsilon}_{i,t}||_{V_{t-1}^{-1}} \\ \leq & 2\alpha_{t-1}||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 4(\alpha_{t-1}/\sqrt{\lambda} + S)\rho_{i,t,\delta}\end{aligned}$$

■

Lemma 5 *For every t, with a probability of at least $1-\delta/t^2-\delta$, the instantaneous pseudo-regret of the algorithm* `SampLinUCB`*, noted $reg_t = \beta^\top \mu_{i^\star} - \beta^\top \mu_{i_t}$, is upper-bounded as:*

$$reg_t \leq \underbrace{2\alpha_{t-1}||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}}_{reg_t^{(1)}} + \underbrace{4(\alpha_{t-1}/\sqrt{\lambda}+S)\rho_{i_t,t,\delta}}_{reg_t^{(2)}}$$

Proof The previous lemma allows us to say that for all t:
$s_{i_t,t} \leq \beta^\top \mu_{i_t} + 2\alpha_{t-1}||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 4(\alpha_{t-1}/\sqrt{\lambda}+S)\rho_{i,t,\delta}$
Now, given the selection policy of `SampLinUCB` and the proposition 6, we get for all t: $s_{i_t,t} \geq s_{i^\star,t} \geq \beta^\top \mu_{i^\star}$. Thus:
$reg_t \leq s_{i_t,t} - \beta^\top \mu_{i_t} \leq 2\alpha_{t-1}||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 4(\alpha_{t-1}/\sqrt{\lambda}+S)\rho_{i,t,\delta}$

■

Next, we use the uniform bound property, the fact that $\sum_{t=2}^{\infty} \frac{\delta}{t^2} = \delta(\pi^2/6-1) \leq \delta$ and the fact that in the proposition 3 the bound is uniform (i.e., it holds for all step t simultaneously) to say that, with a probability of at least $1-2\delta$:

$$\begin{aligned}
\sum_{t=1}^{T} reg_t^{(2)} &\leq C + \sum_{t=2}^{T} 4(\alpha_{t-1}/\sqrt{\lambda}+S)\rho_{i_t,t,\delta} \\
&\leq C + \sum_{t=2}^{T} 4(\alpha_{t-1}/\sqrt{\lambda}+S)L\sqrt{\frac{2d}{n_{i,t}}\log\left(\frac{2dt^2}{\delta}\right)} \\
&\leq C + 4L(\alpha_T/\sqrt{\lambda}+S)\sqrt{2d\log\left(\frac{2dT^2}{\delta}\right)}\sum_{t=2}^{T}\frac{1}{\sqrt{n_{i_t,t}}}
\end{aligned}$$

On another hand, we have:

$$\begin{aligned}
\sum_{t=1}^{T} reg_t^{(1)} &\leq \sum_{t=1}^{T} 2\alpha_{t-1}||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}^2 \\
&\leq \sqrt{T\sum_{t=1}^{T} 4\alpha_{t-1}^2||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}^2} \\
&\leq 2\alpha_T\sqrt{T\sum_{t=1}^{T}||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}^2}
\end{aligned}$$

Now, it remains to upper-bound the term $\sum_{t=1}^{T}||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}^2$.

For that purpose, we introduce the following notation: $\nu_{i,t,\delta} = L\sqrt{2d/(n_{i,t})\log(2dT/\delta)}$. By using again the Hoeffding inequality, with a probability of at least $1-\delta/T$, we get for all $s \leq t-1$:

$$||\hat{x}_{i_s,t}|| \leq ||\mu_{i_s}|| + \nu_{i_s,t,\delta}$$

With $\check{\epsilon}_{i,t} = min(\nu_{i,t,\delta}, ||\mu_i||)\mu_i/||\mu_i||$, we get, for all $s \leq t-1$:

$$1/\sqrt{R_{i_s,s}}||\mu_{i_s} - \check{\epsilon}_{i_s,s}|| \leq 1/\sqrt{R_{i_s,t}}||\hat{x}_{i_s,t}||$$

Then, we arrive to:

$$V_{t-1} = \lambda I + \sum_{s=1}^{t-1}\frac{1}{R_{i_s,t}}\hat{x}_{i_s,t}\hat{x}_{i_s,t}^\top \geq \lambda I + \sum_{s=1}^{t-1}\frac{1}{R_{i_s,s}}(\mu_{i_s} - \check{\epsilon}_{i_s,s})(\mu_{i_s} - \check{\epsilon}_{i_s,s})^\top = W_{t-1}$$

Which means that for every vector x: $||x||_{V_{t-1}^{-1}} \leq ||x||_{W_{t-1}^{-1}}$.
Let us now define $\hat{\epsilon}_{i,t} = \nu_{i,t,\delta}\mu_i/(\sqrt{\lambda}||\mu_i||_{W_{t-1}^{-1}})$, such that for all $s \leq t-1$:

$$||\hat{x}_{i_s,t}||_{W_{t-1}^{-1}} \leq ||\mu_{i_s} + \hat{\epsilon}_{i_s,s}||_{W_{t-1}^{-1}}$$

and

$$||\hat{\epsilon}_{i_s,s}||_{W_{t-1}^{-1}} = \nu_{i_s,s,\delta}/\sqrt{\lambda}$$

Finally, by using the uniform bound property and the fact that $\sum_{t=1}^{T}\frac{\delta}{T} = \delta$, with a probability of at least $1-\delta$:

$$\begin{aligned}
\sum_{t=1}^{T}||\hat{x}_{i_t,t}||_{V_{t-1}^{-1}}^2 &\leq \sum_{t=1}^{T}||\hat{x}_{i_t,t}||_{W_{t-1}^{-1}}^2 \\
&\leq \sum_{t=1}^{T}||\mu_{i_t} + \hat{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 \leq \sum_{t=1}^{T}||\mu_{i_t} + \hat{\epsilon}_{i_t,t} - \check{\epsilon}_{i_t,t} + \check{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 \\
&\leq \sum_{t=1}^{T}||\mu_{i_t} - \check{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 + \sum_{t=1}^{T}||\hat{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 + \sum_{t=1}^{T}||\check{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 \\
&\leq \sum_{t=1}^{T}||\mu_{i_t} - \check{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 + \frac{2}{\lambda}\sum_{t=1}^{T}\nu_{i_t,t,\delta}^2 \\
&\leq \sum_{t=1}^{T}||\mu_{i_t} - \check{\epsilon}_{i_t,t}||_{W_{t-1}^{-1}}^2 + \frac{4L^2d}{\lambda}\log\left(\frac{2dT}{\delta}\right)\sum_{t=1}^{T}\frac{1}{n_{i_t,t}}
\end{aligned}$$

On another hand, we have:

$$\begin{aligned}
det(W_T) &= det(W_{T-1} + \frac{1}{R_{i_T,T}}(\mu_{i_T} - \check{\epsilon}_{i_T,T})(\mu_{a_T} - \check{\epsilon}_{i_T,T})^\top) \\
&= det(W_{T-1}) det(I + \frac{1}{R_{i_T,T}} W_{T-1}^{-1/2}(\mu_{i_T} - \check{\epsilon}_{i_T,T})(W_{T-1}^{-1/2}(\mu_{i_T} - \check{\epsilon}_{i_T,T}))^\top) \\
&= det(W_{T-1})(1 + \frac{1}{R_{i_T,T}} ||\mu_{i_T} - \check{\epsilon}_{i_T,T}||^2_{W_{T-1}^{-1}}) \\
&= det(\lambda I) \prod_{t=1}^{T} (1 + \frac{1}{R_{i_t,t}} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}})
\end{aligned}$$

Where we used the fact that all eigenvalues of $I + xx^\top$ equal 1 except one that is associated to the eigenvector x and thus equals $1 + ||x||^2$.

Since by assumption $\lambda > \max(1, L^2/\sqrt{R^2})$, we have:

$$\frac{1}{R_{i_t,t}} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}} \leq ||\hat{x}_{i_t,t}||^2/\sqrt{R^2}\lambda \leq L^2/\sqrt{R^2}\lambda \leq 1$$

Thus, by using the fact that $x \leq 2\log(1+x)$ when $0 \leq x \leq 1$, we get:

$$\begin{aligned}
2\log\left(\frac{det(W_T)}{det(\lambda I)}\right) &\geq \sum_{t=1}^{T} \frac{1}{R_{i_t,t}} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}} \\
&\geq \min_{t=1..T}\left(\frac{1}{R_{i_t,t}}\right) \sum_{t=1}^{T} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}} \\
&\geq 1/\sqrt{R^2 + L^2S^2} \sum_{t=1}^{T} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}}
\end{aligned}$$

As in the lemma 11 of (Abbasi-Yadkori et al., 2011), we also have:

$$\log\left(\frac{det(W_T)}{det(\lambda I)}\right) \leq d\log\left(1 + \frac{TL^2}{\lambda d}\right)$$

Which leads us to:

$$\sum_{t=1}^{T} ||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}} \leq \sqrt{R^2 + L^2S^2} d\log\left(1 + \frac{TL^2}{\lambda d}\right)$$

Finally, same manner as in the lemma 10 of Abbasi-Yadkori et al. (2011), the trace-determinant inequality gives:

$$\alpha_T \leq \sqrt{d\log\left(\frac{1 + TL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S$$

Gathering all these results together allows us to prove the announced result.

A.7. Proof of the proposition 9

Lemma 6 *When removing dependencies on L, λ, R and S, the bound from proposition 16 for the cumulative regret $\hat{R}_T$ can written as follows (when $T > d$):*

$$\hat{R}_T \leq C + C_1 d\log\left(\frac{T}{\delta}\right) \sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} + C_2 d\log\left(\frac{T}{\delta}\right) \sqrt{T \sum_{t=1}^{T} \frac{1}{n_{i_t,t}}}$$

with C,C_1 and C_2 three constants.

Proof From proposition 16, we have:

$$\hat{R}_T \leq C + \hat{R}_{T,1} + \hat{R}_{T,2}$$

where:

$$\begin{aligned}
\hat{R}_{T,1} &= 4L\left(\sqrt{\frac{d}{\lambda}\log\left(\frac{1 + TL^2/\lambda}{\delta}\right)} + 2S\right) \sqrt{2d\log\left(\frac{2dT^2}{\delta}\right)} \sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} \\
&\leq Constant \times d\sqrt{\log\left(\frac{T}{\delta}\right)\log\left(\frac{dT^2}{\delta}\right)} \sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} \\
&\leq Constant \times d\log\left(\frac{T}{\delta}\right) \sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} \qquad (\text{when } T > d > 0)
\end{aligned}$$

And

$$\begin{aligned}
\hat{R}_{T,2} &= 2\left(\sqrt{d\log\left(\frac{1 + TL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S\right) \\
&\times \sqrt{Td\left(\sqrt{R^2 + L^2S^2}\log\left(1 + \frac{TL^2}{\lambda d}\right) + \frac{4L^2}{\lambda}\log\left(\frac{2dT}{\delta}\right) \sum_{t=1}^{T} \frac{1}{n_{i_t,t}}\right)} \\
&\leq Constant \times \sqrt{d\log\left(\frac{T}{\delta}\right)} \sqrt{Td\left(\log(T) + \log\left(\frac{dT}{\delta}\right) \sum_{t=1}^{T} \frac{1}{n_{i_t,t}}\right)} \\
&\leq Constant \times \sqrt{d\log\left(\frac{T}{\delta}\right)} \sqrt{Td\log\left(\frac{dT}{\delta}\right) \sum_{t=1}^{T} \frac{1}{n_{i_t,t}}} \\
&\leq Constant \times d\log\left(\frac{T}{\delta}\right) \sqrt{T \sum_{t=1}^{T} \frac{1}{n_{i_t,t}}} \qquad (\text{when } T > d > 0)
\end{aligned}$$

■

Thanks to this lemma, we are ready to derive specific bounds for the three considered profile delivery settings.

A.7.1. Case 1:

On one hand, we have:

$$\sum_{t=1}^{T} \frac{1}{n_{i_t,t}} = \sum_{t=1}^{T} \frac{1}{t} \leq 1 + \log(T)$$

On the other hand, we have:

$$\sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} = \sum_{t=1}^{T} \frac{1}{\sqrt{t}} \leq \int_0^T \frac{1}{\sqrt{t}} \, \mathrm{d}t \leq 2\sqrt{T}$$

Finally, we get from lemma 6 that, when $T > d > 0$:

$$\hat{R}_{T,1} \leq C + C_1 d \log\left(\frac{T}{\delta}\right) \sqrt{T} + C_2 d \log\left(\frac{T}{\delta}\right) \sqrt{T \log(T)}$$

with C,C_1 and C_2 three constants. Since the last term is clearly the greatest, we get the announced result.

A.7.2. Case 2:

Lemma 7 *$\forall i, \forall t \geq \lceil 2\log(1/\delta)/p^2 \rceil$, with a probability of at least $1-\delta$:*

$$n_{i,t} \geq \frac{tp}{2}$$

Proof By the Hoeffding inequality, for all $\epsilon > 0$:

$$\mathbb{P}(n_{i,t} \geq tp - \epsilon) \geq 1 - e^{-2\epsilon^2/t}$$

By taking $\epsilon = tp/2$, we get:

$$\mathbb{P}(n_{i,t} \geq tp/2) \geq 1 - e^{-tp^2/2}$$

If $t \geq 2\log(1/\delta)/p^2$, then $1 - e^{-tp^2/2} \geq 1 - \delta$, which proves the lemma. ■

Let us note $u = ceil(2\log(1/\delta)/p^2)$. Thus, following lemma 7, with a probability of at least $1-\delta$, we have:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{n_{i_t,t}} &= \sum_{t=1}^{u} \frac{1}{n_{i_t,t}} + \sum_{t=u+1}^{T} \frac{1}{n_{i_t,t}} \\
&\leq u + \frac{2}{p} \sum_{t=u+1}^{T} \frac{1}{t} \\
&\leq u + \frac{2}{p} \int_u^T \frac{1}{t} \, \mathrm{d}t \\
&\leq u + \frac{2\log(T)}{p}
\end{aligned}$$

From another hand, still thanks to the lemma 7, with a probability of at least $1-\delta$:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} &= \sum_{t=1}^{u} \frac{1}{\sqrt{n_{i_t,t}}} + \sum_{t=u+1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} \\
&\leq u + \sqrt{\frac{2}{p}} \sum_{t=u+1}^{T} \frac{1}{\sqrt{t}} \\
&\leq u + \sqrt{\frac{2}{p}} \int_u^T \frac{1}{\sqrt{t}} \, \mathrm{d}t \\
&\leq u + 2\sqrt{\frac{2T}{p}}
\end{aligned}$$

Finally, we get from lemma 6 that, when $T > d > 0$:

$$\hat{R}_{T,1} \leq C + C_1 d \log\left(\frac{T}{\delta}\right) \sqrt{\frac{T}{p}} + C_2 d \log\left(\frac{T}{\delta}\right) \sqrt{T \frac{\log(T)}{p}}$$

with C,C_1 and C_2 three constants. Since the last term is clearly the greatest, we get the announced result.

A.7.3. Case 3:

First note that the sum $\sum_{t=1}^{T} \frac{1}{n_{i_t,t}}$ is maximized when each action has delivered exactly $\lfloor T/K \rfloor$ samples in the $\lfloor T/K \rfloor K$ first iterations (i.e., every action has been played as many times). Thus:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{n_{i_t,t}} &\leq \sum_{i=1}^{K} \sum_{t=1}^{\lfloor T/K \rfloor + 1} \frac{1}{t} \\
&\leq K \sum_{t=1}^{\lceil T/K \rceil} \frac{1}{t} \\
&\leq K(1 + \log(\lceil T/K \rceil))
\end{aligned}$$

With the same argument, we also get:

$$\begin{aligned}
\sum_{t=1}^{T} \frac{1}{\sqrt{n_{i_t,t}}} &\leq K \sum_{t=1}^{\lceil T/K \rceil} \frac{1}{\sqrt{t}} \\
&\leq 2K\sqrt{\lceil T/K \rceil}
\end{aligned}$$

Finally, by noting that $K\log(\lceil T/K \rceil) \sim K\log(T/K)$, that $K\sqrt{\lceil T/K \rceil} \sim \sqrt{KT}$ and by using the generic bound from the proposition 9, we get the announced result. Finally, since

$K\log(\lceil T/K\rceil) \sim K\log(T/K)$ and $K\sqrt{\lceil T/K\rceil} \sim \sqrt{KT}$, we get from lemma 6 that, when $T > d > 0$:

$$\hat{R}_{T,1} \leq C + C_1 d\log\left(\frac{T}{\delta}\right)\sqrt{KT} + C_2 d\log\left(\frac{T}{\delta}\right)\sqrt{TK\log(T/K)}$$

with C,C_1 and C_2 three constants. Since the last term is clearly the greatest, we get the announced result.

A.8. Proof of the proposition 10

We follow a similar method to the one presented in Qin et al. (2014) for the specific case of sums of individual rewards: Since we consider that the reward obtained by playing a set of actions at a given step t is the sum of rewards observed for every single played action at t, we have:

$$reg_t = \sum_{i\in\mathcal{K}^\star} \mu_i^\top \beta - \sum_{i\in\mathcal{K}_t} \mu_i^\top \beta$$

with $\mathcal{K}^\star$ the set of k optimal actions, i.e., those with greatest expectations $\mu_i^\top\beta$. Then, we use the fact that for every step t:

$$\sum_{i\in\mathcal{K}_t} s_{i,t} \geq \sum_{i\in\mathcal{K}_\star} s_{i^\star,t}$$

This leads to:

$$reg_t \leq \sum_{i\in\mathcal{K}_t} 2\alpha_{t-1}||\hat{x}_{i,t}||_{V_{t-1}^{-1}} + 4\sqrt{d}(\alpha_{t-1}/\sqrt{\lambda} + S)\rho_{i,t,\delta}$$

Where the matrix V_{t-1} is defined by considering the k played actions at each step: $V_{t-1} = \lambda I + \sum_{s=1}^{t-1}\sum_{i\in\mathcal{K}_s}\frac{1}{R_{i,t}}\hat{x}_{i,t}\hat{x}_{i,t}^\top$. The fact that we add k terms to the matrix V at each iteration implies two distinct things:

- First on the confidence ellipsoid of β. For k actions played at each step, we have:

  $$\alpha_T \leq \sqrt{d\log\left(\frac{1+kTL^2/\lambda}{\delta}\right)} + \sqrt{\lambda}S;$$

- On another hand on the upper-bounding of $\sum_{t=1}^{T}\sum_{i_t\in\mathcal{K}_t}||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}}$. We have:

  $$\sum_{t=1}^{T}\sum_{i_t\in\mathcal{K}_t}||\mu_{i_t} - \check{\epsilon}_{i_t,t}||^2_{W_{t-1}^{-1}} \leq \sqrt{R^2+L^2S^2}d\log\left(1+\frac{TkL^2}{\lambda d}\right)$$

Finally, we can use the same methods as in the previous proofs for deriving specific bounds from the generic one, where the selection of k actions at each step appears explicitly in the terms $\sum_{t=1}^{T}\sum_{i\in\mathcal{K}_t}\frac{1}{n_{i,t}}$ and $\sum_{t=1}^{T}\sum_{i\in\mathcal{K}_t}\frac{1}{\sqrt{n_{i,t}}}$.

A.9. Table of the main notations

T	number of steps of the process
$\mathcal{K}$	set of available actions
K	number of available arms
k	number of simultaneaous plays at each step
d	dimension of the arms' profiles
i_t	arm selected at step t
i^*	arm with the highest final cumulative reward
r_i	reward obtained by arm i at step t
$x_{i,t}$	profile sample vector observed for arm i at step t
$\hat{x}_{i,t}$	average of profile sample vectors observed for arm i until step t
L	upper-bound for the profiles' norm
$\mathcal{O}_t$	set of arms delivering a profile context at step t
$n_{i,t}$	number of samples observed for arm i until step t
μ_i	profile vector of arm i
β	mapping parameter between profiles and rewards
$\hat{\beta}_t$	estimator of β at step t
S	upper-bound for the β parameter norm
λ	l2-regularization constant of the β estimator
$\eta_{i,t}$	sub-gaussian noise of the reward of arm i at step t
R	sub-gaussian constant of the rewards distribution
$\epsilon_{i,t}$	deviation between the true profile of arm i and its estimator at step t ($\epsilon_{i,t} = \mu_i - \hat{x}_{i,t}$)
η'_{t-1}	vector of reward deviations of the first $t-1$ selected arms from their expectation at step t: $\eta'_{t-1} = (\eta_{i_s,s} + \epsilon_{i_s,t}^\top\beta)_{s=1..t-1}^\top$
$R_{i,t}$	sub-gaussian constant of the noise of the reward of i at step t w.r.t. $\hat{x}_{i,t}^\top\beta$
X_{t-1}	$(t-1)\times d$ matrix containing the empirical means of the selected actions, where the s-th row corresponds to the estimator at step t of the action selected at step s: $X_{t-1} = (\hat{x}_{i_s,t}^\top)_{s=1..t-1}$
Y_{t-1}	rewards vector of size $t-1$: $Y_{t-1} = (r_{i_s,s})_{s=1..t-1}^\top$
A_{t-1}	diagonal $(t-1)\times(t-1)$ matrix, where the s-th diagonal element equals $1/R_{i_s,t}$: $A_{t-1} = diag(1/R_{i_s,t})_{s=1..t-1}$
δ	parameter controling the confidence level of the regret bound
V_t^{-1}	variance-covariance matrix of the posterior distribution of β at step t
$\rho_{i,t,\delta}$	quantity used to bound the deviation of the estimators of profiles: $\rho_{i,t,\delta} = \min(L\sqrt{\frac{2d}{n_{i,t}}\log\left(\frac{2dt^2}{\delta}\right)}, 2L)$
$s_{i,t}$	selection score for arm i at step t
α_t	exploration coefficient at step t w.r.t. the confidence of the *beta* estimator
$\bar{\epsilon}_t$	quantity used to bound $\mu_i^\top\hat{\beta}_t$
$\tilde{\epsilon}_t$	quantity used to bound $\mu_i^\top(\beta - \hat{\beta}_t)$

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.*, pages 2312–2320, 2011.

Shipra Agrawal and Navin Goyal. Analysis of thompson sampling for the multi-armed bandit problem. In *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, pages 39.1–39.26, 2012.

Shipra Agrawal and Navin Goyal. Thompson sampling for contextual bandits with linear payoffs. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 127–135, 2013.

Jean-Yves Audibert and Sébastien Bubeck. Minimax policies for adversarial and stochastic bandits. In *COLT 2009 - The 22nd Conference on Learning Theory, Montreal, Quebec, Canada, June 18-21, 2009*, 2009.

Jean-Yves Audibert, Rémi Munos, and Csaba Szepesvári. Exploration-exploitation tradeoff using variance estimates in multi-armed bandits. *Theor. Comput. Sci.*, 410(19):1876–1902, April 2009. ISSN 0304-3975.

Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *Journal of Machine Learning Research*, 2003.

Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In *Proceedings of the 25th Conference on Neural Information Processing Systems 2011, Granada, Spain*, pages 2249–2257, 2011.

Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013, Atlanta, GA, USA, 16-21 June 2013*, pages 151–159, 2013.

Wei Chu, Lihong Li, Lev Reyzin, and Robert E. Schapire. Contextual bandits with linear payoff functions. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2011, Fort Lauderdale, USA, April 11-13, 2011*, pages 208–214, 2011.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, pages 355–366, 2008.

V. H. de la Peña, T. L. Lai, and Q. M. Shao. *Self-Normalized Processes: Limit Theory and Statistical Applications*. Springer Series in Probability and its Applications. Springer, 2009.

Sarah Filippi, Olivier Cappé, Aurélien Garivier, and Csaba Szepesvári. Parametric bandits: The generalized linear case. In *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010. Proceedings of a meeting held 6-9 December 2010, Vancouver, British Columbia, Canada.*, pages 586–594, 2010.

Aurlien Garivier. The kl-ucb algorithm for bounded stochastic bandits and beyond. In *COLT*, 2011.

Thibault Gisselbrecht, Ludovic Denoyer, Patrick Gallinari, and Sylvain Lamprier. Whichstreams: A dynamic approach for focused data capture from large social media. In *Proceedings of the Ninth International Conference on Web and Social Media, ICWSM 2015, University of Oxford, Oxford, UK, May 26-29, 2015*, pages 130–139, 2015.

Liangjie Hong and Brian D. Davison. Empirical study of topic modeling in twitter. In *ECIR*, 2010.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On bayesian upper confidence bounds for bandit problems. In *Proceedings of the Fifteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2012, La Palma, Canary Islands, April 21-23, 2012*, pages 592–600, 2012a.

Emilie Kaufmann, Nathaniel Korda, and Rémi Munos. Thompson sampling: An asymptotically optimal finite-time analysis. In *Algorithmic Learning Theory - 23rd International Conference, ALT 2012, Lyon, France, October 29-31, 2012. Proceedings*, pages 199–213, 2012b.

T.L Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985. ISSN 0196-8858.

Lihong Li, Wei Chu, John Langford, and Robert E. Schapire. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 661–670, New York, NY, USA, 2010. ACM. ISBN 978-1-60558-799-8.

Benedict C. May, Nathan Korda, Anthony Lee, and David S. Leslie. Optimistic bayesian sampling in contextual-bandit problems. *J. Mach. Learn. Res.*, 13:2069–2106, 2012. ISSN 1532-4435.

Lijing Qin, Shouyuan Chen, and Xiaoyan Zhu. Contextual combinatorial bandit and its application on diversified online recommendation. In *Proceedings of the 2014 SIAM International Conference on Data Mining, Philadelphia, Pennsylvania, USA, April 24-26, 2014*, pages 461–469, 2014.

Paat Rusmevichientong and John N. Tsitsiklis. Linearly parameterized bandits. *Math. Oper. Res.*, 35(2):395–411, 2010.

W.R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Bulletin of the American Mathematics Society*, 25:285–294, 1933.

Journal of Machine Learning Research 19 (2018) 1-46 Submitted 1/18; Revised 8/18; Published 9/18

How Deep Are Deep Gaussian Processes?

Matthew M. Dunlop MDUNLOP@CALTECH.EDU
Computing and Mathematical Sciences
Caltech
Pasadena, CA 91125, USA

Mark A. Girolami M.GIROLAMI@IMPERIAL.AC.UK
Department of Mathematics
Imperial College London
London, SW7 2AZ, UK
and
The Alan Turing Institute
96 Euston Road
London, NW1 2DB, UK

Andrew M. Stuart ASTUART@CALTECH.EDU
Computing and Mathematical Sciences
Caltech
Pasadena, CA 91125, USA

Aretha L. Teckentrup A.TECKENTRUP@ED.AC.UK
School of Mathematics
University of Edinburgh
Edinburgh, EH9 3FD, UK
and
The Alan Turing Institute
96 Euston Road
London, NW1 2DB, UK

Editor: Neil Lawrence

Abstract

Recent research has shown the potential utility of deep Gaussian processes. These deep structures are probability distributions, designed through hierarchical construction, which are conditionally Gaussian. In this paper, the current published body of work is placed in a common framework and, through recursion, several classes of deep Gaussian processes are defined. The resulting samples generated from a deep Gaussian process have a Markovian structure with respect to the depth parameter, and the effective depth of the resulting process is interpreted in terms of the ergodicity, or non-ergodicity, of the resulting Markov chain. For the classes of deep Gaussian processes introduced, we provide results concerning their ergodicity and hence their effective depth. We also demonstrate how these processes may be used for inference; in particular we show how a Metropolis-within-Gibbs construction across the levels of the hierarchy can be used to derive sampling tools which are robust to the level of resolution used to represent the functions on a computer. For illustration, we consider the effect of ergodicity in some simple numerical examples.

Keywords: deep learning, deep Gaussian processes, deep kernels

1. Introduction

This section provides background on the study and application of Gaussian and deep Gaussian processes, outlines the contribution and setup of the paper, and establishes notation to be used in subsequent sections.

1.1 Background

Gaussian processes have proved remarkably successful as a tool for various statistical inference and machine learning tasks (Rasmussen and Williams, 2006; Kennedy and O'Hagan, 2001; Higdon et al., 2004; Stein, 1999). This success relates in part to the ease with which computations may be performed in the Gaussian framework, and also to the flexible ways in which Gaussian processes may be used, for example when combined with thresholding to perform classification tasks via probit models (Neal, 1997; Rasmussen and Williams, 2006) or to find interfaces in Bayesian inversion (Iglesias et al., 2016). Nonetheless there are limits to the sort of phenomena that are readily expressible via direct use of Gaussian processes, such as in the sparse data scenario, where the constructed probability distribution is far from posterior contraction. Recognizing this fact, there have been a number of interesting research activities which seek to represent new phenomena via the hierarchical cascading of Gaussians. Early work of this type includes the PhD thesis of Paciorek (2003) (see also Paciorek and Schervish, 2004) in which the aim is to reproduce spatially non-stationary phenomena, and this is achieved by means of a Gaussian process whose covariance function itself depends on another Gaussian process. This idea was recently re-visited by Roininen et al. (2016), using the precision operator viewpoint, rather than covariance function, and building on the explicit link between Gaussian processes and stochastic partial differential equations (SPDEs) (Lindgren et al., 2011). A different approach was adopted by Damianou and Lawrence (2013) where a Gaussian process was directly composed with another Gaussian process; furthermore the idea was implemented recursively, leading to what is referred to as deep Gaussian processes (DGP). These ingenious constructions open up new possibilities for problems in non-parametric inference and machine learning and the purpose of this paper is to establish, and utilize, a common framework for their study. Relevant to our analysis is the early work of Diaconis and Freedman (1999) which studied iterations of random Lipschitz functions and the conditions required for their convergence.

1.2 Our Contribution

In the paper we make three main contributions:

- We demonstrate a unifying perspective on the hierarchical Gaussian processes described in the previous subsection, leading to a wide class of deep Gaussian processes, with a common framework within which new deep Gaussian processes can be constructed.
- By exploiting the fact that this common framework has a Markovian structure, we interpret the depth of the process in terms of the ergodicity or non-ergodicity of this process; in simple terms ergodic constructions have effective depth given by the mixing time.

- We demonstrate how these processes may be used for inference; in particular we show how a Metropolis-within-Gibbs construction across the levels of the hierarchy can be used to derive sampling tools which are robust to the level of resolution used to represent the functions on a computer.

We also describe numerical experiments which illustrate the theory, and which demonstrate some of the limitations of the framework in the inference context, suggesting the need for further algorithmic innovation and theoretical understanding. We now summarize the results and contributions by direct reference to the main theorems in the paper.

- Theorem 4 shows that a composition-based deep Gaussian process will, with sufficiently many layers, produce samples that are approximately constant. This pathology can be avoided by, for example, increasing the width of each hidden layer, or allowing each layer to depend on the input layer.
- Theorem 8 shows the ergodicity of a class of discretized deep Gaussian processes, constructed using non-stationary covariance *functions*. As a consequence, there is little benefit in adding additional layers after a certain point. This observation elucidates the mechanism underlying the choices of DGPs with a small number of layers for inference in numerous papers (for example Cutajar et al., 2017; Salimbeni and Deisenroth, 2017; Dai et al., 2015).
- Theorem 14 establishes a similar result as Theorem 8 on function space, for a different class of deep Gaussian processes constructed using non-stationary covariance *operators*.
- Theorem 16 establishes the asymptotic properties of a deep Gaussian process formed by iterated convolution of fairly general classes of Gaussian random fields. Specifically it is shown that such processes will either converge weakly to zero or diverge as the number of layers is increased, and so they will provide little flexibility for inference in practice.

1.3 Overview

The general framework in which we place the existing literature, and which we employ to analyze deep Gaussian processes, and to construct algorithms for related inference tasks, is as follows. We consider sequences of functions $\{u_n\}$ which are conditionally Gaussian:

$$u_{n+1}|u_n \sim N\big(m(u_n), C(u_n)\big); \tag{CovOp}$$

here $m(u_n)$ denotes the mean function and $C(u_n)$ the covariance operator. We will also sometimes work with the covariance function representation, in which case we will write

$$u_{n+1}|u_n \sim \mathrm{GP}\big(m(x;u_n), c(x,x';u_n)\big). \tag{GP}$$

Note that the covariance function is the kernel of the covariance operator when the latter is represented as an integral operator over the approximate domain $D \subseteq \mathbb{R}^d$:

$$\big(C(u_n)\phi\big)(x) = \int c(x,x';u_n)\phi(x')dx'.$$

In most of the paper we consider the centred case where $m \equiv 0$, although the flexibility of allowing for non-zero mean will be important in some applications, as discussed in the conclusions. When the mean is zero, the iterations (CovOp) and (GP) can be written in the form

$$u_{n+1} = L(u_n)\xi_{n+1}, \tag{ZeroMean}$$

where $\{\xi_n\}$ form an i.i.d. Gaussian sequence and, for each u, $L(u)$ is a linear operator. For example if the ξ_n are white then the covariance operator is $C(u) = L(u)L(u)^\top$ with $\top$ denoting the adjoint operation and $L(u)$ is a Cholesky factor of $C(u)$. The formulation (ZeroMean) is useful in much of our analysis. For the purpose of this paper, we will refer to any sequence of functions constructed as in (ZeroMean) as a deep Gaussian process.

In section 2 we discuss the hierarchical Gaussian constructions referenced above, and place them in the setting of equations (CovOp), (GP) and (ZeroMean). Section 3 studies the ergodicity of the resulting deep Gaussian processes, using the Markov chain which defines them. In section 4 we provide supporting numerical experiments; we give illustrations of draws from deep Gaussian process priors, and we discuss inference. In the context of inference we describe a methodology for MCMC, using deep Gaussian priors, which is defined in the function space limit and is hence independent of the level of resolution used to represent the functions u_n; numerical illustrations are given. We conclude in section 5 in which we describe generalizations of the settings considered in this paper, and highlight future directions.

1.4 Notation

The structure of the deep Gaussian processes above means that they can be interpreted as Markov chains on a Hilbert space $\mathcal{H}$ of functions. Let $\mathcal{B}(\mathcal{H})$ denote the Borel σ-algebra on $\mathcal{H}$. We denote by $\mathsf{P} : \mathcal{H} \times \mathcal{B}(\mathcal{H}) \to \mathbb{R}$ the one-step transition probability distribution,

$$\mathsf{P}(u, A) = \mathbb{P}(u_n \in A \,|\, u_{n-1} = u), \tag{1}$$

and denote by $\mathsf{P}^n : \mathcal{H} \times \mathcal{B}(\mathcal{H}) \to \mathbb{R}$ the n-step transition probability distribution,

$$\mathsf{P}^n(u, A) = \mathbb{P}(u_n \in A \,|\, u_0 = u). \tag{2}$$

Thus, for example, in the case of the covariance operator construction (CovOp) we have

$$\mathsf{P}(u, \cdot) = N\big(0, C(u)\big),$$

when the mean is zero. This Markovian structure will be exploited when showing ergodicity, or lack of ergodicity, of the chains.

2. Four Constructions

This section provides examples of four constructions of deep Gaussian processes, all of which fall into our general framework. The reader will readily design others.

2.1 Composition

Let $D \subseteq \mathbb{R}^d$, $D' \subseteq \mathbb{R}^l$, $u_n : D \to \mathbb{R}^m$ and $F : \mathbb{R}^m \to D'$. If $\{\xi_n\}$ is a collection of i.i.d. centred Gaussian processes taking values in the space of continuous functions $C(D';\mathbb{R}^m)$ then we define the Markov chain

$$u_{n+1}(x) = \xi_{n+1}\Big(F\big(u_n(x)\big)\Big). \tag{3}$$

The case $m = l$, $F = \mathrm{id}$ and $D = D' = \mathbb{R}^m$ was introduced by Damianou and Lawrence (2013) and the generalization here is inspired by the formulation of Duvenaud et al. (2014). The case where two layers are employed could be interpreted as a form of warped Gaussian process: a generalization of Gaussian processes that have been used successfully in a number of inference problems (Snelson et al., 2004; Schmidt and O'Hagan, 2003).

We note that the mapping $\xi \mapsto \xi \circ F \circ u$ is linear, and we may thus define $L(u)$ by $L(u)\xi = \xi \circ F \circ u$; hence the Markov chain may be written in the form (ZeroMean). If $\xi_1 \sim N(0,\Sigma)$ then the Markov chain has the form (CovOp), with mean zero and $C(u) = L(u)\Sigma L(u)^*$; if $\xi_1 \sim \mathrm{GP}\big(0, k(z,z')\big)$ then the Markov chain has the form (GP) with mean zero and $c(x,x';u) = k\Big(F\big(u(x)\big), F\big(u(x')\big)\Big)$.

2.2 Covariance Function

Paciorek (2003) gives a general strategy to construct anisotropic versions of isotropic covariance functions. Let $\Sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be such that $\Sigma(z)$ is symmetric positive definite for all $z \in \mathbb{R}^d$, and define the quadratic form

$$Q(x,x') = (x-x')^T \left(\frac{\Sigma(x)+\Sigma(x')}{2}\right)^{-1}(x-x'), \qquad x,x' \in \mathbb{R}^d.$$

If the isotropic correlation function $\rho_S(\cdot)$ is positive definite on $\mathbb{R}^d$, for all $d \in \mathbb{N}$, then the function

$$c(x,x') = \sigma^2 \frac{2^{\frac{d}{2}} \det(\Sigma(x))^{\frac{1}{4}} \det(\Sigma(x'))^{\frac{1}{4}}}{\det(\Sigma(x)+\Sigma(x'))^{\frac{1}{2}}} \rho_S(\sqrt{Q(x,x')})$$

is positive definite on $\mathbb{R}^d \times \mathbb{R}^d$ and may thus be used as a covariance function. We make these statements precise below. If we choose Σ to depend on u_n then this may be used as the basis of a deep Gaussian process. To be concrete we choose

$$\Sigma(x) = F\big(u(x)\big) I_d$$

where $F : \mathbb{R} \to \mathbb{R}_{\geq 0}$ for $u : D \subseteq \mathbb{R}^d \to \mathbb{R}$. We then write $c(x,x';u)$. Now let $u_n : D \to \mathbb{R}$ and consider the Markov chain (GP) in the mean zero case. Paciorek (2003) considered this iteration over one-step with $u_0 \sim \mathrm{GP}(0, \sigma^2\rho_S(\|x-x'\|))$ and u_1 was shown to exhibit interesting non-stationary effects. Here we generalize and consider the deep process that results from this construction for arbitrary $n \in \mathbb{N}$. By considering the covariance operator

$$\big(C(u)\varphi\big)(x) = \int_{\mathbb{R}^d} c(x,x';u)\varphi(x')\,\mathrm{d}x'$$

we may write the iteration in the form (CovOp). The form (ZeroMean) follows with $L(u) = C(u)^{\frac{1}{2}}$ and ξ_{n+1} being white noise.

Various generalizations of this construction are possible, for example allowing the pointwise variance of the process σ^2 to be spatially varying (Heinonen et al., 2016) and to depend on $u_n(x)$. These may be useful in applications, but we confine our analysis to the simpler setting for expository purposes; however in Remark 12 we discuss this generalization.

In order to make the statements made above precise, let $\rho_S : [0,\infty) \to \mathbb{R}$ be a stationary covariance kernel, where the covariance between locations x and y depends only on the Euclidean distance $\|x-y\|_2$. We make the following assumption on ρ_S.

Assumptions 1 *(i) The covariance kernel $\rho_S(\|x-y\|_2)$ is positive definite[1] on $\mathbb{R}^d \times \mathbb{R}^d$: for any $N \in \mathbb{N}$, $b \in \mathbb{R}^N \backslash \{0\}$ and pairwise distinct $\{x_i\}_{i=1}^N \subseteq \mathbb{R}^d$, we have*

$$\sum_{i=1}^N \sum_{j=1}^N b_i b_j \rho_S\big(\|x_i - x_j\|_2\big) > 0.$$

(ii) ρ_S is normalized to be a correlation kernel, i.e. $\rho_S(0) = 1$.

Using (Wendland, 2004, Theorem 6.11), sufficient conditions for ρ_S to fulfill Assumptions 1(i) are that ρ_S, as a function of $x-y$, is continuous, bounded and in $L^1(\mathbb{R}^d)$, with a Fourier transform that is non-negative and non-vanishing. These sufficient conditions are satisfied, for example, for the family of Matèrn covariance functions and the Gaussian covariance. To satisfy Assumptions 1(ii), any positive definite kernel $\tilde{\rho}_S$ can simply be rescaled by $\tilde{\rho}_S(0)$.

We now have the following proposition, a slightly weaker version of which is proved by Paciorek (2003), where it is shown that $\rho(\cdot,\cdot)$ is positive semi-definite if ρ_S is positive semi-definite. Our proof, which is in the Appendix, follows closely that of (Paciorek, 2003, Theorem 1), but sharpens the result using a characterization of positive definite kernels proved in (Wendland, 2004, Theorem 7.14).

Proposition 1 *Let Assumptions 1 hold. Suppose $\Sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ is such that $\Sigma(z)$ is symmetric positive definite for all $z \in \mathbb{R}^d$, and define the quadratic form*

$$Q(x,x') = (x-x')^T \left(\frac{\Sigma(x)+\Sigma(x')}{2}\right)^{-1}(x-x'), \qquad x,x' \in \mathbb{R}^d.$$

Then the function $\rho(\cdot,\cdot)$, defined by

$$\rho(x,x') = \frac{2^{\frac{d}{2}} |\Sigma(x)|^{\frac{1}{4}} |\Sigma(x')|^{\frac{1}{4}}}{|\Sigma(x)+\Sigma(x')|^{\frac{1}{2}}} \rho_S\Big(\sqrt{Q(x,x')}\Big),$$

is positive definite on $\mathbb{R}^d \times \mathbb{R}^d$, for any $d \in \mathbb{N}$, and is a non-stationary correlation function.

1. If the double sum in this definition is only non-negative, we say that the kernel ρ_S is positive semi-definite. We are thus adopting the terminology used by Wendland (2004), where the kernel ρ_S is called positive definite if the double sum in Assumptions 1(i) is positive, and positive semi-definite if the sum is non-negative. For historical reasons, there is an alternative terminology, used by for example Paciorek (2003), where our notion of positive definite is referred to as strictly positive definite, and our notion of positive semi-definite is referred to as positive definite.

Non-stationary covariance functions $c(x,y)$, for which $c(x,x) \neq 1$, can be obtained from the non-stationary correlation function $\rho(x,y)$ through multiplication by a standard deviation function $\sigma : \mathbb{R}^d \to \mathbb{R}$, in which case we have $c(x,y) = \sigma(x)\sigma(y)\rho(x,y)$. Since the product of two positive definite kernels is also positive definite by (Wendland, 2004, Theorem 6.2), the kernel $c(x,y)$ can be ensured to be positive definite by a proper choice of σ. We discuss generalizations such as this in the conclusions section 5.

We are interested in studying the behaviour of Gaussian processes with non-stationary correlation functions $\rho(x,y)$ of the form derived in Proposition 1, in the particular case where the matrices $\Sigma(z)$ are derived from another Gaussian process. Specifically, we consider the following hierarchy of conditionally Gaussian processes on a bounded domain $D \subseteq \mathbb{R}^d$ defined as follows:

$$u_0 \sim \mathrm{GP}(0, \rho_{\mathrm{S}}(\cdot)), \tag{4a}$$

$$u_{n+1}|u_n \sim \mathrm{GP}(0, \rho(\cdot,\cdot;u_n)), \quad \text{for } n \in \mathbb{N}. \tag{4b}$$

Here, $\rho(\cdot,\cdot;u_n)$ denotes a non-stationary correlation function constructed from $\rho_{\mathrm{S}}(\cdot)$ as in Proposition 1, with the map Σ defined through u_n. Typical choices for Σ are $\Sigma(z) = (u_n(z))^2\,\mathrm{I}_d$ and $\Sigma(z) = \exp(u_n(z))\,\mathrm{I}_d$. Choices such as the first of these lead to the possibility of positive semi-definite Σ and, in the worst case, $\Sigma \equiv 0$. If $\Sigma \equiv 0$ the resulting correlation function is given by

$$\rho_{\mathrm{S}}(0) = 1, \quad \text{and} \quad \rho_{\mathrm{S}}(r) = 0 \quad \text{for any } r > 0.$$

This does not correspond to any (function valued) Gaussian process on $\mathbb{R}^d$ (Kallianpur, 2013): heuristically the resulting process would be a white noise process, but normalized to zero. However, it is possible to sample from any set of finite dimensional distributions when $\Sigma \equiv 0$: the correlation matrix is then the identity. To allow for the possibility of $F(\cdot)$ taking the value zero, we therefore only study the finite dimensional process defined as follows:

$$\mathbf{u}_0 \sim N(0, \mathbf{R}_{\mathrm{S}}), \tag{5a}$$

$$\mathbf{u}_{n+1} \mid \mathbf{u}_n \sim N(0, \mathbf{R}(\mathbf{u}_n)), \quad \text{for } n \in \mathbb{N}. \tag{5b}$$

The vector $\mathbf{u}_n$ has entries $(\mathbf{u}_n)_i = u_n(x_i)$. Here, $\mathbf{R}_{\mathrm{S}}$ is the covariance matrix with entries $(\mathbf{R}_{\mathrm{S}})_{ij} = \rho_{\mathrm{S}}(\|x_i - x_j\|_2)$, and $\mathbf{R}(\mathbf{u}_n)$ is the covariance matrix with entries $(\mathbf{R}(\mathbf{u}_n))_{ij} = \rho(x_i, x_j; u_n)$. The set $\{x_j\}$ comprises a finite set of points in $\mathbb{R}^d$.

We may now generalize Proposition 1 to allow for Σ becoming zero. In order to do this we make the following assumptions:

Assumptions 2 *(i) We have $\Sigma(z) = G(z)\mathrm{I}_d$, for some non-negative, bounded function $G : \mathbb{R} \to \mathbb{R}_{\geq 0}$.*

(ii) The correlation function ρ_{S} is continuous, with $\lim_{r\to\infty} \rho_{\mathrm{S}}(r) = 0$.

We then have the following result on the positive-definiteness of $\rho(\cdot,\cdot)$,

Proposition 2 *Let Assumptions 1 and 2 hold. Then the kernel $\rho(\cdot,\cdot)$ defined in Proposition 1 is positive definite on $\mathbb{R}^d \times \mathbb{R}^d$.*

Remark 3 *This proposition applies to the process (5) with $\Sigma(z) = F\big(u_n(z)\big)\mathrm{I}_d$ and $F : \mathbb{R} \to \mathbb{R}_{\geq 0}$ locally bounded, by taking $G = F \circ u_n$, proving that $\rho(\cdot,\cdot;u_n)$ is positive definite on $D \times D$ for all bounded functions u_n on D. Here we generalize the notion of positive-definite in the obvious way to apply on $D \subseteq \mathbb{R}^d$ rather than on the whole of $\mathbb{R}^d$.*

2.3 Covariance Operator

Here we demonstrate how precision (inverse covariance) operators may be used to make deep Gaussian processes. Because precision operators encode conditional independence and sparsity this can be a very attractive basis for fast computations (Lindgren et al., 2011). Our approach is inspired by the hierarchical Gaussian process introduced by Roininen et al. (2016), where one-step of the Markov chain which we introduce here was considered. Let $D \subseteq \mathbb{R}^d$, $u_n : D \to \mathbb{R}$ and $X := C(D;\mathbb{R})$. Assume that $F : \mathbb{R} \to \mathbb{R}_{\geq 0}$ is a bounded function. Let C_- be a covariance operator associated to a Gaussian process taking values in X and let P be the associated precision operator. Define the multiplication operator $\Gamma(u)$ by $\big(\Gamma(u)v\big)(x) = F\big(u(x)\big)v(x)$ and the covariance operator $C(u)$ by

$$C(u)^{-1} = P + \Gamma(u)$$

and consider the Markov chain (CovOp) with mean zero; this defines our deep Gaussian process. We note that formulation (GP) can be obtained by observing that the covariance function $c(u) := c(x,x';u)$ is the Green's function associated with the precision operator for $C(u)$:

$$C(u)^{-1}c(\cdot,x';u) = \delta_{x'}(\cdot)$$

where $\delta_{x'}$ is a Dirac delta function centred at point x'. Computationally we will typically choose P to be a differential operator, noting that then fast methods may be employed to sample the Gaussian process $u_{n+1}|u_n$ by means of SPDEs (Lindgren et al., 2011; Dashti and Stuart, 2017). If P is chosen as a differential operator, then the order of this operator will be related to the order of regularity of samples, and F will be related to the length scale of the samples. These relations are made explicit in the case of certain Whittle-Matérn distributions when F is constant (Lindgren et al., 2011); some boundary effects may be present when $D \neq \mathbb{R}^d$, though methodology is available to ameliorate these (Daon and Stadler, 2018). As in the previous subsection, the form (ZeroMean) follows with $L(u) = C(u)^{\frac{1}{2}}$ and ξ_{n+1} being white noise.

Generalizations of the construction in this subsection are possible, and we highlight these in subsection 5; however for expository purposes we confine our analysis to the setting described in this subsection. For theoretical investigation of the equivalence, as measures, of Gaussians defined by addition of an operator to a given precision operator (see Pinski et al., 2015).

2.4 Convolution

We consider the case (ZeroMean) where $L(u)\xi := u * \xi$ is a convolution. To be concrete we let $D = [0,1]^d$ and construct a sequence of functions $u_n : D \to \mathbb{R}$ (or $u_n : D \to \mathbb{C}$) defined via the iteration

$$u_{n+1}(x) = (u_n * \xi_{n+1})(x) := \int_{[0,1]^d} u_n(x-y)\xi_{n+1}(y)\,dy,$$

where $\{\xi_n\}$ are a sequence of i.i.d. centred real-valued Gaussian random functions on D. Here we implicitly work with periodic extension of u_n from D to the whole of $\mathbb{R}^d$ in order to define the convolution.

3. The Role of Ergodicity

The purpose of this section is to demonstrate that the iteration (ZeroMean) is, in many situations, ergodic. This has the practical implication that the effective depth of the deep Gaussian process is limited by the mixing time of the Markov chain. In some cases the ergodic behaviour may be trivial (convergence to a constant). Furthermore, even if the chain is not ergodic, the large iteration number dynamics may blow-up, prohibiting use of the iteration at significant depth. The take home message is that in many cases the effective depth is not that great. Great care will be needed to design deep Gaussian processes whose depth, and hence approximation power, is substantial. This issue was first identified by Duvenaud et al. (2014), and we here provide a more general analysis of the phenomenon within the broad framework we have introduced for deep Gaussian processes.

3.1 Composition

We first consider the case where the iteration is defined by (3), which includes examples considered by Damianou and Lawrence (2013); Duvenaud et al. (2014). Duvenaud et al. (2014) observed that after a number of iterations, sample paths are approximately piecewise constant. We investigate this effect in the context of ergodicity. We first make two observations:

(i) if u_0 is piecewise constant, then u_n is piecewise constant for all $n \in \mathbb{N}$;

(ii) if u_0 has discontinuity set Z_0, and Z_n denotes the discontinuity set of the nth iterate, then $\mathsf{Z}_{n+1} \subseteq \mathsf{Z}_n$ for all $n \in \mathbb{N}$.

Due to point (ii) above, if the sequence $\{u_n\}$ is to be ergodic, then necessarily it must be the case that $\mathsf{Z}_n \to \varnothing$, or else the process will have retained knowledge of the initial condition. In particular, if the initial condition is piecewise constant, then ergodicity would force the limit to be constant in space.

In what follows we assume that the iteration is given by

$$u_{n+1}(x) = \xi_{n+1}\big(u_n(x)\big), \quad \xi^j_{n+1} \sim \mathrm{GP}\big(0, h(\|x - x'\|_2)\big) \text{ i.i.d.}$$

where h is a stationary covariance function. We therefore make the choice $m = l$ and $F = \mathrm{id}$ in (3) so that we are in the same setup as Damianou and Lawrence (2013); Duvenaud et al. (2014); the inclusion of more general maps F is discussed in Remark 5. Then for any $x, x' \in \mathbb{R}$ we have

$$\begin{pmatrix} u^j_{n+1}(x) \\ u^j_{n+1}(x') \end{pmatrix} \Bigg| u_n \sim N\Bigg(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} h(0) & h\big(\|u_n(x) - u_n(x')\|_2\big) \\ h\big(\|u_n(x) - u_n(x')\|_2\big) & h(0) \end{pmatrix} \Bigg).$$

A common choice of covariance function is the squared exponential kernel:

$$h(z) = \sigma^2 e^{-z^2/2w^2} \tag{6}$$

where $\sigma^2, w^2 > 0$ are scalar parameters. In Duvenaud et al. (2014), in the case $m = d = 1$, the choice $\sigma^2/w^2 = \pi/2$ is made above to ensure that the expected magnitude of the derivative remains constant through iterations. We show in the next proposition that if σ^2, w^2 are chosen such that $\sigma^2 < w^2/m$, then the limiting process is trivial in a sense to be made precise.

Theorem 4 *Assume that $h(\cdot)$ is given by the squared exponential kernel (6) and that u_0 is bounded on bounded sets almost-surely. Then if $\sigma^2 < w^2/m$,*

$$\mathbb{P}\big(\|u_n(x) - u_n(x')\|_2 \to 0 \text{ for all } x, x' \in D\big) = 1$$

where $\mathbb{P}$ denotes the law of the process $\{u_n\}$ over the probability space Ω.

Proof Since $1 - e^{-x} \leq x$ for $x \geq 0$ it follows that, for all $z \in \mathbb{R}$,

$$2h(0) - 2h(z) \leq \frac{\sigma^2}{w^2} z^2,$$

with equality when $z = 0$. Then we have

$$\begin{aligned} \mathbb{E}\big(\|u_n(x) - u_n(x')\|_2^2 \big| u_{n-1}\big) &= \sum_{j=1}^m \mathbb{E}\big(|u^j_n(x) - u^j_n(x')|^2 \big| u_{n-1}\big) \\ &= \sum_{j=1}^m \big(2h(0) - 2h\big(\|u_{n-1}(x) - u_{n-1}(x')\|_2\big)\big) \\ &\leq m\frac{\sigma^2}{w^2}\|u_{n-1}(x) - u_{n-1}(x')\|_2^2 \end{aligned}$$

and so using induction and the tower property of conditional expectations,

$$\begin{aligned} \mathbb{E}\|u_n(x) - u_n(x')\|_2^2 &\leq \left(\frac{m\sigma^2}{w^2}\right)\mathbb{E}\|u_{n-1}(x) - u_{n-1}(x')\|_2^2 \\ &\leq \left(\frac{m\sigma^2}{w^2}\right)^n \mathbb{E}\|u_0(x) - u_0(x')\|_2^2 \\ &\leq \left(\frac{m\sigma^2}{w^2}\right)^n \kappa(x, x') \end{aligned}$$

for some constant $\kappa(x, x')$. By the Markov inequality, we see that for any $\varepsilon > 0$,

$$\mathbb{P}\big(\|u_n(x) - u_n(x')\|_2 \geq \varepsilon\big) \leq \frac{1}{\varepsilon^2}\left(\frac{m\sigma^2}{w^2}\right)^n \kappa(x, x'), \tag{7}$$

and so, since $\sigma^2 < w^2/m$, applying the first Borel-Cantelli lemma we deduce that

$$\mathbb{P}\left(\bigcap_{n=1}^\infty \bigcup_{m=n}^\infty \{\|u_m(x) - u_m(x')\|_2 \geq \varepsilon\}\right) = 0.$$

We therefore have that

$$\begin{aligned}\mathbb{P}(\|u_n(x)-u_n(x')\|_2\to 0) &= \mathbb{P}\left(\bigcap_{k=1}^{\infty}\bigcup_{n=1}^{\infty}\bigcap_{m=n}^{\infty}\{\|u_m(x)-u_m(x')\|_2<1/k\}\right)\\ &= 1-\mathbb{P}\left(\bigcup_{k=1}^{\infty}\left(\bigcap_{n=1}^{\infty}\bigcup_{m=n}^{\infty}\{\|u_m(x)-u_m(x')\|_2\geq 1/k\}\right)\right)\\ &\geq 1-\sum_{k=1}^{\infty}\mathbb{P}\left(\bigcap_{n=1}^{\infty}\bigcup_{m=n}^{\infty}\{\|u_m(x)-u_m(x')\|_2\geq 1/k\}\right)=1.\end{aligned}$$

Hence given any $x,x'\in D$, we can find $\Omega(x,x')\subseteq\Omega$ with $\mathbb{P}(\Omega(x,x'))=1$ such that for any $\omega\in\Omega(x,x')$, $\|u_n(x;\omega)-u_n(x';\omega)\|_2\to 0$. Let $\{q_j\}$ be a countable dense subset of D, and define

$$\Omega_* = \bigcap_{i,j\in\mathbb{N}}\Omega\left(q_i,q_j\right),$$

noting that $\mathbb{P}(\Omega_*)=1$. Then for any $\omega\in\Omega_*$ and any $x,x'\in\{q_j\}$, $\|u_n(x;\omega)-u_n(x';\omega)\|_2\to 0$. Since sample paths are almost-surely continuous, the above can be extended to all $x,x'\in D$, so that

$$\mathbb{P}\big(\|u_n(x)-u_n(x')\|_2\to 0 \text{ for all } x,x'\in D\big)=1.$$

■

Remark 5 *1. If a more general transformation map $F:\mathbb{R}^m\to D'$ is included, then the above result still holds provided we take $\sigma^2<w^2/(\|F'\|_\infty m)$. The convergence to a constant hence occurs when the length scale w is large or $\|F'\|_\infty$ is small (so each Gaussian random field doesn't change too rapidly across the domain), or when the amplitude σ is small (so inputs are not warped too far).*

2. The condition of the above theorem is less likely to be satisfied as the width m of each layer is increased, and so this triviality pathology is unlikely to arise for large m; this may be observed in practice numerically.

3. Following Neal (1995); Duvenaud et al. (2014), recent works (such as Dai et al., 2015; Cutajar et al., 2017) connect all layers to the input layer in order to avoid certain pathologies. The Markovian structure of the process is maintained in this case: with the above notation, the process is then defined by

$$u_{n+1}(x)=\xi_{n+1}(u_n(x),x),\quad \xi_{n+1}^j\sim\mathrm{GP}\big(0,h(\|x-x'\|_2)\big)\ i.i.d,$$

where now $\xi_n:\mathbb{R}^m\times\mathbb{R}^d\to\mathbb{R}^m$. Defining $\beta=m\sigma^2/w^2<1$, if $\sigma\geq 1$ we may use the same argument as the proof above to deduce that

$$\mathbb{E}\big(\|u_n(x)-u_n(x')\|_2^2|u_{n-1}\big)\leq\beta\|u_{n-1}(x)-u_{n-1}(x')\|_2^2+\beta\|x-x'\|_2^2,$$

which leads to

$$\mathbb{E}\|u_n(x)-u_n(x')\|_2^2\leq\beta^n\mathbb{E}\|u_0(x)-u_0(x')\|_2^2+\beta\left(\frac{1-\beta^n}{1-\beta}\right)\|x-x'\|_2^2.$$

The right hand side does not vanish as $n\to\infty$, and so we can no longer use the first Borel-Cantelli lemma to reach the same conclusion as the case where the layers are not connected to the input layer. This could provide some intuition as to why including the connection of each layer to the input layer provides greater stability than not doing so.

3.2 Covariance Function

In order to study ergodicity of the deep Gaussian process defined through covariance functions, we will restrict attention in the remainder of this subsection to hierarchies of finite-dimensional multivariate Gaussian random variables as in (5). Note that although we have here defined $\mathbf{u}_0\sim N(0,\mathbf{R}_S)$, following e.g. Paciorek (2003), the ergodicity of the deep Gaussian process will be proved for fixed $\mathbf{u}_0\in\mathbb{R}^N$ (cf. Theorem 8). The following result is immediate from Proposition 2.

Corollary 6 *Let Assumptions 1 and 2 hold. Then the covariance matrix $\mathbf{R}(\mathbf{u}_n)$ is positive definite for all $\mathbf{u}_n\in C$, for any compact subset of $C\subseteq\mathbb{R}^N$.*

Note that, because we have chosen to work with a correlation kernel, we have

$$\mathrm{Tr}\big(\mathbf{R}(\mathbf{u}_n)\big)=N. \tag{8}$$

We will use this fact explicitly in the ergodicity proof; however it may be relaxed as discussed in the Remark 12 below.

We view the sequence of random variables $\{\mathbf{u}_n\}_{n=0}^\infty$ as a Markov chain, with $\mathbf{u}_0\in\mathbb{R}^N$ given, and we want to show the existence of a stationary distribution. Recall the one-step transition kernel P of the Markov chain given by (1), and its $n-$fold composition given by (2). In order to prove ergodicity of the Markov chain we will follow the proof technique of Mattingly et al. (2002); Meyn and Tweedie (2012), which establishes geometric ergodicity with the following proposition.

Proposition 7 *Suppose the Markov chain $\{\mathbf{u}_n\}_{n=0}^\infty$ satisfies, for some compact set $C\in\mathcal{B}(\mathbb{R}^N)$, the following:*

(i) For some $y^*\in int(C)$ and for any $\delta>0$, we have

$$\mathsf{P}(u,\mathcal{B}_\delta(y^*))>0\qquad \text{for all } u\in C.$$

(ii) The transition kernel $\mathsf{P}(u,\cdot)$ possesses a density $p(u,y)$ in C, precisely

$$\mathsf{P}(u,A)=\int_A p(u,y)\,\mathrm{d}y,\quad \text{for all } u\in C,\ A\in\mathcal{B}(\mathbb{R}^N)\cap\mathcal{B}(C),$$

and $p(u,y)$ is jointly continuous on $C\times C$.

(iii) There is a function $V : \mathbb{R}^N \to [1,\infty)$, with $\lim_{u\to\infty} V(u) = \infty$, and real numbers $\alpha \in (0,1)$ and $\beta \in [0,\infty)$ such that

$$\mathbb{E}(V(\mathbf{u}_{n+1}) \,|\, \mathbf{u}_n) \leq \alpha V(\mathbf{u}_n) + \beta.$$

If we can choose the compact set C such that

$$C = \left\{ u : V(u) \leq \frac{2\beta}{\gamma - \alpha} \right\},$$

for some $\gamma \in (\sqrt{\alpha}, 1)$, then there exists a unique invariant measure π. Furthermore, there is $r(\gamma) \in (0,1)$ and $\kappa(\gamma) \in (0,\infty)$ such that for all $\mathbf{u}_0 \in \mathbb{R}^N$ and all measurable g with $|g(u)| \leq V(u)$ for all $u \in \mathbb{R}^N$, we have

$$|\mathbb{E}^{\mathsf{P}^n(\mathbf{u}_0,\cdot)}(g) - \pi(g)| \leq \kappa r^n V(\mathbf{u}_0).$$

We may verify the assumptions of Proposition 7 leading to the following theorem concerning the ergodicity of deep Gaussian processes defined via the covariance function:

Theorem 8 *Suppose Assumptions 1 and 2 hold. Then the Markov chain $\{\mathbf{u}_n\}_{n=0}^{\infty}$ satisfies the assumptions of Proposition 7. As a consequence, there exists $\varepsilon \in (0,1)$ such that for any $\mathbf{u}_0 \in \mathbb{R}^N$, there is a $K(\mathbf{u}_0) > 0$ with*

$$\|\mathsf{P}^n(\mathbf{u}_0,\cdot) - \pi\|_{TV} \leq K(1-\varepsilon)^n \quad \text{for all } n \in \mathbb{N},$$

and so the chain is ergodic.

The proof rests on the following three lemmas, and is given after stating and proving them. The first lemma shows that, on average, the norm of states of the chain remains constant as the length of the chain is increased. The second shows that, given any current state in $\mathbb{R}^N$ and any ball around the origin in $\mathbb{R}^N$, there is a positive probability that the next state will belong to that ball. The third lemma shows that the probability that the Markov chain moves to a set may be found via integration of a continuous function over that set.

Lemma 9 (Boundedness) *Suppose Assumptions 1 and 2 hold. For all $n \in \mathbb{N}$, we have*

$$\mathbb{E}\big(\|\mathbf{u}_{n+1}\|_2^2 \,|\, \mathbf{u}_n\big) = N.$$

Proof Let $n \geq 0$. Since the random variable $\mathbf{u}_{n+1}|\mathbf{u}_n$ has zero mean, the linearity of expectation implies (using (8)) that

$$\mathbb{E}\left(\|\mathbf{u}_{n+1}\|_2^2 \,|\, \mathbf{u}_n\right) = \mathbb{E}\left(\sum_{j=1}^{N} (\mathbf{u}_{n+1})_j^2 \,\middle|\, \mathbf{u}_n \right) = \mathrm{Tr}(\mathbf{R}(\mathbf{u}_n)) = N,$$

for all $n \in \mathbb{N}$. ■

Lemma 10 (Positive probability of a ball around zero) *Suppose Assumptions 1 and 2 hold. For all $u \in \mathbb{R}^N$ and $\delta > 0$, we have*

$$\mathsf{P}\big(u, \mathcal{B}_\delta(0)\big) > 0.$$

Proof We have the equality $\mathbf{u}_{n+1}|(\mathbf{u}_n = u) = \sqrt{\mathbf{R}(u)}\xi_{n+1}$ in distribution, where $\sqrt{\mathbf{R}(u)}$ denotes the Cholesky factor of the correlation matrix $\mathbf{R}(u)$ and $\xi_{n+1} \sim \mathrm{N}(0, \mathrm{I}_N)$. Then

$$\begin{aligned}
\mathsf{P}\big(u, \mathcal{B}_\delta(0)\big) &= \mathbb{P}\big(\|\mathbf{u}_n\|_2 \leq \delta \,|\, \mathbf{u}_{n-1} = u\big) \\
&= \mathbb{P}\Big(\big\|\sqrt{\mathbf{R}(u)}\xi_{n+1}\big\|_2 \leq \delta\Big) \\
&\geq \mathbb{P}\Big(\big\|\sqrt{\mathbf{R}(u)}\big\|_2 \big\|\xi_{n+1}\big\|_2 \leq \delta\Big) \\
&= \mathbb{P}\Big(\big\|\xi_{n+1}\big\|_2 \leq \delta \big\|\sqrt{\mathbf{R}(u)}\big\|_2^{-1}\Big).
\end{aligned}$$

To show that the latter probability is positive, we need to show that $\delta\big\|\sqrt{\mathbf{R}(u)}\big\|_2^{-1} > 0$. Since $\delta > 0$ is fixed, we only need to show $\big\|\sqrt{\mathbf{R}(u)}\big\|_2 < \infty$. Since $\big\|\sqrt{\mathbf{R}(u)}\big\|_2^2 = \rho(\mathbf{R}(u))$, the spectral radius of $\mathbf{R}(u)$, we have

$$\big\|\sqrt{\mathbf{R}(u)}\big\|_2^2 = \rho(\mathbf{R}(u)) \leq \mathrm{Tr}(\mathbf{R}(u)) = N.$$

The claim then follows. ■

Lemma 11 (Transition probability has a density) *Suppose Assumptions 1 and 2 hold. Then the transition probability $\mathsf{P}\big(u,\cdot\big)$ has a jointly continuous density $p(u,y)$ for all $u \in C$, for any compact set $C \subseteq \mathbb{R}^N$.*

Proof We have $\mathbf{u}_{n+1}|(\mathbf{u}_n = u) \sim N(0, \mathbf{R}(u))$, and the existence of a jointly continuous density of the transition probability in C follows if $\mathbf{R}(u)$ is positive definite for all $u \in C$. The claim then follows by Proposition 2. ■

We may now use the three preceding lemmas to prove the main ergodic theorem for deep Gaussian processes defined through the covariance function.

Proof of Theorem 8 Lemma 10 shows that assumption (i) is satisfied, for any C containing $y^* = 0$, and Lemma 11 shows that assumption (ii) is satisfied, for any compact set C. It follows from Lemma 9 that assumption (iii) is satisfied, with $V(u) = \|u\|_2^2 + 1$, any $\alpha \in (0,1)$ and $\beta = N + 1$. Now choose $\alpha = 1/4$ and $\gamma = 3/4 \in (\sqrt{\alpha}, 1)$, so that the set

$$C = \left\{ u : V(u) \leq \frac{2\beta}{\gamma - \alpha} \right\} = \big\{ u : \|u\|_2^2 \leq 4N + 4 \big\}$$

is compact. Then there is a unique invariant measure π, and there is $r(\gamma) \in (0,1)$ and $\kappa(\gamma) \in (0,\infty)$ such that for $\mathbf{u}_0 \in \mathbb{R}^N$ and all measurable g with $|g(u)| \leq V(u)$ for all $u \in \mathbb{R}^N$, we have

$$|\mathbb{E}^{\mathsf{P}^n(\mathbf{u}_0,\cdot)}(g) - \pi(g)| \leq \kappa r^n V(\mathbf{u}_0). \tag{9}$$

Since $V(u) \geq 1$ for all $u \in \mathbb{R}^N$, the above holds in particular for all measurable g with $\|g\|_\infty \leq 1$. Taking the supremum over all such g in (9) yields the given total variation bound, with $K = \kappa V(\mathbf{u}_0)$ and $\varepsilon = 1 - r$. ■

Remark 12 *(Covariance vs correlation kernels) In this subsection we have restricted our attention to correlation kernels $\rho_S(\|x_i - x_j\|_2)$ and $\rho(x_i, x_j; u_n)$, rather than more general covariance kernels*

$$c_S(\|x_i - x_j\|_2) = \sigma_S^2 \rho_S(\|x_i - x_j\|_2),$$
$$c(x_i, x_j; u_n) = \sigma(x_i; u_n)\sigma(x_j; u_n)\rho(x_i, x_j; u_n),$$

for stationary and non-stationary marginal standard deviation functions $\sigma_S \in (0, \infty)$ and $\sigma : \mathbb{R}^d \to (0, \infty)$ respectively. This restriction is solely for ease of presentation; the analysis presented readily extends to $c(x_i, x_j; u_n)$, under suitable assumptions on σ. In particular the analysis may be adapted to the case of general covariance kernels $c_S(\|x_i - x_j\|_2)$ and $c(x_i, x_j; u_n)$ under the assumption that there exist positive constants σ^-, σ^+ such that $\sigma^- \leq \sigma(z) \leq \sigma^+$, for all $z \in \mathbb{R}^d$. When general covariances are used then it is possible to ensure that every multivariate Gaussian random variable in the hierarchy is of the same amplitude by scaling the corresponding covariance matrix $\mathbf{C}(\mathbf{u}_n)$ to have constant trace N at each iteration n; the average variance over all points $\{x_i\}_{i=1}^N$ is then 1 for every n.

3.3 Covariance Operator

We consider the class of covariance operators introduced in section 2.3 and show that, under precise assumptions detailed below, the iteration (ZeroMean) produces an ergodic Markov chain. Unlike the previous subsection, where we worked on $\mathbb{R}^N$, here we will work on the separable Hilbert space $\mathcal{H} = L^2(D; \mathbb{R})$. To begin with, define the precision operators (densely defined on $\mathcal{H}$: Hairer et al., 2005; Pinski et al., 2015),

$$C_-^{-1} = P,$$
$$C_+^{-1} = P + F_+ I,$$
$$C(u)^{-1} = P + \Gamma(u), \quad u \in \mathcal{H},$$

and the probability measures

$$\mu_- = N(0, C_-),$$
$$\mu_+ = N(0, C_+),$$
$$\mu(\cdot\,; u) = N(0, C(u)), \quad u \in \mathcal{H}.$$

Throughout the rest of this section we make the following assumptions on C_- and F:

Assumptions 3 *1. The operator $C_- : \mathcal{H} \to \mathcal{H}$ is symmetric and positive, and its eigenvalues $\{\lambda_j^2\}$ have algebraic decay $\lambda_j^2 \asymp j^{-r}$ for some $r > 1$.*

2. The function $F : \mathcal{H} \to \mathbb{R}$ is continuous, and there exists $F_+ \geq 0$ such that $0 \leq F(u) \leq F_+$ for all $u \in \mathcal{H}$.

Remark 13 *1. The assumption on algebraic decay of the eigenvalues can be relaxed to the operator C_- being trace-class on $\mathcal{H}$; however the arguments that follow are cleaner when we assume this explicit decay which, of course, implies the trace condition. Note also that, under the stated assumption on algebraic decay, Gaussian measures on $L^2(D; \mathbb{R})$ will be supported on $X = C(D; \mathbb{R})$ under mild conditions on the eigenfunctions of C_- (Dashti and Stuart, 2017) so that $F\big(u(x)\big)$ will be defined for all $x \in D$ rather than x a.e. in D. Then $\Gamma(u)v$ makes sense pointwise when $v \in X$.*

2. The assumed form of the precision operator together with Assumptions 3 mean that the resulting family of measures $\{\mu(\cdot; u)\}_{u \in \mathcal{H}}$ will be mutually equivalent. This allows for the total variation metric between measures to be used, and a concise proof of ergodicity to be obtained. If the measures were singular, a different metric such as the Wasserstein metric would be required to quantify the convergence.

We now prove the following ergodic theorem for the deep Gaussian processes constructed through covariance operators.

Theorem 14 *Let Assumptions 3 hold, and let the Markov chain $\{u_n\}$ be given by (ZeroMean) with $L(u) = C(u)^{\frac{1}{2}}$ as defined above. Then there exists a unique invariant distribution π, and there exists $\varepsilon > 0$ such that for any $u_0 \in \mathcal{H}$,*

$$\|\mathsf{P}^n(u_0, \cdot) - \pi\|_{TV} \leq (1 - \varepsilon)^n \quad \text{for all } n \in \mathbb{N}.$$

In particular, the chain is ergodic.

The following lemma will be used to show a minorization condition, as well as establish further notation, key to the proof of Theorem 14 which follows it. It essentially shows a stronger form of equivalence of the family of measures $\{\mu(\cdot, u)\}_{u \in \mathcal{H}}$.

Lemma 15 *Let Assumptions 3 hold. Then there exists $\varepsilon > 0$ such that for any $u, v \in \mathcal{H}$,*

$$\frac{\mathrm{d}\mu(\cdot\,; u)}{\mathrm{d}\mu_+}(v) \geq \varepsilon.$$

Proof The assumptions on F mean that the measures $\mu(\cdot\,; u)$, μ_- and μ_+ are mutually absolutely continuous, with

$$\frac{\mathrm{d}\mu(\cdot\,; u)}{\mathrm{d}\mu_-}(v) = \frac{1}{Z(u)} \exp\Big(-\frac{1}{2}\langle v, F(u)v\rangle\Big),$$
$$Z(u) = \mathbb{E}^{\mu_-}\Big[\exp\Big(-\frac{1}{2}\langle v, F(u)v\rangle\Big)\Big];$$
$$\frac{\mathrm{d}\mu_+}{\mathrm{d}\mu_-}(v) = \frac{1}{Z_+} \exp\Big(-\frac{1}{2}\langle v, F_+ v\rangle\Big),$$
$$Z_+ = \mathbb{E}^{\mu_-}\Big[\exp\Big(-\frac{1}{2}\langle v, F_+ v\rangle\Big)\Big].$$

Observe that we may bound $Z(u) \leq 1$ uniformly in $u \in H$ since $F \geq 0$. Additionally, we have that

$$Z_+ \geq \mathbb{E}^{\mu_-}\left[\exp\left(-\frac{1}{2}\langle v, F_+ v\rangle\right)\mathbb{1}_{\|v\|^2\leq 1}\right] \geq \exp\left(-\frac{1}{2}F_+\right)\mu_-\big(\|v\|^2 \leq 1\big) =: \varepsilon > 0.$$

Note that ε is positive since $\mathcal{H}$ is separable, and thus all balls have positive measure (Hairer, 2009). It follows that

$$\begin{aligned}\frac{\mathrm{d}\mu(\cdot\,;u)}{\mathrm{d}\mu_+}(v) &= \frac{\mathrm{d}\mu(\cdot\,;u)}{\mathrm{d}\mu_-}(v) \times \left(\frac{\mathrm{d}\mu_+}{\mathrm{d}\mu_-}(v)\right)^{-1} \\ &= \frac{1}{Z(u)}\exp\left(-\frac{1}{2}\langle v, F(u)v\rangle\right) \times Z_+ \exp\left(\frac{1}{2}\langle v, F_+ v\rangle\right) \\ &\geq \varepsilon \exp\left(\frac{1}{2}\langle v, (F_+ - F(u))v\rangle\right) \\ &\geq \varepsilon\end{aligned}$$

since F_+ bounds F above uniformly. ■

Proof of Theorem 14 We first establish existence of at least one invariant distribution by showing that chain $\{u_n\}$ is (strong) Feller, and that for each $u_0 \in \mathcal{H}$ the family $\{\mathsf{P}^n(u_0, \cdot)\}$ of transition kernels is tight. To see the former, let $f : \mathcal{H} \to \mathbb{R}$ be any bounded measurable function. We have that, for any $v \in \mathcal{H}$,

$$\begin{aligned}(\mathsf{P}f)(u) &:= \int_{\mathcal{H}} f(v)\mathsf{P}(u, \mathrm{d}v) \\ &= \int_{\mathcal{H}} f(v)\frac{1}{Z(u)}\exp\left(-\frac{1}{2}\langle v, F(u)v\rangle\right)\mu_-(\mathrm{d}v).\end{aligned}$$

Since $F(u) \leq F_+$ it follows that $Z(u)$ is bounded below by a positive constant, uniformly with respect to u. Additionally F is continuous and non-negative, and so the integrand is bounded and continuous with respect to u. Hence given any sequence $u^{(k)} \to u$ in $\mathcal{H}$, we may apply the dominated convergence theorem to see that $(\mathsf{P}f)(u^{(k)}) \to (\mathsf{P}f)(u)$. The function $\mathsf{P}f$ is therefore continuous, and so the chain $\{u_n\}$ is strong Feller.

We now show tightness. The assumptions on the operator $C_- : \mathcal{H} \to \mathcal{H}$ imply that it is trace-class, and so in particular compact. It is also positive and symmetric, and so by the spectral theorem, admits a complete orthonormal system of eigenvectors $\{\varphi_j\}$ with corresponding positive eigenvalues $\{\lambda_j^2\}$ such that $\lambda_j^2 \to 0$. Given $s > 0$, define the subspace $\mathcal{H}^s \subset \mathcal{H}$ by

$$\mathcal{H}^s = \left\{ v \in \mathcal{H} \;\middle|\; \|v\|_{\mathcal{H}^s}^2 := \sum_{j=1}^{\infty} j^{2s}|\langle \varphi_j, v\rangle|^2 < \infty \right\}.$$

It is standard to show that $\mathcal{H}^s$ is compactly embedded in $\mathcal{H}$ for any $s > 0$ (see for example Robinson, 2001, Appendix A.2). By the Karhunen-Loéve theorem, any $v \sim \mu_-$ may be represented as

$$v = \sum_{j=1}^{\infty} \lambda_j \xi_j \varphi_j, \quad \xi_j \sim N(0,1) \text{ i.i.d.}$$

Hence, by the orthonormality of the $\{\varphi_j\}$ and the assumed decay of the eigenvalues, we have that

$$\mathbb{E}^{\mu_-}\big(\|v\|_{\mathcal{H}^s}^2\big) = \sum_{j=1}^{\infty} j^{2s}\lambda_j^2 \asymp \sum_{j=1}^{\infty} j^{2s-r}$$

and so

$$\mathbb{E}^{\mu_-}\big(\|v\|_{\mathcal{H}^s}^2\big) < \infty \quad \text{if and only if} \quad s < \frac{r}{2} - \frac{1}{2}.$$

Since $r > 1$ by assumption, we can always choose $s > 0$ such that this holds; fix such an s in what follows. Observe that, for any $n \in \mathbb{N}$,

$$\begin{aligned}\mathbb{E}\big(\|u_n\|_{\mathcal{H}^s}^2\big) &= \mathbb{E}\big(\mathbb{E}\big(\|u_n\|_{\mathcal{H}^s}^2 \big| u_{n-1}\big)\big) \\ &= \mathbb{E}\big(\mathbb{E}^{\mu(\cdot\,;u_{n-1})}\big(\|v\|_{\mathcal{H}^s}^2\big)\big) \\ &= \mathbb{E}\left(\int_{\mathcal{H}} \|v\|_{\mathcal{H}^s}^2 \frac{1}{Z(u_{n-1})}\exp\left(-\frac{1}{2}\langle v, F(u_{n-1})v\rangle\right)\mu_-(\mathrm{d}v)\right) \\ &\leq \frac{1}{Z_+}\mathbb{E}^{\mu_-}\big(\|v\|_{\mathcal{H}^s}^2\big) \\ &=: M < \infty.\end{aligned}$$

We have bounded $Z(u_{n-1}) \geq Z_+$ using that $F(u_{n-1}) \leq F_+$. Applying the Chebychev inequality, we have for each $n \in N$ and $R > 0$

$$\mathbb{P}\big(\|u_n\|_{\mathcal{H}^s} > R\big) \leq \frac{\mathbb{E}\big(\|u_n\|_{\mathcal{H}^s}^2\big)}{R^2} \leq \frac{M}{R^2},$$

and so given any $\kappa > 0$,

$$\mathbb{P}\left(\|u_n\|_{\mathcal{H}^s} \leq \sqrt{\frac{M}{\kappa}}\right) \geq 1 - \kappa.$$

This can be rewritten as

$$\mathsf{P}^n(u_0, K_\kappa) \geq 1 - \kappa$$

where $K_\kappa = \big\{u \in \mathcal{H} \,\big|\, \|u\|_{\mathcal{H}^s} \leq \sqrt{M/\kappa}\big\}$ is compact in $\mathcal{H}$, since $\mathcal{H}^s$ is compactly embedded in $\mathcal{H}$; this shows tightness of the sequence of probability measures $\mathsf{P}^n(u_0, \cdot)$. Since tightness implies boundedness in probability on average, an application of (Meyn and Tweedie, 2012, Theorem 12.0.1) gives existence of an invariant distribution.

Lemma 15 shows that $\{u_n\}$ satisfies a global minorization condition for the one-step transition probabilities: for any $u_0 \in \mathcal{H}$ and any measurable $A \subseteq \mathcal{H}$,

$$\mathsf{P}\big(u_0, A\big) = \mathbb{E}^{\mu(\cdot\,;u_0)}\big(\mathbb{1}_A(v)\big) = \mathbb{E}^{\mu_+}\left(\frac{\mathrm{d}\mu(\cdot\,;u_0)}{\mathrm{d}\mu_+}(v)\mathbb{1}_A(v)\right) \geq \varepsilon\mu_+(A).$$

Combined with the existence of an invariant distribution above, a short coupling argument (Meyn and Tweedie, 2012, Theorem 16.2.4) gives the result with the same ε as above. ■

3.4 Convolution

The convolution iteration has the advantage that, through use of Fourier series and the law of large numbers, its long time behaviour can be completely characterized analytically. We consider the convolution as a random map on $\mathcal{H} = L^2(D;\mathbb{C})$, $D = (0,1)^d$. The iteration is given by

$$u_{n+1}(x) = (u_n * \xi_{n+1})(x) := \int_D u_n(x-y)\xi_{n+1}(y)\,dy, \quad \xi_{n+1} \sim N(0,C) \text{ i.i.d.} \tag{10}$$

where we implicitly work with periodic extensions to define the convolution. We assume that C is a negative fractional power of a differential operator so that it diagonalizes in Fourier space; such a form of covariance operator is common in applications, as it includes, for example, Whittle-Matérn distributions (Lindgren et al., 2011). For example, we may take

$$C = (I - \triangle)^{-\alpha}, \quad D(-\triangle) = H^2_{\text{per}}([0,1]^d) \subset \mathcal{H},$$

in which case the samples $\xi_{n+1} \sim N(0,C)$ will (almost surely) possess s fractional Sobolev and Hölder derivatives for any $s < \alpha - d/2$ (see Dashti and Stuart, 2017, for details).

We choose the orthonormal Fourier basis

$$\varphi_k(x) = e^{2\pi i k\cdot x}, \quad k \in \mathbb{Z}^d,$$

which are the eigenvectors of C; we denote the corresponding eigenvalues $\{\lambda_k^2\}$. Given $u \in \mathcal{H}$ and $k \in \mathbb{Z}^d$, define the Fourier coefficient $\hat{u}(k) \in \mathbb{C}$ by

$$\hat{u}(k) := \langle \varphi_k, u\rangle_{L^2} = \int_D \overline{\varphi_k(x)}u(x)\,\mathrm{d}x.$$

Then it can be readily checked that for any $u, v \in \mathcal{H}$ and $k \in \mathbb{Z}^d$,

$$\widehat{(u*v)}(k) = \hat{u}(k)\hat{v}(k). \tag{11}$$

We use this property to establish the following theorem.

Theorem 16 *Let $C : \mathcal{H} \to \mathcal{H}$ be a negative fractional power of a differential operator such that C is positive, symmetric and trace-class, with eigenvectors $\{\psi_k\}$ and eigenvalues $\{\lambda_k^2\}$. Define the Markov chain $\{u_n\}$ by (10). Then for any $u_0 \in \mathcal{H}$,*

$$\lim_{n\to\infty} |\hat{u}_n(k)|^2 = \begin{cases} 0 & |\lambda_k|^2 < 2e^\gamma \\ \infty & |\lambda_k|^2 > 2e^\gamma \end{cases} \quad \textit{almost surely}$$

where $\gamma \approx 0.577$ is the Euler-Mascheroni constant. In particular, if $|\lambda_k|^2 < 2e^\gamma$ for all $k \in \mathbb{Z}^d$, then every Fourier coefficient of u_n tends to zero almost surely and hence $u_n \rightharpoonup 0$ in $\mathcal{H}$ almost surely.

Proof First observe that by the Karhunen-Loéve theorem, we may express $\xi_{n+1} \sim N(0,C)$ as

$$\xi_{n+1} = \sum_{k\in\mathbb{Z}^d} \lambda_k \eta_{n,k}\varphi_k, \quad \eta_{n,k} \sim N(0,1) \text{ i.i.d.}$$

and so, since $\{\varphi_k\}$ is orthonormal,

$$\hat{\xi}_{n+1}(k) = \lambda_k \eta_{n,k}.$$

Then by the property (11), we see that for each $k \in \mathbb{Z}^d$ and $n \in \mathbb{N}$,

$$\hat{u}_{n+1}(k) = \hat{u}_n(k)\hat{\xi}_{n+1}(k) = \hat{u}_n(k)\lambda_k\eta_{n,k} \tag{12}$$

where the second equality is in distribution. The problem has now been reduced to an independent family of scalar problems. We can write $\hat{u}_n(k)$ explicitly as

$$\hat{u}_n(k) = \hat{u}_0(k)\prod_{j=1}^n \lambda_k \eta_{j,k}. \tag{13}$$

Now observe that

$$\begin{aligned} |\hat{u}_n(k)|^2 &= |\hat{u}_0(k)|^2 \prod_{j=1}^n |\lambda_k|^2 |\eta_{j,k}|^2 \\ &= |\hat{u}_0(k)|^2 \exp\left(n \cdot \frac{1}{n}\sum_{j=1}^n \log\left(|\lambda_k|^2|\eta_{j,k}|^2\right)\right) \\ &= |\hat{u}_0(k)|^2 \exp\left(n \cdot \left(\frac{1}{n}\sum_{j=1}^n \log|\eta_{j,k}|^2 + \log|\lambda_k|^2\right)\right). \end{aligned} \tag{14}$$

By the strong law of large numbers, the scaled sum inside the exponential converges almost surely to $\mathbb{E}(\log|\eta_{1,k}|^2)$. This can be calculated as

$$\mathbb{E}(\log|\eta_{1,k}|^2) = -\gamma - \log 2.$$

If the bracketed term inside the exponential in (14) is eventually negative almost surely, then the limit of $|u_n(k)|^2$ will be zero almost surely. This is guaranteed when $-\gamma - \log 2 + \log|\lambda_k|^2 < 0$, i.e. $|\lambda_k|^2 < 2e^\gamma$. Similarly we get divergence if the bracketed term is eventually positive, which happens when $|\lambda_k|^2 > 2e^\gamma$. ■

Remark 17 *It is interesting to note that we may take expectations in (12) to establish that*

$$\mathbb{E}|\hat{u}_n(k)|^2 = |\hat{u}_0(k)|^2|\lambda_k|^{2n}$$

and so

$$\lim_{n\to\infty} \mathbb{E}|\hat{u}_n(k)|^2 = \begin{cases} 0 & |\lambda_k|^2 < 1 \\ \infty & |\lambda_k|^2 > 1 \end{cases}.$$

In particular, if $|\lambda_k|^2 \in (1, 2e^\gamma)$, then $|\hat{u}_n(k)|^2$ converges to zero almost surely, but diverges in mean square.

Via a slight modification of the above proof to account for different boundary conditions, we have the following result.

Corollary 18 *Let $D = (0,1)$ and let $\{u_n\}$ be defined by the iteration (10), where each ξ_{n+1} is a Brownian bridge. Then $u_n \rightharpoonup 0$ almost surely.*

Proof The Brownian bridge on $[0,1]$ has covariance operator $(-\triangle)^{-1}$, where

$$D(-\triangle) = \{u \in H^2_{\text{per}}([0,1]) \,|\, u(0) = u(1) = 0\}.$$

The result of Theorem 16 cannot be applied directly, since the basis functions $\{\varphi_k\}$ do not satisfy the boundary conditions. The eigenfunctions with the correct boundary conditions are given by

$$\psi_j(x) = \sqrt{2}\sin(j\pi x) = \frac{1}{\sqrt{2}i}\big(\varphi_j(x) - \varphi_{-j}(x)\big), \quad j \geq 1$$

with corresponding eigenvalues $\alpha_j^2 = (\pi^2 j^2)^{-1}$. A Brownian bridge $\xi_{n+1} \sim N(0, (-\triangle)^{-1})$ can then be expressed as

$$\xi_{n+1} = \sum_{j=1}^{\infty} \alpha_j \zeta_{n,j} \psi_j, \quad \zeta_{n,j} \sim N(0,1) \text{ i.i.d.}$$

by the Karhunen-Loéve theorem. We calculate

$$\begin{aligned}
\hat{u}_{n+1}(k) &= \hat{u}_n(k)\hat{\xi}_n(k) \\
&= \hat{u}_n(k) \sum_{j=1}^{\infty} \alpha_j \zeta_{n,j} \langle \varphi_k, \psi_j \rangle \\
&= \hat{u}_n(k) \sum_{j=1}^{\infty} \alpha_j \zeta_{n,j} \frac{1}{\sqrt{2}i} \big(\langle \varphi_k, \varphi_j \rangle - \langle \varphi_k, \varphi_{-j} \rangle \big) \\
&= \hat{u}_n(k) \frac{\text{sgn}(k)\alpha_{|k|}}{\sqrt{2}i} \zeta_{n,|k|} \\
&= \hat{u}_n(k) \lambda_k \eta_{n,k}, \quad \eta_{n,k} \sim N(0,1).
\end{aligned}$$

We can now proceed as in Theorem 16 to deduce that $|u_n(k)|^2 \to 0$ whenever $|\lambda_k|^2 < 2e^\gamma$; note that the correlations between $\hat{u}_n(k)$ and $\hat{u}_n(-k)$ do not affect the argument. Now observe that $|\lambda_k|^2 = (2\pi^2 k^2)^{-1} < 1 < 2e^\gamma$ for all k, and the result follows. ■

Remark 19 *The preceding results also holds if we replace the Brownian bridge by a Gaussian process with precision operator the negative Laplacian subject to Neumann boundary conditions and spatial mean zero; the eigenfunctions are then*

$$\psi_j(x) = \sqrt{2}\cos(j\pi x) = \frac{1}{\sqrt{2}}\big(\varphi_j(x) + \varphi_{-j}(x)\big), \quad j \geq 1.$$

The argument is identical, except no sgn(k) *term appears in λ_k.*

4. Numerical Illustrations

We now study two of the constructions of deep Gaussian processes numerically. In subsection 4.1 we look at realizations of the deep Gaussian process constructed using the covariance function formulation, and in subsection 4.2 we perform similar experiments for the covariance operator formulations. Finally we consider Bayesian inverse problems, in which we choose deep Gaussian processes as our prior distributions; we introduce a function space MCMC algorithm, which scales well under mesh refinement of the functions to be inferred, for sampling.

For the composition construction, numerical experiments are given by, for example, Damianou and Lawrence (2013); Duvenaud et al. (2014). We do not provide numerical experiments for the convolution construction; Theorem 16 tells us that interesting behaviour cannot be expected in this case.

4.1 Covariance Function

We start by investigating typical realizations of a deep Gaussian process, constructed through anisotropic covariance kernels as in section 2.2. As the basis of our construction, we choose a stationary Gaussian correlation kernel, given by

$$\rho_S(r) = \exp(-r^2), \qquad r > 0.$$

The function F determining the length scale of the kernel $\rho(\cdot, \cdot; u_n)$ is chosen as $F(x) = x^2$, such that $\Sigma(z) = (u_n(z))^2 \, \mathrm{I}_d$. Similar results are obtained with other choices of F in terms of the distribution of samples u_n. The choice of F does, however, influence the conditioning of the correlation matrix $\mathbf{R}(\mathbf{u}_n)$, and the choice $F(x) = \exp(x)$, for example, can lead to numerical instabilities. As described in section 2.2, we will sample from the finite dimensional distributions obtained by sampling from the Gaussian process at a finite number of points in the domain D. To generate the samples, we use the command `mvnrnd` in MATLAB, and when plotting the samples, we use linear interpolation.

In Figure 1, we show four independent realizations of the first seven layers $u_0, \ldots, u_6$, where u_0 is taken as a sample of the stationary Gaussian process with correlation kernel ρ_S. The domain D is here chosen as the interval $(0,1)$, and the sampling points are given by the uniform grid $x_i = \frac{i-1}{256}$, for $i = 1, \ldots, 257$. Each column in Figure 1 corresponds to one realization, and each row corresponds to a given layer u_n, the first row showing u_0. We can clearly see the non-stationary behaviour in the samples when progressing through the levels. We note that the ergodicity of the chain is also reflected in the samples, with the distribution of the samples u_n looking similar for larger values of n.

Figure 2 shows the same information as Figure 1, in the case where the domain D is $(0,1)^2$ and the sampling points are the tensor product of the one-dimensional points $x_i^1 = \frac{i-1}{64}$, for $i = 1, \ldots, 65$.

4.2 Covariance Operator

We now consider the covariance operator construction of the deep Gaussian process. In order to produce more interesting behaviour in the samples, we move away from the absolutely continuous setting considered in section 3.3 by introducing a rescaling of $C(u)$ that depends

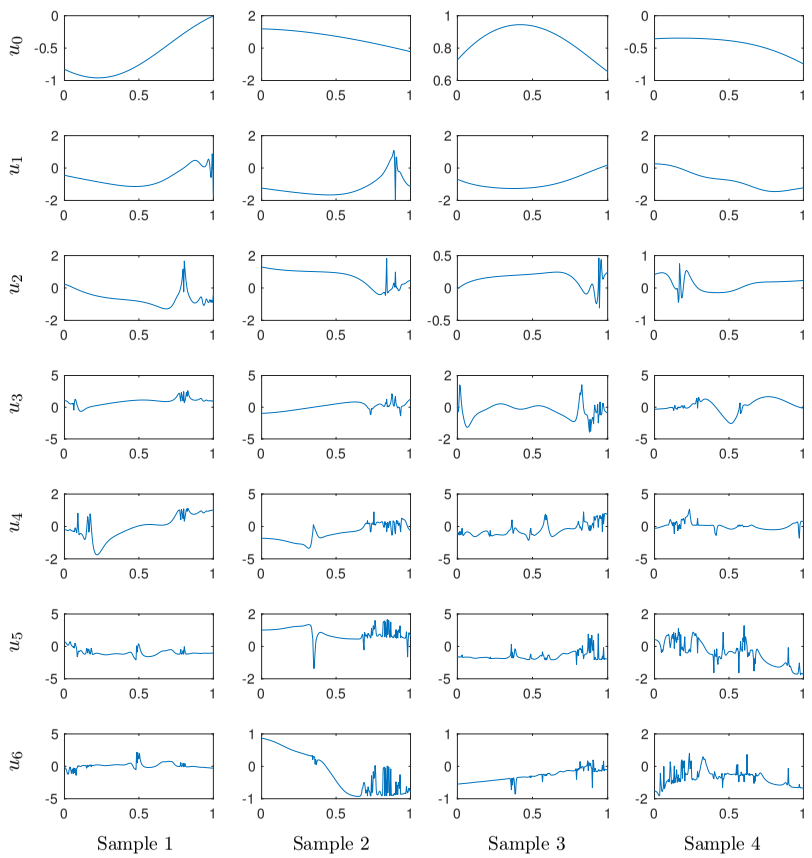


Figure 1: Four independent realizations of the first seven layers of a deep Gaussian process, in one spatial dimension, using the covariance kernel construction described in subsection 2.2. Each column corresponds to an independent chain, and layers $u_0, u_1, \ldots, u_6$ are shown from top-to-bottom.

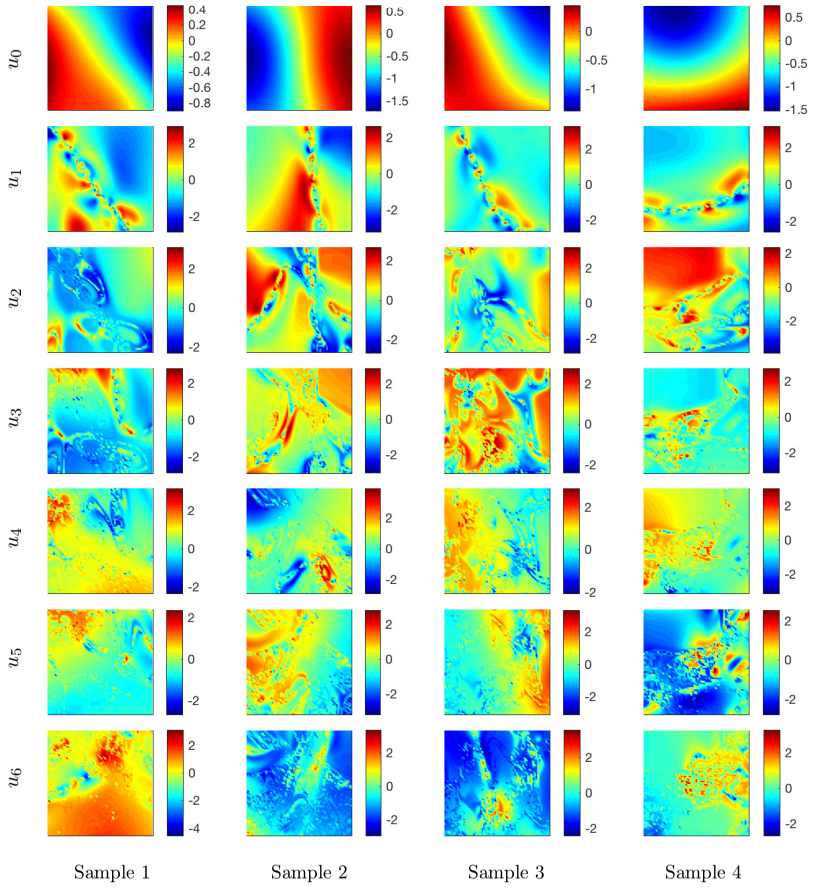


Figure 2: Four independent realizations of the first seven layers of a deep Gaussian process, in two spatial dimensions, using the covariance kernel construction described in subsection 2.2. Each column corresponds to an independent chain, and layers $u_0, u_1, \ldots, u_6$ are shown from top-to-bottom.

on u. This scaling is chosen so that the amplitude of samples is $\mathcal{O}(1)$ with respect to u. The rescaled family can be shown to satisfy Assumptions 3, and a minorization condition as in Lemma 15 can also be shown to hold when the state space is finite-dimensional. From this we can deduce that the resulting discretized process will still be ergodic.

Assume $D \subseteq \mathbb{R}^d$ and define the negative Laplacian $-\triangle$ on $D(-\triangle)$,

$$D(-\triangle) = \left\{ u \in H^2(D;\mathbb{R}) \;\middle|\; \frac{\mathrm{d}u}{\mathrm{d}\nu}(x) = 0 \text{ for } x \in \partial D \right\},$$

where ν is the outward normal to ∂D. Given $\alpha > d/2$, $\sigma > 0$, we define $P = -\triangle$ and

$$C(u)^{-1} = \sigma^{-2}(P + \Gamma(u))^{\alpha/2}\Gamma(u)^{d/2-\alpha}(P + \Gamma(u))^{\alpha/2} \tag{15}$$

where $\big(\Gamma(u)v\big)(x) = F\big(u(x)\big)v(x)$. The scaling introduced is inspired by the SPDE representation of Whittle-Matérn distributions (Lindgren et al., 2011); if $F(u) = \tau^2$ is chosen to be constant, then modulo boundary conditions, samples from a centred Gaussian distribution with covariance $C(u)$ are samples from a Whittle-Matérn distribution. In particular, τ corresponds to the inverse length-scale of samples, and samples almost-surely have s Sobolev and Hölder and derivatives for any $s < \alpha - d/2$.

For numerical experiments, we take

$$F(u) = \min\{F_- + ae^{bu^2}, F_+\}$$

for some $F_+, F_-, a, b > 0$. In particular, in one spatial dimension we take $F_+ = 150^2$, $F_- = 200$, $a = 100$ and $b = 2$. In two dimensions, we take $F_+ = 150^2$, $F_- = 50$, $a = 25$ and $b = 0.3$. We take $\alpha = 4$ in both cases, and choose σ such that $\mathbb{E}\big(u(x)^2\big) \approx 1$. These parameter choices were made empirically to ensure interesting structure of the samples. In order to generate samples at a given level, the negative Laplacian P is constructed using a finite-difference method. Given u, the operator $A(u)$ is then computed,

$$A(u) := \sigma^{-1}\Gamma(u)^{d/4-\alpha/2}(P + \Gamma(u))^{\alpha/2},$$

so that $v \sim N(0, C(u))$ solves the SPDE $A(u)v = \xi$, where ξ is white noise.

In Figure 3 we show samples of the deep Gaussian process on domain $D = (0,1)$, sampled on the uniform grid $x_i = \frac{i-1}{1000}$, for $i = 1, \ldots, 1001$. We show 4 independent realizations of the first seven layers of the process—each row corresponds to a given layer u_n. The anisotropy of the length-scale is evident in levels beyond u_0, and the effect of ergodicity is evident, with deeper levels having similar properties. Compared to the covariance function construction, local effects are less prominent, though a greater level of anisotropy could potentially be obtained by making an alternative choice of $F(\cdot)$. Figure 4 shows the same experiments on domain $D = (0,1)^2$, sampled on the tensor product of the one-dimensional points $x_i^1 = \frac{i-1}{150}$, for $i = 1, \ldots, 151$, and the same effects are observed. Figure 5 shows the trace of the norm of a DGP $\{u_n\}$ with $d = 1$, along with the running mean of these norms; the rapid convergence of the mean reflects the ergodicity of the chain.

We emphasize that our perspective on inference includes quite general inverse problems, and is not limited to the problems of regression and classification which dominate much of

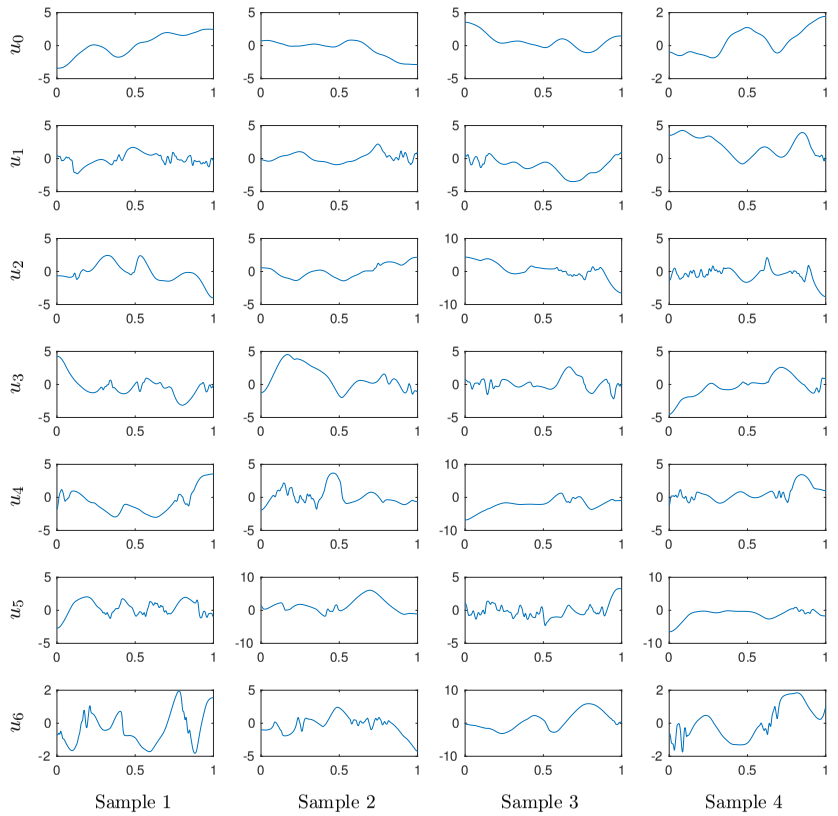


Figure 3: Four independent realizations of the first seven layers of a deep Gaussian process, in one spatial dimension, using the covariance operator construction described in subsection 4.2. Each column corresponds to an independent chain, and layers $u_0, u_1, \ldots, u_6$ are shown from top-to-bottom.

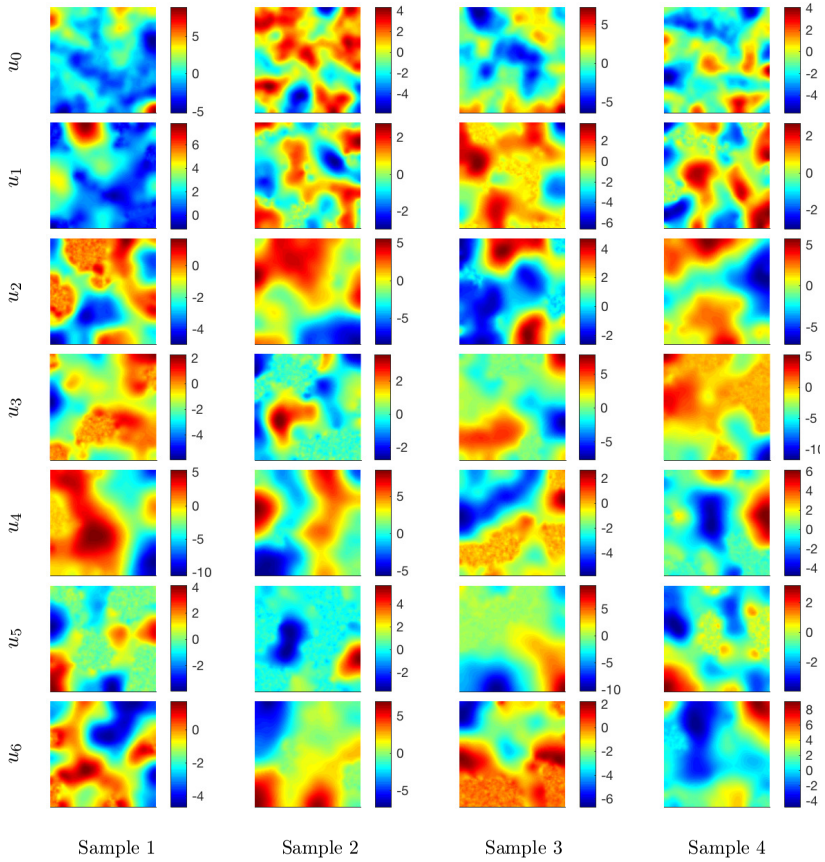


Figure 4: Four independent realizations of the first seven layers of a deep Gaussian process, in two spatial dimensions, using the covariance operator construction described in subsection 4.2. Each column corresponds to an independent chain, and layers $u_0, u_1, \ldots, u_6$ are shown from top-to-bottom.

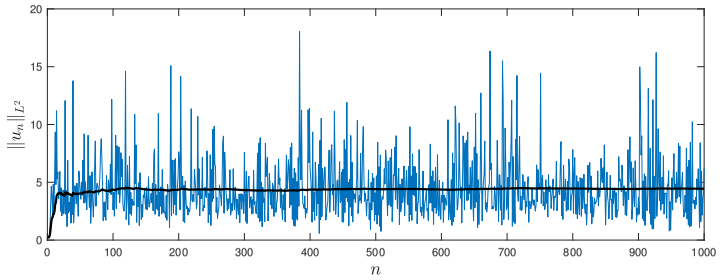


Figure 5: The trace of the norm of u_n versus n for a 1000 layer DGP $\{u_n\}$ as in Figure 3. The thick black curve shows the running mean of the norms.

classical machine learning; this broad perspective on the potential for the methodology affects the choice of algorithms that we study as we do not exploit any of the special structures that arise in regression and classification.

The deep Gaussian processes discussed in the previous sections were introduced with the idea of providing flexible prior distributions for inference, for example in inverse problems. The structure of such problems is as follows. We have data $y \in \mathbb{R}^J$ arising via the model

$$y = \mathcal{G}(u) + \eta \tag{16}$$

where η is a realization of some additive noise, and $\mathcal{G} : X \to \mathbb{R}^J$ is a (typically non-linear) forward map. The map $\mathcal{G}$ may involve, for example, solution of a partial differential equation which takes function u as input, or point evaluations of a function u, regression. In this paper we will fix $X = \mathcal{H}^N$, writing $u = (u_0, \ldots, u_{N-1}) \in X$; our prior beliefs on u will then be characterized by the first N states of a Markov chain of a form considered in the previous sections. Note that the map $\mathcal{G}$ could incorporate a projection map if the dependence is only upon a single state u_{N-1}; indeed this is the canonical example—the variables $(u_0, \ldots, u_{N-2})$ are viewed as hyperparameters in a prior on the parameter u_{N-1}.

4.2.1 Algorithms

We now turn to the design of algorithms for the Bayesian inference problems of sampling $u|y$. As already mentioned above, we are typically only interested in sampling the deepest layer $u_{N-1}|y$. However, due to the hierarchical definition of u_{N-1} given all the components of u, our algorithms work with the full set of layers u. Since the components of u are functions, and hence infinite dimensional objects in general, a guiding principle is to design algorithms which are well-defined on function space, an approach to MCMC inference reviewed by Cotter et al. (2013); the value of this approach is that it leads to algorithms whose mixing time is not dependent on the number of mesh points used to represent the function to be inferred. For simplicity of exposition we assume that the observational noise η is distributed

as $N(0,\Gamma)$; this is not central to our developments but makes the exposition concrete. Recalling that the Markov chain defining the prior beliefs is given by (ZeroMean), we can consider the unknowns in the problem to be the variables $u = (u_0, \ldots, u_{N-1})$, which are correlated under the prior, or the variables $\xi = (\xi_0, \ldots, \xi_{N-1})$, where we define $\xi_0 = u_0$, which are independent under the prior. These variables are related via $u = T(\xi)$, where the components of the deterministic map $T : X \to X$ are defined iteratively by

$$\begin{aligned} T_1(\xi_0, \ldots, \xi_{N-1}) &= \xi_0, \\ T_{n+1}(\xi_0, \ldots, \xi_{N-1}) &= L(T_n(\xi_0, \ldots, \xi_{N-1}))\xi_n, \quad n = 1, \ldots, N-1. \end{aligned}$$

The data may then be expressed in terms of ξ rather than u:

$$y = \tilde{\mathcal{G}}(\xi) + \eta = \mathcal{G}(T(\xi)) + \eta \tag{17}$$

where our prior belief on ξ is that its components are i.i.d. Gaussians. To be consistent with the notation introduced by Papaspiliopoulos et al. (2007); Yu and Meng (2011), (16) will be referred to as the *centred* model and (17) will be referred to as the *non-centred* model. The space $\mathcal{H}$ may be chosen differently in the centred and non-centred cases.

Associated with the two data models are two likelihoods: $\mathbb{P}(y|u)$ and $\mathbb{P}(y|\xi)$. Assuming that the observational noise $\eta \sim N(0,\Gamma)$ is Gaussian, where $\Gamma \in \mathbb{R}^{J\times J}$ is a positive definite covariance matrix, the likelihoods are given by

$$\begin{aligned} \mathbb{P}(y|u) &= \frac{1}{Z(y)} \exp\big(-\Phi(u;y)\big), \quad \Phi(u;y) := \frac{1}{2}\big|\Gamma^{-\frac{1}{2}}(y - \mathcal{G}(u))\big|^2, \\ \mathbb{P}(y|\xi) &= \frac{1}{\tilde{Z}(y)} \exp\big(-\tilde{\Phi}(\xi;y)\big), \quad \tilde{\Phi}(\xi;y) := \frac{1}{2}\big|\Gamma^{-\frac{1}{2}}(y - \tilde{\mathcal{G}}(\xi))\big|^2. \end{aligned}$$

We may then apply Bayes' theorem to write down the posterior distributions $\mathbb{P}(u|y)$ and $\mathbb{P}(\xi|y)$:

$$\begin{aligned} \mathbb{P}(u|y) &\propto \mathbb{P}(y|u)\mathbb{P}(u) \propto \exp\big(-\Phi(u;y)\big)\mathbb{P}(u), \\ \mathbb{P}(\xi|y) &\propto \mathbb{P}(y|\xi)\mathbb{P}(\xi) \propto \exp\big(-\tilde{\Phi}(\xi;y)\big)\mathbb{P}(\xi). \end{aligned}$$

We know (Cotter et al., 2013) that it is straightforward to design algorithms to sample $\mathbb{P}(\xi|y)$ which are well-defined in infinite dimensions, exploiting the fact that $\mathbb{P}(\xi)$ is Gaussian. An example of such an algorithm given by Algorithm 1.

This algorithm produces a chain $\{\xi^{(k)}\}_{k\in\mathbb{N}}$ that samples $\mathbb{P}(\xi|y)$ in stationarity; and $\{T(\xi^{(k)})\}_{k\in\mathbb{N}}$ will be samples of $\mathbb{P}(u|y)$. By working in non-centred coordinates we have been able to design this algorithm which is well-defined on function space. If we were to work with the centred coordinates u directly, the algorithm would not be well-defined on function space: in infinite dimensions, each family of measures $\{\mathbb{P}(u_n|u_{n-1})\}_{u_{n-1}\in\mathcal{H}}$ will typically be mutually singular, and so a proposed update $u \mapsto \hat{u}$ will almost surely be rejected. To see why this rejection occurs in practice, in high finite dimensions K, notice that the acceptance probability for an update $u \mapsto \hat{u}$ will involve the ratios of the Gaussian densities $N(u_n; 0, C(u_{n-1}))$ and $N(u_n; 0, C(\hat{u}_{n-1}))$. These densities will decay to zero as the dimension K is increased, and their ratio will only be well-defined in the limit if the measures are equivalent; consequently, the Markov chain will mix very poorly. Working with the

Algorithm 1 Non-Centred Algorithm

1. Fix $\beta_0, \ldots, \beta_{N-1} \in (0,1]$ and define $B = \mathrm{diag}(\beta_j)$. Choose initial state $\xi^{(0)} \in X$, and set $u^{(0)} = T(\xi^{(0)}) \in X$. Set $k = 0$.
2. Propose $\hat{\xi}^{(k)} = (I - B^2)^{\frac{1}{2}}\xi^{(k)} + B\zeta_j^{(k)}, \quad \zeta^{(k)} \sim N(0, I)$.
3. Set $\xi^{(k+1)} = \hat{\xi}^{(k)}$ with probability
$$\alpha_k = \min\left\{1, \exp\Big(\Phi\big(T(\xi^{(k)});y\big) - \Phi\big(T(\hat{\xi}^{(k)});y\big)\Big)\right\};$$
otherwise set $\xi^{(k+1)} = \xi^{(k)}$.
4. Set $k \mapsto k+1$ and go to 1.

non-centred coordinates ξ, the prior does not appear in the acceptance probability and so this issue is circumvented. Another advantage of using the non-centred coordinates is that there is no need to calculate the (divergent) log determinants which appear in the centred acceptance probability, avoiding potential numerical issues. These issues are discussed in greater depth and generality by Chen et al. (2018). For the reasons set-out in that paper, including those above, we have used only the non-centred algorithm in what follows. When the forward model $\mathcal{G}(u) = Au$ is linear, the non-centred algorithm can be combined with standard Gaussian process regression techniques via the identity

$$\mathbb{P}(\mathrm{d}u_N|y) = \int_X \mathbb{P}(\mathrm{d}u_N|u_{N-1}, y)\mathbb{P}(\mathrm{d}u_{N-1}|y).$$

The distribution $\mathbb{P}(\mathrm{d}u_N|u_{N-1}, y) = N(m_y(u_{N-1}), C_y(u_{N-1}))$ is Gaussian, where expressions for m_y, C_y are known, and so direct sampling methods are available. On the other hand, we have that $\mathbb{P}(y|u_{N-1}) = N(0, AC(u_{N-1})A^* + \Gamma)$, and so we may use the non-centred algorithm to robustly sample the measure

$$\begin{aligned} \mathbb{P}(\mathrm{d}u_{N-1}|y) &= \exp(-\Psi(u_{N-1};y))\mathbb{P}(\mathrm{d}u_{N-1}), \\ \Psi(u_{N-1};y) &= \frac{1}{2}\|y\|^2_{AC(u_{N-1})A^*+\Gamma} + \frac{1}{2}\log\det(AC(u_{N-1})A^* + \Gamma), \end{aligned}$$

after reparametrizing in terms of ξ. This approach can be viable even when the data is particularly informative so that Φ is very singular—this singularity does not in general pass to Ψ. It is this approach that we use for the simulations in the following subsections. An alternative approach not based on MCMC would be to use the non-centred parameterization of the Ensemble Kalman Filter (Chada et al., 2018) which we have successfully implemented in the context of the deep Gaussian processes of this paper, but do not show here for reasons of brevity.

4.3 Application to Regression

We consider the application of the non-centered algorithm described above to simple regression problems in one and two spatial dimensions.

4.3.1 One-Dimensional Simulations

We consider first the case $D = (0, 1)$, where the forward map is given by a number of point evaluations: $\mathcal{G}_j(u) = u(x_j)$ for some sequence $\{x_j\}_{j=1}^J \subseteq D$. We compare the quality of reconstruction versus both the number of point evaluations and the number of levels in the deep Gaussian prior. We use the same parameters for the family of covariance operators as in subsection 4.2. The base layer u_0 is taken to be Gaussian with covariance of the form (15), with $\Gamma(u) \equiv 20^2$.

The true unknown field $u^\dagger$ is given by the indicator function $u^\dagger = \mathbb{1}_{(0.3,0.7)}$, shown in Figure 6. It is generated on a mesh of 400 points, and three data sets are created wherein it is observed on uniform grids of $J = 25$, 50 and 100 points, and corrupted by white noise with standard deviation $\gamma = 0.02$. Sampling is performed on a mesh of 200 points to avoid an inverse crime (Kaipio and Somersalo, 2006). 10^6 samples are generated per chain, with the first 2×10^5 discarded as burn-in when calculating means. The jump parameters β_j are adaptively tuned to keep acceptance rates close to 30%.

In these experiments the deepest field is labelled as u_N, rather than as u_{N-1} as in the statement of the algorithm; this is purely for notational convenience, of course. In Figure 7 the means of the deepest field u_N and of the length-scales associated with each hidden layer are shown, that is, approximations to $\mathbb{E}(u_N)$ and $\mathbb{E}\big(F(u_j)^{\frac{1}{2}}\big)$ for each $j = 0, \ldots, N-1$. We see that, in all cases, the reconstructions of $u^\dagger$ are visually similar when two or more layers are used, and similar length-scale fields $\mathbb{E}\big(F(u_{N-1})^{\frac{1}{2}}\big)$ are obtained in these cases. The sharpness of these length-scale fields is related to the amount of data. Additionally, when $N = 4$ and $J = 100$ the location of the discontinuities is visible in the estimate for $\mathbb{E}\big(F(u_{N-2})^{\frac{1}{2}}\big)$, suggesting the higher quality data can influence the process more deeply. When $J = 50$ or $J = 25$, this layer does not appear to be significantly informed. When a single layer prior is used, the reconstruction fails to accurately capture the discontinuities. Figure 7 also shows bands of quantiles of the values $u(x)$ under the posterior, illustrating their distribution; in particular the lack of symmetry and disagreement of the means and medians show that the posterior is clearly non-Gaussian. Uncertainty increases both as the number of observations J and the layer n in the chain is increased. Note in particular the over-confidence of the shallow Gaussian process posterior: the truth is not contained within 95% credible intervals in all cases.

In Table 1 we show the L^1-errors between the true field and the posterior means arising from the different setups. The errors decrease as the number of observation points is increased, as would be expected. Additionally, when $J = 100$ and $J = 50$, the accuracy of the reconstruction increases with the number of layers, though the most significant increase occurs when increasing from 1 to 2 layers. When $J = 25$, the error increases beyond 2 layers, suggesting that some balance is required between the quality of the data and the flexibility of the prior.

In Figure 8 we replace the uniformly spaced observations with 10^6 randomly placed observations, to illustrate the effect of very high quality data. With 3 or 4 layers, more anisotropic behavior is observed in the length-scale field. Additionally, the layer u_{N-2} is much more strongly informed than the cases with fewer observations, though the layer u_{N-3} in the case $N = 4$ does not appear to be informed at all, indicating a limitation on how deeply the process can be influenced by data. The corresponding errors are shown in Table 1—as

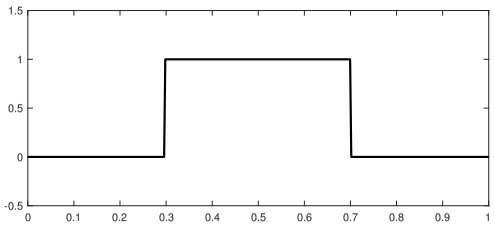


Figure 6: The true field used to generate the data for the one-dimensional inverse problem.

J	1 layer	2 layers	3 layers	4 layers
100	0.0485	0.0200	0.0198	**0.0196**
50	0.0568	0.0339	0.0339	**0.0337**
25	0.0746	**0.0658**	0.0667	0.0670
10^6	0.0131	0.000145	**0.000133**	**0.000133**

Table 1: The L^1-errors $\|u^\dagger - \mathbb{E}(u_N)\|_{L^1}$ between the true field and sample means for the one-dimensional simulations shown in Figure 7, for different numbers of data points J and layers N. Also shown are the corresponding errors for the simulations shown in Figure 8

.

in the cases $N = 50, 100$, more layers increases the accuracy of the mean, with diminishing returns for each additional layer. Note that higher accuracy could be attained in the single layer case by adjusting the constant length-scale parameter.

Finally, in Figure 9, we consider the same experiment as in Figure 7, except observations are limited to the subset $(0, 0.5)$ of the domain. Uncertainty is naturally higher in the unobserved portion of the domain. Uncertainty also increases in the observed layer u_N as N is increased; this could suggest that deep Gaussian processes may provide better generalization to unseen data than shallow Gaussian processes—note that the truth has much higher probability under the posterior with 4 layers versus just 1.

4.3.2 Two-Dimensional Simulations

We now consider the case $D = (0, 1)^2$, again where the forward map is given by a number of point evaluations. We fix the number of point observations $J = 2^{10}$, on a $2^5 \times 2^5$ uniform grid. We again compare quality of reconstruction versus the number of point evaluations and the number of levels in the deep Gaussian prior, and use the same parameters for the

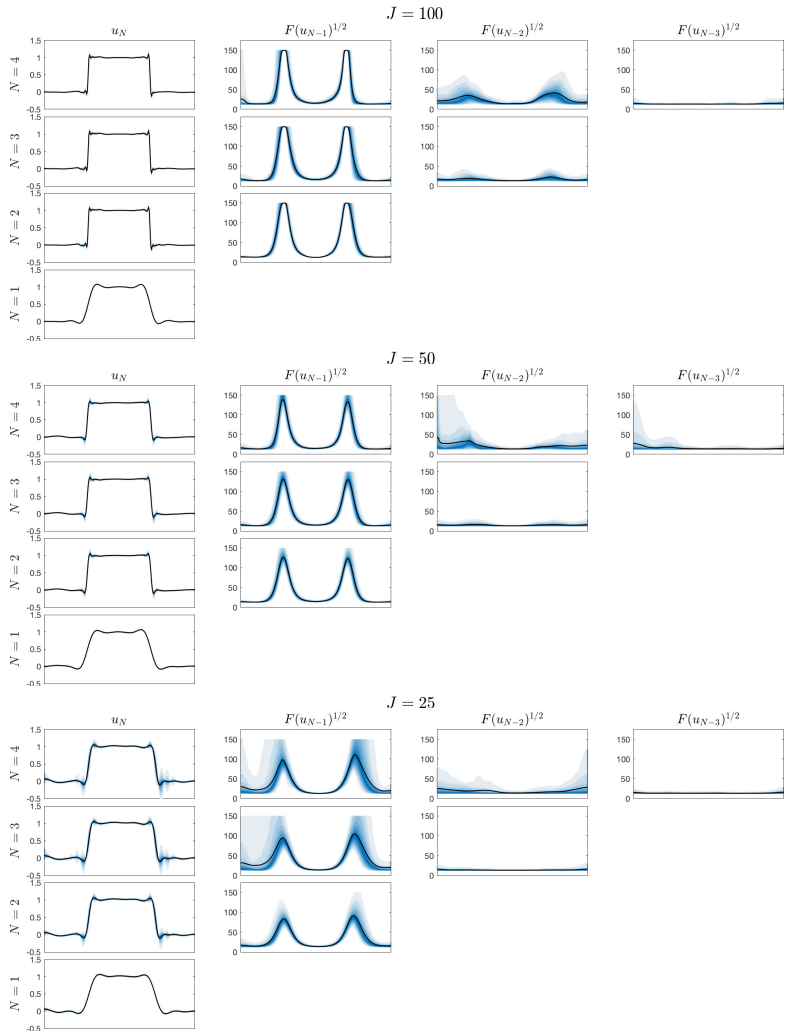


Figure 7: Estimates of posterior means (solid curves) and $5-95\%$ quantiles (shaded regions) arising from one-dimensional inverse problem. Number of data points taken are $J = 100$ (top block), $J = 50$ (middle block), $J = 25$ (bottom block). From left-to right, results for u_N, $F(u_{N-1})^{\frac{1}{2}}$, $F(u_{N-2})^{\frac{1}{2}}$, $F(u_{N-3})^{\frac{1}{2}}$ are shown. From top-to-bottom within each block, $N = 4, 3, 2, 1$.

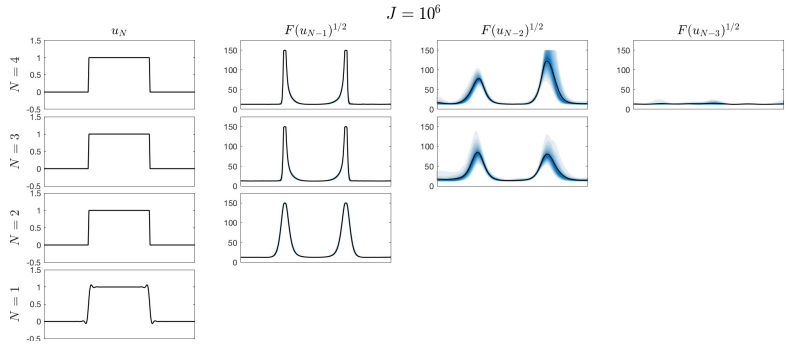


Figure 8: Estimates of posterior means (solid curves) and $5-95\%$ quantiles (shaded regions) arising from one-dimensional inverse problem, with $J = 10^6$ data points. From left-to right, results for u_N, $F(u_{N-1})^{\frac{1}{2}}$, $F(u_{N-2})^{\frac{1}{2}}$, $F(u_{N-3})^{\frac{1}{2}}$ are shown. From top-to-bottom, $N = 4, 3, 2, 1$.

family of covariance operators as in subsection 4.2. The base layer u_0 is taken to be Gaussian with covariance of the form (15), with $\Gamma(u) \equiv 20^2$.

The true unknown field $u^\dagger$ is constructed as a linear combination of truncated trigonometric functions with different length-scales, and shown in Figure 10 along with its contours. It is given by

$$\begin{aligned} u^\dagger(x,y) = &\cos(2\pi x)\cos(2\pi y) + \sin(4\pi x)\sin(4\pi y)\mathbb{1}_{(1/4,3/4)^2}(x,y) \\ &+ \sin(8\pi x)\sin(8\pi y)\mathbb{1}_{(1/2,3/4)^2}(x,y) \\ &+ \sin(16\pi x)\sin(16\pi y)\mathbb{1}_{(1/4,1/2)^2}(x,y). \end{aligned}$$

It is generated on a uniform square mesh of 2^{14} points, and two data sets are created wherein it is observed on uniform square grid of $J = 2^{10}, 2^8$ points, and corrupted by white noise with standard deviation $\gamma = 0.02$. Sampling is performed on a mesh of 2^{12} points to again avoid an inverse crime. 4×10^5 samples are generated per chain, with the first 2×10^5 discarded as burn-in when calculating means. Again the jump parameters β_j are adaptively tuned to keep acceptance rates close to 30%.

In Figure 11, analogously to Figure 7, the means of u_N and of the length-scales associated with each layer are shown, for $N = 1, 2, 3$. When $J = 2^{10}$, reconstructions are similar, though quality is generally proportional to the number of layers. In particular the, effect of too short a length-scale is evident in the case $N = 1$, in the regions where the length-scale should be larger, and conversely the effect of too long a length-scale is evident in the cases $N = 1, 2$ in the region where the length-scale should be the shortest. In the cases $N = 2, 3$, the length-scale fields $\mathbb{E}\big(F(u_{N-1})^{\frac{1}{2}}\big)$ are similar, though in the case $N = 3$ more accurately captures the true length-scales. When $J = 2^8$ the reconstructions are again similar, though

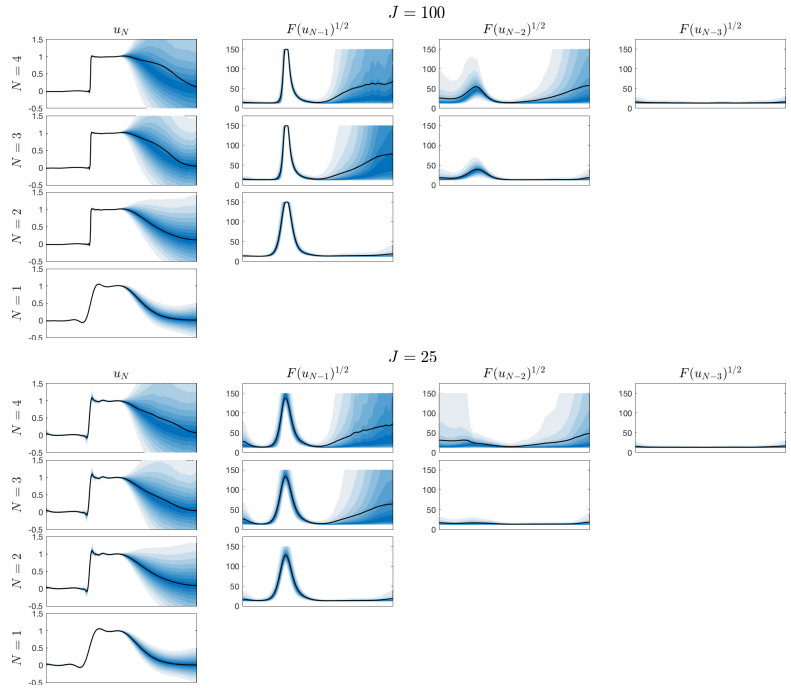


Figure 9: Estimates of posterior means (solid curves) and $5-95\%$ quantiles (shaded regions) arising from one-dimensional inverse problem. Number of data points taken are $J = 100$ (top block), $J = 25$ (bottom block). From left-to right, results for u_N, $F(u_{N-1})^{\frac{1}{2}}$, $F(u_{N-2})^{\frac{1}{2}}$, $F(u_{N-3})^{\frac{1}{2}}$ are shown. From top-to-bottom within each block, $N = 4, 3, 2, 1$.

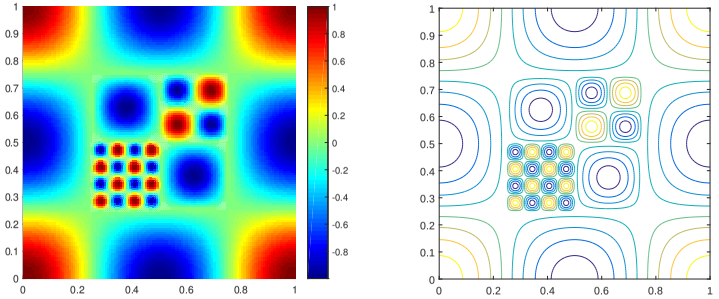


Figure 10: The true field used to generate the data for the two-dimensional inverse problem.

J	1 layer	2 layers	3 layers
2^{10}	0.0856	0.0813	**0.0681**
2^8	0.1310	**0.1260**	0.1279

Table 2: The L^2-errors $\|u^\dagger - \mathbb{E}(u_N)\|_{L^2}$ between the true field and sample means for the two-dimensional simulations shown in Figure 11, for different numbers of data points J and layers N.

there is now less accuracy in the shapes of the contours. In particular, the effect of too short a length-scale is especially evident in the case $N = 1$. The values of the reconstructed fields in the area of shortest length-scale are inaccurate in all cases—the positions of the observation points meant that the actual values of the peaks were not reflected in the data. The fields $\mathbb{E}\big(F(u_{N-1})^{\frac{1}{2}}\big)$ have similar structure to the case $J = 2^{10}$, though less accurately represent the true length scales. The L^2-errors between the means and the truth are shown in Table 2

5. Conclusions, Discussion, and Actionable Advice

In this section we provide an overview of the advantages and disadvantages of each of the four different DGP constructions considered in the paper, and summarize actionable advice that can be taken from the theoretical and numerical results that have been presented. We then outline a number of directions that would be interesting for future study.

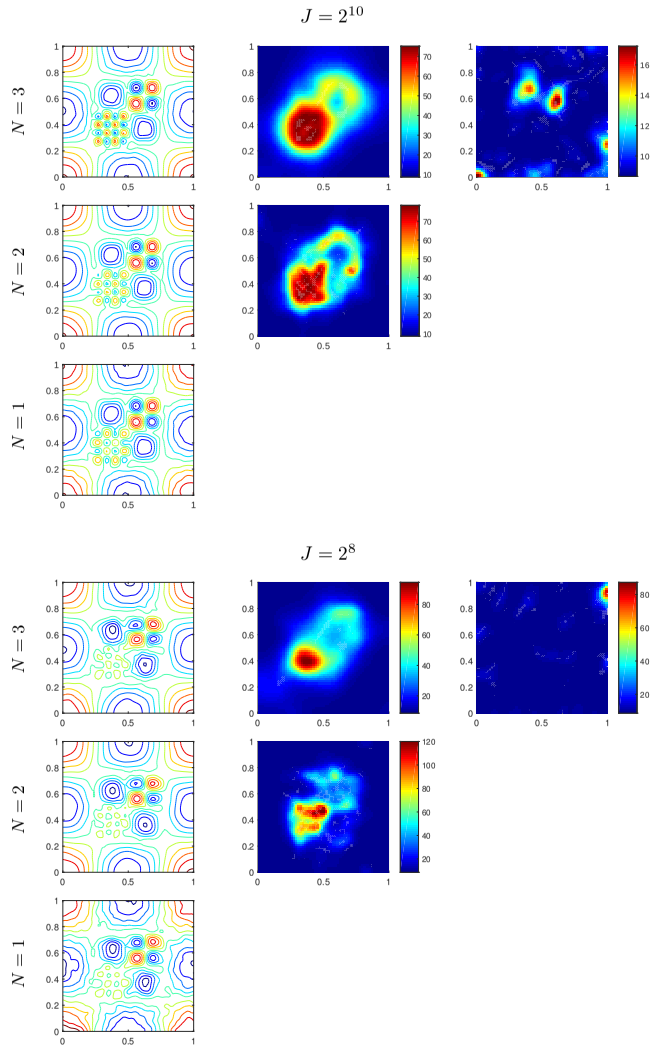


Figure 11: Estimates of posterior means arising from two-dimensional inverse problem. (Top block) $J = 2^{10}$, (Bottom block) $J = 2^8$. From left-to right, $\mathbb{E}\big(u_N\big)$, $\mathbb{E}\big(F(u_{N-1})^{\frac{1}{2}}\big)$, $\mathbb{E}\big(F(u_{N-2})^{\frac{1}{2}}\big)$. From top-to-bottom within each block, $N = 3, 2, 1$.

5.1 Comparison of Deep GP Constructions

We have considered four different constructions of deep GPs and we now discuss their relative merits. We also consider the context of variational inference which is popular in machine learning primarily because of its tractability. We emphasize however that it forms an uncontrolled approximation of the true posterior distribution and may fail to adequately represent the posterior distribution, and uncertainty in particular.

The **composition** construction is the classical construction introduced by Damianou and Lawrence (2013), building a hierarchy of layers using a stationary covariance function and composition. It has received the most study, and methods for variational inference have already been established. It has the advantage of scaling well with respect to data dimension d, however accurate sampling methods such as MCMC are intractable for large numbers of data points, due to the requirement to construct and factor dense covariance matrices at every step.

The **covariance function** construction builds the hierarchy using a stationary covariance function, and iteratively modifying its associated length scale. It has the advantage that each layer can be readily interpreted as the anisotropic length-scale field of the following layer. Its scaling properties are similar to those of the composition construction, however variational inference methods for this construction have not yet been studied.

The **covariance operator** construction builds the hierarchy using an SPDE representation of stationary Matèrn fields, and again iteratively modifies their associated length scale. It allows for fast sampling in low data dimension d via the use of PDE solvers, even when the number of data points is large. Accurate sampling via MCMC methods is tractable with this construction, due to the low cost of constructing and storing the inverse covariance (precision) matrix. Inference when d is large appears to be intractable at present, due to the requirement of dense meshes for PDE solvers.

Finally, the **convolution** construction builds the hierarchy via iterative convolution of Gaussian random fields. It has the advantage of being amenable to analysis, however the results of this analysis indicate that it would likely be a poor construction to use for inference due to trivial behaviour for large depth.

To summarize the numerical results on illustrative regression problems from the previous section, if the data is high quality, a small number of layers in the DGP will be sufficient as the problem becomes closer to interpolation. Conversely, if the data is low quality the likelihood is not strong enough to inform deeper layers in the DGP, and so a small number of layers is again sufficient. As a consequence, when the data lies between these two cases, and the truth has sufficiently rich structure, the use of deeper processes may be advantageous, but care is required to limit the number of layers employed.

5.2 Summary and Future Work

There are a number of interesting ways in which this work may be generalized. Within the context of covariance operators it is of interest to construct covariances $C(u)$ which are defined as $L^{-\alpha}$ with L being the divergence form elliptic operator

$$Lu = -\nabla \cdot \big(F(u)\nabla u\big).$$

Such a construction allows for the conditional distributions of the layers to be viewed as stationary on deformed spaces (Lindgren et al., 2011, §3.4), or to incorporate anisotropy in specific directions (Roininen et al., 2014, §3.1). Similar notions of anisotropy in different directions can be incorporated into the covariance function formulation by choosing the length scale $\Sigma(z)$ different to a multiple of the identity matrix. Additionally, we could consider a non-zero mean in the iteration (GP), as considered by Duvenaud et al. (2014); Salimbeni and Deisenroth (2017), allowing for forcing of the system. For example, with the choice $m(u_n) = u_n$ and a rescaling of the covariance, we obtain the ResNet-type iteration

$$u_{n+1} = u_n + \sqrt{\triangle t} L(u_n)\xi_{n+1}.$$

This may be viewed as a discretization of the continuous-time stochastic differential equation

$$\mathrm{d}u = L(u)L(u)^\top \mathrm{d}W,$$

analogously to what has been considered for neural networks (Haber and Ruthotto, 2017). Study of these systems could be insightful, for example deriving conditions to ensure a lack of ergodicity and hence arbitrary depth. As before $\top$ denotes the adjoint operation.

And finally it is possible to consider processes outside the four categories considered here; for example the one-step transition from u_n to u_{n+1} might be defined via stochastic integration against i.i.d. Brownian motions.

We have shown how a number of ideas in the literature may be recursed to produce deep Gaussian processes, different from those introduced by Damianou and Lawrence (2013). We have studied the effective depth of these processes, either through demonstrating ergodicity, or through showing convergence to a trivial solution (such as 0 or ∞). Together these results demonstrate that, as also shown by Duvenaud et al. (2014) for the original construction of deep Gaussian processes, care is needed in order to design processes with significant depth. Nonetheless, even a few layers can be useful for inference purposes, and we have demonstrated this also. It is an interesting question to ask precisely how the approximation power and effective depth are affected by the number of layers of the process, both in the non-ergodic case, and in the ergodic case before stationarity has been reached.

We also emphasize that the analysis in the paper is based solely on the deep Gaussian process u_n, and not the conditioned process $u_n|y$ in the inference problem with observed data y. The ergodicity properties of u_n do not directly carry over to $u_n|y$. As we have seen in the numerical experiments, the number of layers required in the inference problem in practice depends on the information content in the observed data y, and the analysis in this paper does not fully answer the question as to how many. The results in this paper do show, however, that in the case of ergodic constructions, the expressive power of the *prior* distribution in the inference problem does not increase past a certain number of layers. This provides some justification for using only a moderate number of layers in a deep Gaussian process prior in inference problems.

There are interesting approximation theory questions around deep processes, such as those identified in the context of neural networks by Pinkus (1999). There are also interesting questions around the use of these deep processes for inversion; in particular it seems hard to get significant value from using depth of more than two or three layers for noisy inverse problems. On the algorithmic side the issue of efficiently sampling these deep processes

(even over only two layers), when conditioned on possibly nonlinear observations remains open. We have used non-centred parameterizations because these may be sampled using function-space MCMC (Cotter et al., 2013; Chen et al., 2018); but centred methods, or mixtures, may be desirable for some applications.

Acknowledgments

MG is supported by EPSRC grants [EP/R034710/1, EP/R018413/1, EP/R004889/1, EP/P020720/1], an EPSRC Established Career Fellowship EP/J016934/3, a Royal Academy of Engineering Research Chair, and The Lloyds Register Foundation Programme on Data Centric Engineering. AMS is supported by AFOSR Grant FA9550-17-1-0185 and by US National Science Foundation (NSF) grant DMS 1818977. ALT is partially supported by The Alan Turing Institute under the EPSRC grant EP/N510129/1.

Appendix A. Proofs for Section 2

Proof of Proposition 1 The stationary kernel ρ_{S} is positive definite by Assumption 1, and so by (Wendland, 2004, Theorem 7.14), we have

$$\rho_{\mathrm{S}}(r) = \int_0^\infty \exp(-r^2 t)\,\mathrm{d}\nu(t) \quad \text{for all } r \in [0,\infty),$$

for a finite, non-negative Borel measure ν on $[0,\infty)$ that is not concentrated at 0 (i.e. it is not a multiple of the Dirac measure centred at 0).

For any $x \in \mathbb{R}^d$ and $t \in [0,\infty)$, let us now define the matrix $\tilde{\Sigma}_t(x) := (4t)^{-1}\Sigma(x)$ and the functions

$$K_{x,t}(z) = \frac{1}{(2\pi)^{d/2}|\tilde{\Sigma}_t(x)|^{1/2}} \exp\Big(-\frac{1}{2}(x-z)^T \tilde{\Sigma}_t(x)^{-1}(x-z)\Big).$$

Here $|\cdot|$ denotes determinant and so the preceding is simply an expression for a normal density with mean x and covariance matrix $\tilde{\Sigma}_t(x)$ when $t > 0$; at $t = 0$, we simply have $K_{x,t}(z) = 0$, for all $x, z \in \mathbb{R}^d$. Then $\rho(x,x')$ is given by

$$\begin{aligned}
&\frac{2^{\frac{d}{2}}|\Sigma(x)|^{\frac{1}{4}}|\Sigma(x')|^{\frac{1}{4}}}{|\Sigma(x)+\Sigma(x')|^{\frac{1}{2}}}\rho_{\mathrm{S}}\Big(\sqrt{Q(x,x')}\Big) = \frac{2^{\frac{d}{2}}|\Sigma(x)|^{\frac{1}{4}}|\Sigma(x')|^{\frac{1}{4}}}{|\Sigma(x)+\Sigma(x')|^{\frac{1}{2}}}\int_0^\infty \exp\big(-tQ(x,x')\big)\,\mathrm{d}\nu(t)\\
&= \frac{2^{\frac{d}{2}}|\Sigma(x)|^{\frac{1}{4}}|\Sigma(x')|^{\frac{1}{4}}}{|\Sigma(x)+\Sigma(x')|^{\frac{1}{2}}}\int_0^\infty \exp\left(-t(x-x')^T\left(\frac{\Sigma(x)+\Sigma(x')}{2}\right)^{-1}(x-x')\right)\mathrm{d}\nu(t)\\
&= 2^{\frac{d}{2}}\int_0^\infty \frac{|\tilde{\Sigma}_t(x)|^{\frac{1}{4}}|\tilde{\Sigma}_t(x')|^{\frac{1}{4}}}{|\tilde{\Sigma}_t(x)+\tilde{\Sigma}_t(x')|^{\frac{1}{2}}}\exp\left(-\frac{1}{2}(x-x')^T\left(\tilde{\Sigma}_t(x)+\tilde{\Sigma}_t(x')\right)^{-1}(x-x')\right)\mathrm{d}\nu(t)\\
&= (2\pi)^{\frac{d}{2}}2^{\frac{d}{2}}\int_0^\infty |\tilde{\Sigma}_t(x)|^{\frac{1}{4}}|\tilde{\Sigma}_t(x')|^{\frac{1}{4}}\int_{\mathbb{R}^d} K_{x,t}(z)K_{x',t}(z)\,\mathrm{d}z\,\mathrm{d}\nu(t),
\end{aligned}$$

where in the last step, we have used the fact that the convolution $\int_{\mathbb{R}^d} K_{x,t}(z)K_{x',t}(z)\,\mathrm{d}z$ can be calculated explicitly using properties of normal random variables. More precisely, we

have

$$\int_{\mathbb{R}^d} K_{x,t}(z)K_{x',t}(z)\,\mathrm{d}z = \int_{\mathbb{R}^d} p_X(z-x)p_{X'}(z)\,\mathrm{d}z = \int_{\mathbb{R}^d} p_{X,X'}(z-x,z)\,\mathrm{d}z,$$

where p_X is the density of $X \sim N(0, \tilde{\Sigma}_t(x))$, $p_{X'}$ is the density of $X' \sim N(x', \tilde{\Sigma}_t(x'))$ and X and X' are independent. The change of variable from X, X' to W, X', where $W = X' - X$, has Jacobian 1, and so

$$\int_{\mathbb{R}^d} p_{X,X'}(z-x,z)\,\mathrm{d}z = \int_{\mathbb{R}^d} p_{W,X'}(z-(z-x),z)\,\mathrm{d}z = \int_{\mathbb{R}^d} p_{W,X'}(x,z)\,\mathrm{d}z = p_W(x).$$

Since $W = X' - X \sim N(x', \tilde{\Sigma}_t(x) + \tilde{\Sigma}_t(x'))$, we hence have

$$\begin{aligned}&\int_{\mathbb{R}^d} K_{x,t}(z)K_{x',t}(z)\,\mathrm{d}z \\ &\quad= \frac{1}{(2\pi)^{\frac{d}{2}}|\tilde{\Sigma}_t(x) + \tilde{\Sigma}_t(x')|^{\frac{1}{2}}} \exp\Big(-\frac{1}{2}(x-x')^T\Big(\tilde{\Sigma}_t(x) + \tilde{\Sigma}_t(x')\Big)^{-1}(x-x')\Big),\end{aligned}$$

as required.

Now, for any $b \in \mathbb{R}^N$ and pairwise distinct $\{x_i\}_{i=1}^N$, we then have

$$\begin{aligned}&\sum_{i=1}^N\sum_{j=1}^N b_i b_j \rho(x_i,x_j) \\ &= (2\pi)^{\frac{d}{2}}2^{\frac{d}{2}}\sum_{i=1}^N\sum_{j=1}^N b_i b_j \int_0^\infty \int_{\mathbb{R}^d} |\tilde{\Sigma}_t(x_i)|^{\frac{1}{4}} K_{x_i,t}(z)|\tilde{\Sigma}_t(x_j)|^{\frac{1}{4}} K_{x_j,t}(z)\,\mathrm{d}z\,\mathrm{d}\nu(t) \\ &= (2\pi)^{\frac{d}{2}}2^{\frac{d}{2}}\int_0^\infty \int_{\mathbb{R}^d} \left(\sum_{i=1}^N b_i |\tilde{\Sigma}_t(x_i)|^{\frac{1}{4}} K_{x_i,t}(z)\right)^2 \mathrm{d}z\,\mathrm{d}\nu(t) \\ &\geq 0,\end{aligned}$$

since the Borel measure ν is finite and non-negative. It remains to show that strict inequality also holds.

Firstly, we note that $|\tilde{\Sigma}_0(x_i)|^{\frac{1}{4}} K_{x_i,0}(z) = 0$, for all $x_i, z \in \mathbb{R}^d$, which means that the integrand with respect to t is identically equal to zero at $t = 0$. Secondly, we note that the points $\{x_i\}_{i=1}^N$ are pairwise distinct and the functions $\{|\tilde{\Sigma}_t(x_i)|^{\frac{1}{4}} K_{x_i,t}(\cdot)\}_{i=1}^N$ are hence linearly independent for any $t \in (0,\infty)$. It is thus impossible to make the integrand with respect to z identically equal to 0 for a.e. $z \in \mathbb{R}^d$. As a consequence the integrand with respect to t is positive for all $t \in (0,\infty)$. Since we know that the measure ν is not concentrated at 0 this completes the proof that ρ is positive definite on $\mathbb{R}^d \times \mathbb{R}^d$, for any $d \in \mathbb{N}$.

Finally, we note that the kernel ρ is clearly non-stationary, and is a correlation function since $\rho(x,x) = 1$, for any $x \in \mathbb{R}^d$. ■

Proof of Proposition 2 We note that the definition of positive definite in Assumptions 1(i) refers only to behaviour of the kernel on a finite set of pairwise distinct points $\{x_i\}_{i=1}^N$. By Assumption 2(i), the function G is non-negative and bounded. If $G(z) > 0$ for all $z \in \mathbb{R}^d$,

then the matrix $\Sigma(z)$ is positive definite for all $z \in \mathbb{R}^d$, and the fact that $\rho(\cdot,\cdot)$ is positive definite follows directly from Proposition 1.

It remains to investigate the case where $G(z) = 0$ for some $z \in \mathbb{R}^d$. We will prove that $\rho(\cdot,\cdot)$ is positive definite by showing that the correlation matrix $\mathbf{R}$, with entries $\mathbf{R}_{ij} = \rho(x_i,x_j)$, is positive definite for any pairwise disjoint points $\{x_i\}_{i=1}^N$. Without loss of generality, we will study the case $G(x_1) = 0$; the proof easily adapts to the case where $G(x_i) = 0$, for $i \neq 1$. To define $\rho(x_1,x_j)$ in this case, we start by assuming $G(x_1) > 0, G(x_j) > 0$, and then take limits.

With $\Sigma(z) = G(z)\mathrm{I}_d$, we have

$$\begin{aligned}Q(x_1,x_j) &= (x_1-x_j)^T\left(\frac{\Sigma(x_1)+\Sigma(x_j)}{2}\right)^{-1}(x_1-x_j) \\ &= 2\|x_1-x_j\|_2^2\Big(G(x_1)+G(x_j)\Big)^{-1},\end{aligned}$$

where $\|\cdot\|_2$ is the Euclidean norm, and

$$\frac{2^{\frac{d}{2}}\det(\Sigma(x_1))^{\frac{1}{4}}\det(\Sigma(x_j))^{\frac{1}{4}}}{\det(\Sigma(x_1)+\Sigma(x_j))^{\frac{1}{2}}} = \left(\frac{4G(x_1)G(x_j)}{\big(G(x_1)+G((x_j)\big)^2}\right)^{\frac{d}{4}}.$$

We now study separately three cases:

i) $x_j = x_1$: we have

$$\lim_{G(x_1)\to 0}\left(\frac{4G(x_1)G(x_1)}{\big(G(x_1)+G(x_1)\big)^2}\right)^{\frac{d}{4}} = \lim_{G(x_1)\to 0} 1 = 1, \tag{18}$$

and so using the algebra of limits, the continuity of ρ_{S}, (18) and the fact that $\rho_{\mathrm{S}}(0) = 1$, we have

$$\lim_{G(x_1)\to 0}\rho(x_1,x_1) = \lim_{G(x_1)\to 0}\rho_{\mathrm{S}}\Big(\sqrt{Q(x_1,x_1)}\Big) = \rho_{\mathrm{S}}(0) = 1.$$

ii) $x_j \neq x_1$ and $G(x_j) > 0$: we have

$$\lim_{G(x_1)\to 0} Q(x_1,x_j) = 2\|x_1-x_j\|_2^2\Big(G(x_j)\Big)^{-1},$$

and

$$\lim_{G(x_1)\to 0}\left(\frac{4G(x_1)G(x_j)}{\big(G(x_1)+G(x_j)\big)^2}\right)^{\frac{d}{4}} = 0. \tag{19}$$

Thus, using the continuity of ρ_{S}, together with (19) and the algebra of limits, we have

$$\lim_{G(x_1)\to 0}\rho(x_1,x_j) = 0.$$

iii) $x_j \neq x_1, G(x_j) = 0$: we obtain

$$\lim_{G(x_1),G(x_j)\to 0} Q(x_1, x_j) = \infty,$$

which by Assumptions 2(ii) implies that

$$\lim_{G(x_1),G(x_j)\to 0} \rho_S\Big(\sqrt{Q(x_1, x_j)}\Big) = 0.$$

Since $(a+b)^2 \geq 4ab$ for any positive numbers a and b, we have

$$0 \leq \left(\frac{4G(x_1)G(x_j)}{\big(G(x_1) + G((x_j)\big)^2}\right)^{\frac{d}{4}} \leq 1,$$

for any $G(x_1) > 0, G(x_j) > 0$, and hence

$$\lim_{G(x_1),G(x_j)\to 0} \rho(x_1, x_j) = 0.$$

Hence, when $G(x_i) > 0$, for $i = 2, \ldots, N$, we have $\lim_{G(x_1)\to 0} \mathbf{R} = \mathbf{R}^*$, where the matrix $\mathbf{R}^*$ has the first row and column equal to the first basis vector $e_1 = (1, 0, 0, \ldots, 0) \in \mathbb{R}^N$, and the remaining submatrix $\mathbf{R}^*_{N-1} \in \mathbb{R}^{N-1\times N-1}$ with entries $\rho(x_i, x_j)$, for $i, j = 2, \ldots, N$. The matrix $\mathbf{R}^*_{N-1}$ is positive definite by Proposition 1, from which we can conclude that $\mathbf{R}^*$ is positive definite also. A similar argument holds when $G(x_i) = 0$ for one or more indices $i \in \{2, \ldots, N\}$. ■

References

Neil K Chada, Marco A Iglesias, Lassi Roininen, and Andrew M Stuart. Parameterizations for ensemble Kalman inversion. *Inverse Problems*, 34(5):055009, 2018.

Victor Chen, Matthew M Dunlop, Omiros Papaspiliopoulos, and Andrew M Stuart. Robust MCMC sampling with non-Gaussian and hierarchical priors in high dimensions. arXiv preprint arXiv:1803.03344, 2018.

Simon L Cotter, Gareth O Roberts, Andrew M Stuart, and David White. MCMC methods for functions: modifying old algorithms to make them faster. *Statistical Science*, 28(3): 424–446, 2013.

Kurt Cutajar, Edwin V Bonilla, Pietro Michiardi, and Maurizio Filippone. Random feature expansions for deep Gaussian processes. In *International Conference on Machine Learning*, pages 884–893, 2017.

Zhenwen Dai, Andreas Damianou, Javier González, and Neil Lawrence. Variational auto-encoded deep Gaussian processes. arXiv preprint arXiv:1511.06455, 2015.

Andreas C Damianou and Neil D Lawrence. Deep Gaussian Processes. In *Artificial Intelligence and Statistics*, pages 207–215, 2013.

Yair Daon and Georg Stadler. Mitigating the influence of the boundary on PDE-based covariance operators. *Inverse Problems and Imaging*, 12(5), 2018.

Masoumeh Dashti and Andrew M Stuart. The Bayesian approach to inverse problems. *Handbook of Uncertainty Quantification*, 2017.

Persi Diaconis and David Freedman. Iterated random functions. *SIAM Review*, 41(1):45–76, 1999.

David K Duvenaud, Oren Rippel, Ryan P Adams, and Zoubin Ghahramani. Avoiding pathologies in very deep networks. In *Artificial Intelligence and Statistics*, pages 202–210, 2014.

Eldad Haber and Lars Ruthotto. Stable architectures for deep neural networks. *Inverse Problems*, 34(1):014004, 2017.

Martin Hairer. An introduction to stochastic PDEs. arXiv preprint arXiv:0907.4178, 2009.

Martin Hairer, Andrew M Stuart, Jochen Voss, and Petter Wiberg. Analysis of SPDEs arising in path sampling. Part I: The Gaussian case. *Communications in Mathematical Sciences*, 3(4):587–603, 2005.

Markus Heinonen, Henrik Mannerström, Juho Rousu, Samuel Kaski, and Harri Lähdesmäki. Non-stationary Gaussian process regression with Hamiltonian Monte Carlo. In *Artificial Intelligence and Statistics*, pages 732–740, 2016.

Dave Higdon, Marc Kennedy, James C Cavendish, John A Cafeo, and Robert D Ryne. Combining field data and computer simulations for calibration and prediction. *SIAM Journal on Scientific Computing*, 26(2):448–466, 2004.

Marco A Iglesias, Yulong Lu, and Andrew M Stuart. A Bayesian level set method for geometric inverse problems. *Interfaces and Free Boundaries*, 18(2):181–217, 2016.

Jari Kaipio and Erkki Somersalo. *Statistical and Computational Inverse Problems*, volume 160. Springer Science & Business Media, 2006.

Gopinath Kallianpur. *Stochastic Filtering Theory*, volume 13. Springer Science & Business Media, 2013.

Marc C Kennedy and Anthony O'Hagan. Bayesian calibration of computer models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(3):425–464, 2001.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(4):423–498, 2011.

Jonathan C Mattingly, Andrew M Stuart, and Desmond J Higham. Ergodicity for SDEs and approximations: locally Lipschitz vector fields and degenerate noise. *Stochastic Processes and their Applications*, 101(2):185–232, 2002.

Sean P Meyn and Richard L Tweedie. *Markov Chains and Stochastic Stability*. Springer Science & Business Media, 2012.

Radford M Neal. *Bayesian Learning for Neural Networks*. PhD thesis, University of Toronto, 1995.

Radford M Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. arXiv preprint physics/9701026, 1997.

Christopher J Paciorek. *Nonstationary Gaussian processes for regression and spatial modelling*. PhD thesis, Carnegie Mellon University, 2003.

Christopher J Paciorek and Mark J Schervish. Nonstationary covariance functions for Gaussian process regression. *Advances in Neural Information Processing Systems*, 16:273–280, 2004.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

Allan Pinkus. Approximation theory of the MLP model in neural networks. *Acta Numerica*, 8:143–195, 1999.

Frank J Pinski, Gideon Simpson, Andrew M Stuart, and Hendrik Weber. Kullback–Leibler approximation for probability measures on infinite dimensional spaces. *SIAM Journal on Mathematical Analysis*, 47(6):4091–4122, 2015.

Carl E Rasmussen and Christopher K I Williams. Gaussian processes for machine learning. *the MIT Press*, 2(3):4, 2006.

James C Robinson. *Infinite-Dimensional Dynamical Systems: An Introduction to Dissipative Parabolic PDEs and the Theory of Global Attractors*, volume 28. Cambridge University Press, 2001.

Lassi Roininen, Janne M J Huttunen, and Sari Lasanen. Whittle-Matérn priors for Bayesian statistical inversion with applications in electrical impedance tomography. *Inverse Problems and Imaging*, 8(2):561–586, 2014.

Lassi Roininen, Mark Girolami, Sari Lasanen, and Markku Markkanen. Hyperpriors for Matérn fields with applications in Bayesian inversion. arXiv preprint arXiv:1612.02989, 2016.

Hugh Salimbeni and Marc Deisenroth. Doubly stochastic variational inference for deep Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 4588–4599, 2017.

Alexandra M Schmidt and Anthony O'Hagan. Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65(3):743–758, 2003.

Edward Snelson, Zoubin Ghahramani, and Carl E Rasmussen. Warped Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 337–344, 2004.

Michael L Stein. *Interpolation of Spatial Data: Some Theory for Kriging*. Springer, 1999.

Holger Wendland. *Scattered Data Approximation*, volume 17. Cambridge University Press, 2004.

Yaming Yu and Xiao-Li Meng. To center or not to center: That is not the question—an Ancillarity–Sufficiency Interweaving Strategy (ASIS) for boosting MCMC efficiency. *Journal of Computational and Graphical Statistics*, 20(3):531–570, 2011.

Journal of Machine Learning Research 19 (2018) 1-86 Submitted 3/18; Published 9/18

Fast MCMC Sampling Algorithms on Polytopes

Yuansi Chen[*,◇] YUANSI.CHEN@BERKELEY.EDU
Raaz Dwivedi[*,†] RAAZ.RSK@BERKELEY.EDU
Martin J. Wainwright[◇,†,‡] WAINWRIG@BERKELEY.EDU
Bin Yu[◇,†] BINYU@BERKELEY.EDU
Department of Statistics[◇]
Department of Electrical Engineering and Computer Sciences[†]
University of California, Berkeley
Voleon Group[‡]*, Berkeley*

Editor: Alexander Rakhlin

Abstract

We propose and analyze two new MCMC sampling algorithms, the Vaidya walk and the John walk, for generating samples from the uniform distribution over a polytope. Both random walks are sampling algorithms derived from interior point methods. The former is based on volumetric-logarithmic barrier introduced by Vaidya whereas the latter uses John's ellipsoids. We show that the Vaidya walk mixes in significantly fewer steps than the logarithmic-barrier based Dikin walk studied in past work. For a polytope in $\mathbb{R}^d$ defined by $n > d$ linear constraints, we show that the mixing time from a warm start is bounded as $\mathcal{O}\left(n^{0.5}d^{1.5}\right)$, compared to the $\mathcal{O}\left(nd\right)$ mixing time bound for the Dikin walk. The cost of each step of the Vaidya walk is of the same order as the Dikin walk, and at most twice as large in terms of constant pre-factors. For the John walk, we prove an $\mathcal{O}\left(d^{2.5} \cdot \log^4(n/d)\right)$ bound on its mixing time and conjecture that an improved variant of it could achieve a mixing time of $\mathcal{O}\left(d^2 \cdot \text{poly-log}(n/d)\right)$. Additionally, we propose variants of the Vaidya and John walks that mix in polynomial time from a deterministic starting point. The speed-up of the Vaidya walk over the Dikin walk are illustrated in numerical examples.

Keywords: MCMC methods, interior point methods, polytopes, sampling from convex sets

1. Introduction

Sampling from distributions is a core problem in statistics, probability, operations research, and other areas involving stochastic models (Geman and Geman, 1984; Brémaud, 1991; Ripley, 2009; Hastings, 1970). Sampling algorithms are a prerequisite for applying Monte Carlo methods to order to approximate expectations and other integrals. Recent decades have witnessed great success of Markov Chain Monte Carlo (MCMC) algorithms; for instance, see the handbook by Brooks et al. (2011) and references therein. These methods are based on constructing a Markov chain whose stationary distribution is equal to the target distribution, and then drawing samples by simulating the chain for a certain number of steps. An advantage of MCMC algorithms is that they only require knowledge of the target density up to a proportionality constant. However, the theoretical understanding of MCMC algorithms used in practice is far from complete. In particular, a general challenge is to bound the *mixing time* of a given MCMC algorithm, meaning the number of iterations—as a function of the error tolerance δ, problem dimension d and other parameters—for the chain to arrive at a distribution within distance δ of the target.

In this paper, we study a certain class of MCMC algorithms designed for the problem of drawing samples from the uniform distribution over a polytope. The polytope is specified in the form $\mathcal{K} := \{x \in \mathbb{R}^d \mid Ax \leq b\}$, parameterized by the matrix-vector pair $(A, b) \in \mathbb{R}^{n \times d} \times \mathbb{R}^n$. Our goal is to understand the mixing time for obtaining δ-accurate samples, and how it grows as a function of the pair (n, d).

The problem of sampling uniformly from a polytope is important in various applications and methodologies. For instance, it underlies various methods for computing randomized approximations to polytope volumes. There is a long line of work on sampling methods being used to obtain randomized approximations to the volumes of polytopes and other convex bodies (see, e.g., Lovász and Simonovits, 1990; Lawrence, 1991; Bélisle et al., 1993; Lovász, 1999; Cousins and Vempala, 2014). Polytope sampling is also useful in developing fast randomized algorithms for convex optimization (Bertsimas and Vempala, 2004) and sampling contingency tables (Kannan and Narayanan, 2012), as well as in randomized methods for approximately solving mixed integer convex programs (Huang and Mehrotra, 2013, 2015). Sampling from polytopes is also related to simulations of the hard-disk model in statistical physics (Kapfer and Krauth, 2013), as well as to simulations of error events for linear programming in communication (Feldman et al., 2005).

Many MCMC algorithms have been studied for sampling from polytopes, and more generally, from convex bodies. Some early examples include the Ball Walk (Lovász and Simonovits, 1990) and the hit-and-run algorithm (Bélisle et al., 1993; Lovász, 1999), which apply to sampling from general convex bodies. Although these algorithms can be applied to polytopes, they do not exploit any special structure of the problem. In contrast, the Dikin walk introduced by Kannan and Narayanan (2012) is specialized to polytopes, and thus can achieve faster convergence rates than generic algorithms. The Dikin walk was the first sampling algorithm based on a connection to interior point methods for solving linear programs. More specifically, as we discuss in detail below, it constructs proposal distributions based on the standard logarithmic barrier for a polytope. In a later paper, Narayanan (2016) extended the Dikin walk to general convex sets equipped with self-concordant barriers.

For a polytope defined by n constraints, Kannan and Narayanan (2012) proved an upper bound on the mixing time of the Dikin walk that scales linearly with n. In many applications, the number of constraints n can be much larger than the number of variables d. For example, we could imagine one using many hyperplane constraints to approximate complicated convex sets such as sphere or ellipsoid. For such problems, linear dependence on the number of constraints is not desirable. Consequently, it is natural to ask if it is possible to design a sampling algorithm whose mixing time scales in a sub-linear manner with the number of constraints. Our main contribution is to investigate and answer this question in affirmative—in particular, by designing and analyzing two sampling algorithms with provably faster convergence rates than the the Dikin walk while retaining its advantages over the ball walk and the hit-and-run methods.

. *Yuansi Chen and Raaz Dwivedi contributed equally to this work.

JMLR 19(55):1-86, 2018

Our contributions: We introduce and analyze a new random walk, which we refer to as the *Vaidya walk* since it is based on the *volumetric-logarithmic barrier* introduced by Vaidya (1989). We show that for a polytope in $\mathbb{R}^d$ defined by n-constraints, the Vaidya walk mixes in $\mathcal{O}\left(n^{1/2}d^{3/2}\right)$ steps, whereas the Dikin walk (Kannan and Narayanan, 2012) has mixing time bounded as $\mathcal{O}\left(nd\right)$. So the Vaidya walk is better in the regime $n \gg d$. We also propose the *John walk*, which is based on the *John ellipsoidal algorithm* in optimization. We show that the John walk has a mixing time of $\mathcal{O}\left(d^{2.5} \cdot \log^4(n/d)\right)$ and conjecture that a variant of it could achieve $\mathcal{O}\left(d^2 \cdot \text{poly-log}(n/d)\right)$ mixing time. We show that when compared to the Dikin walk, the per-iteration computational complexities of the Vaidya walk and the John walk are within a constant factor and a poly-logarithmic in n/d factor respectively. Thus, in the regime $n \gg d$, the overall upper bound on the complexity of generating an approximately uniform sample follows the order Dikin walk $\gg$ Vaidya walk $\gg$ John walk.

The remainder of the paper is organized as follows. In Section 2, we discuss many polynomial-time random walks on convex sets and polytopes, and motivate the starting point for the new random walks. In Section 3, we introduce the new random walks and state bounds on their rates of convergence and provide a sketch of the proof in Section 3.5. We discuss the computational complexity of the different random walks and demonstrate the contrast between the random walks for several illustrative examples in Section 4. We present the proof of the mixing time for the Vaidya walk in Section 5 and defer the analysis of the John walk to the appendix. We conclude with possible extensions of our work in Section 6.

Notation: For two sequences a_δ and b_δ indexed by $\delta \in I \subseteq \mathbb{R}$, we say that $a_\delta = \mathcal{O}\left(b_\delta\right)$ if there exists a universal constant $C > 0$ such that $a_\delta \leq C b_\delta$ for all $\delta \in I$. For a set $\mathcal{K} \subset \mathbb{R}^d$, the sets $\text{int}\left(\mathcal{K}\right)$ and $\mathcal{K}^c$ denote the interior and complement of $\mathcal{K}$ respectively. We denote the boundary of the set $\mathcal{K}$ by $\partial\mathcal{K}$. The Euclidean norm of a vector $x \in \mathbb{R}^d$ is denoted by $\|x\|_2$. For any square matrix M, we use $\det(M)$ and $\text{trace}(M)$ to denote the determinant and the trace of the matrix M respectively. For two distributions $\mathcal{P}_1$ and $\mathcal{P}_2$ defined on the same probability space $(\mathcal{X}, \mathcal{B}(\mathcal{X}))$, their total-variation (TV) distance is denoted by $\|\mathcal{P}_1 - \mathcal{P}_2\|_{\text{TV}}$ and is defined as follows

$$\|\mathcal{P}_1 - \mathcal{P}_2\|_{\text{TV}} = \sup_{A \in \mathcal{B}(\mathcal{X})} \left|\mathcal{P}_1(A) - \mathcal{P}_2(A)\right|.$$

Furthermore if $\mathcal{P}_1$ is absolutely continuous with respect to $\mathcal{P}_2$, then the KullbackLeibler divergence from $\mathcal{P}_2$ to $\mathcal{P}_1$ is defined as

$$\text{KL}(\mathcal{P}_1 \| \mathcal{P}_2) = \int_{\mathcal{X}} \log\left(\frac{d\mathcal{P}_1}{d\mathcal{P}_2}\right) d\mathcal{P}_1.$$

2. Background and problem set-up

In this section, we describe general MCMC algorithms and review the rates of convergence of existing random walks on convex sets. After introducing several random walks studied in past work, we introduce the Vaidya and John walks studied in this paper.

2.1 Markov chains and mixing

Suppose that we are interested in drawing samples from a *target distribution* π^* supported on a subset $\mathcal{X}$ of $\mathbb{R}^d$. A broad class of methods are based on first constructing a discrete-time Markov chain that is irreducible and aperiodic, and whose stationary distribution is equal to π^*, and then simulating this Markov chain for a certain number of steps k. As we describe below, the number of steps k to be taken is determined by a mixing time analysis.

In this paper, we consider the class of Markov chains that are of the *Metropolis-Hastings type* (Metropolis et al., 1953; Hastings, 1970); see the books by Robert (2004) and Brooks et al. (2011), as well as references therein, for further background. Any such chain is specified by an initial density π^0 over the set $\mathcal{X}$, and a *proposal function* $p : \mathcal{X} \times \mathcal{X} \in \mathbb{R}_+$, where $p(x, \cdot)$ is a density function for each $x \in \mathcal{X}$. At each time, given a current state $x \in \mathcal{X}$ of the chain, the algorithm first proposes a new vector $z \in \mathcal{X}$ by sampling from the proposal density $p(x, \cdot)$. It then accepts $z \in \mathcal{X}$ as the new state of the Markov chain with probability

$$\alpha(x, z) := \min\left\{1, \frac{\pi^*(z)p(z, x)}{\pi^*(x)p(x, z)}\right\}. \tag{1}$$

Otherwise, with probability equal to $1 - \alpha(x, z)$, the chain stays at x. Thus, the overall transition kernel p for the Markov chain is defined by the function

$$q(x, z) := p(x, z)\alpha(x, z) \qquad \text{for } z \neq x,$$

and a probability mass at x with weight $1 - \int_{\mathcal{X}} q(x, z)dz$. It should be noted that the purpose of the Metropolis-Hastings correction (1) is that ensure that the target distribution π^* satisfies the *detailed balanced condition*, meaning that

$$q(y, x)\pi^*(x) = q(x, y)\pi^*(y) \quad \text{for all } x, y \in \mathcal{X}. \tag{2}$$

It is straightforward to verify that the detailed balance condition (2) implies that the target density π^* is stationary for the Markov chain. Throughout this paper, we analyze the *lazy version* of the Markov chain, defined as follows: when at state x with probability $1/2$ the walk stays at x and with probability $1/2$ it makes a transition as per the original random walk. Given that the Markov chains discussed in this paper are also irreducible, the laziness ensures uniqueness of the stationary distribution.

Overall, this set-up defines an operator $\mathcal{T}_p$ on the space of probability distributions: given an initial distribution μ_0 with $\text{supp}(\mu_0) \subseteq \text{supp}(\pi^*)$, it generates a new distribution $\mathcal{T}_p(\mu_0)$, corresponding to the distribution of the chain at the next step. Moreover, for any positive integer $k = 1, 2, \ldots$, the distribution μ_k of the chain at time k is given by $\mathcal{T}_p^k(\mu_0)$, where $\mathcal{T}_p^k$ denotes the composition of $\mathcal{T}_p$ with itself k times. Furthermore, the transition distribution at any state x is given by $\mathcal{T}_p(\delta_x)$ where δ_x denotes the dirac-delta distribution with unit mass at x.

Given our assumptions and set-up, we are guaranteed that $\lim_{k\to\infty} \mathcal{T}_p^k(\mu_0) = \pi^*$—that is, if we were to run the chain for an infinite number of steps, then we would draw a sample from the target distribution π^*. In practice, however, any algorithm will be run only for a finite number of steps, which suffices to ensure only that the distribution from which the

sample has been drawn is "close" to the target π^*. In order to quantify the closeness, for a given tolerance parameter $\delta \in (0,1)$, we define the *δ-mixing time* as

$$k_{\text{mix}}(\delta;\mu_0) := \min\Big\{k \mid \|\mathcal{T}_p^k(\mu_0) - \pi^*\|_{\text{TV}} \le \delta\Big\}, \tag{3}$$

corresponding to the first time that the chain's distribution is within δ in TV norm of the target distribution, given that it starts with distribution μ_0.

In the analysis of Markov chains, it is convenient to have a rough measure of the distance between the initial distribution μ_0 and the stationary distribution. Warmness is one such measure: For a finite scalar M, the initial distribution μ_0 is said to be M-warm with respect to the stationary distribution π^* if

$$\sup_S \left(\frac{\mu_0(S)}{\pi^*(S)}\right) \le M, \tag{Warm-Start}$$

where the supremum is taken over all measurable sets S. A number of mixing time guarantees from past work (Lovász, 1999; Vempala, 2005) are stated in terms of this notion of M-warmness, and our results make use of it as well. In particular, we provide bounds on the quantity $\sup_{\mu_0 \in \mathcal{P}_M(\pi^*)} k_{\text{mix}}(\delta;\mu_0)$, where $\mathcal{P}_M(\pi^*)$ denotes the set of all distributions that are M-warm with respect to π^*. Naturally, as the value of M decreases, the task of generating samples from the target distribution gets easier. However, access to a warm-start may not be feasible for many applications and thus deriving bounds on mixing time of the Markov chain from a non warm-start is also desirable. Consequently, we provide modifications of our random walks which mix in polynomial time even from deterministic starting points.

2.2 Sampling from polytopes

In this paper, we consider the problem of drawing a sample uniformly from a polytope. Given a full-rank matrix $A \in \mathbb{R}^{n\times d}$ with $n \ge d$, we consider a polytope $\mathcal{K}$ in $\mathbb{R}^d$ of the form

$$\mathcal{K} := \big\{x \in \mathbb{R}^d \mid Ax \le b\big\}, \tag{4}$$

where $b \in \mathbb{R}^n$ is a fixed vector. Since the uniform distribution on the polytope $\mathcal{K}$ is the primary target distribution considered in the paper, in the sequel we use π^* exclusively to denote the uniform distribution on the polytope $\mathcal{K}$. There are various algorithms to sample a vector from the uniform distribution over $\mathcal{K}$, including the ball walk (Lovász and Simonovits, 1990) and hit-and-run algorithms (Lovász, 1999). To be clear, these two algorithms apply to the more general problem of sampling from a convex set; Table 1 shows their complexity, when applied to the polytope $\mathcal{K}$, relative to the Vaidya walk analyzed in this paper. Most closely related to our paper is the Dikin walk proposed by Kannan and Narayanan (2012), and a more general random walk on a Riemannian manifold studied by Narayanan (2016). Both of these random walks, as with the Vaidya and John walks, can be viewed as randomized versions of the interior point methods used to solve linear programs, and more generally, convex programs equipped with suitable barrier functions.

In order to motivate the form of the Vaidya and John walks proposed in this paper, we begin by discussing the ball walk and then the Dikin walk. For the sake of completeness, we end the section with a brief description another popular sampling algorithm Hit-and-run.

Ball walk: The ball walk of Lovász and Simonovits (1990) is simple to describe: when at a point $x \in \mathcal{K}$, it draws a new point u from a Euclidean ball of radius $r > 0$ centered at x. Here the radius r is a step size parameter in the algorithm. If the proposed point u belongs to the polytope $\mathcal{K}$, then the walk moves to u; otherwise, the walk stays at x. On the one hand, unlike the walks analyzed in this paper, the ball walk applies to any convex set, but on the other, its mixing time depends on the condition number $\gamma_{\mathcal{K}}$ of the set $\mathcal{K}$, given by

$$\gamma_{\mathcal{K}} = \inf_{R_{\text{in}},R_{\text{out}}>0}\Big\{\frac{R_{\text{out}}}{R_{\text{in}}} \quad | \quad \mathbb{B}(x,R_{\text{in}}) \subseteq \mathcal{K} \subseteq \mathbb{B}(y,R_{\text{out}}) \quad \text{for some } x,y \in \mathcal{K}\Big\}. \tag{5}$$

Mixing time of the ball walk has been improved greatly since it was introduced (Kannan et al., 1997, 2006; Lee and Vempala, 2018b). Nonetheless, as shown in Table 1, the mixing time of the ball walk gets slower when the condition of the set is large; for instance, it scales[1] as d^6 for a set with condition number $\gamma_{\mathcal{K}} = d^2$. One approach to tackle bad conditioning is to use rounding as a pre-processing step, where the set is rounded to bring it in a near-isotropic position, i.e., reduce the condition $\gamma_{\mathcal{K}}$ to near-constant before sampling from it. Nonetheless, these algorithms are themselves based on several rounds of sampling algorithms and the current best algorithm by Lovász and Vempala (2006b) puts a convex body into approximately isotropic position, i.e., $\mathcal{O}^*(\sqrt{d})$ rounding with a running time of $\mathcal{O}^*(d^4)$ where we have omitted the dependence on log-factors. If one has more information about the structure of the convex set (and not just oracle access as required by the ball walk), one can potentially exploit it to design fast sampling algorithms which are unaffected by the conditioning of the set thereby reducing the need of the (expensive) pre-processing step. One such algorithm is the Dikin walk for polytopes which we describe next.

Dikin walk: The Dikin walk (Kannan and Narayanan, 2012) is similar in spirit to the ball walk, except that it proposes a point drawn uniformly from a *state-dependent* ellipsoid known as the Dikin ellipsoid (Dikin, 1967; Nesterov and Nemirovskii, 1994). It then applies an accept-reject step to adjust for the difference in the volumes of these ellipsoids at different states. The state-dependent choice of the ellipsoid allows the Dikin walk to adapt to the boundary structure. A key property of the Dikin ellipsoid of unit radius—in contrast to the Euclidean ball that underlies the ball walk—is that it is always contained within $\mathcal{K}$, as is known from classic results on interior point methods (Nesterov and Nemirovskii, 1994). Furthermore, the Dikin walk is affine invariant, meaning that its behavior does not change under linear transformations of the problem. As a consequence, the Dikin mixing time does not depend on the condition number $\gamma_{\mathcal{K}}$. In a variant of this random walk (Narayanan, 2016), uniform proposals in the ellipsoid are replaced by Gaussian proposals with covariance specified by the ellipsoid, and it is shown that with high probability, the proposal falls within the polytope.

The Dikin walk is closely related to the interior point methods for solving linear programs. In order to understand the Vaidya and John walks, it is useful to understand this connection in more detail. Suppose that our goal is to optimize a convex function over the polytope $\mathcal{K}$. A barrier method is based on converting this constrained optimization problem to a sequence of unconstrained ones, in particular by using a barrier to enforce the linear

1. Although, very recently Lee and Vempala (2018b) improved the mixing time of the ball walk for isotropic sets which have $\gamma_{\mathcal{K}} = \mathcal{O}(\sqrt{d})$ improved from $\mathcal{O}\left(d^3\right)$ to $\mathcal{O}\left(d^{2.5}\right)$.

constraints defining the polytope. Letting $a_i^\top$ denote the i-th row vector of matrix A, the *logarithmic-barrier* for the polytope $\mathcal{K}$ given by the function

$$\mathcal{F}(x) := -\sum_{i=1}^{n} \log(b_i - a_i^T x). \tag{6}$$

For each $i \in [n]$, we define the scalar $s_{x,i} := (b_i - a_i^T x)$, and we refer to the vector $s_x := (s_{x,1}, \ldots, s_{x,n})^\top$ as the *slackness at* x.

Each step of an interior point algorithm (Boyd and Vandenberghe, 2004) involves (approximately) solving a linear system involving the Hessian of the barrier function, which is given by

$$\nabla^2 \mathcal{F}(x) := \sum_{i=1}^{n} \frac{a_i a_i^\top}{s_{x,i}^2}. \tag{7}$$

In the Dikin walk (Kannan and Narayanan, 2012), given a current iterate x, the algorithm chooses a point uniformly at random from the ellipsoid

$$\{u \in \mathbb{R}^d \mid (u-x)^\top D_x (u-x) \leq R\}, \tag{8}$$

where $D_x := \nabla^2 \mathcal{F}(x)$ is the Hessian of the log barrier function, and $R > 0$ is a user-defined radius. In an alternative form of the Dikin walk (Narayanan, 2016; Sachdeva and Vishnoi, 2016), the proposal vector $u \in \mathbb{R}^d$ is drawn randomly from a Gaussian centered at x, and with covariance equal to a scaled copy of $(D_x)^{-1}$. Note that in contrast to the ball walk, the proposal distribution now depends on the current state.

Vaidya walk: For the *Vaidya walk* analyzed in this paper, we instead generate proposals from the ellipsoids defined, for each $x \in \text{int}(\mathcal{K})$, by the positive definite matrix

$$V_x := \sum_{i=1}^{n} (\sigma_{x,i} + \beta_{\text{V}}) \frac{a_i a_i^\top}{s_{x,i}^2}, \qquad \text{where} \tag{9a}$$

$$\beta_{\text{V}} := d/n \quad \text{and} \quad \sigma_x := \left(\frac{a_1^\top (\nabla^2 \mathcal{F}_x)^{-1} a_1}{s_{x,1}^2}, \ldots, \frac{a_n^\top (\nabla^2 \mathcal{F}_x)^{-1} a_n}{s_{x,n}^2} \right)^\top. \tag{9b}$$

The entries of the the vector σ_x are known as the leverage scores assciated with the matrix $\nabla^2 \mathcal{F}_x$ from equation (7), and are commonly used to measure the importance of rows in a linear system (Mahoney, 2011). The matrix V_x is related to the Hessian of the function $x \mapsto \mathcal{V}_x$ given by

$$\mathcal{V}_x := \log\det \nabla^2 \mathcal{F}_x + \beta_{\text{V}} \mathcal{F}_x. \tag{10}$$

This particular combination of the *volumetric barrier* and the *logarithmic barrier* was introduced by Vaidya (1989) and Vaidya and Atkinson (1993) in the context of interior point methods, hence our name for the resulting random walk.

John walk: We now describe the John walk. For any vector $w \in \mathbb{R}^n$, let $W := \text{diag}(w)$ denote the diagonal matrix with $W_{ii} = w_i$ for each $i \in [n]$. Let $S_x = \text{diag}(s_x)$ denote the slackness matrix at x. It is easy to see that S_x is positive semidefinite for all $x \in \mathcal{K}$, and strictly positive definite for all $x \in \text{int}(\mathcal{K})$. The (scaled) inverse covariance matrix underlying the John walk is given by

$$J_x := \sum_{i=1}^{n} \zeta_{x,i} \frac{a_i a_i^\top}{s_{x,i}^2}, \tag{11}$$

where for each $x \in \text{int}(\mathcal{K})$, the weight vector $\zeta_x \in \mathbb{R}^n$ is obtained by solving the convex program

$$\zeta_x := \arg\min_{w \in \mathbb{R}^n} \left\{ \sum_{i=1}^{n} w_i - \frac{1}{\alpha_{\text{J}}} \log\det(A^\top S_x^{-1} W^{\alpha_{\text{J}}} S_x^{-1} A) - \beta_{\text{J}} \sum_{i=1}^{n} \log w_i \right\}, \tag{12}$$

with $\beta_{\text{J}} := d/2n$ and $\alpha_{\text{J}} := 1 - 1/\log_2(1/\beta_{\text{J}})$. Lee and Sidford (2014) proposed the convex program (12) associated with the *approximate John weights* ζ_x, with the aim of searching for the best member of a family of volumetric barrier functions. They analyzed the use of the John weights in the context of speeding up interior point methods for solving linear programs; here we consider them for improving the mixing time of a sampling algorithm. The convex program (12) is closely related to the problem of finding the largest ellipsoid at any interior point of the polytope, such that the ellipsoid is contained within the polytope. This problem of finding the largest ellipsoid was first studied by John (1948) who showed that each convex body in $\mathbb{R}^d$ contains a unique ellipsoid of maximal volume. The convex program (12) was used by Lee and Sidford (2014) to compute approximate John Ellipsoids for solving linear programs. In a recent work, Gustafson and Narayanan (2018) make use of the exact John ellipsoids and design a polynomial time sampling algorithm for polytopes. See Table 1 for the associated guarantees.

Hit-and-run: We conclude with a brief discussion with another popular sampling algorithm: Hit-and-run. It was introduced by Smith (1984) as a sampling algorithm for general distributions and it was later shown to have polynomial mixing time for sampling from convex sets (Lovász, 1999; Lovász and Vempala, 2003, 2006a). The algorithm proceeds as follows: when at point x, it firsts draws a random line through x and then samples from the one-dimensional marginal of the target distribution restricted to this line. For uniform sampling from convex sets, the second step simplifies to drawing a uniform point from the line restricted to the convex set. Mixing time bounds for this random walk are summarized in Table 1.

2.3 Mixing time comparisons of walks

Table 1 provides a summary of the mixing time bounds and per step complexity and the effective per sample complexity for various random walks, including the Vaidya and John walks analyzed in this paper. In addition to the Ball Walk, Hit-and-Run, Dikin, Vaidya and John walks, we also show scalings for the recently introduced Riemannian Hamiltonian Monte Carlo (RHMC) on polytopes by Lee and Vempala (2016) and the John's walk based on exact John ellipsoids studied by Gustafson and Narayanan (2018). The details of per

iteration cost for the new random walks is discussed in Section 4.1. We now compare and contrast the complexities of these random walks.

Unlike the Ball Walk or hit-and-run which are useful for general convex sets, the Dikin, Vaidya, John and RHMC walks are specialized for polytopes. These latter random walks exploit the definition of the polytope in a particular way so that the transition probability from a point x to y does not change under an affine transformation, i.e., $\mathbb{T}(x, y) = \mathbb{T}(Ax, Ay)$ where $\mathbb{T}$ denotes the transition kernel for the random walk. Consequently, the mixing time bounds for these random walks have no dependence on the condition number of the set $\gamma_{\mathcal{K}}$ (5). We can see from Table 1, that compared to the Ball walk and hit-and-run, Vaidya walk mixes significantly faster if $n \ll d\gamma_{\mathcal{K}}^2$. The condition number $\gamma_{\mathcal{K}}$ of polytopes with polynomially many faces can not be $\mathcal{O}(d^{\frac{1}{2}-\epsilon})$ for any $\epsilon > 0$ but can be arbitrarily larger, even exponential in dimension d (Kannan and Narayanan, 2012). For such polytopes, Vaidya walk mixes faster as long as $n \ll d^3$ (and even for larger n when $\gamma_{\mathcal{K}}$ is large). It takes $\mathcal{O}(\sqrt{n/d})$ fewer steps compared to Dikin walk and thus provides a practical speed up over all range of d.

From a warm start, the Riemannian Hamiltonian Monte Carlo on polytopes introduced by Lee and Vempala (2016) has $\mathcal{O}\left(nd^{2/3}\right)$ mixing time, and thus mixes faster (up to constants) compared than the Vaidya walk (respectively the John walk) when the number of constraints n is is bounded as $n \ll d^{5/3}$ (respectively $n \ll d^{11/6}$). For larger numbers of constraints, the Vaidya and John walks exhibit faster mixing. More generally, it is clear that the rate of John walk has *almost* the best order across all the walks for reasonably large values of $n \gg d^2$.

Finally, let us compare the (exact) John walk due to Gustafson and Narayanan (2018) with the (approximate) John walk studied in our paper. A notable feature of their random walk is that its mixing time is independent of the number of constraints and the per iteration cost also depends linearly on the number of constraints. Nonetheless, the dependence on d, for both the mixing time (d^7) and the per iteration cost ($nd^4 + d^8$) is quite poor. In contrast, the per iteration cost for our John walk is nd^2 and the mixing time has only a poly-logarithmic dependence on n.

2.4 Visualization of three walks' proposal distributions

In order to gain intuition about the three interior point based methods—namely, the Dikin, Vaidya and John walks—it is helpful to discuss how their underlying proposal distributions change as a function of the current point x. All three walks are based on Gaussian proposal distributions with inverse covariance matrices of the general form

$$\sum_{i=1}^{n} w_{x,i} \frac{a_i a_i^\top}{s_{x,i}^2},$$

where $w_{x,i} > 0$ corresponds to a state-dependent weight associated with the i-th constraint. The Dikin walk uses the weights $w_{x,i} = 1$; the Vaidya walk uses the weights $w_{x,i} = \sigma_{x,i} + \beta_{\mathrm{V}}$; and the John walk uses the weights $w_{x,i} = \zeta_{x,i}$. For simplicity, we refer to these weights as the Dikin, Vaidya and John weights. The i-th weight characterize the importance of the i-th linear constraint in constructing the inverse covariance matrix. A larger value of

Random walk	$\mathbf{k}_{\mathrm{mix}}(\delta; \mu_0)$	**Iteration cost**	**Per sample cost**
Ball walk$^{\#}$ (Kannan et al., 2006)	$d^2\gamma_{\mathcal{K}}^2$	nd	$nd^3\gamma_{\mathcal{K}}^2$
Hit-and-Run (Lovász and Vempala, 2006a)	$d^2\gamma_{\mathcal{K}}^2$	nd	$nd^3\gamma_{\mathcal{K}}^2$
Dikin walk (Kannan and Narayanan, 2012)	nd	nd^2	n^2d^3
RHMC walk (Lee and Vempala, 2018a)	$nd^{2/3}$	nd^2	$n^2d^{2.67}$
John's walk† (Gustafson and Narayanan, 2018)	d^7	$nd^4 + d^8$	$nd^{11} + d^{15}$
Vaidya walk (this paper)	$n^{1/2}d^{3/2}$	nd^2	$n^{1.5}d^{3.5}$
John walk (this paper)	$d^{5/2}\log^4\left(\frac{2n}{d}\right)$	$nd^2\log^2 n$	$nd^{4.5}$
Improved John walk‡ (this paper)	$d^2\,\kappa_{n,d}$	$nd^2\log^2 n$	nd^4

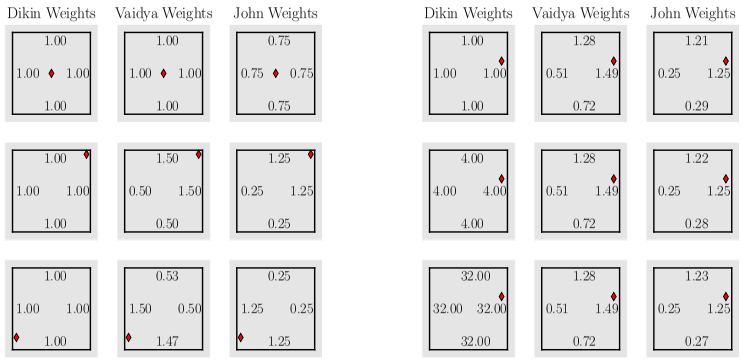
Table 1. Upper bounds on computational complexity of random walks on the polytope $\mathcal{K} = \{x \in \mathbb{R}^d | Ax \leq b\}$ defined by the matrix-vector pair $(A, b) \in \mathbb{R}^{n\times d} \times \mathbb{R}^n$ with a warm-start. For simplicity, here we ignore the logarithmic dependence on the warmness parameter and the tolerance δ. The iteration cost terms of order nd^2 arise from linear system solving, using standard and numerically stable algorithms, for n equations in d dimensions; algorithms with best possible theoretical complexity nd^ω for $\omega < 1.373$ are not numerically stable enough for practical use. $^{\#}$Mixing time of the Ball walk has been improved to $\mathcal{O}\left(d^2\gamma_{\mathcal{K}}\right)$ for near isotropic convex bodies by Lee and Vempala (2018b) during the submission period of this paper. While ball walk, Hit-and-run are affected by the condition number $\gamma_{\mathcal{K}}$ of the set, the Dikin and RHMC walks have quadratic dependence on the number of constraints n. †John's walk by Gustafson and Narayanan (2018) (based on the exact John ellipsoids) has linear dependence on n but poor dependence on d. In contrast, the Vaidya walk has sub-quadratic dependence on n and significantly better dependence on d. Furthermore, the John walk (based on approximate John's ellipsoids) analyzed in this paper has linear dependence with reasonable dependence on the dimensions d. ‡The mixing time bound for the improved John walk with poly-logarithmic factor $\kappa_{n,d}$ is conjectured.

the weight $w_{x,i}$ relative to the total weight $\sum_{i=1}^n w_{x,i}$ signifies more importance for the i-th linear constraint for the point x.

Figure 1a illustrates the difference in three weights as we move points inside the polytope $[-1, 1]^2$. When the point x is in the middle of the unit square formed by the four constraints, all walks exhibit equal weight for every constraint. When the point x is closer to the bottom-left boundary, the Vaidya and John weights assign larger weights to the bottom and the left constraints, while the weights for top and right constraints decrease. Note that the total sum of Vaidya weights and that of John weights remains constant independent of the position of the point x.

In Figure 1b-2b, we demonstrate that the Vaidya walk and the John walk are better at handling repeated constraints. Note that we can define the square $[-1, 1]^2$ as

$$[-1,1]^2 = \left\{ x \in \mathbb{R}^2 \,\middle|\, Ax \leq b, A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ 0 & -1 \end{bmatrix}, b = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \right\}. \tag{13}$$



(a) Weights for different locations and a fixed number of constraints n.

(b) Effective weights for a fixed location and different number of constraints n

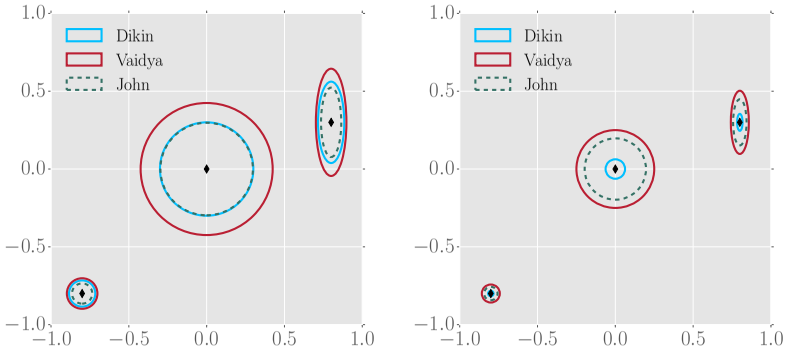
Figure 1. Visualization of the weights on the square with repeated constraints $\mathcal{S}_{n/4}$ for the different random walks. The number mentioned next to the boundary lines denotes the effective weight for the location x (denoted by diamond) for the corresponding constraint. **(a)** $n = 4$ is common across rows and $x = (0, 0)$ for the top row, $(0.9, 0.9)$ for the middle and $(-0.9, -0.7)$ for the bottom row. The Dikin weights are independent of x, the Vaidya and the John weights for a constraint increase if the location x is closer to it. **(b)** $x = (0.85, 0.30)$ is common across rows, and n=4 for the top row, $n = 16$ for the middle and n=128 for the bottom row. The effective Dikin weight for each constraint increases linearly with n but for the Vaidya and John walk adaptively, the weights get adjusted such that the sum of their weights is always of the order of the dimension d.

Simply repeating the rows of the matrix A several times changes the mathematical formulatiton of the polytope, but does not change the shape of the polytope. We define the square with constraints repeated $n/4$ times $\mathcal{S}_{n/4}$ as

$$\mathcal{S}_{n/4} = \left\{ x \in \mathbb{R}^2 \,\middle|\, A_{n/4} x \leq b_{n/4}, A_{n/4} = \begin{bmatrix} A \\ \vdots \\ \times(n/4) \end{bmatrix}, b_{n/4} = \begin{bmatrix} b \\ \vdots \\ \times(n/4) \end{bmatrix}, \right\} \tag{14}$$

where A and b were defined above. We denote effective weight for each distinct constraint as the sum of weights corresponding to the same constraint. Using this definition, the effective Dikin weight, which is $n/4$, is thus affected by the repeating of constraints. Consequently, the Dikin ellipsoid is much smaller for polytopes with repeated constraints. However, the Vaidya and John weights do not change as observed in the Figure 1b. Such a property of these two weights implies that the Vaidya and John ellipsoids are not too small even for very large number of constraints. And we observe such a phenomenon in Figures 2a-2b where the repetition of rows in the matrix A leads to very small Dikin ellipsoid but large Vaidya and John ellipsoid. A few other numerical computations also suggest that the Vaidya and John ellipsoids are moder adaptive when compared to Dikin ellipsoids when the

number of constraints is large. Nonetheless, such a claim is only based on heuristics and is presented simply to provide an intuition that the new ellipsoids are better behaved than Dikin ellipsoids and thereby motivated the design of the new random walks.



(a) $n = 32$

(b) $n = 2048$

Figure 2. Visualization of the proposal distribution on the square with repeated constraints $\mathcal{S}_{n/4}$ for the different random walks. **(a, b)** Unit ellipsoids associated with the covariances of the random walks at different states x on the square with repeated constraints $\mathcal{S}_{n/4}$. Clearly, all these ellipsoids adapt to the boundary but increasing n has a profound impact on the volume of the Dikin ellipsoids and comparatively less impact on the Vaidya and John ellipsoids.

3. Main results

With the basic background in place, we now describe the algorithms more precisely and state upper bounds on the mixing time of the Vaidya and John walks. In Section 3.4, we propose a variant of the John walk, known as the *improved John walk*, and conjecture that it has a better mixing time bound than that of the John walk.

3.1 Vaidya and John walks

In this subsection, we formally define the Vaidya and John walks. In Algorithm 1 and Algorithm 2, we summarize the steps of the Vaidya walk and the John walk.

Vaidya walk: The Vaidya walk with radius parameter $r > 0$, denoted by VW(r) for short, is defined by a Gaussian proposal distribution denoted as $\mathcal{P}_x^{\mathrm{v}}$: given a current state $x \in \mathrm{int}\,(\mathcal{K})$, it proposes a new point by sampling from the multivariate Gaussian distribution

$\mathcal{N}\left(x, \frac{r^2}{\sqrt{nd}} {V_x}^{-1}\right)$. In analytic terms, the proposal density at x is given by

$$p_x^{\mathrm{V}}(z) := p_{\mathrm{Vaidya}(r)}(x,z) = \sqrt{\det V_x}\left(\frac{nd}{2\pi r^2}\right)^{d/2} \exp\left(-\frac{\sqrt{nd}}{2r^2}\,(z-x)^\top V_x (z-x)\right). \quad (15)$$

As the target distribution for our walk is the uniform distribution on $\mathcal{K}$, the proposal step is followed by an accept-reject step as described in Section 2.1 (equation 1). Thus the overall transition distribution for the walk at state x is defined by a density given by

$$q_{\mathrm{Vaidya}(r)}(x,z) = \begin{cases} \min\left\{p_x^{\mathrm{V}}(z), p_z^{\mathrm{V}}(x)\right\}, & z \in \mathcal{K} \text{ and } z \neq x, \\ 0, & z \notin \mathcal{K}, \end{cases}$$

and a probability mass at x, given by $1 - \int_{z\in\mathcal{K}} \min\left\{p_x(z), p_z(x)\right\} dz$. We use $\mathcal{T}_{\mathrm{Vaidya}(r)}$ to denote the resulting transition operator for the Vaidya walk with parameter r.

Algorithm 1: Vaidya Walk with parameter r (VW(r))

Input: Parameter r and $x_0 \in \mathrm{int}\,(\mathcal{K})$
Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  With probability $\frac{1}{2}$ stay at the current state: $x_{i+1} \leftarrow x_i$ % *lazy step*
  With probability $\frac{1}{2}$ perform the following update:
    Proposal step: Draw $z_{i+1} \sim \mathcal{N}\left(x_i, \frac{r^2}{(nd)^{1/2}} V_{x_i}^{-1}\right)$
    Accept-reject step:
      if $z_{i+1} \notin \mathcal{K}$ *then* $x_{i+1} \leftarrow x_i$ % *reject an infeasible proposal*
      else
        compute $\alpha_{i+1} = \min\left\{1, p_{z_{i+1}}(x_{i+1})/p_{x_{i+1}}(z_{i+1})\right\}$
        With probability α_{i+1} accept the proposal: $x_{i+1} \leftarrow z_{i+1}$
        With probability $1 - \alpha_{i+1}$ reject the proposal: $x_{i+1} \leftarrow x_i$
end

John walk: The John walk is similar to the Vaidya walk except that the proposals at state $x \in \mathrm{int}\,(\mathcal{K})$ are generated from the multivariate Gaussian distribution $\mathcal{N}\left(x, \frac{r^2}{d^{3/2}\cdot\log_2^4(2n/d)} {J_x}^{-1}\right)$, where the matrix J_x is defined by equation (11), and $r > 0$ is a constant. The proposal distribution at $x \in \mathrm{int}\,(\mathcal{K})$ is denoted as $\mathcal{P}_x^{\mathrm{J}}$. The proposal step is then followed by an accept-reject step similarly defined as in the Vaidya walk. We use $\mathcal{T}_{\mathrm{John}(r)}$ to denote the resulting transition operator for the John walk with parameter r.

3.2 Mixing time bounds for warm start

We are now ready to state an upper bound on the mixing time of the Vaidya walk. In this and other theorem statements, we use c to denote a universal positive constant. Recall that π^* denotes the uniform distribution on the polytope $\mathcal{K}$, and, that $\mathcal{T}_{\mathrm{Vaidya}(r)}$ denotes the operator on distributions associated with the Vaidya walk.

Algorithm 2: John Walk with parameter r (JW(r))

Input: Parameter r and $x_0 \in \mathrm{int}\,(\mathcal{K})$
Output: Sequence $x_1, x_2, \ldots$

for $i = 0, 1, \ldots$ **do**
  With probability $\frac{1}{2}$ stay at the current state: $x_{i+1} \leftarrow x_i$ % *lazy step*
  With probability $\frac{1}{2}$ perform the following update:
    Proposal step: Draw $z_{i+1} \sim \mathcal{N}\left(x_i, \frac{r^2}{d^{3/2}} J_{x_i}^{-1}\right)$ % *this step is different than the Vaidya walk*
    Accept-reject step:
      if $z_{i+1} \notin \mathcal{K}$ *then* $x_{i+1} \leftarrow x_i$ % *reject an infeasible proposal*
      else
        compute $\alpha_{i+1} = \min\left\{1, p_{z_{i+1}}(x_{i+1})/p_{x_{i+1}}(z_{i+1})\right\}$
        With probability α_{i+1} accept the proposal: $x_{i+1} \leftarrow z_{i+1}$
        With probability $1 - \alpha_{i+1}$ reject the proposal: $x_{i+1} \leftarrow x_i$
end

Theorem 1 *Let μ_0 be any distribution that is M-warm with respect to π^* as defined in equation* (Warm-Start)*. For any $\delta \in (0, 1]$, the Vaidya walk with parameter $r_{\mathrm{V}} = 10^{-4}$ satisfies*

$$\|\mathcal{T}^k_{Vaidya(r_V)}(\mu_0) - \pi^*\|_{TV} \leq \delta \qquad \text{for all } \; k \geq cn^{1/2}d^{3/2}\log\left(\frac{\sqrt{M}}{\delta}\right). \quad (16)$$

The proof of Theorem 1 is provided in Section 5. Theorem 1 precisely quantifies the dependence of mixing time of the Vaidya walk on many parameters of interest such as dimension d, number of constraints n, the error tolerance δ and the warmness M. The specific choice $r_{\mathrm{V}} = 10^{-4}$ is for theoretical purposes; in practice, we find that substantially larger values can be used.[2] Our upper bound for the mixing time of the Vaidya walk has $\mathcal{O}(\sqrt{n/d})$ improvement over the current best upper bound for the mixing time of the Dikin walk. In Section 4.1, we show that the per iteration cost for the two walks is of the same order. Since $n \geq d$ for closed polytopes in $\mathbb{R}^d$, the effective cost until convergence (iteration complexity multiplied by number of iterations required) for the Vaidya walk is at least of the same order as of the Dikin walk, and significantly smaller when $n \gg d$. Comparing the provable mixing time upper bounds, the Vaidya walk has an advantage over the Dikin walk for the problems where the number of constraints is significantly larger than the number of variables involved. Our simulations also confirm this theoretical finding.

Let us now state our result for the mixing time of the John walk:

2. A larger than optimal r leads to an undesirable high rejection rate. In practice, we can fine tune r by performing a binary search over the interval $[10^{-4}, 1]$ and keeping track of the rejection rate of the samples during the run of the Markov chain for a given choice of r. A choice of $r > 1$ is obviously bad because then the Vaidya ellipsoid will have poor overlap with polytopes near the boundary, causing high rejection rate and slow down of the chain.

Theorem 2 *Suppose that $n \leq \exp(\sqrt{d})$, and let μ_0 be any distribution that is M-warm with respect to π^*. Then for any $\delta \in (0, 1]$, the John walk with parameter $r_{\scriptscriptstyle J} = 10^{-5}$ satisfies*

$$\|\mathcal{T}^k_{John(r_{\scriptscriptstyle J})}(\mu_0) - \pi^*\|_{TV} \leq \delta \qquad \text{for all } k \geq c\, d^{2.5}\, \log^4\left(\frac{n}{d}\right) \log\left(\frac{\sqrt{M}}{\delta}\right).$$

The proof of Theorem 2 is provided in Appendix D. Again the specific choice of $r_{\scriptscriptstyle J} = 10^{-5}$ is for theoretical purpose; in practice larger choices are possible. Note that the mixing time bound for the John walk depends only on the number of constraints n via a logarithmic factor, and so is almost independent of n. Consequently, it has a mixing time that is polynomial in d even if the number of constraints n scales exponentially in $\sqrt{d}$. Further, we show in Section 4.1 that the cost to execute one step of the John walk is of the same order as of the Dikin walk up to a poly-logarithmic factor in n. Thus, using John walk, we obtain improved mixing time bounds for the case when $n \gg d^2$.

3.3 Mixing time bounds from deterministic start

The mixing time bounds in Theorem 1 and 2 depend on the warmness M of the initial distribution. In some applications, it may not be easy to find an M-warm initial distribution. In such cases, we can consider starting the random walk from a deterministic point $x_0 \in \text{int}\,(\mathcal{K})$ that is not too close to the boundary $\partial\mathcal{K}$. Indeed, such a point can be found using standard optimization methods—e.g., using a Phase-I method for Newton's algorithm (see Boyd and Vandenberghe, 2004, Section 11.5.4).

Given such a deterministic initialization, our mixing time guarantees depend on the distance of the starting point from the boundary. This dependence involves the following notion of s-centrality:

Definition 3 *A point $x \in \text{int}\,(\mathcal{K})$ is called* s-central *if for any chord $\overline{ef}$ with end points $e, f \in \partial\mathcal{K}$ passing through x, we have $\|e - x\|_2 \,/\, \|f - x\|_2 \leq s$.*

Assuming that it is started at an s-central point x_0, the Dikin walk (Kannan and Narayanan, 2012, algorithm in section 2.1) has a polynomial mixing time. The authors showed that when the walk moves to a new state for the first time, the distribution of the iterate is $\mathcal{O}\left((\sqrt{n}s)^d\right)$-warm with respect to the distribution[3] π^*. Since only constant number of steps is required to get a warm start, for a deterministic start, we can just use the Dikin walk in the beginning to provide a warm start to the Vaidya (or John) walk. This motivates us to define the following hybrid walk.

Given an s-central point x_0, simulate the Dikin walk until we observe a new state. Note that due to *laziness* and the accept-reject step, the chain can stay at the starting point for several steps before making the first move a new state. Let k_1 denote the (random) number of steps taken to make the first move to a new state. After k_1 steps, we run the walk VW(r) with x_{k_1} as the initial point. We call such a walk as *s-central Dikin-start-Vaidya-walk* with parameter r. Let $\mathcal{T}_{\text{Dikin}}$ denote the transition kernel of the Dikin walk stated above. Then, we have the following mixing time bound for this hybrid walk.

3. Obtaining a warmness result for the Vaidya walk from a deterministic start from a central point is non-trivial and it is quite possible that the warmness does not improve. As a result, we simply invoke the established result for the Dikin walk.

Corollary 4 *Any s-central Dikin-start-Vaidya-walk with parameter $r = 10^{-4}$ satisfies*

$$\|\mathcal{T}^k_{Vaidya(r)}\left(\mathcal{T}^{k_1}_{Dikin}(\delta_{x_0})\right) - \pi^*\|_{TV} \leq \delta \qquad \text{for all } k \geq cn^{1/2}d^{5/2}\, \log\left(\frac{ns}{\delta}\right),$$

where k_1 is a geometric random variable with $\mathbb{E}\,[k_1] \leq c'$, and $c, c' > 0$ are universal constants.

The mixing rate is logarithmic in ns and has an extra factor of d compared to the bounds in Theorem 1. However, guaranteeing a warm start for a general polytope is hard but obtaining a central point involves only a few steps of optimization. Consequently, the hybrid walk and the guarantees from Corollary 4 come in handy for all such cases. Once again we observe that the upper bounds for mixing time are improved by a factor of $\mathcal{O}(\sqrt{n/d})$ when compared to the Dikin walk from an s-central start (Kannan and Narayanan, 2012; Narayanan, 2016) which had a mixing time of $\mathcal{O}\left(nd^2\right)$. The proof follows immediately from Theorem 1 by Kannan and Narayanan (2012) and Theorem 1 of this paper and is thereby omitted.

In a similar fashion, we can provide a polynomial time guarantee for a modified John walk from a deterministic start. We can consider a hybrid random walk that starts at an s-central point, simulates the Dikin walk until it makes the first move to a new state, and from there onwards simulates the John walk. Such a chain would have a mixing time of $\mathcal{O}\left(d^{3.5}\text{poly-log}(n, d, s)\right)$. For brevity, we omit a formal statement of this result.

3.4 Conjecture on improved John walk

From our analysis, we suspect that it is possible to improve the mixing time bound of $\mathcal{O}\left(d^{2.5}\text{poly-log}(n/d)\right)$ in Theorem 2 by considering a variant of the John walk. In particular, we conjecture that a random walk with proposal distribution given by $\mathcal{N}\left(x, \frac{r^2}{d\cdot\text{poly-log}(n/d)} {J_x}^{-1}\right)$ for a suitable choice of r has an $\mathcal{O}\left(d^2\text{poly-log}(n/d)\right)$ mixing time from a warm start. We refer to this random walk as the *improved John walk*, and denote its transition operator by $\mathcal{T}_{\text{John}^+}$. Let us now give a formal statement of our conjecture on its mixing rate.

Conjecture 5 *Let μ_0 be any M-warm distribution. Then for any $\delta \in (0, 1]$, the improved John walk with parameter $r = r_0$, satisfies the bound*

$$\|\mathcal{T}^k_{John^+}(\mu_0) - \pi^*\|_{TV} \leq \delta \qquad \text{for all } k \geq c\, d^2\, \log_2^{c'}\left(\frac{2n}{d}\right) \log\left(\frac{\sqrt{M}}{\delta}\right),$$

where r_0, c, c' are universal constants.

Note that this conjecture involves quadratic (degree two) scaling in d; this exponent of two matches the sum of exponents for d and n in the mixing time bounds for both the Dikin and Vaidya walks from a warm-start. Consquently, the improved John walk would have better performance than the Dikin, Vaidya and John walks for almost all ranges of (n, d), apart from possible poly-logarithmic factors in the ratio n/d.

3.5 Proof sketch

In this subsection, we provide a high-level sketch of the main ingredients of the main proof. It is well-known that mixing of a Markov chain is closely related to its *conductance*. Our

main proof relies on the work by Lovász (1999) that characterizes the conductance of Markov chains on a convex set using Hilbert metric. Precisely, Lovász (1999) showed that a Markov chain has good conductance if it makes jumps to regions with large overlaps from two nearby points and the mixing time depends inversely on the maximum Hilbert metric between such nearby points. Using this argument, it remains to make sure that the ellipsoid radius is chosen properly such that the ellipsoids remain inside the polytope and the ellipsoids corresponding to two different points x and y overlap a lot even if the points x and y are relatively far apart.

The conductance-based argument has been used for analyzing the ball walk (Lovász and Simonovits, 1990, 1993), Hit-and-run (Lovász, 1999; Lovász and Vempala, 2006a) and the Dikin walk (Kannan and Narayanan, 2012; Narayanan, 2016; Sachdeva and Vishnoi, 2016). We refer the reader to the survey by Vempala (2005) for a thorough discussion about the relation between the conductance and mixing time for Markov chains. Our proof techniques share a few features with the recent analyses of the Dikin walk by Kannan and Narayanan (2012) and Sachdeva and Vishnoi (2016). However, new technical ideas are needed in order to handle the state-dependent weights σ_x and ζ_x, as defined in equations (9b) and (12) respectively, that underlie the proposal distributions for the Vaidya and John walks. Note that these techniques are not present in the analysis of the Dikin walk, which is based on constant weights.

Specifically, we present the proof of Theorem 1 on the mixing time of the Vaidya walk in Section 5 and defer the intermediate technical results to Appendix A, B and C. We present the proof of Theorem 2 (mixing time bound for the John walk) in Appendix D and provide related auxiliary results and their proofs in Appendices E, F, G, H and I. As alluded to earlier, to keep the paper self-contained, we provide the proof of Lovász's Lemma in Appendix J.

4. Numerical experiments

In this section, we first analyze the per-iteration cost to implement of three walks. We show that while the Dikin walk has the best per-iteration cost, the per-iteration cost of the Vaidya walk is only twice of that of Dikin walk and the per-iteration cost of the John walk is only of order $\log_2(2n/d)$ larger. Second, we demonstrate the speed-up gained by the Vaidya walk over the Dikin walk for a warm start on different polytopes.

4.1 Per iteration cost

We now show that the per iteration cost of the Dikin, Vaidya and John walks is of the same order. The proposal step of Vaidya walk requires matrix operations like matrix inversion, matrix multiplication and singular value decomposition (SVD). The accept-reject step requires computation of matrix determinants, besides a few matrix inverses and matrix-vector products. The complexity of all aforementioned operations is $\mathcal{O}\left(nd^2\right)$. Thus, per iteration computational complexity for the Vaidya walk is $\mathcal{O}\left(nd^2\right)$.[4]

4. In theory, the matrix computations for the Dikin walk can be carried out in time nd^{ν} for an exponent $\nu < 1.373$, but such algorithms are not numerically stable enough for practical use.

Both the Dikin and Vaidya walks requires an SVD computation for inverting the Hessian of Dikin barrier $\nabla^2\mathcal{F}_x$. In addition for the Vaidya walk, we have to invert the matrix V_x, which leads to almost twice the computation time of the Dikin walk per step. This difference can be observed in practice.

For the John walk, we need to compute the weights ζ_x at each point which involves solving the program (12). Lee and Sidford (2014) argued that the convex program (12) for obtaining John walk's weights is strongly convex with a suitably chosen norm. They proved that solving this program requires $\log^2 n$ number of gradient steps, where the computational complexity of each gradient step is equivalent to that of solving an $n \times d$ linear system ($\mathcal{O}\left(nd^2\right)$ using a numerically stable routine). Thus, the overall cost for the John walk is of the same order as of the Dikin walk up to a poly-logarithmic factor in the pair (n, d).

In practice, for the John walk, the combined effect of logarithmic factors in the number of steps and the cost to implement each step cannot be ignored. This extra factor becomes a bottleneck for the overall run time for the convergence of the Markov chain. Consequently, the John walk is not suitable for polytopes with moderate values of n and d, and its mixing time bounds are computationally superior to the Dikin and Vaidya walks only for the polytopes with $n \gg d \gg 1$.

4.2 Simulations

We now present simulation results for the random walks in $\mathbb{R}^d$ for $d = 2, 10$ and 50 with initial distribution $\mu_0 = \mathcal{N}(0, \sigma_d^2 \mathbb{I}_d)$ and target distribution being uniform, on the following polytopes:

Set-up 1 : The set $[-1, 1]^2$ defined by different number of constraints.

Set-up 2 : The set $[-1, 1]^d$ for $d \in \{2, 3, 4, 5, 6, 7\}$ for $n = \{2d, 2d^2, 2d^3\}$ constraints.

Set-up 3 : Symmetric polytopes in $\mathbb{R}^2$ with n-randomly-generated-constraints.

Set-up 4 : The interior of regular n-polygons on the unit circle.

Set-up 5 : Hyper cube $[-1, 1]^d$ for $d = 10$ and 50.

We choose σ_d such that the warmness parameter M is bounded by 100. We provide implementations of the Dikin, Vaidya and John walks in python and a jupyter notebook at the github repository `https://github.com/rzrsk/vaidya-walk`.

We use the following three ways to compare the convergence rate of the Dikin and the Vaidya walks: (1) comparing the approximate mixing time of a particular subset of the polytope—smaller value is associated with a faster mixing chain; (2) comparing the plot of the empirical distribution of samples from multiple runs of the Markov chain after k steps—if it appears *more uniform* for smaller k, the chain is deemed to be faster; and (3) contrasting the sequential plots of one dimensional projection of samples for a single long run of the chain—*less smooth* plot is associated with effective and fast exploration leading to a faster mixing (Yu and Mykland, 1998). Note that MCMC convergence diagnostics is a hard problem, especially in high dimensions, and since the methods outlined above are heuristic in nature we expect our experiments to not fully match our theoretical results.

In **Set-up 1**, we consider the polytope $[-1, 1]^2$ which can be represented by exactly 4 linear constraints (see Section 2.4). Suppose that we repeat the rows of the matrix A, and

then run the Dikin and Vaidya walks with the new A. Given the larger number of constraints, our theory predicts that the random walks should mix more slowly. In Figure 3c and 3d, we plot the empirical distribution obtained by the Dikin walk and Vaidya walk, starting from 200 i.i.d initial samples, for $n = 64$ and 2048. The empirical distribution plot shows that having large n significantly slows the mixing rate of the Dikin walk, while the effect on the Vaidya walk is much less. Further, we also plot the scaling of the approximate mixing time $\hat{k}_{\text{mix}}$ (defined below) for this simulation as a function of the number of constraints n in Figure 3b. For **Set-up 2**, we plot $\hat{k}_{\text{mix}}$ as a function of the dimensions d in Figures 3e-3g, for the random walks on $[-1, 1]^d$ where the hypercube is parametrized by different number of constraints $n \in \{2d, 2d^2, 2d^3\}$. The approximate mixing time is defined with respect to the set $\mathcal{S}_d = \{x \in \mathbb{R}^d | \, |x_i| \geq c_d \ \forall i \in [d]\}$ where c_d is chosen such that $\pi^*(\mathcal{S}_d) = 1/2$. In particular, for a fixed value of n, let $\hat{\mathbb{T}}^k$ denote the empirical measure after k-iterations across 2000 experiments. The approximate mixing time $\hat{k}_{\text{mix}}$ is defined as

$$\hat{k}_{\text{mix}} := \min \left\{ k \middle| \pi^*(\mathcal{S}_d) - \hat{\mathbb{T}}^k(\mathcal{S}_d) \leq \frac{1}{20} \right\}, \tag{17}$$

We choose such a set since the set covers the regions near to the boundary of the polytope which are not covered well by the chosen initial distribution. We make the following observations:

1. The slopes of the best-fit lines, for $\hat{k}_{\text{mix}}$ versus n in the log-log plot in Figure 3b, are 0.88 and 0.45 for Dikin and Vaidya walks respectively. This observation reflects a near-linear and sub-linear dependence on n for a fixed d for the mixing time of the Dikin walk and the Vaidya walk respectively.

2. In Figures 3e-3g, once again we observe a more significant effect of increasing the number of constraints on the approximate mixing time $\hat{k}_{\text{mix}}$. We list the slopes of the best fit lines on these log-log plots in Table 2. These slopes correspond to the exponents for d for the approximate mixing time. From the table, we can observe that these experiments agree with the mixing time bounds of $\mathcal{O}(nd)$ for the Dikin walk and $\mathcal{O}\left(n^{0.5}d^{1.5}\right)$ for the Vaidya walk.

No. of Constraints	DW Theoretical	VW Theoretical	DW Experiments	VW Experiments
$n = 2d$	2.0	2.0	1.58	1.72
$n = 2d^2$	3.0	2.5	2.80	2.48
$n = 2d^3$	4.0	3.0	3.84	2.75

Table 2. Value of the exponent of dimensions d for the theoretical bounds on mixing time and the observed approximate mixing time of the Dikin walk (DW) and the Vaidya walk (VW) for $[-1, 1]^d$ described by $n = 2d, 2d^2, 2d^3$ constraints. The theoretical exponents are based on the mixing time bounds of $\mathcal{O}(nd)$ for the Dikin walk and $\mathcal{O}\left(n^{0.5}d^{1.5}\right)$ for the Vaidya walk. The experimental exponents are based on the results from the simulations described in **Set-up 2** in Section 4.2. Clearly, the exponents observed in practice are in agreement with the theoretical rates and imply the faster convergence of the Vaidya walk compared to the Dikin walk for large number of constraints.

In **Set-up 3**, we compare the plots of the empirical distribution of 200 runs of the Dikin walk and the Vaidya walk for different values of k, for symmetric polytopes in $\mathbb{R}^2$ with n-randomly-generated-constraints. We fix $b_i = 1$. To generate a_i, first we draw two uniform

random variables from $[0, 1]$ and then flip the sign of both of them with probability $1/2$ and assign these values to the vector a_i. The resulting polytope is always a subset of the square $\mathcal{K} = [-1, 1]^2$ and contains the diagonal line connecting the points $(-1, 1)$ and $(1, -1)$. From Figure 4a-4b, we observe that while there is no clear winner for the case $n = 64$, the Vaidya walk mixes mixes significantly faster than the Dikin walk for the polytope defined by 2048 constraints.

In **Set-up 4**, the constraint set is the regular n-polygons inscribed in the unit circle. A similar observation as in **Set-up 3** can be made from Figure 4c-4d: the Vaidya walk mixes at least as fast as the Dikin walk and mixes significantly faster for large n.

In **Set-up 5**, we examine the performance of the Dikin walk and the Vaidya walk on hyper-cube $[-1, 1]^d$ for $d = 10, 50$. We plot the one dimensional projections onto a random normal direction of all the samples from a single run up to $10,000$ steps. The Vaidya sequential plot looks more jagged than that of the Dikin walk for $d = 10, n = 5120$. For other cases, we do not have a clear winner. Such an observation is consistent with the $\mathcal{O}(\sqrt{n/d})$ speed up of the Vaidya walk which is apparent when the ratio n/d is large.

5. Proofs

We begin with auxiliary results in Section 5.1 which we use then to prove Theorem 1 in Section 5.2. Proofs of the auxiliary results are in Sections 5.3 and 5.4, and we defer other technical results to appendices.

5.1 Auxiliary results

Our proof proceeds by formally establishing the following property for the Vaidya walk: if two points are close, then their one-step transition distribution are also close. Consequently, we need to quantify the closeness between two points and the associated transition distributions. We measure the distance between two points in terms of the cross ratio that we define next. For a given pair of points $x, y \in \mathcal{K}$, let $e(x), e(y) \in \partial\mathcal{K}$ denote the intersection of the chord joining x and y with $\mathcal{K}$ such that $e(x), x, y, e(y)$ are in order (see Figure 6a). The cross-ratio $d_{\mathcal{K}}(x, y)$ is given by

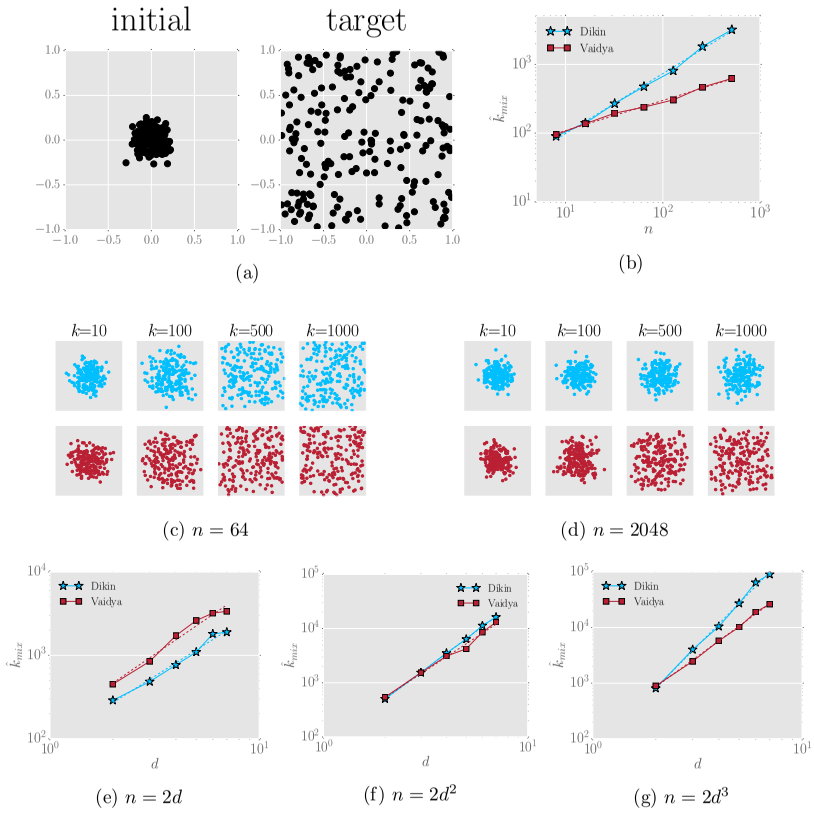
$$d_{\mathcal{K}}(x, y) := \frac{\|e(x) - e(y)\|_2 \, \|x - y\|_2}{\|e(x) - x\|_2 \, \|e(y) - y\|_2}. \tag{18}$$

The ratio $d_{\mathcal{K}}(x, y)$ is related to the Hilbert metric on $\mathcal{K}$, which is given by $\log(1 + d_{\mathcal{K}}(x, y))$; see the paper by Bushell (1973) for more details.

Consider a lazy reversible random walk on a bounded convex set $\mathcal{K}$ with transition operator $\mathcal{T}$ defined via the mapping $\mu_0 \mapsto \mu_0/2 + \tilde{\mathcal{T}}(\mu_0)/2$ and stationary with respect to the uniform distribution on $\mathcal{K}$ (denoted by π^*). (Recall that δ_x denote the dirac-delta distribution with unit mass at x.) The following lemma gives a bound on the mixing-time of the Markov chain.

Lemma 6 (Lovász's Lemma) *Suppose that there exist scalars $\rho, \Delta \in (0, 1)$ such that*

$$\|\tilde{\mathcal{T}}(\delta_x) - \tilde{\mathcal{T}}(\delta_y)\|_{TV} \leq 1 - \rho \qquad \text{for all } x, y \in \text{int}(\mathcal{K}) \text{ with } d_{\mathcal{K}}(x, y) < \Delta. \tag{19a}$$

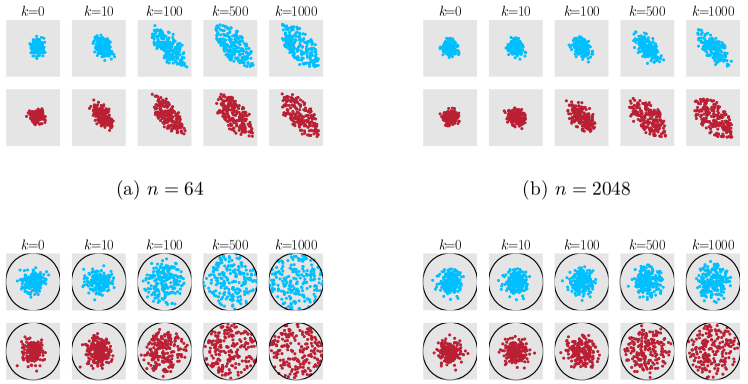


(a) (b) (c) $n = 64$ (d) $n = 2048$ (e) $n = 2d$ (f) $n = 2d^2$ (g) $n = 2d^3$

Figure 3. Comparison of the Dikin and Vaidya walks on the polytope $\mathcal{K} = [-1, 1]^2$. **(a)** Samples from the initial distribution $\mu_0 = \mathcal{N}(0, 0.04\,\mathbb{I}_2)$ and the uniform distribution on $[-1, 1]^2$. **(b)** Log-log plot of $\hat{k}_{\text{mix}}$ (17) versus the number of constraints (n) for a fixed dimension $d = 2$. **(c, d)** Empirical distribution of the samples for the Dikin walk (blue/top rows) and the Vaidya walk (red/bottom rows) for different values of n at iteration $k = 10, 100, 500$ and 1000. **(e, f, g)** Log-log plot of $\hat{k}_{\text{mix}}$ vs the dimension d, for $n \in \{2d, 2d^2, 2d^3\}$ for $d \in \{2, 3, 4, 5, 6, 7\}$. The exponents from these plots are summarized in Table 2. Note that increasing the number of constraints n has more profound effect on the Dikin walk in almost all the cases.

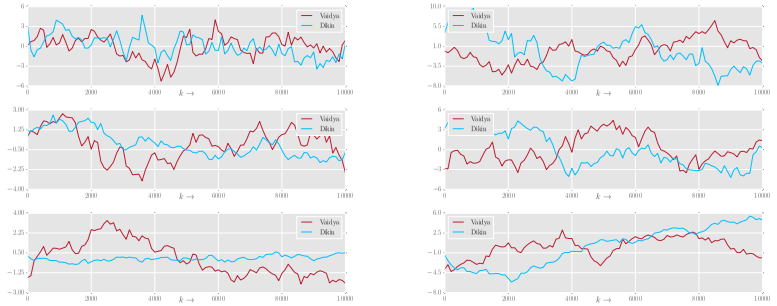
Then for every distribution μ_0 that is M-warm with respect to π^*, the lazy transition operator $\mathcal{T}$ satisfies

$$\|\mathcal{T}^k(\mu_0) - \pi^*\|_{TV} \leq \sqrt{M} \exp\left(-k\,\frac{\Delta^2 \rho^2}{4096}\right) \quad \forall \;\; k = 1, 2, \ldots. \tag{19b}$$



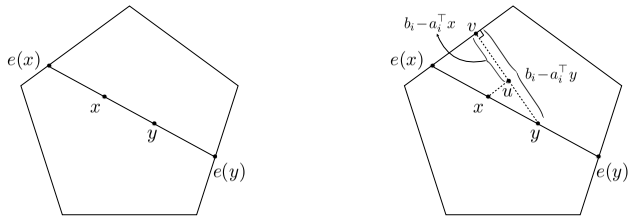
(a) $n = 64$ (b) $n = 2048$ (c) $n = 64$ (d) $n = 2048$

Figure 4. Empirical distribution of the samples from 200 runs for the Dikin walk (blue/top rows) and the Vaidya walk (red/bottom rows) at different iterations k. The 2-dimensional polytopes considered are: **(a, b)** random polytopes with n-constraints, and **(c, d)** regular n-polygons inscribed in the unit circle. For both sets of cases, we observe that higher n slows down the walks, with visibly more effect on the Dikin walk compared to the Vaidya walk.



(a) $d = 10$ (b) $d = 50$

Figure 5. Sequential plots of a one-dimensional random projection of the samples on the hyperbox $\mathcal{K} = [-1, 1]^d$, defined by n constraints. Each plot corresponds to one long run of the Dikin and Vaidya walks, and the projection is taken in a direction chosen randomly from the sphere. **(a)** Plots for $d = 10$ and $n \in \{20, 640, 5120\}$. **(b)** Plots for $d = 50$ and $n \in \{100, 400, 1600\}$. Relative to the Dikin walk, the Vaidya walk has a more jagged plot for pairs (n, d) in which the ratio n/d is relatively large: for instance, see the plots corresponding to $(n, d) = (640, 10)$ and $(5120, 10)$. The same claim cannot be made for pairs (n, d) for which the ratio n/d is relatively small; e.g., the plot with $(n, d) = (20, 10)$. These observations are consistent with our results that the Vaidya walk mixes more quickly by a factor of order $\mathcal{O}(\sqrt{n/d})$ over the Dikin walk.



(a) (b)

Figure 6. Polytope $\mathcal{K} = \{x \in \mathbb{R}^d | Ax \leq b\}$. (a) The points $e(x)$ and $e(y)$ denote the intersection points of the chord joining x and y with $\mathcal{K}$ such that $e(x), x, y, e(y)$ are in order. (b) A geometric illustration of the argument (23). It is straightforward to observe that $\|x-y\|_2/\|e(x)-x\|_2 = \|u-y\|_2/\|u-v\|_2 = \left|a_i^\top (y-x)\right| / \left(b_i - a_i^\top x\right)$.

This result is implicit in the paper by Lovász (1999), though not explicitly stated. In order to keep the paper self-contained, we provide a proof of this result in Appendix J.

Our proof of Theorem 1 is based on applying Lovász's Lemma; the main challenge in our work is to establish that our random walks satisfy the condition (19a) with suitable choices of Δ and ρ. In order to proceed with the proof, we require a few additional notations. Recall that the slackness at x was defined as $s_x := (b_1 - a_1^\top x, \ldots, b_n - a_n^\top x)^\top$. For all $x \in \text{int}\,(\mathcal{K})$, define the *Vaidya local norm of v at x* as

$$\|v\|_{V_x} := \left\|V_x^{1/2} v\right\|_2 = \sqrt{\sum_{i=1}^n (\sigma_{x,i} + \beta_{\text{V}}) \frac{(a_i^\top v)^2}{s_{x,i}^2}}, \tag{20a}$$

and the *Vaidya slack sensitivity at x* as

$$\theta_{V_x} := \left(\left\|\frac{a_1}{s_{x,1}}\right\|_{V_x}^2, \ldots, \left\|\frac{a_n}{s_{x,n}}\right\|_{V_x}^2\right)^\top = \left(\frac{a_1^\top V_x^{-1} a_1}{s_{x,1}^2}, \ldots, \frac{a_n^\top V_x^{-1} a_n}{s_{x,n}^2}\right)^\top. \tag{20b}$$

Similarly, we define the *John local norm of v at x* and the *John slack sensitivity at x* as

$$\|v\|_{J_x} := \left\|J_x^{1/2} v\right\|_2 \quad \text{and} \quad \theta_{J_x} := \left(\left\|\frac{a_1}{s_{x,1}}\right\|_{J_x}^2, \ldots, \left\|\frac{a_n}{s_{x,n}}\right\|_{J_x}^2\right)^\top. \tag{20c}$$

The following lemma provides useful properties of the leverage scores σ_x from equation (9b), the weights ζ_x obtained from solving the program (12), and the slack sensitivities θ_{V_x} and θ_{J_x}.

Lemma 7 *For any $x \in \text{int}\,(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} \in [0,1]$ *for all* $i \in [n]$,

(b) $\sum_{i=1}^n \sigma_{x,i} = d$,

(c) $\theta_{V_x,i} \in \left[0, \sqrt{n/d}\right]$ *for all* $i \in [n]$,

(d) $\zeta_{x,i} \in [\beta_J, 1+\beta_J]$ *for all* $i \in [n]$,

(e) $\sum_{i=1}^n \zeta_{x,i} = 3d/2$, *and*

(f) $\theta_{J_x,i} \in [0,4]$ *for all* $i \in [n]$.

We prove this lemma in Section 5.3.

Let $\mathcal{P}_x^{\text{V}}$ to denote the proposal distribution of the random walk VW(r) at state x. Next, we state a lemma that shows that if two points $x, y \in \text{int}\,(\mathcal{K})$ are close in Vaidya local norm at x, then for a suitable choice of the parameter r, the proposal distributions $\mathcal{P}_x^{\text{V}}$ and $\mathcal{P}_y^{\text{V}}$ are close. In addition, we show that the proposals are accepted with high probability at any point $x \in \text{int}\,(\mathcal{K})$. To establish the latter result, we now define the non-lazy transition operator of the Vaidya walk. Since the Vaidya walk is lazy with probability $1/2$, there exists a valid (non-lazy) transition operator $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}$ such that for any distribution μ_0, we have

$$\mathcal{T}_{\text{Vaidya}(r)}(\mu_0) = \mu_0/2 + \tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\mu_0)/2.$$

We call $\tilde{\mathcal{T}}_{\text{Vaidya}}$ the non-lazy transition operator for the Vaidya walk. Note that the one-step non-lazy transition distribution $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$ denotes the distribution of proposals after the accept-reject step if the chain was not lazy. Thus to establish that proposals are accepted with high probability, it suffices to establish that the transition distribution $\tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$ at any point $x \in \mathcal{K}$ is close to the proposal distribution $\mathcal{P}_x^{\text{V}}$. We now state these two results formally:

Lemma 8 *There exists a continuous non-decreasing function $f : [0, 1/4] \to \mathbb{R}_+$ with $f(1/15) \geq 10^{-4}$ such that for any $\epsilon \in (0, 1/15]$, the random walk VW(r) with $r \in [0, f(\epsilon)]$ satisfies*

$$\|\mathcal{P}_x^V - \mathcal{P}_y^V\|_{TV} \leq \epsilon \quad \forall\ x, y \in \text{int}\,(\mathcal{K})\ \text{s.t.}\ \|x-y\|_{V_x} \leq \frac{\epsilon r}{2(nd)^{1/4}}, \quad \text{and} \tag{21a}$$

$$\|\tilde{\mathcal{T}}_{Vaidya(r)}(\delta_x) - \mathcal{P}_x^V\|_{TV} \leq 5\epsilon\ \forall\ x \in \text{int}\,(\mathcal{K}). \tag{21b}$$

See Section 5.4 for the proof of this lemma.

With these lemmas in hand, we are now equipped to prove Theorem 1. To simplify notation, for the rest of this section, we adopt the shorthands $\mathbb{T}_x = \tilde{\mathcal{T}}_{\text{Vaidya}(r)}(\delta_x)$, $\mathcal{P}_x = \mathcal{P}_x^{\text{V}}$ and $\|\cdot\|_{V_x} = \|\cdot\|_x$.

5.2 Proof of Theorem 1

In order to invoke Lovász's Lemma for the random walk VW(10^{-4}), we need to verify the condition (19a) for suitable choices of ρ and Δ. Doing so involves two main steps:

(A): First, we relate the cross-ratio $d_{\mathcal{K}}(x,y)$ to the local norm (20a) at x.

(B): Second, we use Lemma 8 to show that if $x, y \in \text{int}\,(\mathcal{K})$ are close in local-norm, then the transition distributions $\mathbb{T}_x$ and $\mathbb{T}_y$ are close in TV-distance.

Step (A): We claim that for all $x, y \in \text{int}(\mathcal{K})$, the cross-ratio can be lower bounded as

$$d_{\mathcal{K}}(x,y) \geq \frac{1}{\sqrt{2d}} \|x-y\|_x . \tag{22}$$

Note that we have

$$d_{\mathcal{K}}(x,y) = \frac{\|e(x)-e(y)\|_2 \|x-y\|_2}{\|e(x)-x\|_2 \|e(y)-y\|_2} \overset{(i)}{\geq} \max\left\{\frac{\|x-y\|_2}{\|e(x)-x\|_2}, \frac{\|x-y\|_2}{\|e(y)-y\|_2}\right\} \overset{(ii)}{\geq} \max\left\{\frac{\|x-y\|_2}{\|e(x)-x\|_2}, \frac{\|x-y\|_2}{\|e(y)-x\|_2}\right\},$$

where step (i) follows from the inequality $\|e(x)-e(y)\|_2 \geq \max\{\|e(y)-y\|_2, \|e(x)-x\|_2\}$; and step (ii) follows from the inequality $\|e(x)-x\|_2 \leq \|e(y)-x\|_2$. Furthermore, from Figure 6b, we observe that

$$\max\left\{\frac{\|x-y\|_2}{\|e(x)-x\|_2}, \frac{\|x-y\|_2}{\|e(y)-x\|_2}\right\} = \max_{i\in[n]} \left|\frac{a_i^\top (x-y)}{s_{x,i}}\right|. \tag{23}$$

This argument of equation (14) has also been used (Sachdeva and Vishnoi, 2016, lemma 9). Note that maximum of a set of non-negative numbers is greater than the mean of the numbers. Combining this fact with properties (a) and (b) from Lemma 7, we find that

$$d_{\mathcal{K}}(x,y) \geq \sqrt{\frac{1}{\sum_{i=1}^n (\sigma_{x,i}+\beta_{\text{V}})} \sum_{i=1}^n (\sigma_{x,i}+\beta_{\text{V}}) \frac{(a_i^\top (x-y))^2}{s_{x,i}^2}} = \frac{\|x-y\|_x}{\sqrt{2d}},$$

thereby proving the claim (22).

Step (B): By the triangle inequality, we have

$$\|\mathbb{T}_x - \mathbb{T}_y\|_{\text{TV}} \leq \|\mathbb{T}_x - \mathcal{P}_x\|_{\text{TV}} + \|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} + \|\mathcal{P}_y - \mathbb{T}_y\|_{\text{TV}}.$$

Thus, for any (r,ϵ) such that $\epsilon \in [0, 1/15]$ and $r \leq f(\epsilon)$, Lemma 8 implies that

$$\|\mathbb{T}_x - \mathbb{T}_y\|_{\text{TV}} \leq 11\epsilon, \quad \forall x, y \in \text{int}(\mathcal{K}) \text{ such that } \|x-y\|_x \leq \frac{r\epsilon}{2(nd)^{1/4}}.$$

Consequently, the walk VW(r) satisfies the assumptions of Lovász's Lemma with

$$\Delta := \frac{1}{\sqrt{2d}} \cdot \frac{r\epsilon}{2(nd)^{1/4}} \quad \text{and} \quad \rho := 1 - 11\epsilon.$$

Since $f(1/15) \geq 10^{-4}$, we can set $\epsilon = 1/15$ and $r = 10^{-4}$, whence

$$\Delta^2\rho^2 = \frac{(1-11\epsilon)^2\epsilon^2 r^2}{8d\sqrt{nd}} = \frac{4^2}{15^2}\frac{1}{15^2}\frac{1}{10^{-8}} \cdot \frac{1}{d\sqrt{nd}} \geq 10^{-12}\frac{1}{d\sqrt{nd}}.$$

Observing that $\Delta < 1$ yields the claimed upper bound for the mixing time of Vaidya Walk.

5.3 Proof of Lemma 7

In order to prove part (a), observe that for any $x \in \text{int}(\mathcal{K})$, the Hessian $\nabla^2\mathcal{F}_x := \sum_{i=1}^n a_i a_i^\top / s_{x,i}^2$ is a sum of rank one positive semidefinite (PSD) matrices. Also, we can write $\nabla^2\mathcal{F}_x = A_x^\top A_x$ where

$$A_x := \begin{bmatrix} a_1^\top / s_{x,1} \\ \vdots \\ a_n^\top / s_{x,n} \end{bmatrix}.$$

Since $\text{rank}(A_x) = d$, we conclude that the matrix $\nabla^2\mathcal{F}_x$ is invertible and thus, both the matrices $\nabla^2\mathcal{F}_x$ and $\left(\nabla^2\mathcal{F}_x\right)^{-1}$ are PSD. Since $\sigma_{x,i} = a_i^\top \left(\nabla^2\mathcal{F}_x\right)^{-1} a_i / s_{x,i}^2$, we have $\sigma_{x,i} \geq 0$. Further, the fact that $a_i a_i^\top / s_{x,i}^2 \preceq \nabla^2\mathcal{F}_x$ implies that $\sigma_{x,i} \leq 1$.

Turning to the proof of part (b), from the equality $\text{trace}(AB) = \text{trace}(BA)$, we obtain

$$\sum_{i=1}^n \sigma_{x,i} = \text{trace}\left(\sum_{i=1}^n \frac{a_i^\top \left(\nabla^2\mathcal{F}_x\right)^{-1} a_i}{s_{x,i}^2}\right) = \text{trace}\left(\left(\nabla^2\mathcal{F}_x\right)^{-1} \sum_{i=1}^n \frac{a_i a_i^\top}{s_{x,i}^2}\right) = \text{trace}(\mathbb{I}_d) = d.$$

Now we prove part (c). Using the fact that $\sigma_{x,i} \geq 0$, and an argument similar to part (a) we find that that the matrices V_x and V_x^{-1} are PSD. Since $\theta_{V_x,i} = a_i^\top V_x^{-1} a_i / s_{x,i}^2$, we have $\theta_{V_x,i} \geq 0$. It is straightforward to see that $\beta_{\text{V}} \nabla^2\mathcal{F}_x \preceq V_x$ which implies that $\theta_{V_x,i} \leq \sigma_{x,i}/\beta$. Further, we also have $(\sigma_{x,i}+\beta_{\text{V}}) \frac{a_i a_i^\top}{s_{x,i}^2} \preceq V_x$ and whence $\theta_{V_x,i} \leq 1/(\sigma_{x,i}+\beta_{\text{V}})$. Combining the two inequalities yields the claim.

The other parts of the Lemma follow from Lemma 13, 14 and 15 by Lee and Sidford (2014) and are thereby omitted here.

5.4 Proof of Lemma 8

We prove the lemma for the following function

$$f(\epsilon) := \min\left\{\frac{1}{20\left(1+\sqrt{2}\log^{\frac{1}{2}}\left(\frac{4}{\epsilon}\right)\right)}, \frac{\epsilon}{\sqrt{18\log(2/\epsilon)}}, \sqrt{\frac{\epsilon}{86\sqrt{3}\chi_2}}, \frac{\epsilon}{22\sqrt{5/3}\chi_3}, \sqrt{\frac{\epsilon}{50\sqrt{105}\chi_4}}\right\}, \tag{24}$$

where $\chi_k = (2e/k \cdot \log(4/\epsilon))^{k/2}$ for $k = 2, 3$ and 4. A numerical calculation shows that $f(1/15) \geq 10^{-4}$.

5.4.1 Proof of claim (21a)

In order to bound the total variation distance $\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}}$, we apply Pinsker's inequality, which provides an upper bound on the TV-distance in terms of the KL divergence:

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} \leq \sqrt{2\,\text{KL}(\mathcal{P}_x \| \mathcal{P}_y)}.$$

For Gaussian distributions, the KL divergence has a closed form expression. In particular, for two normal-distributions $\mathcal{G}_1 = \mathcal{N}(\mu_1, \Sigma_1)$ and $\mathcal{G}_2 = \mathcal{N}(\mu_2, \Sigma_2)$, the Kullback-Leibler

divergence between the two is given by

$$\mathrm{KL}(\mathcal{G}_1\|\mathcal{G}_2) = \frac{1}{2}\Big(\mathrm{trace}(\Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}) - d - \log\det(\Sigma_1^{-1/2}\Sigma_2\Sigma_1^{-1/2}) + (\mu_1-\mu_2)^\top\Sigma_1^{-1}(\mu_1-\mu_2)\Big).$$

Recall from equation (15) that the proposal distribution for Vaidya walk is Gaussian, i.e., $\mathcal{P}_x = \mathcal{N}\left(x, \frac{r}{\sqrt{nd}}V_x^{-1}\right)$. Substituting $\mathcal{G}_1 = \mathcal{P}_x$ and $\mathcal{G}_2 = \mathcal{P}_y$ into the above expression and applying Pinsker's inequality, we find that

$$\begin{aligned}\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}}^2 \le 2\,\mathrm{KL}(\mathcal{P}_y\|\mathcal{P}_x) &= \mathrm{trace}(V_x^{-1/2}V_yV_x^{-1/2}) - d - \log\det(V_x^{-1/2}V_yV_x^{-1/2}) + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2 \\ &= \left\{\sum_{i=1}^d\left(\lambda_i - 1 + \log\frac{1}{\lambda_i}\right)\right\} + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2, \qquad (25)\end{aligned}$$

where $\lambda_1, \ldots, \lambda_d > 0$ denote the eigenvalues of the matrix $V_x^{-1/2}V_yV_x^{-1/2}$, and we have used the facts that $\det(V_x^{-1/2}V_yV_x^{-1/2}) = \prod_{i=1}^d\lambda_i$ and $\mathrm{trace}(V_x^{-1/2}V_yV_x^{-1/2}) = \sum_{i=1}^d\lambda_i$. The following lemma is useful in bounding expression (25).

Lemma 9 *For any scalar $t \in [0, 1/12]$ and any pair $x, y \in \mathrm{int}(\mathcal{K})$ such that $\|x-y\|_x \le t/(nd)^{1/4}$, we have*

$$\left(1 - \frac{8t}{\sqrt{d}}\right)\mathbb{I}_d \preceq V_x^{-1/2}V_yV_x^{-1/2} \preceq \left(1 + \frac{8t}{\sqrt{d}}\right)\mathbb{I}_d,$$

where $\preceq$ denotes ordering in the PSD cone, and $\mathbb{I}_d$ is the d-dimensional identity matrix.

See Appendix B for the proof of this lemma.

For $\epsilon \in (0, 1/15]$ and $r \in [0, 1/12]$, we have $t = \epsilon r/2 \le 1/12$, whence the eigenvalues $\{\lambda_i, i \in [d]\}$ can be sandwiched as

$$\frac{1}{2} \le 1 - \frac{4\epsilon r}{\sqrt{d}} \le \lambda_i \le 1 + \frac{4\epsilon r}{\sqrt{d}} \quad \text{for all } i \in d. \qquad (26)$$

We are now ready to bound the TV distance between $\mathcal{P}_x$ and $\mathcal{P}_y$. Using the bound (25) and the inequality $\log\omega \le \omega - 1$, valid for $\omega > 0$, we obtain

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}}^2 \le \sum_{i=1}^d\left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2.$$

Using the assumption that $\|x-y\|_x \le \epsilon r/\left(2(nd)^{1/4}\right)$, and plugging in the bounds (26) for the eigenvalues $\{\lambda_i, i \in [d]\}$, we find that

$$\sum_{i=1}^d\left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\sqrt{nd}}{r^2}\|x-y\|_x^2 \le 32\epsilon^2r^2 + \frac{\epsilon^2}{4}.$$

In asserting this inequality, we have used the facts that according to equation (26), for any $i \in [d]$,

$$\lambda_i - 2 + \frac{1}{\lambda_i} = \frac{(\lambda_i - 1)^2}{\lambda_i} \le 2\cdot\left(\frac{4\epsilon r}{\sqrt{d}}\right)^2.$$

Note that for any $r \in [0, 1/12]$ we have that $32r^2 \le 1/2$. Putting the pieces together yields $\|\mathcal{P}_x - \mathcal{P}_y\|_{\mathrm{TV}} \le \epsilon$, as claimed.

5.4.2 Proof of claim (21b)

Note that

$$\mathbb{T}_x(\{x\}) = \mathcal{P}_x(\mathcal{K}^c) + \int_{\mathcal{K}}\left(1 - \min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}\right)p_x(z)dz, \qquad (27)$$

where $\mathcal{K}^c$ denotes the complement of $\mathcal{K}$. Consequently, we find that

$$\begin{aligned}\|\mathcal{P}_x - \mathbb{T}_x\|_{\mathrm{TV}} &= \frac{1}{2}\left(\mathbb{T}_x(\{x\}) + \int_{\mathbb{R}^d}p_x(z)dz - \int_{\mathcal{K}}\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}p_x(z)dz\right) \\ &= \frac{1}{2}\left(2 - 2\int_{\mathbb{R}^d}\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}p_x(z)dz + 2\int_{\mathcal{K}^c}\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}p_x(z)dz\right) \\ &\le \underbrace{\mathcal{P}_x(\mathcal{K}^c)}_{=:S_1} + \underbrace{1 - \mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}\right]}_{=:S_2}, \qquad (28)\end{aligned}$$

Consequently, it suffices to show that both S_1 and S_2 are small, where the probability is taken over the randomness in the proposal z. In particular, we show that $S_1 \le \epsilon$ and $S_2 \le 4\epsilon$.

Bounding the term S_1: Since z is multivariate Gaussian with mean x and covariance $\frac{r^2}{\sqrt{nd}}V_x^{-1}$, we can write

$$z \overset{d}{=} x + \frac{r}{(nd)^{1/4}}V_x^{-1/2}\xi, \qquad (29)$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and $\overset{d}{=}$ denotes equality in distribution. Using equation (29) and definition (20b) of $\theta_{V_x,i}$, we obtain the bound

$$\frac{\left(a_i^\top(z-x)\right)^2}{s_{x,i}^2} = \frac{r^2}{(nd)^{\frac{1}{2}}}\left[\frac{a_i^\top V_x^{-1/2}\xi}{s_{x,i}}\right]^2 \overset{(i)}{\le} \frac{r^2}{(nd)^{\frac{1}{2}}}\theta_{V_x,i}\|\xi\|_2^2 \overset{(ii)}{\le} \frac{r^2}{d}\|\xi\|_2^2, \qquad (30)$$

where step (i) follows from Cauchy-Schwarz inequality, and step (ii) from the bound on $\theta_{V_x,i}$ from Lemma 7(c). Define the events

$$\mathcal{E} := \left\{\frac{r^2}{d}\|\xi\|_2^2 < 1\right\} \quad \text{and} \quad \mathcal{E}' := \{z \in \mathrm{int}(\mathcal{K})\}.$$

Inequality (30) implies that $\mathcal{E} \subseteq \mathcal{E}'$ and hence $\mathbb{P}[\mathcal{E}'] \ge \mathbb{P}[\mathcal{E}]$. Using a standard Gaussian tail bound and noting that $r \le \frac{1}{1+\sqrt{2/d\log(1/\epsilon)}}$, we obtain $\mathbb{P}[\mathcal{E}] \ge 1-\epsilon$ and whence $\mathbb{P}[\mathcal{E}'] \ge 1-\epsilon$. Thus, we have shown that $\mathbb{P}[z \notin \mathcal{K}] \le \epsilon$ which implies that $S_1 \le \epsilon$.

Bounding the term S_2: By Markov's inequality, we have

$$\mathbb{E}_{z\sim\mathcal{P}_x}\left[\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}\right] \ge \alpha\mathbb{P}\left[p_z(x) \ge \alpha p_x(z)\right] \quad \text{for all } \alpha \in (0, 1]. \qquad (31)$$

By definition (15) of p_x, we obtain

$$\frac{p_z(x)}{p_x(z)} = \exp\left(-\frac{\sqrt{nd}}{2r^2}\left(\|z-x\|_z^2 - \|z-x\|_x^2\right) + \frac{1}{2}\left(\log\det V_z - \log\det V_x\right)\right).$$

The following lemma provides us with useful bounds on the two terms in this expression, valid for any $x \in \operatorname{int}(\mathcal{K})$.

Lemma 10 *For any $\epsilon \in (0, 1/15]$ and $r \in (0, f(\epsilon)]$, we have*

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\frac{1}{2}\log\det V_z - \frac{1}{2}\log\det V_x \geq -\epsilon\right] \geq 1-\epsilon, \quad \textit{and} \tag{32a}$$

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\|z-x\|_z^2 - \|z-x\|_x^2 \leq 2\epsilon\frac{r^2}{\sqrt{nd}}\right] \geq 1-\epsilon. \tag{32b}$$

See Appendix C for the proof of this claim.

Using Lemma 10, we now complete the proof. For $r \leq f(\epsilon)$, we obtain

$$\frac{p_z(x)}{p_x(z)} \geq \exp(-2\epsilon) \geq 1-2\epsilon$$

with probability at least $1-2\epsilon$. Substituting $\alpha = 1-2\epsilon$ in inequality (31) yields that $S_2 \leq 4\epsilon$, as claimed.

6. Discussion

In this paper, we focused on improving mixing rate of MCMC sampling algorithms for polytopes by building on the advancements in the field of interior point methods. We proposed and analyzed two different barrier based MCMC sampling algorithms for polytopes that outperforms the existing sampling algorithms like the ball walk, the hit-and-run and the Dikin walk for a large class of polytopes. We provably demonstrated the fast mixing of the Vaidya walk, $\mathcal{O}\left(n^{0.5}d^{1.5}\right)$ and the John walk, $\mathcal{O}\left(d^{2.5}\text{poly-log}(n/d)\right)$ from a warm start. Our numerical experiments, albeit simple, corroborated with our theoretical claims: the Vaidya walk mixes at least as fast the Dikin walk and significantly faster when the number of constraints is quite large compared to the dimension of the underlying space. For the John walk, the logarithmic factors were dominant in all our experiments and thereby we deemed the result of importance only for set-ups with polytopes in very high dimensions with number of constraints overwhelmingly larger than the dimensions. Besides, proving the mixing time guarantees for the improved John walk (Conjecture 5) is still an open question.

Narayanan (2016) analyzed a generalized version of the Dikin walk for arbitrary convex sets equipped with self-concordant barrier. From his results, we were able to derive mixing time bounds of $\mathcal{O}\left(nd^4\right)$ and $\mathcal{O}\left(d^5\text{poly-log}(n/d)\right)$ from a warm start for the Vaidya walk and the John walk respectively. Our proof takes advantage of the specific structure of the Vaidya and John walk, resulting a better mixing rate upper bound the the general analysis provided by Narayanan (2016).

While our paper has mainly focused on sampling algorithms on polytopes, the idea of using logarithmic barrier to guide sampling can be extended to more general convex sets. The self-concordance property of the logarithmic barrier for polytopes is extended by Anstreicher (2000) to more general convex sets defined by semidefinite constraints, namely, linear matrix inequality (LMI) constraints. Moreover, Narayanan (2016) showed that for a convex set in $\mathbb{R}^d$ defined by n LMI constraints and equipped with the log-determinant barrier—the semidefinite analog of the logarithmic barrier for polytopes—the mixing time of the Dikin walk from a warm start is $\mathcal{O}\left(nd^2\right)$. It is possible that an appropriate Vaidya walk on such sets would have a speed-up over the Dikin walk. Narayanan and Rakhlin (2013) used the Dikin walk to generate samples from time varying log-concave distributions with appropriate scaling of the radius for different class of distributions. We believe that suitable adaptations of the Vaidya and John walks for such cases would provide significant gains.

Acknowledgements

This research was supported by Office of Naval Research grant DOD ONR-N00014 to MJW and in part by ARO grant W911NF1710005, NSF-DMS 1613002, the Center for Science of Information (CSoI), a US NSF Science and Technology Center, under grant agreement CCF-0939370 and the Miller Professorship (2016-2017) at UC Berkeley to BY. In addition, MJW was partially supported by National Science Foundation grant NSF-DMS-1612948 and RD was partially supported by the Berkeley Fellowship.

Appendix

Appendix A. Auxiliary results for the Vaidya walk

In this appendix, we first summarize a few notations used in the proofs related to Theorem 1, and collect the auxiliary results for the later proofs.

A.1 Notation

We begin with introducing the notation. Recall $A \in \mathbb{R}^{n\times d}$ is a matrix with $a_i^\top$ as its i-th row. For any positive integer p and any vector $v = (v_1, \ldots, v_p)^\top$, $\mathrm{diag}(v) = \mathrm{diag}(v_1, \ldots, v_p)$ denotes a $p\times p$ diagonal matrix with the i-th diagonal entry equal to v_i. Recall the definition of S_x:

$$S_x = \mathrm{diag}\left(s_{x,1}, \ldots, s_{x,n}\right) \text{ where } s_{x,i} = b_i - a_i^\top x \text{ for each } i \in [n]. \tag{33}$$

Furthermore, define $A_x = S_x^{-1}A$ for all $x \in \mathrm{int}\,(\mathcal{K})$, and let Υ_x denote the projection matrix for the column space of A_x, i.e.,

$$\Upsilon_x := A_x(A_x^\top A_x)^{-1}A_x^\top = A_x\nabla^2\mathcal{F}_x^{-1}A_x^\top. \tag{34}$$

Note that for the scores σ_x (9b), we have $\sigma_{x,i} = (\Upsilon_x)_{ii}$ for each $i \in [n]$. Let Σ_x be an $n \times n$ diagonal matrix defined as

$$\Sigma_x = \mathrm{diag}\left(\sigma_{x,1}, \ldots, \sigma_{x,n}\right). \tag{35}$$

Let $\sigma_{x,i,j} := (\Upsilon_x)_{ij}$, and let $\Upsilon_x^{(2)}$ denote the Hadamard product of Υ_x with itself, i.e.,

$$(\Upsilon_x^{(2)})_{ij} = \sigma_{x,i,j}^2 = \frac{\left(a_i^\top\nabla^2\mathcal{F}_x^{-1}a_j\right)^2}{s_{x,i}^2 s_{x,j}^2} \quad \text{for all } i,j \in [n]. \tag{36}$$

Using the shorthand $\theta_x := \theta_{V_x}$, we define

$$\Theta_x := \mathrm{diag}\left(\theta_{x,1}, \ldots, \theta_{x,m}\right) \text{ where } \theta_{x,i} = \frac{a_i^\top V_x^{-1}a_i}{s_{x,i}^2} \quad \text{for } i \in [n], \text{ and}$$

$$\Xi_x := (\theta_{x,i,j}^2) \text{ where } \theta_{x,i,j}^2 = \frac{\left(a_i^\top V_x^{-1}a_j\right)^2}{s_{x,i}^2 s_{x,j}^2} \quad \text{for } i,j \in [n].$$

In our new notation, we can re-write the Vaidya matrix V_x defined in equation (9a) as $V_x = A_x^\top\left(\Sigma_x + \beta_{\mathrm{V}}\mathbb{I}\right)A_x$, where $\beta_{\mathrm{V}} = d/n$.

A.2 Basic Properties

We begin by summarizing some key properties of various terms involved in our analysis.

Lemma 11 *For any vector $x \in \mathrm{int}\,(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} = \sum_{j=1}^n \sigma_{x,i,j}^2 = \sum_{j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}$ *for each* $i \in [n]$,

(b) $\Sigma_x \succeq \Upsilon_x^{(2)}$,

(c) $\sum_{i=1}^n \theta_{x,i}\,(\sigma_{x,i}+\beta_V) = d$,

(d) $\forall i \in [n],\ \theta_{x,i} = \sum_{j=1}^n (\sigma_{x,j}+\beta_V)\,\theta_{x,i,j}^2$, *for each* $i \in [n]$,

(e) $\theta_x^\top\,(\Sigma_x + \beta_V \mathbb{I})\,\theta_x = \sum_{i=1}^n \theta_{x,i}^2\,(\sigma_{x,i}+\beta_V) \le \sqrt{nd}$, *and*

(f) $\beta_V\,\nabla^2\mathcal{F}_x \preceq V_x \preceq (1+\beta_V)\,\nabla^2\mathcal{F}_x$.

where $\beta_V = d/n$ *was defined in equation* (9b).

Proof We prove each property separately.

Part (a): Using $\mathbb{I}_d = \nabla^2\mathcal{F}_x\left(\nabla^2\mathcal{F}_x\right)^{-1}$, we find that

$$\sigma_{x,i} = \frac{a_i^\top\left(\nabla^2\mathcal{F}_x\right)^{-1}\nabla^2\mathcal{F}_x\left(\nabla^2\mathcal{F}_x\right)^{-1}a_i}{s_{x,i}^2} = \frac{a_i^\top\left(\nabla^2\mathcal{F}_x\right)^{-1}\nabla^2\sum_{j=1}^n \frac{a_j^\top a_j}{s_{x,j}^2}\left(\nabla^2\mathcal{F}_x\right)^{-1}a_i}{s_{x,i}^2} = \sum_{i,j=1}^n \sigma_{x,i,j}^2.$$

Applying a similar trick twice and performing some algebra, we obtain

$$\sigma_{x,i} = \frac{a_i^\top\left(\nabla^2\mathcal{F}_x\right)^{-1}\nabla^2\mathcal{F}_x\left(\nabla^2\mathcal{F}_x\right)^{-1}\nabla^2\mathcal{F}_x\left(\nabla^2\mathcal{F}_x\right)^{-1}a_i}{s_{x,i}^2} = \sum_{i,j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}.$$

Part (b): From part (a), we have that $\Sigma_x - \Upsilon_x^{(2)}$ is a symmetric and diagonally dominant matrix with non-negative entries on the diagonal. Applying Gershgorin's theorem (Bhatia, 2013; Horn and Johnson, 2012), we conclude that it is PSD.

Part (c): Since $\mathrm{trace}(AB) = \mathrm{trace}(BA)$, we have

$$\sum_{i=1}^n \theta_{x,i}\,(\sigma_{x,i}+\beta_V) = \mathrm{trace}\left(V_x^{-1}\sum_{i=1}^n (\sigma_{x,i}+\beta_V)\,\frac{a_i a_i^\top}{s_{x,i}^2}\right) = \mathrm{trace}\,(\mathbb{I}_d) = d.$$

Part (d): An argument similar to part (a) implies that

$$\theta_{x,i} = \frac{a_i^\top V_x^{-1} V_x V_x^{-1} a_i}{s_{x,i}^2} = \frac{a_i^\top V_x^{-1}\sum_{j=1}^n (\sigma_{x,i}+\beta_V)\,\frac{a_j^\top a_j}{s_{x,j}^2} V_x^{-1} a_i}{s_{x,i}^2} = \sum_{i,j=1}^n (\sigma_{x,i}+\beta_V)\,\theta_{x,i,j}^2.$$

Part (e): Using part (c) and Lemma 7(c) yields the claim.

Part (f): The left inequality is by the definition of V_x. The right inequality uses the fact that $\Sigma_x \preceq \mathbb{I}_d$. ■

We now prove an important result that relates the *slackness* s_x and s_y at two points, in terms of $\|x-y\|_x$.

Lemma 12 *For all* $x, y \in \mathrm{int}\,(\mathcal{K})$, *we have*

$$\left|1 - \frac{s_{y,i}}{s_{x,i}}\right| \le \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x-y\|_x \quad \textit{for each } i \in [n].$$

Proof For any pair $x, y \in \mathrm{int}\,(\mathcal{K})$ and index $i \in [n]$, we have

$$\begin{aligned}\left(a_i^\top(x-y)\right)^2 = \left((V_x^{-\frac{1}{2}}a_i)^\top V_x^{\frac{1}{2}}(x-y)\right)^2 &\overset{(i)}{\le} \|V_x^{-\frac{1}{2}}a_i\|_2^2\,\|V_x^{\frac{1}{2}}(x-y)\|_2^2 \\ &= a_i^T V_x^{-1} a_i\ \|x-y\|_x^2 \\ &= \theta_{x,i}s_{x,i}^2\ \|x-y\|_x^2 \\ &\overset{(ii)}{\le} \sqrt{\frac{n}{d}}s_{x,i}^2\ \|x-y\|_x^2,\end{aligned}$$

where step (i) follows from the Cauchy-Schwarz inequality, and step (ii) uses the bound $\theta_{x,i}$ from Lemma 7(c). Noting the fact that $a_i^\top(x-y) = s_{y,i} - s_{x,i}$, the claim follows after simple algebra. ■

Appendix B. Proof of Lemma 9

In this appendix section, we prove Lemma 9 using results from the previous appendix. As a direct consequence of Lemma 12, we find that

$$\left|1 - \frac{s_{y,i}}{s_{x,i}}\right| \le \frac{t}{\sqrt{d}}, \quad \text{for any } x, y \in \mathrm{int}\,(\mathcal{K}) \text{ such that } \|x-y\|_x \le \frac{t}{(nd)^{1/4}}.$$

The Hessian $\nabla^2\mathcal{F}_y$ is thus sandwiched in terms of the Hessian $\nabla^2\mathcal{F}_x$ as

$$\left(1-\frac{t}{\sqrt{d}}\right)^2\nabla^2\mathcal{F}_x \preceq \nabla^2\mathcal{F}_y \preceq \left(1+\frac{t}{\sqrt{d}}\right)^2\nabla^2\mathcal{F}_x.$$

By the definition of $\sigma_{x,i}$ and $\sigma_{y,i}$, we have

$$\frac{\left(1-\frac{t}{\sqrt{d}}\right)^2}{\left(1+\frac{t}{\sqrt{d}}\right)^2}\sigma_{x,i} \le \sigma_{y,i} \le \frac{\left(1+\frac{t}{\sqrt{d}}\right)^2}{\left(1-\frac{t}{\sqrt{d}}\right)^2}\sigma_{x,i} \quad \text{for all } i \in [n]. \tag{37}$$

Consequently, we find that

$$\frac{\left(1-\frac{t}{\sqrt{d}}\right)^2}{\left(1+\frac{t}{\sqrt{d}}\right)^4}V_x \preceq V_y \preceq \frac{\left(1+\frac{t}{\sqrt{d}}\right)^2}{\left(1-\frac{t}{\sqrt{d}}\right)^4}V_x.$$

Note that

$$\frac{(1-\omega)^2}{(1+\omega)^4} \ge 1-8\omega \quad \text{and} \quad \frac{(1+\omega)^2}{(1-\omega)^4} \le 1+8\omega \quad \text{for any } \omega \in \left[0, \tfrac{1}{12}\right].$$

Applying this sandwiching pair of inequalities with $\omega = t/\sqrt{d}$ yields the claim.

Appendix C. Proof of Lemma 10

We begin by defining

$$\varphi_{x,i} := \frac{\sigma_{x,i} + \beta_{\mathrm{V}}}{s_{x,i}^2} \text{ for } i \in [n], \quad \text{and} \quad \Psi_x := \frac{1}{2} \log\det V_x, \quad \text{for all } x \in \operatorname{int}(\mathcal{K}). \tag{38}$$

Further, for any two points x and z, let $\overline{xz}$ denote the set of points on the line segment joining x and z. The proof of Lemma 10 is based on a Taylor series expansion, and so requires careful handling of σ, φ, Ψ and their derivatives. At a high level, the proof involves the following steps: (1) perform a Taylor series expansion around x and along the line segment $\overline{xz}$; (2) transfer the bounds of terms involving some point $y \in \overline{xz}$ to terms involving only x and z; and then (3) use concentration of Gaussian polynomials to obtain high probability bounds.

C.1 Auxiliary results for the proof of Lemma 10

We now introduce some auxiliary results involved in these three steps. The following lemma provides expressions for gradients of σ, φ and Ψ and bounds for directional Hessian of φ and Ψ. Let $e_i \in \mathbb{R}^d$ denote a vector with 1 in the i-th position and 0 otherwise. For any $h \in \mathbb{R}^d$ and $x \in \operatorname{int}(\mathcal{K})$, define $\eta_{x,h,i} = \eta_{x,i} := a_i^\top h / s_{x,i}$ for each $i \in [n]$.

Lemma 13 *The following relations hold;*

(a) Gradient of σ: $\nabla \sigma_{x,i} = 2A_x^\top (\Sigma_x - \Upsilon_x^{(2)}) e_i$ *for each* $i \in [n]$.

(b) Gradient of φ: $\nabla \varphi_{x,i} = \frac{2}{s_{x,i}^2} A_x^\top \left[2\Sigma_x + \beta_V \mathbb{I} - \Upsilon_x^{(2)}\right] e_i$ *for each* $i \in [n]$;

(c) Gradient of Ψ: $\nabla \Psi_x = A_x^\top \left(2\,\Sigma_x + \beta_V \mathbb{I} - \Upsilon_x^{(2)}\right) \theta_x$;

(d) Bound on $\nabla^2 \varphi$: $s_{x,i}^2 \left|\frac{1}{2} h^\top \nabla^2 \varphi_{x,i} h\right| \le 14\,(\sigma_{x,i} + \beta_V)\,\eta_{x,i}^2 + 11 \sum_{j=1}^n \sigma_{x,i,j}^2 \eta_{x,j}^2$ *for* $i \in [n]$;

(e) Bound on $\nabla^2 \Psi$: $\left|\frac{1}{2} h^\top \left(\nabla^2 \Psi_x\right) h\right| \le 13 \sum_{i=1}^n (\sigma_{x,i} + \beta_V)\,\theta_{x,i} \eta_{x,i}^2 + \frac{17}{2} \sum_{i,j=1}^n \sigma_{x,i,j}^2 \theta_{x,i} \eta_{x,j}^2$.

See Section C.6 for the proof of this claim.

The following lemma that shows that for a random variable $z \sim \mathcal{P}_x$, the slackness $s_{z,i}$ is close to $s_{x,i}$ with high probability.

Lemma 14 *For any $\epsilon \in (0, 1/4], r \in (0,1)$ and $x \in \operatorname{int}(\mathcal{K})$, we have*

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in \left(1 - r\,(1+\delta), 1 + r\,(1+\delta)\right)\right] \ge 1 - \epsilon/4,$$

where $\delta = \sqrt{\frac{2\log(4/\epsilon)}{d}}$. Thus for any $d \ge 1$ and $r \le 1/\left[20\left(1 + \sqrt{2\log\left(\frac{4}{\epsilon}\right)}\right)\right]$, we have

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in (0.95, 1.05)\right] \ge 1 - \epsilon/4.$$

See Section C.4 for the proof which is based on combining the bound on $\frac{s_{x,i}}{s_{v,i}}$ from Lemma 12 with standard Gaussian tail bounds.

This result comes in handy for transferring bounds for different expressions in Taylor expansion involving an arbitrary y on $\overline{xz}$ to bounds on terms involving simply x. The proof follows from Lemma 12 and a simple application of the standard Gaussian tail bounds and is thereby omitted. For brevity, we define the shorthand

$$\hat{a}_{x,i} = \frac{1}{s_{x,i}} V_x^{-1/2} a_i \quad \text{for each } i \in [n]. \tag{39}$$

In the following lemma, we state some tail bounds for particular Gaussian polynomials that arise in our analysis.

Lemma 15 *For any $\epsilon \in (0, 1/15]$, define $\chi_k = (2e/k \cdot \log(4/\epsilon))^{k/2}$ for $k = 2, 3$ and 4. Then for $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and any $x \in \operatorname{int}(\mathcal{K})$ the following high probability bounds hold:*

$$\mathbb{P}\left[\sum_{i=1}^n (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^2 \le \chi_2 \sqrt{3} d\right] \ge 1 - \frac{\epsilon}{4}, \tag{40a}$$

$$\mathbb{P}\left[\left|\sum_{i=1}^n (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^3\right| \le \chi_3 \sqrt{15}\,(nd)^{1/4}\right] \ge 1 - \frac{\epsilon}{4}, \tag{40b}$$

$$\mathbb{P}\left[\left|\sum_{i,j=1}^n \sigma_{x,i,j}^2 \left(\left(\frac{\hat{a}_{x,i} + \hat{a}_{x,j}}{2}\right)^\top \xi\right)^3\right| \le \chi_3 \sqrt{15}\,(nd)^{1/4}\right] \ge 1 - \frac{\epsilon}{4}, \tag{40c}$$

$$\mathbb{P}\left[\sum_{i=1}^n (\sigma_{x,i} + \beta_V)\left(\hat{a}_{x,i}^\top \xi\right)^4 \le \chi_4 \sqrt{105}\,(nd)^{1/2}\right] \ge 1 - \frac{\epsilon}{4}. \tag{40d}$$

See Section C.5 for the proof of these claims.

Now we summarize the final ingredients needed for our proofs. Recall that the Gaussian proposal z is related to the current state x via the equation

$$z \overset{d}{=} x + \frac{r}{(nd)^{1/4}} V_x^{-1/2} \xi, \tag{41}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. We also use the following elementary inequalities:

Cauchy-Schwarz inequality:	$\lvert u^\top v\rvert \le \lVert u\rVert_2 \lVert v\rVert_2$	(C-S)
AM-GM inequality:	$\nu\kappa \le \frac{1}{2}(\nu^2 + \kappa^2)$.	(AM-GM)
Sum of squares inequality:	$\frac{1}{2}\lVert a+b\rVert_2^2 \le \lVert a\rVert_2^2 + \lVert b\rVert_2^2$,	(SSI)

Note that the sum-of-squares inequality is simply a vectorized version of the AM-GM inequality. With these tools, we turn to the proof of Lemma 10. We split our analysis into parts.

C.2 Proof of claim (32a)

Using the second degree Taylor expansion, we have

$$\Psi_z - \Psi_x = (z-x)^\top \nabla\Psi_x + \frac{1}{2}(z-x)^\top \nabla^2\Psi_y (z-x), \quad \text{for some } y \in \overline{xz}.$$

We claim that for $r \le f(\epsilon)$, we have

$$\mathbb{P}_z\left[(z-x)^\top \nabla\Psi_x \ge -\epsilon/2\right] \ge 1-\epsilon/2, \quad \text{and} \tag{42a}$$

$$\mathbb{P}_z\left[\frac{1}{2}(z-x)\nabla^2\Psi_y (z-x) \ge -\epsilon/2\right] \ge 1-\epsilon/2. \tag{42b}$$

Note that the claim (32a) is a consequence of these two auxiliary claims, which we now prove.

C.2.1 Proof of bound (42a)

Equation (41) implies that $(z-x)^\top \nabla\Psi_x \sim \mathcal{N}\left(0, \frac{r^2}{\sqrt{nd}}\nabla\Psi_x^\top V_x^{-1}\nabla\Psi_x\right)$. We claim that

$$\nabla\Psi_x^\top V_x^{-1}\nabla\Psi_x \le 9\sqrt{nd} \quad \text{for all } x \in \operatorname{int}(\mathcal{K}). \tag{43}$$

We prove this inequality at the end of this subsection. Taking it as given for now, let $\xi' \sim \mathcal{N}(0, 9r^2)$. Then using inequality (43) and a standard Gaussian tail bound, we find that

$$\mathbb{P}\left[(z-x)^\top \nabla\Psi_x \ge -\omega\right] \ge \mathbb{P}\left[\xi' \ge -\omega\right] \ge 1 - \exp(-\omega^2/(18r^2)), \quad \text{valid for all } \omega \ge 0.$$

Setting $\omega = \epsilon/2$ and noting that $r \le \frac{\epsilon}{\sqrt{18\log(2/\epsilon)}}$ completes the claim.

C.2.2 Proof of bound (42b)

Let $\eta_{x,i} = \frac{a_i^\top(z-x)}{s_{x,i}} = \frac{r}{(mn)^{\frac{1}{4}}}\hat{a}_{x,i}^\top\xi$. Using Lemma 13(e), we have

$$\begin{aligned}\left|\frac{1}{2}(z-x)^\top\nabla^2\Psi_y(z-x)\right| &\le 13\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\theta_{y,i}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2 + \frac{17}{2}\sum_{i,j=1}^n\sigma_{y,i,j}^2\theta_{y,i}\frac{s_{x,j}^2}{s_{y,j}^2}\eta_{x,j}^2 \\ &\le \frac{43}{2}\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\frac{(\sigma_{y,i}+\beta_{\mathrm{V}})}{(\sigma_{x,i}+\beta_{\mathrm{V}})}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2. \end{aligned}\tag{44}$$

The last inequality comes from Lemma 7(c) and Lemma 11(a). Setting $\tau = 1.05$, we define the events $\mathcal{E}_1$ and $\mathcal{E}_2$ as follows:

$$\mathcal{E}_1 = \left\{\forall i \in [n], \frac{s_{x,i}}{s_{y,i}} \in [2-\tau, \tau]\right\}, \quad \text{and} \tag{45a}$$

$$\mathcal{E}_2 = \left\{\forall i \in [n], \frac{\sigma_{x,i}}{\sigma_{y,i}} \in \left[0, \frac{\tau^2}{(2-\tau)^2}\right]\right\}. \tag{45b}$$

It is straightforward to see that $\mathcal{E}_1 \subseteq \mathcal{E}_2$ following a similar argument we used to obtain equation (37) in the proof of Lemma 9. Since $r \le 1/\left[20\left(1+\sqrt{2}\log^{1/2}\left(\frac{4}{\epsilon}\right)\right)\right]$, Lemma 14 implies that $\mathbb{P}[\mathcal{E}_1] \ge 1-\epsilon/4$ whence $\mathbb{P}[\mathcal{E}_2] \ge 1-\epsilon/4$. Using these high probability bounds and the setting $\tau = 1.05$, we obtain that with probability at least $1-\epsilon/4$

$$\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\frac{(\sigma_{y,i}+\beta_{\mathrm{V}})}{(\sigma_{x,i}+\beta_{\mathrm{V}})}\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2 \le 2\sqrt{\frac{n}{d}}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\eta_{x,i}^2 = \frac{2r^2}{d}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})(\hat{a}_{x,i}^\top\xi)^2. \tag{46}$$

Applying the high probability bound Lemma 15 (40a) and the condition

$$r \le \sqrt{\frac{\epsilon}{86\sqrt{3}\chi_2}}, \tag{47}$$

we obtain that with probability at least $1-\epsilon/2$,

$$\frac{1}{2}(z-x)^\top\nabla^2\Psi_y(z-x) \ge -\epsilon/2,$$

as claimed.

C.2.3 Proof of bound (43)

We now return to prove our earlier inequality (43). Using the expression for the gradient $\nabla\Psi_x$ from Lemma 13(c), we have that for any vector $u \in \mathbb{R}^n$

$$\begin{aligned} u^\top\nabla\Psi_x\nabla\Psi_x^\top u &= \left\langle u, A_x^\top\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)\theta_x\right\rangle^2 \\ &= \left\langle A_x u, \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)\theta_x\right\rangle^2 \\ &= \left\langle (\Sigma_x+\beta_{\mathrm{V}}\mathbb{I})^{\frac{1}{2}}A_x u, (\Sigma_x+\beta_{\mathrm{V}}\mathbb{I})^{-1/2}\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)\theta_x\right\rangle^2 \\ &\le u^\top V_x u \cdot \theta_x^\top\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)(\Sigma_x+\beta_{\mathrm{V}}\mathbb{I})^{-1}\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)\theta_x \end{aligned}\tag{48}$$

where the last step follows from the Cauchy-Schwarz inequality. As a consequence of Lemma 11(b), the matrix $\Sigma_x - \Upsilon_x^{(2)}$ is PSD. Thus, we have

$$0 \preceq 2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I} \preceq 3(\Sigma_x + \beta_{\mathrm{V}}\mathbb{I}).$$

Consequently, we find that

$$0 \preceq \underbrace{(3\Sigma_x+3\beta_{\mathrm{V}}\mathbb{I})^{-1/2}\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)(3\Sigma_x+3\beta_{\mathrm{V}}\mathbb{I})^{-1/2}}_{=:L} \preceq \mathbb{I}.$$

We deduce that all eigenvalues of the matrix L lie in the interval $[0,1]$ and hence all the eigenvalues of the matrix L^2 belong to the interval $[0,1]$. As a result, we have

$$\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)(3\Sigma_x+3\beta_{\mathrm{V}}\mathbb{I})^{-1}\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right) \preceq (3\Sigma_x+3\beta_{\mathrm{V}}\mathbb{I}).$$

Thus, we obtain

$$\theta_x^\top \left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)(\Sigma_x + \beta_{\mathrm{V}}\mathbb{I})^{-1}\left(2\Sigma_x - \Upsilon_x^{(2)} + \beta_{\mathrm{V}}\mathbb{I}\right)\theta_x \le 9\theta_x^\top(\Sigma_x + \beta_{\mathrm{V}}\mathbb{I})\,\theta_x. \quad (49)$$

Finally, applying Lemma 11 and combining bounds (48) and (49) yields the claim.

C.3 Proof of claim (32b)

The quantity of interest can be written as

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \sum_{i=1}^n \left(a_i^\top(z-x)\right)^2(\varphi_{z,i} - \varphi_{x,i}).$$

We can write $z = x + \alpha u$, where α is a scalar and u is a unit vector in $\mathbb{R}^d$. Then we have

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \alpha^2 \sum_{i=1}^n \left(a_i^\top u\right)^2(\varphi_{z,i} - \varphi_{x,i}).$$

We apply a Taylor series expansion for $\sum_{i=1}^n \left(a_i^\top u\right)^2(\varphi_{z,i} - \varphi_{x,i})$ around the point x, along the line u. There exists a point $y \in \overline{xz}$ such that

$$\sum_{i=1}^n \left(a_i^\top u\right)^2(\varphi_{z,i} - \varphi_{x,i}) = \sum_{i=1}^n \left(a_i^\top u\right)^2\left((z-x)^\top\nabla\varphi_{x,i} + \frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x)\right).$$

Multiplying both sides by α^2, and using the shorthand $\eta_{x,i} = \frac{a_i^\top(z-x)}{s_{x,i}}$, we obtain

$$\|z-x\|_z^2 - \|z-x\|_x^2 = \sum_{i=1}^n \eta_{x,i}^2 s_{x,i}^2 (z-x)^\top\nabla\varphi_{x,i} + \sum_{i=1}^n \eta_{x,i}^2 s_{x,i}^2 \frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x). \quad (50)$$

Substituting the expression for $\nabla\varphi_{x,i}$ from Lemma 13(b) in equation (50) and performing some algebra, the first term on the RHS of equation (50) can be written as

$$\sum_{i=1}^n \eta_{x,i}^2 s_{x,i}^2 (z-x)^\top\nabla\varphi_{x,i} = 2\sum_{i=1}^n\left(\frac{7}{3}\sigma_{x,i} + \beta_{\mathrm{V}}\right)\eta_{x,i}^3 - \frac{1}{3}\sum_{i,j=1}^n \sigma_{x,i,j}^2\left(\eta_{x,i} + \eta_{x,j}\right)^3. \quad (51)$$

On the other hand, using Lemma 13 (d), we have

$$\frac{1}{2}s_{x,i}^2\left|(z-x)^\top\nabla^2\varphi_{y,i}(z-x)\right| \le \frac{s_{x,i}^2}{s_{y,i}^2}\left[14\left(\sigma_{y,i} + \beta_{\mathrm{V}}\right)\frac{s_{x,i}^2}{s_{y,i}^2}\eta_{x,i}^2 + 11\left(\sum_{j=1}^n \sigma_{y,i,j}^2\eta_{x,j}^2\frac{s_{x,j}^2}{s_{y,j}^2}\right)\right]. \quad (52)$$

Now, we use a fourth degree Gaussian polynomial to bound both the terms on the RHS of inequality (52). To do so, we use high probability bound for $s_{x,i}/s_{y,i}$. In particular, we use the high probability bounds for the events $\mathcal{E}_1$ and $\mathcal{E}_2$ defined in equations (45a) and (45b).

Multiplying both sides of inequality (52) by $\eta_{x,i}^2$ and summing over the index i, we obtain that with probability at least $1 - \epsilon/4$, we have

$$\begin{aligned}
\sum_{i=1}^n \eta_{x,i}^2 s_{x,i}^2\left|\frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x)\right| &\le \left[14\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\frac{s_{x,i}^4}{s_{y,i}^4}\eta_{x,i}^4 + 11\sum_{i,j=1}^n\sigma_{y,i,j}^2\eta_{x,i}^2\eta_{x,j}^2\frac{s_{x,i}^2 s_{x,j}^2}{s_{y,i}^2 s_{y,j}^2}\right] \\
&\overset{\text{(hpb.(45a))}}{\le} \tau^4\left[14\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\eta_{x,i}^4 + 11\sum_{i,j=1}^n\sigma_{y,i,j}^2\eta_{x,i}^2\eta_{x,j}^2\right] \\
&\overset{\text{(AM-GM)}}{\le} \tau^4\left[14\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\eta_{x,i}^4 + \frac{11}{2}\sum_{i,j=1}^n\sigma_{y,i,j}^2(\eta_{x,i}^4+\eta_{x,j}^4)\right] \\
&\overset{\text{(Lem. 11(a))}}{\le} 25\tau^4\sum_{i=1}^n(\sigma_{y,i}+\beta_{\mathrm{V}})\eta_{x,i}^4 \\
&\overset{\text{(hpb.(45b))}}{\le} 50\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\eta_{x,i}^4, \quad (53)
\end{aligned}$$

where "hpb" stands for high probability bound for events $\mathcal{E}_1$ and $\mathcal{E}_2$. In the last step, we have used the fact that $\tau^6/(2-\tau)^2 \le 2$ for $\tau = 1.05$. Combining equations (50), (51) and (53) and noting that $\eta_{x,i} = r\hat{a}_i^\top\xi/(nd)^{1/4}$, we find that

$$\begin{aligned}
\left|\|z-x\|_z^2 - \|z-x\|_x^2\right| &\le \frac{14}{3}\left|\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\eta_{x,i}^3\right| + \frac{8}{3}\left|\sum_{i,j=1}^n\sigma_{x,i,j}^2\left(\left(\eta_{x,i}+\eta_{x,j}\right)/2\right)^3\right| + 38\sum_{i=1}^n\sigma_{x,i}\eta_{x,i}^4 \\
&\le \frac{14}{3}\frac{r^3}{(nd)^{3/4}}\left|\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})\left(\hat{a}_{x,i}^\top\xi\right)^3\right| + \frac{8}{3}\frac{r^3}{(nd)^{3/4}}\left|\sum_{i,j=1}^n\sigma_{x,i,j}^2\left(\frac{1}{2}(\hat{a}_{x,i}+\hat{a}_{x,j})^\top\xi\right)^3\right| \\
&\quad + 50\frac{r^4}{nd}\sum_{i=1}^n(\sigma_{x,i}+\beta_{\mathrm{V}})(\hat{a}_{x,i}^\top\xi)^4, \quad (54)
\end{aligned}$$

where the last step follows from the fact that $0 \le \sigma_{x,i} \le \sigma_{x,i} + \beta_{\mathrm{V}}$. In order to show that $\left|\|z-x\|_z^2 - \|z-x\|_x^2\right|$ is bounded as $\mathcal{O}\left(1/\sqrt{nd}\right)$ with high probability, it suffices to show that with high probability, the third and fourth degree polynomials of $\hat{a}_{x,i}^\top\xi$, that appear in bound (54), are bounded by $\mathcal{O}\left((nd)^{1/4}\right)$ and $\mathcal{O}\left(\sqrt{nd}\right)$ respectively.

Applying the bounds (40b), (40c) and (40d) from Lemma 15, we have with probability at least $1-\epsilon$,

$$\|z-x\|_z^2 - \|z-x\|_x^2 \le \frac{r^3}{\sqrt{nd}}\left(\frac{22\sqrt{15}\chi_3}{3}\right) + \frac{r^4}{\sqrt{nd}}\left(50\sqrt{105}\chi_4\right).$$

Using the condition

$$r \le \min\left\{\frac{\epsilon}{22\sqrt{5/3}\chi_3}, \sqrt{\frac{\epsilon}{50\sqrt{105}\chi_4}}\right\}, \quad (55)$$

completes our proof of claim (32b).

C.4 Proof of Lemma 14

The proof is based on Lemma 12 and a simple application of the standard chi-square tail bounds. According to Lemma 12, we have that for $v \in \overline{xz}$,

$$\left|1 - \frac{s_{v,i}}{s_{x,i}}\right| \leq \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - v\|_x \leq \left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - z\|_x .$$

According to equation (41), the proposal follows Gaussian distribution

$$\left(\frac{n}{d}\right)^{\frac{1}{4}} \|x - z\|_x = \frac{r}{d^{1/2}} \|\xi\|_2 ,$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. Using the standard chi-square tail bound we have that for $\delta > 0$,

$$\mathbb{P}\left[\|\xi\|_2 / \sqrt{d} \geq 1 + \delta\right] \leq \exp\left(-d\delta^2/2\right).$$

Plugging in $\delta = \sqrt{\frac{2}{d}} \log^{\frac{1}{2}}\left(\frac{4}{\epsilon}\right)$ concludes the lemma.

C.5 Proof of Lemma 15

The proof relies on the classical fact that the tails of a polynomial in Gaussian random variables decay exponentially independently of dimension. In particular, Theorem 6.7 by Janson (1997) ensures that for any integers $d, k \geq 1$, any polynomial $f : \mathbb{R}^d \to \mathbb{R}$ of degree k, and any scalar $t \geq (2e)^{k/2}$, we have

$$\mathbb{P}\left[|f(\xi)| \geq t\left(\mathbb{E}f(\xi)^2\right)^{\frac{1}{2}}\right] \leq \exp\left(-\frac{k}{2e}t^{2/k}\right), \tag{56}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_n)$ denotes a standard Gaussian vector in n dimensions. Also, the following observations on the behavior of the vectors $\hat{a}_{x,i}$ defined in equation (39) are useful:

$$\|\hat{a}_{x,i}\|_2^2 = \theta_{x,i} \overset{(i)}{\leq} \sqrt{\frac{n}{d}} \quad \text{for all } i \in [n], \quad \text{and} \tag{57a}$$

$$(\hat{a}_{x,i}^\top \hat{a}_{x,j})^2 = \theta_{x,i,j}^2 \quad \text{for all } i, j \in [n], \tag{57b}$$

where inequality (i) follows from Lemma 7 (c).

C.5.1 Proof of bound (40a)

We have

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\left(\hat{a}_{x,i}^\top \xi\right)^2\right)^2 &= \sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,j} + \beta_{\mathrm{V}})\,\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^2 \left(\hat{a}_{x,j}^\top \xi\right)^2 \\
&= \sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,j} + \beta_{\mathrm{V}})\left(\|\hat{a}_{x,i}\|_2^2 \|\hat{a}_{x,j}\|_2^2 + 2\left(\hat{a}_{x,i}^\top \hat{a}_{x,j}\right)^2\right) \\
&= \sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,j} + \beta_{\mathrm{V}})\left(\theta_{x,i}\theta_{x,j} + 2\theta_{x,i,j}^2\right) \\
&\overset{(i)}{=} d^2 + 2d \\
&\leq 3d^2,
\end{aligned}$$

where step (i) follows from properties (c) and (d) from Lemma 11. Applying the bound (56) with $k = 2, t = e\log(\frac{4}{\epsilon})$ yields the claim. We verify that for $\epsilon \in (0, 1/15]$, $t \geq 2e$.

C.5.2 Proof of bound (40b)

Using Isserlis' theorem (Isserlis, 1918) for Gaussian moments, we obtain

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\left(\hat{a}_{x,i}^\top \xi\right)^3\right)^2 &= \sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,i} + \beta_{\mathrm{V}})\,\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^3 \left(\hat{a}_{x,j}^\top \xi\right)^3 \\
&= 9\underbrace{\sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,j} + \beta_{\mathrm{V}})\|\hat{a}_{x,i}\|_2^2 \|\hat{a}_{x,j}\|_2^2 \left(\hat{a}_{x,i}^\top \hat{a}_{x,j}\right)}_{=:N_1} \\
&\quad + 6\underbrace{\sum_{i,j=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})(\sigma_{x,j} + \beta_{\mathrm{V}})\left(\hat{a}_{x,i}^\top \hat{a}_{x,j}\right)^3}_{=:N_2}.
\end{aligned} \tag{58}$$

We claim that the two terms in this sum are bounded as $N_1 \leq \sqrt{nd}$ and $N_2 \leq \sqrt{nd}$. Assuming the claims as given, we now complete the proof. Plugging in the bounds for N_1 and N_2 in equation (58) we find that $\mathbb{E}\big(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\left(\hat{a}_{x,i}^\top \xi\right)^3\big)^2 \leq 15\sqrt{nd}$. Applying the bound (56) with $k = 3, t = \left(\frac{2e}{3}\log(4/\epsilon)\right)^{3/2}$ yields the claim. We also verify that for $\epsilon \in (0, 1/15]$, $t \geq (2e)^{3/2}$. We now turn to proving the bounds on N_1 and N_2.

Bounding N_1: Let B be an $n \times d$ matrix with its i-th row given by $\sqrt{(\sigma_{x,i} + \beta_{\mathrm{V}})}\hat{a}_{x,i}^\top$. Observe that

$$\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_i \hat{a}_{x,i}^\top = V_x^{-1/2}\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\frac{a_i a_i^\top}{s_{x,i}^2}\right)V_x^{-1/2} = V_x^{-1/2} V_x V_x^{-1/2} = \mathbb{I}_d. \tag{59}$$

Thus we have $B^\top B = \mathbb{I}_d$, which implies that $BB^\top$ is an orthogonal projection matrix. Letting $v \in \mathbb{R}^n$ be a vector such that $v_i = \sqrt{(\sigma_{x,i} + \beta_{\mathrm{V}})}\,\|\hat{a}_{x,i}\|_2^2$, we then have

$$\sum_{i,j=1}^n (\sigma_{x,i}+\beta_{\mathrm{V}})\|\hat{a}_{x,i}\|_2^2\,\hat{a}_{x,i}^\top\,(\sigma_{x,j}+\beta_{\mathrm{V}})\|\hat{a}_{x,j}\|_2^2\,\hat{a}_{x,j} = \left\|\sum_{i=1}^n (\sigma_{x,i}+\beta_{\mathrm{V}})\|\hat{a}_{x,i}\|_2^2\,\hat{a}_{x,i}\right\|_2^2 = \left\|B^\top v\right\|_2^2 \overset{(i)}{\leq} \|v\|_2^2,$$

where inequality (i) follows from the fact that $v^\top P v \leq \|v\|_2^2$ for any orthogonal projection matrix P. Equation (57a) implies that $v_i^2 = (\sigma_{x,i} + \beta_{\mathrm{V}})\,\theta_{x,i}^2$. Using Lemma 11(e), we find that

$$\|v\|_2^2 = \sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\theta_{x,i}^2 \leq \sqrt{nd}.$$

Bounding N_2: We see that

$$\sum_{i,j=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}})(\sigma_{x,j}+\beta_{\mathrm{V}})\left(\hat{a}_{x,i}^{\top}\hat{a}_{x,j}\right)^3 \overset{\text{(C–S)}}{\leq} \sum_{i,j=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}})(\sigma_{x,j}+\beta_{\mathrm{V}})\left(\hat{a}_{x,i}^{\top}\hat{a}_{x,j}\right)^2 \|\hat{a}_{x,i}\|_2 \|\hat{a}_{x,j}\|_2$$
$$\overset{\text{(eqns.(57a),(57b))}}{\leq} \sum_{i,j=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}})(\sigma_{x,j}+\beta_{\mathrm{V}})\,\theta_{x,i,j}^2\sqrt{\theta_{x,i}\theta_{x,j}}$$
$$\overset{\text{(Lem. 7(c))}}{\leq} \sqrt{\frac{n}{d}} \sum_{i,j=1}^{n} (\sigma_{x,i}+\beta_{\mathrm{V}})(\sigma_{x,j}+\beta_{\mathrm{V}})\,\theta_{x,i,j}^2.$$

We now apply Lemma 11(d) followed by Lemma 11(c) to obtain the claimed bound on N_2.

C.5.3 Proof of bound (40c)

Let $c_{i,j} = \dfrac{(\hat{a}_{x,i}+\hat{a}_{x,j})}{2}$ for $i,j \in [n]$. Using Isserlis' theorem for Gaussian moments, we obtain

$$\mathbb{E}\left(\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \left(c_{i,j}^{\top}\xi\right)^3\right)^2 = \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \mathbb{E}\left(c_{i,j}^{\top}\xi\right)^3\left(c_{k,l}^{\top}\xi\right)^3$$
$$= 9\underbrace{\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \|c_{i,j}\|_2^2 \|c_{k,l}\|_2^2 \left(c_{i,j}^{\top}c_{k,l}\right)}_{=:C_1} + 6\underbrace{\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(c_{i,j}^{\top}c_{k,l}\right)^3}_{=:C_2}$$

We claim that $C_1 \leq \sqrt{nd}$ and $C_2 \leq \sqrt{nd}$. Assuming the claims as given, the result follows using similar arguments as in the previous part. We now bound $C_i, i = 1,2$, using arguments similar to the ones used in Section C.5.2 to bound $N_i, i = 1,2$, respectively. The following bounds on $\|c_{i,j}\|_2^2$ are used in the arguments that follow:

$$\|c_{i,j}\|_2^2 \overset{\text{SSI}}{\leq} \frac{1}{2}\left(\|\hat{a}_i\|_2^2 + \|\hat{a}_j\|_2^2\right) = \frac{1}{2}(\theta_{x,i}+\theta_{x,j}) \tag{60a}$$
$$\overset{\text{Lem. 7(c)}}{\leq} \sqrt{\frac{n}{d}}. \tag{60b}$$

Bounding C_1: Let B be the same $n \times d$ matrix as in the proof of previous part with its i-th row given by $\sqrt{(\sigma_{x,i}+\beta_{\mathrm{V}})}\hat{a}_{x,i}^{\top}$. Define the vector $u \in \mathbb{R}^d$ with entries given by

$u_i = \sum_{j=1}^n \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2/(\sigma_{x,i}+\beta_{\mathrm{V}})^{1/2}$. We have

$$\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \|c_{i,j}\|_2^2 \|c_{k,l}\|_2^2 \left(c_{i,j}^{\top}c_{k,l}\right) \leq \left\|\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 c_{i,j}\right\|_2^2$$
$$\overset{\text{(SSI)}}{\leq} \frac{1}{2}\left(\left\|\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 \hat{a}_{x,i}\right\|_2^2 + \left\|\sum_{i,j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2 \hat{a}_{x,j}\right\|_2^2\right)$$
$$= \left\|B^{\top}u\right\|_2^2$$
$$\overset{(i)}{\leq} \|u\|_2^2,$$

where inequality (i) follows from the fact that $v^{\top}Pv \leq \|v\|_2^2$ for any orthogonal projection matrix P. It is left to bound the term u_i^2. We see that

$$u_i^2 = \frac{1}{\sigma_{x,i}+\beta_{\mathrm{V}}} \sum_{j,k=1}^{n} \sigma_{x,i,j}^2\sigma_{x,i,k}^2 \|c_{i,j}\|_2^2 \|c_{i,k}\|_2^2 \overset{\text{(bnd. (60b))}}{\leq} \sqrt{\frac{n}{d}} \frac{1}{\sigma_{x,i}+\beta_{\mathrm{V}}} \sum_{j,k=1}^{n} \sigma_{x,i,j}^2\sigma_{x,i,k}^2 \|c_{i,j}\|_2^2$$
$$\overset{\text{(Lem. 11(a))}}{\leq} \sqrt{\frac{n}{d}} \frac{\sigma_{x,i}}{\sigma_{x,i}+\beta_{\mathrm{V}}} \sum_{j=1}^{n} \sigma_{x,i,j}^2 \|c_{i,j}\|_2^2$$
$$\overset{\text{(bnd. (60a))}}{\leq} \sqrt{\frac{n}{d}} \sum_{j=1}^{n} \sigma_{x,i,j}^2 \frac{\theta_{x,i}+\theta_{x,j}}{2}.$$

Now, summing over i and using symmetry of indices i,j, we find that

$$\|u\|_2^2 \leq \sqrt{\frac{n}{d}} \sum_{i=1}^{n}\sum_{j=1}^{n} \sigma_{x,i,j}^2\theta_{x,i} \overset{\text{(Lem. 11(a))}}{=} \sqrt{\frac{n}{d}} \sum_{i=1}^{n} \sigma_{x,i}\theta_{x,i} \overset{\text{(Lem. 11(c))}}{\leq} \sqrt{nd},$$

thereby implying that $C_1 \leq \sqrt{nd}$.

Bounding C_2: Using the Cauchy-Schwarz inequality and the bound (60b), we find that

$$\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(c_{i,j}^{\top}c_{k,l}\right)^3 \leq \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(c_{i,j}^{\top}c_{k,l}\right)^2 \|c_{i,j}\|_2 \|c_{k,l}\|_2$$
$$\leq \sqrt{\frac{n}{d}} \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(c_{i,j}^{\top}c_{k,l}\right)^2.$$

Using SSI and the symmetry of pairs of indices (i,j) and (k,l), we obtain

$$\sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(c_{i,j}^{\top}c_{k,l}\right)^2 \leq \sum_{i,j,k,l=1}^{n} \sigma_{x,i,j}^2\sigma_{x,k,l}^2 \left(\hat{a}_{x,i}^{\top}\hat{a}_k\right)^2 = \sum_{i,k=1}^{n} \sigma_{x,i}\sigma_{x,k}\left(\hat{a}_{x,i}^{\top}\hat{a}_k\right)^2.$$

The resulting expression can be bounded as follows:

$$\sum_{i,k=1}^{n} \sigma_{x,i}\sigma_{x,k}\left(\hat{a}_{x,i}^{\top}\hat{a}_k\right)^2 \overset{\text{(eqn.(57b))}}{=} \sum_{i,k=1}^{n} \sigma_{x,i}\sigma_{x,k}\theta_{x,i,k}^2 \overset{\text{(Lem. 11(d))}}{\leq} \sum_{i=1}^{n} \sigma_{x,i}\theta_{x,i} \overset{\text{(Lem. 11(c))}}{\leq} n.$$

Putting the pieces together yields the claimed bound on C_2.

C.5.4 Proof of bound (40d)

Observe that $\hat{a}_{x,i}^\top \xi \sim \mathcal{N}(0, \theta_{x,i})$ and hence $\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^8 = 105\,\theta_{x,i}^4$. Thus we have

$$\begin{aligned}
\mathbb{E}\left(\sum_{i=1}^n \sigma_{x,i}\left(\hat{a}_{x,i}^\top \xi\right)^4\right)^2 &\overset{\text{C–S}}{\leq} \sum_{i,j=1}^n \sigma_{x,i}\sigma_{x,j}\left(\mathbb{E}\left(\hat{a}_{x,i}^\top \xi\right)^8\right)^{\frac{1}{2}}\left(\mathbb{E}\left(\hat{a}_{x,j}^\top \xi\right)^8\right)^{\frac{1}{2}} \\
&= 105\sum_{i,j=1}^n \sigma_{x,i}\sigma_{x,j}\theta_{x,i}^2\theta_{x,j}^2 \\
&= 105\left(\sum_{i=1}^n \sigma_{x,i}\theta_{x,i}^2\right)^2 \\
&\overset{\text{(Lem. 11}(e)\text{)}}{\leq} 105nd.
\end{aligned}$$

Applying the bound (56) with $k = 4, t = \left(\frac{e}{2}\log(4/\epsilon)\right)^2$ yields the result. We also verify that for $\epsilon \in (0, 1/15]$, we have $t \geq (2e)^2$

C.6 Proof of Lemma 13

We now derive the different expressions for derivatives and prove the bounds for Hessians of $x \mapsto \varphi_{x,i}$, $i \in [n]$ and $x \mapsto \Psi_x$. In this section we use the simpler notation $H_x := \nabla^2 \mathcal{F}_x$.

C.6.1 Gradient of σ

Using $s_{x+h,i} = (b_i - a_i^\top(x+h)) = s_{x,i} - a_i^\top h$, we define the Hessian difference matrix

$$\Delta_{x,h}^H := H_{x+h} - H_x = \sum_{i=1}^n a_i a_i^\top \left(\frac{1}{(s_{x,i} - a_i^\top h)^2} - \frac{1}{s_{x,i}^2}\right). \tag{61}$$

Up to second order terms, we have

$$\frac{1}{s_{x+h,i}^2} = \frac{1}{s_{x,i}^2}\left[1 + \frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + \mathcal{O}\left(\|h\|_2^3\right), \tag{62a}$$

$$\Delta_{x,h}^H = \sum_{i=1}^n \frac{a_i a_i^\top}{s_{x,i}^2}\left[\frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + \mathcal{O}\left(\|h\|_2^3\right), \tag{62b}$$

$$a_i^T H_{x+h}^{-1} a_i = a_i^\top H_x^{-1} a_i - a_i^\top H_x^{-1}\Delta_{x,h}^H H_x^{-1} a_i + a_i^\top H_x^{-1}\Delta_{x,h}^H H_x^{-1}\Delta_{x,h}^H H_x^{-1} a_i + \mathcal{O}\left(\|h\|_2^3\right). \tag{62c}$$

Collecting different first order terms in $\sigma_{x+h,i} - \sigma_{x,i}$, we obtain

$$\begin{aligned}
\sigma_{x+h,i} - \sigma_{x,i} &= 2\,\frac{a_i^\top H_x^{-1} a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}} - 2\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right) H_x^{-1} a_i}{s_{x,i}^2} + \mathcal{O}\left(\|h\|_2^2\right) \\
&= 2\left[\sigma_{x,i}\frac{a_i^\top h}{s_{x,i}} - \sum_{j=1}^n \sigma_{x,i,j}^2\,\frac{a_j^\top h}{s_{x,j}}\right] + \mathcal{O}\left(\|h\|_2^2\right) \\
&= 2\left[(\Sigma_x - \Upsilon_x^{(2)})S_x^{-1}A\right]_i h + \mathcal{O}\left(\|h\|_2^2\right).
\end{aligned}$$

Dividing both sides by h and letting $h \to 0$ yields the claim.

C.6.2 Gradient of φ

Using the chain rule and the fact that $\nabla s_{x,i} = -a_i$, we find that

$$\begin{aligned}
\nabla\varphi_{x,i} &= \frac{\nabla\sigma_{x,i}}{s_{x,i}^2} - 2\,(\sigma_{x,i} + \beta_{\mathrm{V}})\frac{\nabla s_{x,i}}{s_{x,i}^3} \\
&= \frac{2}{s_{x,i}^2} A^\top S_x^{-1}\left[2\Sigma_x + \beta_{\mathrm{V}}\,\mathbb{I} - \Upsilon_x^{(2)}\right] e_i,
\end{aligned}$$

as claimed.

C.6.3 Gradient of Ψ

For convenience, let us restate equations (39) and (59):

$$\hat{a}_{x,i} = \frac{1}{s_{x,i}} V_x^{-1/2} a_i, \quad \text{and} \quad \sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top = \mathbb{I}_d.$$

For a unit vector h, we have

$$h^\top \nabla \log\det V_x = \lim_{\delta\to 0}\frac{1}{\delta}\left[\operatorname{trace}\log\left(\sum_{i=1}^n \frac{(\sigma_{x+\delta h,i} + \beta_{\mathrm{V}})}{\left(1 - \delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right) - \operatorname{trace}\log\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right]. \tag{63}$$

Let $\log L$ denote the logarithm of the matrix L. Keeping track of the first order terms on RHS of equation (63), we find that

$$\begin{aligned}
&\operatorname{trace}\left[\log\left(\sum_{i=1}^n (\sigma_{x+\delta h,i} + \beta_{\mathrm{V}})\,\frac{\hat{a}_{x,i}\hat{a}_{x,i}^\top}{\left(1 - \delta a_i^\top h/s_{x,i}\right)^2}\right)\right] - \operatorname{trace}\left[\log\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right] \\
&= \operatorname{trace}\left[\log\left(\sum_{i=1}^n \left(\sigma_{x+\delta h,i} + \beta_{\mathrm{V}} + \delta h^\top\nabla\sigma_{x,i}\right)\left(1 + 2\delta\frac{a_i^\top h}{s_{x,i}^2}\right)\right)\right] - \operatorname{trace}\left[\log\left(\sum_{i=1}^n (\sigma_{x,i} + \beta_{\mathrm{V}})\,\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right] + \mathcal{O}\left(\delta\right. \\
&= \operatorname{trace}\left[\sum_{i=1}^n \delta\left(2\,(\sigma_{x,i} + \beta_{\mathrm{V}})\,\frac{a_i^\top h}{s_{x,i}^2} + h^\top\nabla\sigma_{x,i}\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right] + \mathcal{O}\left(\delta^2\right) \\
&= \delta\left(\sum_{i=1}^n \left(2\,(\sigma_{x,i} + \beta_{\mathrm{V}})\,\frac{a_i^\top h}{s_{x,i}^2} + h^\top\nabla\sigma_{x,i}\right)\theta_i\right) + \mathcal{O}\left(\delta^2\right),
\end{aligned}$$

where we have used the fact $\operatorname{trace}(\log \mathbb{I}) = 0$. Letting $\delta \to 0$ and substituting expression of $h^\top \nabla \sigma_x$ from part (a), we obtain

$$h^\top \nabla \log\det V_x = A_x^\top \left(4\Sigma_x + 2\beta_{\mathrm{V}}\mathbb{I} - 2\Upsilon_x^{(2)}\right)\Theta_x h.$$

C.6.4 Bound on Hessian $\nabla^2\varphi$

In terms of the shorthand $E_{ii} = e_i e_i^\top$, we claim that for any $h \in \mathbb{R}^d$,

$$h^\top \nabla^2 \varphi_{x,i} h = \frac{2}{s_{x,i}^2} h^\top A_x^\top \Big[E_{ii}\left(3\left(\Sigma_x + \beta_{\mathrm{V}}\mathbb{I}\right) + 7\Sigma_x - 8\operatorname{diag}(\Upsilon_x^{(2)} e_i)\right) E_{ii} + \operatorname{diag}(\Upsilon_x e_i)(4\Upsilon_x - 3\mathbb{I})\operatorname{diag}(\Upsilon_x e_i)\Big] A_x h. \tag{64}$$

Note that

$$\varphi_{x+h,i} - \varphi_{x,i} = \underbrace{\left(\frac{a_i^\top H_{x+h,i}^{-1} a_i}{s_{x+h,i}^4} - \frac{a_i^\top H_{x,i}^{-1} a_i}{s_{x,i}^4}\right)}_{=:A_1} + \underbrace{\beta_{\mathrm{V}}\left(\frac{1}{s_{x+h,i}^2} - \frac{1}{s_{x,i}^2}\right)}_{=:A_2}. \tag{65}$$

The second order Taylor expansion of $1/s_{x,i}^4$ is given by

$$\frac{1}{s_{x+h,i}^4} = \frac{1}{s_{x,i}^4}\left[1 + \frac{4a_i^\top h}{s_{x,i}} + \frac{10(a_i^\top h)^2}{s_{x,i}^2}\right] + \mathcal{O}\left(\|h\|_2^3\right).$$

Let B_1 and B_2 denote the second order terms, i.e., the terms that are of order $\mathcal{O}\left(\|h\|_2^2\right)$, in Taylor expansion of A_1 and A_2 around x, respectively. Borrowing terms from equations (62a)-(62c) and simplifying we obtain

$$B_1 = 10\sigma_{x,i}\frac{(a_i^\top h)^2}{s_{x,i}^2} - 8\frac{a_i^\top h}{s_{x,i}}\sum_{j=1}^n \frac{\sigma_{x,i,j}^2}{s_{x,i}^2}\frac{a_j^\top h}{s_{x,j}} - 3\sum_{j=1}^n \frac{\sigma_{x,i,j}^2}{s_{x,i}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2} + 4\sum_{j=1}^n\sum_{l=1}^n \frac{\sigma_{x,i,j}}{s_{x,i}}\sigma_{x,j,l}\frac{\sigma_{x,l,i}}{s_{x,i}}\frac{a_j^\top h}{s_{x,j}}\frac{a_l^\top h}{s_{x,l}},$$

and $B_2 = 3\beta_{\mathrm{V}}\dfrac{(a_i^\top h)^2}{s_{x,i}^2}$.

Observing that the second order term in the Taylor expansion of $\varphi_{x+h,i}$ around x, is exactly $\frac{1}{2}h^\top\nabla^2\varphi_{x,i}h$ yields the claim (64). We now turn to prove the bound on the directional

Hessian. Recall $\eta_{x,i} = a_i^\top h / s_{x,i}$. We have

$$\begin{aligned}
&s_{y,i}^2\left|\frac{1}{2}h^\top\nabla^2\varphi_{x,i}h\right| \\
&= \left|3\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 7\sigma_{x,i}\eta_{x,i}^2 - 8\sum_{j=1}^n \sigma_{x,i,j}^2\eta_{x,j}\eta_{x,i} - 3\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2 + 4\sum_{j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}\eta_{x,j}\eta_{x,k}\right| \\
&\overset{(i)}{\le} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 8\sum_{j=1}^n \sigma_{x,i,j}^2\left|\eta_{x,i}\eta_{x,j}\right| + 7\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2 \\
&\overset{(ii)}{\le} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 4\sum_{j=1}^n \sigma_{x,i,j}^2\left(\eta_{x,i}^2+\eta_{x,j}^2\right) + 7\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2 \\
&\overset{(iii)}{\le} 10\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 4\sum_{j=1}^n \sigma_{x,i}\eta_{x,i}^2 + 4\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2 + 7\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2, \\
&\overset{(iv)}{\le} 14\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2 + 11\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,j}^2,
\end{aligned}$$

where step (i) follows from the fact that $\operatorname{diag}(\Upsilon_y e_i)\Upsilon_y\operatorname{diag}(\Upsilon_y e_i) \preceq \operatorname{diag}(\Upsilon_y e_i)\operatorname{diag}(\Upsilon_y e_i)$ since Υ_y is an orthogonal projection matrix; step (ii) follows from AM-GM inequality; step (iii) follows from the symmetry of indices i and j and Lemma 11(a), and step (iv) from the fact that $\sigma_{x,i} \le \sigma_{x,i} + \beta_{\mathrm{V}}$.

C.6.5 Bound on Hessian $\nabla^2\Psi$

We have

$$\begin{aligned}
\frac{1}{2}h^\top\left(\nabla^2\log\det V_x\right)h = &\frac{1}{2}\lim_{\delta\to 0}\frac{1}{\delta^2}\Bigg[\operatorname{trace}\log\left(\sum_{i=1}^n \frac{\left(\sigma_{x+\delta h,i}+\beta_{\mathrm{V}}\right)}{\left(1-\delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right) \\
&+ \operatorname{trace}\log\left(\sum_{i=1}^n \frac{\left(\sigma_{x-\delta h,i}+\beta_{\mathrm{V}}\right)}{\left(1+\delta a_i^\top h/s_{x,i}\right)^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right) \\
&- 2\operatorname{trace}\log\left(\sum_{i=1}^n\left(\sigma_x+\beta_{\mathrm{V}}\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\Bigg].
\end{aligned} \tag{66}$$

Up to second order terms, we have

$$\begin{aligned}
&\text{trace}\left[\log\left(\sum_{i=1}^n (\sigma_{x+\delta h,i}+\beta_{\mathrm{V}})\,\frac{\hat{a}_{x,i}\hat{a}_{x,i}^\top}{\left(1-\delta a_i^\top h/s_{x,i}\right)^2}\right)\right]\\
&=\text{trace}\left[\log\left(\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2 h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)\right]\\
&=\text{trace}\left[\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2 h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right]\\
&\quad-\text{trace}\left[\frac{1}{2}\left(\sum_{i=1}^n\left(\sigma_{x,i}+\beta_{\mathrm{V}}+\delta h^\top\nabla\sigma_{x,i}+\frac{1}{2}\delta^2 h^\top\nabla^2\sigma_{x,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{x,i}}+3\delta^2\left(\frac{a_i^\top h}{s_{x,i}}\right)^2\right)\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)^2\right].
\end{aligned}$$

We can similarly obtain the second order expansion of the term $\text{trace}\log\left(\sum_{i=1}^n\frac{(\sigma_{x-\delta h,i}+\beta_{\mathrm{V}})}{(1+\delta a_i^\top h/s_{x,i})^2}\hat{a}_{x,i}\hat{a}_{x,i}^\top\right)$.

Recall $\eta_{x,i}=\frac{a_i^\top h}{s_{x,i}}$. Using part (a) to substitute $h^\top\nabla\sigma_{x,i}$, we obtain

$$\begin{aligned}
\frac{1}{2}h^\top\left(\nabla^2\log\det V_x\right)h=&\sum_{i=1}^n\left(3\left(\sigma_{x,i}+\beta_{\mathrm{V}}\right)\eta_{x,i}^2+4\left(\sigma_{x,i}\eta_{x,i}^2-\sum_{j=1}^n\sigma_{x,i,j}^2\eta_{x,i}\eta_{x,j}\right)+\frac{1}{2}h^\top\nabla^2\sigma_{x,i}h\right)\theta_i\\
&-2\Bigg[\sum_{i,j=1}^n\left(2\sigma_{x,i}+\beta_{\mathrm{V}}\right)\left(2\sigma_{x,j}+\beta_{\mathrm{V}}\right)\eta_{x,i}\eta_{x,j}\theta_{x,i,j}^2-2\sum_{i,j,k=1}^n\left(2\sigma_{x,i}+\beta_{\mathrm{V}}\right)\sigma_{x,j,k}^2\theta_{x,i,k}^2\eta_{x,i}\eta_{x,j}\\
&+\sum_{i,j,k,l=1}^n\sigma_{x,i,l}^2\sigma_{x,j,k}^2\theta_{x,k,l}^2\eta_{x,i}\eta_{x,j}\Bigg].
\end{aligned}\tag{67}$$

We claim that the directional Hessian $h^\top\nabla^2\sigma_{x,i}h$ is given by

$$h^\top\nabla^2\sigma_{x,i}h=2\,h^\top A_x^\top\left[E_{ii}(3\Sigma_x-4\,\text{diag}(\Upsilon_x^{(2)}e_i))E_{ii}+\text{diag}(\Upsilon_x e_i)(4\Upsilon_x-3\mathbb{I})\,\text{diag}(\Upsilon_x e_i)\right]A_x h.\tag{68}$$

Assuming the claim at the moment we now bound $\left|h^\top\nabla^2\Psi_x h\right|$. To shorten the notation, we drop the x-dependence of the terms $\sigma_{x,i},\sigma_{x,i,j},\theta_{x,i}$ and $\eta_{x,i}$. Since Υ_x is an orthogonal projection matrix, we have

$$\text{diag}(\Upsilon_x e_i)\Upsilon_x\,\text{diag}(\Upsilon_x e_i)\preceq\text{diag}(\Upsilon_x e_i)\,\text{diag}(\Upsilon_x e_i).$$

Using this fact and substituting the expression for $h^\top\nabla^2\sigma_{x,i}h$ from equation (68) in equation (67), we obtain

$$\begin{aligned}
&\left|h^\top\nabla^2\Psi_x h\right|\\
&\le\sum_{i=1}^n\Bigg[3\Big(\sigma_i+\beta_{\mathrm{V}}\Big)\eta_i^2+4\big(\sigma_i\eta_i^2+\sum_{j=1}^n\sigma_{i,j}^2\eta_i\eta_j\big)+3\sigma_i\eta_i^2+4\sum_{j=1}^n\sigma_{i,j}^2\eta_i\eta_j+7\sum_{j=1}^n\sigma_{i,j}^2\eta_j^2\Bigg]\theta_i\\
&\quad+\Bigg[8\sum_{i,j=1}^n\left(\sigma_i+\beta_{\mathrm{V}}\right)\left(\sigma_j+\beta_{\mathrm{V}}\right)\eta_i\eta_j\theta_{i,j}^2+8\sum_{i,j,k=1}^n\left(\sigma_i+\beta_{\mathrm{V}}\right)\sigma_{j,k}^2\theta_{i,k}^2\eta_i\eta_j+2\sum_{i,j,k,l=1}^n\sigma_{i,l}^2\sigma_{j,k}^2\theta_{k,l}^2\eta_i\eta_j\Bigg].
\end{aligned}$$

Rearranging terms, we find that

$$\begin{aligned}
&\left|h^\top\nabla^2\Psi_x h\right|\\
&\le\sum_{i=1}^n\left[10\left(\sigma_i+\beta_{\mathrm{V}}\right)\eta_i^2+8\sum_{j=1}^n\sigma_{i,j}^2\eta_i\eta_j+7\sum_{j=1}^n\sigma_{i,j}^2\eta_j^2\right]\theta_i\\
&\quad+\left[8\sum_{i,j=1}^n\left(\sigma_i+\beta_{\mathrm{V}}\right)\left(\sigma_j+\beta_{\mathrm{V}}\right)\eta_i\eta_j\theta_{i,j}^2+8\sum_{i,j,k=1}^n\left(\sigma_i+\beta_{\mathrm{V}}\right)\sigma_{j,k}^2\theta_{i,k}^2\eta_i\eta_j+2\sum_{i,j,k,l=1}^n\sigma_{i,l}^2\sigma_{j,k}^2\theta_{k,l}^2\eta_i\eta_j\right]\\
&\overset{(i)}{\le}\sum_{i=1}^n\left[10\left(\sigma_i+\beta_{\mathrm{V}}\right)\eta_i^2+4\sum_{j=1}^n\sigma_{i,j}^2\left(\eta_i^2+\eta_j^2\right)+7\sum_{j=1}^n\sigma_{i,j}^2\eta_j^2\right]\theta_i\\
&\quad+\left[4\sum_{i,j=1}^n\big(\sigma_i+\beta_{\mathrm{V}}\big)\big(\sigma_j+\beta_{\mathrm{V}}\big)\theta_{i,j}^2(\eta_i^2+\eta_j^2)+4\sum_{i,j,k=1}^n\big(\sigma_i+\beta_{\mathrm{V}}\big)\sigma_{j,k}^2\theta_{i,k}^2(\eta_i^2+\eta_j^2)+\sum_{i,j,k,l=1}^n\sigma_{i,l}^2\sigma_{j,k}^2\theta_{k,l}^2(\eta_i^2+\eta_j^2)\right]
\end{aligned}$$

where in step (i) we have used the AM-GM inequality. Simplifying further, we obtain

$$\begin{aligned}
\left|h^\top\nabla^2\Psi_y h\right|&\le\sum_{i=1}^n\left[14\left(\sigma_i+\beta_{\mathrm{V}}\right)\eta_i^2+11\sum_{j=1}^n\sigma_{i,j}^2\eta_j^2\right]\theta_i+\left[\sum_{i=1}^n12\left(\sigma_i+\beta_{\mathrm{V}}\right)\theta_i\eta_i^2+\sum_{i,j=1}^n6\sigma_{i,j}^2\theta_i\eta_j^2\right]\\
&=26\sum_{i=1}^n\left(\sigma_i+\beta_{\mathrm{V}}\right)\theta_i\eta_i^2+17\sum_{i,j=1}^n\sigma_{i,j}^2\theta_i\eta_j^2.
\end{aligned}$$

Dividing both sides by two completes the proof.

Proof of claim (68)**:** In order to compute the directional Hessian of $x\mapsto\sigma_{x,i}$, we need to track the second order terms in equations (62a)-(62c). Collecting the second order terms (denoted by $\sigma_h^{(2)}$) in the expansion of $\sigma_{x+h,i}-\sigma_{x,i}$, we obtain

$$\begin{aligned}
\sigma_h^{(2)}=&\,3\,\frac{a_i^\top H_x^{-1}a_i}{s_{x,i}^2}\frac{(a_i^\top h)^2}{s_{x,i}^2}-4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n\frac{a_ja_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right)H_x^{-1}a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}}\\
&-3\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n\frac{a_ja_j^\top}{s_{x,j}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2}\right)H_x^{-1}a_i}{s_{x,i}^2}\\
&+4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n\frac{a_ja_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right)H_x^{-1}\left(\sum_{l=1}^n\frac{a_la_l^\top}{s_{x,l}^2}\frac{a_l^\top h}{s_{x,l}}\right)a_i}{s_{x,i}^2}.
\end{aligned}$$

We simply each term on the RHS one by one. Simplifying the first term, we obtain

$$3\,\frac{a_i^\top H_x^{-1}a_i}{s_{x,i}^2}\frac{(a_i^\top h)^2}{s_{x,i}^2}=3\,\sigma_{x,i}\eta_{x,i}^2=h^\top 3\,A_x^\top E_{ii}\Sigma_x E_{ii}A_x\,h.$$

For the second term, we have

$$4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right)H_x^{-1}a_i}{s_{x,i}^2}\frac{a_i^\top h}{s_{x,i}} = 4\,\eta_{x,i}\sum_{j=1}^n \sigma_{x,i,j}^2\,\eta_{x,j}$$
$$= 4\,h^\top A_x^\top E_{ii}\,\mathrm{diag}\left(\Upsilon_x^{(2)}e_i\right)E_{ii}A_x h.$$

The third term can be simplified as follows:

$$3\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{(a_j^\top h)^2}{s_{x,j}^2}\right)H_x^{-1}a_i}{s_{x,i}^2} = 3\sum_{j=1}^n \sigma_{x,i,j}^2\eta_{x,j}^2$$
$$= 3\,h^\top A_x^\top\,\mathrm{diag}\left(\Upsilon_x e_i\right)\mathrm{diag}\left(\Upsilon_x e_i\right)A_x h$$

For the last term, we find that

$$4\,\frac{a_i^\top H_x^{-1}\left(\sum_{j=1}^n \frac{a_j a_j^\top}{s_{x,j}^2}\frac{a_j^\top h}{s_{x,j}}\right)H_x^{-1}\left(\sum_{l=1}^n \frac{a_l a_l^\top}{s_{x,l}^2}\frac{a_l^\top h}{s_{x,l}}\right)a_i}{s_{x,i}^2} = 4\sum_{j,l=1}^n \sigma_{x,i,j}\,\sigma_{x,j,l}\,\sigma_{x,l,i}\,\eta_{x,j}\,\eta_{x,l}$$
$$= 4\,h^\top A_x^\top\,\mathrm{diag}\left(\Upsilon_x e_i\right)\Upsilon_x\,\mathrm{diag}\left(\Upsilon_x e_i\right)A_x h.$$

Putting together the pieces yields the expression (68).

Appendix D. Analysis of the John walk

We recap the key ideas of the John walk for convenience. We have designed a new proposal distribution by making use of an *optimal set of weights* to define the new covariance structure for the Gaussian proposals, where optimality is defined with respect to the convex program defined below (69). The optimality condition is closely related to the problem of finding the largest ellipsoid at any interior point of the polytope, such that the ellipsoid is contained within the polytope. This problem of finding the largest ellipsoid was first studied by John (1948) who showed that each convex body in $\mathbb{R}^d$ contains a unique ellipsoid of maximal volume. More recently, Lee and Sidford (2014) make use of approximate John Ellipsoids to improve the convergence rate of interior point methods for linear programming. We refer the readers to their paper for more discussion about the use of John Ellipsoids for optimization problems. In this work, we make use of these ellipsoids for designing sampling algorithms with better theoretical bounds on the mixing times.

The vector $\zeta_x = (\zeta_{x,1},\ldots,\zeta_{x,n})^\top$ defined in the John walk's inverse covariance matrix (11) is computed by solving the following optimization problem:

$$\zeta_x = \arg\min_{w\in\mathbb{R}^n} c_x(w) := \sum_{i=1}^n w_i - \frac{1}{\alpha_{\text{J}}}\log\det\left(A^\top S_x^{-1}W^{\alpha_{\text{J}}}S_x^{-1}A\right) - \beta_{\text{J}}\sum_{i=1}^n \log w_i, \qquad (69)$$

where the parameters $\alpha_{\text{J}}, \beta_{\text{J}}$ are given by

$$\alpha_{\text{J}} = 1 - \frac{1}{\log_2(2n/d)} \quad \text{and} \quad \beta_{\text{J}} = \frac{d}{2n},$$

and W denotes an $n\times n$ diagonal matrix with $W_{ii} = w_i$ for each $i\in[n]$. In particular, for our proposal the inverse covariance matrix is proportional to J_x, where

$$J_x = \sum_{i=1}^n \zeta_{x,i}\frac{a_i a_i^\top}{(b_i - a_i^\top x)^2}. \qquad (70)$$

where $\kappa := \kappa_{n,d} = \log_2(2n/d) = (1-\alpha_{\text{J}})^{-1}$.

Recall that for John walk with parameter $\frac{r}{d^{3/4}\kappa^2}$, the proposals at state x are drawn from the multivariate Gaussian distribution given by $\mathcal{N}\left(x, \frac{r^2}{d^{3/2}\kappa^4}J_x^{-1}\right)$, which we denote by $\mathcal{P}_x^{\text{J}}$. In particular, the proposal density at point $x\in\mathrm{int}\,(\mathcal{K})$ is given by

$$p_x(z) := p(x,z) = \sqrt{\det J_x}\left(\frac{\kappa^4 d^{3/2}}{2\pi r^2}\right)^{d/2}\exp\left(-\frac{\kappa^4 d^{3/2}}{2r^2}\,(z-x)^\top J_x(z-x)\right). \qquad (71)$$

Here we restate our result for the mixing time of the John walk.

Theorem 2 *Let μ_0 be any distribution that is M-warm with respect to π^* and let $n < \exp(\sqrt{d})$. Then for any $\delta\in(0,1]$, the John walk with parameter $r_{John} = 10^{-5}$ satisfies*

$$\|\mathcal{T}_{John(r)}^k(\mu_0) - \pi^*\|_{TV} \le \delta \qquad \textit{for all } k \ge C\,d^{2.5}\,\log_2^4\left(\frac{2n}{d}\right)\log\left(\frac{\sqrt{M}}{\delta}\right).$$

D.1 Auxiliary results

We begin by proving basic properties of the weights ζ_x which are used throughout the paper. For $x\in\mathrm{int}\,(\mathcal{K}), w\in\mathbb{R}^n_{++}$, define the projection matrix $\Upsilon_{x,w}$ as follows

$$\Upsilon_{x,w} = W^{\alpha/2}A_x(A_x^\top W^\alpha A_x)^{-1}A_x^\top W^{\alpha/2}, \qquad (72)$$

where $A_x = S_x^{-1}A$ and W is the $n\times n$ diagonal matrix with i-th diagonal entry given by w_i. Also, let

$$\sigma_{x,i} := (\Upsilon_{x,\zeta_x})_{ii} \quad \text{for } x\in\mathrm{int}\,(\mathcal{K}) \text{ and } i\in[n]. \qquad (73)$$

Define the *John slack sensitivity* θ_x^{J} as

$$\theta_x := \theta_x^{\text{J}} := \left(\frac{a_1^\top J_x^{-1}a_1}{s_{x,1}^2},\ldots,\frac{a_n^\top J_x^{-1}a_n}{s_{x,n}^2}\right)^\top \quad \text{for all } x\in\mathrm{int}\,(\mathcal{K}). \qquad (74)$$

Further, for any $x\in\mathrm{int}\,(\mathcal{K})$, define the *John local norm at x* as

$$\|\cdot\|_{J_x} : v\mapsto\left\|J_x^{1/2}v\right\|_2 = \sqrt{\sum_{i=1}^n \zeta_{x,i}\frac{(a_i^\top v)^2}{s_{x,i}^2}}. \qquad (75)$$

We now collect some basic properties of the weights ζ_x and the local sensitivity θ_x and restate parts of Lemma 7 for clarity here.

Lemma 3 *For any $x \in \text{int}(\mathcal{K})$, the following properties are true:*

(a) (Implicit weight formula) $\zeta_{x,i} = \sigma_{x,i} + \beta_J$ for all $i \in [n]$,

(b) (Uniformity) $\zeta_{x,i} \in [\beta_J, 1 + \beta_J]$ for all $i \in [n]$,

(c) (Total size) $\sum_{i=1}^n \zeta_{x,i} = 3d/2$, and

(d) (Slack sensitivity) $\theta_{x,i} \in [0, 4]$ for all $i \in [n]$.

Lemma 3 follows from Lemmas 14 and 15 by Lee and Sidford (2014) and thereby we omit its proof.

Next, we state a key lemma that is crucial for proving the convergence rate of John walk. In this lemma, we provide bounds on difference in total variation norm between the proposal distributions of two nearby points.

Lemma 4 *There exists a continuous non-decreasing function $h : [0, 1/30] \to \mathbb{R}_+$ with $h(1/30) \geq 10^{-5}$, such that for any $\epsilon \in (0, 1/30]$, the John walk with $r \in [0, h(\epsilon)]$ satisfies*

$$\|\mathcal{P}_x^J - \mathcal{P}_y^J\|_{TV} \leq \epsilon, \quad \text{for all } x, y \in \text{int}(\mathcal{K}) \text{ such that } \|x - y\|_{J_x} \leq \frac{\epsilon r}{2\kappa^2 d^{3/4}}, \quad \text{and} \tag{76a}$$

$$\|\mathcal{T}_{John(r)}(\delta_x) - \mathcal{P}_x^J\|_{TV} \leq 5\epsilon, \text{ for all } x \in \text{int}(\mathcal{K}). \tag{76b}$$

See Section D.3 for its proof.

With these lemmas in hand, we are now ready to prove Theorem 2.

D.2 Proof of Theorem 2

The proof is similar to the proof of Theorem 1, and relies on the Lovász's Lemma. Here onwards, we use the following simplified notation

$$\mathbb{T}_x = \mathcal{T}_{\text{John}(r)}(\delta_x), \mathcal{P}_x = \mathcal{P}_x^{\text{J}} \text{ and } \|\cdot\|_x = \|\cdot\|_{J_x}.$$

In order to invoke Lovász's Lemma, we need to show that for any two points $x, y \in \text{int}(\mathcal{K})$ with small cross-ratio $d_{\mathcal{K}}(x, y)$, the TV-distance $\|\mathbb{T}_x - \mathbb{T}_y\|_{\text{TV}}$ is also small.

We proceed with the proof in two steps: (A) first, we relate the cross-ratio $d_{\mathcal{K}}(x, y)$ to the John local norm of $x - y$ at x, and (B) we then use Lemma 4 to show that if $x, y \in \text{int}(\mathcal{K})$ are close in the John local-norm, then the transition kernels $\mathbb{T}_x$ and $\mathbb{T}_y$ are close in TV-distance.

Step (A): We claim that for all $x, y \in \text{int}(\mathcal{K})$, the cross-ratio can be lower bounded as

$$d_{\mathcal{K}}(x, y) \geq \frac{1}{\sqrt{3d/2}} \|x - y\|_x. \tag{77}$$

From the arguments in the proof of Theorem 1 (proof for the Vaidya Walk), we have

$$d_{\mathcal{K}}(x, y) \geq \max_{i \in [n]} \left| \frac{a_i^\top (x - y)}{s_{x,i}} \right|. \tag{78}$$

Using the fact that maximum of a set of non-negative numbers is greater than the weighted mean of the numbers and Lemma 3, we find that

$$d_{\mathcal{K}}(x, y) \geq \sqrt{\frac{1}{\sum_{i=1}^n \zeta_{x,i}} \sum_{i=1}^n \zeta_{x,i} \frac{(a_i^\top (x - y))^2}{s_{x,i}^2}} = \frac{\|x - y\|_x}{\sqrt{3d/2}},$$

thereby proving the claim (77).

Step (B): By the triangle inequality, we have

$$\|\mathbb{T}_x - \mathbb{T}_y\|_{\text{TV}} \leq \|\mathbb{T}_x - \mathcal{P}_x\|_{\text{TV}} + \|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} + \|\mathcal{P}_y - \mathbb{T}_y\|_{\text{TV}}.$$

Using Lemma 4, we obtain that

$$\|\mathbb{T}_x - \mathbb{T}_y\|_{\text{TV}} \leq 11\epsilon, \quad \forall x, y \in \text{int}(\mathcal{K}) \text{ such that } \|x - y\|_x \leq \frac{\epsilon r}{2\kappa^2 d^{3/4}}.$$

Consequently, the John walk satisfies the assumptions of Lovász's Lemma with

$$\Delta := \frac{1}{\sqrt{3d/2}} \cdot \frac{\epsilon r}{2\kappa^2 d^{3/4}} \quad \text{and} \quad \rho := 1 - 11\epsilon.$$

Plugging in $\epsilon = 1/30$, $r = 10^{-5}$, we obtain the claimed upper bound of $\mathcal{O}\left(\kappa^4 d^{5/2}\right)$ on the mixing time of the random walk.

D.3 Proof of Lemma 4

We prove the lemma for the following function,

$$h(\epsilon) = \min \left\{ \frac{1}{25\sqrt{1 + \sqrt{2}\log(4/\epsilon)}}, \frac{\epsilon}{(2\sqrt{32}\chi_{1,\epsilon})}, \sqrt{\frac{\epsilon}{386\sqrt{24}\chi_{2,\epsilon}}}, \frac{\epsilon}{5\sqrt{60}\chi_{3,\epsilon}}, \right.$$
$$\left. \sqrt{\frac{\epsilon}{8\sqrt{1680}\chi_{4,\epsilon}}}, \sqrt{\frac{\epsilon}{40\left(\chi_{2,\epsilon}\chi_{6,\epsilon}\sqrt{24}\sqrt{15120}\right)^{1/2}}}, \sqrt{\frac{\epsilon}{204800\chi_{2,\epsilon}\sqrt{24}\log(32/\epsilon)}} \right\}.$$

where $\chi_{1,\epsilon} = \log(2/\epsilon)$and $\chi_{k,\epsilon} = \left(2e/k \cdot \log(16/\epsilon)\right)^{k/2}$ for $k = 2, 3, 4$ and 6. A numerical calculation shows that $h(1/30) \geq 10^{-5}$.

We now prove the two parts (76a) (76b) of the Lemma separately.

D.3.1 Proof of claim (76a)

Applying Pinsker's inequality, and plugging in the closed formed expression for the KL divergence between two Gaussian distributions we find that

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}}^2 \leq 2\,\text{KL}(\mathcal{P}_y \| \mathcal{P}_x) = \text{trace}(J_x^{-1/2} J_y J_x^{-1/2}) - d - \log\det(J_x^{-1/2} J_y J_x^{-1/2}) + \frac{\kappa^4 d^{3/2}}{r^2} \|x - y\|_x^2$$
$$= \sum_{i=1}^d \left(\lambda_i - 1 + \log\frac{1}{\lambda_i} \right) + \frac{\kappa^4 d^{3/2}}{r^2} \|x - y\|_x^2, \tag{79}$$

where $\lambda_1, \ldots, \lambda_d > 0$ denote the eigenvalues of the matrix $J_x^{-1/2} J_y J_x^{-1/2}$. To bound the expression (79), we make use of the following lemma:

Lemma 5 *For any scalar $t \in [0, 1/64]$ and pair of points $x, y \in \operatorname{int}(\mathcal{K})$ such that $\|x - y\|_x \le t/\kappa^2$, we have*

$$\left(1 - 48t + 4t^2\right) \mathbb{I}_d \preceq J_x^{-1/2} J_y J_x^{-1/2} \preceq \left(1 + 48t + 4t^2\right),$$

where $\preceq$ denotes ordering in the PSD cone and $\mathbb{I}_d$ denotes the d-dimensional identity matrix.

See Section F for the proof of this lemma.

For $\epsilon \in (0, 1/30]$ and $r = 10^{-5}$, we have $t = \epsilon r/(2d^{3/4}) \le 1/64$, whence the eigenvalues $\{\lambda_i, i \in [d]\}$ can be sandwiched as

$$1 - \frac{24\epsilon r}{d^{3/4}} + \frac{\epsilon^2 r^2}{d^{3/2}} \le \lambda_i \le 1 + \frac{24\epsilon r}{d^{3/4}} + \frac{\epsilon^2 r^2}{d^{3/2}} \quad \text{for all } i \in d. \tag{80}$$

We are now ready to bound the TV distance between $\mathcal{P}_x$ and $\mathcal{P}_y$. Using the bound (79) and the inequality $\log \omega \le \omega - 1$, valid for $\omega > 0$, we obtain

$$\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}}^2 \le \sum_{i=1}^{d} \left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\kappa^4 d^{3/2}}{r^2} \|x - y\|_x^2.$$

Using the assumption that $\|x - y\|_x \le \epsilon r / \left(2\kappa^2 d^{3/4}\right)$, and plugging in the bounds (80) for the eigenvalues $\{\lambda_i, i \in [d]\}$, we find that

$$\sum_{i=1}^{d} \left(\lambda_i - 2 + \frac{1}{\lambda_i}\right) + \frac{\kappa^4 d^{3/2}}{r^2} \|x - y\|_x^2 \le \frac{2000\epsilon^2 r^2}{\sqrt{d}} + \frac{\epsilon^2}{4}.$$

In asserting this inequality, we have used the facts that

$$\frac{1}{1 - 24\omega + \omega^2} \le 1 + 24\omega + 1000\omega^2, \quad \text{and} \quad \frac{1}{1 + 24\omega + \omega^2} \le 1 - 24\omega + 1000\omega^2 \quad \text{for all } \omega \in \left[0, \tfrac{1}{100}\right].$$

Note that for any $r \in [0, 1/100]$, we have that $2000r^2/\sqrt{d} \le 1/2$. Putting the pieces together yields $\|\mathcal{P}_x - \mathcal{P}_y\|_{\text{TV}} \le \epsilon$, as claimed.

D.3.2 Proof of claim (76b)

We have

$$\|\mathcal{P}_x - \mathbb{T}_x\|_{\text{TV}} \le \underbrace{\frac{3}{2}\mathcal{P}_x(\mathcal{K}^c)}_{=: S_1} + \underbrace{1 - \mathbb{E}_{z \sim \mathcal{P}_x}\left[\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}\right]}_{=: S_2}, \tag{81}$$

where $\mathcal{K}^c$ denotes the complement of $\mathcal{K}$. We now show that $S_1 \le \epsilon$ and $S_2 \le 4\epsilon$, from which the claim follows.

Bounding the term S_1: Note that for $z \sim \mathcal{N}(x, \frac{r^2}{\kappa^2 d^{3/2}} J_x^{-1})$, we can write

$$z \stackrel{d}{=} x + \frac{r}{\kappa d^{3/4}} J_x^{-1/2} \xi, \tag{82}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$ and $\stackrel{d}{=}$ denotes equality in distribution. Using equation (82) and definition (74) of $\theta_{x,i}$, we obtain the bound

$$\frac{\left(a_i^\top (z - x)\right)^2}{s_{x,i}^2} = \frac{r^2}{\kappa^2 d^{3/2}} \left[\frac{a_i^\top J_x^{-1/2} \xi}{s_{x,i}}\right]^2 \stackrel{(i)}{\le} \frac{r^2}{\kappa^2 d^{3/2}} \theta_{x,i} \|\xi\|_2^2 \stackrel{(ii)}{\le} \frac{4r^2}{d} \|\xi\|_2^2, \tag{83}$$

where step (i) follows from Cauchy-Schwarz inequality, and step (ii) from part (d) of Lemma 3. Define the events

$$\mathcal{E} := \left\{\frac{r^2}{d} \|\xi\|_2^2 < \frac{1}{4}\right\} \quad \text{and} \quad \mathcal{E}' := \{z \in \operatorname{int}(\mathcal{K})\}.$$

Inequality (83) implies that $\mathcal{E} \subseteq \mathcal{E}'$ and hence $\mathbb{P}[\mathcal{E}'] \ge \mathbb{P}[\mathcal{E}]$. Using a standard Gaussian tail bound and noting that $r \le \frac{1/2}{1 + \sqrt{2/d \log(2/\epsilon)}}$, we obtain $\mathbb{P}[\mathcal{E}] \ge 1 - \epsilon/2$ and whence $\mathbb{P}[\mathcal{E}'] \ge 1 - \epsilon/2$. Thus, we have shown that $\mathbb{P}[z \notin \mathcal{K}] \le \epsilon/2$ which implies that $S_1 \le \epsilon$.

Bounding the term S_2: By Markov's inequality, we have

$$\mathbb{E}_{z \sim \mathcal{P}_x}\left[\min\left\{1, \frac{p_z(x)}{p_x(z)}\right\}\right] \ge \alpha \mathbb{P}\left[p_z(x) \ge \alpha p_x(z)\right] \quad \text{for all } \alpha \in (0, 1]. \tag{84}$$

By definition (71) of p_x, we obtain

$$\frac{p_z(x)}{p_x(z)} = \exp\left(-\frac{d^{3/2}\kappa^4}{2r^2}\left(\|z - x\|_z^2 - \|z - x\|_x^2\right) + \frac{1}{2}\left(\log\det J_z - \log\det J_x\right)\right).$$

The following lemma provides us with useful bounds on the two terms in this expression, valid for any $x \in \operatorname{int}(\mathcal{K})$.

Lemma 6 *For any $\epsilon \in (0, \frac{1}{4}]$ and $r \in (0, h(\epsilon)]$, we have*

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\frac{1}{2}\log\det J_z - \frac{1}{2}\log\det J_x \ge -\epsilon\right] \ge 1 - \epsilon, \quad \textit{and} \tag{85a}$$

$$\mathbb{P}_{z \sim \mathcal{P}_x}\left[\|z - x\|_z^2 - \|z - x\|_x^2 \le 2\epsilon \frac{r^2}{\kappa^4 d^{3/2}}\right] \ge 1 - \epsilon. \tag{85b}$$

We provide the of this lemma in Section G.

Using Lemma 6, we now complete the proof of the Theorem 2. For $r \le h(\epsilon)$, we obtain

$$\frac{p_z(x)}{p_x(z)} \ge \exp(-2\epsilon) \ge 1 - 2\epsilon$$

with probability at least $1 - 2\epsilon$. Substituting $\alpha = 1 - 2\epsilon$ in inequality (84) yields that $S_2 \le 4\epsilon$, as claimed.

Appendix E. Technical Lemmas for the John walk

We begin by summarizing a few key properties of various terms involved in our analysis.

Let $\Sigma_{x,w}$ be an $n \times n$ diagonal matrix defined as

$$\Sigma_{x,w} = \operatorname{diag}(\sigma_{x,w,i}, \ldots, \sigma_{x,w,n}) \text{ where } \sigma_{x,\zeta_x,w,i} = (\Upsilon_{x,w})_{ii}, i \in [n]. \tag{86a}$$

Let $\Upsilon^{(2)}_{x,w}$ denote the hadamard product of $\Upsilon_{x,w}$ with itself. Further define

$$\Lambda_{x,w} := \Sigma_{x,w} - \Upsilon^{(2)}_{x,w}. \tag{86b}$$

Lee and Sidford (2014) proved that the weight vector ζ_x is the unique solution of the following fixed point equation:

$$w_i = \sigma_{x,w,i} + \beta_{\text{J}}, i \in [n]. \tag{87a}$$

To simplify notation, we use the following shorthands:

$$\sigma_x = \sigma_{x,\zeta_x}, \quad \Upsilon_x = \Upsilon_{x,\zeta_x}, \quad \Upsilon^{(2)}_x = \Upsilon^{(2)}_{x,\zeta_x}, \quad \Sigma_x = \Sigma_{x,\zeta_x}, \quad \Lambda_x = \Lambda_{x,\zeta_x}. \tag{87b}$$

Thus, we have the following relation:

$$\zeta_x = \sigma_{x,\zeta_x} + \beta_{\text{J}}\mathbf{1} = \sigma_x + \beta_{\text{J}}\mathbf{1}. \tag{87c}$$

E.1 Deterministic expressions and bounds

We now collect some properties of various terms defined above.

Lemma 7 *For any $x \in \operatorname{int}(\mathcal{K})$, the following properties hold:*

(a) $\sigma_{x,i} = \sum_{j=1}^n \sigma^2_{x,i,j} = \sum_{j,k=1}^n \sigma_{x,i,j}\sigma_{x,j,k}\sigma_{x,k,i}$ *for each* $i \in [n]$,

(b) $\Sigma_x \succeq \Upsilon^{(2)}_x$,

(c) $\sum_{i=1}^n \zeta_{x,i}\theta_{x,i} = d$,

(d) $\theta_{x,i} = \sum_{j=1}^n \zeta_{x,i}\theta^2_{x,i,j}$, *for each* $i \in [n]$,

(e) $\theta_x^\top \Sigma_x \theta_x = \sum_{i=1}^n \theta^2_{x,i}\zeta_{x,i} \le 4d$, *and*

(f) $\beta_{\text{J}} \nabla^2 \mathcal{F}_x \preceq J_x \preceq (1+\beta_{\text{J}})\nabla^2\mathcal{F}_x$.

The proof is based on the ideas similar to Lemma 5 in the proof of the Vaidya walk and is thereby omitted.

The next lemma relates the change in *slackness* $s_{x,i} = b_i - a_i^\top x$ to the John-local norm at x.

Lemma 8 *For all $x, y \in \operatorname{int}(\mathcal{K})$, we have*

$$\max_{i\in[n]} \left| 1 - \frac{s_{y,i}}{s_{x,i}} \right| \le 2\|x-y\|_x.$$

Proof For any pair $x, y \in \operatorname{int}(\mathcal{K})$ and index $i \in [n]$, we have

$$\left(a_i^\top (x-y)\right)^2 \overset{(i)}{\le} \|J_x^{-\frac{1}{2}} a_i\|_2^2 \, \|J_x^{\frac{1}{2}}(x-y)\|_2^2 = \theta_{x,i} s^2_{x,i} \, \|x-y\|_x^2 \overset{(ii)}{\le} 4 s^2_{x,i} \, \|x-y\|_x^2,$$

where step (i) follows from the Cauchy-Schwarz inequality, and step (ii) uses the bound $\theta_{x,i}$ from Lemma 3(d). Noting the fact that $a_i^\top(x-y) = s_{y,i} - s_{x,i}$, the claim follows after simple algebra. ■

We now state various expressions and bounds for the first and second order derivatives of the different terms. To lighten notation, we introduce some shorthand notation. For any $y \in \operatorname{int}(\mathcal{K})$ and $h \in \mathbb{R}^d$, define the following terms:

$$d_{y,i} = \frac{a_i^\top h}{s_{y,i}}, \ i \in [n] \qquad D_y = \operatorname{diag}(d_{y,1}, \ldots, d_{y,n}), \tag{88a}$$

$$f_{y,i} = \frac{\nabla \zeta_{y,i}^\top h}{\zeta_{y,i}}, \ i \in [n] \qquad F_y = \operatorname{diag}(f_{y,1}, \ldots, f_{y,n}), \tag{88b}$$

$$\ell_{y,i} = \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h / \zeta_{y,i}, \ i \in [n] \qquad L_y = \operatorname{diag}(\ell_{y,1}, \ldots, \ell_{y,n}), \tag{88c}$$

$$\rho_y := (G_y - \alpha\Lambda_y) \begin{bmatrix} \ell_{y,1} \\ \vdots \\ \ell_{y,n} \end{bmatrix}, \tag{88d}$$

where for brevity in our notation we have omitted the dependence on h. The choice of h is specified as per the context. Further, we define for each $x \in \operatorname{int}(\mathcal{K})$ and $i \in [n]$

$$\varphi_{x,i} := \frac{\zeta_{x,i}}{s^2_{x,i}}, \qquad \text{and} \qquad \Psi_x := \frac{1}{2}\log\det J_x, \tag{89}$$

$$\hat{a}_{x,i} := \frac{J_x^{-1/2} a_{x,i}}{s^2_{x,i}}, \qquad \text{and} \qquad \hat{b}_{x,i} := J_x^{-1/2} A_x \Lambda_x (G_x - \alpha\Lambda_x)^{-1} e_i. \tag{90}$$

Next, we state expressions for gradients of ζ, φ and Ψ and bounds for directional Hessian of σ, φ and Ψ which are used in various Taylor series expansions and bounds in our proof.

Lemma 9 (Calculus) *For any $y \in \operatorname{int}(\mathcal{K})$ and $h \in \mathbb{R}^n$, the following relations hold;*

(a) Gradient of ζ: $(f_{y,1}, \ldots, f_{y,n})^\top = 2(G_y - \alpha\Lambda_y)^{-1}\Lambda_y A_y h$;

(b) Hessian of ζ:

$$\|\rho_y\|_1 \le 56\kappa^2 \sum_{i=1}^n \zeta_{y,i} d^2_{y,i}. \tag{91}$$

(c) Gradient of Ψ: $\nabla\Psi^\top h = \theta_y^\top G_y \left(\mathbb{I}_n + (G_y - \alpha\Lambda_y)^{-1}\Lambda_y\right) A_y h$.

(d) Gradient of φ: $\nabla\varphi_{y,i}^\top h = \varphi_{y,i}(2d_{y,i} + f_{y,i})$.

(e) Bound on $\nabla^2\Psi$: $\frac{1}{2}\left|h^\top(\nabla^2\Psi)h\right| \le \frac{1}{2}\left[\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\left[9\,d_{y,i}^2 + 4f_{y,i}^2\right] + \left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right|\right]$

(f) Bound on $\nabla^2\varphi$:

$$\left|\sum_{i=1}^n d_{y,i}^2 s_{y,i}^2 \frac{1}{2} h^\top \nabla^2\varphi_{y,i} h\right| \le 3\sum_{i=1}^n \zeta_{y,i} d_{y,i}^4 + 2\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^3 f_{y,i}\right| + \left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right|.$$

The proof is provided in Section H.1.

Next, we state some results that would be useful to provide explicit bounds for various terms like f_y, ℓ_y and ρ_y that appear in the statements of the previous lemma. Note that the following results do not have a corresponding analog in our analysis of the Vaidya walk.

Lemma 10 *For any* $c_1, c_2 \ge 0$, $y \in \operatorname{int}(\mathcal{K})$, *we have*

$$\left(c_1\mathbb{I}_n + c_2\Lambda_y\left(G_y - \alpha\Lambda_y\right)^{-1}\right) G_y \left(c_1\mathbb{I}_n + c_2\left(G_y - \alpha\Lambda_y\right)^{-1}\Lambda_y\right) \preceq \left(c_1 + c_2\right)^2 \kappa^2 G_y,$$

where $\preceq$ *denotes the ordering in the PSD cone.*

Lemma 11 *Let* μ_y *denote the* $n \times n$ *matrix* $\left(G_y - \alpha\Lambda_y\right)^{-1} G_y$, *and let* $\mu_{y,i,j}$ *denote its* ij*-th entry. Then for each* $i \in [n]$ *and* $y \in \operatorname{int}(\mathcal{K})$, *we have*

$$\mu_{y,i,i} \in [0, \kappa], \qquad \textit{and,} \tag{92a}$$

$$\sum_{j \ne i, j \in [n]} \frac{\mu_{y,i,j}^2}{\zeta_{y,j}} \le \kappa^3. \tag{92b}$$

Corollary 12 *Let* $e_i \in \mathbb{R}^n$ *denote the unit vector along* i*-th axis. Then for any* $y \in \operatorname{int}(\mathcal{K})$, *we have*

$$\left\|G_y\left(G_y - \alpha\Lambda_y\right)^{-1} e_i\right\|_1 \le 3\sqrt{d}\kappa^{3/2}, \qquad \textit{for all } i \in [n]. \tag{93}$$

Consequently, we also have $\left\|\left(G_y - \alpha\Lambda_y\right)^{-1} G_y\right\|_\infty \le 3\sqrt{d}\kappa^{3/2}$.

See Section H.2, H.3 and H.4 for the proofs of Lemma 10, Lemma 11 and Corollary 12 respectively.

E.2 Tail Bounds

We now collect lemmas that provide us with useful tail bounds.

We start with a result that shows that for a random variable $z \sim \mathcal{P}_x$, the slackness $s_{z,i}$ is close to $s_{x,i}$ with high probability and consequently the weights $\zeta_{z,i}$ are also close to $\zeta_{x,i}$. This result comes in handy for transferring the remainder terms in Taylor expansions to the reference point (around which the series is being expanded).

Lemma 13 *For any point* $x \in \operatorname{int}(\mathcal{K})$ *and* $r \le \frac{1}{25\cdot\sqrt{1+\sqrt{2}\log(4/\epsilon)}}$, *we have*

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\forall i \in [n], \forall v \in \overline{xz},\ \frac{s_{x,i}}{s_{v,i}} \in [0.99, 1.01] \ \textit{ and } \frac{\zeta_{x,i}}{\zeta_{v,i}} \in [0.96, 1.04]\right] \ge 1 - \epsilon/4 \tag{94a}$$

See Section I.1 for the proof of this lemma.

Next, we state high probability results for some Gaussian polynomials. These results are useful to bound various polynomials of the form $\sum_{i=1}^n \zeta_{x,i} d_{x,i}^k$, where $d_{x,i} = a_i^\top(z - x)/s_{x,i}$ and z is drawn from the transition distribution for the John walk at point x.

Lemma 14 (Gaussian moment bounds) *To simplify notations, all subscripts on* x *are omitted in the following statements. For any* $\epsilon \in (0, 1/30]$, *define* $\chi_k := \chi_{k,\epsilon} = (2e/k \cdot \log(16/\epsilon))^{k/2}$, *for* $k = 2, 3, 4$ *and* 6, *then we have*

$$\mathbb{P}\left[\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^2 \le \chi_2\sqrt{24}d\right] \ge 1 - \frac{\epsilon}{16}, \tag{95a}$$

$$\mathbb{P}\left[\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^3 \le \chi_3\sqrt{60}d^{1/2}\right] \ge 1 - \frac{\epsilon}{16}, \tag{95b}$$

$$\mathbb{P}\left[\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^2\left(\hat{b}_i^\top\xi\right) \le \chi_3\sqrt{240}\kappa d^{1/2}\right] \ge 1 - \frac{\epsilon}{16}, \tag{95c}$$

$$\mathbb{P}\left[\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^4 \le \chi_4\sqrt{1680}d\right] \ge 1 - \frac{\epsilon}{16}, \tag{95d}$$

$$\mathbb{P}\left[\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^6 \le \chi_6\sqrt{15120}d\right] \ge 1 - \frac{\epsilon}{16}. \tag{95e}$$

See Section I.2 for the proof.

Appendix F. Proof of Lemma 5

As a direct consequence of Lemma 8, for any $x, y \in \operatorname{int}(\mathcal{K})$ such that $\|x - y\|_x \le t/\kappa^2$, we have

$$\max_{i\in[n]}\left|1 - \frac{s_{y,i}}{s_{x,i}}\right| \le \frac{2t}{\kappa^2}. \tag{96}$$

Bounding the terms in $\nabla^2\mathcal{F}_x$ one by one, we obtain

$$\left(1 - \frac{2t}{\kappa^2}\right)^2 \nabla^2\mathcal{F}_y \preceq \nabla^2\mathcal{F}_x \preceq \left(1 + \frac{2t}{\kappa^2}\right)^2 \nabla^2\mathcal{F}_y.$$

We claim that

$$\|\log\zeta_y - \log\zeta_x\|_\infty \le 16t. \tag{97}$$

Assuming the claim as given at the moment, we now complete the proof. Putting the result (97) in matrix form, we obtain that $\exp(-16t)\,\mathbb{I}_n \preceq G_x^{-1}G_y \preceq \exp(16t)\,\mathbb{I}_n$, and hence

$$\exp(-16t)\,\zeta_{x,i} \le \zeta_{y,i} \le \exp(16t)\,\zeta_{x,i}. \tag{98}$$

Consequently, using the definition of J_x we have,

$$\underbrace{\left(1-\frac{2t}{\kappa^2}\right)^2 \exp\left(-16t\right)}_{\omega_\ell} J_x \preceq J_y \preceq \underbrace{\left(1+\frac{2t}{\kappa^2}\right)^2 \exp\left(16t\right)}_{\omega_u} J_y.$$

Letting $\omega = 2t$, we obtain

$$\omega_\ell \geq (1-\omega)^2 \cdot \exp\left(-8\omega\right) \overset{(i)}{\geq} 1 - 24\omega + \omega^2, \quad \text{and} \quad \omega_u \leq (1+\omega)^2 \cdot \exp\left(8\omega\right) \overset{(ii)}{\leq} 1 + 24\omega + \omega^2,$$

where inequalities (i) and (ii) hold since $\omega \leq 1/24$. Putting the pieces together, we find that

$$\left(1 - 48t + 4t^2\right) J_x \preceq J_y \preceq \left(1 - 48t + 4t^2\right) J_x$$

for $t \in [0, 1/48]$.

Now, we return to the proof of our earlier claim (97). We use an argument based on the continuity of the function $x \mapsto \log \zeta_x$. (Such an argument appeared in a similar scenario in Lee and Sidford (2014).) For $\lambda \in [0,1]$, define $u_\lambda = \lambda y + (1-\lambda)\, x$. Let

$$\lambda^{\max} := \sup\left\{\lambda \in [0,1] \,\middle|\, \left\|\log \zeta_{u_\lambda} - \log \zeta_x\right\|_\infty \leq 16t\right\}. \tag{99}$$

It suffices to establish that $\lambda^{\max} = 1$. Note that $\lambda = 0$ is feasible on the RHS of equation (99) and hence $\lambda^{\max}$ exists. Now for any $\lambda \in [0, \lambda^{\max}]$ and $i \in \{1, \ldots, n\}$, there exists v on the segment $\overline{u_\lambda x}$ such that

$$\left|\log \zeta_{u_\lambda, i} - \log \zeta_{x,i}\right| = \left|\left(\frac{\nabla \zeta_{v,i}}{\zeta_{v,i}}\right)^\top (u_\lambda - x)\right| \overset{(i)}{\leq} \left\|G_v^{-1} G_v' \,(y-x)\right\|_\infty = 2\left\|\left(G_v - \alpha\Lambda_v\right)^{-1} \Lambda_v A_v \,(y-x)\right\|_\infty.$$

where in step (i) we have used the fact that $u_\lambda - x = \lambda(y-x)$ and $\lambda \in [0,1]$. We claim that

$$\left\|\left(G_v - \alpha\Lambda_v\right)^{-1} \Lambda_v v_1\right\|_\infty \leq \kappa \left\|v_1\right\|_\infty + 2\kappa^2 \left\|G_v^{1/2} v_1\right\|_2 \quad \text{for any } v_1 \in \mathbb{R}^n. \tag{100}$$

We prove the claim at the end of this section. We now derive bounds for the two terms on the RHS of the equation (100) for $v_1 = A_v(y-x)$. Note that

$$\left\|A_v\,(y-x)\right\|_\infty = \max_i \left|\frac{s_{y,i} - s_{x,i}}{s_{v,i}}\right| = \max_i \left|\frac{s_{y,i} - s_{x,i}}{s_{x,i}}\right| \left|\frac{s_{x,i}}{s_{v,i}}\right| \overset{(i)}{\leq} \frac{2t}{\kappa^2\left(1 - 2t/\kappa^2\right)} \overset{(ii)}{\leq} \frac{3t}{\kappa^2}.$$

Inequality (i) uses bound (96) and inequality (ii) follows by plugging in $t \leq 1/64$. Next, we have

$$\begin{aligned}
\left\|G_v^{1/2} A_v\,(y-x)\right\|_2^2 = \sum_{i=1}^n \zeta_{x,i} \frac{\left(a_i^\top\,(y-x)\right)^2}{s_{x,i}^2} \frac{\zeta_{v,i}}{\zeta_{x,i}} \frac{s_{v,i}^2}{s_{x,i}^2} &\overset{(i)}{\leq} \left\|x-y\right\|_x^2 \max_{i\in[n]} \frac{\zeta_{v,i}}{\zeta_{x,i}} \frac{s_{v,i}^2}{s_{x,i}^2} \\
&\overset{(ii)}{\leq} \frac{t^2}{\kappa^4}\left(1 + (16t) + (16t)^2\right)\left(1 + \frac{2t}{\kappa^2}\right)^2 \\
&\overset{(iii)}{\leq} \frac{1.5t}{\kappa^4},
\end{aligned}$$

where step (i) follows from the definition of the local norm; step (ii) follows from bounds (96) and (99) and the fact that $e^x \leq 1 + x + x^2$ for all $x \in [0, 1/4]$; and inequality (iii) follows by plugging in $t \leq 1/64$. Putting the pieces together, we obtain

$$\left\|\log \zeta_{u_\lambda} - \log \zeta_x\right\|_\infty \leq 2(\kappa \cdot 3t/\kappa^2 + 2\kappa^2 \cdot 1.5t/\kappa^4) \leq 12t < 16t.$$

The strict inequality is valid for $\lambda = \lambda^{\max}$. Consequently, using the continuity of $x \mapsto \log \zeta_x$, we conclude that $\lambda^{\max} = 1$.

It is left to prove claim (100). Let $w := \left(G_v - \alpha\Lambda_v\right)^{-1} \Lambda_v v_1$. which implies $\left(G_v - \alpha\Lambda_v\right) w = \Lambda_v v_1$. Plugging the expression of G_v and Λ_v, we have

$$\left((1-\alpha)\Sigma_v + \beta_\jmath \mathbb{I}_n + \alpha\Upsilon_v^{(2)}\right) w = \left(\Sigma_v - \Upsilon_v^{(2)}\right) v_1.$$

Writing component wise, we find that for any $i \in [n]$, we have

$$\begin{aligned}
\left|\left((1-\alpha)\sigma_{v,i} + \beta_\jmath\right) w_i\right| &\leq \alpha\left|e_i^\top \Upsilon_v^{(2)} w\right| + \sigma_{v,i}\left|v_{1,i}\right| + \left|e_i^\top \Upsilon_v^{(2)} v_1\right| \\
&\overset{(i)}{\leq} \alpha\sigma_{v,i}\left\|\Sigma_v^{1/2} w\right\|_2 + \sigma_{v,i}\left\|v_1\right\|_\infty + \sigma_{v,i}\left\|\Sigma_v^{1/2} v_1\right\|_2 \\
&\overset{(ii)}{\leq} \alpha\sigma_{v,i}\left\|G_v^{1/2} w\right\|_2 + \sigma_{v,i}\left\|v_1\right\|_\infty + \sigma_{v,i}\left\|G_v^{1/2} v_1\right\|_2 \\
&\overset{(iii)}{\leq} \alpha\sigma_{v,i}\kappa\left\|G_v^{1/2} v_1\right\|_2 + \sigma_{v,i}\left\|v_1\right\|_\infty + \sigma_{v,i}\left\|G_v^{1/2} v_1\right\|_2,
\end{aligned} \tag{101}$$

where inequality (ii) from the fact that $\Sigma_y \preceq G_y$ and inequality (iii) from Lemma 10 with $c_1 = 0, c_2 = 1$. To assert inequality (i), observe the following

$$\left|\sum_{j=1}^n \sigma_{y,i,j}^2 w_j\right| \leq \sum_{j=1}^n \sigma_{y,i,j}^2 \left|w_j\right| \overset{(a)}{\leq} \sigma_{y,i} \sum_{j=1}^n \sigma_{y,j}\left|w_j\right| \overset{(b)}{\leq} \sigma_{y,i} \sum_{j=1}^n \sqrt{\sigma_{y,j}}\left|w_j\right| = \sigma_{y,i}\left\|\Sigma_v^{1/2} w\right\|_2,$$

where step (a) follows from the fact that $\sigma_{y,i,j}^2 \leq \sigma_{y,i}\sigma_{y,j}$, and step (b) from the fac that $\sigma_{y,i} \in [0,1]$. Dividing both sides of inequality (101) by $((1-\alpha)\sigma_{v,i} + \beta_\jmath)$ and observing that $\sigma_{v,i}/\left((1-\alpha)\sigma_{v,i} + \beta_\jmath\right) \leq \kappa$, and $\alpha \in [0,1]$, yields the claim.

Appendix G. Proof of Lemma 6

We prove Lemma 6 in two parts: claim (85a) in Section G.1 and claim (85b) in Section G.2.

G.1 Proof of claim (85a)

Using the second order Taylor expansion, we have

$$\Psi_z - \Psi_x = (z-x)^\top \nabla\Psi_x + \frac{1}{2}(z-x)^\top \nabla^2\Psi_y\,(z-x), \quad \text{for some } y \in \overline{xz}.$$

We claim that for $r \leq h(\epsilon)$, we have

$$\mathbb{P}\left[(z-x)^\top \nabla\Psi_x \geq -\epsilon/2\right] \geq 1 - \epsilon/2, \text{ and} \tag{102a}$$

$$\mathbb{P}\left[\frac{1}{2}(z-x)\,\nabla^2\Psi_y\,(z-x) \geq -\epsilon/2\right] \geq 1 - \epsilon/2. \tag{102b}$$

Note that the claim (85a) follows from the above two claims.

G.1.1 Proof of bound (102a)

We observe that

$$(z-x)^\top \nabla\Psi_x \sim \mathcal{N}\left(0, \frac{r^2}{\kappa^2 n}\nabla\Psi_x^\top J_x^{-1}\nabla\Psi_x\right).$$

Let $E_x = \mathbb{I}_n + (G_x - \alpha\Lambda_x)^{-1}\Lambda_x$. Substituting the expression of $\nabla\Psi_x$ from Lemma 9 (c) and applying Cauchy-Schwarz inequality, we have that for any vector $v \in \mathbb{R}^d$

$$v^\top \nabla\Psi_x \nabla\Psi_x^\top v = (\theta_x^\top G_x E_x A_x v)^2 \le \left(v^\top A_x^\top G_x A_x v\right)\cdot\left(\theta_x^\top G_x E_x G_x^{-1} E_x G_x \theta_x\right). \tag{103}$$

Observe that

$$G_x^{1/2} E_x G_x^{-1/2} = \mathbb{I}_n + (\mathbb{I}_n - \alpha G_x^{-1/2}\Lambda_x G_x^{-1/2})^{-1}(G_x^{-1/2}\Lambda_x G_x^{-1/2}).$$

Now, using the intermediate bound (126) from the proof of Lemma 10, we obtain that

$$\mathbb{I}_n \preceq G_x^{1/2} E_x G_x^{-1/2} \preceq 2\kappa \mathbb{I}_n,$$

and hence $G_x \preceq G_x E_x G_x^{-1} E_x G_x \preceq 4\kappa^2 G_x$. Consequently, we have

$$\theta_x^\top G_x E_x G_x^{-1} E_x G_x \theta_x \le 4\kappa^2 \theta_x^\top G_x \theta_x = 4\kappa^2 \sum_{i=1}^n \zeta_{x,i}\theta_{x,i}^2 \le 16\kappa^2 d,$$

where the last step follows from Lemma 7. Putting the pieces together into equation (103), we obtain $\nabla\Psi_x\nabla\Psi_x^\top \preceq 16\kappa^2 d J_x$ whence $J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2} \preceq 16\kappa^2 d\mathbb{I}_d$. Noting that the matrix $J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2}$ has rank one, we have

$$\nabla\Psi_x^\top J_x^{-1}\nabla\Psi_x = \text{trace}\left(J_x^{-1/2}\nabla\Psi_x\nabla\Psi_x^\top J_x^{-1/2}\right) \le 16\kappa^2 d.$$

Using standard Gaussian tail bound, we have $\mathbb{P}\left((z-x)^\top\nabla\Psi_x \ge -\sqrt{32}\chi_1 r\right) \ge 1 - \exp\left(-\chi_1^2\right)$.

Choosing $\chi_1 = \log(2/\epsilon)$, and observing that

$$r \le \frac{\epsilon}{\left(2\sqrt{32}\chi_1\right)}, \tag{104}$$

yields the claim.

G.1.2 Proof of bound (102b)

In the following proof, we use $h = z - x$ for definitions (88a)-(88d). According to Lemma 9(e), we have

$$\left|\frac{1}{2}(z-x)^\top\nabla^2\Psi_y(z-x)\right| \le \sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\left[\frac{9}{2}d_{y,i}^2 + 2f_{y,i}^2\right] + \frac{1}{2}\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right|$$

We claim that

$$\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\left[\frac{9}{2}d_{y,i}^2 + 2f_{y,i}^2\right] + \frac{1}{2}\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right| \le 386\sqrt{d}\kappa^4 \sum_{i=1}^n \zeta_{y,i}d_{y,i}^2. \tag{105}$$

Assuming the claim as given at the moment, we now complete the proof. Note that y is some particular point on $\overline{xz}$ and its dependence on z is hard to characterize. Consequently, we transfer all the terms with dependence on y, to terms with dependence on x only. We have

$$\sum_{i=1}^n \zeta_{y,i}d_{y,i}^2 = \sum_{i=1}^n \zeta_{x,i}d_{x,i}^2 \underbrace{\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{s_{x,i}^2}{s_{y,i}^2}}_{\tau_{y,i}}.$$

We now invoke the following high probability bounds implied by Lemma 13 and Lemma 14 (95a) respectively

$$\mathbb{P}\left[\sup_{y\in\overline{xz},i\in[n]}\tau_{y,i} \le 1.1\right] \ge 1-\epsilon/4, \quad \text{and,} \quad \mathbb{P}\left[\sum_{i=1}^n \zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2 \le \chi_2\sqrt{24}d\right] \ge 1-\epsilon/16. \tag{106}$$

Since $h = z - x$, we have that $d_{x,i}^2 = \frac{r^2}{\kappa^2 d^{3/2}}\left(\hat{a}_{x,i}^\top\xi\right)^2$. Consequently, for

$$r \le \sqrt{\frac{\epsilon}{386\sqrt{24}\chi_2}}, \tag{107}$$

with probability at least $1-\epsilon/2$, we have

$$\left|\frac{1}{2}(z-x)^\top\nabla^2\Psi_y(z-x)\right| \overset{\text{eqn. (105)}}{\le} 386\sqrt{d}\kappa^4\sum_{i=1}^n \zeta_{y,i}d_{y,i}^2 \overset{\text{hpb (106)}}{\le} \epsilon,$$

which completes the proof.

We now turn to the proof of claim (105). First we observe the following relationship between the terms $d_{y,i}$ and $f_{y,i}$:

$$\sum_{i=1}^n \zeta_{y,i}f_{y,i}^2 \overset{(i)}{=} 4h^\top A_y^\top\Lambda_y\left(G_y-\alpha\Lambda_y\right)^{-1}G_y\left(G_y-\alpha\Lambda_y\right)^{-1}\Lambda_y A_y h \overset{(ii)}{\le} 4\kappa^2 h^\top A_y^\top G_y A_y h = 4\kappa^2\sum_{i=1}^n \zeta_{y,i}d_{y,i}^2, \tag{108}$$

where step (i) follows by plugging in the definition of $f_{y,i}$ (88b) and step (ii) by invoking Lemma 10 with $c_1 = 0$ and $c_2 = 1$. Next, we relate the term on the LHS of equation (105) involving $\ell_{y,i}$ to a polynomial in $d_{y,i}$. Using Lemma 9, we find that

$$\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right| = \left|\left(\left(G_y-\alpha\Lambda_y\right)^{-1}G_y\theta_y\right)^\top\left(G_y-\alpha\Lambda_y\right)\ell_y\right| \le \left\|\underbrace{\left(G_y-\alpha\Lambda_y\right)^{-1}G_y\theta_y}_{v_1}\right\|_\infty \left\|\underbrace{\left(G_y-\alpha\Lambda_y\right)\ell_y}_{\rho_y}\right\|_1,$$

where the last step follows from the Holder's inequality: for any two vectors $u, v \in \mathbb{R}^d$, we have that $u^\top v \le \|u\|_\infty \|v\|_1$. Substituting the bound for the norm $\|v_1\|_\infty$ from Corollary 12 and the bound on $\rho_{y,i}$ from Lemma 9(b), we obtain that

$$\left|\sum_{i=1}^n \zeta_{y,i}\,\theta_{y,i}\ell_{y,i}\right| \le 12\sqrt{n}\kappa^{3/2}\sum_{i=1}^n\left[7\zeta_{y,i}d_{y,i}^2+3\zeta_{y,i}f_{y,i}^2+\sum_{j=1}^n\left(13d_{y,j}^2+6f_{y,j}^2\right)\Upsilon_{y,i,j}^2\right] \le 672\sqrt{n}\kappa^4\sum_{i=1}^n\zeta_{y,i}d_{y,i}^2,$$

where the last step follows from Lemma 7(a) and the bound (108). The claim now follows.

G.2 Proof of claim (85b)

Writing $z = x + tu$, where t is a scalar and u is a unit vector in $\mathbb{R}^d$, we obtain

$$\|z-x\|_z^2 - \|z-x\|_x^2 = t^2\sum_{i=1}^n\left(a_i^\top u\right)^2(\varphi_{z,i}-\varphi_{x,i}).$$

Now, we use a Taylor series expansion for $\sum_{i=1}^n\left(a_i^\top u\right)^2(\varphi_{z,i}-\varphi_{x,i})$ around the point x, along the line u. There exists a point $y \in \overline{xz}$ such that

$$\sum_{i=1}^n\left(a_i^\top u\right)^2(\varphi_{z,i}-\varphi_{x,i}) = \sum_{i=1}^n\left(a_i^\top u\right)^2\left((z-x)^\top\nabla\varphi_{x,i}+\frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x)\right).$$

Note that the point y in this discussion is not the same as the point y used in previous proofs, in particular in Section G.1. Multiplying both sides by t^2, and using the shorthand $d_{x,i} = \frac{a_i^\top(z-x)}{s_{x,i}}$, we obtain

$$\|z-x\|_z^2-\|z-x\|_x^2 = \sum_{i=1}^n d_{x,i}^2s_{x,i}^2(z-x)^\top\nabla\varphi_{x,i}+\sum_{i=1}^n d_{x,i}^2s_{x,i}^2\frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x). \quad (109)$$

We claim that for $r \le h(\epsilon)$, we have

$$\mathbb{P}_{z\sim\mathbb{T}_x^{\mathrm{J}}}\left[\sum_{i=1}^n d_{x,i}^2s_{x,i}^2(z-x)^\top\nabla\varphi_{x,i} \le \epsilon\frac{r^2}{\kappa^4 d^{3/2}}\right] \ge 1-\epsilon/2, \text{ and} \quad (110a)$$

$$\mathbb{P}_{z\sim\mathbb{T}_x^{\mathrm{J}}}\left[\sup_{y\in\overline{xz}}\left(\sum_{i=1}^n d_{x,i}^2s_{x,i}^2\frac{1}{2}(z-x)^\top\nabla^2\varphi_{y,i}(z-x)\right) \le \epsilon\frac{r^2}{\kappa^4 d^{3/2}}\right] \ge 1-\epsilon/2. \quad (110b)$$

We now prove each claim separately.

G.2.1 Proof of bound (110a)

Using Lemma 9(d) and using $h = z - x$ where z is given by the relation (82), we find that

$$\begin{aligned}\sum_{i=1}^n d_{x,i}^2s_{x,i}^2(z-x)^\top\nabla\varphi_{x,i} &= \sum_{i=1}^n\zeta_{x,i}d_{x,i}^2(2d_{x,i}+f_{x,i})\\ &= \frac{r^3}{d^{9/4}\kappa^6}\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^3+\frac{2r^3}{d^{9/4}\kappa^6}\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2\left(\hat{b}_{x,i}^\top\xi\right) \quad (111)\end{aligned}$$

Using high probability bounds for the two terms in equation (111) from Lemma 14, part (95b) and part (95c), we obtain that

$$\left|\sum_{i=1}^n d_{x,i}^2s_{x,i}^2(z-x)^\top\nabla\varphi_{x,i}\right| \le \frac{5\sqrt{60}\chi_3 r^3}{\kappa^5 d^{7/4}} \le \epsilon\frac{r^2}{\kappa^4 d^{3/2}},$$

with probability at least $1-\epsilon/2$. The last inequality uses the condition that

$$r \le \frac{\epsilon}{5\sqrt{60}\chi_3}. \quad (112)$$

The claim now follows.

G.2.2 Proof of bound (110b)

Note that $d_{x,i}s_{x,i} = a_i^\top h = d_{y,i}s_{y,i}$ for any h. Using this equality for $h = z - x$, we find that

$$\begin{aligned}\left|\sum_{i=1}^n d_{x,i}^2s_{x,i}^2\frac{1}{2}h^\top\nabla^2\varphi_{y,i}h\right| &= \left|\sum_{i=1}^n d_{y,i}^2s_{y,i}^2\frac{1}{2}h^\top\nabla^2\varphi_{y,i}h\right|\\ &\overset{(i)}{\le} 3\underbrace{\sum_{i=1}^n\zeta_{y,i}d_{y,i}^4}_{C_1}+2\underbrace{\left|\sum_{i=1}^n\zeta_{y,i}d_{y,i}^3f_{y,i}\right|}_{C_2}+\underbrace{\left|\sum_{i=1}^n\zeta_{y,i}d_{y,i}^2\ell_{y,i}\right|}_{C_3}, \quad (113)\end{aligned}$$

where step (i) follows from Lemma 9(f). We can write C_1 as follows

$$\sum_{i=1}^n\zeta_{y,i}d_{y,i}^4 = \sum_{i=1}^n\zeta_{x,i}d_{x,i}^4\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^4}{d_{x,i}^4} = \frac{r^4}{n^3\kappa^8}\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^4\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^4}{d_{x,i}^4}. \quad (114)$$

Now, we claim the following:

$$C_2 \le 2\frac{r^4}{n^3\kappa^7}\cdot\sqrt{\left[\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^2}{d_{x,i}^2}\right]\cdot\left[\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^6\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^6}{d_{x,i}^6}\right]}, \quad \text{and,} \quad (115a)$$

$$C_3 \le 56\frac{r^4}{n^3\kappa^{4.5}}\left(\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^2}{d_{x,i}^2}\right)\left(\max_i\left(\hat{a}_{x,i}^\top\xi\right)^2\frac{d_{y,i}^2}{d_{x,i}^2}+\sqrt{\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^4\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^4}{d_{x,i}^4}}\right) \quad (115b)$$

Assuming the claims as given, we now complete the proof. Using Lemma 13, we have

$$\mathbb{P}\left[\frac{\zeta_{y,i}}{\zeta_{x,i}}\frac{d_{y,i}^6}{d_{x,i}^6} \le 1.2\right] \ge 1-\epsilon/4,$$

and consequently

$$\begin{aligned}3C_1+2C_2+C_3 \le \frac{r^4}{d^3\kappa^{4.5}}\Bigg[&4\cdot\sum_{i=1}^n\zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^4+10\cdot\left(\sum_{i=1}^n\zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^2\cdot\sum_{i=1}^n\zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^6\right)^{1/2}\\ &+100\cdot\sum_{i=1}^n\zeta_{x,i}\left(\hat{a}_{x,i}^\top\xi\right)^2\cdot\left(\max_i(\hat{a}_{x,i}^\top\xi)^2+\left(\sum_{i=1}^n\zeta_{x,i}(\hat{a}_{x,i}^\top\xi)^4\right)^{1/2}\right)\Bigg], \quad (116)\end{aligned}$$

with probability at least $1-\epsilon/4$. Now, we observe that for all $i \in [n]$ and $x \in \operatorname{int}(\mathcal{K})$, we have

$$\left(\hat{a}_{x,i}^\top \xi\right) \sim \mathcal{N}(0, \theta_{x,i}) \quad \text{and} \quad \theta_{x,i} \leq 4.$$

Invoking the standard tail bound for maximum of Gaussian random variables, we obtain

$$\mathbb{P}\left[\max_i \left|\left(\hat{a}_{x,i}^\top \xi\right)\right| \leq 8 \cdot \left(\sqrt{\log n} + \sqrt{\log(32/\epsilon)}\right)\right] \geq 1 - \epsilon/16.$$

Using the fact that $2c_1c_2 \geq c_1 + c_2$ for all $c_1, c_2 \geq 1$, we obtain

$$\mathbb{P}\left[\max_i \left|\left(\hat{a}_{x,i}^\top \xi\right)\right| \leq 16 \cdot \sqrt{\log n} \cdot \sqrt{\log(32/\epsilon)}\right] \geq 1 - \epsilon/16.$$

Combining this bound with the tail bounds for various Gaussian polynomials (95a), (95d), (95e) from Lemma 14, and substituting in inequality (116), we obtain that

$$\left|\sum_{i=1}^n d_{x,i}^2 s_{x,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h\right| \leq \frac{r^4}{\kappa^{6.5} d^3}\left[4 \cdot \chi_4 \sqrt{1680} d + 10\left(\chi_2 \sqrt{24} d \cdot \chi_6 \sqrt{15120} d\right)^{1/2} + 100 \cdot \chi_2 \sqrt{24} d \cdot \left(256 \cdot \log n \cdot \log(32/\epsilon) + \left(\chi_4 \sqrt{1680} d\right)^{1/2}\right)\right]$$

with probability at least $1-\epsilon/2$. In the above expression, the terms χ_i are a function of ϵ as defined in Lemma 14. In particular, $\chi_i = \chi_{i,\epsilon} = (2e/i \cdot \log(16/\epsilon))^{i/2}$ for $i \in \{2,3,4,6\}$. Observing that $256 \log(32/\epsilon) \geq \left(\chi_4 \sqrt{1680}\right)^{1/2}$, and that our choice of r satisfies

$$r^2 \leq \min\left\{\frac{\epsilon}{8\sqrt{1680}\chi_4}, \frac{\epsilon}{40\left(\chi_2\chi_6\sqrt{24}\sqrt{15120}\right)^{1/2}}, \frac{\epsilon}{204800\chi_2\sqrt{24}\log(32/\epsilon)}\right\}, \tag{117}$$

we obtain

$$\left|\sum_{i=1}^n d_{x,i}^2 s_{x,i}^2 \frac{1}{2} h^\top \nabla^2 \varphi_{y,i} h\right| \leq \frac{r^2}{\kappa^4 d^{3/2}}\left[\frac{\epsilon}{2} + \frac{\epsilon}{4} + \frac{\epsilon}{8}\left(\frac{\log n}{\sqrt{d}} + 1\right)\right].$$

Asserting the additional condition $\sqrt{d} \geq \log n$, yields the claim.

It is now left to prove the bounds (115a) and (115b). We prove these bounds separately.

Bounding C_2: Applying Cauchy-Schwarz inequality, we have

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^3 f_{y,i}\right| \leq \left(\sum_{i=1}^n \zeta_{y,i} f_{y,i}^2 \cdot \sum_{i=1}^n \zeta_{y,i} d_{y,i}^6\right)^{1/2}$$

Using the bound (108), we obtain

$$\sum_{i=1}^n \zeta_{y,i} f_{y,i}^2 \leq 4\kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 = 4\kappa^2 \sum_{i=1}^n \zeta_{x,i} d_{x,i}^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}.$$

Substituting $h = z - x$ where z is given by relation (82), we obtain that $d_{x,i} = \frac{r}{d^{3/4}\kappa} \hat{a}_{x,i}^\top \xi$, and thereby

$$\sum_{i=1}^n \zeta_{y,i} f_{y,i}^2 \leq 4\kappa^2 \frac{r^2}{d^{3/2}\kappa^4} \sum_{i=1}^n \zeta_{x,i} (\hat{a}_{x,i}^\top \xi)^2 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^2}{d_{x,i}^2}.$$

Doing similar algebra, we obtain $\sum_{i=1}^n \zeta_{y,i} d_{y,i}^6 = \frac{r^6}{d^{9/2}\kappa^{12}} \sum_{i=1}^n \zeta_{x,i} \left(\hat{a}_{x,i}^\top \xi\right)^6 \frac{\zeta_{y,i}}{\zeta_{x,i}} \frac{d_{y,i}^6}{d_{x,i}^6}$. Putting the pieces together yields the claim.

Bounding C_3: Recall that $\rho_y = (G_y - \alpha\Lambda_y)\ell_y$ (Lemma 9) and $\mu_y = (G_y - \alpha\Lambda_y)^{-1} G_y$ (Lemma 11). We have

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| = \mathbf{1} D_y^2 G_y \ell_y = \underbrace{\mathbf{1} D_y^2 G_y (G_y - \alpha\Lambda_y)^{-1}}_{=: v_y^\top} \underbrace{(G_y - \alpha\Lambda_y)\ell_y}_{\rho_y}.$$

Using the definition of v_y and μ_y, we obtain

$$v_{y,i} := e_i^\top v_y = e_i^\top (G_y - \alpha\Lambda_y)^{-1} G_y D_y^2 \mathbf{1} = e_i^\top \mu_y D_y^2 \mathbf{1} = \mu_{y,i,i} d_{y,i}^2 + \sum_{j \in [n], j \neq i} \mu_{y,i,j} d_{y,j}^2.$$

Consequently, we have

$$\left|\sum_{i=1}^n v_{y,i} \rho_{y,i}\right| \leq \overbrace{\sum_{i=1}^n |\rho_{y,i}| \cdot \left|\mu_{y,i,i} d_{y,i}^2\right|}^{=:C_4} + \overbrace{\sum_{i=1}^n |\rho_{y,i}| \cdot \left(\sum_{j \in [n], j \neq i} \left|\mu_{y,i,j} d_{y,j}^2\right|\right)}^{=:C_5}$$

From Lemma 11, we have that $\mu_{y,i,i} \in [0, \kappa]$. Hence, we have $C_4 \leq \|\rho_y\|_1 \cdot \kappa \cdot \max_{i \in [n]} d_{y,i}^2$. To bound C_5, we note that

$$\sum_{j \in [n], j \neq i} \left|\mu_{y,i,j} d_{y,j}^2\right| \overset{(i)}{\leq} \left(\sum_{j \in [n], j \neq i} \frac{\mu_{y,i,j}^2}{\zeta_{y,j}} \cdot \sum_{j=1}^n \zeta_{y,j} d_{y,j}^4\right)^{1/2} \overset{(ii)}{\leq} \left(\kappa^3 \cdot \sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2},$$

where step (i) follows from Cauchy-Schwarz inequality and step (ii) from Lemma 11. Putting the pieces together, we obtain that

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| \leq \|\rho_y\|_1 \cdot \left[\kappa \cdot \max_{i \in [n]} d_{y,i}^2 + \kappa^{3/2} \left(\sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2}\right].$$

Using the bound on $\|\rho_y\|_1$ from Lemma 9, we have

$$\left|\sum_{i=1}^n \zeta_{y,i} d_{y,i}^2 \ell_{y,i}\right| \leq \left(56\kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2\right) \cdot \left[\kappa \cdot \max_{i \in [n]} d_{y,i}^2 + \kappa^{3/2} \left(\sum_{j=1}^n \zeta_{x,j} d_{x,j}^4 \frac{\zeta_{y,j}}{\zeta_{x,j}} \frac{d_{y,j}^4}{d_{x,j}^4}\right)^{1/2}\right].$$

Substituting the expression for $d_{x,i} = \frac{r}{\kappa^2 d^{3/4}} \left(\hat{a}_{x,i}^\top \xi\right)$ yields the claim.

Appendix H. Proofs of Lemmas from Section E.1

In this section we collect proofs of lemmas from Section E.1. Each lemma is proved in a different subsection.

H.1 Proof of Lemma 9

Up to second order terms, we have

$$\frac{1}{s_{x+h,i}^2} = \frac{1}{s_{x,i}^2}\left[1 + \frac{2a_i^\top h}{s_{x,i}} + \frac{3(a_i^\top h)^2}{s_{x,i}^2}\right] + \mathcal{O}\left(\|h\|_2^3\right), \tag{118a}$$

$$\zeta_{y+h,i} = \zeta_{y,i} + h^\top \nabla \zeta_{y,i} + \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h + \mathcal{O}\left(\|h\|_2^3\right), \tag{118b}$$

$$\zeta_{y+h,i}^\alpha = \zeta_{y,i}^\alpha + \alpha \zeta_{y,i}^{\alpha-1}\left(h^\top \nabla \zeta_{y,i} + \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h\right) + \frac{\alpha(\alpha-1)}{2}\zeta_{y,i}^{\alpha-2}\left(h^\top \nabla \zeta_{y,i}\right)^2 + \mathcal{O}\left(\|h\|_2^3\right), \tag{118c}$$

Further, let

$$\tilde{J}_y := A_y^\top G_y^\alpha A_y = \sum_{i=1}^n \zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}. \tag{118d}$$

Using equations (118a) and (118c), and substituting $d_{y,i} = a_i^\top h / s_{y,i}$, $f_{y,i} = h^\top \nabla \zeta_{y,i}/\zeta_{y,i}$ and $\ell_{y,i} = \frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h / \zeta_{y,i}$, we find that

$$\tilde{J}_{y+h} = \sum_{i=1}^n \left[1 + \alpha f_{y,i} + \alpha \ell_{y,i} + \frac{\alpha(\alpha-1)}{2} f_{y,i}^2\right]\left[1 + 2d_{y,i} + 3d_{y,i}^2\right]\zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2} + \mathcal{O}\left(\|h\|_2^3\right).$$

Note that $d_{y,i}$ and $f_{y,i}$ are first order terms in $\|h\|_2$ and $\ell_{y,i}$ is a second order term in $\|h\|_2$.

Thus we obtain

$$\tilde{J}_{y+h} - \tilde{J}_y = \underbrace{\sum_{i=1}^n (2d_{y,i} + \alpha f_{y,i})\,\zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}}_{=:\Delta_{y,h}^{(1)}} + \underbrace{\sum_{i=1}^n \left[3d_{y,i}^2 + 2\alpha d_{y,i} f_{y,i} + \alpha \ell_{y,i} + \frac{\alpha(\alpha-1)}{2} f_{y,i}^2\right]\zeta_{y,i}^\alpha \frac{a_i a_i^\top}{s_{y,i}^2}}_{=:\Delta_{y,h}^{(2)}} + \mathcal{O}\left(\|h\|_2^3\right).$$

Let $\Delta_{y,h} := \Delta_{y,h}^{(1)} + \Delta_{y,h}^{(2)}$. Note that $\Delta_{y,h}^{(i)}$ denotes the i-th order term in $\|h\|_2$. Finally, the following expansion also comes in handy for our derivations:

$$a_i^T \tilde{J}_{y+h}^{-1} a_i = a_i^\top \tilde{J}_y^{-1} a_i - a_i^\top \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} a_i + a_i^\top \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} \Delta_{y,h} \tilde{J}_y^{-1} a_i + \mathcal{O}\left(\|h\|_2^3\right). \tag{118e}$$

H.1.1 Proof of part (a): Gradient of weights

The expression for the gradient $\nabla \zeta_{y,i}$ is derived in Lemma 14 of the paper (Lee and Sidford, 2014) and is thereby omitted.

H.1.2 Proof of part (b): Hessian of weights

We claim that

$$\begin{aligned}\rho_y = \left(\mathbb{I} - \alpha \Lambda_y G_y^{-1}\right)\begin{bmatrix} \frac{1}{2} h^\top \nabla^2 \zeta_{y,1} h \\ \cdots \\ \frac{1}{2} h^\top \nabla^2 \zeta_{y,m} h \end{bmatrix} &= (2D_y + \alpha F_y)\Upsilon_y^{(2)}(2D_y + \alpha F_y)\mathbf{1} \\ &+ \left(\Sigma_y - \Upsilon_y^{(2)}\right)\left[2\alpha D_y F_y + 3D_y^2 + \tau_\alpha F_y^2\right]\mathbf{1} \\ &+ \operatorname{diag}\left(\Upsilon_y (2D_y + \alpha F_y)\Upsilon_y (2D_y + \alpha F_y)\Upsilon_y\right),\end{aligned} \tag{119}$$

where we have used $\operatorname{diag}(B)$ to denote the diagonal vector $(B_{1,1}, \ldots, B_{n,n})$ of the matrix B. Deferring the proof of this expression for the moment, we now derive a bound on the ℓ_1 norm of ρ_y. Expanding the i-th term of $\rho_{y,i}$ from equation (119), we obtain

$$\begin{aligned}\rho_{y,i} = (2d_{y,i} + \alpha f_{y,i})\sum_{j=1}^n (2d_{y,j} + \alpha f_{y,j})\Upsilon_{y,i,j}^2 + \left[2\alpha d_{y,i} f_{y,i} + 3d_{y,i}^2 + \tau_\alpha f_{y,i}^2\right]\sigma_{y,i} \\ - \sum_{j=1}^n \left[2\alpha d_{y,j} f_{y,j} + 3d_{y,j}^2 + \tau_\alpha f_{y,j}^2\right]\Upsilon_{y,i,j}^2 + \sum_{j,l=1}^n (2d_{y,j} + \alpha f_{y,j})(2d_{y,l} + \alpha f_{y,l})\Upsilon_{y,i,j}\Upsilon_{y,j,l}\Upsilon_{y,l,i}.\end{aligned}$$

Recall that $\alpha = 1 - 1/\log_2(2n/d)$. Since $n \geq d$ for polytopes, we have $\alpha \in [0,1]$ and consequently $|\tau_\alpha| = |\alpha(\alpha-1)/2| \in [0,1]$. Further note that Υ_x is an orthogonal projection matrix, and hence we have

$$\operatorname{diag}(\Upsilon_x e_i)\Upsilon_x \operatorname{diag}(\Upsilon_x e_i) \preceq \operatorname{diag}(\Upsilon_x e_i)\operatorname{diag}(\Upsilon_x e_i).$$

Combining these observations with the AM-GM inequality, we have

$$|\rho_{y,i}| \leq 7\sigma_{y,i} d_{y,i}^2 + 3\sigma_{y,i} f_{y,i}^2 + \sum_{j=1}^n \left(13 d_{y,j}^2 + 6 f_{y,j}^2\right)\Upsilon_{y,i,j}^2.$$

Summing both sides over the index i, we find that

$$\sum_{i=1}^n |\rho_{y,i}| \overset{(i)}{\leq} \sum_{i=1}^n 20\sigma_{y,i} d_{y,i}^2 + 9\sigma_{y,i} f_{y,i}^2 \overset{(ii)}{\leq} \sum_{i=1}^n 20\zeta_{y,i} d_{y,i}^2 + 9\zeta_{y,i} f_{y,i}^2 \overset{(iii)}{\leq} 56\kappa^2 \sum_{i=1}^n \zeta_{y,i} d_{y,i}^2,$$

where step (i) follows from Lemma 7 (a), step (ii) from Lemma 3 (a) and step (iii) from the bound (108).

We now return to the proof of expression (119). Using equation (87c), we find that

$$\frac{1}{2} h^\top \nabla^2 \zeta_{y,i} h = \frac{1}{2} h^\top \nabla^2 \sigma_{y,i} h \quad \text{for all } i \in [n]. \tag{120}$$

Next, we derive the Taylor series expansion of $\sigma_{y,i}$. Using the definition of $\tilde{J}_x$ (118d) in equation (72), we find that $\sigma_{y,i} = \zeta_{y,i}^\alpha \frac{a_i^\top \tilde{J}_y^{-1} a_i}{s_{y,i}^2}$. To compute the difference $\sigma_{y+h,i} - \sigma_{y,i}$, we

use the expansions (118a), (118c) and (118e). Letting $\tau_\alpha = \alpha(\alpha-1)/2$, we have

$$\begin{aligned}
\sigma_{y+h,i} &= \zeta^\alpha_{y+h,i}\frac{a_i^\top \tilde{J}^{-1}_{y+h}a_i}{s^2_{y+h,i}} \\
&= \zeta^\alpha_{y,i}\frac{a_i^\top \tilde{J}^{-1}_{y+h}a_i}{s^2_{y,i}}\left[1+\alpha f_{y,i}+\alpha\ell_{y,i}+\tau_\alpha f_{y,i}^2\right]\left[1+2d_{y,i}+3d_{y,i}^2\right]+\mathcal{O}\left(\|h\|_2^3\right) \\
&= \sigma_{y,i}+(2d_{y,i}+\alpha f_{y,i})\sigma_{y,i}-\sum_{j=1}^n(2d_{y,j}+\alpha f_{y,j})\Upsilon^2_{y,i,j}+(2d_{y,i}+\alpha f_{y,i})\sum_{j=1}^n(2d_{y,j}+\alpha f_{y,j})\Upsilon^2_{y,i,j} \\
&\quad+2\alpha d_{y,i}f_{y,i}\sigma_{y,i}+\left[\alpha\ell_{y,i}+\tau_\alpha f_{y,i}^2+3d_{y,i}^2\right]\sigma_{y,i}-\sum_{j=1}^n\left[3d_{y,j}^2+2\alpha d_{y,j}f_{y,j}+\alpha\ell_{y,j}+\tau_\alpha f_{y,j}^2\right]\Upsilon^2_{y,i,j} \\
&\quad+\sum_{j,l=1}^n(2d_{y,j}+\alpha f_{y,j})(2d_{y,l}+\alpha f_{y,l})\Upsilon_{y,i,j}\Upsilon_{y,j,l}\Upsilon_{y,l,i}+\mathcal{O}\left(\|h\|_2^3\right).
\end{aligned}$$

We identify the second order (in $\mathcal{O}\left(\|h\|_2^2\right)$) terms in the previous expression. Using the equation (120), these are indeed the terms that correspond to the terms $\frac{1}{2}h^\top\nabla^2\zeta_{y,i}h$, $i\in[n]$. Substituting $\ell_{y,i}=\frac{1}{2}h^\top\nabla^2\zeta_{y,i}h/\zeta_{y,i}$, we have

$$\begin{aligned}
&\frac{1}{2}h^\top\nabla^2\zeta_{y,i}h \\
&=(2d_{y,i}+\alpha f_{y,i})\sum_{j=1}^n(2d_{y,j}+\alpha f_{y,j})\Upsilon^2_{y,i,j}+2\alpha d_{y,i}f_{y,i}\sigma_{y,i}+\left[\frac{\alpha}{2}\frac{h^\top\nabla^2\zeta_{y,i}h}{\zeta_{y,i}}+\tau_\alpha f_{y,i}^2+3d_{y,i}^2\right]\sigma_{y,i} \\
&-\sum_{j=1}^n\left[3d_{y,j}^2+2\alpha d_{y,j}f_{y,j}+\frac{\alpha}{2}\frac{h^\top\nabla^2\zeta_{y,j}h}{\zeta_{y,j}}+\tau_\alpha f_{y,j}^2\right]\Upsilon^2_{y,i,j}+\sum_{j,l=1}^n(2d_{y,j}+\alpha f_{y,j})(2d_{y,l}+\alpha f_{y,l})\Upsilon_{y,i,j}\Upsilon_{y,j,l}\Upsilon_{y,l,i}.
\end{aligned}$$

Collecting the different terms and doing some algebra yields the result (119).

H.1.3 Proof of part (c): Gradient of logdet

For a unit vector $h\in\mathbb{R}^d$, we have

$$h^\top\log\det J_y=\lim_{\delta\to0}\frac{1}{\delta}(\log\det J_{y+\delta h}-\log\det J_y)=\lim_{\delta\to0}\frac{1}{\delta}(\log\det J_y^{-1/2}J_{y+\delta h}J_y^{-1/2}-\log\det\mathbb{I}_d)$$

Let $\hat{a}_{y,i}:=J_{y,i}^{-1/2}a_i/s_{y,i}$ for each $i\in[n]$. Using the property $\log\det B=\operatorname{trace}\log B$, where $\log B$ denotes the logarithm of the matrix and that $\log\det\mathbb{I}_d=0$, we obtain

$$h^\top\log\det J_y=\lim_{\delta\to0}\frac{1}{\delta}\left[\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\zeta_{y+\delta h}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)\right],$$

where we have substituted $s_{y+\delta h,i}=s_{y,i}-\delta a_i^\top h$. Keeping track of first order terms in δ, and noting that $\sum_{i=1}^n\zeta_{y,i}\hat{a}_{y,i}\hat{a}_{y,i}^\top=\mathbb{I}_d$, we find that

$$\begin{aligned}
\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\zeta_{y+\delta h,i}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)&=\operatorname{trace}\log\left[\sum_{i=1}^n\left(\zeta_{y,i}+\delta h^\top\nabla\zeta_{y,i}\right)\left(1+\frac{2\delta a_i^\top h}{s_{y,i}}\right)\hat{a}_{y,i}\hat{a}_{y,i}^\top\right]+\mathcal{O}\left(\delta^2\right) \\
&=\operatorname{trace}\left[\sum_{i=1}^n\delta\left(\frac{2a_i^\top h}{s_{y,i}}+h^\top\nabla\zeta_{y,i}\right)\hat{a}_{y,i}\hat{a}_{y,i}^\top\right]+\mathcal{O}\left(\delta^2\right) \\
&=\sum_{i=1}^n\delta\left(\frac{2a_i^\top h}{s_{y,i}}+h^\top\nabla\zeta_{y,i}\right)\theta_{y,i}+\mathcal{O}\left(\delta^2\right)
\end{aligned}$$

where in the last step we have used the fact that $\operatorname{trace}(\hat{a}_{y,i}\hat{a}_{y,i}^\top)=\hat{a}_{y,i}^\top\hat{a}_{y,i}=\theta_{y,i}$ for each $i\in[n]$. Substituting the expression for $\nabla\zeta_y$ from part (a), and rearranging the terms yields the claimed expression in the limit $\delta\to0$.

H.1.4 Proof of part (d): Gradient of φ

Using the chain rule and the fact that $\nabla s_{y,i}=-a_i$, yields the result.

H.1.5 Proof of part (e)

We claim that

$$\frac{1}{2}h^\top\nabla^2\Psi_y h=\frac{1}{2}\left[\sum_{i=1}^n\zeta_{y,i}\theta_{y,i}(3d_{y,i}^2+2d_{y,i}f_{y,i}+\ell_{y,i})-\frac{1}{2}\sum_{i,j=1}^n\zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2\left(2d_{y,i}+f_{y,i}\right)\left(2d_{y,j}+f_{y,j}\right)\right].$$

The desired bound on $\left|h^\top\nabla^2\Psi_y h\right|/2$ now follows from an application of AM-GM inequality with Lemma 7(d).

We now derive the claimed expression for the directional Hessian of the function Ψ. We have

$$\begin{aligned}
&\frac{1}{2}h^\top\left(\nabla^2\log\det J_y\right)h=\lim_{\delta\to0}\frac{1}{2\delta^2}(\log\det J_y^{-1/2}J_{y+\delta h}J_y^{-1/2}+\log\det J_y^{-1/2}J_{y-\delta h}J_y^{-1/2}-2\log\det\mathbb{I}_d) \\
&=\frac{1}{2}\lim_{\delta\to0}\frac{1}{\delta^2}\left[\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\zeta_{y+\delta h}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)+\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\zeta_{y-\delta h}}{(1+\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)\right].
\end{aligned}$$

Expanding the first term in the above expression, we find that

$$\begin{aligned}
&\operatorname{trace}\log\left(\sum_{i=1}^n\frac{\zeta_{y+\delta h,i}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right) \\
&=\operatorname{trace}\log\underbrace{\left[\sum_{i=1}^n\left(\zeta_{y,i}+\delta h^\top\nabla\zeta_{y,i}+\frac{\delta^2}{2}h^\top\nabla^2\zeta_{y,i}h\right)\left(1+2\delta\frac{a_i^\top h}{s_{y,i}}+3\delta^2\frac{(a_i^\top h)^2}{s_{y,i}^2}\right)\hat{a}_{y,i}\hat{a}_{y,i}^\top\right]}_{=:\mathbb{I}_d+B}+\mathcal{O}\left(\delta^3\right).
\end{aligned}$$

Substituting the shorthand notation from equations (88a), (88b) and (88c), we have

$$B = \sum_{i=1}^{n} \zeta_{y,i} \left[\delta(2d_{y,i} + f_{y,i}) + \delta^2(3d_{y,i}^2 + 2d_{y,i}f_{y,i} + \ell_{y,i})\right] \hat{a}_{y,i}\hat{a}_{y,i}^\top + \mathcal{O}\left(\delta^3\right).$$

Now we make use of the following facts (1) $\operatorname{trace}\log(\mathbb{I}_d + B) = \operatorname{trace}\left[B - \frac{B^2}{2} + \mathcal{O}\left(\|B\|^3\right)\right]$, (2) for each $i, j \in [n]$, we have $\operatorname{trace}(\hat{a}_{y,i}\hat{a}_j^\top) = \hat{a}_{y,i}^\top \hat{a}_j = \theta_{y,i,j}$, and (3) for each $i \in [n]$, we have $\theta_{y,i,i} = \theta_{y,i}$. Thus we obtain

$$\begin{aligned}\operatorname{trace}\log\left(\sum_{i=1}^{n} \frac{\zeta_{y+\delta h,i}}{(1-\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right) &= \sum_{i=1}^{n} \zeta_{y,i}\theta_{y,i}\left[\delta(2d_{y,i} + f_{y,i}) + \delta^2(3d_{y,i}^2 + 2d_{y,i}f_{y,i} + \ell_{y,i})\right] \\ &\quad - \frac{1}{2}\sum_{i,j=1}^{n} \zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2\delta^2(2d_{y,i} + f_{y,i})(2d_{y,j} + f_{y,j}) + \mathcal{O}\left(\delta^3\right).\end{aligned}$$

Similarly, we can obtain an expression for $\operatorname{trace}\log\left(\sum_{i=1}^{n} \frac{\zeta_{y-\delta h}}{(1+\delta a_i^\top h/s_{y,i})}\hat{a}_{y,i}\hat{a}_{y,i}^\top\right)$. Putting the pieces together, we obtain

$$\frac{1}{2}h^\top\left(\nabla^2 \log\det J_y\right)h = \sum_{i=1}^{n} \zeta_{y,i}\theta_{y,i}(3d_{y,i}^2 + 2d_{y,i}f_{y,i} + \ell_{y,i}) - \frac{1}{2}\sum_{i,j=1}^{n} \zeta_{y,i}\zeta_{y,j}\theta_{y,i,j}^2(2d_{y,i} + f_{y,i})(2d_{y,j} + f_{y,j}). \tag{121}$$

H.1.6 Proof of part (f)

We claim that

$$\frac{1}{2}h^\top\nabla^2\varphi_{y,i}h = \varphi_{y,i}\left(2d_{y,i}f_{y,i} + 3d_{y,i}^2 + \ell_{y,i}\right). \tag{122}$$

The claim follows from a straightforward application of chain rule and substitution of the expressions for $\nabla\zeta_{y,i}$ and $\nabla^2\zeta_{y,i}$ in terms of the shorthand notation $d_{y,i}, f_{y,i}$ and $\ell_{y,i}$. Multiplying both sides of equation (122) with $d_{y,i}^2 s_{y,i}^2$ and summing over index i, we find that

$$\begin{aligned}\sum_{i=1}^{n} d_{y,i}^2 s_{y,i}^2 \frac{1}{2}h^\top\nabla\varphi_{y,i}^2 h = \sum_{i=1}^{n} d_{y,i}^2 s_{y,i}^2\varphi_{y,i}\left[\ell_{y,i} + 2d_{y,i}f_{y,i} + 3d_{y,i}^2\right] &= \sum_{i=1}^{n} d_{y,i}^2\zeta_{y,i}\left[\ell_{y,i} + 2d_{y,i}f_{y,i} + 3d_{y,i}^2\right] \\ &\le \sum_{i=1}^{n} d_{y,i}^2\zeta_{y,i}\left[\ell_{y,i} + f_{y,i}^2 + 4d_{y,i}^2\right],\end{aligned}$$

where in the last step we have used the AM-GM inequality. The claim follows.

H.2 Proof of Lemma 10

We claim that

$$0 \preceq G_y^{-1/2}\left(c_1\mathbb{I}_n + c_2\Lambda_y\left(G_y - \alpha\Lambda_y\right)^{-1}\right)G_y^{1/2} \preceq (c_1 + c_2)\,\kappa\mathbb{I}_n. \tag{123}$$

The proof of the lemma is immediate from this claim, as for any PSD matrix $H \le c\mathbb{I}_n$, we have $H^2 \le c^2\mathbb{I}_n$.

We now prove claim (123). Note that

$$G_y^{-1/2}\Lambda_y\left(G_y - \alpha\Lambda_y\right)^{-1}G_y^{1/2} = \underbrace{G_y^{-1/2}\Lambda_y G_y^{-1/2}}_{:=B_y}(\mathbb{I}_n - \alpha_\jmath G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1}. \tag{124}$$

Note that the RHS is equal to the matrix $B_y(\mathbb{I}_n - \alpha_\jmath B_y)^{-1}$ which is symmetric. Observe the following ordering of the matrices in the PSD cone

$$\Sigma_y + \beta_\jmath\mathbb{I}_n = G_y \succeq \Sigma_y \succeq \Lambda_y = \Sigma_y - \Upsilon_y^{(2)} \succeq 0.$$

For the last step we have used the fact that $\Sigma_y - \Upsilon_y^{(2)}$ is a diagonally dominant matrix with non negative entries on the diagonal to conclude that it is a PSD matrix. Consequently, we have

$$B_y = G_y^{-1/2}\Lambda_y G_y^{-1/2} \preceq \mathbb{I}_n. \tag{125}$$

Further, recall that $\alpha_\jmath = (1 - 1/\kappa) \Leftrightarrow \kappa = (1 - \alpha_\jmath)^{-1}$. As s result, we obtain

$$0 \preceq (\mathbb{I}_n - \alpha_\jmath G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1} \preceq \kappa\mathbb{I}_n.$$

Multiplying both sides by $B_y^{1/2}$ and using the relation (125), we obtain

$$0 \preceq B_y^{1/2}(\mathbb{I}_n - \alpha_\jmath G_y^{-1/2}\Lambda_y G_y^{-1/2})^{-1}B_y^{1/2} \preceq \kappa\mathbb{I}_n. \tag{126}$$

Using the fact that B_y commutes with $(\mathbb{I}_n - B_y)^{-1}$, we obtain $B_y(\mathbb{I}_n - \alpha_\jmath B_y)^{-1} \preceq \kappa\mathbb{I}_n$. Using observation (124) now completes the proof.

H.3 Proof of Lemma 11

Without loss of generality, we can first prove the result for $i = 1$. Let $\nu := \mu_y^\top e_1$ denote the first row of the matrix μ_y. Observe that

$$e_1 = \left(G_y - \alpha\Lambda_y\right)G_y^{-1}\nu = \nu - \alpha\Sigma_y G_y^{-1}\nu + \alpha\Upsilon_y^{(2)}G_y^{-1}\nu \tag{127}$$

We now prove bounds (92a) and (92b) separately.

Proof of bound (92a)**:** Multiplying the equation (127) on the left by $\nu^\top G_y^{-1}$, we obtain

$$\begin{aligned}g_1^{-1}\nu_1 &= \nu^\top G_y^{-1}\nu - \alpha\nu^\top G_y^{-1}\Sigma_y G_y^{-1}\nu + \alpha\nu^\top G_y^{-1}\Upsilon_y^{(2)}G_y^{-1}\nu \\ &\ge \nu^\top G_y^{-1}\nu - \alpha\nu^\top G_y^{-1}\Sigma_y G_y^{-1}\nu \\ &\ge \left(g_1^{-1} - \alpha\sigma_{y,1}/g_1^2\right)\nu_1^2.\end{aligned} \tag{128}$$

Rearranging terms, we obtain

$$0 \le \nu_1 \le \frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1}} \overset{(i)}{\le} \kappa, \tag{129}$$

where inequality (i) follows from the facts that $\zeta_{y,j} \ge \sigma_{y,j}$ and $(1 - \alpha) = \kappa$.

Proof of bound (92b)**:** In our proof, we use the following improved lower bound for the term $\mu_{y,1,1} = \nu_1$.

$$\nu_1 \geq \frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2}, \tag{130}$$

Deferring the proof of this claim at the moment, we now complete the proof.

We begin by deriving a weighted ℓ_2-norm bound for the vector $\tilde{\nu} = (\nu_2, \ldots, \nu_n)^\top$. Equation (128) implies

$$\zeta_{y,1}^{-1}\nu_1\left(1 - \nu_1 + \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\nu_1\right) \geq \sum_{j=2}^n \nu_j^2\left(\zeta_{y,j}^{-1} - \alpha\zeta_{y,j}^{-2}\sigma_{y,j}\right) \overset{(i)}{\geq} (1-\alpha)\sum_{j=2}^n \frac{\nu_j^2}{\zeta_{y,j}},$$

where step (i) follows from the fact that $\zeta_{y,i} \geq \sigma_{y,i}$. Now, we upper bound the expression on the left hand side of the above inequality using the upper (129) and lower (130) bounds on ν_1:

$$\begin{aligned}
\zeta_{y,1}^{-1}\nu_1\left(1 - \nu_1 + \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\nu_1\right) &\leq \zeta_{y,1}^{-1}\frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1}}\left(1 - \left(1 - \alpha\frac{\sigma_{y,1}}{\zeta_{y,1}}\right)\frac{\zeta_{y,1}}{\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2}\right) \\
&= \frac{\alpha\sigma_{y,1}^2}{(\zeta_{y,1} - \alpha\sigma_{y,1})\left(\zeta_{y,1} - \alpha\sigma_{y,1} + \alpha\sigma_{y,1}^2\right)} \\
&\leq \kappa^2,
\end{aligned}$$

where in the last step we have used the facts that $\zeta_{y,1} \geq \sigma_{y,1}$ and $(1-\alpha)^{-1} = \kappa$. Putting the pieces together, we obtain $\sum_{j=2}^n \nu_j^2\zeta_{y,j}^{-1} \leq \kappa^3$, which is equivalent to our claim (92b) for $i = 1$.

It remains to prove our earlier claim (130). Writing equation (127) separately for the first coordinate and for the rest of the coordinates, we obtain

$$1 = \left(1 - \alpha\sigma_{y,1}\zeta_{y,1}^{-1} + \alpha\sigma_{y,1,1}^2\zeta_{y,j}^{-1}\right)\nu_1 + \alpha\sum_{j=2}^n \sigma_{y,1,j}^2\zeta_{y,j}^{-1}\nu_j, \quad \text{and} \tag{131a}$$

$$0 = \left(\mathbb{I}_{n-1} - \alpha\Sigma_y' G_y'^{-1}\right)\begin{pmatrix}\nu_2\\ \vdots \\ \nu_n\end{pmatrix} + \alpha\Upsilon_y'^{(2)}G_y'^{-1}\begin{pmatrix}\nu_2\\ \vdots \\ \nu_n\end{pmatrix} + \alpha\zeta_{y,1}^{-1}\nu_1\begin{pmatrix}\sigma_{y,1,2}^2\\ \vdots \\ \sigma_{y,1,n}^2\end{pmatrix}, \tag{131b}$$

where G_y' (respectively Σ_y', $\Upsilon_y'^{(2)}$) denotes the principal minor of G_y (respectively Σ_y, $\Upsilon_y^{(2)}$) obtained by excluding the first column and the first row. Multiplying both sides of the equation (131b) from the left by $\left(\nu_2, \cdots, \nu_n\right) G_y'^{-1}$, we obtain

$$0 = \sum_{j=2}^n \underbrace{\frac{1}{\zeta_{y,j}}\left(1 - \frac{\alpha\sigma_{y,j}}{\zeta_{y,j}}\right)\nu_j^2}_{c_{y,j}} + \underbrace{\alpha\left(\nu_2, \cdots, \nu_n\right) G_y'^{-1}\Upsilon_y'^{(2)}G_y'^{-1}\begin{pmatrix}\nu_2\\ \vdots \\ \nu_n\end{pmatrix}}_{C_{y.2}} + \alpha\frac{\nu_1}{\zeta_{y,1}}\sum_{j=2}^n \frac{\sigma_{y,j}^2}{\zeta_{y,j}}\nu_j. \tag{132}$$

Observing that $\alpha \in [0,1]$ and $\zeta_{y,j} \geq \sigma_{y,j}$ for all $y \in \operatorname{int}(\mathcal{K})$ and $j \in [n]$, we obtain $c_{y,j} \geq 0$. Further, note that $G_y'^{-1}\Upsilon_y'^{(2)}G_y'^{-1}$ is a PSD matrix and hence we have that $C_{y,2} \geq 0$. Putting the pieces together, we have

$$\alpha\frac{\nu_1}{\zeta_{y,1}}\sum_{j=2}^n \frac{\sigma_{y,j}^2}{\zeta_{y,j}}\nu_j \leq 0.$$

Combining this inequality with equation (131a) yields the claim.

H.4 Proof of Corollary 12

Without loss of generality, we can prove the result for $i = 1$. Applying Cauchy-Schwarz inequality, we have

$$\|\nu\|_1 = \nu_1 + \sum_{j=2}^n |\nu_j| \leq \nu_1 + \sqrt{\sum_{j=2}^n \frac{\nu_j^2}{\zeta_{y,j}} \cdot \sum_{j=2}^n \zeta_{y,j}} \leq \kappa + \kappa^{3/2} \cdot \sqrt{1.5\, d} \leq 3\sqrt{d}\kappa^{3/2},$$

where to assert the last inequality we have used Lemma 11 and Lemma 3(c). The claim (93) follows. Further, noting that the infinity norm of a matrix is the ℓ_1-norm of its transpose, we obtain $\left\| \left(G_y - \alpha\Lambda_y\right)^{-1} G_y \right\|_\infty \leq 3\sqrt{d}\kappa^{3/2}$ as claimed.

Appendix I. Proof of Lemmas from Section E.2

In this section, we collect proofs of auxiliary lemmas from Section E.2.

I.1 Proof of Lemma 13

Using Lemma 8, and the relation (82) we have

$$\left(1 - \frac{s_{z,i}}{s_{x,i}}\right)^2 \leq 4\frac{r^2}{\kappa^4 d^{3/2}}\xi^\top\xi, \tag{133}$$

where $\xi \sim \mathcal{N}(0, \mathbb{I}_d)$. Define

$$\Delta_s := \max_{i\in[n],\ v\in\overline{xz}} \left|1 - \frac{s_{v,i}}{s_{x,i}}\right|. \tag{134}$$

Using the standard Gaussian tail bound, we observe that $\mathbb{P}_{\xi\sim\mathcal{N}(0,\mathbb{I}_n)}\left[\xi^\top\xi \geq d(1+\delta)\right] \leq 1 - \epsilon/4$ for $\delta = \sqrt{\frac{2}{d}}$. Plugging this bound in the inequality (133) and noting that for all $v \in \overline{xz}$ we have $\|v - x\|_{J_x} \leq \|z - x\|_{J_x}$, we obtain that

$$\mathbb{P}_{z\sim\mathcal{P}_x}\left[\Delta_s \leq \frac{2r^2(1 + \sqrt{2/d}\log(4/\epsilon)}{\kappa^4\sqrt{d}}\right] \geq 1 - \epsilon/4.$$

Setting

$$r \leq 1/(25\sqrt{1 + \sqrt{2}\log(4/\epsilon)}), \tag{135}$$

and noting that $\kappa^4\sqrt{d} \geq 1$ implies the claim (94a). Hence, we obtain that $\Delta_s < .005/\kappa^2$ and consequently $\max_{i\in[n],v\in\overline{xz}} s_{x,i}/s_{v,i} \in (0.99, 1.01)$ with probability at least $1-\epsilon/4$.

We now claim that

$$\max_{i\in[n],v\in\overline{xz}} \frac{\zeta_{x,i}}{\zeta_{v,i}} \in \left[1-24\kappa^2\Delta_s, 1+24\kappa^2\Delta_s\right], \quad \text{if } \Delta_s \leq \frac{1}{32\kappa^2}.$$

The result follows immediately from this claim. To prove the claim, note that equation (98) implies that if $\Delta_s \leq \frac{1}{32\kappa^2}$, then

$$\frac{\zeta_{v,i}}{\zeta_{x,i}} \in (e^{-8\kappa^2\Delta_s}, e^{8\kappa^2\Delta_s}) \quad \text{for all } i\in[n] \text{ and } v\in\overline{xz},$$

which implies that

$$\max_{i\in[n],v\in\overline{xz}} \frac{\zeta_{x,i}}{\zeta_{v,i}} \in (e^{-8\kappa^2\Delta_s}, e^{8\kappa^2\Delta_s}).$$

Asserting the facts that $e^x \leq 1+3x$ and $e^{-x} \geq 1-3x$, for all $x\in[0,1]$ yields the claim.

I.2 Proof of Lemma 14

The proof once again makes use of the classical tail bounds for polynomials in Gaussian random variables. We restate the classical result stated in equation (136) for convenience. For any $d \geq 1$, any polynomial $P:\mathbb{R}^d \to \mathbb{R}$ of degree k, and any $t \geq (2e)^{k/2}$, we have

$$\mathbb{P}\left[|P(\xi)| \geq t\left(\mathbb{E}P(\xi)^2\right)^{\frac{1}{2}}\right] \leq \exp\left(-\frac{k}{2e}t^{2/k}\right), \tag{136}$$

where $\xi \sim \mathcal{N}(0,\mathbb{I}_n)$ denotes a standard Gaussian vector in n dimensions.

Recall the notation from equation (90) and observe that

$$\|\hat{a}_{x,i}\|_2^2 = \theta_{x,i}, \quad \text{and} \quad \hat{a}_{x,i}^\top \hat{a}_{x,j} = \theta_{x,i,j}. \tag{137}$$

We also have

$$\sum_{i=1}^n \zeta_{x,i}\hat{a}_{x,i}\hat{a}_{x,i}^\top = J_x^{-1/2}\sum_{i=1}^n \zeta_{x,i}\frac{a_i a_i^\top}{s_{x,i}^2}J_x^{-1/2} = \mathbb{I}_d. \tag{138}$$

Further, using Lemma 10 we obtain

$$\sum_{i=1}^n \zeta_{x,i}\hat{b}_{x,i}\hat{b}_{x,i}^\top = J_x^{-1/2}A_x\Lambda_x\left(G_x-\alpha\Lambda_x\right)^{-1}G_x\left(G_x-\alpha\Lambda_x\right)^{-1}\Lambda_x A_x^\top J_x^{-1/2} = 4\kappa^2\mathbb{I}_d. \tag{139}$$

Throughout this section, we consider a fixed point $x\in\operatorname{int}(\mathcal{K})$. For brevity in our notation, we drop the dependence on x for terms like $\zeta_{x,i}, \theta_{x,i}, \hat{a}_{x,i}$ (etc.) and denote them simply by $\zeta_i, \theta_i, \hat{a}_i$ respectively.

We introduce some matrices and vectors that would come in handy for our proofs.

$$B = \begin{bmatrix}\sqrt{\zeta_1}\hat{a}_1^\top \\ \vdots \\ \sqrt{\zeta_n}\hat{a}_n^\top\end{bmatrix}, \quad B_b = \begin{bmatrix}\sqrt{\zeta_1}\hat{b}_1^\top \\ \vdots \\ \sqrt{\zeta_n}\hat{b}_n^\top\end{bmatrix}, \quad v = \begin{bmatrix}\sqrt{\zeta_1}\|\hat{a}_1\|_2^2 \\ \vdots \\ \sqrt{\zeta_n}\|\hat{a}_n\|_2^2\end{bmatrix}, \quad \text{and} \quad v^{ab} = \begin{bmatrix}\sqrt{\zeta_1}\hat{a}_1^\top\hat{b}_1 \\ \vdots \\ \sqrt{\zeta_n}\hat{a}_n^\top\hat{b}_n\end{bmatrix}. \tag{140}$$

We claim that

$$BB^\top \preceq \mathbb{I}_n, \quad \text{and} \quad B_bB_b^\top \preceq 4\kappa^2\mathbb{I}_n. \tag{141a}$$

To see these claims, note that equation (138) implies that $B^\top B = \mathbb{I}_d$ and consequently, $BB^\top$ is an orthogonal projection matrix and $BB^\top \preceq \mathbb{I}_n$. Next, note that from equation (139) we have that $B_b^\top B_b \preceq \kappa^2\mathbb{I}_d$, which implies that $B_bB_b^\top \preceq \kappa^2\mathbb{I}_n$. In asserting both these arguments, we have used the fact that for any matrix B, the matrices $BB^\top$ and $B^\top B$ are PSD and have same set of eigenvalues.

Next, we bound the ℓ_2 norm of the vectors v and v^{ab}:

$$\|v\|_2^2 = \sum_{i=1}^n \zeta_i\theta_i^2 \overset{\text{Lem. 7 } (e)}{\leq} 4d, \quad \text{and} \tag{141b}$$

$$\left\|v^{ab}\right\|_2^2 = \sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\hat{b}_i\right)^2 \leq \sum_{i=1}^n \zeta_i\|\hat{a}_i\|_2^2\left\|\hat{b}_i\right\|_2^2 \leq 4\sum_{i=1}^n \zeta_i\left\|\hat{b}_i\right\|_2^2 = 4\operatorname{trace}(B_b^\top B_b) \overset{\text{eqn. (141a)}}{\leq} 16\kappa^2 d. \tag{141c}$$

We now prove the five claims of the lemma separately.

I.2.1 Proof of bound (95a)

Using Isserlis theorem (Isserlis, 1918) for fourth order Gaussian moments, we have

$$\mathbb{E}\left(\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^2\right)^2 = \sum_{i,j=1}^n \zeta_i\zeta_j\left(\|\hat{a}_i\|_2^2\|\hat{a}_j\|_2^2 + 2\left(\hat{a}_i^\top\hat{a}_j\right)^2\right) = \sum_{i,j=1}^n \zeta_i\zeta_j\left(\theta_i\theta_j + 2\theta_{i,j}^2\right) \leq 24d^2,$$

where the last follows from Lemma 7. Applying the bound (136) with $k=2$ and $t = e\log(\frac{16}{\epsilon})$. Note that the bound is valid since $t \geq (2e)$ for all $\epsilon\in(0,1/30]$.

I.2.2 Proof of bound (95b)

Applying Isserlis theorem for Gaussian moments, we obtain

$$\mathbb{E}\left(\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^3\right)^2 = 9\underbrace{\sum_{i,j=1}^n \zeta_i\zeta_j\|\hat{a}_i\|_2^2\|\hat{a}_j\|_2^2\left(\hat{a}_i^\top\hat{a}_j\right)}_{=:N_1} + 6\underbrace{\sum_{i,j=1}^n \zeta_i\zeta_j\left(\hat{a}_i^\top\hat{a}_j\right)^3}_{=:N_2}.$$

We claim that $N_1 \leq 4d$ and $N_2 \leq 4d$. Assuming these claims as given at the moment, we now complete the proof. We have $\mathbb{E}\left(\sum_{i=1}^n \zeta_i\left(\hat{a}_i^\top\xi\right)^3\right)^2 \leq 60d$. Applying the bound (136)

with $k = 3$ and $t = \left(\frac{2e}{3}\log\left(\frac{16}{\epsilon}\right)\right)^{3/2}$, and verifying that $t \geq (2e)^{3/2}$ for $\epsilon \in (0, 1/30]$ yields the claim.

We now turn to prove the bounds on N_1 and N_2. We have

$$N_1 = \sum_{i,j=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \hat{a}_i^\top \zeta_j \|\hat{a}_j\|_2^2 \hat{a}_j = \left\| \sum_{i=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \hat{a}_i \right\|_2^2 = \left\| B^\top v \right\|_2^2 \overset{\text{eqn. (141a)}}{\leq} \|v\|_2^2 \overset{\text{eqn. (141b)}}{\leq} 4d.$$

Next, applying Cauchy-Schwarz inequality and using equation (137), we obtain

$$N_2 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^3 \leq \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_{i,j}^2 \sqrt{\theta_i \theta_j} \overset{\text{(Lem. 3 (d))}}{\leq} 4 \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_{i,j}^2 \overset{\text{(Lem. 7 (d))}}{\leq} 4 \sum_{i=1}^{n} \zeta_i \theta_i \quad = 4d.$$

I.2.3 Proof of bound (95c)

Using Isserlis theorem for Gaussian moments, we have

$$\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \left(\hat{b}_{x,i}^\top \xi\right)\right)^2 = \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \|\hat{a}_i\|_2^2 \|\hat{a}_j\|_2^2 \left(\hat{b}_i^\top \hat{b}_j\right)}_{:=N_3} + 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_i\right) \left(\hat{a}_j^\top \hat{b}_j\right)}_{:=N_4}$$
$$+ 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \|\hat{a}_i\|_2^2 \left(\hat{b}_i^\top \hat{a}_j\right) \left(\hat{a}_j^\top \hat{b}_j\right)}_{:=N_5} + 2 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\hat{b}_i^\top \hat{b}_j\right)}_{:=N_6} + 4 \underbrace{\sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_j\right) \left(\hat{b}_i^\top \hat{a}_j\right)}_{:=N_7}$$

We claim that all terms $N_k \leq 16\kappa^2 d, k \in \{3, 4, 5, 6, 7\}$. Putting the pieces together, we have

$$\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^2 \left(\hat{b}_{x,i}^\top \xi\right)\right)^2 \leq 240\kappa^2 d.$$

Applying the bound (136) with $k = 3$ and $t = \left(\frac{2e}{3}\log\left(\frac{16}{\epsilon}\right)\right)^{3/2}$ yields the claim. Note that for the given definition of t, we have $t \geq (2e)^{3/2}$ for $\epsilon \in (0, 1/30]$ so that the bound (136) is valid.

It is now left to prove the bounds on N_k for $k \in \{3, 4, 5, 6, 7\}$. We have

$$N_3 = \sum_{i,j=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \hat{b}_i^\top \zeta_j \|\hat{a}_j\|_2^2 \hat{b}_j = \left\| \sum_{i=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \hat{b}_i \right\|_2^2 = \left\| B_b^\top v \right\|_2^2 \overset{\text{eqn. (141a)}}{\leq} 4\kappa^2 \|v\|_2^2 \overset{\text{eqn. (141b)}}{=} 16\kappa^2 d,$$

$$N_4 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right) \left(\hat{a}_i^\top \hat{b}_i\right) \left(\hat{a}_j^\top \hat{b}_j\right) = \left\| B^\top v^{ab} \right\|_2^2 \overset{\text{eqn. (141a)}}{\leq} \left\| v^{ab} \right\|_2^2 \overset{\text{eqn. (141c)}}{\leq} 16\kappa^2 d, \quad \text{and}$$

$$N_5 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \|\hat{a}_i\|_2^2 \left(\hat{b}_i^\top \hat{a}_j\right) \left(\hat{a}_j^\top \hat{b}_j\right) = \left(B^\top v^{ab}\right)^\top \left(B_b^\top v\right) \overset{\text{C–S}}{\leq} \left\| B^\top v^{ab} \right\|_2 \left\| B_b^\top v \right\|_2 \leq 16\kappa^2 d.$$

For the term N_6, we have

$$\begin{aligned} N_6 = \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\hat{b}_i^\top \hat{b}_j\right) &\overset{\text{(C–S)}}{\leq} \frac{1}{2} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left(\left\|\hat{b}_i\right\|_2^2 + \left\|\hat{b}_j\right\|_2^2\right) \\ &\overset{\text{(symm in } i,j)}{=} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\hat{a}_i^\top \hat{a}_j\right)^2 \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(eqn. (138))}}{\leq} \sum_{i=1}^{n} \zeta_i \|\hat{a}_i\|_2^2 \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(Lem. 3(d))}}{\leq} 4 \sum_{i=1}^{n} \zeta_i \left\|\hat{b}_i\right\|_2^2 \\ &\overset{\text{(eqn. (141c))}}{\leq} 16\kappa^2 d. \end{aligned}$$

The bound on the term N_7 can be obtained in a similar fashion.

I.2.4 Proof of bound (95d)

Observe that $\hat{a}_i^\top \xi \sim \mathcal{N}(0, \theta_i)$ and hence $\mathbb{E}\left(\hat{a}_i^\top \xi\right)^8 = 105\,\theta_i^4$. Thus, we have

$$\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^4\right)^2 \overset{\text{C–S}}{\leq} \sum_{i,j=1}^{n} \zeta_i \zeta_j \left(\mathbb{E}\left(\hat{a}_i^\top \xi\right)^8\right)^{\frac{1}{2}} \left(\mathbb{E}\left(\hat{a}_j^\top \xi\right)^8\right)^{\frac{1}{2}} = 105 \sum_{i,j=1}^{n} \zeta_i \zeta_j \theta_i^2 \theta_j^2 = 105 \left(\sum_{i=1}^{n} \zeta_i \theta_i^2\right)^2.$$

Now applying Lemma 7, we obtain that $\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^4\right)^2 \leq 1680d^2$. Consequently, applying the bound (136) with $k = 4$ and $t = \left(\frac{e}{2}\log\left(\frac{16}{\epsilon}\right)\right)^2$ and noting that $t \geq (2e)^2$ for $\epsilon \in (0, 1/30]$, yields the claim.

I.2.5 Proof of bound (95e)

Using the fact that $\mathbb{E}\left(\hat{a}_i^\top \xi\right)^{12} = 945\,\theta_i^6$ and an argument similar to the previous part yields that $\mathbb{E}\left(\sum_{i=1}^{n} \zeta_i \left(\hat{a}_i^\top \xi\right)^6\right)^2 \leq 15120d^2$.

Finally, applying the bound (136) with $k = 6$ and $t = \left(\frac{e}{3}\log\left(\frac{16}{\epsilon}\right)\right)^3$, and verifying that $t \geq (2e)^3$ for $\epsilon \in (0, 1/30]$, yields the claim.

Appendix J. Proof of Lovász's Lemma

We begin by formally defining the conductance (Φ) of a Markov chain on $(\mathcal{K}, \mathbb{B}(\mathcal{K}))$ with arbitrary transition operator $\mathcal{T}$ and stationary distribution π^*. We assume that the operator $\mathcal{T}$ is lazy and thereby the stationary distribution π^* is unique. Let $\mathbb{T}_x = \mathcal{T}(\delta_x)$ denote the transition distribution at point x, then the conductance Φ is defined as

$$\Phi := \inf_{\substack{\mathcal{S} \in \mathbb{B}(\mathcal{K}) \\ \pi^*(\mathcal{S}) \in (0, 1/2)}} \frac{\Phi(\mathcal{S})}{\pi^*(\mathcal{S})} \quad \text{where} \quad \Phi(\mathcal{S}) := \int_{\mathcal{S}} \mathbb{T}_u(\mathcal{K} \cap \mathcal{S}^c) d\pi^*(u) \quad \text{for any } \mathcal{S} \subseteq \mathcal{K}.$$

The conductance denotes the measure of the flow from a set to its complement relative to its own measure, when initialized in the stationary distribution. If the conductance is high, the following result shows that the Markov chain mixes fast.

Lemma 15 *(Lovász and Simonovits, 1993, Theorem 1.4) For any M-warm start μ_0, the mixing time of the Markov chain with conductance Φ is bounded as*

$$\left\|\mathcal{T}^k(\mu_0) - \pi^*\right\|_{TV} \leq \sqrt{M}\left(1 - \frac{\Phi^2}{2}\right)^k \leq \sqrt{M}\exp\left(-k\frac{\Phi^2}{2}\right).$$

Note that this result holds for a general distribution π^* although we apply for uniform π^*. The result can be derived from Cheeger's inequality for continuous-space discrete-time Markov chain and elementary results in Calculus. See, e.g., Theorem 1.4 and Corollary 1.5 by Lovász and Simonovits (1993) for a proof. For ease in notation define $\mathcal{K}\backslash\mathcal{S} := \mathcal{K} \cap \mathcal{S}^c$. We now state a key isoperimetric inequality.

Lemma 16 *(Lovász, 1999, Theorem 6) For any measurable sets $\mathcal{S}_1, \mathcal{S}_2 \subseteq \mathcal{K}$, we have*

$$\mathrm{vol}(\mathcal{K}\backslash\mathcal{S}_1\backslash\mathcal{S}_2) \cdot \mathrm{vol}(\mathcal{K}) \geq d_{\mathcal{K}}(\mathcal{S}_1, \mathcal{S}_2) \cdot \mathrm{vol}(\mathcal{S}_1) \cdot \mathrm{vol}(\mathcal{S}_2),$$

where $d_{\mathcal{K}}(\mathcal{S}_1, \mathcal{S}_2) := \inf_{x\in\mathcal{S}_1, y\in\mathcal{S}_2} d_{\mathcal{K}}(x, y)$.

Since π^* is the uniform measure on $\mathcal{K}$, this lemma implies that

$$\pi^*(\mathcal{K}\backslash\mathcal{S}_1\backslash\mathcal{S}_2) \geq d_{\mathcal{K}}(\mathcal{S}_1, \mathcal{S}_2) \cdot \pi^*(\mathcal{S}_1) \cdot \pi^*(\mathcal{S}_2). \tag{142}$$

In fact, such an inequality holds for an arbitrary log-concave distribution (Lovász and Vempala, 2003). In words, the inequality says that for a bounded convex set any two subsets which are far apart, can not have a large volume. Taking these lemmas as given, we now complete the proof.

Proof of (Lovász's) Lemma 6: We first bound the conductance of the Markov chain using the assumptions of the lemma. From Lemma 15, we see that the Markov chain mixes fast if all the sets $\mathcal{S}$ have a high conductance $\Phi(\mathcal{S})$. We claim that

$$\Phi \geq \frac{\rho\Delta}{64}, \tag{143}$$

from which the proof follows by applying Lemma 15. We now prove the claim (143) along the lines of Theorem 11 in the paper by Lovász (1999). In particular, we show that under the assumptions in the lemma, the sets with bad conductance are far apart and thereby have a small measure under π^*, whence the ratio $\Phi(\mathcal{S})/\pi^*(\mathcal{S})$ is not arbitrarily small. Consider a partition $\mathcal{S}_1, \mathcal{S}_2$ of the set $\mathcal{K}$ such that $\mathcal{S}_1$ and $\mathcal{S}_2$ are measurable. To prove claim (143), it suffices to show that

$$\frac{1}{\mathrm{vol}(\mathcal{K})}\int_{\mathcal{S}_1} \mathcal{T}_u(\mathcal{S}_2)du \geq \frac{\rho\Delta}{64} \cdot \min\left\{\pi^*(\mathcal{S}_1), \pi^*(\mathcal{S}_2)\right\}, \tag{144}$$

Define the sets

$$\mathcal{S}_1' := \left\{u \in \mathcal{S}_1 \middle| \tilde{\mathcal{T}}_u(\mathcal{S}_2) < \frac{\rho}{2}\right\}, \quad \mathcal{S}_2' := \left\{v \in \mathcal{S}_2 \middle| \tilde{\mathcal{T}}_v(\mathcal{S}_1) < \frac{\rho}{2}\right\}, \quad \text{and} \quad \mathcal{S}_3' := \mathcal{K}\backslash\mathcal{S}_1'\backslash\mathcal{S}_2'. \tag{145}$$

Case 1: If we have $\mathrm{vol}(\mathcal{S}_1') \leq \mathrm{vol}(\mathcal{S}_1)/2$ and consequently $\mathrm{vol}(\mathcal{K}\backslash\mathcal{S}_1') \geq \mathrm{vol}(\mathcal{S}_1)/2$, then

$$\int_{\mathcal{S}_1} \mathcal{T}_u(\mathcal{S}_2)du \overset{(i)}{\geq} \frac{1}{2}\int_{\mathcal{S}_1\backslash\mathcal{S}_1'} \tilde{\mathcal{T}}_u(\mathcal{S}_2)du \overset{(ii)}{\geq} \frac{\rho}{4}\mathrm{vol}(\mathcal{S}_1) \overset{(iii)}{\geq} \frac{\rho\Delta}{4} \cdot \min\left\{\mathrm{vol}(\mathcal{S}_1), \mathrm{vol}(\mathcal{S}_2)\right\},$$

which implies the inequality (144) since π^* is the uniform measure on $\mathcal{K}$. In the above sequence of inequalities, step (i) follows from the definition of the kernel $\mathcal{T}$, step (ii) follows from the definition of the set $\mathcal{S}_1'$ (145) and step (iii) from the fact that $\Delta < 1$. Dividing both sides by $\mathrm{vol}(\mathcal{K})$ yields the inequality (144) and we are done.

Case 2: It remains to establish the inequality (144) for the case when $\mathrm{vol}(\mathcal{S}_i') \geq \mathrm{vol}(\mathcal{S}_i)/2$ for each $i \in \{1, 2\}$. Now for any $u \in \mathcal{S}_1'$ and $v \in \mathcal{S}_2'$ we have

$$\left\|\tilde{\mathcal{T}}_u - \tilde{\mathcal{T}}_v\right\|_{\mathrm{TV}} \geq \tilde{\mathcal{T}}_u(\mathcal{S}_1) - \tilde{\mathcal{T}}_v(\mathcal{S}_1) = 1 - \tilde{\mathcal{T}}_u(\mathcal{S}_2) - \tilde{\mathcal{T}}_v(\mathcal{S}_1) > 1 - \rho,$$

and hence by assumption we have $d_{\mathcal{K}}(\mathcal{S}_1', \mathcal{S}_2') \geq \Delta$. Applying Lemma 16 and the definition of $\mathcal{S}_3'$ (145) we find that

$$\mathrm{vol}(\mathcal{S}_3') \cdot \mathrm{vol}(\mathcal{K}) \geq \Delta \cdot \mathrm{vol}(\mathcal{S}_1') \cdot \mathrm{vol}(\mathcal{S}_2') \geq \frac{\Delta}{4} \cdot \mathrm{vol}(\mathcal{S}_1) \cdot \mathrm{vol}(\mathcal{S}_2). \tag{146}$$

Using this inequality and the fact that for any $x \in [0, 1]$ we have $x(1-x) \geq \min\left\{x, (1-x)\right\}/2$ we obtain that

$$\pi^*(\mathcal{S}_3') \geq \frac{\Delta}{4} \cdot \pi^*(\mathcal{S}_1) \cdot \pi^*(\mathcal{S}_2) \geq \frac{\Delta}{8}\min\left\{\pi^*(\mathcal{S}_1), \pi^*(\mathcal{S}_2)\right\}. \tag{147}$$

We claim that

$$\int_{\mathcal{S}_1} \mathcal{T}_u(\mathcal{S}_2)du = \int_{\mathcal{S}_2} \mathcal{T}_v(\mathcal{S}_1)dv. \tag{148}$$

Assuming the claim as given, we now complete the proof. Using the equation (148), we have

$$\begin{aligned}
\frac{1}{\mathrm{vol}(\mathcal{K})}\int_{\mathcal{S}_1} \mathcal{T}_u(\mathcal{S}_2)du &= \frac{1}{2\,\mathrm{vol}(\mathcal{K})}\left(\int_{\mathcal{S}_1} \mathcal{T}_u(\mathcal{S}_2)du + \int_{\mathcal{S}_2} \mathcal{T}_v(\mathcal{S}_1)dv\right) \\
&\overset{(i)}{\geq} \frac{1}{2\,\mathrm{vol}(\mathcal{K})}\left(\frac{1}{2}\int_{\mathcal{S}_1\backslash\mathcal{S}_1'} \tilde{\mathcal{T}}_u(\mathcal{S}_2)du + \frac{1}{2}\int_{\mathcal{S}_2\backslash\mathcal{S}_2'} \tilde{\mathcal{T}}_v(\mathcal{S}_2)dv\right) \\
&\overset{(ii)}{\geq} \frac{\rho}{8}\frac{\mathrm{vol}(\mathcal{S}_3')}{\mathrm{vol}(\mathcal{K})} \\
&\overset{(iii)}{\geq} \frac{\rho\Delta}{64}\min\left\{\pi^*(\mathcal{S}_1), \pi^*(\mathcal{S}_2)\right\},
\end{aligned}$$

where step (i) follows from the definition of the kernel $\mathcal{T}$, step (ii) follows from the definition of the set $\mathcal{S}_3'$ (145) and step (iii) follows from the inequality (147). Putting together the pieces yields the claim (143).

It remains to prove the claim (148). We make use of the following result

$$\Phi(\mathcal{S}) = \Phi(\mathcal{K}\backslash\mathcal{S}) \quad \text{for any measurable } \mathcal{S} \subseteq \mathcal{K}. \tag{149}$$

Using equation (149) and noting that $\mathcal{S}_1 = \mathcal{K}\backslash\mathcal{S}_2$, we have

$$\frac{1}{\text{vol}(\mathcal{K})}\int_{\mathcal{S}_1}\mathcal{T}_u(\mathcal{S}_2)du = \int_{\mathcal{S}_1}\mathcal{T}_u(\mathcal{S}_2)\pi^*(u)du = \Phi(\mathcal{S}_1) = \Phi(\mathcal{K}\backslash\mathcal{S}_1) = \frac{1}{\text{vol}(\mathcal{K})}\int_{\mathcal{S}_2}\mathcal{T}_v(\mathcal{S}_1)dv,$$

which yields equation (148).

Proof of result (149)**:** Note that $\int_{\mathcal{K}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) = \pi^*(\mathcal{S})$. Thus, we have

$$\Phi(\mathcal{K}\backslash\mathcal{S}) = \int_{\mathcal{K}\backslash\mathcal{S}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) = \int_{\mathcal{K}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) - \int_{\mathcal{S}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) = \pi^*(\mathcal{S}) - \int_{\mathcal{S}}\mathcal{T}_u(\mathcal{S})d\pi^*(u).$$

Using the fact that $1 - \mathcal{T}_u(\mathcal{S}) = \mathcal{T}_u(\mathcal{K}\backslash\mathcal{S})$, we obtain

$$\pi^*(\mathcal{S}) - \int_{\mathcal{S}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) = \int_{\mathcal{S}}d\pi^*(u) - \int_{\mathcal{S}}\mathcal{T}_u(\mathcal{S})d\pi^*(u) = \int_{\mathcal{S}}\mathcal{T}_u(\mathcal{K}\backslash\mathcal{S})d\pi^*(u) = \Phi(\mathcal{S}),$$

thereby yielding the claim (149).

References

Kurt M Anstreicher. The volumetric barrier for semidefinite programming. *Mathematics of Operations Research*, 25(3):365–380, 2000.

Claude J. P. Bélisle, H. Edwin Romeijn, and Robert L. Smith. Hit-and-run algorithms for generating multivariate distributions. *Mathematics of Operations Research*, 18(2): 255–266, 1993.

Dimitris Bertsimas and Santosh Vempala. Solving convex programs by random walks. *Journal of the ACM (JACM)*, 51(4):540–556, 2004.

Rajendra Bhatia. *Matrix Analysis*, volume 169. Springer Science & Business Media, 2013.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

Pierre Brémaud. *Markov chains, Gibbs fields, Monte Carlo simulation, and queues*. Springer, 1991.

Steve Brooks, Andrew Gelman, Galin L Jones, and Xiao-Li Meng. *Handbook of Markov Chain Monte Carlo*. Chapman and Hall/CRC, 2011.

Peter J Bushell. Hilbert's metric and positive contraction mappings in a Banach space. *Archive for Rational Mechanics and Analysis*, 52(4):330–338, 1973.

Ben Cousins and Santosh Vempala. A cubic algorithm for computing Gaussian volume. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1215–1228. Society for Industrial and Applied Mathematics, 2014.

I. Dikin. Iterative solution to problems of linear and quadratic programming. *Doklady Akademii Nauk SSSR*, 174(4):747, 1967.

Jon Feldman, Martin J Wainwright, and David R Karger. Using linear programming to decode binary linear codes. *IEEE Transactions on Information Theory*, 51(3):954–972, 2005.

Stuart Geman and David Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. PAMI*, 6:721–741, 1984.

Adan Gustafson and Hariharan Narayanan. John's walk. *arXiv preprint arXiv:1803.02032*, 2018.

W. Keith Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

Roger A. Horn and Charles R. Johnson. *Matrix Analysis*. Cambridge University Press, 2012.

Kuo-Ling Huang and Sanjay Mehrotra. An empirical evaluation of walk-and-round heuristics for mixed integer linear programs. *Computational Optimization and Applications*, 55 (3):545–570, 2013.

Kuo-Ling Huang and Sanjay Mehrotra. An empirical evaluation of a walk-relax-round heuristic for mixed integer convex programs. *Computational Optimization and Applications*, 60(3):559–585, 2015.

Leon Isserlis. On a formula for the product-moment coefficient of any order of a normal frequency distribution in any number of variables. *Biometrika*, 12(1/2):134–139, 1918.

Svante Janson. *Gaussian Hilbert Spaces*, volume 129. Cambridge University Press, 1997.

Fritz John. Extremum problems with inequalities as subsidiary conditions. In O.E. Neugebauer In K. O. Friedrichs and J. J. Stoker, editors, *Studies and Essays: Courant Anniversary Volume*, pages 187–204. Wiley-Interscience, New York, 1948.

Ravi Kannan, László Lovász, and Miklós Simonovits. Random walks and an o*(n5) volume algorithm for convex bodies. *Random Structures & Algorithms*, 11(1):1–50, 1997.

Ravi Kannan, László Lovász, and Ravi Montenegro. Blocking conductance and mixing in random walks. *Combinatorics, Probability and Computing*, 15(4):541–570, 2006.

Ravindran Kannan and Hariharan Narayanan. Random walks on polytopes and an affine interior point method for linear programming. *Mathematics of Operations Research*, 37 (1):1–20, 2012.

Sebastian C. Kapfer and Werner Krauth. Sampling from a polytope and hard-disk Monte Carlo, 2013.

Jim Lawrence. Polytope volume computation. *Mathematics of Computation*, 57(195):259–271, 1991.

Yin Tat Lee and Aaron Sidford. Path finding methods for linear programming: Solving linear programs in $\tilde{O}(\sqrt{\text{rank}})$ iterations and faster algorithms for maximum flow. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 424–433. IEEE, 2014.

Yin Tat Lee and Santosh S. Vempala. Geodesic walks in polytopes. *arXiv preprint arXiv:1606.04696*, 2016.

Yin Tat Lee and Santosh S Vempala. Convergence rate of Riemannian Hamiltonian Monte Carlo and faster polytope volume computation. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1115–1121. ACM, 2018a.

Yin Tat Lee and Santosh S Vempala. Stochastic localization+ Stieltjes barrier= tight bound for log-sobolev. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1122–1129. ACM, 2018b.

László Lovász. Hit-and-run mixes fast. *Mathematical Programming*, 86(3):443–461, 1999.

László Lovász and Miklós Simonovits. The mixing rate of Markov chains, an isoperimetric inequality, and computing the volume. In *Proceedings of 31st Annual Symposium on Foundations of Computer Science, 1990*, pages 346–354. IEEE, 1990.

László Lovász and Miklós Simonovits. Random walks in a convex body and an improved volume algorithm. *Random Structures & Algorithms*, 4(4):359–412, 1993.

László Lovász and Santosh Vempala. Hit-and-run is fast and fun. *Tehnical Report, Microsoft Research*, 2003.

László Lovász and Santosh Vempala. Hit-and-run from a corner. *SIAM Journal on Computing*, 35(4):985–1005, 2006a.

László Lovász and Santosh Vempala. Simulated annealing in convex bodies and an $O^*(n^4)$ volume algorithm. *Journal of Computer and System Sciences*, 72(2):392–417, 2006b.

Michael W. Mahoney. Randomized algorithms for matrices and data. *Foundations and Trends in Machine Learning*, 3(2):123–224, 2011.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Hariharan Narayanan. Randomized interior point methods for sampling and optimization. *The Annals of Applied Probability*, 26(1):597–641, 2016.

Hariharan Narayanan and Alexander Rakhlin. Efficient sampling from time-varying log-concave distributions. *arXiv preprint arXiv:1309.5977*, 2013.

Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*. SIAM, 1994.

Brian D. Ripley. *Stochastic simulation*, volume 316. John Wiley & Sons, 2009.

Christian P. Robert. *Monte Carlo methods*. Wiley Online Library, 2004.

Sushant Sachdeva and Nisheeth K. Vishnoi. The mixing time of the Dikin walk in a polytope—a simple proof. *Operations Research Letters*, 44(5):630–634, 2016.

Robert L Smith. Efficient Monte Carlo procedures for generating points uniformly distributed over bounded regions. *Operations Research*, 32(6):1296–1308, 1984.

Pravin M. Vaidya. A new algorithm for minimizing convex functions over convex sets. In *30th Annual Symposium on Foundations of Computer Science, 1989*, pages 338–343. IEEE, 1989.

Pravin M. Vaidya and David S. Atkinson. A technique for bounding the number of iterations in path following algorithms. In *Complexity in Numerical Optimization*, pages 462–489. World Scientific, 1993.

Santosh Vempala. Geometric random walks: a survey. *Combinatorial and Computational Geometry*, 52(573-612):2, 2005.

Bin Yu and Per Mykland. Looking at Markov samplers through cusum path plots: a simple diagnostic idea. *Statistics and Computing*, 8(3):275–286, 1998.

Journal of Machine Learning Research 19 (2018) 1-82 Submitted 11/13; Revised 10/18; Published 11/18

Modular Proximal Optimization for Multidimensional Total-Variation Regularization

Álvaro Barbero ALVARO.BARBERO@INV.UAM.ES
Instituto de Ingeniería del Conocimiento and Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11, Madrid, Spain

Suvrit Sra* SUVRIT@MIT.EDU
Laboratory for Information and Decision Systems
Massachusetts Institute of Technology (MIT), Cambridge, MA

Editor: Vishwanathan S V N

Abstract

We study *TV regularization*, a widely used technique for eliciting structured sparsity. In particular, we propose efficient algorithms for computing prox-operators for ℓ_p-norm TV. The most important among these is ℓ_1-norm TV, for whose prox-operator we present a new geometric analysis which unveils a hitherto unknown connection to taut-string methods. This connection turns out to be remarkably useful as it shows how our geometry guided implementation results in efficient weighted and unweighted 1D-TV solvers, surpassing state-of-the-art methods. Our 1D-TV solvers provide the backbone for building more complex (two or higher-dimensional) TV solvers within a modular proximal optimization approach. We review the literature for an array of methods exploiting this strategy, and illustrate the benefits of our modular design through extensive suite of experiments on (i) image denoising, (ii) image deconvolution, (iii) four variants of fused-lasso, and (iv) video denoising. To underscore our claims and permit easy reproducibility, we provide all the reviewed and our new TV solvers in an easy to use multi-threaded C++, Matlab and Python library.

Keywords: proximal optimization, total variation, regularized learning, sparsity, non–smooth optimization

1. Introduction

Sparsity impacts the entire data analysis pipeline, touching algorithmic, modeling, as well as practical aspects. Most commonly, sparsity is elicited via ℓ_1-norm regularization (Tibshirani, 1996; Candès and Tao, 2004). However, numerous applications rely on more refined "structured" notions of sparsity, e.g., groupwise-sparsity (Meier et al., 2008; Liu and Zhang, 2009; Yuan and Lin, 2006; Bach et al., 2011), hierarchical sparsity (Bach, 2010; Mairal et al., 2010), gradient sparsity (Rudin et al., 1992; Vogel and Oman, 1996; Tibshirani et al., 2005), or sparsity over structured 'atoms' (Chandrasekaran et al., 2012).

Such regularizers typically arise in optimization problems of the form

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \Phi(\boldsymbol{x}) := \ell(\boldsymbol{x}) + r(\boldsymbol{x}), \tag{1.1}$$

where $\ell : \mathbb{R}^n \to \mathbb{R}$ is a smooth loss function (often convex), while $r : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ is a lower semicontinuous, convex, and nonsmooth regularizer that induces sparsity.

We focus on instances of (1.1) where r is a weighted *anisotropic Total-Variation* (TV) regularizer[1], which for a vector $\boldsymbol{x} \in \mathbb{R}^n$ and fixed weights $\boldsymbol{w} \geq 0$ is defined as

$$r(\boldsymbol{x}) \stackrel{\text{def}}{=} \mathrm{Tv}_p^1(\boldsymbol{w};\boldsymbol{x}) \stackrel{\text{def}}{=} \Big(\sum\nolimits_{j=1}^{n-1} w_j|x_{j+1}-x_j|^p\Big)^{1/p} \quad p \geq 1. \tag{1.2}$$

More generally, if X is an order-m tensor in $\mathbb{R}^{\prod_{j=1}^m n_j}$ with entries $\mathsf{X}_{i_1,i_2,\ldots,i_m}$ ($1 \leq i_j \leq n_j$ for $1 \leq j \leq m$); we define the weighted *m-dimensional anisotropic TV* regularizer as

$$\mathrm{Tv}_{\boldsymbol{p}}^m(\mathsf{W};\mathsf{X}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} \sum_{I_k=\{i_1,..,i_m\}\setminus i_k} \Big(\sum_{j=1}^{n_k-1} w_{I_k,j}|\mathsf{X}_{j+1}^{[k]} - \mathsf{X}_j^{[k]}|^{p_k}\Big)^{1/p_k}, \tag{1.3}$$

where $\mathsf{X}_j^{[k]} \equiv \mathsf{X}_{i_1,..,i_{k-1},j,i_{k+1},..,i_m}$, $w_{I_k,j} \geq 0$ are weights, and $\boldsymbol{p} \equiv [p_k \geq 1]$ for $1 \leq k \leq m$. If X is a matrix, expression (1.3) reduces to (note, $p, q \geq 1$)

$$\mathrm{Tv}_{p,q}^2(\mathsf{W};\mathsf{X}) = \sum_{i=1}^{n_1}\Big(\sum_{j=1}^{n_2-1} w_{1,j}|x_{i,j+1}-x_{i,j}|^p\Big)^{1/p} + \sum_{j=1}^{n_2}\Big(\sum_{i=1}^{n_1-1} w_{2,i}|x_{i+1,j}-x_{i,j}|^q\Big)^{1/q}, \tag{1.4}$$

These definitions look formidable; already 2D-TV (1.4) or even the simplest 1D-TV (1.2) are fairly complex, which further complicates the overall optimization problem (1.1). Fortunately, this complexity can be "localized" by invoking *prox-operators* (Moreau, 1962), which are now widely used across machine learning (Sra et al., 2011; Parikh et al., 2014).

The main idea of using prox-operators while solving (1.1) is as follows. Suppose Φ is a convex lsc function on a set $\mathcal{X} \subset \mathbb{R}^n$. The *prox-operator* of Φ is defined as the map

$$\mathrm{prox}_\Phi \stackrel{\text{def}}{=} \boldsymbol{y} \mapsto \operatorname*{argmin}_{\boldsymbol{x}\in\mathcal{X}} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \Phi(\boldsymbol{x}) \quad \text{for} \quad \boldsymbol{y} \in \mathbb{R}^n. \tag{1.5}$$

A popular method based on prox-operators is the *proximal gradient method* (also known as 'forward backward splitting'), which performs a gradient (forward) step followed by a proximal (backward) step to iterate

$$\boldsymbol{x}_{k+1} = \mathrm{prox}_{\eta_k r}(\boldsymbol{x}_k - \eta_k \nabla\ell(\boldsymbol{x}_k)), \quad k = 0, 1, \ldots. \tag{1.6}$$

Numerous other proximal methods exist—see e.g., (Beck and Teboulle, 2009; Nesterov, 2007; Combettes and Pesquet, 2009; Kim et al., 2010; Schmidt et al., 2011).

To implement the proximal-gradient iteration (1.6) efficiently, we require a subroutine that computes the prox-operator prox_r. An additional concern is whether the overall algorithm requires an *exact* computation of prox_r, or merely a moderately *inexact* computation. This concern is justified: rarely does r admit an exact algorithm for computing prox_r. Fortunately, proximal methods easily admit inexactness, e.g., (Schmidt et al., 2011; Salzo and Villa, 2012; Sra, 2012), which allows approximate prox-operators (as long as the approximation is sufficiently accurate).

We study both exact and inexact prox-operators in this paper, contingent upon the ℓ_p-norm used and on the data dimensionality m.

∗. An initial version of this work was performed during 2013-14, when the author was with the Max Planck Institute for Intelligent Systems, Tübingen, Germany, and with Carnegie Mellon University, Pittsburgh.

1. We use the term "anisotropic" to refer to the specific TV penalties considered in this paper.

1.1. Contributions

In particular, we review, analyze, implement, and experiment with a variety of fast algorithms. The ensuing contributions of this paper are summarized below.

- Geometric analysis that leads to a new, efficient version of the classic Taut String Method (Davies and Kovac, 2001), whose origins can be traced back to (Barlow, 1972) – this version turns out to perform better than most of the recently developed TV proximity methods.
- A previously unknown connection between (a variation of) this classic algorithm and Condat's *unweighted* TV method (Condat, 2012). This connection provides a geometric, more intuitive interpretation and helps us define a hybrid taut-string algorithm that combines the strengths of both methods, while also providing a new efficient algorithm for *weighted* ℓ_1-norm 1D-TV proximity.
- Efficient prox-operators for general ℓ_p-norm ($p \geq 1$) 1D-TV. In particular,
  - For $p = 2$, we present a specialized Newton method based on the root-finding strategy of Moré and Sorensen (1983),
  - For the general $p \geq 1$ case we describe both "projection-free" and projection based first-order methods.
- Scalable proximal-splitting algorithms for computing 2D (1.4) and higher-D TV (1.3) prox-operators. We review an array of methods in the literature that use prox-splitting, and through extensive experiments show that a splitting strategy based on alternating reflections is the most effective in practice. Furthermore, this modular construction of 2D and higher-D TV solvers allows reuse of our fast 1D-TV routines and exploitation of the massive parallelization inherent in matrix and tensor TV.
- The final most important contribution of our paper is a well-tuned, multi-threaded open-source C++, Matlab and Python implementation of all the reviewed and developed methods.[2]

To complement our algorithms, we illustrate several applications of TV prox-operators to: (i) image and video denoising; (ii) image deconvolution; and (iii) four variants of fused-lasso.

Note: We have invested great efforts to ensure reproducibility of our results. In particular, given the vast attention that TV problems have received in the literature, we believe it is valuable to both users of TV and other researchers to have access to our code, data sets, and scripts, to independently verify our claims, if desired.[3]

1.2. Related Work

The literature on TV is too large to permit a comprehensive review here. Instead, we mention the most directly related work to help place our contributions in perspective.

We focus on *anisotropic*-TV, in contrast to *isotropic*-TV (Rudin et al., 1992). Several proposals for designing an anisotropic variant of TV have been proposed in the literature:

2. See https://github.com/albarji/proxTV
3. This material shall be made available at: http://suvrit.de/work/soft/tv.html

in this paper we use the definition given in Bioucas-Dias and Figueiredo (2007), which follows the already presented Equation (1.2). Alternative definitions of anisotropic TV include instances such as a general TV defined in the continuous domain in terms of Wulff shapes (Esedoglu and Osher, 2004), or making use of estimates of the directional information (Steidl and Teuber , 2009), to name a few. Although the definition used here is simpler, it arises frequently in image denoising and signal processing, and quite a few TV-based denoising algorithms exist (Zhu and Chan, 2008, see e.g.).

The anisotropic TV regularizers $\mathrm{Tv}_1^{1\mathrm{D}}$ and $\mathrm{Tv}_{1,1}^{2\mathrm{D}}$ arise in image denoising and deconvolution (Dahl et al., 2010), in the fused-lasso (Tibshirani et al., 2005), in logistic fused-lasso (Kolar et al., 2010), in change-point detection (Harchaoui and Lévy-Leduc, 2010), in graph-cut based image segmentation (Chambolle and Darbon, 2009), in submodular optimization (Jegelka et al., 2013); see also the related work in (Vert and Bleakley, 2010). This broad applicability and importance of anisotropic TV is the key motivation towards developing carefully tuned proximity operators.

There is a rich literature of methods tailored to anisotropic TV, e.g., those developed in the context of fused-lasso (Friedman et al., 2007; Liu et al., 2010), graph-cuts (Chambolle and Darbon, 2009), ADMM-style approaches (Combettes and Pesquet, 2009; Wahlberg et al., 2012), fast methods based on dynamic programming (Johnson, 2013) or KKT conditions analysis (Condat, 2012). However, it seems that anisotropic TV norms other than ℓ_1 have not been studied much in the literature, although recognized as a form of Sobolev semi-norms (Pontow and Scherzer, 2009).

For 1D-TV and for the particular ℓ_1 norm, there exist several direct methods that are exceptionally fast. We treat this problem in detail in Section 2, and hence refer the reader to that section for discussion of closely related work on fast solvers. We note here, however, that in contrast to many of the previous fast solvers, our solvers allow weights, a capability that can be very important in applications (Jegelka et al., 2013).

Regarding 2D-TV, Goldstein T. (2009) presented a so-called "Split-Bregman" (SB). It turns out that this method is essentially a variant of the well-known ADMM method. In contrast to the 2D approach presented here, the SB strategy followed by Goldstein T. (2009) is to rely on ℓ_1-soft thresholding substeps instead of 1D-TV substeps. From an implementation viewpoint, the SB approach is somewhat simpler, but not necessarily more accurate. Incidentally, sometimes such direct ADMM approaches turn out to be less effective than ADMM methods that rely on more complex 1D-TV prox-operators (Ramdas and Tibshirani, 2014).

It is worth highlighting that it is not just proximal solvers such as FISTA (Beck and Teboulle, 2009), SpaRSA (Wright et al., 2009), SALSA (Afonso et al., 2010), TwIST (Bioucas-Dias and Figueiredo, 2007), Trip (Kim et al., 2010), that can benefit from our fast prox-operators. All other 2D and higher-D TV solvers, e.g., (Yang et al., 2013), as well as the recent ADMM based trend-filtering solvers of Tibshirani (2014) immediately benefit, not only in speed but also by gaining the ability to solve weighted problems.

1.3. Summary of the Paper

The remainder of the paper is organized as follows. In Section 2 we consider prox operators for 1D-TV problems when using the most common ℓ_1 norm. The highlight of this section

is our analysis on taut-string TV solvers, which leads to the development a new hybrid method and a weighted TV solver (Sections 2.3, 2.4). Thereafter, we discuss variants of 1D-TV (Section 3), including a specialized Tv_2^{1D} solver, and a more general Tv_p^{1D} method based on a gradient projection strategy. Subsequently, we describe multi-dimensional TV problems and study their prox-operators in Section 4, paying special attention to 2D-TV; for both 2D and multi-D, prox-splitting methods are used. After these theoretical sections, we describe experiments and applications in Section 5. In particular, extensive experiments for 1D-TV are presented in Section 5.1 and Section 5.2; 2D-TV experiments are in Section 5.3, while an application of multi-D TV is the subject of Section 5.4. The appendices to the paper include further technical details and additional information about the experimental setup.

2. TV-L1: Fast Prox-Operators for Tv_1^{1D}

We begin with the 1D-TV problem (1.2) for an ℓ_1 norm choice, for which we review several carefully tuned algorithms. Using such well–tuned algorithms pays off: we can find fast, robust, and low-memory (in fact, in place) algorithms, which are not only of independent value, but also ideal building blocks for scalably solving 2D- and higher-D TV problems.

Computation of the ℓ_1-norm TV prox-operator can be compactly written as the problem

$$\min_{\boldsymbol{x}\in\mathbb{R}^n} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{D}\boldsymbol{x}\|_1, \tag{2.1}$$

where $\boldsymbol{D}$ is the *differencing matrix*, all zeros except $d_{ii} = -1$ and $d_{i,i+1} = 1$ $(1 \le i \le n-1)$.

To solve (2.1) we will analyze an approach based on the line of "taut-string" methods. We first introduce these methods for the unweighted *TV-L1* problem (2.1), before discussing the elementwise weighted TV problem (2.6). Most of the previous fastest methods handle only unweighted-TV. It is often nontrivial to extend them to handle weighted-TV, a problem that is crucial to several applications, e.g., segmentation (Chambolle and Darbon, 2009) and certain submodular optimization problems (Jegelka et al., 2013).

A remarkably efficient approach to TV-L1 was presented in (Condat, 2012). We will show Condat's fast algorithm can be interpreted as a "linearized" version of the taut-string approach, a view that paves the way to obtain an equally fast solver for weighted TV-L1.

Before proceeding we note that other than (Condat, 2012), other efficient methods to address unweighted Tv_1^{1D} proximity have been proposed. Johnson (2013) shows how solving Tv_p^{1D} proximity is equivalent to computing the data likelihood of an specific Hidden Markov Model (HMM), which suggests a dynamic programming approach based on the well-known Viterbi algorithm for HMMs. The resulting algorithm is very competitive, and guarantees an overall $O(n)$ performance while requiring approximately $8n$ storage. Another similarly performing algorithm was presented by Kolmogorov et al (2015) in the form of a message passing method. We will also consider these algorithms in our experimental comparison in §5.1.

Yet another family of methods is based on projected-Netwon (PN) techniques: we also present in Appendix E a PN approach for its instructive value, and also because it provides key subroutines for solving TV problems with $p > 1$. Our derivation may also be helpful to readers seeking to implement efficient prox-operators for problems that have structure

similar to TV, for instance ℓ_1-trend filtering (Kim et al., 2009; Tibshirani, 2014). Indeed, the PN approach proves to be foundational for the fast "group fused-lasso" algorithms of (Wytock et al., 2014).

2.1. The Taut-String Method for Tv_1^{1D}

While taut-string methods seem to be largely unknown in machine learning, they have been widely applied in statistics—see e.g., (Grasmair, 2007; Davies and Kovac, 2001; Barlow, 1972).

We start by transforming the problem as follows. For TV-L1, elementary manipulations, e.g., using Proposition A.4, yield the dual (re-written as a minimization problem)

$$\min_{\boldsymbol{u}} \ \tfrac{1}{2}\|\boldsymbol{D}^T\boldsymbol{u}\|_2^2 - \boldsymbol{u}^T\boldsymbol{D}\boldsymbol{y}, \ \text{s.t.} \ \|\boldsymbol{u}\|_\infty \le \lambda. \tag{2.2}$$

Without changing the minimizer, the objective (2.2) can be replaced by $\|\boldsymbol{D}^T\boldsymbol{u} - \boldsymbol{y}\|_2^2$, which then unfolds into

$$(u_1 - y_1)^2 + \sum\nolimits_{i=2}^{n-1} (-u_{i-1} + u_i - y_i)^2 + (-u_{n-1} - y_n)^2 .$$

Introducing the fixed extreme points $u_0 = u_n = 0$, we can replace the problem (2.2) by

$$\min_{\boldsymbol{u}} \ \sum\nolimits_{i=1}^{n} (-u_{i-1} + u_i - y_i)^2 , \ \text{s.t.} \ \|\boldsymbol{u}\|_\infty \le \lambda, \quad u_0 = u_n = 0. \tag{2.3}$$

Now we perform a change of variables by defining the new set of variables $\boldsymbol{s} = \boldsymbol{r} - \boldsymbol{u}$, where $r_i := \sum_{k=1}^{i} y_k$ is the cumulative sum of input signal values. Thus, (2.3) becomes

$$\min_{\boldsymbol{s}} \ \sum\nolimits_{i=1}^{n} (-r_{i-1} + s_{i-1} + r_i - s_i - y_i)^2 , \ \text{s.t.} \ \|\boldsymbol{s} - \boldsymbol{r}\|_\infty \le \lambda, \quad r_0 - s_0 = r_n - s_n = 0,$$

which upon simplification becomes

$$\min_{\boldsymbol{s}} \ \sum\nolimits_{i=1}^{n} (s_{i-1} - s_i)^2 , \quad \text{s.t.} \quad \|\boldsymbol{s} - \boldsymbol{r}\|_\infty \le \lambda, s_0 = 0, \ s_n = r_n. \tag{2.4}$$

Now the key trick: problem (2.4) can be shown to share the same optimum as

$$\min_{\boldsymbol{s}} \ \sum_{i=1}^{n} \sqrt{1 + (s_{i-1} - s_i)^2}, \ \text{s.t.} \ \|\boldsymbol{s} - \boldsymbol{r}\|_\infty \le \lambda, \quad s_0 = 0, \ s_n = r_n. \tag{2.5}$$

A proof of this relationship may be found in (Steidl et al., 2005); for completeness, and also because it will help us generalize to the weighted Tv_1^{1D} variant, we include an alternative proof in Appendix C.

The name "taut-string" is explained as follows. The objective in (2.5) can be interpreted as the Euclidean length of a polyline through the points $(i, \boldsymbol{s}_i)$. Thus, (2.5) seeks the minimum length polyline (the *taut-string*) crossing a tube of height λ with center the cumulative sum $\boldsymbol{r}$ and having the fixed endpoints (s_0, s_n). An example illustrating this description is shown in Figure 1.

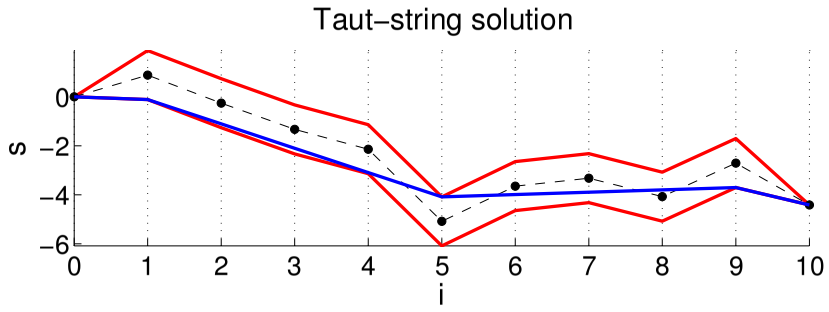


Figure 1: Example of the taut string method. The cumulative sum $\boldsymbol{r}$ of the input signal values $\boldsymbol{y}$ is shown as the dashed line; the black dots mark the points (i, r_i). The bottom and top of the λ-width tube are shown in red. The taut string solution $\boldsymbol{s}$ is shown as a blue line.

Once the taut string is found, the solution for the original TV problem (2.1) can be recovered by observing that

$$s_i - s_{i-1} \quad = \quad r_i - u_i - (r_{i-1} - u_{i-1}) \quad = \quad y_i - u_i + u_{i-1} \quad = \quad x_i,$$

where we used the primal-dual relation $\boldsymbol{x} = \boldsymbol{y} - \boldsymbol{D}^T\boldsymbol{u}$. Intuitively, the above argument shows that the solution to the TV-L1 proximity problem is obtained as the discrete gradient of the taut string, or as the slope of its segments.

It remains to describe how to find the taut string. The most widely used approach seems to be the one due to Davies and Kovac (2001). This approach starts from the fixed point $s_0 = 0$, and incrementally computes the *greatest convex minorant* of the upper bounds on the λ tube, as well as the *smallest concave majorant* of the lower bounds on the λ tube. When both curves intersect, the *left-most* point where either the majorant or the minorant touched the tube is used to fix a first segment of the taut string. The procedure is then resumed at the end of the identified segment, and iterated until all taut string segments have been obtained. Pseudocode of this method is presented as Algorithm 1, while an example of this procedure is shown in Figure 2.

It is important to note that since we have a discrete number of points in the tube, the greatest convex minorant can be expressed as a piecewise linear function with segments of monotonically increasing slope, while the smallest concave majorant is another piecewise linear function with segments of monotonically decreasing slope. Another relevant fact is that each segment in the tube upper/lower bound enters the minorant/majorant exactly once in the algorithm, and is also removed exactly once. This limits the extent of the inner loops in the algorithm, and in fact an analysis of the computational complexity of this behavior leads to an overall $O(n)$ performance (Davies and Kovac, 2001).

In spite of this, Condat (2012) notes that maintaining the minorant and majorant functions in memory is inefficient, and views a taut-string approach as potentially inferior to

Algorithm 1 Taut string algorithm for TV-L1-proximity

Inputs: input signal $\boldsymbol{y}$ of length n, regularizer λ.
Initialize $i = 0$, $concmajorant = \varnothing$, $convminorant = \varnothing$, $\boldsymbol{r}_i = \sum_{k=1}^{i} \boldsymbol{y}_k$.
while $i < n$ **do**
  Add new segment: $concmajorant = concmajorant \cup ((i-1, \boldsymbol{r}_{i-1} - \lambda) \to (i, \boldsymbol{r}_i - \lambda))$.
  while $concmajorant$ is not concave **do**
    Merge the last two segments of $concmajorant$
  end while
  Add new segment: $convminorant = convminorant \cup ((i-1, \boldsymbol{r}_{i-1} + \lambda) \to (i, \boldsymbol{r}_i + \lambda))$.
  while $convminorant$ is not convex **do**
    Merge the last two segments of $convminorant$
  end while
  if slope(left-most segment in $concmajorant$) > slope(lest-most segment in $convminorant$) **then**
    $break$ = left-most point where either the majorant or the minorant touched the tube
    if $break \in convminorant$ **then**
      Remove left-most segment of the minorant, add it to the taut-string solution $\boldsymbol{x}$.
      Majorant is recalculated as a straight line from $break$ to its last point.
    end if
    if $break \in concmajorant$ **then**
      Remove left-most segment of the majorant, add it to the taut-string solution $\boldsymbol{x}$.
      Minorant is recalculated as a straight line from $break$ to its last point.
    end if
  end if
  $i++$
end while
Add last segment from either the majorant or minorant to the solution $\boldsymbol{x}$.

his proposed method. To this observation we make two claims: Condat's method can be interpreted as a linearized version of the taut-string method (see Section 2.2); and that a careful implementation of the taut-string method can be highly competitive in practice.

2.1.1. Efficient Implementation of Taut-Strings

We propose now an efficient implementation of the taut-string method. The main idea is to carefully use double-ended queues (Knuth, 1997) to store the majorant and minorant information. Therewith, all majorant/minorant operations such as appending a segment or removing segments from either the beginning or the end of the majorant can be performend in constant time. Note however that usual double-ended queue implementations use doubly linked lists, dynamic arrays or circular buffers: these approaches require dynamically reallocating memory chunks at some of the insert or remove operations. But in the taut-string algorithm, the maximum number of segments of the majorant/minorant is just the size of the input signal (n), and also the number of segments to be inserted in the queue throughout the algorithm will be n. Making use of these facts we implement a specialized queue based on a contiguous array of fixed length n. New segments are added from the start of the array on, and a couple of pointers are maintained to keep track of the first and last valid segments in the array, much in the way of a circular buffer. This implementation, however, does not require of the usual circular logic. Overall, this double-ended queue

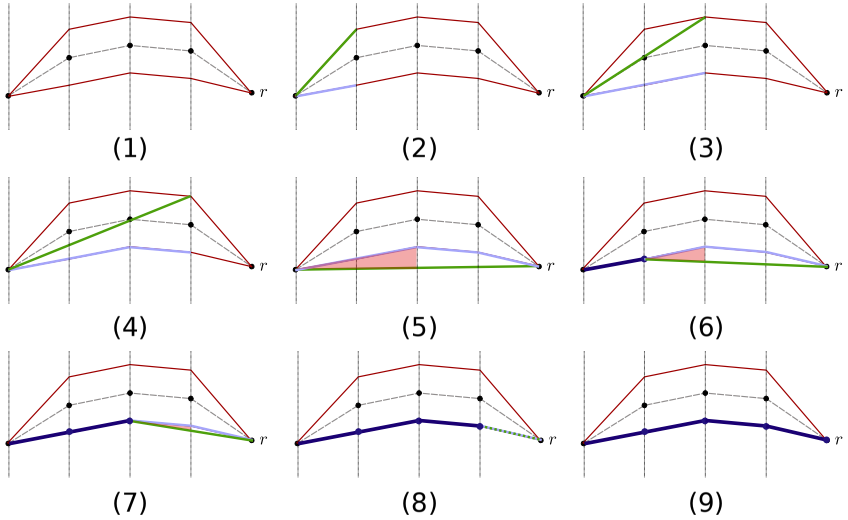


Figure 2: Example of the evolution of the taut string method. The smallest concave majorant (blue) and largest convex minorant (green) are updated are every step. At step (1) the algorithm is initialized. Steps (2) to (4) successfully manage to update majorant and minorant without producing crossings between them. Note how while the concave majorant keeps adding segments without issue, the convex minorant must remove and merge existing segments with new ones to mantain a convex function from the origin to the new points. At step (5) the end of the tube is reached, but the minorant and majorant slopes overlap, and so it is necessary to break the segment at the left-most point where the majorant/minorant touched the tube. Since the left-most touching point is in the concave majorant it's leftmost segment is removed and placed in the solution, while the convex minorant is updated as a straight line from the detected breakpoint to the last explored point, resulting in (6). The algorithm would then continue adding segments, but since the majorant/minorant slopes are still crossing, the procedure of fixing segments to the solution is repeated through steps (6), (7) and (8). Finally at step (9) the slopes are no longer crossing and the method would continue adding tube segments, but since the end of the tube has already been reached the algorithm stops.

requires a single memory allocation at the beginning of the algorithm, keeping the rest of queue operations free from memory management and all but the simplest pointer or index algebra.

We also store for each segment the following values: x length of the segment, y length and slope. Slopes might seem as redundant given the other two factors, but given the number of times the algorithm requires comparing slopes between segments (e.g., to preserve convexity/concavity) it pays off to precompute these values. This fact together with

other calculation and code optimization details produces our implementation; these can be reviewed in the code itself at `https://github.com/albarji/proxTV`.

2.2. Linearized Taut-String Method for $\mathbf{TV}_1^{\mathbf{1D}}$

We now present a variant, linearized version of the taut-string method. Surprisingly, the resulting algorithm turns out to be equivalent to the fast algorithm of Condat (2012), though now with a clearer interpretation based on taut-strings.

The key idea is to build linear approximations to the greatest convex minorant and smallest concave majorant, producing exactly the same results but significantly reducing the bookkeeping of the method to a handful of simple variables. We therefore replace the greatest convex minorant and smallest convex majorant by a *greatest affine minorant* and *smallest affine majorant*.

An example of the method is presented in Figure 3. A proof showing that this linearization does not change the resultant taut-string is given in Appendix D. In what follows, we describe the linearized method in depth.

Details. Linearized taut-string requires only the following bookkeeping variables:

1. i_0: index of the current segment start
2. $\bar{\delta}$: slope of the majorant
3. $\underline{\delta}$: slope of the minorant
4. $\bar{h}$: height of majorant w.r.t. the λ-tube center
5. $\underline{h}$: height of minorant w.r.t. λ-tube center
6. $\bar{i}$: index of last point where $\bar{\delta}$ was updated—potential majorant break point
7. $\underline{i}$: index of last point where $\underline{\delta}$ was updated—potential minorant break point.

Figure 4 gives a geometric interpretation of these variables; we use these variables to detect minorant-majorant intersections, without the need to compute or store them explicitly.

Algorithm 2 presents full pseudocode of the linearized taut-string method. Broadly, the algorithm proceeds in the same fashion as the classic taut-string method, updating the affine approximations to the majorant and minorant at each step, and introducing a breakpoint whenever the slopes of these two functions cross.

More precisely, at each each iteration the method steps one point further through the tube, updating the minorant/majorant slopes ($\underline{\delta}$, $\bar{\delta}$) as well as their heights at the current point ($\underline{h}$, $\bar{h}$). To check for minorant/majorant crossings it suffices to compare the slopes ($\underline{\delta}$, $\bar{\delta}$), or equivalently, to check whether the height of the minorant $\underline{h}$ falls below the tube bottom (since the minorant follows the tube ceiling) or the height of the majorant $\bar{h}$ grows above the tube ceiling (since the majorant follows the tube bottom). We make use of this last variant, since updating heights turns out to be slightly cheaper than updating slopes, and so it is faster to ensure no crossing will take place before performing such updates.

When a crossing is detected, we perform similar steps as in the classic taut-string method but with one significant difference: the algorithm is completely restarted at the newly introduced breakpoint. This restart idea is in contrast with the classic method, where we simply re-use the previously computed information about the minorant and majorant to update their estimates and continue working with them. In the linearized version we do

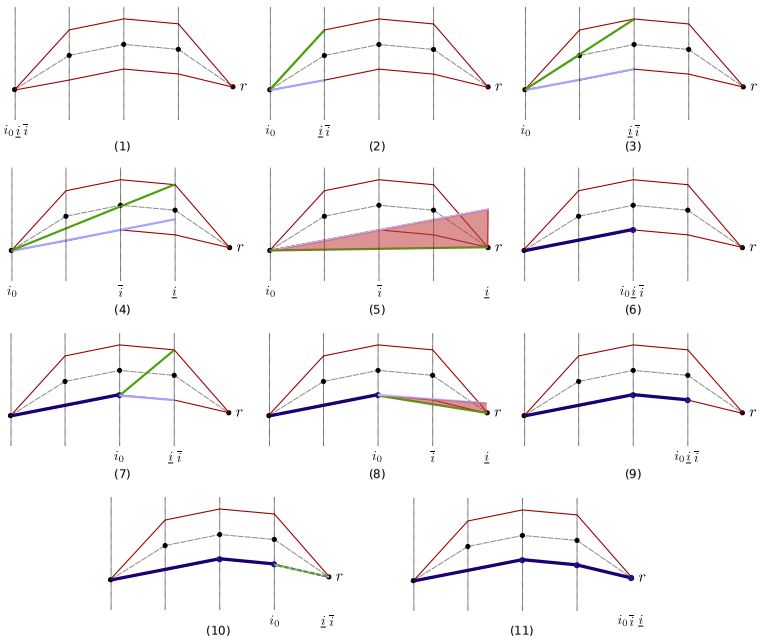


Figure 3: Example of the evolution of the linearized taut string method. The smallest affine majorant of the tube bottom (blue) and greatest affine minorant of the tube ceiling (green) are updated at every step. At step (1) the algorithm is initialized. Steps (2) to (4) successfully manage to update majorant/minorant without crossings. At step (5), however, the slopes cross, and so it is necessary to break the segment. Since the left-most tube touching point is the one in the majorant, the majorant is broken down at that point and its left-hand side is added to the solution, resulting in (6). The method is then restarted at the break point, with majorant/minorant being updated at step (7), though at step (8) once again a crossing is detected. Hence, at step (9) a breaking point is introduced again and the algorithm is restarted once more. Following this, step (10) manages to update majorant/minorant slopes up to the end of the tube, and so at step (11) the final segment is built using the (now equal) slopes.

not keep enough information to perform such an operation, so all data about minorant and majorant is discarded and the algorithm begins anew. Because of this choice the same tube segment might be reprocessed up to $O(n)$ times in the method, and therefore the overall worst case performance is $O(n^2)$. This fact was already observed in (Condat, 2012).

In what follows we describe the rationale behind the height update formulae.

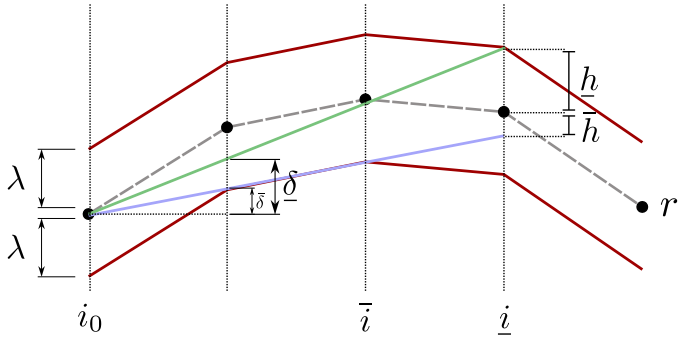


Figure 4: Illustration of the geometric concepts involved in the linearized taut string method. The greatest linear minorant (of the tube ceiling) is depicted in green, while the smallest linear majorant (of the tube bottom) is shown in blue. The δ slopes and h heights are presented updated up to the index shown as $\underline{\imath}$.

Height variables. To implement the method described above, the height variables h are not strictly necessary as they can be obtained from the slopes δ. However, explicitly including them leads to efficient updating rules at each iteration, as we show below.

Suppose we are updating the heights and slopes from their estimates at step $i-1$ to step i. Updating the heights is immediate given the slopes, since

$$h_i = h_{i-1} + \delta - y_i.$$

In other words, since we are following a line with slope δ, the change in height from one step to the next is given by precisely such a slope. Note, however, that in this algorithm we do not compute absolute heights but instead relative heights with respect to the λ–tube center. Therefore we need to account for the change in the tube center between steps $i-1$ and i, which is given by $r_i - r_{i-1} = y_i$. This completes the update, which is shown in Algorithm 2 as lines 4 and 11.

However, it is possible that the new height h runs over or under the tube. This would mean that we cannot continue using the current slope in the majorant or minorant, and a recalculation is needed, which again can be done efficiently by using the height information. Assume without loss of generality that the starting index of the current segment is 0 and the absolute height of the starting point of the segment is given by α. Then, for adjusting the minorant slope $\bar{\delta}_i$ so that it touches the tube ceiling at the current point, we note that

$$\bar{\delta}_i = \frac{\lambda + r_i - \alpha}{i} = \frac{\lambda + (\bar{h}_i - \bar{h}_i) + r_i - \alpha}{i},$$

where we have also added and subtracted the current value of $\bar{h}_i$. Observe that this value was computed using the estimate δ_{i-1} of the slope so far, so we can rewrite it as the projection of the initial point in the segment following such a slope, that is, as $\bar{h}_i = i\bar{\delta}_i - r_i + \alpha$. Doing

Algorithm 2 Linearized taut string algorithm for TV-L1-proximity

1: Initialize $i = \bar{i} = \underline{i} = \bar{h} = \underline{h} = 0$, $\underline{\delta} = y_0 + \lambda$, $\bar{\delta} = y_0 - \lambda$
2: **while** $i < n$ **do**
3: Find tube height: $\tilde{\lambda} = \lambda$ if $i < n-1$, else $\tilde{\lambda} = 0$
4: Update majorant height following current slope: $\bar{h} = \bar{h} + \bar{\delta} - y_i$.
5: /* Check for ceiling violation: majorant is above tube ceiling */
6: **if** $\bar{h} > \tilde{\lambda}$ **then**
7: Build valid segment up to last majorant breaking point: $\boldsymbol{x}_{i_0+1:\bar{i}} = \bar{\delta}$.
8: Start new segment after break: $(i_0, \underline{i}) = \bar{i}$, $\underline{\delta} = y_i + 2\lambda$, $\bar{\delta} = y_i$, $\underline{h} = \lambda$, $\bar{h} = -\lambda$, $i = \bar{i} + 1$
9: **continue**
10: **end if**
11: Update minorant height following current slope: $\underline{h} = \underline{h} + \underline{\delta} - y_i$.
12: /* Check for bottom violation: minorant is below tube bottom */
13: **if** $\underline{h} < -\tilde{\lambda}$ **then**
14: Build valid segment up to last minorant breaking point: $\boldsymbol{x}_{i_0+1:\underline{i}} = \underline{\delta}$.
15: Start new segment after break: $(i_0, \bar{i}) = \underline{i}$, $\underline{\delta} = y_i$, $\bar{\delta} = -2\lambda + y_i$, $\underline{h} = \lambda$, $\bar{h} = -\lambda$, $i = \underline{i} + 1$
16: **continue**
17: **end if**
18: /* Check if majorant height is below the floor */
19: **if** $\bar{h} \leq -\tilde{\lambda}$ **then**
20: Correct slope: $\bar{\delta} = \bar{\delta} + \frac{\tilde{\lambda} - \bar{h}}{i - i_0}$
21: The majorant now touches the floor: $\bar{h} = -\tilde{\lambda}$
22: This is a possible majorant breaking point: $\bar{i} = i$
23: **end if**
24: /* Check if minorant height is above the ceiling */
25: **if** $\underline{h} \geq \tilde{\lambda}$ **then**
26: Correct slope: $\underline{\delta} = \underline{\delta} + \frac{-\tilde{\lambda} - \underline{h}}{i - i_0}$
27: The minorant now touches the ceiling: $\underline{h} = \tilde{\lambda}$
28: This is a possible minorant breaking point: $\underline{i} = i$
29: **end if**
30: Continue building current segment: $i = i + 1$
31: **end while**
32: Build last valid segment: $\boldsymbol{x}_{i_0+1:n} = \bar{\delta}$.

so for one of the added heights $\bar{h}_i$ produces

$$\bar{\delta}_i = \frac{\lambda + (i\bar{\delta}_{i-1} - r_i + \alpha) - \bar{h}_i + r_i - \alpha}{i} = \bar{\delta}_{i-1} + \frac{\lambda - \bar{h}_i}{i},$$

which generates a simple updating rule. A similar derivation holds for the minorant. The resulting updates are included in the algorithm in lines 20 and 26. After recomputing this slope we need to adjust the corresponding height back to the tube: since the heights are relative to the tube center we can just set $\bar{h} = \lambda$, $\underline{h} = -\lambda$; this is done in lines 21 and 27.

Notice also that the special case of the last point in the tube where the taut-string must meet $s_n = r_n$ is handled by line 3, where $\tilde{\lambda}$ is set to 0 at such a point to enforce this constraint. Overall, one iteration of the method is very efficient, as mostly just additions and subtractions are involved with the sole exception of the division required for the slope updates, which are not performed at every iteration. Moreover, no additional memory is

	Classic	**Linearized (Condat's)**
Worst-case performance	$O(n)$	$O(n^2)$
In–memory	No	Yes
Other considerations	Fast bookkeeping through double-ended queues	Very fast iteration, cache friendly

Table 1: Comparison of the main features of reviewed taut-string algorithms.

required beyond the constant number of bookkeeping variables, and in-place updates are also possible because y_i values for already fixed sections of the taut-string are not required again, so the output $\boldsymbol{x}$ and the input $\boldsymbol{y}$ can both refer to the same memory locations.

The resulting algorithm turns out to be equivalent, almost line by line, to the method of Condat (2012), even though its theoretical grounds are radically different: while the approach presented here has a strong geometric basis due to its taut-string relationship, (Condat, 2012) is based solely on analysis of KKT conditions. Therefore, we have shown that Condat's fast TV method is, in fact, a linearized taut-string algorithm.

2.3. Comparison of Taut-String Methods and a Hybrid Strategy

Table 1 summarizes the main features of the classic and linearized taut-string methods reviewed so far. Although the classic taut-string method has been largely neglected in the machine learning literature, its guarantee in linear performance makes it an attractive choice. Furthermore, although we could not find any references on implementation details of this method, we have empirically seen that a very efficient solver can be produced by making use of a double-ended queue to bookkeep the majorant/minorant information.

In contrast to this, the linearized taut-string method (equivalent to Condat (2012)) features a much better performance per step in the tube traversal, mainly due to not requiring additional memory and making use of only a small constant number of variables, making the method friendly for CPU cache or registers calculation. As a tradeoff of keeping such scarce information in memory, the method does not guarantee linear performance, falling to a quadratic theoretical runtime in the worst case. This fact was already observed in (Condat, 2012), though such worst case was deemed as pathological, claiming a $O(n)$ performance in all practical situations. We shall review these claims in the experimental sections in this manuscript.

The key points of Table 1 show that no taut-string variant is clearly superior. While the classic method provides a safe linear time solution to the problem, the linearized method is potentially faster but riskier in terms of worst case performance. Following these observations we propose here a simple hybrid method combining both approaches: run the linearized algorithm up to a prefixed number of steps n^S, $S \in (1, 2)$, and if the solution has not yet been found, we switch to the classic method. We therefore limit the worst-case scenario to $O(n^S) + O(n) \simeq O(n^S)$, because once the classic method kicks, it will ensure an $O(n)$ performance guarantee.

Implementation of this hybrid method is easy upon realizing the similarities between algorithms: a switch–check is added to the linearized method every time a segment of the taut-string has been identified (Algorithm 2, lines 7, 14). If it is confirmed that the method

has already run for n^S steps without reaching the solution, the remaining part of the signal for which the taut-string has not yet been found is passed on to the classic method, whose solution is concatenated to the part the linearized method managed to find so far. We also report the empirical performance of this method in the experimental section.

2.4. Taut-string Methods for Weighted Tv_1^{1D}

Several applications TV require penalizing the discrete gradients individually, which can be done by solving the *weighted TV-L1* problem

$$\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \sum_{i=1}^{n-1} w_i|x_{i+1}-x_i|, \tag{2.6}$$

where the weights $\{w_i\}_{i=1}^{n-1}$ are all positive. To solve (2.6) using a taut-string approach, we again begin with its dual (written as a minimization problem)

$$\min_{\boldsymbol{u}} \quad \tfrac{1}{2}\|\boldsymbol{D}^T\boldsymbol{u}\|_2^2 - \boldsymbol{u}^T\boldsymbol{D}\boldsymbol{y} \quad \text{s.t.} \quad |u_i| \le w_i, \quad 1 \le i < n. \tag{2.7}$$

Then, we repeat the derivation of the unweighted taut-string method but with a few key modifications. More precisely, we transform (2.7) by introducing $u_0 = u_n = 0$ to obtain

$$\min_{\boldsymbol{u}} \ \sum_{i=1}^{n} (y_i - u_i + u_{i-1})^2 \quad \text{s.t.} \ |u_i| \le w_i, \quad 1 \le i < n.$$

Then, we perform the change of variables $\boldsymbol{s} = \boldsymbol{r} - \boldsymbol{u}$, where $r_i := \sum_{k=1}^{i} y_k$, and consider

$$\min_{\boldsymbol{s}} \sum_{i=1}^{n} (s_i - s_{i-1})^2 \quad \text{s.t.} \ |s_i - r_i| \le w_i, \quad 1 \le i < n \ s_0 = 0, \ s_n = r_n.$$

Finally, applying Theorem C.1 we obtain the equivalent *weighted taut-string* problem

$$\min_{\boldsymbol{s}} \sum_{i=1}^{n} \sqrt{1 + (s_i - s_{i-1})^2} \quad \text{s.t.} \ |s_i - r_i| \le w_i, \ 1 \le i < n, \ s_0 = 0, \ s_n = r_n. \tag{2.8}$$

Problem (2.8) differs from its unweighted counterpart (2.5) in the constraints $|s_i - r_i| \le w_i$ $(1 \le i < n)$, which allow different weights for each component instead of using the same value λ. Our geometric intuition also carries over to the weighted problem, albeit with a slight modification: the tube we are trying to traverse now has varying widths at each step instead of the previous fixed λ width—Figure 5 illustrates this idea.

As a consequence of the above derivation and intuition, taut-string methods can be produced to solve the weighted Tv_1^{1D} problem. The original formulation of the classic taut-string method in (Davies and Kovac, 2001) defines the limits of the tube through possibly varying bottom and ceiling values (l_i, u_i) $\forall i$, and so this method easily extends to solve the weighted TV problem by assigning $l_i = r_i - w_i$, $u_i = r_i + w_i$. In our pseudocode in Algorithm 1 we just need to replace λ by the appropriate w_i values.

Similar considerations apply for the linearized version (Algorithm 2), in particular, when checking ceiling/floor violations as well as when checking slope recomputations and restarts, we must account for varying tube heights. Algorithm 3 presents the precise modifications that we must make to Algorithm 2 to handle weights. Regarding the convergence of this method, the proof of equivalence with the classic taut-string method still holds in the weighted case (see Appendix D).

The very same analysis as portrayed in Table 1 applies here: both the benefits and problems of the two taut-string solvers carry on to the weighted variant of the problem.

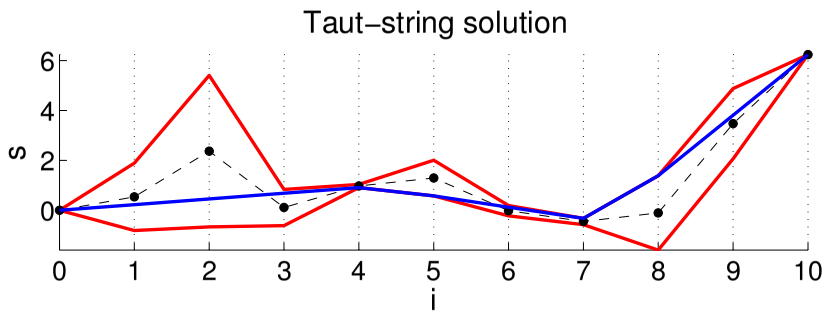


Figure 5: Example of the weighted taut string method with $\boldsymbol{w} = (1.35, 3.03, 0.73, 0.06, 0.71, 0.20, 0.12, 1.49, 1.41)$. The cumulative sum $\boldsymbol{r}$ of the input signal values $\boldsymbol{y}$ is shown as the dashed line, with the black dots marking the points $(i, \boldsymbol{r}_i)$. The bottom and ceiling of the tube are shown in red, which vary in width at each step following the weights $\boldsymbol{w}_i$. The weighted taut string solution $\boldsymbol{s}$ is shown as a blue line.

Algorithm 3 Modified lines for weighted version of Algorithm 2

3: Find tube height: $\tilde{\lambda} = w_{i+1}$ if $i < n-1$, else $\tilde{\lambda} = 0$
8: Start new segment after break: $(i_0, \underline{i}) = \bar{i}$, $\underline{\delta} = y_i + w_{i-1} + w_i$, $\bar{\delta} = y_i + w_{i-1} - w_i$, $\underline{h} = w_i$, $\bar{h} = -w_i$, $i = \bar{i} + 1$
15: Start new segment after break: $(i_0, \bar{i}) = \underline{i}$, $\underline{\delta} = y_i + w_{i-1} - w_i$, $\bar{\delta} = y_i + w_{i-1} + w_i$, $\underline{h} = w_i$, $\bar{h} = -w_i$, $i = \underline{i} + 1$

3. Other One-Dimensional TV Variants

While more infrequent, replacing the ℓ_1 norm of the standard TV regularizer by an ℓ_p-norm version can also be useful. In this section we focus first on a specialized solver for $p = 2$, before discussing a less efficient but more general solver for any ℓ_p with $p \ge 1$. We also briefly cover the $p = \infty$ case.

3.1. TV-L2: Proximity for Tv_2^{1D}

For TV-L2 proximity ($p = 2$) the dual to the prox-operator for (1.2) reduces to

$$\min_{\boldsymbol{u}} \ \phi(\boldsymbol{u}) := \tfrac{1}{2}\|\boldsymbol{D}^T\boldsymbol{u}\|_2^2 - \boldsymbol{u}^T\boldsymbol{D}\boldsymbol{y}, \ \text{ s.t. } \|\boldsymbol{u}\|_2 \le \lambda. \tag{3.1}$$

Problem (3.1) is nothing but a version of the well-known trust-region subproblem (TRS), for which a variety of numerical approaches are known (Conn et al., 2000).

We derive a specialized algorithm based on the classic Moré-Sorensen Newton (MSN) method of (Moré and Sorensen, 1983). This method in general can be quite expensive, but for (3.1) the Hessian is tridiagonal which can be well-exploited (see Appendix E). Curiously, experiments show that for a limited range of λ values, even ordinary gradient-projection

(GP) can be competitive. But for overall best performance, a hybrid MSN-GP approach is preferable.

Towards solving (3.1), consider its KKT conditions:

$$\begin{aligned} &(\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})\boldsymbol{u} = \boldsymbol{D}\boldsymbol{y}, \\ &\alpha(\|\boldsymbol{u}\|_2 - \lambda) = 0, \quad \alpha \geq 0, \end{aligned} \tag{3.2}$$

where α is a Lagrange multiplier. There are two possible cases: either $\|\boldsymbol{u}\|_2 < \lambda$ or $\|\boldsymbol{u}\|_2 = \lambda$.

If $\|\boldsymbol{u}\|_2 < \lambda$, then the KKT condition $\alpha(\|\boldsymbol{u}\|_2 - \lambda) = 0$, implies that $\alpha = 0$ must hold and $\boldsymbol{u}$ can be obtained immediately by solving the linear system $\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} = \boldsymbol{D}\boldsymbol{y}$. This can be done in $O(n)$ time owing to the bidiagonal structure of $\boldsymbol{D}$. Conversely, if the solution to $\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} = \boldsymbol{D}\boldsymbol{y}$ lies in the interior of the ball $\|\boldsymbol{u}\|_2 \leq \lambda$, then it solves (3.2). Therefore, this case is trivial, and we need to consider only the harder case $\|\boldsymbol{u}\|_2 = \lambda$.

For any given α one can obtain the corresponding vector $\boldsymbol{u}$ as $\boldsymbol{u}_\alpha = (\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{D}\boldsymbol{y}$. Therefore, optimizing for $\boldsymbol{u}$ reduces to the problem of finding the "true" value of α.

An obvious approach is to solve $\|\boldsymbol{u}_\alpha\|_2^2 = \lambda^2$. Less obvious is the *MSN equation*

$$h_\alpha := \lambda^{-1} - \|\boldsymbol{u}_\alpha\|_2^{-1} = 0, \tag{3.3}$$

which has the benefit of being almost linear in the search interval, which results in fast convergence (Moré and Sorensen, 1983). Thus, the task is to find the root of the function h_α, for which we use Newton's method, which in this case leads to the iteration

$$\alpha \leftarrow \alpha - h_\alpha / h'_\alpha. \tag{3.4}$$

Some calculation shows that the derivative h' can be computed as

$$\frac{1}{h'_\alpha} = \frac{\|\boldsymbol{u}_\alpha\|_2^3}{\boldsymbol{u}_\alpha^T(\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{u}_\alpha}. \tag{3.5}$$

The key idea in MSN is to eliminate the matrix inverse in (3.5) by using the Cholesky decomposition $\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I} = \boldsymbol{R}_\alpha^T\boldsymbol{R}_\alpha$ and defining a vector $\boldsymbol{q}_\alpha = (\boldsymbol{R}_\alpha^T)^{-1}\boldsymbol{u}$, so that $\|\boldsymbol{q}_\alpha\|_2^2 = \boldsymbol{u}_\alpha^T(\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{u}_\alpha$. As a result, the Newton iteration (3.4) becomes

$$\begin{aligned} \alpha - \frac{h_\alpha}{h'_\alpha} &= \alpha - (\|\boldsymbol{u}_\alpha\|_2^{-1} - \lambda^{-1}) \cdot \frac{\|\boldsymbol{u}_\alpha\|_2^3}{\boldsymbol{u}_\alpha^T(\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})^{-1}\boldsymbol{u}_\alpha}, \\ &= \alpha - \frac{\|\boldsymbol{u}_\alpha\|_2^2 - \lambda^{-1}\|\boldsymbol{u}_\alpha\|_2^3}{\|\boldsymbol{q}_\alpha\|_2^2}, \\ &= \alpha - \frac{\|\boldsymbol{u}_\alpha\|_2^2}{\|\boldsymbol{q}_\alpha\|_2^2}\left(1 - \frac{\|\boldsymbol{u}_\alpha\|_2}{\lambda}\right), \end{aligned}$$

and therefore

$$\alpha \quad \leftarrow \quad \alpha - \frac{\|\boldsymbol{u}_\alpha\|_2^2}{\|\boldsymbol{q}_\alpha\|_2^2}\left(1 - \frac{\|\boldsymbol{u}_\alpha\|_2}{\lambda}\right). \tag{3.6}$$

As shown for TV-L$_1$ (Appendix E), the tridiagonal structure of $(\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I})$ allows one to compute both $\boldsymbol{R}_\alpha$ and $\boldsymbol{q}_\alpha$ in linear time, so the overall iteration runs in $O(n)$ time.

Algorithm 4 MSN based TV-L2 proximity

Initialize: $\alpha = 0$, $\boldsymbol{u}_\alpha = 0$.
while $\left|\|\boldsymbol{u}_\alpha\|_2^2 - \lambda\right| > \epsilon_\lambda$ **or** $\text{gap}(\boldsymbol{u}_\alpha) > \epsilon_{\text{gap}}$ **do**
  Compute Cholesky decomp. $\boldsymbol{D}\boldsymbol{D}^T + \alpha\boldsymbol{I} = \boldsymbol{R}_\alpha^T\boldsymbol{R}_\alpha$.
  Obtain $\boldsymbol{u}_\alpha$ by solving $\boldsymbol{R}_\alpha^T\boldsymbol{R}_\alpha\boldsymbol{u}_\alpha = \boldsymbol{D}\boldsymbol{y}$.
  Obtain $\boldsymbol{q}_\alpha$ by solving $\boldsymbol{R}_\alpha^T\boldsymbol{q}_\alpha = \boldsymbol{u}_\alpha$.
  $\alpha = \alpha - \frac{\|\boldsymbol{u}_\alpha\|_2^2}{\|\boldsymbol{q}_\alpha\|_2^2}\left(1 - \frac{\|\boldsymbol{u}_\alpha\|_2}{\lambda}\right)$.
end while
return $\boldsymbol{u}_\alpha$

Algorithm 5 GP algorithm for TV-L$_2$ proximity

Initialize $\boldsymbol{u}^0 \in \mathbb{R}^N$, $t = 0$.
while ($\neg$ converged) **do**
  Gradient update: $\boldsymbol{v}^t = \boldsymbol{u}^t - \frac{1}{4}\nabla f(\boldsymbol{u}^t)$.
  Projection: $\boldsymbol{u}^{t+1} = \max(1 - \lambda/\|\boldsymbol{v}^t\|_2, 0) \cdot \boldsymbol{v}^t$.
  $t \leftarrow t + 1$.
end while
return $\boldsymbol{u}^t$.

The above ideas are presented as pseudocode in Algorithm 4. As a stopping criterion two conditions are checked: whether the duality gap is small enough, and whether $\boldsymbol{u}$ is close enough to the boundary. This latter check is useful because intermediate solutions could be dual-infeasible, thus making the duality gap an inadequate optimality measure on its own. In practice we use tolerance values $\epsilon_\lambda = 10^{-6}$ and $\epsilon_{\text{gap}} = 10^{-5}$.

Even though Algorithm 4 requires only linear time per iteration, it is fairly sophisticated, and in fact a much simpler method can be devised. This is illustrated here by a gradient-projection method with a *fixed* stepsize α_0, whose iteration is

$$\boldsymbol{u}^{t+1} = P_{\|\cdot\|_2 \leq \lambda}(\boldsymbol{u}^t - \alpha_0\nabla\phi(\boldsymbol{u}^t)). \tag{3.7}$$

The theoretically ideal choice for the stepsize α_0 is given by the inverse of the Lipschitz constant L of the gradient $\nabla\phi(\boldsymbol{u})$ (Nesterov, 2007; Beck and Teboulle, 2009). Since $\phi(\boldsymbol{u})$ is a convex quadratic, L is simply the largest eigenvalue of the Hessian $\boldsymbol{D}\boldsymbol{D}^T$. Owing to its special structure, the eigenvalues of the Hessian have closed-form expressions, namely $\lambda_i = 2 - 2\cos\left(\frac{i\pi}{n+1}\right)$ (for $1 \leq i \leq n$). The largest one is $\lambda_n = 2 - 2\cos\left(\frac{(n-1)\pi}{n}\right)$, which tends to 4 as $n \to \infty$; thus the choice $\alpha_0 = 1/4$ is a good and cheap approximation. Pseudocode showing the whole procedure is presented in Algorithm 5. Combining this with the fact that the projection $P_{\|\cdot\|_2 \leq \lambda}$ is also trivial to compute, the GP iteration (3.7) turns out to be very attractive. Indeed, sometimes it can even outperform the more sophisticated MSN method, though only for a very limited range of λ values. Therefore, in practice we recommend a hybrid of GP and MSN, as suggested by our experiments (see §5.2.1).

3.2. TV-Lp: Proximity for $\mathbf{Tv}_p^{1D}$

For TV-L_p proximity (for $1 < p < \infty$) the dual problem becomes

$$\min_{\boldsymbol{u}} \ \phi(\boldsymbol{u}) := \tfrac{1}{2}\|\boldsymbol{D}^T\boldsymbol{u}\|_2^2 - \boldsymbol{u}^T\boldsymbol{D}\boldsymbol{y}, \ \text{ s.t. } \|\boldsymbol{u}\|_q \leq \lambda, \tag{3.8}$$

where $q = 1/(1-1/p)$. Problem (3.8) is not particularly amenable to Newton-type approaches, as neither PN (Appendix E), nor MSN-type methods (§3.1) can be applied easily. It is partially amenable to gradient-projection (GP), for which the same update rule as in (3.7) applies, but unlike the $q = 2$ case, the projection step here is much more involved. Thus, to complement GP, we may favor the projection-free Frank-Wolfe (FW) method. As expected, the overall best performing approach is actually a hybrid of GP and FW. We summarize both choices below.

3.2.1. Efficient Projection onto the ℓ_q-ball

The problem of projecting onto the ℓ_q-norm ball is

$$\min_{\boldsymbol{w}} \quad d(\boldsymbol{w}) := \tfrac{1}{2}\|\boldsymbol{w}-\boldsymbol{u}\|_2^2, \quad \text{s.t. } \ \|\boldsymbol{w}\|_q \leq \lambda. \tag{3.9}$$

For this problem, it turns out to be more convenient to address its Fenchel dual

$$\min_{\boldsymbol{w}} \quad d^*(\boldsymbol{w}) := \tfrac{1}{2}\|\boldsymbol{w}-\boldsymbol{u}\|_2^2 + \lambda\|\boldsymbol{w}\|_p, \tag{3.10}$$

which is actually nothing but $\operatorname{prox}_{\lambda\|\cdot\|_p}(\boldsymbol{u})$. The optimal solution, say $\boldsymbol{w}^*$, to (3.9) can be obtained by solving (3.10), by using the Moreau-decomposition (A.6) which yields

$$\boldsymbol{w}^* = \boldsymbol{u} - \operatorname{prox}_{\lambda\|\cdot\|_p}(\boldsymbol{u}).$$

Projection (3.9) is computed many times within GP, so it is crucial to solve it rapidly and accurately. To this end, we first turn (3.10) into a differentiable problem and then derive a projected-Newton method following our approach presented in Appendix E.

Assume therefore, without loss of generality that $\boldsymbol{u} \geq 0$, so that $\boldsymbol{w} \geq 0$ also holds (the signs can be restored after solving this problem). Thus, instead of (3.10), we solve

$$\min_{\boldsymbol{w}} \quad d^*(\boldsymbol{w}) := \tfrac{1}{2}\|\boldsymbol{w}-\boldsymbol{u}\|_2^2 + \lambda\big(\textstyle\sum_i w_i^p\big)^{1/p} \quad \text{s.t. } \boldsymbol{w} \geq 0. \tag{3.11}$$

The gradient of d^* may be compactly written as

$$\nabla d^*(\boldsymbol{w}) = \boldsymbol{w} - \boldsymbol{u} + \lambda\|\boldsymbol{w}\|_p^{1-p}\boldsymbol{w}^{p-1}, \tag{3.12}$$

where $\boldsymbol{w}^{p-1}$ denotes elementwise exponentiation of $\boldsymbol{w}$. Elementary calculation yields

$$\begin{aligned}\tfrac{\partial^2}{\partial w_i \partial w_j} d^*(\boldsymbol{w}) &= \delta_{ij}\big(1 + \lambda(p-1)\big(\tfrac{w_i}{\|w\|_p}\big)^{p-2}\|\boldsymbol{w}\|_p^{-1}\big) + \lambda(1-p)\big(\tfrac{w_i}{\|w\|_p}\big)^{p-1}\big(\tfrac{w_j}{\|w\|_p}\big)^{p-1}\|\boldsymbol{w}\|_p^{-1} \\ &= \delta_{ij}\big(1 - c\hat{\boldsymbol{w}}_i^{p-2}\big) + c\bar{\boldsymbol{w}}_i\bar{\boldsymbol{w}}_j,\end{aligned}$$

where $c := \lambda(1-p)\|\boldsymbol{w}\|_p^{-1}$, $\hat{\boldsymbol{w}} := \boldsymbol{w}/\|\boldsymbol{w}\|_p$, $\bar{\boldsymbol{w}} := (\boldsymbol{w}/\|\boldsymbol{w}\|_p)^{p-1}$, and δ_{ij} is the Dirac delta. In matrix notation, this Hessian's diagonal plus rank-1 structure becomes apparent

$$\boldsymbol{H}(\boldsymbol{w}) = \operatorname{Diag}\big(1 - c\hat{\boldsymbol{w}}^{p-2}\big) + c\bar{\boldsymbol{w}}\cdot\bar{\boldsymbol{w}}^T \tag{3.13}$$

To develop an efficient Newton method it is imperative to exploit this structure. It is not hard to see that for a set of non-active variables $\bar{I}$ the reduced Hessian takes the form

$$\boldsymbol{H}_{\bar{I}}(\boldsymbol{w}) = \operatorname{Diag}\big(\mathbf{1} - c\hat{\boldsymbol{w}}_{\bar{I}}^{p-2}\big) + c\bar{\boldsymbol{w}}_{\bar{I}}\bar{\boldsymbol{w}}_{\bar{I}}^T. \tag{3.14}$$

With the shorthand $\Delta = \operatorname{Diag}\big(\mathbf{1} - c\hat{\boldsymbol{w}}_{\bar{I}}^{p-2}\big)$, the matrix-inversion lemma yields

$$\boldsymbol{H}_{\bar{I}}^{-1}(\boldsymbol{w}) = \big(\Delta + c\bar{\boldsymbol{w}}_{\bar{I}}\bar{\boldsymbol{w}}_{\bar{I}}^T\big)^{-1} = \Delta^{-1} - \frac{\Delta^{-1}c\bar{\boldsymbol{w}}_{\bar{I}}\bar{\boldsymbol{w}}_{\bar{I}}^T\Delta^{-1}}{1 + c\bar{\boldsymbol{w}}_{\bar{I}}^T\Delta^{-1}\bar{\boldsymbol{w}}_{\bar{I}}}. \tag{3.15}$$

Furthermore, since in PN the inverse of the reduced Hessian always operates on the reduced gradient, we can rearrange the terms in this operation for further efficiency; that is,

$$\boldsymbol{H}_{\bar{I}}(\boldsymbol{w})^{-1}\nabla_{\bar{I}}f(\boldsymbol{w}) = \boldsymbol{v}\odot\nabla_{\bar{I}}f(\boldsymbol{w}) - \frac{\big(\boldsymbol{v}\odot\bar{\boldsymbol{w}}_{\bar{I}}\big)\big(\boldsymbol{v}\odot\bar{\boldsymbol{w}}_{\bar{I}}\big)^T\nabla_{\bar{I}}f(\boldsymbol{w})}{1/c + \bar{\boldsymbol{w}}_{\bar{I}}\big(\boldsymbol{v}\odot\bar{\boldsymbol{w}}_{\bar{I}}\big)}, \tag{3.16}$$

where $\boldsymbol{v} := \big(\mathbf{1} - c\hat{\boldsymbol{w}}_{\bar{I}}^{p-2}\big)^{-1}$, and $\odot$ denotes componentwise product.

The relevant point of the above derivations is that the Newton direction, and thus the overall PN iteration can be computed in $O(n)$ time, which results in a highly effective solver.

3.2.2. Frank-Wolfe Algorithm for TV-L_p Proximity

The Frank-Wolfe (FW) algorithm (see e.g., Jaggi (2013) for a recent overview), also known as the conditional gradient method (Bertsekas, 1999) solves differentiable optimization problems over compact convex sets, and can be quite effective if we have access to a subroutine to solve linear problems over the constraint set.

The generic FW iteration is illustrated in Algorithm 6. FW offers an attractive strategy for TV-L_p because both the descent-direction as well as stepsizes can be computed easily. Specifically, to find the descent direction we need to solve

$$\min_{\boldsymbol{s}} \quad \boldsymbol{s}^T\left(\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} - \boldsymbol{D}\boldsymbol{y}\right), \quad \text{s.t.} \quad \|\boldsymbol{s}\|_q \leq \lambda. \tag{3.17}$$

This problem can be solved by observing that $\max_{\|\boldsymbol{s}\|_q\leq 1}\boldsymbol{s}^T\boldsymbol{z}$ is attained by some vector $\boldsymbol{s}$ proportional to $\boldsymbol{z}$, of the form $|\boldsymbol{s}^*| \propto |\boldsymbol{z}|^{p-1}$. Therefore, $\boldsymbol{s}^*$ in (3.17) is found by taking $\boldsymbol{z} = \boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} - \boldsymbol{D}\boldsymbol{y}$, computing $\boldsymbol{s} = -\operatorname{sgn}(\boldsymbol{z})\odot|\boldsymbol{z}|^{p-1}$ and then rescaling $\boldsymbol{s}$ to meet $\|\boldsymbol{s}\|_q = \lambda$.

Algorithm 6 Frank-Wolfe (FW)

Inputs: f, compact convex set $\mathcal{D}$.
Initialize $\boldsymbol{x}_0 \in \mathcal{D}$, $t = 0$.
while stopping criteria not met **do**
  Find descent direction: $\min_{\boldsymbol{s}} \boldsymbol{s}\cdot\nabla f(\boldsymbol{x}_t)$ s.t. $\boldsymbol{s}\in\mathcal{D}$.
  Determine stepsize: $\min_\gamma f(\boldsymbol{x}_t + \gamma(\boldsymbol{s}-\boldsymbol{x}_t))$ s.t. $\gamma\in[0,1]$.
  Update: $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t + \gamma(\boldsymbol{s}-\boldsymbol{x}_t)$
  $t \leftarrow t+1$.
end while
return $\boldsymbol{x}_t$.

The stepsize can also be computed in closed form owing to the objective function being quadratic. Note the update in FW takes the form $\boldsymbol{u} + \gamma(\boldsymbol{s} - \boldsymbol{u})$, which can be rewritten as $\boldsymbol{u} + \gamma\boldsymbol{d}$ with $\boldsymbol{d} = \boldsymbol{s} - \boldsymbol{u}$. Using this notation the optimal stepsize is obtained by solving

$$\min_{\gamma\in[0,1]} \tfrac{1}{2}\|\boldsymbol{D}^T(\boldsymbol{u} + \gamma\boldsymbol{d})\|_2^2 - (\boldsymbol{u} + \gamma\boldsymbol{d})^T\boldsymbol{D}\boldsymbol{y}.$$

A brief calculation on the above problem yields

$$\gamma^* = \min\left\{\max\left\{\hat{\gamma}, 1\right\}, 0\right\},$$

where $\hat{\gamma} = -(\boldsymbol{d}^T\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{u} + \boldsymbol{d}^T\boldsymbol{D}\boldsymbol{y})/(\boldsymbol{d}^T\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{d})$ is the unconstrained optimal stepsize. We note that following (Jaggi, 2013) we also check a "surrogate duality-gap"

$$g(\boldsymbol{x}) = \boldsymbol{x}^T\nabla f(\boldsymbol{x}) - \min_{\boldsymbol{s}\in\mathcal{D}} \boldsymbol{s}^T\nabla f(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{s}^*)^T\nabla f(\boldsymbol{x}),$$

at the end of each iteration. If this gap is smaller than the desired tolerance, the real duality gap is computed and checked; if it also meets the tolerance, the algorithm stops.

3.3. Prox Operator for TV-L$_\infty$

The final case is $\text{Tv}_\infty^{\text{1D}}$ proximity. We mention this case only for completeness. The dual to the prox-operator here is

$$\min_{\boldsymbol{u}} \quad \tfrac{1}{2}\|\boldsymbol{D}^T\boldsymbol{u}\|_2^2 - \boldsymbol{u}^T\boldsymbol{D}\boldsymbol{y}, \quad \text{s.t. } \|\boldsymbol{u}\|_1 \leq \lambda. \tag{3.18}$$

This problem can be again easily solved by invoking GP, where the only non-trivial step is projection onto the ℓ_1-ball. But the latter is an extremely well-studied operation (see e.g., Condat (2016); Liu and Ye (2009); Kiwiel (2008)), and so $O(n)$ time routines for this purpose are readily available. By integrating them in our GP framework an efficient prox solver is obtained.

4. Prox Operators for Multidimensional TV

We now move onto discussing how to use the efficient 1D-TV prox operators derived above within a prox-splitting framework to handle multidimensional TV (1.3) proximity.

4.1. Proximity Stacking

The basic composite objective (1.1) is a special case of the more general class of models where one may have several regularizers, so that we now solve

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + \sum\nolimits_{i=1}^{m} r_i(\boldsymbol{x}), \tag{4.1}$$

where each r_i (for $1 \leq i \leq m$) is lsc and convex.

Just like the basic problem (1.1), the more complex problem (4.1) can also be tackled via proximal methods. The key to doing so is to use *inexact proximal methods* along with a technique we should call **proximity stacking**. Inexact proximal methods allow one to use approximately computed prox operators without impeding overall convergence, while

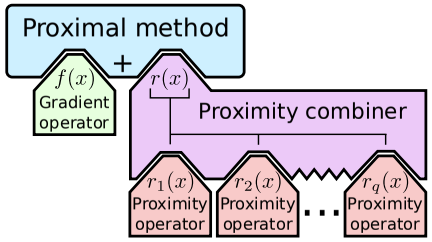


Figure 6: Design schema in proximal optimization for minimizing the function $f(\boldsymbol{x}) + \sum_{i=1}^m r_i(\boldsymbol{x})$. Proximal stacking makes the sum of regularizers appear as a single one to the proximal method, while retaining modularity in the design of each proximity step through the use of a combiner method. For non-smooth f the same schema applies by just replacing the f gradient operator by its corresponding proximity operator.

proximity stacking allows one to compute the prox operator for the entire sum $r(\boldsymbol{x}) = \sum_{i=1}^m r_i(\boldsymbol{x})$ by "stacking" the individual r_i prox operators. This stacking leads to a highly modular design; see Figure 6 for a visualization. In other words, proximity stacking involves computing the prox operator

$$\text{prox}_r(\boldsymbol{y}) := \operatorname*{argmin}_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum\nolimits_{i=1}^{m} r_i(\boldsymbol{x}), \tag{4.2}$$

by iteratively invoking the individual prox operators prox_{r_i} and then combining their outputs. This mixing is done by means of a combiner method, which guarantees convergence to the solution of the overall $\text{prox}_r(\boldsymbol{y})$.

Different proximal combiners can used for computing prox_r (4.2). In what follows we briefly describe some of the possibilities. The crux of all of them is that their key steps will be proximity steps over the individual r_i terms. Thus, using proximal stacking and combination, any convex machine learning problem with multiple regularizers can be solved in a highly modular proximal framework. After this section we exemplify these ideas by applying them to two- and higher-dimensional TV proximity, which we then use within proximal solvers for addressing a wide array of applications.

4.1.1. Proximal Dykstra (PD)

The Proximal Dykstra method (Combettes and Pesquet, 2009) solves problems of the form

$$\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}),$$

which is a particular case of (4.2) for $m = 2$. The method follows the procedure detailed in Algorithm 7, which is guaranteed to converge to the desired solution. Using PD for proximal stacking for 2D Total-Variation was previously proposed in (Barbero and Sra, 2011).

It has also been shown that the application of this method is equivalent to performing alternating projections onto certain dual polytopes (Jegelka et al., 2013), a procedure whose

Algorithm 7 Proximal Dykstra

Inputs: r_1, r_2, input signal $\boldsymbol{y} \in \mathbb{R}^n$.
Initialize $\boldsymbol{x}_0 = \boldsymbol{y}$, $\boldsymbol{p}_0 = \boldsymbol{q}_0 = 0$, $t = 0$.
while stopping criteria not met **do**
  r_2 proximity operator: $\boldsymbol{z}_t = \text{prox}_{r_2}(\boldsymbol{x}_t + \boldsymbol{p}_t)$.
  r_2 step: $\boldsymbol{p}_{t+1} = \boldsymbol{x}_t + \boldsymbol{p}_t - \boldsymbol{z}_t$.
  r_1 proximity operator: $\boldsymbol{x}_{t+1} = \text{prox}_{r_1}(\boldsymbol{z}_t + \boldsymbol{q}_t)$.
  r_1 step: $\boldsymbol{q}_{t+1} = \boldsymbol{z}_t + \boldsymbol{q}_t - \boldsymbol{x}_{t+1}$.
  $t \leftarrow t + 1$.
end while
Return $\boldsymbol{x}_t$.

Algorithm 8 Parallel-Proximal Dykstra

Inputs: $r_1, \ldots, r_m$, input signal $\boldsymbol{y} \in \mathbb{R}^n$.
Initialize $\boldsymbol{x}_0 = \boldsymbol{y}$, $\boldsymbol{z}_0^i = 0$, for $i = 1, \ldots, m$; $t = 0$
while stopping criterion not met **do**
  for $i = 1$ to m in *parallel* **do**
    $\boldsymbol{p}_t^i = \text{prox}_{r_i}(\boldsymbol{z}_t^i)$
  end for
  $\boldsymbol{x}_{t+1} = \frac{1}{m}\sum_i \boldsymbol{p}_t^i$
  for $i = 1$ to m in *parallel* **do**
    $\boldsymbol{z}_{t+1}^i = \boldsymbol{x}_{t+1} + \boldsymbol{z}_t^i - \boldsymbol{p}_t^i$
  end for
  $t \leftarrow t + 1$
end while
Return $\boldsymbol{x}_t$

effectiveness varies depending on the relative orientation of such polytopes. A more efficient method based on reflections instead of projections is possible, as we will see below.

More generally, if more than two regularizers are present (i.e., $m > 2$), then it is more fitting to use *Parallel-Proximal Dykstra* (PPD) (Combettes, 2009) (see Alg. 8), a generalization obtained via the "product-space trick" of Pierra (1984). This parallel proximal method is attractive because it not only combines an arbitrary number of regularizers, but also allows parallelizing the calls to the individual prox operators. This feature allows us to develop a highly parallel implementation for multidimensional TV proximity (§4.3).

4.1.2. Alternating Reflections – Douglas-Rachford (DR)

The Douglas-Rachford (DR) method was originally devised for minimizing the sum of two (nonsmooth) convex functions (Combettes and Pesquet, 2009), in the form:

$$\min_{\boldsymbol{x}} \quad f_1(\boldsymbol{x}) + f_2(\boldsymbol{x}), \tag{4.3}$$

such that $(\text{ri dom } f_1) \cap (\text{ri dom } f_2) \neq \emptyset$. The method operates by iterating a series of reflections, and in its simplest form can be written as

$$\boldsymbol{z}_{k+1} = \tfrac{1}{2}\left[R_{f_1} R_{f_2} + I\right] \boldsymbol{z}_k, \tag{4.4}$$

where the *reflection operator* $R_\phi := 2\,\text{prox}_\phi - I$. This method is not cleanly applicable to problem (4.2) because of the squared norm term. Nevertheless in (Jegelka et al., 2013) a suitable transformation was proposed by making use of arguments from submodular optimization; a minimal background on this topic is given in Appendix A. We summarize the key ideas from (Jegelka et al., 2013) below.

Assume $m = 2$ and r_1, r_2 being Lovász extensions to some submodular functions (Total-Variation is the Lovász extension of a submodular graph-cut problem, see Bach (2013)). Defining $\hat{r}_1(\boldsymbol{x}) = r_1(\boldsymbol{x}) - \boldsymbol{x}^T\boldsymbol{y}$, $\hat{r}_1$ is also a Lovász extension of some submodular function (see Appendix A). Therefore, we may consider the problem

$$\text{prox}_r(\boldsymbol{y}) := \operatorname*{argmin}_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x}\|_2^2 + \hat{r}_1(\boldsymbol{x}) + r_2(\boldsymbol{x}),$$

which can be rewritten (using Proposition A.11) as

$$\min_{a,b} \|a - b\|_2, \quad \text{s.t.} \quad a \in -B_{\hat{r}_1},\, b \in B_{r_2}, \tag{4.5}$$

where B_r denotes the base polytope of submodular function corresponding to r (see Appendix A). The original solution can be recovered through $\boldsymbol{x} = \boldsymbol{a} - \boldsymbol{b}$. Problem (4.5) is still not in a form amenable to DR (4.3)—nevertheless, if we apply DR to the indicator functions of the sets $-B_{\hat{r}_1}, B_{r_2}$, that is, to the problem

$$\min_{\boldsymbol{x}} \quad \delta_{-B_{\hat{r}_1}}(\boldsymbol{x}) + \delta_{B_{r_2}}(\boldsymbol{x}),$$

it can be shown (Bauschke, 2004) that the sequence (4.4) generated by DR is divergent, but that after a correction through projection converges to the desired solution of (4.5). Such solution is given by the pair

$$\boldsymbol{b} = \Pi_{B_{r_2}}(\boldsymbol{z}_k), \quad \boldsymbol{a} = \Pi_{-B_{\hat{r}_1}}(\boldsymbol{b}). \tag{4.6}$$

Although in this derivation many concepts have been introduced, suprisingly all the operations in the algorithm can be reduced to performing proximity steps. Note first that the projections onto a base polytope required to get a solution (4.6) can be written in terms of proximity operators (Proposition A.12), which in this case implies

$$\Pi_{B_{r_2}}(\boldsymbol{z}) = \boldsymbol{z} - \text{prox}_{r_2}(\boldsymbol{z}),$$
$$\Pi_{-B_{\hat{r}_1}}(\boldsymbol{z}) = \boldsymbol{z} + \text{prox}_{\hat{r}_2}(-\boldsymbol{z}) = \boldsymbol{z} + \text{prox}_{r_2}(-\boldsymbol{z} + \boldsymbol{y}),$$

where we use the fact that for $f(\boldsymbol{x}) = \phi(\boldsymbol{x}) + \boldsymbol{u}^T\boldsymbol{x}$, $\text{prox}_f(\boldsymbol{x}) = \text{prox}_\phi(\boldsymbol{x} - \boldsymbol{u})$. The reflection operations in which the DR iteration is based (4.4) can also be written in terms of proximity steps, as we are applying DR to the indicator functions $\delta_{-B_{\hat{r}_1}}, \delta_{B_{r_2}}$, and proximity for an indicator function equals projection.

This alternating reflections variant of DR is presented in Algorithm 9. Note that in contrast with the original DR method, this variant does not require tuning any hyperparameters, thus enhancing its practicality.

Algorithm 9 Alternating reflections – Douglas Rachford (DR)

Inputs: r_1, r_2 Lovász extensions of some submodular function, input signal $\boldsymbol{y} \in \mathbb{R}^n$.
Initialize $\boldsymbol{z}_0 \in \mathbb{R}^n$, $t = 0$.
Define the following operations:
$\Pi_{-B_{\hat{r}_1}}(\boldsymbol{z}) \stackrel{\text{def}}{=} \boldsymbol{z} + \text{prox}_{r_1}(-\boldsymbol{z} + \boldsymbol{y})$.
$\Pi_{B_{r_2}}(\boldsymbol{z}) \stackrel{\text{def}}{=} \boldsymbol{z} - \text{prox}_{r_2}(\boldsymbol{z})$.
$R_{-B_{\hat{r}_1}}(\boldsymbol{z}) \stackrel{\text{def}}{=} 2\Pi_{-B_{\hat{r}_1}}(\boldsymbol{z}) - \boldsymbol{z}$.
$R_{B_{r_2}}(\boldsymbol{z}) \stackrel{\text{def}}{=} 2\Pi_{B_{r_2}}(\boldsymbol{z}) - \boldsymbol{z}$.
while stopping criteria not met **do**
$\boldsymbol{z}_{t+1} = \frac{1}{2}\left[R_{-B_{\hat{r}_1}} R_{B_{r_2}} + I\right] \boldsymbol{z}_k$
$t \leftarrow t + 1$.
end while
$\boldsymbol{b} = \Pi_{B_{r_2}}(z_t), \quad \boldsymbol{a} = \Pi_{-B_{\hat{r}_1}}(\boldsymbol{b})$.
Return $\boldsymbol{x}^* = \boldsymbol{a} - \boldsymbol{b}$.

4.1.3. Alternating-Direction Method of Multipliers (ADMM)

Although many times presented as a particular algorithm for solving problems involving the minimization of a certain objetive $f(x) + g(Lx)$ with L a linear operator (Combettes and Pesquet, 2009), the Alternating-Direction Method of Multipliers can be thought as a general splitting strategy for solving the unconstrained minimization of a sum of functions. This strategy boils down to transforming a problem in the form $\min_{\boldsymbol{x}} \sum_{i=1}^m f_i(\boldsymbol{x})$ into a saddle-point problem by introducing consensus constraints and incorporating them into the objective through augmented Lagrange multipliers,

$$\begin{aligned}\min_{\boldsymbol{x}} \quad \sum_{i=1}^m f_i(\boldsymbol{x}) &= \min_{\boldsymbol{x},\boldsymbol{z}_1,\ldots,\boldsymbol{z}_m} \quad \sum_{i=1}^m f_i(\boldsymbol{z}_i) \quad \text{s.t. } \boldsymbol{z}_1 = \boldsymbol{x}, \ldots, \boldsymbol{z}_m = \boldsymbol{x}, \\ &\equiv \min_{\boldsymbol{x},\boldsymbol{z}_1,\ldots,\boldsymbol{z}_m} \max_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m} \quad \sum_{i=1}^m \left(f_i(\boldsymbol{z}_i) + \boldsymbol{u}_i^T(\boldsymbol{z}_i - \boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{z}_i - \boldsymbol{x}\|_2 \right).\end{aligned}$$

The method then proceeds to solve this problem by alternating steps of minimization on $\boldsymbol{x}$, minimization on every $\boldsymbol{z}_i$, and a gradient step on every $\boldsymbol{u}_i$.

In (Yang et al., 2013) a proposal using this method was presented to solve m–dimensional anisotropic TV (1.3). This approach applies equally to the more general proximal stacking framework under discussion here (4.2), by the transformation

$$\begin{aligned}\text{prox}_r(\boldsymbol{y}) &:= \operatorname*{argmin}_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \textstyle\sum_{i=1}^m r_i(\boldsymbol{x}), \\ &\equiv \min_{\boldsymbol{x},\boldsymbol{z}_1,\ldots,\boldsymbol{z}_m} \max_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_m} \quad \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \sum_{i=1}^m \left(f_i(\boldsymbol{z}_i) + \boldsymbol{u}_i^T(\boldsymbol{z}_i - \boldsymbol{x}) + \frac{\rho}{2}\|\boldsymbol{z}_i - \boldsymbol{x}\|_2 \right).\end{aligned}$$

The steps for obtaining a solution then follow as Algorithm 10. Similar to Parallel Proximal Dykstra, this approach allows computing the prox-operator of each function r_i in parallel.

Algorithm 10 Alternating Direction Method of Multipliers (ADMM)

Inputs: $r_1, \ldots, r_m$, input signal $\boldsymbol{y} \in \mathbb{R}^n$.
Initialize $\boldsymbol{x}_0 = \boldsymbol{z}_0^i = \boldsymbol{y}$ for $i = 1, \ldots, m$; $t = 0$
while stopping criterion not met **do**
$\boldsymbol{x}_{t+1} = \frac{\boldsymbol{y} + \sum_{i=1}^m (\boldsymbol{u}_t^i + \rho \boldsymbol{z}_t^i)}{1 + m\rho}$.
for $i = 1$ to m in *parallel* **do**
$\boldsymbol{z}_t^i = \text{prox}_{\frac{\lambda}{\rho} r_i}(-\frac{1}{\rho}\boldsymbol{u}_t^i + \boldsymbol{x}_{t+1})$
$\boldsymbol{u}_{t+1}^i = \boldsymbol{u}_{t+1} + \rho(\boldsymbol{z}_{t+1}^i - \boldsymbol{x}_{t+1})$
end for
$t \leftarrow t + 1$
end while
Return $\boldsymbol{x}_t$

4.1.4. Dual Proximity Methods

Another family of approaches to solve (4.2) is to compute the global proximity operator using the Fenchel duals $\text{prox}_{r_i^*}$. This can be advantageous in settings where the dual prox-operator is easier to compute than the primal operator; isotropic Total-Variation problems are an instance of such a setting, and thus investigating this approach for their anisotropic variants is worthwhile.

Indeed, in the context of image processing a popular splitting approach is given by Chambolle and Pock (2011), which consider a problem in the form

$$\min_{\boldsymbol{x}} \quad F(\boldsymbol{K}\boldsymbol{x}) + G(\boldsymbol{x}),$$

for $\boldsymbol{K}$ some linear operator, F, G convex lower-semicontinuous functions. Through a strategy similar to ADMM an equivalent saddle point problem can be obtained,

$$\min_{\boldsymbol{x}} \max_{\boldsymbol{y}} \quad (\boldsymbol{K}\boldsymbol{x})^T\boldsymbol{y} + G(\boldsymbol{x}) - F^*(\boldsymbol{y}),$$

with F^* convex conjugate of F. This problem is then solved by alternating maximization on $\boldsymbol{y}$ and minimization on $\boldsymbol{x}$ through proximity steps, as

$$\begin{aligned}\boldsymbol{y}_{t+1} &= \text{prox}_{\sigma F^*}(\boldsymbol{y}_t + \sigma \boldsymbol{K}\bar{\boldsymbol{x}}_t) \\ \boldsymbol{x}_{t+1} &= \text{prox}_{\tau G}(\boldsymbol{x}_t - \tau \boldsymbol{K}^*\boldsymbol{y}_{t+1}) \\ \bar{\boldsymbol{x}}_{t+1} &= \boldsymbol{x}_{t+1} + \theta(\boldsymbol{x}_{t+1} - \boldsymbol{x}_t),\end{aligned}$$

where $\boldsymbol{K}^*$ is the conjugate transpose of $\boldsymbol{K}$. σ, τ and θ are algorithm parameters that should be either selected under some bounds (Chambolle and Pock, 2011, Algorithm 1) or readjusted every iteration making use of Lipschitz convexity of G (Chambolle and Pock, 2011, Algorithm 2), resulting in an accelerating scheme much in the style of FISTA (Beck and Teboulle, 2009). The overall procedure can also be shown to be an instance of preconditioned ADMM, where the preconditioning is given by the application of a proximity step for the maximization of $\boldsymbol{y}$ (instead of the usal dual gradient step of ADMM) and the auxiliary point $\bar{\boldsymbol{x}}$. Note also how proximity is computed over the dual F^* instead of the primal prox_F.

Now, this decomposition strategy can be applied for some instances of proximal stacking (4.2) when the r_i terms allow the particular composition

$$\sum_{i=1}^{m} r_i(\boldsymbol{x}) = F\left(\begin{bmatrix} \boldsymbol{K}_1 \\ \vdots \\ \boldsymbol{K}_m \end{bmatrix} \boldsymbol{x}\right) = F(\boldsymbol{K}\boldsymbol{x}),$$

which does not hold in general but holds for 2D TV (1.4) when taking the identities

$$F(\boldsymbol{x}) = \|\boldsymbol{x}\|_1, G(\boldsymbol{x}) = \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2,$$
$$\boldsymbol{K} = \begin{bmatrix} \boldsymbol{I}\otimes\boldsymbol{D} \\ \boldsymbol{D}\otimes\boldsymbol{I} \end{bmatrix},$$

with $\boldsymbol{D}$ the differencing matrix as before, $\otimes$ denotes Kronecker product, and $\boldsymbol{x}$ a vectorization of the 2D input. The iterates above can then be applied easily: proximity over G is trivial and proximity over F^* is also easy upon realizing that $\operatorname{prox}_{\|\cdot\|_1^*} = \operatorname{prox}_{\delta_{\|\cdot\|_\infty \le 1}} = \Pi_{\|\cdot\|_\infty \le 1}$, which is solved through thresholding.

A generalization of this approach is presented by Condat (2014), who considers

$$\min_{\boldsymbol{x}} \quad f(\boldsymbol{x}) + g(\boldsymbol{x}) + \sum_{i=1}^{m} r_i(\boldsymbol{L}_i\boldsymbol{x}),$$

a problem that cleanly fits into (4.2) with $f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2$, $g(\boldsymbol{x}) = 0$, $\boldsymbol{L} = \boldsymbol{I}$. The procedure to find a solution is proposed as

$$\begin{aligned}
\bar{\boldsymbol{x}}^{t+1} &= \operatorname{prox}_{\tau g^*}\left(\boldsymbol{x}^t - \tau\nabla f(\boldsymbol{x}^t) - \tau\sum_{i=1}^{m}\boldsymbol{L}_i^*\boldsymbol{u}_i^t\right) \\
\boldsymbol{x}_{n+1} &= \rho\bar{\boldsymbol{x}}^{t+1} + (1-\rho)\boldsymbol{x}^t \\
\bar{\boldsymbol{u}}_i^{t+1} &= \operatorname{prox}_{\sigma h_i^*}(\boldsymbol{u}_i^t + \sigma\boldsymbol{L}_i(2\bar{\boldsymbol{x}}_{t+1} - \boldsymbol{x}_t)) \quad \forall i = 1,\dots,m\,, \\
\boldsymbol{u}_i^{t+1} &= \rho\bar{\boldsymbol{u}}_i^{t+1} + (1-\rho)\boldsymbol{u}_i^t \quad \forall i = 1,\dots,m\,,
\end{aligned}$$

for τ, ρ parameters of the algorithm. When applying this procedure to 2D TV ($m = 2$, $r_1(\boldsymbol{x})$ = proximity over rows, $r_2(\boldsymbol{x})$ = proximity over columns) an algorithm almost equivalent to Chambolle and Pock (2011) is obtained, the only difference being that here the gradient of f is used, instead of the prox_G operation.

Yet another related method is the splitting approach of Kolmogorov et al (2015), which for $m = 2$ performs the following splitting:

$$\begin{aligned}
&\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}), \\
\equiv &\min_{\boldsymbol{x},\boldsymbol{x}'} \quad \|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}') \quad \text{s.t. } \boldsymbol{x} = \boldsymbol{x}', \\
\equiv &\min_{\boldsymbol{x},\boldsymbol{x}'}\max_{\boldsymbol{z}} \quad \|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}') + \boldsymbol{z}^T(\boldsymbol{x}-\boldsymbol{x}'), \\
\equiv &\min_{\boldsymbol{x}}\max_{\boldsymbol{z}} \quad \|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) - r_2^*(\boldsymbol{z}) + \boldsymbol{x}^T\boldsymbol{z}.
\end{aligned}$$

where we have made use of the Fenchel dual $r_2^*(\boldsymbol{z}) = \max_{\boldsymbol{x}'} \boldsymbol{z}^T\boldsymbol{x}' - r_2(\boldsymbol{x}')$. This problem can be solved through a primal-dual minimization:

$$\begin{aligned}
\boldsymbol{z}^{t+1} &= \operatorname{prox}_{\sigma^t r_2^*}\left(\boldsymbol{z}^t + \sigma^t(\boldsymbol{x}^t + \theta^t(\boldsymbol{x}^t - \boldsymbol{x}^{t-1}))\right), \\
\boldsymbol{x}^{t+1} &= \operatorname{prox}_{\tau^t(\|\cdot-\boldsymbol{y}\|_2^2 + r_1)}\left(\boldsymbol{x}^t - \tau^t\boldsymbol{z}^{t+1}\right).
\end{aligned}$$

The primal proximity operator over the squared norm term plus r_1 can be rewritten in terms of $\operatorname{prox}_{r_1}$ as

$$\begin{aligned}
\operatorname{prox}_{\tau(r_1 + \frac{1}{2}\|\cdot-\boldsymbol{y}\|_2^2)}(\boldsymbol{w}) &= \operatorname*{argmin}_{\boldsymbol{x}} r_1(\boldsymbol{x}) + \frac{1+\tau^{-1}}{2}\|\boldsymbol{x} - (1+\tau^{-1})^{-1}(\boldsymbol{y} + \tau^{-1}\boldsymbol{w})\|_2^2, \\
&= \operatorname{prox}_{(1+\tau^{-1})^{-1}r_1}\left((1+\tau^{-1})^{-1}(\boldsymbol{y} + \tau^{-1}\boldsymbol{w})\right).
\end{aligned}$$

Regarding the dual step, in the previously presented methods the decompositions allowed to disentangle the effect of a linear operator L_i from each r_i. The present decomposition, however, does not take into account this possibility, thus increasing the complexity of computing r_2^*. To address this difficulty the Moreau decomposition (A.3) is helpful, as

$$\begin{aligned}
\operatorname{prox}_{\sigma r_2^*}(\boldsymbol{w}) &= \boldsymbol{w} - \sigma\left(\operatorname*{argmin}_{\boldsymbol{x}} r_2(\boldsymbol{x}) + \frac{\sigma}{2}\|\boldsymbol{x} - \sigma^{-1}\boldsymbol{w}\|_2^2\right), \\
&= \boldsymbol{w} - \sigma\operatorname{prox}_{\sigma^{-1}r_2}(\sigma^{-1}\boldsymbol{w}),
\end{aligned}$$

thus solving the dual proximity operator in terms of the primal $\operatorname{prox}_{r_2}$. Regarding the algorithm parameters θ, τ and σ, they can be adjusted at every iteration for greater performance making use of Lipschitz convexity (Chambolle and Pock, 2014).

Lastly, and again for $m = 2$, both r_1 and r_2 can be exploited in their dual forms as shown in Chambolle and Pock (2015) through the splitting

$$\begin{aligned}
&\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}) + r_2(\boldsymbol{x}), \\
\equiv &\min_{\boldsymbol{x},\boldsymbol{x}_1,\boldsymbol{x}_2} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}_1) + r_2(\boldsymbol{x}_2) \quad \text{s.t. } \boldsymbol{x} = \boldsymbol{x}_1, \boldsymbol{x} = \boldsymbol{x}_2 \\
\equiv &\min_{\boldsymbol{x},\boldsymbol{x}_1,\boldsymbol{x}_2}\max_{\boldsymbol{z}_1,\boldsymbol{z}_2} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + r_1(\boldsymbol{x}_1) + \boldsymbol{z}_1^T(\boldsymbol{x}-\boldsymbol{x}_1) + r_2(\boldsymbol{x}_2) + \boldsymbol{z}_2^T(\boldsymbol{x}-\boldsymbol{x}_2).
\end{aligned}$$

Minimizing this Lagrangian over $\boldsymbol{x}, \boldsymbol{x}_1, \boldsymbol{x}_2$ and making use of Fenchel duals we arrive at

$$\max_{\boldsymbol{z}_1,\boldsymbol{z}_2} \quad -\tfrac{1}{2}\|\boldsymbol{z}_1 + \boldsymbol{z}_2\|_2^2 - r_1^*(\boldsymbol{u}_1) - r_2(\boldsymbol{u}_2^*) + (\boldsymbol{u}_1 + \boldsymbol{u}_2)^T\boldsymbol{y},$$

which can be solved through an accelerated alternating minimization as

$$\begin{aligned}
t_{k+1} &= \frac{1 + \sqrt{1 + 4t_k^2}}{2}, \\
\bar{\boldsymbol{x}}^{k+1} &= \boldsymbol{x}_2^k + \frac{t_k - 1}{t_{k+1}}(\boldsymbol{x}_2^k - \boldsymbol{x}_2^{k-1}), \\
\boldsymbol{x}_1^{k+1} &= \operatorname{prox}_{r_1^*}(y - \bar{\boldsymbol{x}}_2^k), \\
\boldsymbol{x}_2^{k+1} &= \operatorname{prox}_{r_2^*}(y - \boldsymbol{x}_1^{k+1}),
\end{aligned}$$

where once again we can resort to the Moreau decomposition to compute the dual proximity operators.

4.2. Two-Dimensional TV

Recall that for a matrix $\boldsymbol{X} \in \mathbb{R}^{n_1 \times n_2}$, the anisotropic 2D-TV regularizer takes the form

$$\mathrm{Tv}^2_{p,q}(\boldsymbol{X}) := \sum_{i=1}^{n_1} \Big(\sum_{j=1}^{n_2-1} |x_{i,j+1} - x_{i,j}|^p \Big)^{1/p} + \sum_{j=1}^{n_2} \Big(\sum_{i=1}^{n_1-1} |x_{i+1,j} - x_{i,j}|^q \Big)^{1/q}. \quad (4.7)$$

This regularizer applies a $\mathrm{Tv}^{1\mathrm{D}}_p$ regularization over each row of $\boldsymbol{X}$, and a $\mathrm{Tv}^{1\mathrm{D}}_q$ regularization over each column. Introducing differencing matrices $\boldsymbol{D}_n$ and $\boldsymbol{D}_m$ for the row and column dimensions, the regularizer (4.7) can be rewritten as

$$\mathrm{Tv}^{2\mathrm{D}}_{p,q}(\boldsymbol{X}) = \sum_{i=1}^{n} \|\boldsymbol{D}_n \boldsymbol{x}_{i,:}\|_p + \sum_{j=1}^{m} \|\boldsymbol{D}_m \boldsymbol{x}_{:,j}\|_q, \quad (4.8)$$

where $\boldsymbol{x}_{i,:}$ denotes the i-th row of $\boldsymbol{X}$, and $\boldsymbol{x}_{:,j}$ its j-th column. The corresponding $\mathrm{Tv}^{2\mathrm{D}}_{p,q}$-proximity problem is

$$\min_{\boldsymbol{X}} \quad \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|^2_{\mathrm{F}} + \lambda\, \mathrm{Tv}^{2\mathrm{D}}_{p,q}(\boldsymbol{X}), \quad (4.9)$$

where we use the Frobenius norm $\|\boldsymbol{X}\|_{\mathrm{F}} = \sqrt{\sum_{ij} x^2_{i,j}} = \|\mathrm{vec}(\boldsymbol{X})\|_2$, where $\mathrm{vec}(\boldsymbol{X})$ is the vectorization of $\boldsymbol{X}$. Using (4.8), problem (4.9) becomes

$$\min_{\boldsymbol{X}} \quad \tfrac{1}{2}\|\boldsymbol{X} - \boldsymbol{Y}\|^2_{\mathrm{F}} + \lambda \Big(\sum\nolimits_i \|\boldsymbol{D}_n x_{i,:}\|_p \Big) + \lambda \Big(\sum\nolimits_j \|\boldsymbol{D}_m x_{:,j}\|_q \Big), \quad (4.10)$$

where the parentheses make explicit that $\mathrm{Tv}^{2\mathrm{D}}_{p,q}$ is a combination of two regularizers: one acting over the rows and the other over the columns. Formulation (4.10) fits the model solvable by the strategies presented above, though with an important difference: each of the two regularizers that make up $\mathrm{Tv}^{2\mathrm{D}}_{p,q}$ is itself composed of a sum of several (n or m) 1D-TV regularizers. Moreover, each of the 1D row (column) regularizers operates on a different row (columns), and can thus be solved independently.

4.3. Higher-Dimensional TV

Going even beyond $\mathrm{Tv}^{2\mathrm{D}}_{p,q}$ is the general multidimensional TV (1.3), which we recall below.

Let X be an order-m tensor in $\mathbb{R}^{\prod_{j=1}^m n_j}$, whose components are indexed as $\mathsf{X}_{i_1,i_2,\ldots,i_m}$ ($1 \le i_j \le n_j$ for $1 \le j \le m$); we define TV for X as

$$\mathrm{Tv}^m_{\boldsymbol{p}}(\mathsf{X}) \stackrel{\text{def}}{=} \sum_{k=1}^{m} \sum_{\{i_1,\ldots,i_m\}\setminus i_k} \Big(\sum_{j=1}^{n_k-1} |\mathsf{X}_{i_1,\ldots,i_{k-1},j+1,i_{k+1},\ldots,i_m} - \mathsf{X}_{i_1,\ldots,i_{k-1},j,i_{k+1},\ldots,i_m}|^{p_k} \Big)^{1/p_k}, \quad (4.11)$$

where $\boldsymbol{p} = [p_1, \ldots, p_m]$ is a vector of scalars $p_k \ge 1$. This corresponds to applying a 1D-TV to each of the 1D fibers of X along each of the dimensions.

Introducing the *multi-index* $\boldsymbol{i}(k) = (i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_m)$, which iterates over every 1-dimensional fiber of X along the k-th dimension, the regularizer (4.11) can be written more compactly as

$$\mathrm{Tv}^m_{\boldsymbol{p}}(\mathsf{X}) = \sum_{k=1}^{m} \sum_{\boldsymbol{i}(k)} \|\boldsymbol{D}_{n_k} \boldsymbol{x}_{\boldsymbol{i}(k)}\|_{p_k}, \quad (4.12)$$

where $\boldsymbol{x}_{\boldsymbol{i}(k)}$ denotes a row of X along the k-th dimension, and $\boldsymbol{D}_{n_k}$ is a differencing matrix of appropriate size for the 1D-fibers along dimension k (of size n_k). The corresponding m-dimensional-TV proximity problem is

$$\min_{\mathsf{X}} \ \tfrac{1}{2}\|\mathsf{X} - \mathsf{Y}\|^2_{\mathrm{F}} + \lambda\, \mathrm{Tv}^m_{\boldsymbol{p}}(\mathsf{X}), \quad (4.13)$$

where $\lambda > 0$ is a penalty parameter, and the Frobenius norm for a tensor just denotes the ordinary sum-of-squares norm over the vectorization of such tensor.

Problem (4.13) looks very challenging, but it enjoys decomposability as suggested by (4.12) and made more explicit by writing it as a sum of $\mathrm{Tv}^{1\mathrm{D}}$ terms

$$\min_{\mathsf{X}} \ \tfrac{1}{2}\|\mathsf{X} - \mathsf{Y}\|^2_{\mathrm{F}} + \sum_{k=1}^{m} \sum_{\boldsymbol{i}(k)} \mathrm{Tv}^{1\mathrm{D}}_{p_k}\big(\boldsymbol{x}_{\boldsymbol{i}(k)}\big). \quad (4.14)$$

The proximity task (4.14) can be regarded as the sum of m proximity terms, each of which further decomposes into a number of inner $\mathrm{Tv}^{1\mathrm{D}}$ terms. These inner terms are trivial to address since, as in the 2D-TV case, each of the $\mathrm{Tv}^{1\mathrm{D}}$ terms operates on different entries of X. Regarding the m major terms, we can handle them by applying any of the combiner strategies presented above for $m > 2$, which ultimately yield the prox operator for $\mathrm{Tv}^m_{\boldsymbol{p}}$ by just repeatedly calling $\mathrm{Tv}^{1\mathrm{D}}$ prox operators. Most importantly, both proximal stacking and the natural decomposition of the problem provide a vast potential for parallel multithreaded computing, which is valuable when dealing with such complex and high-dimensional data.

5. Experiments and Applications

We will now demostrate the effectiveness of the various solvers covered in a wide array of experiments, as well as showing many of their practical applications. We will start by focusing on the $\mathrm{Tv}^{1\mathrm{D}}_1$ methods, moving then to other 1D-TV variants, and then to multidimensional TV.

All the solvers implemented for this paper were coded in C++ for efficiency. Our publicy available library **proxTV** includes all these implementations, plus bindings for easy usage in Matlab or Python: https://github.com/albarji/proxTV. Matrix operations have been implented by exploiting the LAPACK (Fortran) library (Anderson et al., 1999).

5.1. $\mathrm{Tv}^{1\mathrm{D}}_1$ Experiments and Applications

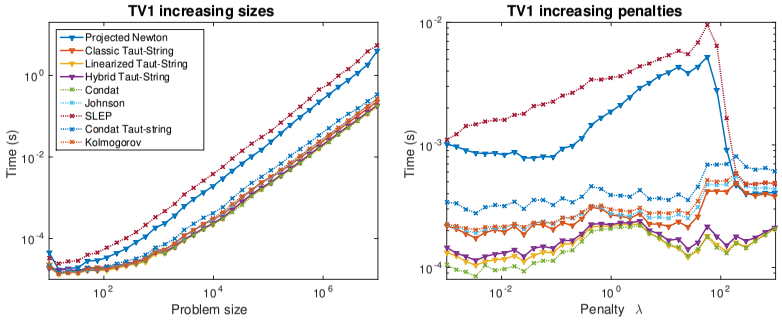
Since the most important components of the presented modular framework are the efficient $\mathrm{Tv}^{1\mathrm{D}}_1$ prox operators, let us begin by highlighting their empirical performance. We will do so both on synthetic and natural images data.

5.1.1. Running Time Results for Synthetic Data

We test the solvers under two scenarios of synthetic signals:

I) Increasing input size ranging from $n = 10^1$ to $n = 10^7$. A penalty $\lambda \in [0, 50]$ is chosen at random for each run, and the data vector $\boldsymbol{y}$ with uniformly random entries $y_i \in [-2\lambda, 2\lambda]$ (proportionally scaled to λ).

II) Varying penalty parameter λ ranging from 10^{-3} (negligible regularization) to 10^3 (the TV term dominates); here n is set to 1000 and y_i is randomly generated in the range $[-2, 2]$ (uniformly).



(a) (b)

Figure 7: Running times (in secs) for proposed and state of the art solvers for $\mathrm{Tv}_1^{1\mathrm{D}}$-proximity with increasing a) input sizes, b) penalties. Both axes are on a log-scale.

We benchmark the performance of the following methods, including both our proposals and state of the art methods found in the literature:

- Our proposed Projected Newton method (Appendix E).
- Our efficient implementation of the classic taut string method.
- Another implementation of the classic taut string method by Condat (2012).
- An implementation of the linearized taut string method.
- Our proposed hybrid taut string approach.
- The **FLSA** function (C implementation) of the SLEP library of Liu et al. (2009) for $\mathrm{Tv}_1^{1\mathrm{D}}$-proximity (Liu et al., 2010).
- The state-of-the-art method of Condat (2012), which we have seen to be equivalent to a linearized taut-string method.
- The dynamic programming method of Johnson (2013), which guarantees linear running time.
- The message passing method of Kolmogorov et al (2015), which allows generalization for computing a Total Variation regularizer on a tree.

Another implementation of the classic taut string method, found in the literature, has been added to the benchmark to test whether the implementation we have proposed is on par with the state of the art. We would like to note the surprising lack of widely available implementations of this method: the only working and efficient code we could find was part of the same paper where Condat's method was proposed.

For Projected Newton and SLEP a duality gap of 10^{-5} is used as the stopping criterion. For the hybrid taut-string method the switch parameter is set as $S = 1.05$. The rest of algorithms do not have parameters.

Timing results are presented in Figure 7 for both experimental scenarios. The following interesting facts are drawn from these results

- Direct methods (Taut string methods, Condat, Johnson, Kolmogorov) prove to be much faster than iterative methods (Projected Newton, SLEP).
- Although Condat's (and hence linearized taut string) method, has a theoretical worst-case performance of $O(n^2)$, the practical performance seems to follow an $O(n)$ behavior, at least for these synthetic signals.
- Even if Johnson and Kolmogorov methods have a guaranteed running time of $O(n)$, they turn out to be slower than the linearized taut string and Condat's methods. This is in line with our previous observations of the cache-friendly properties of in-memory methods; in contrast Johnson's method requires an extra $\sim 8n$ memory storage. Kolmogorov's method has less memory requirements but nevetheless shows similar behavior.
- The same performance observation applies to the classic taut string method. It is also noticeable that our implementation of this method turns out to be faster than previously available implementations (Condat's Taut-string), even becoming slightly faster than the state of the art Johnson and Kolmogorov methods. This result is surprising, and shows that the full potential of the classic taut-string method has been largely unexploited by the research community, or at least that proper efficient implementations of this method have not been made readily available so far.

5.1.2. Worst Case Scenario

The point about comparing $O(n)$ and $O(n^2)$ algorithms deserves more attention. As an illustrative experiment we have generated a signal following the worst case description in Condat (2012), and tested again the methods above on it, for increasing signal lengths. Figure 8 plots the results. Condat's method and consequently the linearized taut string method shows much worse performance than the rest of the direct methods. It is also remarkable how the hybrid method manages to avoid quadratic runtimes in this case.

5.1.3. Running Times on Natural Images

In the light of the previous results the following question arises: in practical settings, are the problems to be solved closer to the worst or the average runtime scenario? This fact will determine whether the guaranteed linear time or the more risky quadratic methods are more apt for practical use. To test this we devise the following experiment: we take a large benchmark of natural images and run each solver over all the rows and columns of all the images in the set, counting total running times, for different regularization values. The benchmark is made from images obtained from the data sets detailed in Table 2. We run this benchmark for the methods showing better performance in the experiments above: our implementation of the classic taut-string method, Condat's method ($\equiv$ linearized taut-string method), our proposed Hybrid taut-string method, Johnson's method and Kolmogorov et al's method.

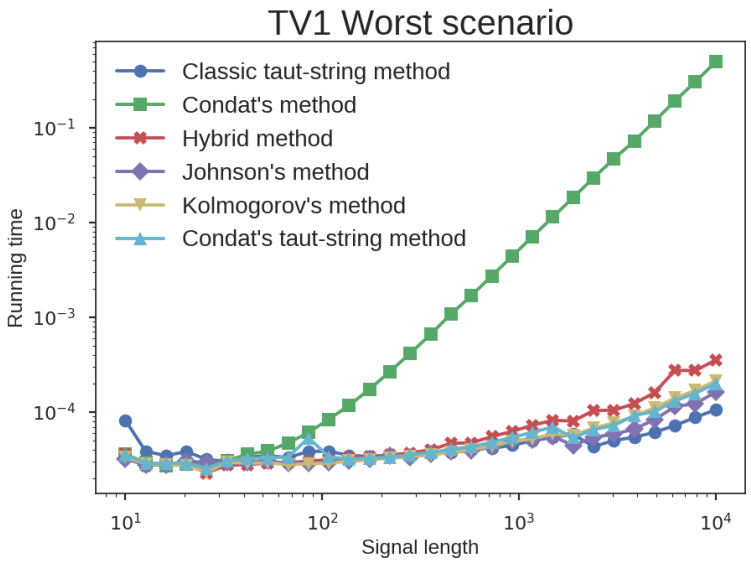


Figure 8: Running times (in secs) for proposed and state of the art solvers for Tv_1^{1D}-proximity in the worst-case scenario for Condat's method, for increasing input sizes. Both axes are on a log-scale.

Data set	Images	Average image size
INRIA holidays (Jegou et al, 2008)	812	1817 × 2233 × 3 px
LSVRC 2010 val set (Russakovsky et al, 2015)	50000	391 × 450 × 3 px

Table 2: Detail of image data sets used for large-scale Tv_1^{1D} experiments.

Figure 9 shows runtime results for different penalty values over the whole INRIA holidays data set (Jegou et al, 2008), while Figure 10 shows similar results for the whole Large Scale Visual Recognition Challenge 2010 validation data set (Russakovsky et al, 2015). The following facts of interest can be observed:

- Condat's method (linearized taut-string) shows top performance for low penalty values, but bad scaling when moving to higher penalties. This can be explained using the geometric intuition developed above: for large penalty values the width of the tube is very large, and thus the taut-string will be composed of very long segments.

This is troublesome for a linearized taut-string method, as each backtrack will require recomputing a large number of steps. On the contrary for smaller penalties the tube will be narrow, and the taut-string composed of many small segments, thus resulting in very cheap backtracking costs.

- The performance of Classic taut-string, Johnson and Kolmogorov becomes slightly worse for large penalties, but suffers significantly less than the linearized taut-string. Surprisingly, the best performing approach tends to be the classic taut-string method.

- The proposed hybrid strategy closely follows the performance of Condat's method for the low penalty regime, while adapting to a behaviour akin to Kolmogorov for large penalties, thus resulting in very good performances over the whole regularization spectrum.

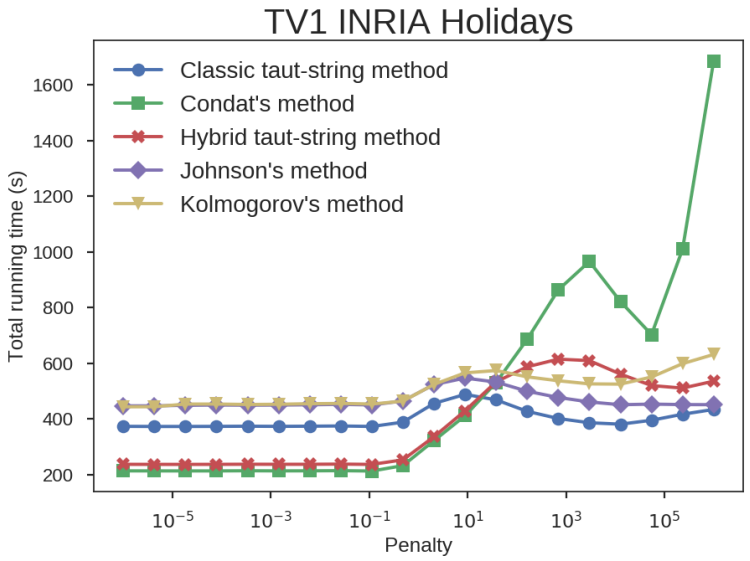


Figure 9: Running times (in secs) for the top performing proposed and state of the art solvers for Tv_1^{1D}-proximity over the whole INRIA Holidays data set, for increasing penalties.

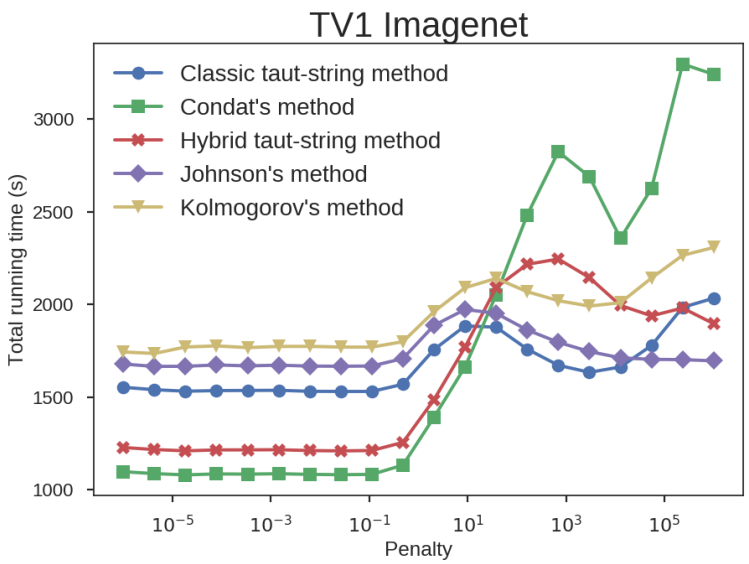


Figure 10: Running times (in secs) for the top performing proposed and state of the art solvers for $\mathrm{Tv}_1^{1\mathrm{D}}$-proximity over the whole Large Scale Visual Recognition Challenge 2010 validation data set, for increasing penalties.

5.1.4. Running Time Results for Weighted TV-L1

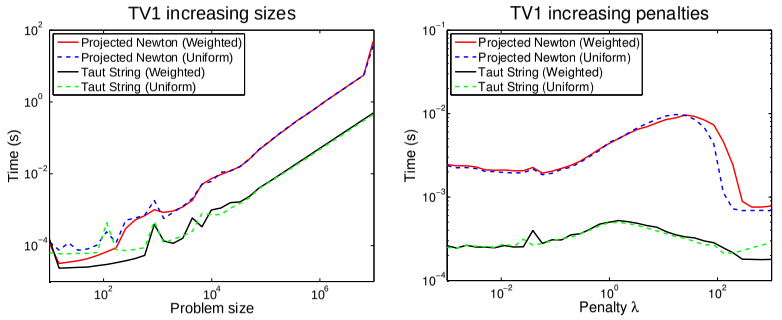
An advantage of the solvers proposed in this paper is their flexibility to easily deal with the more difficult, weighted version of the TV-L1 proximity problem. To illustrate this, Figure 11 shows the running times of the Projected Newton and (linearized) Taut String methods when solving both the standard and weighted TV-L1 prox operators.

Since for this set of experiments a whole vector of weights $\boldsymbol{w}$ is needed, we have adjusted the experimental scenarios as follows:

I) n is generated as in the general setting, penalties $\boldsymbol{w} \in [0, 100]$ are chosen at random for each run, and the data vector $\boldsymbol{y}$ with uniformly random entries $y_i \in [-2\lambda, 2\lambda]$, with λ the mean of $\boldsymbol{w}$, using also this λ choice for the uniform (unweighted) case.

II) λ and n are generated as in the general setting, and the weights vector $\boldsymbol{w}$ is drawn randomly from the uniform distribution $\boldsymbol{w}_i \in [0.5\lambda, 1.5\lambda]$.

As can be readily observed, performance for both versions of the problem is almost identical, even if the weighted problem is conceptually harder. Conversely, adapting the



(a) (b)

Figure 11: Running times (in secs) for Projected Newton and Taut String solvers for weighted and uniform $\mathrm{Tv}_1^{1\mathrm{D}}$-proximity with increasing a) input sizes, b) penalties. Both axes are on a log-scale.

other reviewed algorithms to address this problem while keeping up with performance is not a straightforward task.

We would also like to point out that in the paper Kumar et al (2015) a practical application of this method for energy minimization in computer vision is presented, where exactly the code behind this paper has been put to use.

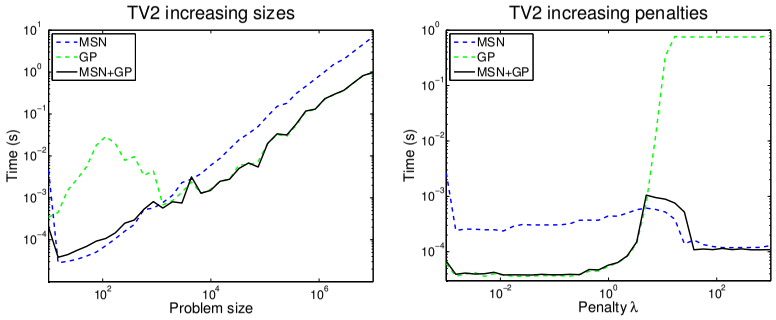
5.2. Experiments for other 1D-TV Variants

In this section we present experiments for other choices of the ℓ_p norm in $\mathrm{Tv}_p^{1\mathrm{D}}$, namely $p = 2$, $p = \infty$ and any general $p \geq 1$.

5.2.1. Running Time Results for TV-L2

Next we show results for $\mathrm{Tv}_2^{1\mathrm{D}}$ proximity. To our knowledge, this version of TV has not been explicitly treated before, so there do not exist highly-tuned solvers for it. Thus, we show running time results only for the MSN and GP methods. We use a duality gap of 10^{-5} as the stopping criterion; we also add an extra boundary check for MSN with tolerance 10^{-6} to avoid early stopping due to potentially infeasible intermediate iterates. Figure 12 shows results for the two experimental scenarios under test.

The results indicate that the performance of MSN and GP differs noticeably in the two experimental scenarios. While the results for the first scenario (Figure 12(a)) might suggest that GP converges faster than MSN for large inputs, it actually does so depending on the size of λ relative to $\|\boldsymbol{y}\|_2$. Indeed, the second scenario (Figure 12(b)) shows that although for small values of λ, GP runs faster than MSN, as λ increases, GP's performance worsens dramatically, so much that for moderately large λ, it is unable to find an acceptable solution even after 10,000 iterations (an upper limit imposed in our implementation). Conversely,



(a) (b)

Figure 12: Running times (in secs) for MSN, GP and a hybrid MSN+GP approach for $\mathrm{Tv}_2^{1\mathrm{D}}$-proximity with increasing a) input sizes, b) penalties. Both axes are on a log-scale.

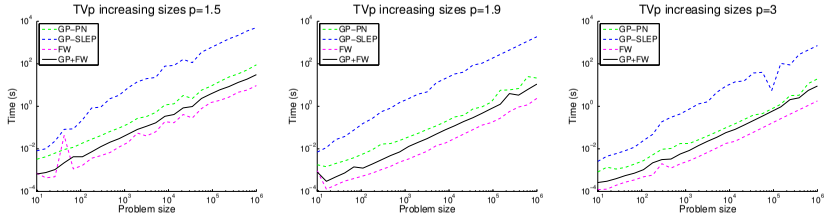
MSN finds a solution satisfying the stopping criterion under every situation, thus showing a more robust behavior.

These results suggest that it is preferable to employ a hybrid approach that combines the strengths of MSN and GP. Such a hybrid approach is guided using the following (empirically determined) rule of thumb: if $\lambda < \|\boldsymbol{y}\|_2$ use GP, otherwise use MSN. Further, as a safeguard, if GP is invoked but fails to find a solution within 50 iterations, the hybrid should switch to MSN. This combination guarantees rapid convergence in practice. Results for this hybrid approach are also included in the plots in Figure 12, and show how it successfully mimics the behavior of the better algorithm amongst MSN and GP.

5.2.2. Running Time Results for TV-Lp

Now we show results for $\mathrm{Tv}_p^{1\mathrm{D}}$ proximity. Again, to our knowledge efficient solvers for this version of TV are not available; still proposals for solving the ℓ_q-ball projection problem do exist. For these experiments we decided to use a method based on a zero finding approach readily available as the *epp* function in SLEP library (Liu et al., 2009). Consequently, we present here a comparison between this reference projection subroutine and our PN–based projection when embedded in our proposed Gradient Projection solver of §3.2. The alternative proposal given by the Frank–Wolfe algorithm of §3.2.2 is also present in the comparison. We use a duality gap of 10^{-5} as stopping criterion both for GP and FW. Figure 13 shows results for the two experimental scenarios under test, for p values of 1.5, 1.9 and 3.

A number of interesting conclusions can be drawn from the results. First, our Projected Newton ℓ_q-ball subroutine is far more efficient than *epp* when in the context of the GP solver. Two factors seem to be the cause of this: in the first place our Projected Newton



(a) (b) (c)

Figure 13: Running times (in secs) for GP with PN projection, GP with SLEP's *epp* projection, FW and a hybrid GP+FW algorithm, for $\mathrm{Tv}_p^{1\mathrm{D}}$-proximity with increasing input sizes and three different choices of p. Both axes are on a log-scale.

approach proves to be faster than the zero finding method used by *epp*. Secondly, in order for the GP solver to find a solution within the desired duality gap, the projection subroutine must provide very accurate results (about 10^{-12} in terms of duality gap). Given its Newton nature, our ℓ_q-ball subroutine scales better in term of running times as a factor of the desired accuracy, which explains he observed differences in performance.

It is also of relevance noting that Frank–Wolfe is significantly faster than Projected Newton. This should discourage the use of Projected Newton, but we find it to be extremely useful in the range of λ penalties where λ is large, but not enough to render the problem trivial ($\boldsymbol{w} = 0$ solution). In this range the two variants of PN and also FW are unable to find a solution within the desired duality gap (10^{-5}), getting stuck at suboptimal solutions. We solve this issue by means of a hybrid GP+FW algorithm, in which updates from both methods are interleaved at a ratio of 10 FW updates per 1 GP update, as FW updates are faster. As both algorithms guarantee improvement in each iteration but follow different procedures for doing so, they complement each other nicely, resulting in a superior method attaining the objective duality gap and performing faster than GP.

5.2.3. Running Time Results for TV-L∞

For completeness we also include results for our $\mathrm{Tv}_\infty^{1\mathrm{D}}$ solver based on GP + a standard ℓ_1-projection subroutine. Figure 15 presents running times for the two experimental scenarios under test. Since ℓ_1-projection is an easier problem than the general ℓ_q-projection the resultant algorithm converges faster to the solution than the general GP $\mathrm{Tv}_p^{1\mathrm{D}}$ prox solver, as expected.

5.2.4. Application: Proximal Optimization for Fused-Lasso

We now present a key application that benefits from our TV prox operators: **Fused-Lasso** (FL) (Tibshirani et al., 2005), a model that takes the form

$$\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2 \mathrm{Tv}_1^{1\mathrm{D}}(\boldsymbol{x}). \tag{5.1}$$

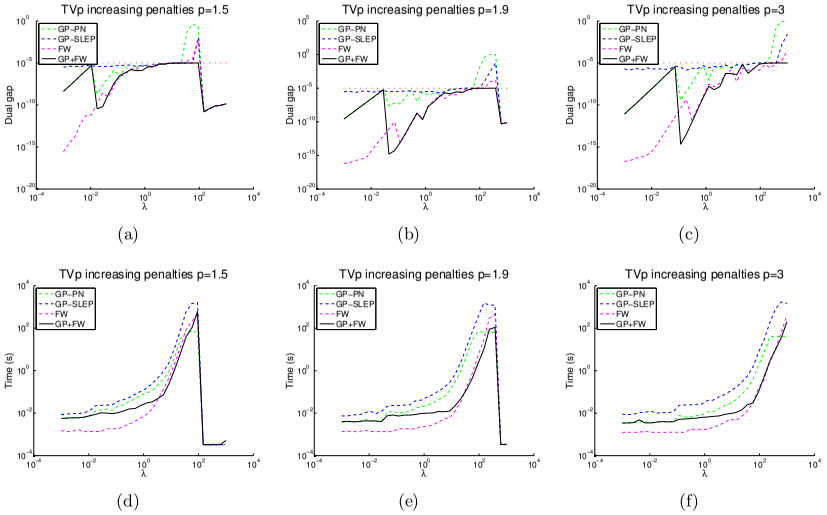


Figure 14: Attained duality gaps (a-c) and running times (d-f, in secs) for GP with PN projection, GP with SLEP's *epp* projection, FW and a hybrid GP+FW algorithm, for $\mathrm{Tv}_p^{1\mathrm{D}}$-proximity with increasing penalties and three different choices of p. Both axes are on a log-scale.

The ℓ_1-norm in (5.1) forces many x_i to be zero, while $\mathrm{Tv}_1^{1\mathrm{D}}$ favors nonzero components to appear in blocks of equal values $x_{i-1} = x_i = x_{i+1} = \ldots$. The FL model has been successfully applied in several bioinformatics applications (Tibshirani and Wang, 2008; Rapaport and Vert, 2008; Friedman et al., 2007), as it encodes prior knowledge about consecutive elements in microarrays becoming active at once.

Following the ideas presented in Sec. 4, since the FL model uses two regularizers, we can use Proximal Dykstra as the combiner to handle the prox operator. To illustrate the benefits of this framework in terms of reusability, we apply it to several variants of FL.

- **Fused-Lasso (FL):** Least-squares loss $+\ell_1 + \mathrm{Tv}_1^{1\mathrm{D}}$ as in (5.1)
- **ℓ_p-Variable Fusion (VF):** Least-squares loss $+\ell_1 + \mathrm{Tv}_p^{1\mathrm{D}}$. Though Variable Fusion was already studied by Land and Friedman (1997), their approach proposed an ℓ_p^p-like regularizer in the sense that $r(\boldsymbol{x}) = \sum_{i=1}^{n-1} |x_{i+1} - x_i|^p$ is used instead of the TV regularizer $\mathrm{Tv}_p^{1\mathrm{D}}(x) = \left(\sum_{i=1}^{n-1} |x_{i+1} - x_i|^p\right)^{1/p}$. Using Tv_p leads to a more conservative penalty that does not oversmooth the estimates. This FL variant seems to be new.

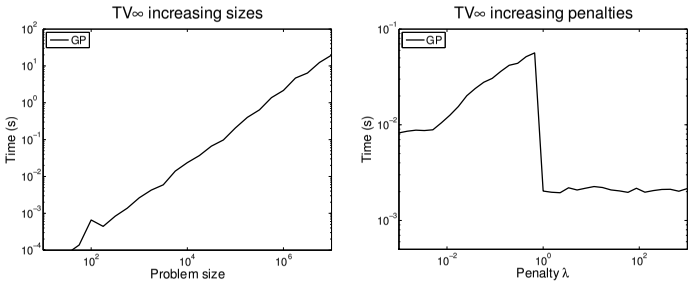


Figure 15: Running times (in secs) for GP for $\mathrm{Tv}_\infty^{1\mathrm{D}}$-proximity with increasing a) input sizes, b) penalties. Both axes are on a log-scale.

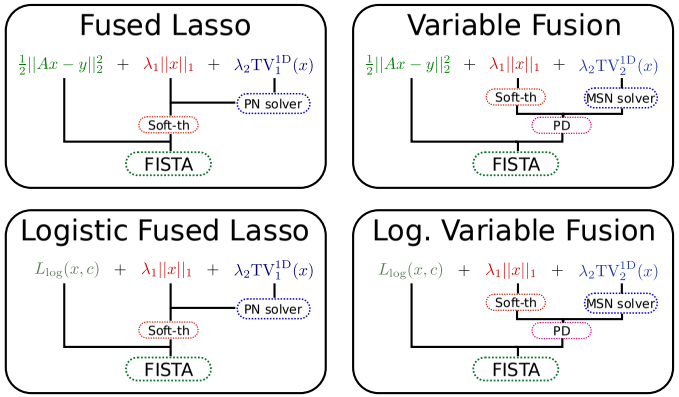


Figure 16: Fused-Lasso models addressed by proximal splitting.

- **Logistic-fused lasso (LFL):** Logistic-loss $+\ell_1 + \mathrm{Tv}_1^{1\mathrm{D}}$, where the loss takes the form $\ell(\boldsymbol{x}, c) = \sum_i \log\left(1 + e^{-y_i(\boldsymbol{a}_i^T \boldsymbol{x} + c)}\right)$, and can be used in a FL formulation to obtain models more appropriate for classification on a data set $\{(\boldsymbol{a}_i, y_i)\}$ (Kolar et al., 2010).
- **Logistic + ℓ_p-fusion (LVF):** Logistic loss $+\ell_1 + \mathrm{Tv}_p^{1\mathrm{D}}$.

To solve these variants of FL, all that remains is to compute the gradients of the loss functions, but this task is trivial. Each of these four models can be then solved easily by invoking any proximal splitting method by appropriately plugging in gradient and prox operators. Incidentally, the **SLEP** library (Liu et al., 2010) includes an implementation

of FISTA (Beck and Teboulle, 2009) carefully tuned for Fused Lasso, which we base our experiments on. Figure 16 shows a schematic of the algorithmic modules for solving each FL model.

Remark: A further algorithmic improvement can be obtained by realizing that for $r(\boldsymbol{x}) = \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2 \mathrm{Tv}_1^{1\mathrm{D}}(\boldsymbol{x})$ the prox operator $\mathrm{prox}_r \equiv \mathrm{prox}_{\lambda_1\|\cdot\|_1} \circ \mathrm{prox}_{\lambda_2 \mathrm{Tv}_1^{1\mathrm{D}}(\cdot)}$. Such a decomposition does not usually hold, but it can be shown to hold for this particular case (Yu, 2013; Rinaldo, 2009; Tibshirani et al., 2005). Therefore, for FL and LFL we can compute the proximal operator for the combined regularizer r directly, thus removing the need for a combiner algorithm. This is also shown in Figure 16.

5.2.5. Fused-Lasso Experiments: Simulation

The standard FL model has been well-studied in the literature, so a number of practical algorithms addressing it have already been proposed. The aforementioned Fused-Lasso algorithm in the **SLEP** library can be regarded as the state of the art, making extensive use of an efficient proximity subroutine (FLSA). Our experiments on $\mathrm{Tv}_1^{1\mathrm{D}}$-proximity (§5.1) have already shown superiority of our prox solvers over FLSA; what remains to be checked is whether this benefit has a significant impact on the overall FL solver. To do so, we compare running times with synthetic data.

We generate random matrices $\boldsymbol{A} \in \mathbb{R}^{n\times m}$ with i.i.d. entries drawn from a zero mean, unit variance gaussian. We set the penalties to $\lambda_1 = \lambda_2 = 10$. We select the vector of responses $\boldsymbol{y}$ using the formula $\boldsymbol{y} = \mathrm{sgn}(\boldsymbol{A}\boldsymbol{x}_t + \boldsymbol{v})$, where $\boldsymbol{x}_t$, and $\boldsymbol{v}$ are random vectors whose entries have variances 1 and 0.01, respectively. The numerical results are summarized in Figure 17, which compares out of the box SLEP (version 4.0) (Liu et al., 2009) against the very same algorithm employing our fast taut–string $\mathrm{Tv}_1^{1\mathrm{D}}$ solver instead of the default FLSA subroutine of SLEP. Comparison is done by showing the relative distance to the problem's optimum versus time. The optimal values in each setting were estimated by running both algorithms for a very large number of iterations.

The plots show a clear trend: when the input matrices feature a very large column dimension the use of our taut-string $\mathrm{Tv}_1^{1\mathrm{D}}$ solver turns into speedups in optimization times, which however become negligible for matrices with a more balanced rows/columns ratio. This result is reasonable, as the vector x under optimization has size equal to the number of columns of the data matrix A. If A has a large number of columns the cost of solving $\mathrm{Tv}_1^{1\mathrm{D}}$ is significant, and thus any improvement in this step has a noticeable impact on the overall algorithm. Conversely, when the number of rows in A is large the cost of computing the gradient of the loss function ($\nabla \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 = \boldsymbol{A}^T(\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y})$) dominates, getting limited benefits from such improvements in prox computations. Therefore, it is for data with a very large number of features where our proposed method can provide a useful speedup.

5.2.6. Fused-Lasso Experiments: Microarray Classification

Now we report results of applying the four FL models on a series of problems from bioinformatics. We test the FL models on binary classification tasks for the following real microarray data sets: ArrayCGH (Stransky et al., 2006), Leukemias (Golub et al., 1999), Colon (U. Alon et al., 1999), Ovarian (Rogers et al., 2005) and Rat (Hua et al., 2009). Each data set was split into three equal parts (ensuring similar proportion of classes in every split) for

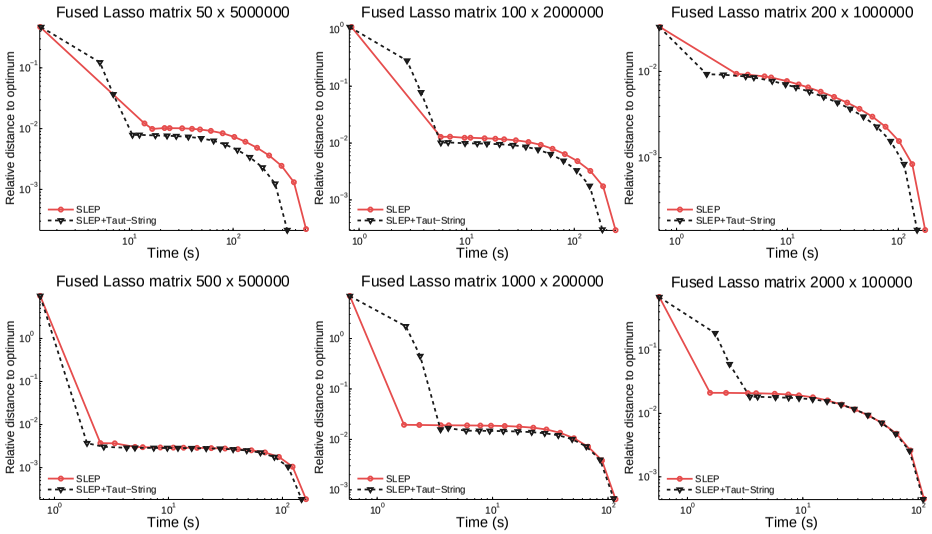


Figure 17: Relative distance to optimum vs time of the Fused Lasso optimizers under comparison, for the different layouts of synthetic matrices.

Data set	FL	VF-ℓ_2	LFL	LVF-ℓ_2
ArrayCGH	73.6%	73.6%	**84.2%**	73.6%
Leukemias	**92.0%**	88.0%	**92.0%**	88.0%
Colon	**77.2%**	**77.2%**	**77.2%**	**77.2%**
Ovarian	**88.8%**	83.3%	83.3%	83.3%
Rat	68.8%	65.5%	**72.1%**	**72.1%**

Table 3: Classification accuracies for the presented Fused–Lasso models on microarray data. For the Variable Fusion models an ℓ_2 version of TV was employed.

training, validation and test. The penalty parameters were found by exhaustive grid search in the range $\lambda_1, \lambda_2 \in [10^{-4}, 10^2]$ to maximize classification accuracy on the validation splits.

Table 3 shows test accuracies. In general, as expected the logistic-loss based FL models yield better classification accuracies than those based on least-squares, as such loss function tends to be more appropriate for classification problems. However the Ovarian data set proves to be an exception, showing better performance under a squared loss. Regarding

the TV-regularizer, the classic $\text{Tv}_1^{1\text{D}}$-penalty seems to perform better in general, with the $\text{Tv}_2^{1\text{D}}$-penalty showing competitive results in some settings.

5.3. 2D-TV: Experiments and Applications

We address now several practical applications that benefit from two-dimensional TV regularization; our results show again how the presented $\text{Tv}_{p,q}^{2\text{D}}$ prox operators fits in seamlessly into our modular framework to produce efficient proximal splitting solvers.

5.3.1. Image Denoising through Anisotropic Filtering

Our first example is related to the classic problem of image denoising, but with the twist that we deal with noise of an anisotropic character. More specifically, suppose that the true image $\mu \in \mathbb{R}^{n\times m}$ is contaminated by additive noise $\boldsymbol{N}$, so that only $\mu_0 = \mu + \boldsymbol{N}$ is observed. The denoising problem estimates μ given just the noisy version μ_0. This problem is highly ill-posed and as such not approachable unless additional assumptions on the noise (or on the underlying image) are made.

Isotropic and anisotropic models: an extremely common choice is to simply assume the noise to be gaussian, or some other zero-mean distribution. Under these conditions, a classic method to perform such denoising task is the **Rudin-Osher-Fatemi** (ROF) model (Rudin et al., 1992), which finds an approximation $\boldsymbol{X}$ to the original image by solving

$$\min_{\boldsymbol{X}} \quad \|\boldsymbol{X} - \mu_0\|_{\text{F}}^2 + \lambda \sum_{i=2}^{n}\sum_{j=2}^{m} \|\partial x_{i,j}\|_2, \tag{5.2}$$

where $\partial x_{i,j}$ is the *discrete gradient*

$$\partial x_{i,j} = \begin{bmatrix} x_{i,j} - x_{i-1,j} \\ x_{i,j} - x_{i,j-1} \end{bmatrix}.$$

That is, it is the vector of differences of $\boldsymbol{X}_{i,j}$ and its neighbors along both axes.

The objective of the first term in the ROF model is to penalize any deviation of $\boldsymbol{X}$ from the observed image μ_0, while the second term can be readily recognized as a mixed $(2,1)$-norm over the discrete gradient of $\boldsymbol{X}$. This regularizer models caters to some prior knowledge: in natural images sharp discontinuities in intensity between neighboring points only appear in borders of objects, while the rest of the pixels usually show smooth variations in intensity. It makes sense, therefore, to penalize large values of the gradient, as sharp changes have a higher probability of having being produced by noise. Conversely, as the mean of the noise is zero, it is also sensible to maintain the denoised image $\boldsymbol{X}$ close to the observed μ_0. Merging these two goals produces the ROF model (5.2).

A closer look at the ROF regularizer reveals that it follows the spirit of the reviewed 2D-TV regularizer which also penalizes sharp variations between neighboring pixels. Indeed, all such regularizers are broadly categorized as TV regularizers within the image processing community. It is clear, though, that the ROF regularizer (5.2) does not coincide with the $\text{Tv}_{p,q}^{2\text{D}}$ regularizer used in this paper. Some authors (Bioucas-Dias and Figueiredo, 2007) differentiate between these regularizers by naming the ROF approach as **isotropic TV** and the $\text{Tv}_{p,q}^{2\text{D}}$-style approach as **anisotropic TV**. This naming comes from the fact that

isotropic TV penalizes each component of the discrete gradient $\partial x_{i,j}$ following an ℓ_2 norm, whereas the anisotropic $\text{Tv}_{p,q}^{2\text{D}}$-norm and in particular $\text{Tv}_{1,1}^{2\text{D}}$-norm, penalize rows and columns independently.

While image filtering using isotropic TV is generally preferred for natural images denoising (Bioucas-Dias et al., 2006), in some settings anisotropic filtering can produce better results, and in fact has been favored by some authors in the past (Choksi et al., 2010; Li and Santosa, 1996). This is specially true on those images that present a "blocky" structure, and thus are better suited to the structure modeled by the $\text{Tv}_{p,q}^{2\text{D}}$-norm. Therefore, efficient methods to perform anisotropic filtering are also important.

Anisotropic denoising experiments: denoising using the anisotropic $\text{Tv}_{p,q}^{2\text{D}}$-norm reduces to solving

$$\min_{\boldsymbol{X}} \quad \|\boldsymbol{X} - \mu_0\|_{\text{F}}^2 + \lambda \text{Tv}_{p,q}^{2\text{D}}(\boldsymbol{X}). \tag{5.3}$$

But (5.3) is nothing but the $\text{Tv}_{p,q}^{2\text{D}}$-proximity problem, and hence can be directly solved by applying the 2D-TV prox operators described above. We solve (5.3) below for the choice $p = q = 1$ (which is common in practice), for the following selection of algorithms:

- Proximal Dykstra (§ 4.1.1)
- The Douglas-Rachford variant based on alternating projections (§ 4.1.2)
- The Split Bregman method of Goldstein T. (2009), which follows an ADMM–like approach to split the ℓ_1 norm apart from the discrete gradient operator, thus not requiring the use of a 1D-TV prox operator.
- Chambolle-Pock's method applied to 2D TV (§ 4.1.4).
- Condat's general splitting method (§ 4.1.4).
- Kolmogorov et al primal-dual method (§ 4.1.4).
- Yang's method (ADMM) (§ 4.1.3)
- The maximum flow approach by Goldfarb and Yin (2009), which shows the relationship between the 2D-TV proximity minimization and the maximum flow problem over a grid, and thus applies an efficient maximum flow method to solve a discrete-valued version of 2D-TV.

In Proximal Dykstra, Douglas-Rachford and ADMM we use the linearized taut–string strategy presented before as solver for the base proximity operators. All algorithm parameters were set as recommended in their corresponding papers or public implementations, except for Proximal Dykstra and Douglas-Rachford, which are parameter free. For Chambolle-Pock we tried both the scheme with fixed algorithm parameters (Chambolle and Pock, 2011, Algorithm 1) and the scheme with acceleration (Chambolle and Pock, 2011, Algorithm 2); however the accelerated version did not converge to the desired solution within enough accuracy (relative difference of 10^{-5}), therefore only the results for the fixed version are reported. For Kolmogorov we follow the recommendations in Chambolle and Pock (2014), taking into account the Lipschitz constants of the optimized functions and selecting the parameter updating strategy that produced faster performance in the experiments: $\theta^{t+1} = \frac{1}{\sqrt{1+\tau^t}}, \tau^{t+1} = \theta^{t+1}\tau^t, \sigma^{t+1} = \frac{\sigma^t}{\theta^{t+1}}, \theta^0 = 1, \tau^0 = \frac{1}{2}, \sigma^0 = 1$.

Image	Gaussian	Speckle	Poisson	Salt & Pepper
randomQR	0.2	0.3	∅	∅
shape	0.05	∅	∅	∅
trollface	∅	1	∅	∅
diagram	∅	∅	✓	∅
text	∅	∅	∅	0.1
comic	0.05	∅	✓	∅
contour	∅	∅	✓	0.4
phantom	∅	2	✓	∅

Table 4: Types of noise and parameters for each test image. A ∅ indicates that such noise was not applied for the image. *Gaussian* and *Speckle* correspond to gaussian additive and multiplicative (respectively) noises with zero mean and the indicated variance. *Salt & Pepper* noise turns into black or white the indicated fraction of image pixels. *Poisson* regenerates each pixel by drawing a random value from a Poisson distribution with mean equal to the original pixel value, thus producing a more realistic noise.

The images used in the experiments are displayed in Appendix F as Figure 25. To test the filters under a variety of scenarios, different kinds of noise were introduced for each image. Table 4 gives details on this, while the noisy images are shown in Figure 26. All QR barcode images used the same kind and parameters of noise. Noise was introduced using Matlab's *imnoise* function.

Values for the regularization parameter λ were found by maximizing the quality of the reconstruction, measured using **Improved Signal-to-Noise Ratio** (ISNR) (Afonso et al., 2010). ISNR is defined as

$$\mathrm{ISNR}(\boldsymbol{X}, \mu, \mu_0) = 10 \log_{10} \frac{\|\mu_0 - \boldsymbol{X}\|_{\mathrm{F}}^2}{\|\boldsymbol{X} - \mu\|_{\mathrm{F}}^2},$$

where μ is the original image, μ_0 its noisy variant, and $\boldsymbol{X}$ the reconstruction.

To compare the algorithms we run all of them for each image and measured its ISNR and relative distance to the optimal objective value of the current solution at each iteration through their execution. The only exception to this procedure is the method of Goldfarb and Yin, which is non–iterative and thus always returns an exact solution, and so we just measure the time required to finish. The optimal objective value was estimated by running all methods for a very large number of iterations and taking the minimum value of them all. This produced the plots shown in Figures 18–19. From them the following observations are of relevance:

- Condat's method and Chambolle-Pock's method are reduced to essentially the same algorithm when applied to the particular case of anisotropic 2D TV denoising. Furthermore, they seem to perform slowly when compared to other methods.
- ADMM (Yang's method) exhibits slow performance at the beginning, but when run for sufficient time is able to achieve a good approximation to the optimum.

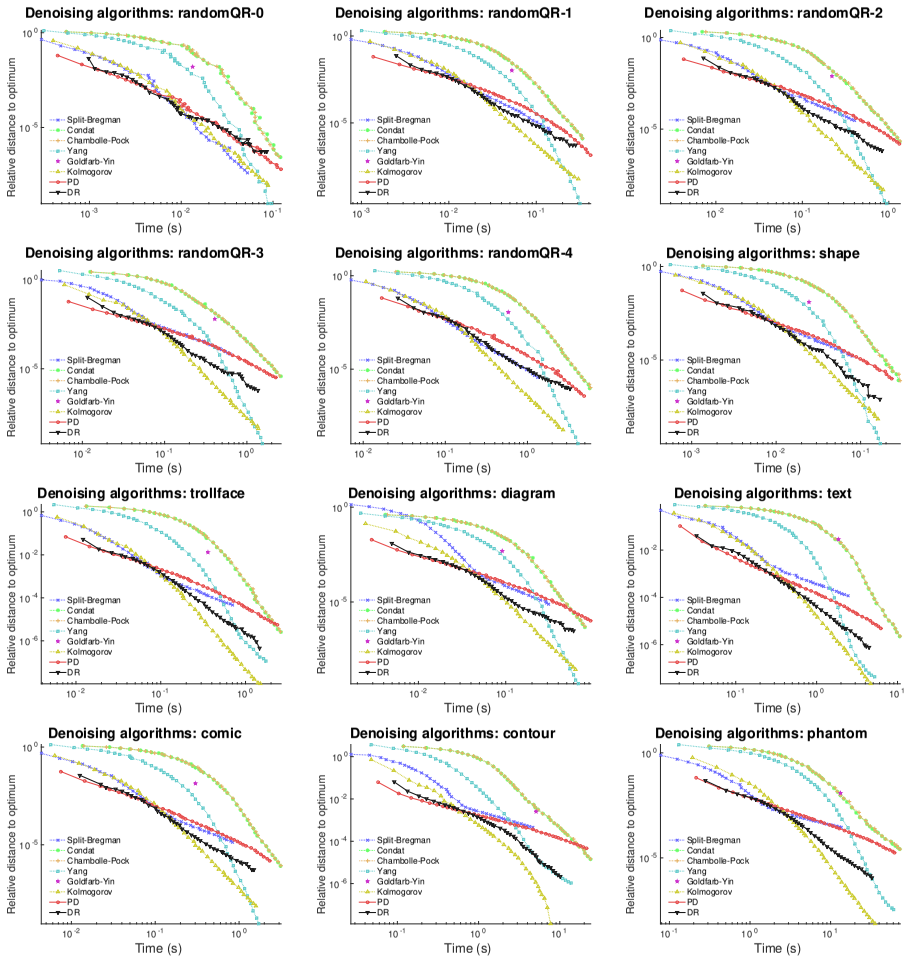


Figure 18: Relative distance to optimum vs time of the denoising 2D-TV algorithms under comparison, for the different images considered in the experiments.

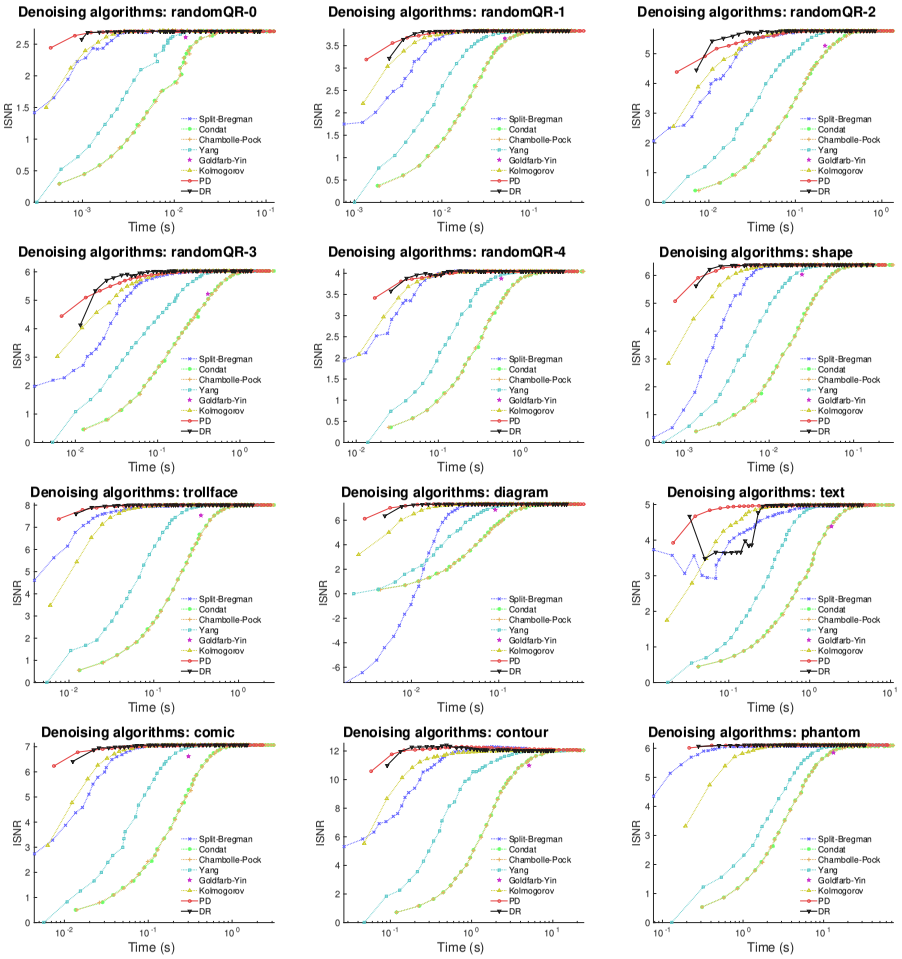


Figure 19: Increased Signal to Noise Ratio (ISNR) vs time of the denoising 2D-TV algorithms under comparison, for the different images considered in the experiments.

- The Split Bregman method, in spite of being an ADMM–like method much like Condat's or Chambolle-Pock, performs significantly better than those. We attribute this to the very efficient implementation provided by its authors, and to the fact that a fast approximate method is employed to compute the required matrix inversions throughout the method.

- The method by Goldfarb and Yin is slower than other approaches and seems to provide suboptimal solutions. We attribute this to the fact that this method solves a discrete (integer–rounded) approximation to the problem. We acknowledge that other methods exploiting the Total Variation - Minimum-cut relationship have been proposed with varying speed results, e.g. (Duan and Tai, 2012), however the suboptimality issues still apply.

- The method by Kolmogorov et al, when properly accelerated by a suitable choice of adaptive stepsizes, seems to be the best choice for finding very accurate solutions, though it is very closely followed by ADMM.

- The parameter free methods PD and DR are the fastest to achieve a mid-quality solution, with Douglas-Rachford performing better than Proximal Dykstra.

Considering these facts, the method of choice among the ones considered depends on the desired accuracy. We argue, however, that for the purpose of image processing a mid-quality solution is sufficient. The ISNR plots of Figure 19 certainly seem to support this, as the perceived quality of the reconstruction, roughly approximated by the ISNR, saturates rapidly and no significant improvements are obtained through further optimization. Given this, the proposed methods seem to be the best suited for the considered task.

For quick reference, Table 5 presents a summary of key points of the compared methods, along with some recommendations about when to put them to use.

5.3.2. Parallelization Experiments

In addition to the previous experiments and to illustrate the parallelization potential of the presented anisotropic filtering method, Figure 20 plots running times for the PD algorithm as the number of processor core ranges from 1 through 16. We see that for the smaller images, the gains due to more processors essentially flatten out by 8 cores, where synchronization and memory contention offsets potential computational gains (first row). For the larger images, there is steadier speedup as the number of cores increase (in each plot there seems to be a "bump" at 14 processors; we attribute this to a quirk of the multicore machine that we used). From all the plots, however, the message is clear: our TV prox operators exploit parallelization well, and show substantial speedups as more processor cores become available.

We should also note in passing that the Split Bregman method, which in the previous experiments showed a reasonable performance, turns out to be much harder to parallelize. This fact was already observed by Jie Wang et al. (2014) in the context of isotropic TV. Therefore when several processor cores are available the proposed modular strategy seems to be even more suitable to the task.

Method		Key points
Douglas Rachford	+	Fast convergence to medium-quality
	+	Embarrassingly parallel
	−	Slow for higher accuracies
	⇒	Ideal for standard denoising tasks
Proximal Dykstra	+	Attainable accuracies similar to DR
	−	But slower than DR
	⇒	Use DR instead
Split Bregman	+	Eventually performs similarly to DR
	−	Slow convergence at first iterations
	⇒	Use DR instead
Chambolle–Pock	−	Slow
	⇒	Use other method instead
Condat	+	Solves objectives involving a sum of smooth/non–smooth functions with linear operators
	−	Reduces to Chambolle–Pock when solving basic image denoising
	⇒	Use only when dealing with more complex functionals
ADMM (Yang)	+	More accurate
	−	Slightly slower than Kolmogorov
	−	Bad behavior for mid-quality solutions
	⇒	Use Kolmogorov instead
Kolmogorov	+	More accurate
	−	Slower than DR for low accuracies
	⇒	Useful when extremely accurate solutions are required
Goldfarb-Yin	+	Solves the discrete version of the problem
	−	Slow
	−	Poor accuracy for the continuous version
	⇒	Apply only when solving the discrete problem

Table 5: Summary of key points of the compared $\mathrm{Tv}^{\mathrm{2D}}_{1,1}$ proximity (denoising) methods.

5.3.3. Anisotropic Image Deconvolution

Taking a step forward we now confront the problem of **image deconvolution** (or image deblurring). This setting is more complex since the task of image recovery is made harder by the presence of a **convolution kernel** K that distorts the image as

$$\mu_0 = \boldsymbol{K} * \mu + \boldsymbol{N},$$

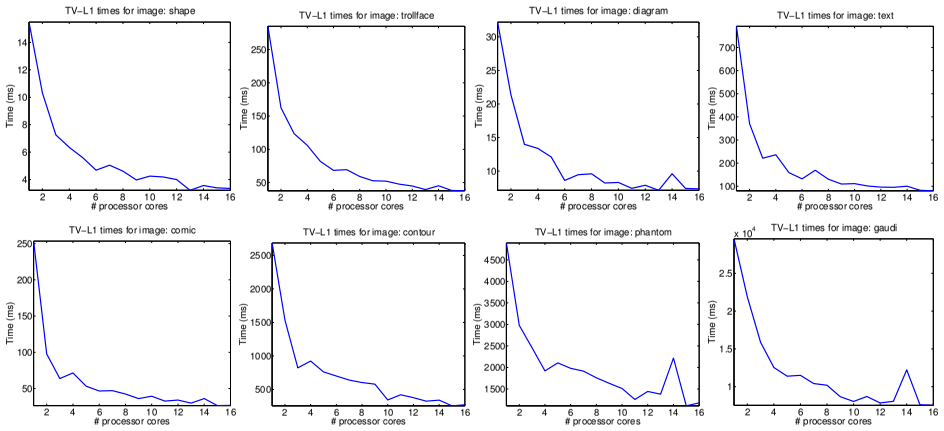


Figure 20: Multicore speedups on different images

where $\boldsymbol{N}$ is noise as before and $*$ denotes convolution. To recover the original image μ from the observed μ_0, it is common to solve the following deconvolution problem

$$\min_{\boldsymbol{X}} \quad \tfrac{1}{2}\|\boldsymbol{K} * \boldsymbol{X} - \mu\|_{\mathrm{F}}^2 + \lambda r(\boldsymbol{X}). \tag{5.4}$$

As before, the regularizer $r(\boldsymbol{X})$ can be isotropic or anisotropic TV, among others. Here we focus again on the anisotropic TV case to show how the presented solvers can also be used for this image task.

Problem (5.4) also fits the proximal splitting framework, and so we employ the popular FISTA (Beck and Teboulle, 2009) method for image processing. The gradient of the loss can be dealt efficiently by exploiting $\boldsymbol{K}$ being a convolution operator, which through the well–known convolution theorem is equivalent to a dot product in the frequencies space, and so the computation is done by means of fast Fourier transforms and products. Several other solvers that explicitly deal with convolution operators are also available (Afonso et al., 2010; Bioucas-Dias and Figueiredo, 2007). A notable solver specific for the isotropic case is given by the work of Krishnan and Fergus (2009), that handles even nonconvex isotropic TV-norms ($0 < p < 1$). But this approach does not extend to the anisotropic case, so we focus on general proximal splitting.

We use the same test images as for our denoising experiments (Figure 25), with identical noise patterns (Table 4) for the QR images, and gaussian noise with variance 0.05 for the rest. In addition, we convolve each image with a different type of kernel to assess the behavior for a variety of convolutions; Table 6 shows the kernels applied. We constructed these kernels using Matlab's *fspecial* function; the convolved images are shown in Figure 28.

The values for the regularizer λ were determined by maximizing the reconstruction quality measured in ISNR. Since deconvolution is much more expensive than denoising,

Image	Convolution	Parameters
randomQR	Motion	Length 5, Angle 35^o
shape	Average	Size 3×3
trollface	Disk	Radius 5
diagram	Motion	Length 5, Angle 0^o
text	Average	Size 1×10
comic	Gaussian	Size 15, Deviation 2
contour	Disk	Radius 5
phantom	Motion	Length 100, Angle 240^o

Table 6: Convolution kernels used for each test image. *Average* substitutes each pixel with the average of its surrounding $n \times m$ neighbors. *Disk* performs the same operation within a disk-shaped neighborhood of the shown radius. *Gaussian* uses a $n \times n$ neighborhood and assigns different weights to each neighbor following the value of a gaussian distribution of the indicated deviation centered at the current pixel. *Motion* emulates the distortions produced when taking a picture in motion, defining a neighborhood following a vector of the indicated length and angle.

instead of performing an exhaustive search for the best λ, we used a Focused Grid Search strategy (Barbero et al., 2008, 2009) to find the best performing values.

Any denoising subroutine can be plugged into the aforementioned deconvolution methods, however for comparison purposes we run our experiments with the best proposed method, Douglas Rachford (Alternating Reflections), and the best competing method among those reviewed from the literature, Kolmogorov et al. A key parameter in deconvolution performance is for how long should these methods be run at each FISTA iteration. To select this, we first run FISTA with 100 iterations of Douglas Rachford per step, for a large number of FISTA steps, and take the final objective value as an estimate of the optimum. Then we find the minimum number of Douglas Rachford and Kolmogorov iterations for which FISTA can achieve a relative distance to such optimum below 10^{-3}. The reason for doing this is that for larger distances the attained ISNR values are still far from convergence. This turned to be 5 iterations for Douglas Rachford an 10 for Kolmogorov. We then run FISTA for such configurations of the inner solvers, and others with a larger number of inner iterations, for comparison purposes.

Figures 21-22 show the evolution of objective values and ISNR for all the tested configurations. In general, Douglas Rachford seems to be slightly better at finding more accurate solutions, and also faster at converging to the final ISNR value. We explain this by the fact that the major advantage of Douglas Rachford is its aforementioned ability to find medium–quality solutions in a very small number of iterations: this is why with a small number of inner DR iterates we can converge to good ISRN levels.

For reference we also provide the resultant deconvoluted images as Figure 29.

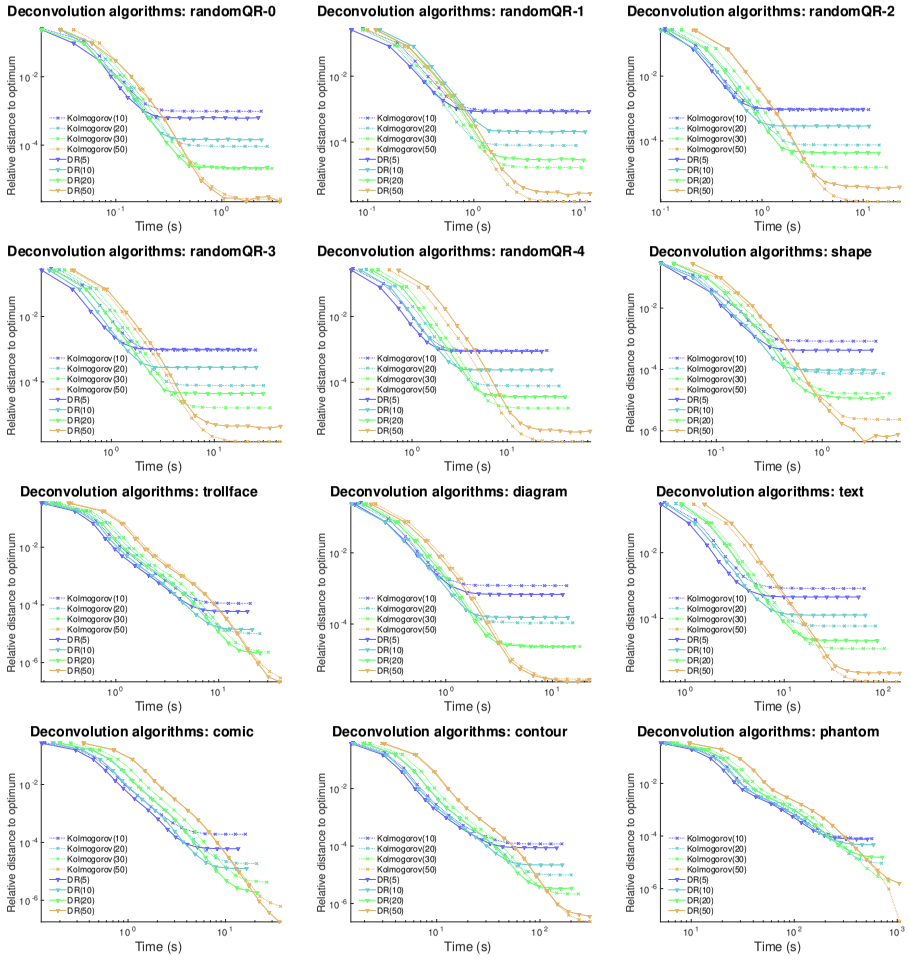


Figure 21: Relative distance to optimum vs time of the deconvolution 2D-TV algorithms under comparison, for the different images considered in the experiments.

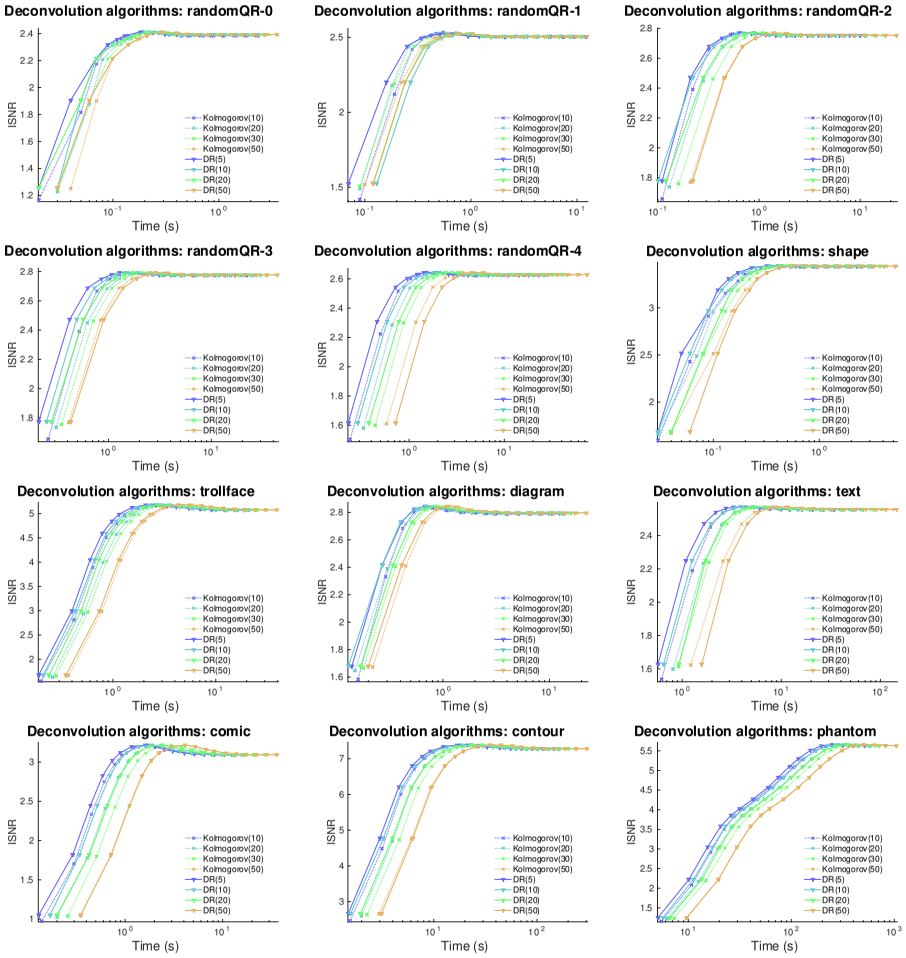


Figure 22: Increased Signal to Noise Ratio (ISNR) vs time of the deconvolution 2D-TV algorithms under comparison, for the different images considered in the experiments.

5.3.4. 2D Fused-Lasso Signal Approximator

The **Fused–Lasso Signal Approximator** (FLSA) (Friedman et al., 2007) can be regarded as a particular case of Fused-Lasso where the input matrix $\boldsymbol{A}$ is the identity matrix $\boldsymbol{I}$, i.e.,

$$\min_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda_1\|\boldsymbol{x}\|_1 + \lambda_2 \mathrm{Tv}_1^{1\mathrm{D}}(\boldsymbol{x}).$$

This problem can be solved immediately using the methods presented in §5.2.4. A slightly less trivial problem is the one posed by the 2D variant of FLSA:

$$\min_{\boldsymbol{X}} \quad \tfrac{1}{2}\|\boldsymbol{X}-\boldsymbol{Y}\|_{\mathrm{F}}^2 + \lambda_1\|\mathrm{vec}(\boldsymbol{X})\|_1 + \lambda_2 \mathrm{Tv}_{1,1}^{2\mathrm{D}}(\boldsymbol{X}). \tag{5.5}$$

Friedman et al. (2007) used this model for denoising images where a large number of pixels are known to be completely black (intensity 0), which aligns well with the structure imposed by the ℓ_1 regularizer.

Akin to the 1D-case, 2D-FLSA (5.5) can also be solved by decomposing its computation into two prox operators (Friedman et al., 2007); formally,

$$\mathrm{prox}_{\lambda_1\|\cdot\|_1+\lambda_2\mathrm{Tv}_{1,1}^{2\mathrm{D}}(\cdot)}(\boldsymbol{Y}) = \mathrm{prox}_{\lambda_1\|\cdot\|_1}\big(\mathrm{prox}_{\lambda_2\mathrm{Tv}_{1,1}^{2\mathrm{D}}(\cdot)}(\boldsymbol{Y})\big).$$

Thus, to solve (5.5) we merely invoke one of the presented $\mathrm{Tv}_{1,1}^{2\mathrm{D}}$ prox operators and then apply soft-thresholding to the results. Since soft-thresholding is done in closed form, the performance of a 2D-FLSA solver depends only on its ability to compute $\mathrm{Tv}_{1,1}^{2\mathrm{D}}$-proximity efficiently. We can then safely claim that the results summarized in table 5 apply equivalently to 2D-FLSA, and so the proposed Douglas Rachford method performs best when reconstruction ISNR is the primary concern.

5.4. Application of Higher-Dimensional TV

We now apply the presented multidimensional TV regularizer to anisotropic filtering for **video denoising**. The extension to videos from images is natural. Say a video contains f frames of size $n \times m$ pixels; this video can be viewed as a 3D-tensor $\mathsf{X} \in \mathbb{R}^{n\times m\times f}$, on which a 3D-TV based filter can be effected by

$$\min_{\mathsf{X}} \quad \tfrac{1}{2}\|\mathsf{X}-\mathsf{U}_0\|_{\mathrm{F}}^2 + \lambda \mathrm{Tv}_{p_1,p_2,p_3}^{3\mathrm{D}}(\mathsf{X}), \tag{5.6}$$

where U_0 is the observed noisy video, and $\mathrm{Tv}_{p_1,p_2,p_3}^{3\mathrm{D}} = \mathrm{Tv}_{\boldsymbol{p}}^3$ with $\boldsymbol{p} = [p_1, p_2, p_3]$. Application of the filter (5.6) is nothing but computation of the prox operator, which can be done using the Parallel-Proximal Dykstra (PPD) algorithm presented in Sec. 4.

We apply this idea to the video sequences detailed in Table 7. All of the sequences are made of grayscale pixels. Figure 30 in the Appendix shows some of the frames of the *salesman* sequence. We noise every frame of these sequences by applying gaussian noise with zero mean and variance 0.01, using Matlab's *imnoise* function. Then we solve problem 5.6 for each sequence, adjusting the regularization value so as to maximize ISNR of the reconstructed signal. We test the following algorithms, which have been previously applied in the literature for solving 3D-TV, with the only exception Parallel Proximal Dykstra:

- Parallel Proximal Dykstra (§ 4.1.1).

Sequence	Frame resolution	Number of frames	Total number of pixels
salesman	288 × 352	50	5 million
coastguard	176 × 144	300	7.6 million
bicycle	720 × 576	30	12.4 million

Table 7: Size details of video sequences used in the video denoising experiments.

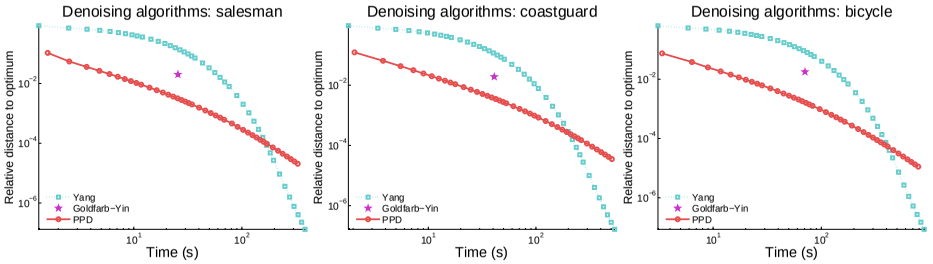


Figure 23: Relative distance to optimum vs time of the denoising 3D-TV algorithms under comparison, for the different video sequences considered in the experiments.

- Yang's method, which is based on ADMM (§ 4.1.1)
- The maximum flow approach by Goldfarb and Yin (2009), which features an implementation for 3D grids, thus solving a discrete-valued version of 3D-TV.

For both PPD and ADMM we again make use of linearized taut-string 1D TV solver. We must also point out that other image denoising methods seem amenable for extension into the multidimensional setting, such as Condat's and Chambolle-Pock methods. However in the light of our image denoising results we do not deem them as good choices for this problem. A more reasonable choice might be to extend Split-Bregman to multiple dimensions, but such an extension has not been implemented or proposed as far as we know. We would also like to note that we have considered extending the Douglas Rachford method to a multidimensional setting, however such task is complex and thus we decided to focus on Parallel Proximal Dykstra.

Similarly to our previous image denoising experiments, we ran the algorithms under comparison for each video sequence and measured its ISNR and relative distance to the optimal objective value of the current solution at each iteration through their execution. Again the exception is the Goldfarb-Yin method, which is non–iterative and so we only report the time required for its termination. The optimal objective value was estimated by running all methods for a very large number of iterations and taking the minimum value of them all. This produced the plots shown in Figures 23–24. From them the following observations are of relevance:

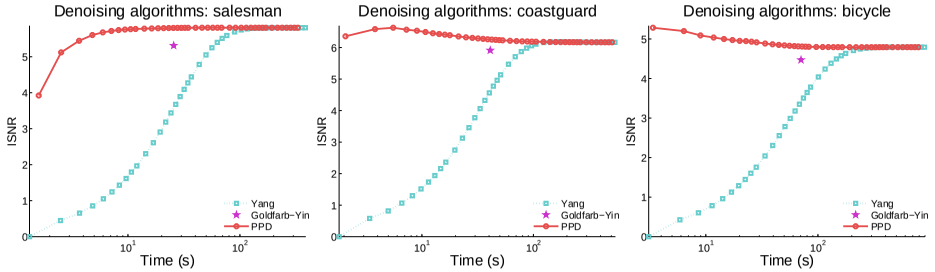


Figure 24: Increased Signal to Noise Ratio (ISNR) vs time of the denoising 3D-TV algorithms under comparison, for the different video sequences considered in the experiments.

- Following the pattern observed in the image denoising experiments, ADMM (Yang's method) is best suited for finding very accurate solutions.
- The method by Goldfarb and Yin again provides suboptimal solutions, due to the discrete approximation it uses.
- Parallel Proximal Dykstra is the fastest to achieve a mid-quality solution.
- Intermediate solutions prior to convergence of the PPD run result in better ISNR values for the *coastguard* and *bicycle* data sets. This hints that the denoising model used in this experiment may not be optimal for these kind of signals; indeed, more advanced denoising models abound in the signal processing literature. Hence we do not claim novel results in terms of ISNR quality, but just in solving this classic denoising model more efficiently.

The ISNR plots in Figure 24 also show how both Parallel Proximal Dykstra and ADMM (Yang's method) converge to equivalent solutions in practice. Therefore, for the purpose of video denoising PPD seems to be the best choice, unless for some reason a high degree of accuracy is required, for which ADMM should be preferred.

Acknowledgments

ÁB acknowledges partial financial support from Spain's grants TIN2010-21575-C02-01, TIN2013-42351-P, S2013/ICE-2845 CASI-CAM-CM, TIN2016-76406-P, TIN2015-70308-REDT (MINECO/FEDER EU) and project "FACIL—Ayudas Fundación BBVA a Equipos de Investigación Científica 2016" during the long research period leading to the writing of this manuscript. We thank R. Tibshirani for bringing (Johnson, 2013) to our attention, and S. Jegelka for alerting us to the importance of weighted total-variation problems.

Appendix A. Mathematical Background

We begin by recalling a few basic ideas from convex analysis; we recommend the recent book (Bauschke and Combettes, 2011) for more details.

Let $\mathcal{X} \subset \mathbb{R}^n$ be any set. A function $r : \mathcal{X} \to \mathbb{R} \cup \{-\infty, +\infty\}$ is called *lower semicontinuous* if for every $\boldsymbol{x} \in \mathcal{X}$ and a sequence $(\boldsymbol{x}_k)$ that converges to $\boldsymbol{x}$, it holds that

$$\boldsymbol{x}_k \to \boldsymbol{x} \implies r(\boldsymbol{x}) \leq \liminf_k r(\boldsymbol{x}_k). \tag{A.1}$$

The set of proper lsc convex functions on $\mathcal{X}$ is denoted by $\Gamma_0(\mathcal{X})$ (such functions are also called *closed convex functions*). The *indicator function* of a set C is defined as

$$\delta_C : \mathcal{X} \to [0, \infty] : \boldsymbol{x} \mapsto \begin{cases} 0, & \text{if } \boldsymbol{x} \in C; \\ \infty, & \text{if } \boldsymbol{x} \notin C, \end{cases} \tag{A.2}$$

which is lsc if and only if C is closed.

The *convex conjugate* of r is given by $r^*(\boldsymbol{z}) := \sup_{\boldsymbol{x} \in \operatorname{dom} r} \langle \boldsymbol{x}, \boldsymbol{z} \rangle - r(\boldsymbol{x})$, and a particularly important example is the Fenchel conjugate of a norm $\|\cdot\|$

$$\text{if } r = \|\cdot\|, \quad \text{then } \; r^* = \delta_{\|\cdot\|_* \leq 1}, \tag{A.3}$$

where the norm $\|\cdot\|_*$ is dual to $\|\cdot\|$. Let r and h be proper convex functions. The *infimal convolution* of r with h is the convex function given by $(r \square h)(\boldsymbol{x}) := \inf_{\boldsymbol{y} \in \mathcal{X}} \big(r(\boldsymbol{y}) + h(\boldsymbol{x} - \boldsymbol{y})\big)$. For our purposes, the most important special case is infimal convolution of a convex function with the squared euclidean norm, which yields the *Moreau envelope* (Moreau, 1962).

Proposition A.1 *Let $r \in \Gamma_0(\mathcal{X})$ and let $\gamma > 0$. The* Moreau envelope *of r indexed by γ is*

$$E_r^\gamma(\cdot) := r \,\square\, (\tfrac{1}{2\gamma}\|\cdot\|_2^2). \tag{A.4}$$

The Moreau envelope (A.4) *is convex, real-valued, and continuous.*

Proof See e.g. (Bauschke and Combettes, 2011, Prop. 12.15). ∎

Using the Moreau envelope (A.4), we now formally introduce prox operators.

Definition A.2 (Prox operator) *Let $r \in \Gamma_0(\mathcal{X})$, and let $\boldsymbol{y} \in \mathcal{X}$. Then* $\operatorname{prox}_r \boldsymbol{y}$ *is the unique point in $\mathcal{X}$ that satisfies $E_r^1(\boldsymbol{y}) = \min_{\boldsymbol{x} \in \mathcal{X}} (r(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2)$, i.e.,*

$$\operatorname{prox}_r(\boldsymbol{y}) := \operatorname*{argmin}_{\boldsymbol{x} \in \mathcal{X}} r(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{y}\|_2^2, \tag{A.5}$$

and the nonlinear map $\operatorname{prox}_r : \mathcal{X} \to \mathcal{X}$ is called the prox operator *of r.*

Sometimes the Fenchel conjugate r^* is easier to use than r; similarly, sometimes the operator $\operatorname{prox}_{r^*}$ is easier to compute than prox_r. The result below shows the connection.

Proposition A.3 (Moreau decomposition) *Let $r \in \Gamma_0(\mathcal{X})$, $\gamma > 0$, and $\boldsymbol{y} \in \mathcal{X}$. Then,*

$$\boldsymbol{y} = \operatorname{prox}_{\gamma r} \boldsymbol{y} + \gamma \operatorname{prox}_{r^*/\gamma}(\gamma^{-1}\boldsymbol{y}). \tag{A.6}$$

Proof A brief exercise; see e.g., (Bauschke and Combettes, 2011, Thm. 14.3). ∎

This decomposition provides the necessary tools to exploit useful primal–dual relations. For the sake of clarity we also present an additional result regarding a particular primal-dual relation that plays a key role in our algorithms.

Proposition A.4 *Let $f \in \Gamma_0(\mathcal{X})$ and $r \in \Gamma_0(\mathcal{Z})$. The problems below form a primal-dual pair.*

$$\inf_{\boldsymbol{x} \in \mathcal{X}} \quad f(\boldsymbol{x}) + r(\boldsymbol{B}\boldsymbol{x}) \quad \textit{s.t.} \quad \boldsymbol{B}\boldsymbol{x} \in \mathcal{Z} \tag{A.7}$$

$$\inf_{\boldsymbol{u} \in \mathcal{Z}} \quad f^*(-\boldsymbol{B}^T\boldsymbol{u}) + r^*(\boldsymbol{u}). \tag{A.8}$$

Proof Introduce an extra variable $\boldsymbol{z} = \boldsymbol{B}\boldsymbol{x}$, dual function is

$$g(\boldsymbol{u}) = \inf_{\boldsymbol{x} \in \mathcal{X}} \; f(\boldsymbol{x}) + \boldsymbol{u}^T\boldsymbol{B}\boldsymbol{x} + \inf_{\boldsymbol{z} \in \mathcal{Z}} \; r(\boldsymbol{z}) - \boldsymbol{u}^T\boldsymbol{z},$$

which upon rewriting using Fenchel conjugates yields (A.8). ∎

Notions on submodular optimization are also required to introduce some of the decomposition techniques for 2D-TV in this paper. For a more thorough read on this topic we recommend the monograph Bach (2013).

Definition A.5 (Submodular function) *A set-function $F : 2^V \to \mathbb{R}$, for 2^V the power set of some set V, is submodular if and only if it fulfills the diminishing returns property, that is, for $A \subseteq B \subseteq V$ and $k \in V$, $k \notin B$ we have*

$$F(A \cup \{k\}) - F(A) \geq F(B \cup \{k\}) - F(B).$$

Intuitively, a set-function is submodular if adding a new element to the set results in less value as the set grows in size.

Definition A.6 (Modular function) *A set-function $F : 2^V \to \mathbb{R}$, for 2^V the power set of some set V, $F(\emptyset) = 0$ is modular (and also submodular) if and only if there exists $\boldsymbol{s} \in \mathbb{R}^p$ such that $F(A) = \sum_{k \in A} \boldsymbol{s}_k$.*

That is, a function is modular if it always assigns the same value for each element added to the set, regardless of the other elements in the set. A common shorthand for modular functions is $s(A) = \sum_{k \in A} \boldsymbol{s}_k$.

Submodular functions can be thought as convex functions in the realm of discrete optimization, in the sense that they feature useful properties that allow for efficient optimization. Similarly, modular functions are connected to linear functions. To make such connections explicit we require of the following geometric concepts.

Definition A.7 (Base polytope) *The base polytope B_F of a submodular function F is the polyhedron given by*

$$B_F = \{y \in \mathbb{R}^n : y(A) \leq F(A) \; \forall A \subseteq V, \quad y(V) = F(V)\}.$$

That is, the base polytope is a polyhedron defined through linear inequality constraints on the values of F for every one of the n elements of the powerset 2^V, and an equality constraint for the complete set. This results in a combinatorial number of contraints, but fortunately this polytope will not be used directly.

Definition A.8 (Support function) *The support function h_A for some non-empty closed convex set $A \in \mathbb{R}^n$ is given by*

$$h_A(\boldsymbol{x}) = \sup\left\{\boldsymbol{x}^T\boldsymbol{a} : \boldsymbol{x} \in A\right\}.$$

The support function is useful when connected with the following definition.

Definition A.9 (Lovász extension) *Suppose a set-function F such that $F(\emptyset) = 0$. Its Lovász extension $f : \mathbb{R}^p \to \mathbb{R}$ is defined through the following mechanism. Take $\boldsymbol{w} \in \mathbb{R}^p$ input to f, and order its components in decreasing order $\boldsymbol{w}_{j_1} \geq \ldots \geq \boldsymbol{w}_{j_p}$, then*

$$f(\boldsymbol{w}) = \sum_{k=1}^{p}[F(\{j_1, \ldots, j_k\}) - F(\{j_1, \ldots, j_{k-1}\})].$$

Other equivalent definitions are possible: see Bach (2013) for details. The following result links all the definitions so far.

Proposition A.10 *For F submodular function such that $F(\emptyset) = 0$ we have*

- *Its Lovász extension f is a convex function.*
- *The support function of its base polytope is equal to its Lovász extension, that is, $h_{B_F}(\boldsymbol{x}) = f(\boldsymbol{x})$.*
- *The problem $\min_{S\subseteq V} F(S)$ is dual to $\min_{\boldsymbol{x}} f(\boldsymbol{x}) + \frac{1}{2}\|\boldsymbol{x}\|_2^2$, with $S^* = \{k|\boldsymbol{x}_k^* \geq 0\}$.*

For proofs on these points we refer to Bach (2013). The takeaway from them is that any minimization on a submodular function can be cast into a convex optimization problem. Furthermore, for those convex minimization problems whose objective turns out to be the Lovász extension of some other function, we can trace the steps the other way round, obtaining the minimization of a submodular function.

Consider now a composite problem $\min_{S\subseteq V}\sum_j F_j(S)$. The following results hold

Proposition A.11 *The problem $\min_{S\subseteq V}\sum_j F_j(S)$ is equivalent to $\min_x \sum_j f_j(x) + \frac{1}{2}\|x\|_2^2$, with $S^* = \{k|x_k^* \geq 0\}$. Furthermore it is also equivalent to $\min_{y_j\in B_{F_j}\,\forall j}\frac{1}{2}\|\sum_j y_j\|_2^2$, with $x^* = -\sum_j y_j^*$.*

Proof The first equivalence is a direct result of the properties of Lovász extensions (Bach, 2013), in particular that for F, G set-functions with Lovász extensions f, g, the Lovász extension of $F + G$ is $f + g$. For the second equivalence we have:

$$\begin{aligned}
\min_{\boldsymbol{x}} \sum_j f_j(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x}\|_2^2 &= \min_{\boldsymbol{x}} \sum_j h_{B_{F_j}} + \tfrac{1}{2}\|\boldsymbol{x}\|_2^2, \\
&= \min_{\boldsymbol{x}} \sum_j \max_{y_j\in B_{F_j}} \boldsymbol{y}_j^T\boldsymbol{x} + \tfrac{1}{2}\|\boldsymbol{x}\|_2^2, \\
&= \max_{\boldsymbol{y}_j\in B_{F_j}\,\forall j} \min_{\boldsymbol{x}} \left(\sum_j \boldsymbol{y}_j^T\boldsymbol{x} + \tfrac{1}{2}\|\boldsymbol{x}\|_2^2\right), \\
&= \min_{\boldsymbol{y}_j\in B_{F_j}\,\forall j} \tfrac{1}{2}\|\sum_j \boldsymbol{y}_j\|_2^2,
\end{aligned}$$

and the dual relationship $\boldsymbol{x}^* = -\sum_j \boldsymbol{y}_j^*$ comes from solving the inner $\min_{\boldsymbol{x}}$ problem for $\boldsymbol{x}$. ■

Therefore any decomposable submodular minimization, or sum of Lovász extensions plus ℓ_2 term, can be casted into a geometric problem in terms of the base polytopes. For two functions the resultant problem is of special interest if rewritten as

$$\min_{\substack{\boldsymbol{y}_1\in B_{F_1}\\ \boldsymbol{y}_2\in B_{F_2}}} \tfrac{1}{2}\|\boldsymbol{y}_1 + \boldsymbol{y}_2\|_2^2 \quad = \quad \min_{\substack{\boldsymbol{y}_1\in B_{F_1}\\ -\boldsymbol{y}_2\in -B_{F_2}}} \tfrac{1}{2}\|\boldsymbol{y}_1 - (-\boldsymbol{y}_2)\|_2^2 \quad = \quad \min_{\substack{\boldsymbol{a}\in B_{F_1}\\ \boldsymbol{b}\in -B_{F_2}}} \tfrac{1}{2}\|\boldsymbol{a} - \boldsymbol{b}\|_2^2$$

with $\boldsymbol{a} = \boldsymbol{y}_1$, $\boldsymbol{b} = -\boldsymbol{y}_2$, as this results in the classic geometric problem of finding the closest points between two convex sets. Many algorithms have been proposed to tackle problems in this form, most of them making use of alternating projection operations onto the two sets. Thus, a legitimate concern is how easy it is to compute such projections for B_{F_1} and $-B_{F_2}$.

Proposition A.12 *Given a submodular function F and its base polytope B_F, the projections $\Pi_{B_F}(\boldsymbol{z})$ and $\Pi_{-B_F}(\boldsymbol{z})$ of a point $\boldsymbol{z}$ onto B_F or its negated counterpart can be computed as*

$$\begin{aligned}
\Pi_{B_F}(\boldsymbol{z}) &= \boldsymbol{z} - \operatorname{prox}_f(\boldsymbol{z}), \\
\Pi_{-B_F}(\boldsymbol{z}) &= \boldsymbol{z} + \operatorname{prox}_f(-\boldsymbol{z}),
\end{aligned}$$

with prox *proximity operator of a function, f the Lovász extension of F.*

Proof We start with the proximity of f and work our way to a relationship with the projection operator,

$$\begin{aligned}
\operatorname{prox}_f(\boldsymbol{z}) &\equiv \min_{\boldsymbol{x}} f(\boldsymbol{x}) + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \\
&= \max_{\boldsymbol{y}\in B_F} \min_{\boldsymbol{x}} \boldsymbol{y}^T\boldsymbol{x} + \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{z}\|_2^2, \\
&= \max_{\boldsymbol{y}\in B_F} \boldsymbol{y}^T(\boldsymbol{z} - \boldsymbol{y}) + \tfrac{1}{2}\|(\boldsymbol{z} - \boldsymbol{y}) - \boldsymbol{z}\|_2^2, \\
&= \min_{\boldsymbol{y}\in B_F} \tfrac{1}{2}\|\boldsymbol{y}\|_2^2 - \boldsymbol{y}^T\boldsymbol{z}, \\
&\equiv \min_{\boldsymbol{y}\in B_F} \tfrac{1}{2}\|\boldsymbol{y} - \boldsymbol{z}\|_2^2 = \Pi_{B_F}(\boldsymbol{z}),
\end{aligned}$$

where solving the inner minimization problem for $\boldsymbol{x}$ gives the primal–dual relationship $\boldsymbol{x}^* = \boldsymbol{z} - \boldsymbol{y}^*$. Using this we can obtain the solution for the projection problem from the proximity problem, as $\Pi_{B_F}(\boldsymbol{z}) = \boldsymbol{z} - \text{prox}_f(\boldsymbol{z})$. Projection onto the negated base polytope follows from the basic geometric argument $\Pi_{-B_F}(\boldsymbol{z}) = -\Pi_{B_F}(-\boldsymbol{z})$. ■

Appendix B. proxTV Toolbox

All the Total–Variation proximity solvers in this paper have been implemented as the **proxTV** toolbox for C++, Matlab and Python, available at `https://github.com/albarji/proxTV`. The toolbox has been designed to be used out of the box in a user friendly way; for instance, the top–level Matlab function `TV` solves Total–Variation proximity for a given signal under a variety of settings. For instance

>> TV(X, lambda)

solves Tv_1 proximity for a signal `X` of any dimension and a regularization value `lambda`. The weighted version of this problem is also seamlessly tackled by just providing a vector of weights of the appropriate length as the `lambda` parameter.

If a third parameter `p` is provided as

>> TV(X, lambda, p)

the general Tv_p proximity problem is addressed, whereupon an adequate solver is chosen by the library.

More advanced uses of the library are possible, allowing to specify which norm `p` and regularizer `lambda` values to use for each dimension of the signal, and even applying combinations of several different Tv_p regularizers along the same dimension. Please refer to the documentation within the toolbox for further information.

Appendix C. Proof on the Equality of Taut-String Problems

Theorem C.1 (Equality of taut-string problems) *Given the problems*

$$\min_{\boldsymbol{s}} \sum_{i=1}^{n} (\boldsymbol{s}_i - \boldsymbol{s}_{i-1})^2, \ s.t. \ |\boldsymbol{s}_i - \boldsymbol{r}_i| \leq \boldsymbol{w}_i \, \forall i = 1, \dots, n-1, \boldsymbol{s}_0 = 0, \boldsymbol{s}_n = \boldsymbol{r}_n, \tag{C.1}$$

and

$$\min_{\hat{\boldsymbol{s}}} \sum_{i=1}^{n} \sqrt{1 + (\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{i-1})^2}, \ s.t. \ |\hat{\boldsymbol{s}}_i - \boldsymbol{r}_i| \leq \boldsymbol{w}_i \, \forall i = 1, \dots, n-1, \hat{\boldsymbol{s}}_0 = 0, \hat{\boldsymbol{s}}_n = \boldsymbol{r}_n, \tag{C.2}$$

for a non-zero vector $\boldsymbol{w}$, both problems share the same minimum $\boldsymbol{s}^* = \hat{\boldsymbol{s}}^*$.

Proof

The Lagrangian of problem C.1 takes the form

$$L(\boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{\beta}) = \sum_{i=1}^{n} (\boldsymbol{s}_i - \boldsymbol{s}_{i-1})^2 + \sum_{i=1}^{n-1} \boldsymbol{\alpha}_i (\boldsymbol{s}_i - \boldsymbol{r}_i - \boldsymbol{w}_i) + \sum_{i=1}^{n-1} \boldsymbol{\beta}_i (-\boldsymbol{w}_i - \boldsymbol{s}_i + \boldsymbol{r}_i),$$

and its Karush-Kuhn-Tucker optimality conditions are given by

$$(\boldsymbol{s}_{i+1} - \boldsymbol{s}_i) - (\boldsymbol{s}_i - \boldsymbol{s}_{i-1}) = \boldsymbol{\alpha}_i - \boldsymbol{\beta}_i, \tag{C.3}$$

$$|\boldsymbol{s}_i - \boldsymbol{r}_i| \leq \boldsymbol{w}_i, \tag{C.4}$$

$$\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \geq 0, \tag{C.5}$$

$$\boldsymbol{\alpha}_i (\boldsymbol{s}_i - \boldsymbol{r}_i - \boldsymbol{w}_i) = 0, \tag{C.6}$$

$$\boldsymbol{\beta}_i (-\boldsymbol{w}_i - \boldsymbol{s}_i + \boldsymbol{r}_i) = 0, \tag{C.7}$$

$\forall i = 1, \dots, n-1$, and where the first equation comes from the fact that $\frac{\partial L(\boldsymbol{s}, \boldsymbol{\alpha}, \boldsymbol{\beta})}{\partial \boldsymbol{s}} = 0$ at the minimum.

As the only difference between problems C.1 and C.2 is in the form of the objective, the KKT conditions for problem C.2 take the same form, but for the first one,

$$\frac{(\hat{\boldsymbol{s}}_{i+1} - \hat{\boldsymbol{s}}_i)}{\sqrt{1 + (\hat{\boldsymbol{s}}_{i+1} - \hat{\boldsymbol{s}}_i)^2}} - \frac{(\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{i-1})}{\sqrt{1 + (\hat{\boldsymbol{s}}_i - \hat{\boldsymbol{s}}_{i-1})^2}} = \hat{\boldsymbol{\alpha}}_i - \hat{\boldsymbol{\beta}}_i, \tag{C.8}$$

$$|\hat{\boldsymbol{s}}_i - \boldsymbol{r}_i| \leq \boldsymbol{w}_i, \tag{C.9}$$

$$\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i \geq 0, \tag{C.10}$$

$$\hat{\boldsymbol{\alpha}}_i (\hat{\boldsymbol{s}}_i - \boldsymbol{r}_i - \boldsymbol{w}_i) = 0, \tag{C.11}$$

$$\hat{\boldsymbol{\beta}}_i (-\boldsymbol{w}_i - \hat{\boldsymbol{s}}_i + \boldsymbol{r}_i) = 0, \tag{C.12}$$

$\forall i = 1, \dots, n-1$, and where we use hat notation for the dual coefficients to tell them apart from those of problem C.1.

Suppose $\boldsymbol{s}^*$ minimizer to problem C.1, hence fulfilling the conditions C.3-C.7. In particular this means that it is feasible to assign values to the dual coefficients $\boldsymbol{\alpha}, \boldsymbol{\beta}$ in such a way that the conditions above are met. If we set $\hat{\boldsymbol{s}} = \boldsymbol{s}^*$ in the conditions C.8-C.12 the following observations are of relevance

- Condition C.9 becomes the same as condition C.4, and so it is immediately met.
- The operator $f(x) = \frac{x}{\sqrt{1+x^2}}$ is contractive and monotonous.
- The couple $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ cannot be both non–zero at the same time, since $\boldsymbol{\alpha}_i > 0$ enforces $\boldsymbol{s}_i = \boldsymbol{r}_i + \boldsymbol{w}_i$ and $\boldsymbol{\beta}_i > 0$ enforces $\boldsymbol{s}_i = \boldsymbol{r}_i - \boldsymbol{w}_i$, and $\boldsymbol{w}_i$ is non–zero.
- Hence and because $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i \geq 0$ and condition C.3 holds, when $(\boldsymbol{s}_{i+1} - \boldsymbol{s}_i) - (\boldsymbol{s}_i - \boldsymbol{s}_{i-1}) > 0$ then $\boldsymbol{\alpha}_i > 0$, $\boldsymbol{\beta}_i = 0$, and when $(\boldsymbol{s}_{i+1} - \boldsymbol{s}_i) - (\boldsymbol{s}_i - \boldsymbol{s}_{i-1}) < 0$ then $\boldsymbol{\alpha}_i = 0$, $\boldsymbol{\beta}_i > 0$.
- $f(\boldsymbol{s}_{i+1} - \boldsymbol{s}_i) - f(\boldsymbol{s}_i - \boldsymbol{s}_{i-1})$ has the same sign as $(\boldsymbol{s}_{i+1} - \boldsymbol{s}_i) - (\boldsymbol{s}_i - \boldsymbol{s}_{i-1})$, since f is monotonous and as such preserves ordering.
- Since f is contractive, condition C.8 can be met by setting $(\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i) = (k\boldsymbol{\alpha}_i, k\boldsymbol{\beta}_i)$ for some $0 \leq k < 1$. Note that this works because $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i)$ cannot be both zero at the same time.
- Condition C.10 is met for those choices of $\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i$, as C.5 was met for $\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i$ and $0 \leq k < 1$.

- Conditions C.11 and C.12 are also met for those choices of $\hat{\boldsymbol{\alpha}}_i, \hat{\boldsymbol{\beta}}_i$, as $\hat{\boldsymbol{\alpha}}_i(\boldsymbol{s}_i - \boldsymbol{r}_i - \boldsymbol{w}_i) = k\boldsymbol{\alpha}_i(\boldsymbol{s}_i - \boldsymbol{r}_i - \boldsymbol{w}_i) = 0$ and $\hat{\boldsymbol{\beta}}_i(-\boldsymbol{w}_i - \boldsymbol{s}_i + \boldsymbol{r}_i) = k\boldsymbol{\beta}_i(-\boldsymbol{w}_i - \boldsymbol{s}_i + \boldsymbol{r}_i) = 0$.

Therefore, all of the optimality conditions C.8-C.12 for problem C.2 are met for $\boldsymbol{s}^*$ solution of problem C.1, and so a minimum of problem C.1 is also a minimum for problem C.2.

The proof can be repeated the other way round by setting $\boldsymbol{s} = \hat{\boldsymbol{s}}^*$ optimal for problem C.2, defining the operator $f^{-1}(x) = \frac{x}{\sqrt{1-x^2}}$, and observing that this operator is monotonous and expansive, so we can establish $(\boldsymbol{\alpha}_i, \boldsymbol{\beta}_i) = (k\hat{\boldsymbol{\alpha}}_i, k\hat{\boldsymbol{\beta}}_i)$ for some $k \geq 1$ and the optimality conditions C.3-C.7 for problem C.1 are met following a similar reasoning to the one presented above. Thus, a minimum for problem C.2 is also a minimum for problem C.1, which joined with the previous result completes the proof. ■

Appendix D. Proof on the Equivalence of Linearized Taut-String Method

Proposition D.1 *Using affine approximations to the greatest convex minorant and the smallest concave majorant does not change the solution of the taut-string method.*

Proof Let us note $\cap(f)$ as the smallest concave majorant of some function f taking integer values, $\cup(f)$ as the greatest concave minorant, $\bar{a}(f)$ as the smallest affine majorant and $\underline{a}(f)$ as the greatest affine minorant. By definition we have

$$\underline{a}(f(i)) \leq \cup(f(i)) \leq f(i) \leq \cap(f(i)) \leq \bar{a}(f(i)) \quad \forall i \in \mathbb{Z}$$

Consider now the nature of the taut-string problem, where a vertically symmetric tube of radius λ_i at each section is modelled by following the majorant of the tube bottom $(f-\lambda)$ and the minorant of the tube ceiling $(f+\lambda)$. We work the inequalities above as:

$$f(i) - \lambda_i \leq \cap(f(i) - \lambda_i) \leq \bar{a}(f(i) - \lambda_i)$$
$$\underline{a}(f(i) + \lambda_i) \leq \cup(f(i) + \lambda_i) \leq f(i) + \lambda_i$$

We will show that an overlap of smallest concave majorant / greatest convex minorant takes place iff the same overlap happens when using the affine approximations. We formally define overlap as the setting where for a point i we have $\cup(f_i + \lambda_i) \leq \cap(f_i - \lambda_i)$.

One side of the implication is easy: if $\cup(f(i) + \lambda_i) \leq \cap(f(i) - \lambda)$ for some i, then using the relations above we have $\underline{a}(f(i) + \lambda_i) \leq \cup(f(i) + \lambda_i) \leq \cap(f(i) - \lambda_i) \leq \bar{a}(f(i) - \lambda_i)$, and so the affine approximation detects any overlap taking place in the concave/convex counterpart.

The opposite requires the key observation that in the taut-string method both majorant and minorant functions are clamped to the same point of origin: $f(0) = 0$ at the start of the method, or the point where the last segment was fixed after each restart. Let us assume $f(0) = 0$ without loss of generality. Suppose now that an overlap is detected by the affine approximation. Because of this affine nature the majorant/minorant slopes are constant,

i.e.

$$\bar{\delta}_1 = \bar{\delta}_2 = \ldots = \bar{\delta}_n = \bar{\delta}, \quad \underline{\delta}_1 = \underline{\delta}_2 = \ldots = \underline{\delta}_n = \underline{\delta}.$$

However, if we consider the convex/concave approximations these slopes can increase/decrease as the segment progresses, that is:

$$\delta_1^{\cup} \leq \delta_2^{\cup} \leq \ldots \leq \delta_n^{\cup}, \quad \delta_1^{\cap} \geq \delta_2^{\cap} \geq \ldots \geq \delta_n^{\cap}.$$

Consider now the majorant/minorant values, expressed through the slopes and taking into account the observation above about the starting point.

$$\cap(f(i) - \lambda_i) = \sum_{j=1}^{i} \delta_j^{\cap}, \quad \cup(f(i) + \lambda_i) \quad = \sum_{j=1}^{i} \delta_j^{\cup}, \quad \bar{a}(f(i) - \lambda_i) = i\bar{\delta}, \quad \underline{a}(f(i) + \lambda_i) \quad = i\underline{\delta}.$$

Since an overlap has been detected in the affine approximation, we have that for some point i

$$i\underline{\delta} = \underline{a}(f(i) + \lambda_i) \leq \bar{a}(f(i) - \lambda_i) = i\bar{\delta},$$

so $\underline{\delta} \leq \bar{\delta}$. Consider now the values of the affine minorant/majorant at the point immediately after the origin,

$$\underline{a}(f_1 - \lambda_1) = \underline{\delta}, \quad \bar{a}(f_1 + \lambda_1) = \bar{\delta}.$$

We will show now that the convex/convex counterpart must take exactly the same values at these points. To do so we take into account the following fact: there must exist points x and y, $x, y \leq i$, where

$$\underline{a}(f_x + \lambda_x) = f_x + \lambda_x = \cup(f_x + \lambda_x), \quad \bar{a}(f_y - \lambda_y) = f_y - \lambda_y = \cap(f_y - \lambda_y),$$

that is to say, the affine minorant/majorant must touch the tube ceiling/bottom at some point, otherwise we could obtain a greater minorant / smaller majorant by reducing this distance. The equalities to the convex minorant / concave majorant are then obtained by exploiting the inequalities at the beginning of the proof.

By the already presented inequalities $\cup(f_1 + \lambda_1) \geq \underline{a}(f_1 + \lambda_1)$, but let us suppose for a moment $\cup(f_1 + \lambda_1) > \underline{a}(f_1 + \lambda_1)$. This would imply $\delta_1^{\cup} > \underline{\delta}$. We then would have that at the touching point x

$$f_x + \lambda_x = \underline{a}(f_x + \lambda_x) = x\underline{\delta} < x\delta_1^{\cup} \leq \cup(f_1 + \lambda_1),$$

as the slopes in a convex minorant must be monotonically increasing. However, such function would not be a valid convex minorant, as it would grow over $f + \lambda$. Therefore $\cup(f_1 + \lambda_1) = \underline{a}(f_1 + \lambda_1)$ must hold. Using a symmetric argument, $\cap(f_1 - \lambda_1) = \bar{a}(f_1 - \lambda_1)$ can also be shown to hold. Joining this with the previous facts we have that

$$\cup(f_1 + \lambda_1) = \underline{a}(f_1 + \lambda_1) = \underline{\delta} \leq \bar{\delta} = \bar{a}(f_1 - \lambda_1) = \cap(f_1 - \lambda_1),$$

and therefore the overlap detected by the affine approximation is detected through its convex/concave version as well through $\cup(f_1 + \lambda_1) \leq \cap(f_1 - \lambda_1)$. ■

Appendix E. Projected-Newton for Weighted $\mathbf{Tv}_1^{\mathbf{1D}}$

In this appendix we present details of a projected-Newton (PN) approach to solving the weighted-TV problem (2.6). Although taut-string approaches are empirically superior to this PN approach, the details of this derivation prove to be useful when developing subroutines for handling ℓ_p-norm TV prox-operators, but perhaps their greatest use lies in presenting a general method that could be applied to other problems that have structures similar to TV, e.g., group total-variation (Alaız et al., 2013; Wytock et al., 2014) and ℓ_1-trend filtering (Kim et al., 2009; Tibshirani, 2014).

The weighted-TV dual problem (2.7) is a bound-constrained QP, so it could be solved using a variety of methods such as TRON (Lin and Moré, 1999), L-BFGS-B (Byrd et al., 1994), or projected-Newton (PN) (Bertsekas, 1982). Obviously, these methods will be inefficient if invoked off-the-shelf; exploitation of problem structure is a must for solving (2.7) efficiently. PN lends itself well to such structure exploitation; we describe the details below.

PN runs iteratively in three key steps: first it identifies a special subset of *active variables* and uses these to compute a *reduced* Hessian. Then, it uses this Hessian to scale the gradient and move in the direction opposite to it, damping with a stepsize, if needed. Finally, the next iterate is obtained by projecting onto the constraints, and the cycle repeats. PN can be regarded as an extension of the gradient-projection method (GP, Bertsekas (1999)), where the components of the gradient that make the updating direction infeasible are removed; in PN both the gradient and the Hessian are *reduced* to guarantee this feasibility.

At each iteration PN selects the active variables

$$I := \{i \mid (u_i = -w_i \text{ and } [\nabla\phi(\boldsymbol{u})]_i > \epsilon) \quad \text{or} \quad (u_i = w_i \text{ and } [\nabla\phi(\boldsymbol{u})]_i < -\epsilon)\}\,, \tag{E.1}$$

where $\epsilon \geq 0$ is small scalar. This corresponds to the set of variables at a bound, and for which the gradient points inside the feasible region; that is, for these variables to further improve the objective function we would have to step out of bounds. It is thus clear that these variables are of no use for this iteration, so we define the complementary set $\bar{I} := \{1 \dots n\} \setminus I$ of indices not in I, which are the variables we are interested in updating. From the Hessian $\boldsymbol{H} = \nabla^2\phi(u)$ we extract the *reduced Hessian* $\boldsymbol{H}_{\bar{I}}$ by selecting rows and columns indexed by $\bar{I}$, and in a similar way the *reduce gradient* $[\nabla\phi(\boldsymbol{u})]_{\bar{I}}$. Using these we perform a Newton–like "reduced" update in the form

$$\boldsymbol{u}_{\bar{I}} \leftarrow P(\boldsymbol{u}_{\bar{I}} - \alpha \boldsymbol{H}_{\bar{I}}^{-1}[\nabla\phi(\boldsymbol{u})]_{\bar{I}}), \tag{E.2}$$

where α is a stepsize, and P denotes projection onto the constraints, which for box–constraints reduces to simple element–wise projection. Note that only the variables in the set $\bar{I}$ are updated in this iterate, leaving the rest unchanged. While such update requires computing the inverse of the reduced Hessian $\boldsymbol{H}_{\bar{I}}$, which in the general case can amount to computational costs in the $O(n^3)$ order, we will see now how exploiting the structure of the problem allows us to perform all the steps above efficiently.

First, observe that for (2.7) the Hessian is

$$\boldsymbol{H} = \boldsymbol{D}\boldsymbol{D}^T = \begin{pmatrix} 2 & -1 & & & \\ -1 & 2 & -1 & & \\ & -1 & 2 & \ddots & \\ & & \ddots & \ddots & -1 \\ & & & -1 & 2 \end{pmatrix} \in \mathbb{R}^{(n-1)\times(n-1)}.$$

Next, observe that whatever the active set I, the corresponding reduced Hessian $\boldsymbol{H}_{\bar{I}}$ remains symmetric tridiagonal. This observation is crucial because then we can quickly compute the updating direction $\boldsymbol{d}_{\bar{I}} = \boldsymbol{H}_{\bar{I}}^{-1}[\nabla\phi(\boldsymbol{u})]_{\bar{I}}$, which can be done by solving the linear system $\boldsymbol{H}_{\bar{I}}\boldsymbol{d}_{\bar{I}} = [\nabla\phi(\boldsymbol{u}^t)]_{\bar{I}}$ as follows:

1. Compute the Cholesky decomposition $\boldsymbol{H}_{\bar{I}} = \boldsymbol{R}^T\boldsymbol{R}$.
2. Solve the linear system $\boldsymbol{R}^T\boldsymbol{v} = [\nabla\phi(\boldsymbol{u})]_{\bar{I}}$ to obtain $\boldsymbol{v}$.
3. Solve the linear system $\boldsymbol{R}\boldsymbol{d}_{\bar{I}} = \boldsymbol{v}$ to obtain $\boldsymbol{d}_{\bar{I}}$.

Because the reduced Hessian is also tridiagonal, its Cholesky decomposition can be computed in *linear time* to yield a bidiagonal matrix $\boldsymbol{R}$, which in turn allows to solve the subsequent linear systems also in linear time. Extremely efficient routines to perform all these tasks are available in the LAPACK libraries (Anderson et al., 1999).

The next crucial ingredient is efficient selection of the stepsize α. The original PN algorithm Bertsekas (1982) recommends Armijo-search along projection arc. However, for our problem this search is inordinately expensive. So we resort to a backtracking strategy using quadratic interpolation (Nocedal and Wright, 2000), which works admirably well. This strategy is as follows: start with an initial stepsize $\alpha_0 = 1$. If the current stepsize α_k does not provide sufficient decrease in ϕ, build a quadratic model using $\phi(\boldsymbol{u})$, $\phi(\boldsymbol{u} - \alpha_k\boldsymbol{d})$, and $\partial_{\alpha_k}\phi(\boldsymbol{u})$. Then, the stepsize α_{k+1} is set to the value that minimizes this quadratic model. In the event that at some point of the procedure the new α_{k+1} is larger than or too similar to α_k, its value is halved. In this fashion, quadratic approximations of ϕ are iterated until a good enough α is found. The goodness of a stepsize is measured using the following Armijo-like sufficient descent rule

$$\phi(\boldsymbol{u}) - \phi(P\left[\boldsymbol{u} - \alpha_k\boldsymbol{d}\right]) \geq \sigma \cdot \alpha_k \cdot (\nabla\phi(\boldsymbol{u}) \cdot \boldsymbol{d})\,,$$

where a tolerance $\sigma = 0.05$ works well practice.

Note that the gradient $\nabla\phi(\boldsymbol{u})$ might be misleading in the condition above if $\boldsymbol{u}$ has components at the boundary and $\boldsymbol{d}$ points outside this boundary (because then, due to the subsequent projection no real improvement would be obtained by stepping outside the feasible region). To address this concern, we modify the computation of the gradient $\nabla\phi(\boldsymbol{u})$, zeroing our the entries that relate to direction components pointing outside the feasible set.

The whole stepsize selection procedure is shown in Algorithm 11. The costliest operation in this procedure is the evaluation of ϕ, which, nevertheless can be done in linear time. Furthermore, in practice a few iterations more than suffice to obtain a good stepsize.

Overall, a full PN iteration as described above runs at $O(n)$ cost. Thus, by exploiting the structure of the problem, we manage to reduce the $O(n^3)$ cost per iteration of a general

Algorithm 11 Stepsize selection for Projected Newton

Initialize: $\alpha_0 = 1$, $k = 0$, $\boldsymbol{d}$, tolerance parameter σ
while $\phi(\boldsymbol{u}) - \phi(P[\boldsymbol{u} - \alpha_k \boldsymbol{d}]) < \sigma \cdot \alpha_k \cdot (\nabla\phi(\boldsymbol{u}) \cdot \boldsymbol{d})$ **do**
  Minimize quadratic model: $\alpha_{k+1} = \frac{\alpha_k^2 \partial_{\alpha_k}\phi(\boldsymbol{u})}{2(\phi(\boldsymbol{u}) - \phi(\boldsymbol{u} - \alpha_k) + \alpha_k \partial_{\alpha_k}\phi(\boldsymbol{u}))}$.
  if $\alpha_{k+1} > \alpha_k$ **or** $\alpha_{k+1} \simeq \alpha_k$, **then** $\alpha_{k+1} = \frac{1}{2}\alpha_k$.
  $k \leftarrow k + 1$
end while
return α_k

Algorithm 12 PN algorithm for TV-L1-proximity

Let $\boldsymbol{W} = \mathrm{Diag}(w_i)$; solve $\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{W}\boldsymbol{u}^* = \boldsymbol{D}\boldsymbol{y}$.
if $\|\boldsymbol{W}^{-1}\boldsymbol{u}^*\|_\infty \leq 1$, **return** $\boldsymbol{u}^*$.
$\boldsymbol{u}^0 = P[\boldsymbol{u}^*]$, $t = 0$.
while $\mathrm{gap}(\boldsymbol{u}) > \epsilon$ **do**
  Identify set of active constraints I; let $\bar{I} = \{1 \dots n\} \setminus I$.
  Construct reduced Hessian $\boldsymbol{H}_{\bar{I}}$.
  Solve $\boldsymbol{H}_{\bar{I}}\boldsymbol{d}_{\bar{I}} = [\nabla\phi(\boldsymbol{u}^t)]_{\bar{I}}$.
  Compute stepsize α using backtracking + interpolation (Alg. 11).
  Update $\boldsymbol{u}_{\bar{I}}^{t+1} = P[\boldsymbol{u}_{\bar{I}}^t - \alpha\boldsymbol{d}_{\bar{I}}]$.
  $t \leftarrow t + 1$.
end while
return $\boldsymbol{u}^t$.

PN algorithm to a linear-cost method. The pseudocode of the resulting method is shown as Algorithm 12. Note that in the special case when the weights $\boldsymbol{W} := \mathrm{Diag}(w_i)$ are so large that the unconstrained optimum coincides with the constrained one, we can obtain $\boldsymbol{u}^*$ directly via solving $\boldsymbol{D}\boldsymbol{D}^T\boldsymbol{W}\boldsymbol{u}^* = \boldsymbol{D}\boldsymbol{y}$ (which can also be done at $O(n)$ cost). The duality gap of the current solution is used as a stopping criterion, where we use a tolerance of $\epsilon = 10^{-5}$ in practice.

Appendix F. Testing Images and Videos, and Experimental Results

The images used in the experiments are displayed in what follows, along with their noisy/denoised and convoluted/deconvoluted versions for each algorithm tested. QR barcode images were generated by encoding random text using Google chart API[4]. Images *shape* and *phantom* [5] are publicly available and frequently used in image processing. *trollface* and *comic* [6] are also publicly available. *gaudi*, used in the multicore experiments, is a high resolution 3197×3361 photograph of Gaudi's Casa Batlló[7]. The rest of the images were originally created by the authors.

4. `http://code.google.com/intl/en-EN/apis/chart/`
5. Extracted from `http://en.wikipedia.org/wiki/File:Shepp_logan.png`
6. Author: Francisco Molina. `http://www.afrikislife.net/english/`
7. Extracted from `http://www.flickr.com/photos/jeffschwartz/202423023/`

For the video experiments, the *salesman*, *coastguard* and *bicycle* sequences were used, which are publicly available at BM3D (2013). As an example, frames from the first video are displayed in what follows, along with their noisy/denoised versions.

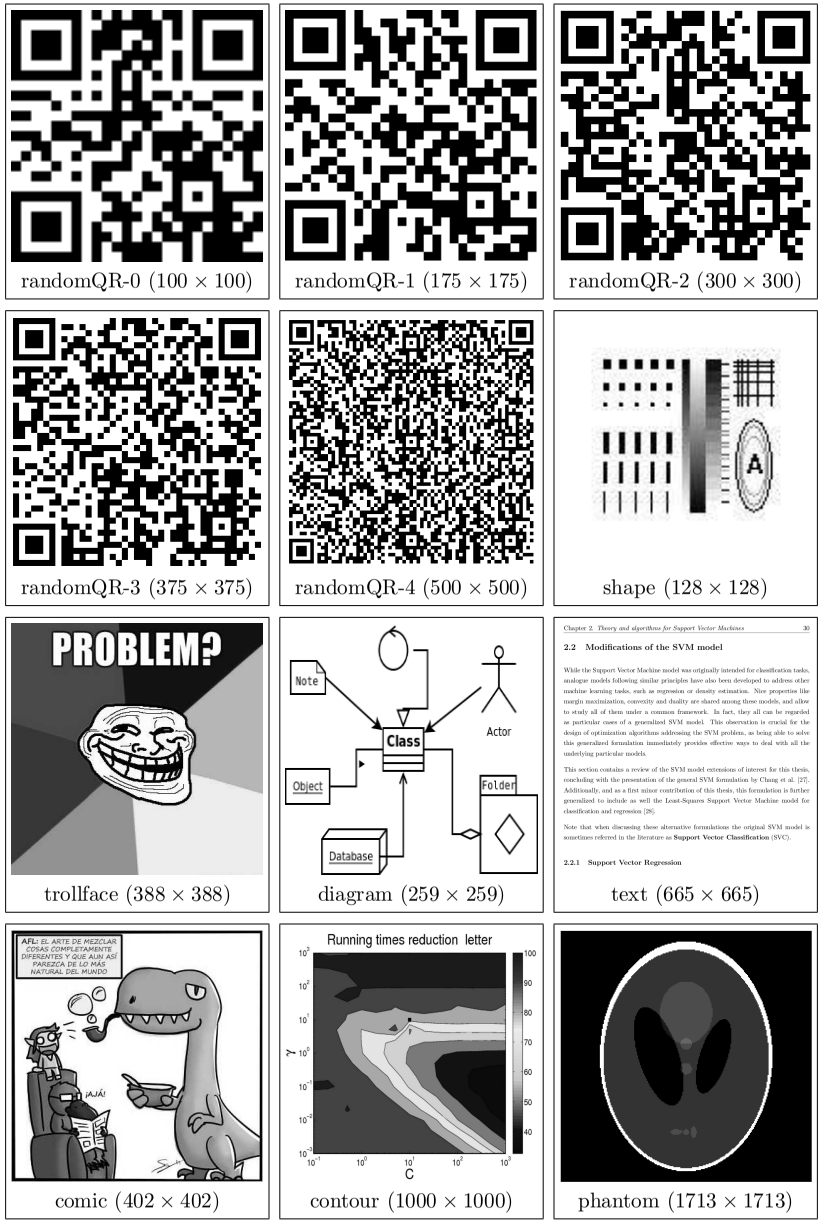


Figure 25: Test images used in the experiments together with their sizes in pixels. Images displayed have been scaled down to fit in page.

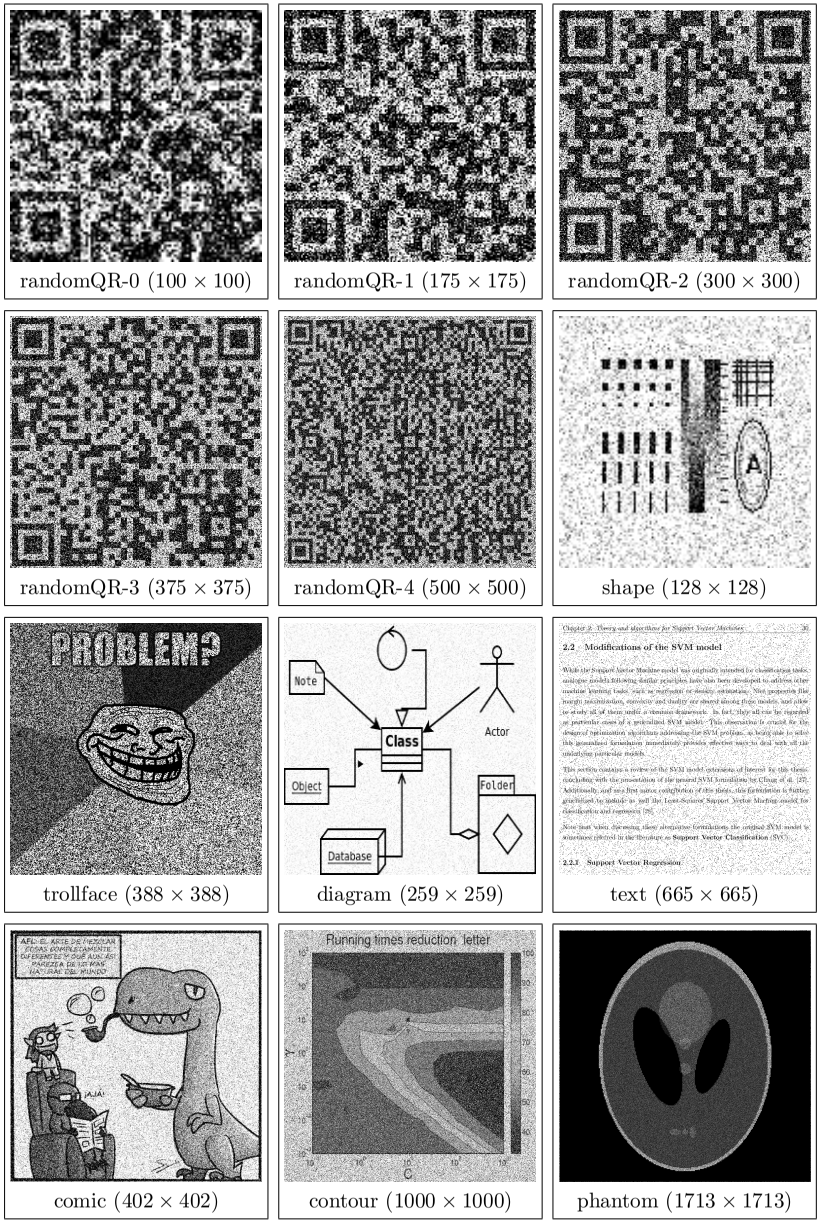


Figure 26: Noisy versions of images used in the experiments.

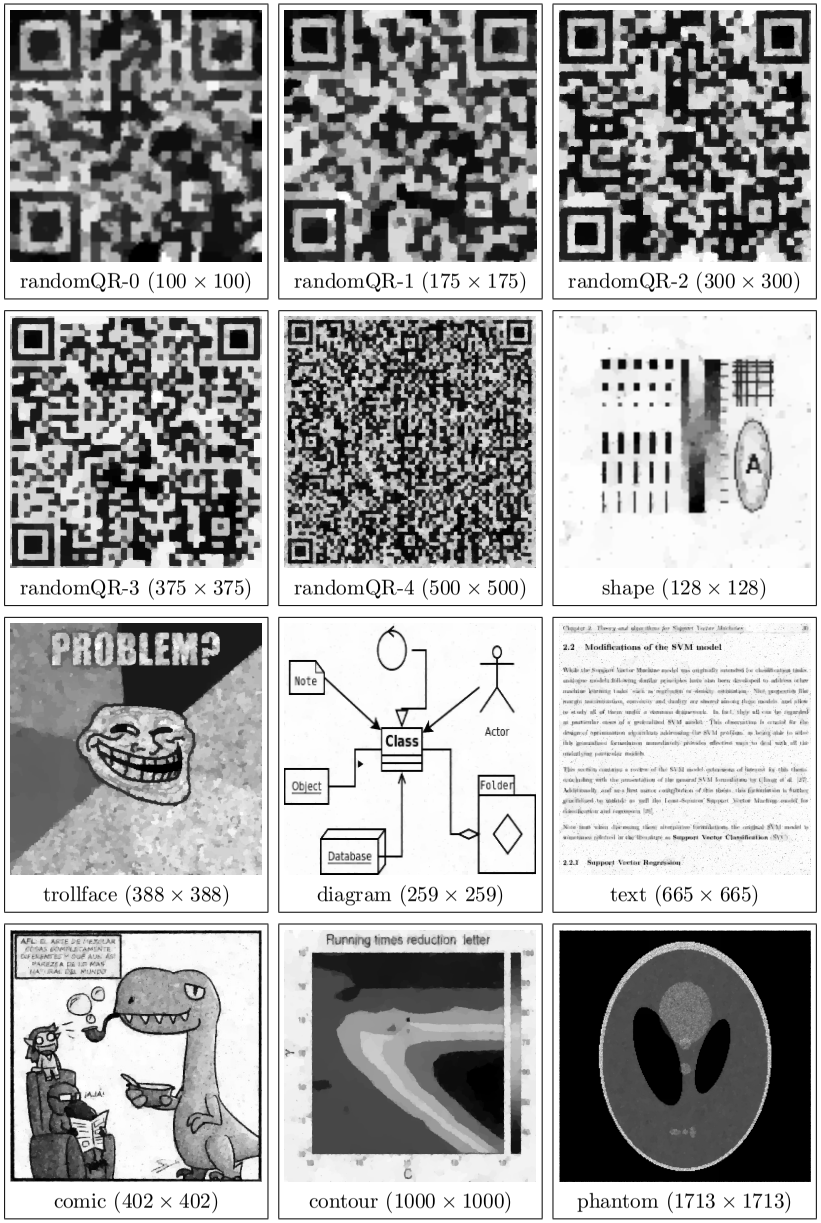


Figure 27: Denoising results for the test images.

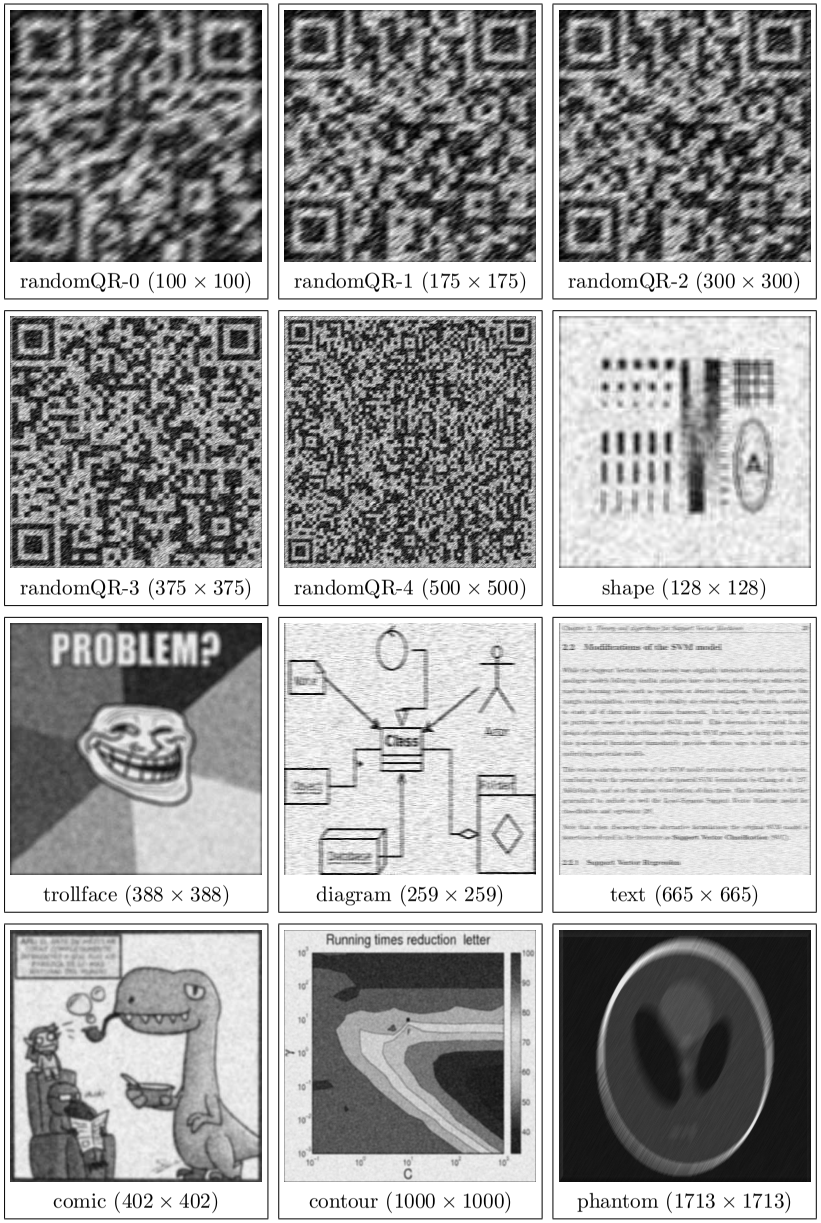
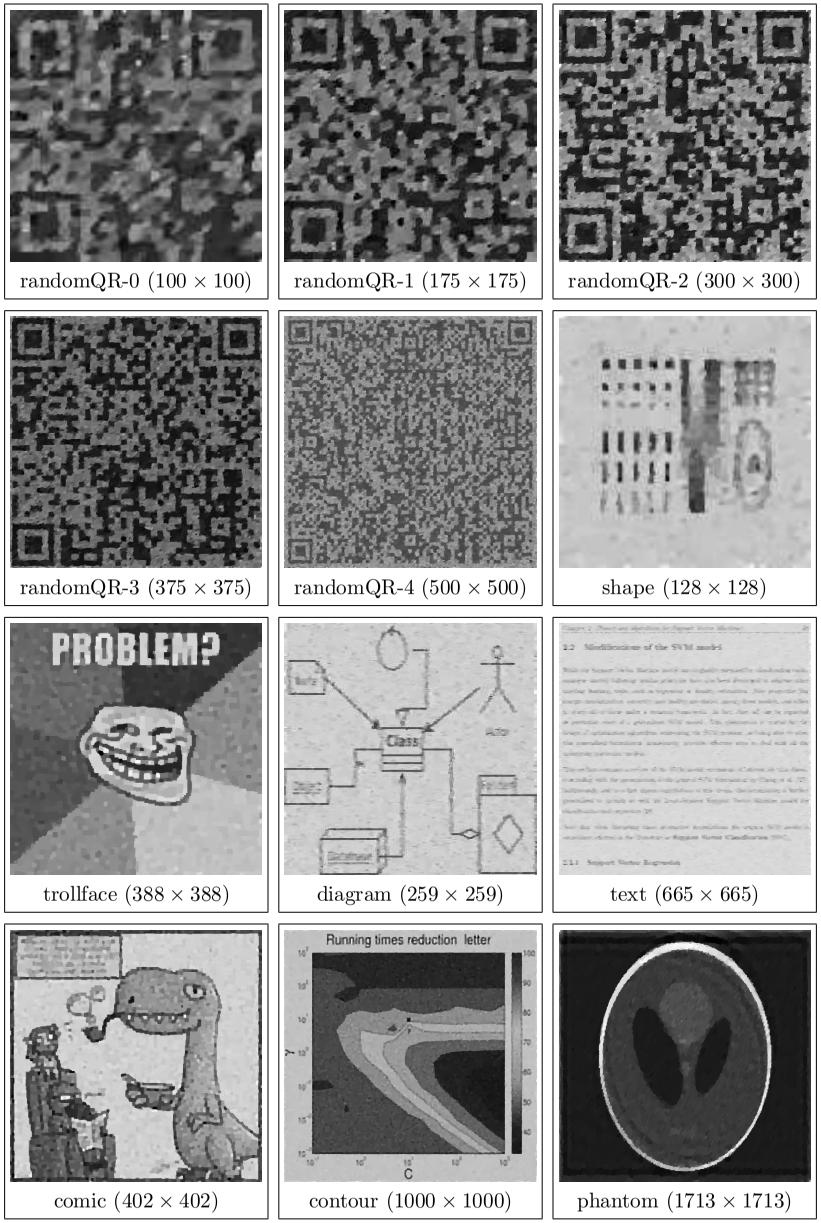


Figure 28: Noisy and convoluted versions of images used in the experiments.



randomQR-0 (100×100) | randomQR-1 (175×175) | randomQR-2 (300×300)

randomQR-3 (375×375) | randomQR-4 (500×500) | shape (128×128)

trollface (388×388) | diagram (259×259) | text (665×665)

comic (402×402) | contour (1000×1000) | phantom (1713×1713)

Figure 29: Deconvolution results for the test images.

Figure 30: A selection of frames from the *salesman* video sequence.

Figure 31: Noisy frames from the *salesman* video sequence.

Figure 32: Denoised frames from the *salesman* video sequence.

References

M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *IEEE Transactions on Image Processing*, 19(9), September 2010.

C. M. Alaız, Á. Barbero, and J. R. Dorronsoro. Group fused lasso. *Artificial Neural Networks and Machine Learning–ICANN 2013*, page 66, 2013.

E. Anderson et al. *LAPACK Users' Guide*. Society for Industrial and Applied Mathematics, Philadelphia, PA, third edition, 1999. ISBN 0-89871-447-8 (paperback).

F. Bach. Structured sparsity-inducing norms through submodular functions. In *NIPS*, 2010.

Bach, Francis Learning with Submodular Functions: A Convex Optimization Perspective *arXiv preprint arXiv:1111.6453*

F. Bach, R. Jenatton, J. Mairal, and G. Obozinski. Convex optimization with sparsity-inducing norms. In S. Sra, S. Nowozin, and S. J. Wright, editors, *Optimization for Machine Learning*. MIT Press, 2011.

Á. Barbero, J. López, and J. R. Dorronsoro. Finding Optimal Model Parameters by Discrete Grid Search. In *Advances in Soft Computing: Innovations in Hybrid Intelligent Systems 44*, pages 120–127. Springer, 2008.

Barbero, A., Sra, S. Fast Newton-type methods for total variation regularization. *In Proceedings of the 28th International Conference on Machine Learning (ICML-11) (pp. 313-320).*

Á. Barbero, J. López, and J. R. Dorronsoro. Finding Optimal Model Parameters by Deterministic and Annealed Focused Grid Search. *Neurocomputing*, 72(13-15):2824–2832, 2009. ISSN 0925-2312. doi: DOI:10.1016/j.neucom.2008.09.024.

Barlow, R. E., Bartholomew, D. J., Bremner, J. M., Brunk, H. D. Statistical inference under order restrictions: The theory and application of isotonic regression *New York: Wiley, 1972*

H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics. Springer, 2011.

Heinz H. Bauschke, Patrick L. Combettes, D. Russell Luke Finding best approximation pairs relative to two closed convex sets in Hilbert spaces *Journal of Approximation Theory 127 (2004) 178–192*

A. Beck and M. Teboulle. A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems. *SIAM Journal of Imgaging Sciences*, 2(1):183–202, 2009.

D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, 20(2), March 1982.

D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 2nd edition, September 1999.

J. M. Bioucas-Dias and M. A. T. Figueiredo. A new twist: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, December 2007.

J. M. Bioucas-Dias, M. A. T. Figueiredo, and J. P. Oliveira. Total variation-based image deconvolution: A majorization-minimization approach. In *ICASSP Proceedings*, 2006.

BM3D. Bm3d software and test sequences, 2013. URL `http://www.cs.tut.fi/~foi/GCF-BM3D/`.

R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. Technical report, Northwestern University, 1994.

E. J. Candès and T. Tao. Near-optimal signal recovery from random projections: universal encoding strategies. *IEEE Trans. Info. Theory*, 52:5406–5425, 2004.

A. Chambolle and J. Darbon. On total variation minimization and surface evolution using parametric maximum flows. *International Journal of Computer Vision*, 84(3), 2009.

A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011.

Chambolle, A., Pock, T. On the ergodic convergence rates of a first-order primal-dual algorithm *Mathematical Programming. September 2016, Volume 159, Issue 1, pp 253–287*

Chambolle, A., Pock, T. A remark on accelerated block coordinate descent for computing the proximity operators of a sum of convex functions. *SMAI Journal of Computational Mathematics*, 129-54

V. Chandrasekaran, B. Recht, P. A. Parrilo, and A. S. Willsky. The Convex Geometry of Linear Inverse Problems. *Foundations of Computational Mathematics*, 12(6), 2012.

R. Choksi, Y. van Gennip, and A. Oberman. Anisotropic Total Variation Regularized L1-Approximation and Denoising/Deblurring of 2D Bar Codes. Technical report, McGill University, July 2010.

P. L. Combettes. Iterative construction of the resolvent of a sum of maximal monotone operators. *Journal of Convex Analysis*, 16:727–748, 2009.

P. L. Combettes and J.-C. Pesquet. Proximal splitting methods in signal processing. *arXiv:0912.3522*, 2009.

L. Condat. A direct algorithm for 1d total variation denoising. Technical report, GREYC laboratory, CNRS-ENSICAEN-Univ. of Caen, 2012.

L. Condat. A generic proximal algorithm for convex optimization - application to total variation minimization. *IEEE SIGNAL PROC. LETTERS*, 21(8):985–989, 2014.

L. Condat. Fast projection onto the simplex and the ℓ_1 ball. *Mathematical Programming*, 158(1-2):575-585, 2016.

A. R. Conn, N. I. M. Gould, and P. L. Toint. *Trust-Region Methods*. SIAM, 2000.

J. Dahl, P. C. Hansen, S. H. Jensen, and T. L. Jensen. Algorithms and software for total variation image reconstruction via first-order methods. *Numer Algor*, 53:67–92, 2010.

P. L. Davies and A. Kovac. Local extremes, runs, strings and multiresolution. *The Annals of Statistics*, 29(1):1–65, 2001.

Y. Duan and X.-C. Tai. Domain decomposition methods with graph cuts algorithms for total variation minimization. *Adv Comput Math*, 36:175–199, 2012. doi: 10.1007/s10444-011-9213-4.

Esedoglu, Selim and Osher, Stanley J. Decomposition of images by the anisotropic Rudin-Osher-Fatemi model *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 77:(12): 1609–1626, 2014.

J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, Aug. 2007.

D. Goldfarb and W. Yin. Parametric maximum flow algorithms for fast total variation minimization. *SIAM Journal on Scientific Computing*, 31(5):3712–3743, 2009.

O. S. Goldstein T. The Split Bregman Method for L1 Regularized Problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

T. R. Golub et al. Molecular classification of cancer. *Science*, 286(5439):531–537, October 1999.

M. Grasmair. The equivalence of the taut string algorithm and bv-regularization. *Journal of Mathematical Imaging and Vision*, 27(1):59–66, 2007. ISSN 0924-9907. doi: 10.1007/s10851-006-9796-4. URL `http://dx.doi.org/10.1007/s10851-006-9796-4`.

Z. Harchaoui and C. Lévy-Leduc. Multiple Change-Point Estimation With a Total Variation Penalty. *Journal of the American Statistical Association*, 105(492):1480–1493, 2010.

J. Hua, W. D. Tembe, and E. R. Dougherty. Performance of feature-selection methods in the classification of high-dimension data. *Pattern Recognition*, 42:409–424, 2009.

K. Ito and K. Kunisch. An active set strategy based on the augmented lagrangian formulation for image restoration. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33 (1):1–21, 1999. URL `http://eudml.org/doc/193911`.

M. Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of the 30th International Conference on Machine Learning,*, 2013.

S. Jegelka, F. Bach, and S. Sra. Reflection methods for user-friendly submodular optimization. *Advances in Neural Information Processing Systems* 2013: 1313–1321.

Jegou, H., Douze, M., Schmid, C. Hamming Embedding and Weak geometry consistency for large scale image search *Proceedings of the 10th European conference on Computer vision, October, 2008* `http://lear.inrialpes.fr/~jegou/data.php#holidays`

N. A. Johnson. A dynamic programming algorithm for the fused Lasso and l_0-segmentation. *J. Computational and Graphical Statistics*, 2013.

D. Kim, S. Sra, and I. Dhillon. A scalable trust-region algorithm with application to mixed-norm regression. In *International Conference on Machine Learning*, 2010.

S. Kim, K. Koh, S. Boyd, and D. Gorinevsky. ℓ_1 trend filtering. *SIAM Review*, 51(2): 339–360, 2009. doi: 10.1137/070690274.

K. C. Kiwiel. Variable fixing algorithms for the continuous quadratic knapsack problem. *J. Optim. Theory Appl.*, 136:445–458, 2008.

Knuth, Donald E. The art of computer programming, volume 1: fundamental algorithms. *CA, USA: Addison Wesley Longman Publishing Co., Inc*

M. Kolar, L. Song, A. Ahmed, and E. Xing. Estimaging time-varying networks. *The Annals of Applied Statistics*, 4(1):94–123, 2010.

Kolmogorov, V., Pock, T., Rolinek, M. Total variation on a tree *SIAM J. Imaging Sci., 9(2), 605–636.*

D. Krishnan and R. Fergus. Fast image deconvolution using hyper-laplacian priors. In *Advances in Neural Information Processing Systems*, 2009.

Kumar, K.S., Barbero, A., Jegelka, S., Sra, S., and Bach, F. Convex optimization for parallel energy minimization. *arXiv preprint arXiv:1503.01563.*

S. R. Land and J. H. Friedman. Variable fusion: A new adaptive signal regression method. Technical Report 656, Department of Statistics, Carnegie Mellon University Pittsburgh,

1997.

Y. Li and F. Santosa. A computational algorithm for minimizing total variation in image restoration. *IEEE Transactions on Image Processing*, 5(6):987–995, 1996. URL `http://dblp.uni-trier.de/db/journals/tip/tip5.html#LiS96`.

C.-J. Lin and J. J. Moré. Newton's method for large bound-constrained optimization problems. *SIAM Journal on Optimization*, 9(4):1100–1127, 1999.

H. Liu and J. Zhang. Estimation Consistency of the Group Lasso and its Applications. In *Int. Conf. Mach. Learning (ICML)*, 2009.

J. Liu and J. Ye. Efficient Euclidean projections in linear time. In *ICML*, Jun. 2009.

J. Liu, S. Ji, and J. Ye. *SLEP: Sparse Learning with Efficient Projections*. Arizona State University, 2009. *http://www.public.asu.edu/ jye02/Software/SLEP*.

J. Liu, L. Yuan, and J. Ye. An efficient algorithm for a class of fused lasso problems. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2010.

J. Mairal, R. Jenatton, G. Obozinski, and F. Bach. Network Flow Algorithms for Structured Sparsity. In *NIPS*, 2010. To appear.

B. Martinet. Régularisation d'inéquations variationnelles par approximations successives. *Modélisation Mathématique et Analyse Numérique*, 4(R3):154–158, 1970.

L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *J. R. Statist. Soc.*, 70:53–71, 2008.

J. J. Moré and D. C. Sorensen. Computing a trust region step. *SIAM Journal of Scientific Computing*, 4(3), September 1983.

J. J. Moreau. Fonctions convexes duales et points proximaux dans un espace hilbertien. *C. R. Acad. Sci. Paris Sér. A Math.*, 255:2897–2899, 1962.

Y. Nesterov. Gradient methods for minimizing composite objective function. Technical Report 76, Catholic University of Louvain, CORE, 2007.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Verlag, 2000.

N. Parikh, S. Boyd, et al. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

G. Pierra. Decomposition through formalization in a product space. *Mathematical Programming*, 28(1):96–115, 1984.

C. Pontow and O. Scherzer. A derivative free approach for total variation regularization. *arXiv:0911.1293*, 2009. URL `http://arxiv.org/abs/0911.1293`.

A. Ramdas and R. J. Tibshirani. Fast and flexible admm algorithms for trend filtering. *arXiv:1406.2082*, 2014.

F. Rapaport and E. B. J.-P. Vert. Classification of arrayCGH data using fused SVM. *Bioinformatics*, 24(13):i375–i382, 2008.

A. Rinaldo. Properties and refinements of the fused lasso. *Annals of Statistics*, 37(5B): 2922–2952, 2009.

R. T. Rockafellar. Monotone operators and hte proximal point algorithm. *SIAM J. Control and Opt.*, 14(5):877–898, 1976.

S. Rogers, M. Girolami, C. Campbell, and R. Breitling. The latent process decomposition of cdna microarray data sets. *IEEE/ACM Trans. Comp. Bio. and Bioinformatics*, 2(2), April-June 2005.

L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268, 1992.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge *International Journal of Computer Vision (IJCV), Year 2015, Volume 115, Number 3, pages 211-252* `http://image-net.org/challenges/LSVRC/2010/download-public`

S. Salzo and S. Villa. Inexact and accelerated proximal point algorithms. *J. Convex Analysis*, 19(4), 2012.

M. Schmidt, N. L. Roux, and F. Bach. Convergence Rates of Inexact Proximal-Gradient Methods for Convex Optimization. In *Advances in Neural Information Processing Systems (NIPS)*, 2011.

S. Sra. Scalable nonconvex inexact proximal splitting. In *Advances in Neural Information Processing Systems*, 2012.

S. Sra, S. Nowozin, and S. Wright, editors. *Optimization for machine learning*. MIT Press, 2011.

G. Steidl, S. Didas, and J. Neumann. Relations between higher order tv regularization and support vector regression. In *Scale-Space*, pages 515–527, 2005.

G. Steidl and T. Teuber Anisotropic smoothing using double orientations. In *International Conference on Scale Space and Variational Methods in Computer Vision*, pages 477–489, 2009. Springer, Berlin, Heidelberg.

N. Stransky et al. Regional copy number-independent deregulation of transcription in cancer. *Nature Genetics*, 38(12):1386–1396, December 2006.

R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.*, 58(1): 267–288, 1996.

R. Tibshirani and P. Wang. Spatial smoothing and hot spot detection for CGH data using the fused lasso. *Biostatistics*, 9(1):18–29, 2008.

R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused lasso. *J. Royal Stat. Soc.: Series B*, 67(1):91–108, 2005.

R. J. Tibshirani. Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics*, 42(1):285–323, 02 2014. doi: 10.1214/13-AOS1189.

U. Alon et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA*, 96:6745–6750, June 1999.

J.-P. Vert and K. Bleakley. Fast detection of multiple change-points shared by many signals using group LARS. In *Advances in Neural Information Processing Systems*, 2010.

C. R. Vogel and M. E. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, 1996.

B. Wahlberg, S. Boyd, M. Annergren, and Y. Wang. An ADMM Algorithm for a Class of Total Variation Regularized Estimation Problems. In *Proceedings 16th IFAC Symposium on System Identification*, volume 16, 2012.

J. Wang and Q. Li and S. Yang and W. Fan and P. Wonka and J. Ye. A Highly Scalable Parallel Algorithm for Isotropic Total Variation Models In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 235-243, 2014.

S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo. Sparse reconstruction by separable approximation. *IEEE Trans. Sig. Proc.*, 57(7):2479–2493, 2009.

M. Wytock, S. Sra, and J. Z. Kolter. Fast Newton Methods for the Group Fused Lasso. In *Conference on Uncertainty in Artificial Intelligence*, 2014.

S. Yang, J. Wang, W. Fan, X. Zhang, P. Wonka, and J. Ye. An Efficient ADMM Algorithm for Multidimensional Anisotropic Total Variation Regularization Problems. In *ACM Knowledge Discovery and Data Mining (KDD)*, Chicago, Illinois, USA, August 2013.

Y. Yu. On decomposing the proximal map. In *Advances in Neural Information Processing Systems*, 2013.

M. Yuan and Y. Lin. Model Selection and Estimation in Regression with Grouped Variables. *J. R. Statist. Soc. B*, 68(1):49–67, 2006.

M. Zhu and T. Chan. An efficient primal-dual hybrid gradient algorithm for total variation image restoration. Technical report, UCLA CAM, 2008.

Journal of Machine Learning Research 19 (2018) 1-59 Submitted 9/15; Revised 10/18; Published 10/18

On Semiparametric Exponential Family Graphical Models

Zhuoran Yang ZY6@PRINCETON.EDU
Department of Operations Research and Financial Engineering
Princeton University
Princeton, NJ 08544, USA

Yang Ning YN265@CORNELL.EDU
Department of Statistical Science
Cornell University
Ithaca, NY 14853, USA

Han Liu HANLIU@NORTHWESTERN.EDU
Department of Electrical Engineering and Computer Science
Northwestern University
Evanston, IL 60208, USA

Editor: David Blei

Abstract

We propose a new class of semiparametric exponential family graphical models for the analysis of high dimensional mixed data. Different from the existing mixed graphical models, we allow the nodewise conditional distributions to be semiparametric generalized linear models with unspecified base measure functions. Thus, one advantage of our method is that it is unnecessary to specify the type of each node and the method is more convenient to apply in practice. Under the proposed model, we consider both problems of parameter estimation and hypothesis testing in high dimensions. In particular, we propose a symmetric pairwise score test for the presence of a single edge in the graph. Compared to the existing methods for hypothesis tests, our approach takes into account of the symmetry of the parameters, such that the inferential results are invariant with respect to the different parametrizations of the same edge. Thorough numerical simulations and a real data example are provided to back up our theoretical results.

Keywords: Graphical Models, Exponential Family, High Dimensional Inference

1. Introduction

Given a d-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_d)^T$, inferring the conditional independence among $\boldsymbol{X}$ and quantifying its uncertainty are important tasks in statistics. We propose a unified framework for modeling, estimation, and uncertainty assessment for a new type of graphical model, named as semiparametric exponential family graphical model. Let $G = (V, E)$ be an undirected graph with node set $V = \{1, 2, \ldots, d\}$ and edge set $E \subseteq \{(j, k)\colon 1 \le j < k \le d\}$. The semiparametric exponential family graphical model specifies the joint distribution of $\boldsymbol{X}$ such that for each $j \in V$, the conditional distribution of X_j given $\boldsymbol{X}_{\setminus j} := (X_1, \ldots, X_{j-1}, X_{j+1}, \ldots, X_d)^T$ is of the form

$$p(x_j \mid \boldsymbol{x}_{\setminus j}) = \exp\big[\eta_j(\boldsymbol{x}_{\setminus j}) \cdot x_j + f_j(x_j) - b_j(\eta_j, f_j)\big], \tag{1}$$

where $\boldsymbol{x}_{\setminus j} = (x_1, \ldots, x_{j-1}, x_{j+1}, \ldots, x_d)$, $\eta_j(\boldsymbol{x}_{\setminus j}) = \alpha_j + \sum_{k \neq j} \beta_{jk} x_k$ is the canonical parameter, $f_j(\cdot)$ is an unknown base measure function, and $b_j(\cdot, \cdot)$ is the log-partition function. Besides, we assume $\beta_{jk} = \beta_{kj}$ for all $j \neq k$. By definition, the unknown parameter contains $\{(\alpha_j, \beta_{jk}, f_j)\colon 1 \le j < k \le d\}$. To make the model identifiable, we set $\alpha_j = 0$ and absorb the term $\alpha_j x_j$ into $f_j(x_j)$. By the Hammersley-Clifford theorem (Besag, 1974), we have $\beta_{jk} \neq 0$ if and only if X_j and X_k are conditionally independent given $\{X_\ell\colon \ell \neq j, k\}$. Therefore, we set $(j, k) \in E$ if and only if $\beta_{jk} \neq 0$. The graph G thus characterizes the conditional independence relationship among the high dimensional distribution of $\boldsymbol{X}$. The key feature of the proposed model is that (1) it is a general semiparametric model and (2) it can be used to handle mixed data, which means that $\boldsymbol{X}$ may contain both continuous and discrete random variables. Unlike the existing mixed graphical models, we allow the nodewise conditional distributions to be semiparametric generalized linear models with unspecified base measure functions. Thus, our method does not need to specify the type of each node and is more convenient to apply in practice. In addition to the proposed new model, our paper has the following two novel contributions.

First, for the purpose of estimating β_{jk}, we extend the multistage relaxation algorithm (Zhang, 2010) and conduct a localized analysis for a more sophisticated loss function obtained by a statistical chromatography method (Liang and Qin, 2000; Diao et al., 2012; Chan, 2012; Ning et al., 2017b). The gradient and Hessian matrix of the loss function are nonlinear U-statistics with unbounded kernel functions. This makes our technical analysis more challenging than that in Zhang (2010). Under the assumption that the sparse eigenvalue condition holds locally, we prove the same optimal statistical rates for parameter estimation as in high dimensional linear models.

Second, we propose a symmetric pairwise score test for the null hypothesis $H_0\colon \beta_{jk} = 0$. This is equivalent to testing whether X_j and X_k are conditionally independent given $\{X_\ell\colon \ell \neq j, k\}$. Compared with Ning et al. (2017b), the novelty of our method is that we consider a more sophisticated cross type inference which incorporates the symmetry of the parameter, i.e., $\beta_{jk} = \beta_{kj}$. By considering this unique structure of the graphical model, our proposed method achieves the invariance property of the inferential results. That means the same p-values are obtained for testing $\beta_{jk} = 0$ and $\beta_{kj} = 0$. In contrast, the asymmetric method in Ning et al. (2017b) may lead to different conclusions for testing these two equivalent null hypotheses.

1.1. Related Works

There is a huge literature on estimating undirected graphical models (Lauritzen, 1996; Edwards, 2000; Whittaker, 2009). For modeling continuous data, the most commonly used methods are Gaussian graphical models (Yuan and Lin, 2007; Banerjee et al., 2008; Friedman et al., 2008; Ravikumar et al., 2011; Rothman et al., 2008; Lam and Fan, 2009; Shen et al., 2012; Yuan, 2010; Cai et al., 2011; Sun and Zhang, 2013; Guo et al., 2011; Danaher et al., 2014; Mohan et al., 2014; Meinshausen and Bühlmann, 2006; Peng et al., 2009; Friedman et al., 2010). To relax the Gaussian assumption, Liu et al. (2009); Xue et al. (2012b); Liu et al. (2012); Ning and Liu (2013) propose the Gaussian copula model and Voorman et al. (2014) study the joint additive models for graph estimation. For modeling binary data, the Ising graphical model is considered by Lee et al. (2006); Höfling and Tibshirani (2009);

Ravikumar et al. (2010); Xue et al. (2012a); Cheng et al. (2014). In addition to binary data, Allen and Liu (2012) and Yang et al. (2013b) consider the Poisson data and Guo et al. (2015) consider the ordinal data. Moreover, Yang et al. (2013a) propose exponential family graphical models, and Tan et al. (2014) propose a general framework for graphical models with hubs.

Recently, modeling the mixed data attracts increasing interests (Lee and Hastie, 2015; Fellinghauer et al., 2013; Cheng et al., 2017; Chen et al., 2015; Fan et al., 2017; Yang et al., 2014). Compared with Lee and Hastie (2015); Cheng et al. (2017); Chen et al. (2015); Yang et al. (2014), our model has the following two main advantages. First, it is a semiparametric model, which does not need to specify the parametric conditional distribution for each node. Therefore, it provides a more flexible modeling framework than the existing ones. Second, under our proposed model, the estimation and inference methods are easier to implement. Unlike these existing methods, we propose a unified estimation and inference procedure, which does not need to distinguish whether the node satisfies the Gaussian distribution or the Bernoulli distribution. In addition, our estimation and inference methods are more efficient than the nonparametric approach in Fellinghauer et al. (2013). Finally, our method is more convenient for modeling the count data than the latent Gaussian copula approach in Fan et al. (2017).

Though significant progress has been made towards developing new graph estimation procedures, the research on uncertainty assessment of the estimated graph lags behind. In low dimensions, Drton et al. (2007); Drton and Perlman (2008) establish confidence subgraph of Gaussian graphical models. In high dimensions, Ren et al. (2015); Janková and van de Geer (2015); Gu et al. (2015) study the confidence interval for a single edge under Gaussian (copula) graphical models and Liu et al. (2013) study the false discovery rate control. However, all these methods rely on the Gaussian or sub-Gaussian assumption and cannot be easily applied to the discrete data and more generally the mixed data in high dimensions.

1.2. Notation

We adopt the following notation throughout this paper. For any vector $\mathbf{v} = (v_1, \ldots, v_d)^T \in \mathbb{R}^d$, we define its support as $\mathrm{supp}(\mathbf{v}) = \{t\colon v_t \neq 0\}$. We define its ℓ_0-norm, ℓ_p-norm, and ℓ_∞-norm as $\|\mathbf{v}\|_0 = |\mathrm{supp}(\mathbf{v})|$, $\|\mathbf{v}\|_p = (\sum_{j\in[d]} |v_j|^p)^{1/p}$ and $\|\mathbf{v}\|_\infty = \max_{j\in[d]} |v_j|$, respectively, where $p > 1$. Let $\mathbf{v}^{\otimes 2} = \mathbf{v}\mathbf{v}^T$ be the Kronecker product of a vector $\mathbf{v}$ and itself. We write $\mathbf{v} \circ \mathbf{u} = (v_1u_1, \ldots, v_du_d)^T$ as the Hadamard product of two vectors $\mathbf{u}, \mathbf{v} \in \mathbb{R}^d$. In addition, we use $|\mathbf{v}| = (|v_1|, \ldots, |v_d|)^T$ to denote the elementwise absolute value of vector $\mathbf{v}$ and define $\|\mathbf{v}\|_{\min} = \min_{j\in[d]} |v_j|$. For any matrix $\mathbf{A} = [a_{jk}] \in \mathbb{R}^{d_1\times d_2}$, let $\mathbf{A}_{S_1S_2} = [a_{jk}]_{j\in S_1, k\in S_2}$ be the submatrix of $\mathbf{A}$ with indices in $S_1 \times S_2$; let $\mathbf{A}_{j\setminus j} = [a_{jk}]_{k\neq j}$. Besides, let $\|\mathbf{A}\|_2$, $\|\mathbf{A}\|_1$, $\|\mathbf{A}\|_\infty$, $\|\mathbf{A}\|_{\ell_p}$ be the spectral norm, elementwise ℓ_1-norm, elementwise ℓ_∞-norm, and operator ℓ_p-norm of $\mathbf{A}$, respectively. Furthermore, for two matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, we write $\mathbf{A}_1 \preceq \mathbf{A}_2$ if $\mathbf{A}_2 - \mathbf{A}_1$ is positive semidefinite and write $\mathbf{A}_1 \leq \mathbf{A}_2$ if every entry of $\mathbf{A}_2 - \mathbf{A}_1$ is nonnegative. For a function $f(\boldsymbol{x})\colon \mathbb{R}^d \to \mathbb{R}$, we write $\nabla f(\boldsymbol{x})$, $\nabla_S f(\boldsymbol{x})$, $\nabla^2 f(\boldsymbol{x})$ and $\partial f(\boldsymbol{x})$ as the gradient of $f(\boldsymbol{x})$, the gradient of $f(\boldsymbol{x})$ with respect to $\boldsymbol{x}_S$, the Hessian of $f(\boldsymbol{x})$, and the subgradient of $f(\boldsymbol{x})$, respectively. Moreover, we write $\{1, 2, \ldots, d\}$ as $[d]$. For a sequence of random vectors $\{\boldsymbol{Y}_i\}_{i\geq 1}$ and a random vector $\boldsymbol{Y}$, we write $\boldsymbol{Y}_i \rightsquigarrow \boldsymbol{Y}$ if $\{\boldsymbol{Y}_i\}_{i\geq 1}$ converges to $\boldsymbol{Y}$ in distribution.

Finally, for functions $f(n)$ and $g(n)$, we write $f(n) \lesssim g(n)$ to denote that $f(n) \leq cg(n)$ for a universal constant $c \in (0, +\infty)$ and we write $f(n) \asymp g(n)$ when $f(n) \lesssim g(n)$ and $g(n) \lesssim f(n)$ hold simultaneously.

1.3. Paper Organization

The rest of this paper is organized as follows. In §2 we introduce the semiparametric exponential family graphical models. In §3 we present our methods for graph estimation and uncertainty assessment. In §4 we lay out the assumptions and main theoretical results. We study the finite-sample performance of our method on both simulated and real-world datasets in §5 and conclude the paper in §6 with some discussion.

2. Semiparametric Exponential Family Graphical Models

The semiparametric exponential family graphical models are defined by specifying the conditional distribution of each variable X_j given the rest of the variables $\{X_k\colon k \neq j\}$.

Definition 1 (Semiparametric exponential family graphical model) *A d-dimensional random vector $\boldsymbol{X} = (X_1, \ldots, X_d)^T \in \mathbb{R}^d$ follows a semiparametric exponential graphical model with graph $G = (V, E)$ if for any node $j \in V$, the conditional density of X_j given $\boldsymbol{X}_{\setminus j}$ satisfies*

$$p(x_j \,|\, \boldsymbol{x}_{\setminus j}) = \exp\big[x_j(\boldsymbol{\beta}_j^T \boldsymbol{x}_{\setminus j}) + f_j(x_j) - b_j(\boldsymbol{\beta}_j, f_j)\big], \tag{2}$$

where $f_j(\cdot)$ is an unknown base measure function and $b_j(\cdot, \cdot)$ is a known log-partition function. In particular, $(j, k) \in E$ if and only if $\beta_{jk} \neq 0$.

This model is semiparametric since we treat both $\boldsymbol{\beta}_j = (\beta_{j1}, \ldots, \beta_{jj-1}, \beta_{jj+1}, \ldots, \beta_{jd})^T \in \mathbb{R}^{d-1}$ and the univariate function $f_j(\cdot)$ as parameters, where $\boldsymbol{\beta}_j$ and $f_j(\cdot)$ are the parametric and nonparametric components, respectively. Because the model in Definition 1 is only specified by the conditional distributions of each variable, it is important to understand the conditions under which a valid joint distribution of $\boldsymbol{X}$ exists. This problem has been addressed by Chen et al. (2015). As shown in their Proposition 1, one sufficient condition for the existence of joint distribution of $\boldsymbol{X}$ is that, (i) $\beta_{jk} = \beta_{kj}$ for $1 \leq j, k \leq d$ and (ii) $g(\boldsymbol{x}) := \exp\big[\sum_{j<k} \beta_{jk} x_j x_k + \sum_{j=1}^d f_j(x_j)\big]$ is integrable.

Hereafter, we assume that the above two conditions hold. Thus, there exists a joint probability distribution for the model defined in (2), whose density has the form of

$$p(\boldsymbol{x}) = \exp\bigg[\sum_{k<\ell} \beta_{k\ell} x_k x_\ell + \sum_{j=1}^d f_j(x_j) - A\big(\{\boldsymbol{\beta}_i, f_i\}_{i\in[d]}\big)\bigg], \tag{3}$$

where $\beta_{k\ell} \neq 0$ if and only if $(k, \ell) \in E$. Here $A(\cdot)$ is the log-partition function given by

$$A\big(\{\boldsymbol{\beta}_i, f_i\}_{i\in[d]}\big) := \log\bigg\{\int_{\mathbb{R}^d} \exp\bigg[\sum_{k<\ell} \beta_{k\ell} x_k x_\ell + \sum_{j=1}^d f_j(x_j)\bigg] \nu(\mathrm{d}\boldsymbol{x})\bigg\}, \tag{4}$$

where $\nu(\cdot)$ is a product measure satisfying $\nu(\mathrm{d}\boldsymbol{x}) = \prod_{j\in[d]} \nu_j(\mathrm{d}x_j)$, and each ν_j is either a Lebesgue or a counting measure on the domain of X_j, depending whether X_j is discrete or

continuous. Since $\beta_{k\ell} = \beta_{\ell k}$ for all pairs of nodes (k, ℓ), in the sequel, we will use $\beta_{k\ell}$ and $\beta_{\ell k}$ interchangeably for notational simplicity.

Furthermore, we remark that, without the knowledge of $\{f_j\}_{j\in[d]}$, estimating parameters $\{\boldsymbol{\beta}_j\}_{j\in[d]}$ is insufficient to learn the distribution of $\boldsymbol{X}$. In this paper, we focus on the statistical inference of the underlying conditional independence graph specified by $\{\boldsymbol{\beta}_j\}_{j\in[d]}$. In the next section, by adopting a loss function for $\{\boldsymbol{\beta}_j\}_{j\in[d]}$ that is free of the base measures, we obtain estimators of these parameters, which are used to construct an estimator of the underlying graph. Moreover, by further considering the hypothesis testing problem for each β_{jk}, we are able to assess the uncertainty of the estimated graph.

2.1. Examples

We provide some widely used parametric examples in the class of semiparametric exponential family graphical models.

Gaussian Graphical Models: The Gaussian graphical models assume that $\boldsymbol{X} \in \mathbb{R}^d$ follows a multivariate Gaussian distribution $N(\mathbf{0}, \boldsymbol{\Theta}^{-1})$, where $\boldsymbol{\Theta} \in \mathbb{R}^{d\times d}$ is the precision matrix satisfying $\boldsymbol{\Theta}_{jj} = 1$ for $j \in [d]$. The conditional distribution of X_j given $\boldsymbol{X}_{\backslash j}$ satisfies

$$X_j \,|\, \boldsymbol{X}_{\backslash j} = \boldsymbol{\alpha}_j^T \boldsymbol{X}_{\backslash j} + \epsilon_j \quad \text{with} \quad \epsilon_j \sim N(0,1),$$

where $\boldsymbol{\alpha}_j = \boldsymbol{\Theta}_{\backslash j,j}$. The conditional density is given by

$$p(x_j \,|\, \boldsymbol{x}_{\backslash j}) = \sqrt{1/(2\pi)}\exp\big[-x_j(\boldsymbol{\Theta}_{\backslash j,j}^T \boldsymbol{x}_{\backslash j}) - 1/2 \cdot x_j^2 - 1/2 \cdot (\boldsymbol{\Theta}_{\backslash j,j}^T \boldsymbol{x}_{\backslash j})^2\big].$$

Compared with (2), we obtain $\boldsymbol{\beta}_j = -\boldsymbol{\Theta}_{\backslash j,j}$, $f_j(x) = -x^2/2$ and $b_j(\boldsymbol{\beta}_j, f_j) = (\boldsymbol{\beta}_j^T \boldsymbol{x}_{\backslash j})^2/2 + \log(2\pi)/2$.

Ising Models: In an Ising model with no external field, $\boldsymbol{X}$ takes value in $\{0,1\}^d$ and the joint probability mass function $p(\boldsymbol{x}) \propto \exp(\sum_{j<k} \theta_{jk} x_j x_k)$. Let $\boldsymbol{\theta}_j = (\theta_{j1}, \ldots, \theta_{j,j-1}, \theta_{j,j+1}, \ldots, \theta_{jd})^T$. The conditional distribution of X_j given $\boldsymbol{X}_{\backslash j}$ is of the form

$$p(x_j \,|\, \boldsymbol{x}_{\backslash j}) = \frac{\exp\big(\sum_{k<\ell} \theta_{k\ell} x_k x_\ell\big)}{\sum_{x_j\in\{0,1\}} \exp\big(\sum_{k<\ell} \theta_{k\ell} x_k x_\ell\big)} = \exp\Big\{x_j\big(\boldsymbol{\theta}_j^T \boldsymbol{x}_{\backslash j}\big) - \log\big[1 + \exp(\boldsymbol{\theta}_j^T \boldsymbol{x}_{\backslash j})\big]\Big\}.$$

Therefore, in this case we have $\boldsymbol{\beta}_j = \boldsymbol{\theta}_j$, $f_j(x) = 0$ and $b_j(\boldsymbol{\beta}_j, f_j) = \log[1 + \exp(\boldsymbol{\beta}_j^T \boldsymbol{x}_{\backslash j})]$.

Exponential Graphical Models: For exponential graphical models, $\boldsymbol{X}$ takes values in $[0, +\infty)^d$ and the joint probability density satisfies $p(\boldsymbol{x}) \propto \exp(-\sum_{j=1}^d \phi_j x_j - \sum_{k<\ell} \theta_{k\ell} x_k x_\ell)$. In order to ensure that this probability distribution is normalizable, we require that $\phi_j > 0, \theta_{jk} \geq 0$ for all $j, k \in [d]$. Then we obtain the following conditional probability density of X_j given $\boldsymbol{X}_{\backslash j}$:

$$\begin{aligned} p(x_j \,|\, \boldsymbol{x}_{\backslash j}) &= \exp\Big(-\sum_{k=1}^d \phi_k x_k - \sum_{k<\ell} \theta_{k\ell} x_k x_\ell\Big) \Big/ \int_{x_j\geq 0} \exp\Big(-\sum_{k=1}^d \phi_k x_k - \sum_{k<\ell} \theta_{k\ell} x_k x_\ell\Big) \mathrm{d}x_j \\ &= \exp\big[-x_j\big(\phi_j + \boldsymbol{\theta}_j^T \boldsymbol{x}_{\backslash j}\big) - \log\big(\phi_j + \boldsymbol{\theta}_j^T \boldsymbol{x}_{\backslash j}\big)\big]. \end{aligned}$$

Thus, we have $\boldsymbol{\beta}_j = -\boldsymbol{\theta}_j$, $f_j(x) = -\phi_j x$ and $b_j(\boldsymbol{\beta}_j, f_j) = \log(\boldsymbol{\beta}_j^T \boldsymbol{x}_{\backslash j} + \phi_j)$.

Poisson Graphical Models: In a Poisson graphical model, every node X_j is a discrete random variable taking values in $\mathbb{N} = \{0, 1, 2, \ldots\}$. The joint probability mass function is given by

$$p(\boldsymbol{x}) \propto \exp\Big[\sum_{j=1}^d \phi_j x_j - \sum_{j=1}^d \log(x_j!) + \sum_{k<\ell} \theta_{k\ell} x_k x_\ell\Big].$$

Similar to the exponential graphical models, we also need to impose some restrictions on the parameters so that the probability mass function is normalizable. Here we require that $\theta_{jk} \leq 0$ for all $j, k \in [d]$. By direct computation, the conditional probability mass function of X_j given $\boldsymbol{X}_{\backslash j}$ is given by

$$p(x_j \,|\, \boldsymbol{x}_{\backslash j}) = \exp\big[x_j\big(\boldsymbol{\theta}_j^T \boldsymbol{x}_{\backslash j}\big) + \phi_j x_j - \log(x_j!) - b_j(\boldsymbol{\theta}_j, f_j)\big],$$

where we have $\boldsymbol{\beta}_j = \boldsymbol{\theta}_j$, $f_j(x) = \phi_j x - \log(x!)$ and $b_j(\boldsymbol{\beta}_j, f_j) = \log\big\{\sum_{y=0}^\infty \exp\big[y(\boldsymbol{\beta}_j^T \boldsymbol{x}_{\backslash j}) + f_j(y)\big]\big\}$.

3. Graph Estimation and Uncertainty Assessment

In this section, we lay out the procedures for graph estimation and uncertainty assessment. Throughout our analysis, we use $\{\boldsymbol{\beta}_i^*, f_i^*\}_{i\in[d]}$ to denote the true parameters, and $\mathbb{E}(\cdot)$ to denote the expectation with respect to the joint density in (3) with the true parameters. We first introduce a pseudo-likelihood loss function for the parametric components $\{\boldsymbol{\beta}_j\}_{j=1}^d$ that is invariant to the nuisance parameters $\{f_j\}_{j\in[d]}$. Based on such a loss function, we present an Adaptive Multi-stage Convex Relaxation algorithm to estimate each $\boldsymbol{\beta}_j^*$ by minimizing the loss function regularized by a nonconvex penalty function. We then proceed to introduce the inferential procedure for accessing the uncertainty of a given edge in the graph.

3.1. A Nuisance-Free Loss Function

For graph estimation, we treat $\boldsymbol{\beta}_j$ as the parameter of interest and the base measures $f_j(\cdot)$ as nuisance parameter. Let $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ be n i.i.d. copies of $\boldsymbol{X}$. Due to the presence of $f_j(\cdot)$, finding the conditional maximum likelihood estimator of $\boldsymbol{\beta}_j$ is intractable. To solve this problem, we exploit a pseudo-likelihood loss function proposed in Ning et al. (2017b) that is invariant to the nuisance parameters $\{f_j\}_{j\in[d]}$. This pseudo-likelihood loss is based on pairwise local order statistics, which have been previously studied in Liang and Qin (2000); Diao et al. (2012); Chan (2012) for semiparametric regression models. More details are presented as follows.

Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n$ be n data points that are realizations of $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$. For any $1 \leq i < i' \leq n$, let

$$\mathcal{A}_{ii'}^j := \big\{(X_{ij}, X_{i'j}) = (x_{ij}, x_{i'j}), \boldsymbol{X}_{i\backslash j} = \boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j} = \boldsymbol{x}_{i'\backslash j}\big\}$$

be the event that we observe $\boldsymbol{X}_{i\backslash j} = \boldsymbol{x}_{i\backslash j}$ and $\boldsymbol{X}_{i'\backslash j} = \boldsymbol{x}_{i'\backslash j}$ and the order statistics of X_{ij} and $X_{i'j}$ (but not the relative ranks of X_{ij} and $X_{i'j}$). More specifically, we denote $\max\{X_{ij}, X_{i'j}\}$ and $\min\{X_{ij}, X_{i'j}\}$ by O_1 and O_2, and let o_1 and o_2 be the observed values of O_1 and O_2. Then $\mathcal{A}_{ii'}^j$ can be equivalently written as $\{O_1 = o_1, O_2 = o_2, \boldsymbol{X}_{i\backslash j} = \boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j} = \boldsymbol{x}_{i'\backslash j}\}$.

Let $R \in \{(1,2),(2,1)\}$ be the relative rank of X_{ij} and $X_{i'j}$, and r be the observed value. Then, by definition, we have

$$\begin{aligned}&\mathbb{P}\big(X_{ij}=x_{ij}, X_{i'j}=x_{i'j} \,\big|\, \boldsymbol{X}_{i\backslash j}=\boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j}=\boldsymbol{x}_{i'\backslash j}\big)\\ &\quad=\mathbb{P}\big(O_1=o_1, O_2=o_2 \,\big|\, \boldsymbol{X}_{i\backslash j}=\boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j}=\boldsymbol{x}_{i'\backslash j}\big)\cdot\mathbb{P}\big(R=r \,\big|\, \mathcal{A}_{ii'}^j\big).\end{aligned}$$

Furthermore, we have

$$\begin{aligned}\mathbb{P}\big(R=r \,\big|\, \mathcal{A}_{ii'}^j\big) &= \left[1+\frac{\mathbb{P}(X_{ij}=x_{i'j}, X_{i'j}=x_{ij} \,|\, \mathcal{A}_{ii'}^j)}{\mathbb{P}(X_{ij}=x_{ij}, X_{i'j}=x_{i'j} \,|\, \mathcal{A}_{ii'}^j)}\right]^{-1}\\ &= \left[1+\frac{\mathbb{P}(X_{ij}=x_{i'j}, X_{i'j}=x_{ij} \,|\, \boldsymbol{X}_{i\backslash j}=\boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j}=\boldsymbol{x}_{i'\backslash j})}{\mathbb{P}(X_{ij}=x_{ij}, X_{i'j}=x_{i'j} \,|\, \boldsymbol{X}_{i\backslash j}=\boldsymbol{x}_{i\backslash j}, \boldsymbol{X}_{i'\backslash j}=\boldsymbol{x}_{i'\backslash j})}\right]^{-1} = \big[1+R_{ii'}^j(\boldsymbol{\beta}_j)\big]^{-1},\end{aligned} \tag{5}$$

where $R_{ii'}^j(\boldsymbol{\beta}_j) := \exp[-(x_{ij}-x_{i'j})\boldsymbol{\beta}_j{}^T(\boldsymbol{x}_{i\backslash j}-\boldsymbol{x}_{i'\backslash j})]$. Based on the conditional likelihood in (5), we construct the following pseudo-likelihood loss function for $\boldsymbol{\beta}_j$:

$$L_j(\boldsymbol{\beta}_j) := \frac{2}{n(n-1)}\sum_{1\le i<i'\le n}\log\big[1+R_{ii'}^j(\boldsymbol{\beta}_j)\big]. \tag{6}$$

Obviously, $L_j(\cdot)$ only involves $\boldsymbol{\beta}_j$. Since its form resembles the logistic loss, to find a minimizer of this loss function, we could readily apply any logistic regression solver.

3.2. Adaptive Multi-stage Convex Relaxation Algorithm

Now we are ready to present the algorithm for parameter estimation. For high dimensional sparse estimation, to promote sparsity, we minimize the sum of the loss functions $L_j(\boldsymbol{\beta}_j)$ and some penalty function. Two of the most prevalent methods are the LASSO (ℓ_1-penalization) (Tibshirani, 1996) and the folded concave penalization (Fan et al., 2014). Although the ℓ_1-penalization enjoys good computational properties as a convex optimization problem, it is known to incur significant estimation bias for parameters with large absolute values (Zhang and Huang, 2008). In contrast, nonconvex penalties such as smoothly clipped absolute deviation (SCAD) penalty, minimax concave penalty (MCP) and capped-ℓ_1 penalty can eliminate such bias and attain improved rates of convergence. Therefore, we consider the nonconvex optimization problem

$$\widehat{\boldsymbol{\beta}}_j = \underset{\mathbb{R}^{d-1}}{\operatorname{argmin}}\ \Big\{L_j(\boldsymbol{\beta}_j)+\sum_{k\neq j}p_\lambda(|\beta_{jk}|)\Big\}, \tag{7}$$

where $\lambda>0$ is a regularization parameter and $p_\lambda(\cdot):[0,+\infty)\to[0,+\infty)$ is a penalty function satisfying the following three conditions:

(C.1) The penalty function $p_\lambda(u)$ is continuously nondecreasing and concave with $p_\lambda(0)=0$.

(C.2) The right-hand derivative at $u=0$ satisfies $p'_\lambda(0)=p'_\lambda(0+)=\lambda$.

(C.3) There exist constants $c_1\in[0,1]$ and $c_2\in(0,+\infty)$ such that $p'_\lambda(u+)\ge c_1\lambda$ for $u\in[0,c_2\lambda]$.

Note that we only require the penalty function to be right-differentiable. In what follows, we denote by $p'_\lambda(u)$ the right-hand derivative. By (C.1), $p'_\lambda(u)$ is nonincreasing and nonnegative in $[0,\infty)$. It is easy to verify that SCAD, MCP and capped-ℓ_1 penalty all satisfy (C.1)–(C.3).

Due to the penalty term, the optimization problem in (7) is nonconvex and may have multiple local solutions. To overcome such difficulty, we exploit the local linear approximation algorithm (Zou and Li, 2008; Fan et al., 2014) or equivalently, the multi-stage convex relaxation (Zhang, 2010; Zhang et al., 2013; Fan et al., 2018) to attain an estimator of $\boldsymbol{\beta}_j^*$. Compared with previous works that mainly focus on sparse linear regression, our loss function $L_j(\boldsymbol{\beta}_j)$ is a U-statistics based logistic loss, which requires nontrivial extensions of the existing theoretical analysis.

We present the proposed adaptive multi-stage convex relaxation method in Algorithm 1. Our algorithm solves a sequence of convex optimization problems corresponding to finer and finer convex relaxations of the original nonconvex optimization problem. More specifically, for each $j=1,\ldots,d$, in the first iteration, step 4 of Algorithm 1 is equivalent to a ℓ_1-regularized optimization problem and we obtain the first-step solution $\widehat{\boldsymbol{\beta}}_j^{(1)}$. Then, in each subsequent iteration, we solve an adaptive ℓ_1-regularized optimization problem where the weights of the penalty depend on the solution of the previous step. For example, in the ℓ-th iteration, the regularization parameter $\lambda_{jk}^{(\ell-1)}$ in (8) is updated using the $(\ell-1)$-th step estimator $\widehat{\boldsymbol{\beta}}_j^{(\ell-1)}$. Note that $p'_\lambda\big(|\beta_{jk}^{(\ell)}|\big)$ is the right-hand derivative of $p_\lambda(u)$ evaluated at $u=|\beta_{jk}^{(\ell)}|$.

Since the optimization problem in step 4 is convex, our method is computationally efficient. Besides, note that (8) with $\ell=1$ corresponds to the ℓ_1-regularized problem. Hence, our approach can be viewed as a refinement of LASSO. As we will show in §4.1, the estimator $\widehat{\boldsymbol{\beta}}_j$ of $\boldsymbol{\beta}_j^*$ constructed by Algorithm 1 attains the optimal statistical rates of convergence for parameter estimation.

Algorithm 1 Adaptive Multi-stage Convex Relaxation algorithm for parameter estimation

1: Initialize $\lambda_{jk}^{(0)}=\lambda$ for $1\le j,k\le d$.
2: **for** j= 1,2,. . . ,d **do**
3: **for** $\ell=1,2,\ldots,$ until convergence **do**
4: Solve the convex optimization problem

$$\widehat{\boldsymbol{\beta}}_j^{(\ell)} = \underset{\mathbb{R}^{d-1}}{\operatorname{argmin}}\Big\{L_j(\boldsymbol{\beta}_j)+\sum_{k\neq j}\lambda_{jk}^{(\ell-1)}|\beta_{jk}|\Big\}. \tag{8}$$

5: Update $\lambda_{jk}^{(\ell)}$ by $\lambda_{jk}^{(\ell)}=p'_\lambda(|\widehat{\beta}_{jk}^{(\ell)}|)$ for $1\le k\le d, k\neq j$.
6: **end for**
7: **Output** $\widehat{\boldsymbol{\beta}}_j=\widehat{\boldsymbol{\beta}}_j^{(\ell)}$, where ℓ is the number of iterations until convergence is attained.
8: **end for**

3.3. Graph Inference: Composite Pairwise Score Test

For any given $1 \le j < k \le d$, we are interested in testing if $(j,k) \in E$, i.e., we consider the hypothesis testing problem $H_0 : \beta^*_{jk} = 0$ versus $H_1 : \beta^*_{jk} \neq 0$. To simplify the notation, we write $\boldsymbol{\beta}_{j\backslash k} = (\beta_{j1}, \ldots, \beta_{jj-1}, \beta_{jj+1}, \ldots, \beta_{jk-1}, \beta_{jk+1}, \ldots, \beta_{jd})^T \in \mathbb{R}^{d-2}$ and denote the parameters associated with node j and node k by $\boldsymbol{\beta}_{j\vee k} := \big(\beta_{jk}; \boldsymbol{\beta}^T_{j\backslash k}, \boldsymbol{\beta}^T_{k\backslash j}\big)^T \in \mathbb{R}^{2d-3}$. In addition, let $\mathbf{H}^j := \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}^*_j)\big]$ be the expected Hessian of $L_j(\boldsymbol{\beta}_j)$ evaluated at $\boldsymbol{\beta}^*_j$. We define two submatrices $\mathbf{H}^j_{jk,j\backslash k}$ and $\mathbf{H}^j_{j\backslash k,j\backslash k}$ of $\mathbf{H}^j$ as

$$\mathbf{H}^j_{jk,j\backslash k} := \left[\mathbb{E}\frac{\partial^2 L_j(\boldsymbol{\beta}^*_j)}{\partial\beta_{jk}\partial\beta_{jv}}\right]_{v\neq k} \in \mathbb{R}^{d-2} \text{ and } \mathbf{H}^j_{j\backslash k,j\backslash k} := \left[\mathbb{E}\frac{\partial^2 L_j(\boldsymbol{\beta}^*_j)}{\partial\beta_{ju}\partial\beta_{jv}}\right]_{u\neq k, v\neq k} \in \mathbb{R}^{(d-2)\times(d-2)},$$

and we define $\mathbf{H}^k_{jk,k\backslash j}$ and $\mathbf{H}^k_{k\backslash j,k\backslash j}$ similarly. Furthermore, we define

$$\mathbf{w}^*_{j,k} = \mathbf{H}^j_{jk,j\backslash k}\big[\mathbf{H}^j_{j\backslash k,j\backslash k}\big]^{-1} \quad \text{and} \quad \mathbf{w}^*_{k,j} = \mathbf{H}^k_{jk,k\backslash j}\big[\mathbf{H}^k_{k\backslash j,k\backslash j}\big]^{-1}. \tag{9}$$

Following the general approach in Ning et al. (2017a); Neykov et al. (2018), the composite pairwise score function for parameter β_{jk} is defined as

$$S_{jk}(\boldsymbol{\beta}_{j\vee k}) = \nabla_{jk} L_j(\boldsymbol{\beta}_j) + \nabla_{jk} L_k(\boldsymbol{\beta}_k) - \mathbf{w}^{*T}_{j,k} \nabla_{j\backslash k} L_j(\boldsymbol{\beta}_j) - \mathbf{w}^{*T}_{k,j} \nabla_{k\backslash j} L_k(\boldsymbol{\beta}_k). \tag{10}$$

where we write $\nabla_{jk} L_j(\boldsymbol{\beta}_j) = \partial L_j(\boldsymbol{\beta}_j)/\partial\beta_{jk}$ and $\nabla_{j\backslash k} L_j(\boldsymbol{\beta}_j) = \partial L_j(\boldsymbol{\beta}_j)/\partial\boldsymbol{\beta}_{j\backslash k}$. Here, the last two terms in (10) are constructed to reduce the effect of nuisance parameters $\boldsymbol{\beta}_{j\backslash k}$ and $\boldsymbol{\beta}_{k\backslash j}$ on assessing the uncertainty of β^*_{jk}, which is the parameter of interest. A key feature of $S_{jk}(\boldsymbol{\beta}_{j\vee k})$ is that the symmetry of β_{jk} and β_{kj} (i.e., $\beta_{jk} = \beta_{kj}$) is taken into account, which is distinct from the existing works such as Ren et al. (2015); Janková and van de Geer (2015); Liu et al. (2013) for Gaussian graphical models and Ning et al. (2017b) in the regression setup.

Note that both $\mathbf{w}^*_{j,k}$ and $\mathbf{w}^*_{k,j}$ are computed from $\mathbf{H}$, which is unknown. We estimate them using the Dantzig-type estimators (Candés et al., 2007). Specifically, we define the empirical versions of $\mathbf{H}^j_{jk,j\backslash k}$ and $\mathbf{H}^j_{j\backslash k,j\backslash k}$ as

$$\nabla^2_{jk,j\backslash k} L_j(\boldsymbol{\beta}_j) = \left[\frac{\partial^2 L_j(\boldsymbol{\beta}_j)}{\partial\beta_{jk}\partial\beta_{jv}}\right]_{v\neq k} \quad \text{and} \quad \nabla^2_{j\backslash k,j\backslash k} L_j(\boldsymbol{\beta}_j) = \left[\frac{\partial^2 L_j(\boldsymbol{\beta}_j)}{\partial\beta_{ju}\partial\beta_{jv}}\right]_{u\neq k, v\neq k}.$$

We also define $\nabla^2_{jk,k\backslash j} L_k(\boldsymbol{\beta}_k)$ and $\nabla^2_{k\backslash j,k\backslash j} L_k(\boldsymbol{\beta}_k)$ similarly. Then we estimate $\mathbf{w}^*_{j,k}$ by solving

$$\widehat{\mathbf{w}}_{j,k} = \operatorname{argmin} \|\mathbf{w}\|_1 \quad \text{such that} \quad \big\|\nabla^2_{jk,j\backslash k} L_j(0, \widehat{\boldsymbol{\beta}}_{j\backslash k}) - \mathbf{w}^T \nabla^2_{j\backslash k,j\backslash k} L_j(0, \widehat{\boldsymbol{\beta}}_{j\backslash k})\big\|_\infty \le \lambda_D, \tag{11}$$

where $\widehat{\boldsymbol{\beta}}_j$ is the estimator of $\boldsymbol{\beta}^*_j$ obtained from Algorithm 1 and λ_D is a regularization parameter. An estimator $\widehat{\mathbf{w}}_{k,j}$ of $\mathbf{w}^*_{k,j}$ can be similarly obtained. Based on $\widehat{\mathbf{w}}_{j,k}$ and $\widehat{\mathbf{w}}_{k,j}$, we construct the composite pairwise score statistic for β^*_{jk} by

$$\widehat{S}_{jk} = \nabla_{jk} L_j(0, \widehat{\boldsymbol{\beta}}_{j\backslash k}) + \nabla_{jk} L_k\big(0, \widehat{\boldsymbol{\beta}}_{k\backslash j}\big) - \widehat{\mathbf{w}}^T_{j,k} \nabla_{j\backslash k} L_j\big(0, \widehat{\boldsymbol{\beta}}_{j\backslash k}\big) - \widehat{\mathbf{w}}^T_{k,j} \nabla_{k\backslash j} L_k\big(0, \widehat{\boldsymbol{\beta}}_{k\backslash j}\big). \tag{12}$$

Comparing (10) and (12), we see that $\widehat{S}_{jk}$ is obtained by replacing $\boldsymbol{\beta}_j$ and $\boldsymbol{\beta}_k$ in (10) by $(0, \widehat{\boldsymbol{\beta}}_{j\backslash k})$ and $(0, \widehat{\boldsymbol{\beta}}_{k\backslash j})$ respectively and replacing $\mathbf{w}^*_{j,k}$ and $\mathbf{w}^*_{k,j}$ in (10) by $\widehat{\mathbf{w}}_{j,k}$ and $\widehat{\mathbf{w}}_{k,j}$.

To obtain a valid hypothesis test, we need to establish the limiting distribution of $\widehat{S}_{jk}$ under the null hypothesis. Note that $\widehat{S}_{jk}$ is a linear combination of entries of $\nabla L_j(\boldsymbol{\beta}_j)$ and $\nabla L_k(\boldsymbol{\beta}_k)$, both of which are U-statistics. In the next section, we prove the asymptotic normality of $\widehat{S}_{jk}$. More specifically, under the null hypothesis, we have $\sqrt{n}\widehat{S}_{jk}/2 \rightsquigarrow N(0, \sigma^2_{jk})$, where the limiting variance can be estimated consistently by $\widehat{\sigma}^2_{jk}$ (More details will be explained in the following section). With a significance level $\alpha \in (0,1)$, the test function $\psi_{jk}(\alpha)$ is defined as

$$\psi_{jk}(\alpha) = \begin{cases} 1 & \text{if } \big|\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk})\big| > \Phi^{-1}(1-\alpha/2) \\ 0 & \text{if } \big|\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk})\big| \le \Phi^{-1}(1-\alpha/2) \end{cases}, \tag{13}$$

where $\Phi(t)$ is the cumulative distribution function of a standard normal random variable.

In sum, the composite pairwise score test for the null hypothesis $H_0 \colon \beta^*_{jk} = 0$ consists of the following four steps: (i) Calculate $\widehat{\boldsymbol{\beta}}_j$ and $\widehat{\boldsymbol{\beta}}_k$ from Algorithm 1; (ii) Obtain $\widehat{\mathbf{w}}_{j,k}$ and $\widehat{\mathbf{w}}_{k,j}$ by solving two Dantzig-type problems defined in (11); (iii) Compute the limiting variance $\widehat{\sigma}^2_{jk}$; (iv) Evaluate the test function (13).

4. Theoretical Properties

In this section, we present our theoretical results. We first prove that the proposed procedure attains the optimal rate of convergence for parameter estimation. Then, we provide theory for the composite pairwise score test.

4.1. Theoretical Results for Parameter Estimation

We first establish the rates of convergence of the adaptive multi-stage convex relaxation estimator. We begin by listing several required assumptions. The first is about moment conditions of $\{X_j\}$ and the local smoothness of the log-partition function $A(\cdot)$ defined in (4). This assumption also appears in Yang et al. (2013a) and Chen et al. (2015) as a pivotal technical condition for theoretical analysis.

Assumption 2 *For all $j \in [d]$, we assume that the first two moments of X_j are bounded. That is, there exist two constants κ_m and κ_v such that $|\mathbb{E}(X_j)| \le \kappa_m$ and $\mathbb{E}(X_j^2) \le \kappa_v$. Denote the true parameters by $\{\boldsymbol{\beta}^*_j, f^*_j\}_{j\in[d]}$ and define d univariate functions $\bar{A}_j(\cdot) \colon \mathbb{R} \to \mathbb{R}$ as*

$$\bar{A}_j(u) := \log\left\{\int_{\mathbb{R}^d} \exp\left[u x_j + \sum_{k<\ell} \beta^*_{k\ell} x_k x_\ell + \sum_{i=1}^d f^*_i(x_i)\right] \mathrm{d}\nu(\boldsymbol{x})\right\}, \quad j \in [d].$$

We assume that there exists a constant κ_h such that $\max_{u:\,|u|\le 1} \bar{A}''_j(u) \le \kappa_h$ for all $j \in [d]$.

Unlike the Ising graphical models, $\{X_j\}_{j\in[d]}$ are not bounded in general for semiparametric exponential family graphical models. Instead, we impose mild conditions as in

Assumption 2 to obtain a loose control of the tail behaviors of the distribution of $\boldsymbol{X}$. As shown in Yang et al. (2013a), Assumption 2 implies that for all $j \in [d]$,

$$\max\big\{\log \mathbb{E}[\exp(X_j)], \log \mathbb{E}[\exp(-X_j)]\big\} \leq \kappa_m + \kappa_h/2.$$

Markov inequality implies for any $x > 0$,

$$\mathbb{P}\big(|X_j| \geq x\big) \leq 2\exp(\kappa_m + \kappa_h/2) \cdot \exp(-x). \tag{14}$$

Thus, by setting $x = C\log d$ in (14) with constant C sufficiently large, we have $\|\boldsymbol{X}\|_\infty \leq C\log d$ with high probability. In addition to Assumption 2, we also impose conditions to control the curvature of function $L_j(\cdot)$.

Definition 3 (Sparse eigenvalue condition) *For any $j, s \in [d]$, we define the s-sparse eigenvalues of $\mathbb{E}[\nabla^2 L_j(\boldsymbol{\beta}_j^*)]$ as*

$$\rho_{j+}^*(s) \coloneqq \sup_{\mathbf{v}\in\mathbb{R}^{d-1}} \big\{\mathbf{v}^T \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\mathbf{v} \colon \|\mathbf{v}\|_0 \leq s, \|\mathbf{v}\|_2 = 1\big\};$$
$$\rho_{j-}^*(s) \coloneqq \inf_{\mathbf{v}\in\mathbb{R}^{d-1}} \big\{\mathbf{v}^T \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\mathbf{v} \colon \|\mathbf{v}\|_0 \leq s, \|\mathbf{v}\|_2 = 1\big\}.$$

Assumption 4 *Let $s^* = \max_{j\in[d]} \|\boldsymbol{\beta}_j^*\|_0$. We assume that for any $j \in [d]$, there exist an integer $k^* \geq 2s^*$ satisfying $\lim_{n\to\infty} k^*(\log^9 d/n)^{1/2} = 0$ and a positive number ρ_* such that the sparse eigenvalues of $\mathbb{E}[\nabla^2 L_j(\boldsymbol{\beta}_j^*)]$ satisfy*

$$0 < \rho_* \leq \rho_{j-}^*(2s^* + 2k^*) < \rho_{j+}^*(k^*) < +\infty \quad \text{and}$$
$$\rho_{j+}^*(k^*)\big/\rho_{j-}^*(2s^* + 2k^*) \leq 1 + 0.2k^*/s^* \quad \text{for any } j \in [d].$$

The condition $\rho_{j+}^*(k^*)\big/\rho_{j-}^*(2s^*+2k^*) \leq 1+0.2k^*/s^*$ requires the eigenvalue ratio $\rho_{j+}^*(k)/\rho_{j-}^*(2k+2s^*)$ to grow sub-linearly in k. Assumption 4 is commonly referred to as sparse eigenvalue condition, which is standard for sparse estimation problems and has been studied by Bickel et al. (2009); Raskutti et al. (2010); Zhang (2010); Negahban et al. (2012); Xiao and Zhang (2013); Loh and Wainwright (2015) and Wang et al. (2014). Our assumption is similar to that in Zhang (2010) and is weaker than the restricted isometry property (RIP) proposed in Candés and Tao (2005). We claim that this assumption is true in general and will be verified for Gaussian graphical models in the appendix.

Now we are ready to present the main theorem of this section. Recall that the penalty function $p_\lambda(u)$ satisfies conditions (C.1)–(C.3) in §3.2. We use $p'_\lambda(u)$ to denote its right-hand derivative. For convenience, we will set $p'_\lambda(u) = 1$ when $u < 0$.

Theorem 5 (ℓ_2- and ℓ_1-rates of convergence) *For all $j \in [d]$, we define the support of $\boldsymbol{\beta}_j^*$ as $S_j \coloneqq \big\{(j,k) \colon \beta_{jk}^* \neq 0, k \in [d]\big\}$ and let $s^* = \max_{j\in[d]} \|\boldsymbol{\beta}_j^*\|_0$. Let $\rho_* > 0$ be defined in Assumption 4. Under Assumptions 2 and 4, there exists an absolute constant $K > 0$ such that $\|\nabla L_j(\boldsymbol{\beta}_j^*)\|_\infty \leq K\sqrt{\log d/n}, \forall j \in [d]$ with probability at least $1 - (2d)^{-1}$. Moreover, the penalty function $p_\lambda(\cdot)$ in (7) satisfies (C.1)–(C.3) listed in §3.2 with $c_1 = 0.91$ and $c_2 \geq 24/\rho_*$ for condition (C.3). We set the regulization parameter $\lambda = C\sqrt{\log d/n}$ with $C \geq 25K$. We denote constants $\varrho = c_2(c_2\rho_* - 11)^{-1}$, $A_1 = 22\varrho$, $A_2 = 2.2c_2$, $B_1 = 32\varrho$,*

$B_2 = 3.2c_2$, $\gamma = 11c_2^{-1}\rho_*^{-1} < 1$, and define $\Upsilon_j \coloneqq [\sum_{(j,k)\in S_j} p'_\lambda(|\beta_{jk}^*| - c_2\lambda)^2]^{1/2}$. Then, with probability at least $1 - d^{-1}$, we have the following statistical rates of convergence:

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq A_1\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + A_2\sqrt{s^*}\lambda\gamma^\ell \quad \text{and} \tag{15}$$
$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq B_1\sqrt{s^*}\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + B_2 s^*\lambda\gamma^\ell, \forall j \in [d]. \tag{16}$$

By Theorem 5, the statistical rates are dominated by the second term if $p'_\lambda(|\beta_{jk}^*| - c_2\lambda)$ is not negligible. If the signal strength is large enough such that $p'_\lambda(\beta - c_2\lambda) = 0$ where $\beta = \min_{(j,k)\in S_j} |\beta_{jk}^*|$, after sufficient number of iterations, the statistical rates will be of the order

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 = \mathcal{O}_{\mathbb{P}}\big(\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2\big) \quad \text{and} \quad \big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 = \mathcal{O}_{\mathbb{P}}\big(\sqrt{s^*}\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2\big).$$

However, if the signals are uniformly small such that $p'_\lambda\big(|\beta_{jk}^*| - c_2\lambda\big) > 0$ for all $(j,k) \in S_j$, the rates of convergence will be of the order

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 = \mathcal{O}_{\mathbb{P}}\big(\sqrt{s^*}\lambda\big) \quad \text{and} \quad \big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 = \mathcal{O}_{\mathbb{P}}\big(s^*\lambda\big),$$

which are identical to the ℓ_2- and ℓ_1-rates of the LASSO estimator, respectively (Ning et al., 2017b). Thus $c_2\lambda$ can be viewed as the threshold of signal strength. Therefore, after sufficient numbers of iterations, the final estimator $\widehat{\boldsymbol{\beta}}_j$ obtained by Algorithm 1 attains the following more refined rates of convergence:

$$\big\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*\big\|_2 = \mathcal{O}_{\mathbb{P}}\Big(\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\Big) \text{ and } \big\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*\big\|_1 = \mathcal{O}_{\mathbb{P}}\Big(\sqrt{s^*}\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big]\Big).$$

These statistical rates of convergence are optimal in the sense that they cannot be improved in terms of the order.

Finally, we comment that the sparsity level s^* in (15) and (16) can be replaced by the sparsity level of each $\boldsymbol{\beta}_j^*$. Let $s_j^* = \|\boldsymbol{\beta}_j^*\|_0$ be the sparsity level of $\boldsymbol{\beta}_j^*$ and λ_j be the regularization parameter for optimization problem (7) such that $\lambda_j \asymp \|\nabla L_j(\boldsymbol{\beta}_j^*)\|_\infty$. The statistical rates of convergence for each $\widehat{\boldsymbol{\beta}}_j^{(\ell)}$ can be improved to

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 = \mathcal{O}_{\mathbb{P}}\big(\sqrt{s_j^*}\lambda_j\big) \quad \text{and} \quad \big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 = \mathcal{O}_{\mathbb{P}}\big(s_j^*\lambda_j\big).$$

We use the uniform sparsity level $s^* = \max_{j\in[d]} s_j^*$ and the same regularization parameter λ for simplicity, but the proof can be easily adapted to individual s_j^* and λ_j for each $j \in [d]$.

4.2. Theoretical Results for Composite Pairwise Score Test

In the composite pairwise score test for the null hypothesis $H_0 : \beta_{jk}^* = 0$, we construct the test statistic by combining the loss functions $L_j(\cdot)$ and $L_k(\cdot)$ together because β_{jk} appears in both $L_j(\boldsymbol{\beta}_j)$ and $L_k(\boldsymbol{\beta}_k)$ (recall that we use β_{jk} and β_{kj} interchangeably). In the sequel, we present the theoretical results that guarantee the validity of the proposed inferential method.

Recall that we define the pairwise score function $S_{jk}(\boldsymbol{\beta}_{j\vee k})$ and the pairwise score statistic $\widehat{S}_{jk}$ in (10) and (12) respectively. According to a fixed pair of nodes (j,k), entries in

$\boldsymbol{\beta}_j$ and $\boldsymbol{\beta}_k$ can be categorized into three types: (i) β_{jk}, (ii) $\boldsymbol{\beta}_{j\backslash k} = (\beta_{j\ell}; \ell \neq k)^T$, and (iii) $\boldsymbol{\beta}_{k\backslash j} = (\beta_{k\ell}; \ell \neq j)^T$. Recall that we write $\boldsymbol{\beta}_{j\vee k} = (\beta_{jk}, \boldsymbol{\beta}_{j\backslash k}^T, \boldsymbol{\beta}_{k\backslash j}^T)^T$ for notational simplicity. Moreover, letting $L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big) := L_j(\boldsymbol{\beta}_j) + L_k(\boldsymbol{\beta}_j)$, the entries of $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ are given by

$$\nabla_{jk} L_{jk}(\boldsymbol{\beta}_{j\vee k}) = \nabla_{jk} L_j(\boldsymbol{\beta}_j) + \nabla_{kj} L_k(\boldsymbol{\beta}_k); \quad \nabla_{j\backslash k} L_{jk}(\boldsymbol{\beta}_{j\vee k}) = \nabla_{j\backslash k} L_j(\boldsymbol{\beta}_j), \quad \text{and}$$
$$\nabla_{k\backslash j} L_{jk}(\boldsymbol{\beta}_{j\vee k}) = \nabla_{k\backslash j} L_k(\boldsymbol{\beta}_k).$$

Let $\widehat{\boldsymbol{\beta}}_j$ and $\widehat{\boldsymbol{\beta}}_k$ be the estimators of $\boldsymbol{\beta}_j^*$ and $\boldsymbol{\beta}_k^*$ obtained from Algorithm 1. Note that we can write the pairwise score function $S_{jk}(\cdot)$ and the test statistic $\widehat{S}_{jk}$ as

$$S_{jk}(\boldsymbol{\beta}_{j\vee k}) = \nabla_{jk} L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big) - \mathbf{w}_{j,k}^{*T} \nabla_{j\backslash k} L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big) - \mathbf{w}_{k,j}^{*\ T} \nabla_{k\backslash j} L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big) \quad \text{and} \tag{17}$$
$$\widehat{S}_{jk} = \nabla_{jk} L_{jk}\big(\widehat{\boldsymbol{\beta}}'_{j\vee k}\big) - \widehat{\mathbf{w}}_{j,k}^T \nabla_{j\backslash k} L_{jk}\big(\widehat{\boldsymbol{\beta}}'_{j\vee k}\big) - \widehat{\mathbf{w}}_{k,j}^T \nabla_{k\backslash j} L_{jk}\big(\widehat{\boldsymbol{\beta}}'_{j\vee k}\big), \tag{18}$$

where we write $\widehat{\boldsymbol{\beta}}'_{j\vee k} := \big(0, \widehat{\boldsymbol{\beta}}_{j\backslash k}^T, \widehat{\boldsymbol{\beta}}_{k\backslash j}^T\big)^T$, $\mathbf{w}_{j,k}^*$ and $\mathbf{w}_{k,j}^*$ are defined in (9), $\widehat{\mathbf{w}}_{j,k}$ is obtained from the Dantzig-type problem in (11), and $\widehat{\mathbf{w}}_{k,j}$ can be obtained similarly. To derive the asymptotic distribution of $\widehat{S}_{jk}$ under the null hypothesis, we first show that $\sqrt{n}\big[\widehat{S}_{jk} - S_{jk}(\boldsymbol{\beta}_{j\vee k}^*)\big] = o_{\mathbb{P}}(1)$. Then the problem is reduced to finding the limiting distribution of $S_{jk}(\boldsymbol{\beta}_{j\vee k}^*)$ under H_0. Thanks to its structure of being a U-statistics, we can characterize the limiting distribution of $S_{jk}(\boldsymbol{\beta}_{j\vee k}^*)$ using the method of Hájek projection (Van der Vaart, 2000), which approximates a U-statistic with a sum of independent random variables.

To begin with, we denote the kernel functions of $\nabla L_j(\boldsymbol{\beta}_j)$, $\nabla L_k(\boldsymbol{\beta}_k)$ and $\nabla L_{jk}(\boldsymbol{\beta}_{j\vee k})$ as $\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j)$, $\mathbf{h}_{ii'}^k(\boldsymbol{\beta}_k)$ and $\mathbf{h}_{ii'}^{jk}(\boldsymbol{\beta}_{j\vee k})$ respectively. It can be shown that $\mathbb{E}[\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*)] = \mathbb{E}[\mathbf{h}_{ii'}^k(\boldsymbol{\beta}_k^*)] = 0$; hence $\mathbf{h}_{ii'}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)$ is also centered. We define

$$\mathbf{g}_{jk}(\boldsymbol{X}_i) := n/2 \cdot \mathbb{E}\big[\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)\big|\boldsymbol{X}_i\big] = \mathbb{E}\big[\mathbf{h}_{ii'}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)\big|\boldsymbol{X}_i\big] \quad \text{and} \tag{19}$$
$$\mathbf{U}_{jk} := \frac{2}{n} \sum_{i=1}^n \mathbf{g}_{jk}(\boldsymbol{X}_i) = \sum_{i=1}^n \mathbb{E}\big[\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)\big|\boldsymbol{X}_i\big]. \tag{20}$$

Thus $2/n \cdot \mathbf{g}_{jk}\big(\boldsymbol{X}_i\big)$ is the projection of $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)$ onto the σ-filed generated by $\boldsymbol{X}_i$ and $\mathbf{U}_{jk}$ is the Hájek projection of $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)$. Under mild conditions, $\mathbf{U}_{jk}$ in (20) is a good approximation of $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)$, which enables us to characterize the limiting distribution of $S_{jk}(\boldsymbol{\beta}_{j\vee k}^*)$. We present the following assumption that guarantees the non-degeneracy of $\mathbf{g}_{jk}\big(\boldsymbol{X}_i\big)$.

Assumption 6 *Under Assumption 2, for $\mathbf{g}_{jk}(\boldsymbol{X}_i)$ defined in (19), we denote the covariance matrix of $\mathbf{g}_{jk}(\boldsymbol{X}_i)$ as $\boldsymbol{\Sigma}^{jk} := \mathbb{E}[\mathbf{g}_{jk}(\boldsymbol{X}_i)\mathbf{g}_{jk}(\boldsymbol{X}_i)^T]$. We assume that there exists a constant $c_\Sigma > 0$ such that $\lambda_{\min}(\boldsymbol{\Sigma}^{jk}) \geq c_\Sigma$ for all $1 \leq j < k \leq d$.*

Assumption 6 requires the minimum eigenvalue of $\boldsymbol{\Sigma}^{jk}$ to be bounded away from 0, which implies $\mathrm{Var}(\mathbf{v}^T\mathbf{U}_{jk}) \geq 4c_\Sigma$ for all $\mathbf{v} \in \mathbb{R}^{2d-3}$ with $\|\mathbf{v}\|_2 = 1$. Thus, this assumption guarantees the asymptotic variance of $\sqrt{n}S_{jk}(\boldsymbol{\beta}_{j\vee k}^*)$ is bounded away from 0. We also present the following assumption that specifies the scaling of the Dantzig selector problem in (11).

Assumption 7 *We assume that $\mathbf{H}^j$ is invertible for all $j \in [d]$. In addition, we assume that there exist an integer $s_0^\star$ and a positive number w_0 such that $\|\mathbf{w}_{j,k}^*\|_0 \leq s_0^\star - 1$ and $\|\mathbf{w}_{j,k}^*\|_1 \leq w_0$. Besides, the regularization parameter λ_D in (11) satisfies $\lambda_D \asymp \max\{1, w_0\} s^*\lambda \log^2 d$. Moreover, we assume that*

$$\lim_{n\to\infty} (1+w_0+w_0^2) s^*\lambda \log^2 d = 0, \quad \lim_{n\to\infty} (1+w_0) s_0^\star \lambda_D = 0, \quad \text{and} \quad \lim_{n\to\infty} \sqrt{n}(s^*+s_0^\star)\lambda\lambda_D = 0. \tag{21}$$

In addition, recall that we denote the s-sparse eigenvalues of $\mathbb{E}[\nabla^2 L_j(\boldsymbol{\beta}_j^*)]$ by $\rho_{j-}^*(s)$ and $\rho_{j+}^*(s)$. We further assume that there exist an integer $k_0^\star \geq s_0^\star$ and a positive number ν_* such that

$$\lim_{n\to\infty} k_0^\star \big(\log^9 d/n\big)^{1/2} = 0, \quad 0 < \nu_* \leq \rho_{j-}^*(s_0^\star + k_0^\star) < \rho_{j+}^*(k_0^\star) \leq \big(1 + 0.5k_0^\star/s_0^\star\big)\nu_*, \quad 1 \leq j \leq d.$$

If we can treat w_0 as a constant, and k^* and $k_0^\star$ is of the same order of s^* and $s_0^\star$, respectively, Assumption 7 is reduced to $\lambda_D \asymp s^*\lambda \log^2 d$, $s_0^\star\lambda_D = o(1)$, $s^*\lambda\log^2 d = o(1)$, and $(s^*+s^\star)\lambda\lambda_D = o(n^{1/2})$. Since $\lambda \asymp \sqrt{\log d/n}$, we can choose $\lambda_D = Cs^*(\log^5 d/n)^{1/2}$ with a sufficiently large C, provided $(s^* + s_0^\star)(\log^9 d/n)^{1/2} = o(1)$, $s_0^\star s^*(\log^5 d/n)^{1/2} = o(1)$, and $(s^* + s_0^\star)s^* \log^3 d/n = o(n^{-1/2})$. Hence this condition is fulfilled if

$$\log d = o\Big(\min\big\{(\sqrt{n}/s^*)^{2/9}, (\sqrt{n}/s_0^\star)^{2/9}, (\sqrt{n}/s^{*2})^{1/3}, (\sqrt{n}/s^*s^\star)^{1/3}\big\}\Big).$$

Now we are ready to present the main theorem of composite pairwise score test.

Theorem 8 *Under the Assumptions 2, 4, 6 and 7, it holds uniformly for all $j \neq k$ and $j, k \in [d]$ that $\sqrt{n}\widehat{S}_{jk} = \sqrt{n}S_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big) + o_{\mathbb{P}}(1)$. Furthermore, we let $\widehat{\boldsymbol{\beta}}'_{j\vee k} = (0, \widehat{\boldsymbol{\beta}}_{j\backslash k}^T, \widehat{\boldsymbol{\beta}}_{k\backslash j}^T)^T$ and define $\widehat{\boldsymbol{\Sigma}}^{jk} := n^{-1}\sum_{i=1}^n \big\{(n-1)^{-1}\sum_{i'\neq i} \mathbf{h}_{ii'}^{jk}(\widehat{\boldsymbol{\beta}}'_{j\vee k})\big\}^{\otimes 2}$, where $\mathbf{h}_{ii'}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ is the kernel function of the second-order U-statistic $\nabla L_{jk}(\boldsymbol{\beta}_{j\vee k})$. In addition, we define $\widehat{\sigma}_{jk}$ by*

$$\widehat{\sigma}_{jk}^2 := \widehat{\boldsymbol{\Sigma}}_{jk,jk}^{jk} - 2\widehat{\boldsymbol{\Sigma}}_{jk,j\backslash k}^{jk}\widehat{\mathbf{w}}_{j,k} - 2\widehat{\boldsymbol{\Sigma}}_{jk,k\backslash j}^{jk}\widehat{\mathbf{w}}_{k,j} + \widehat{\mathbf{w}}_{j,k}^T\widehat{\boldsymbol{\Sigma}}_{j\backslash k,j\backslash k}^{jk}\widehat{\mathbf{w}}_{j,k} + \widehat{\mathbf{w}}_{k,j}^T\widehat{\boldsymbol{\Sigma}}_{k\backslash j,k\backslash j}^{jk}\widehat{\mathbf{w}}_{k,j}.$$

Then, under the null hypothesis $H_0\colon \beta_{jk}^* = 0$, we have $\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk}) \rightsquigarrow N(0,1)$.

By Theorem 8, to test the null hypothesis $H_0\colon \beta_{jk}^* = 0$ against the alternative hypothesis $H_1 : \beta_{jk}^* \neq 0$, we reject H_0 if the studentized test statistic $\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk})$ is too extreme. Recall that the test function of the composite pairwise score test with significance level α is deboted by $\psi_{jk}(\alpha)$ in (13). The associated p-value is defined as $p_\psi^{jk} := 2\big[1 - \Phi\big(\big|\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk})\big|\big)\big]$. By Theorem 8, under H_0, we have

$$\lim_{n\to\infty} \mathbb{P}\big(\psi_{jk}(\alpha) = 1 \mid H_0\big) = \alpha \quad \text{and} \quad p_\psi^{jk} \rightsquigarrow \mathrm{Unif}[0,1] \ \text{ under } H_0,$$

where $\mathrm{Unif}[0,1]$ is the uniform distribution over $[0,1]$.

We note that our inferential approach is still valid if we replace $\widehat{\boldsymbol{\beta}}'_{j\vee k}$ in (18) by other estimators of $\boldsymbol{\beta}_{j\vee k}^*$, provided such an estimator converges to $\boldsymbol{\beta}_{j\vee k}^*$ at an appropriate statistical rate. Our theory still holds after simple modification on the proof when controlling the order of the remainder terms.

Remark 9 *There are a number of recent works on the uncertainty assessment for high dimensional linear models or generalized linear models with ℓ_1-penalty; see Lee et al. (2016); Lockhart et al. (2014); Belloni et al. (2012, 2013); Zhang and Zhang (2014); Javanmard and Montanari (2014); van de Geer et al. (2014). These works utilize the convexity and the Karush-Kuhn-Tuker conditions of the LASSO problem. Compared with these works, our pairwise score test is constructed using a nonconvex penalty function and is applicable to a larger model class. Ning et al. (2017b) consider the score test for ℓ_1-penalized semiparametric generalized linear models in the regression setting. Compared with this work, we adopt a composite score test with a nonconvex penalty and relax many technical assumptions including the bounded covariate assumption. For nonconvex penalizations, Fan and Lv (2011); Bradic et al. (2011) establish the asymptotic normality for the low dimensional and nonzero parameters in high dimensional models based on the oracle properties. However, their approach depends on the minimal signal strength assumption, which is not needed in our approach.*

5. Numerical Results

In this section we study the finite-sample performance of the proposed graph inference methods on both simulated and real-world datasets.

5.1. Simulation Studies

We first examine the numerical performance of the proposed pairwise score tests for the null hypothesis $H_0 : \beta^*_{jk} = 0$. We simulate data from the following three settings:

(i) Gaussian graphical model. We set $n = 100$ and $d = 200$. The graph structure is a 4-nearest-neighbor graph, that is, for $j, k \in [d]$, $j \neq k$, node j is connected with node k if $|j - k| = 1, 2, d-2, d-1$. More specifically, we sample $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ from a Gaussian distribution $N_d(\mathbf{0}, \boldsymbol{\Sigma})$. For the precision matrix $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$, we set $\boldsymbol{\Theta}_{jj} = 1$, $\big|\boldsymbol{\Theta}_{jk}\big| = \mu \in [0, 0.25)$ for $|j - k| = 1, 2, d-2, d-1$ and $\boldsymbol{\Theta}_{jk} = 0$ for $2 \leq |j - k| \leq d - 2$. Note that μ denotes the signal strength of the graph inference problem and $\mu \leq 0.25$ ensures that $\boldsymbol{\Theta}$ is diagonal dominant and invertible.

(ii) Ising graphical model. We set $n = 100$ and $d = 200$. The graph structure is a 10×20 grid with the sparsity level $s^* = 4$. We use Markov Chain Monte Carlo method (MCMC) to simulate n data from an Ising model with joint distribution $p(\boldsymbol{x}) \propto \exp\big(\sum_{j \neq k} \beta^*_{jk} x_j x_k\big)$ (using the package `IsingSampler` (Epskamp, 2015)). We set $|\beta^*_{jk}| = \mu \in [0, 1]$ if there exists an edge connecting node j and node k, and $\beta^*_{jk} = 0$ otherwise.

(iii) Mixed graphical model. We set $n = 100$ and $d = 200$. The graph structure is a $10 \times 10 \times 2$ grid with the sparsity level $s^* = 5$. We set the nodes in the first layer to be binomial and nodes in the second layer to be Gaussian. We set $|\beta^*_{jk}| = \mu \in [0, 1]$ if there exists an edge connecting node j and node k, and $\beta^*_{jk} = 0$ otherwise. We refer to Lee and Hastie (2015) for details.

We denote the true parameters of the graphical models as $\{\beta^*_{jk}, j \neq k\}$. We also denote $\boldsymbol{\beta}^*_j = (\beta^*_{j1}, \ldots, \beta^*_{jd})^T$. For the Gaussian graphical model, we have $\beta^*_{jk} = \boldsymbol{\Theta}_{jk}$. We first

obtain a point estimate of $\boldsymbol{\beta}^*_j$ by solving (7) using Algorithm 1 with the capped-ℓ_1 penalty $p_\lambda(u) = \lambda \min\{u, \lambda\}$. The parameter λ is chosen by 10-fold cross validation as suggested by Ning et al. (2017b).

Recall that the form of the loss function $L_j(\boldsymbol{\beta}_j)$ is exactly the loss function for logistic regression, where we use Rademacher random variables $y_{ii'}$ as response and $y_{ii'}(x_{ij} - x_{i'j})\boldsymbol{\beta}_j^T(\boldsymbol{x}_{i\backslash j} - \boldsymbol{x}_{i'\backslash j})$ as covariates, Algorithm 1 can be easily implemented by using the ℓ_1-regularized logistic regression such as the `PICASSO` package (Ge et al., 2017). In particular, the algorithm converges quickly after a few iterations, indicating that it attains a good balance between computational efficiency and statistical accuracy. Once $\widehat{\boldsymbol{\beta}}_j$ is obtained, we solve the Dantzig-type problem (11) using $\widehat{\boldsymbol{\beta}}_j$ as input. We set the regularization parameter λ_D to be 1. In practice, the performance of the proposed method is not very sensitive to the choice of λ_D.

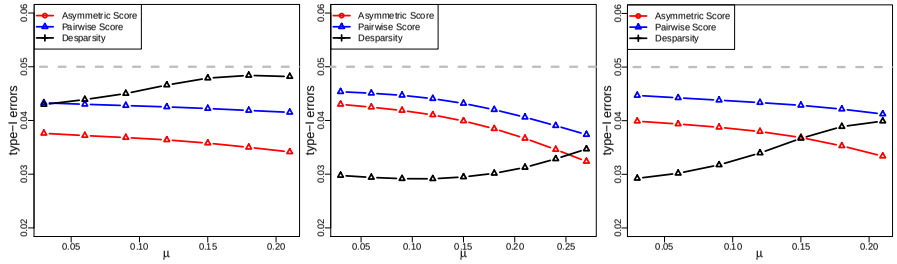
To examine the performance of our semiparametric modeling approach, we compare the pairwise score test with the desparsity method in van de Geer et al. (2014). Although this method is proposed for hypothesis tests in generalized linear models (GLMs), it can be adapted for graphical models by performing nodewise regression, assuming the base measures $\{f_j\}_{j \in [d]}$ are correctly specified. When testing $H_0 : \beta^*_{jk} = 0$ with $j < k$, we apply the desparsity method with X_j and $\boldsymbol{X}_{\backslash j}$ being the response and covariates, respectively. Furthermore, to show that combining both $L_j(\boldsymbol{\beta}_j)$ and $L_k(\boldsymbol{\beta}_k)$ is beneficial for inferring β^*_{jk}, we also compare our method with the asymmetric score test, which constructs a score test statistic similar to that in (12) based solely on $L_j(\boldsymbol{\beta}_j)$.

To examine the validity of our method, we test $H_0 : \beta^*_{jk} = 0$ versus $H_1 : \beta^*_{jk} \neq 0$ for all (j, k). Recall that $\beta^*_{jk} = \mu$ when there is an edge. Here, we let μ increase from 0 to a sufficiently large number. We calculate the type I errors and powers as

$$\text{Type I error} = \frac{\text{the number of rejected hypotheses when there is no edge}}{d(d-1)/2 - \text{the total number of edges}},$$

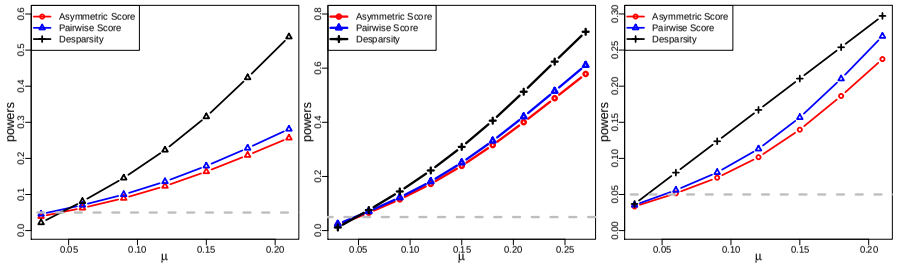
$$\text{Power} = \frac{\text{the number of rejected hypotheses when there is an edge}}{\text{the total number of edges}}.$$

We report the type I errors and powers of the hypothesis tests at the 0.05 significance level in Figure 1 and Figure 2, respectively. The simulation is repeated 100 times. As revealed in Figure 1, both the asymmetric and the pairwise score test achieve accurate type I errors, which is comparable to the desparsity method. Moreover, in terms of the power of the test, in Figure 2, the two score tests based on the loss function defined in (6) are less powerful than the desparsity method, which shows the loss of efficiency by only considering the relative rank. However, as shown in Figure 2-(b) and (c), the two score tests are nearly as powerful as the desparsity method in the Ising and mixed graphical models. In addition, we emphasize that for mixed graphical models the desparsity method needs to know the type (or distribution) of each nodes as a priori. Such phenomenon suggests that we may sacrifice little efficiency for model generality/robustness. Furthermore, comparing the performances of these two score tests, we see that the pairwise score test achieves uniformly higher power than the asymmetric one, which perfectly illustrates that taking into consideration of the symmetry of β^*_{jk} and β^*_{kj} may improve the inference accuracy.



(a). Gaussian graphical model. (b). Ising model. (c). Mixed graphical model.

Figure 1: Type-I errors of the composite pairwise score test, asymmetric score test, and the desparsity method for the three graphical models at the 0.05 significance level. These figures are based on 100 independent simulations.



(a). Gaussian graphical model. (b). Ising model. (c). Mixed graphical model.

Figure 2: Powers of the composite pairwise score test, asymmetric score test, and the desparsity method for the three graphical models at the 0.05 significance level. These figures are based on 100 independent simulations.

5.2. Real Data Analysis

We then apply the proposed methods to analyze a publicly available dataset named `Computer Audition Lab 500-Song (CAL500)` dataset (Turnbull et al., 2008). The data can be obtained from the `Mulan` database (Tsoumakas et al., 2011). The `CAL500` dataset consists of 502 popular music tracks each of which is annotated by at least three listeners. The attributes of this dataset include two subsets: (i) continuous numerical features extracted from the time series of the audio signal and (ii) discrete binary labels assigned by human listeners to give semantic descriptions of the song. For each music track, short time Fourier transform is implemented for a sequence of half-overlapping 23ms time windows over the song's digital audio file. This procedure generates four types of continuous features: *spectral centroids, spectral flux, zero crossings* and a time series of Mel-frequency cepstral coefficient

(MFCC). For the MFCC vectors, every consecutive 502 short time windows are grouped together as a block window to produce the following four types of features: (i) overall mean of MFCC vectors in each block window, (ii) mean of standard deviations of MFCC vectors in each block window, (iii) standard deviation of the means of MFCC vectors in each block window, and (iv) standard deviation of the standard deviations of MFCC vectors in each block window. More details on the feature extraction can be found in Tzanetakis and Cook (2002). In addition to these continuous variables, binary variables in the `CAL500` dataset include a 174-dimensional array indicating the existence of each annotation. These 174 annotations can be grouped into six categories: emotions (36 variables), instruments (33), usages (15), genres (47), song characteristics (27) and vocal types (16). Our goal is to infer the association between these different types of variables using graphical models. This dataset has been analyzed in Cheng et al. (2017) where they exploit a nodewise group-LASSO regression to estimate the graph structure. In what follows, we use the proposed pairwise score test to examine the graph structure.

Similar to Turnbull et al. (2008) and Cheng et al. (2017), we only keep the MFCC features because they can be interpreted as the amplitude of the audio signal and the other continuous features are not readily interpretable. Unlike Cheng et al. (2017), we keep all the binary labels. Thus the processed dataset has $n = 502$ data points of dimension $d = 226$ with 52 continuous variables and 174 binary variables. We apply the pairwise score test to each pair of variables to determine the presence of an edge between them. The p-values for the null hypothesis that two variables are conditionally independence given the rest of variables are calculated. We then apply the Bonferroni correction to control the familywise error rate at 0.05. We set the nonconvex penalty function in optimization problem (7) to be capped-ℓ_1 penalty $p_\lambda(u) = \lambda \min\{u, \lambda\}$ with the regularization parameter λ selected by 10-fold cross-validation as in the previous section.

We compare the pairwise score test with the desparsity method and the asymmetric score test, which are constructed in the same way as in the simulation. We present the fitted graphs obtained by these three methods in Figure 3-(a)–(c), where we plot the connected components and omit the singletons. Moreover, in Figure 3-(d), we plot the intersection of these three graphs. To better display the graphical structure, we use a square to represent each type of 13 MFCC features respectively. If a node is connected to any node within the group of variables in a MFCC node, then we draw an edge. We use circles to represent the binary variables and use different colors to indicate their categories. The obtained graphs have some interesting properties. While all three tests create different graphs, the graphs obtained by the pairwise score test and the asymmetric score test have more common edges, which agrees with our simulation results. Indeed, our test can correct the inconsistency of the asymmetric score test, in the sense that the asymmetric score tests for $\beta^*_{jk} = 0$ and $\beta^*_{kj} = 0$ may yield different test results. To show this inconsistency problem, we also plot the graph obtained by the asymmetric score test based on the loss function $L_k(\boldsymbol{\beta}_k)$ in Figure 4 in the appendix. Comparing with Figure 3-(b), we can see that the asymmetric score test indeed leads to many contradictory edges.

In Figure 3, both the pairwise score test and this asymmetric score test discover that songs that are danceable (circle 92) are suited for parties (circle 93), but such a connection is not found by the desparsity method. This is also true for the connection between the rapping vocals (circle 119) and the rap genre (circle 48) and the edge between strong vocals

(circle 122) and songs with strong emotions (circle 19). Moreover, in all three graphs, the continuous features are densely connected within themselves, which is similar to the results in Cheng et al. (2017). All three tests find that the noisiness of the music (square 4) is connected with the quality of songs (circle 85). Furthermore, the common edges connecting two binary variables also display interesting patterns. For instance, we find that awakening emotions (circle 6) are connected with soothing emotions (circle 8); laid-back emotions (circle 14) are connected with songs with high energy (circle 32); sad emotions (circle 20) are connected with songs with positive feelings (circle 84); songs with female lead vocals (circle 62) are connected with those with male lead vocals (circle 66). In addition, songs using drum sets (circle 59) are connected with the electronica genre (circle 46), which is also connected with the acoustic texture (circle 88). All these edges have fairly intuitive explanations.

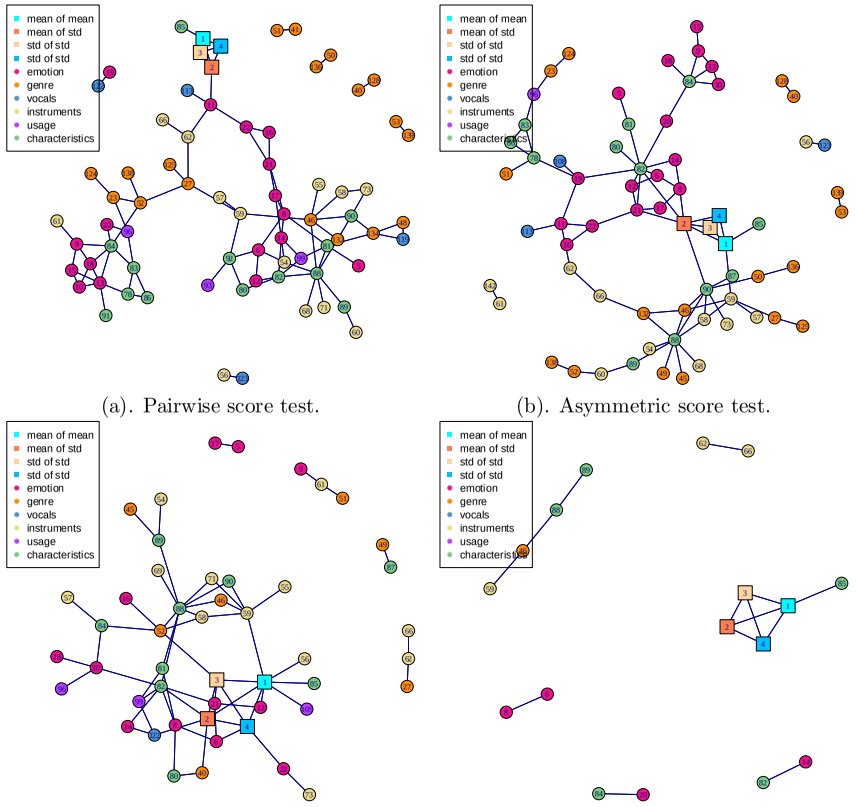
In summary, the proposed method reveals some interesting associations between these variables and can be used as a useful complement to analyze high dimensional datasets with more complex distributions.

6. Conclusion

We propose an integrated framework for uncertainty assessment of a new semiparametric exponential family graphical model. The novelty of our model is that the base measures of each nodewise conditional distribution are treated as unknown nuisance functions. Towards the goal of uncertainty assessment, we first adopt the adaptive multi-stage relaxation algorithm to perform the parameter estimation. Then we propose a composite pairwise score test of the graph structure. Our method provides a rigorous justification for the uncertainty assessment, and is further supported by extensive numerical results. In a followup paper (Tan et al., 2016), the proposed model is further extended to account for the unobserved latent variables in the graphical model.

Acknowledgments

The authors are grateful for the support of NSF CAREER Award DMS1454377, NSF IIS1408910, NSF IIS1332109, NIH R01MH102339, NIH R01GM083084, and NIH R01HG06841.



(a). Pairwise score test. (b). Asymmetric score test.

(c). Desparsity method. (d). The common edges.

Figure 3: Estimated graphs in the `CAL500` dataset inferred by the pairwise score test, asymmetric score test, and the desparsity method. We plot the connected components of the estimated graph. In (a)-(c) we plot the graphs obtained by these three approaches, respectively, and plot the common edges in (d). For better illustration, we only plot the connected components, combine the same type of continuous variables, display them as a square and draw each binary variable as a circle. The edges of the estimated graph show the association between these variables.

Appendix A. Proof of the Main Results

In this appendix we lay out the proof of the main results. In §A.1 we prove the result of parameter estimation. The proof is based an induction argument that Algorithm 1 keeps penalizing most of the irrelevant features and gradually reduces the bias in relevant features.

A.1. Proof of Theorem 5

Proof We only need to prove the theorem for one node $j \in [d]$, the proof is identical for the rest. To begin with, we first define a few index sets that play a significant role in our analysis. For all $j \in [d]$, we let $S_j := \{(j,k)\colon \beta^*_{jk} \neq 0, k \in [d]\}$ be the support of $\boldsymbol{\beta}^*_j$. For the number of iterations $\ell = 1, 2, \ldots$, let $G^\ell_j := \big\{(j,k) \notin S_j\colon \lambda^{(\ell-1)}_{jk} \geq p'_\lambda(c_2\lambda), k \in [d]\big\}$. By condition (C.3) of the penalty function $p_\lambda(u)$ (see §3.2), we have $p'_\lambda(c_2\lambda) \geq 0.91\lambda$. In addition, we let J^ℓ_j be the largest k^* components of $\big[\widehat{\boldsymbol{\beta}}^{(\ell)}_j\big]_{G^\ell_j}$ in absolute value where k^* is defined in Assumption 4. In addition, we let $I^\ell_j = (G^\ell_j)^c \cup J^\ell_j$. Moreover, for notational simplicity, we denote $\big[\boldsymbol{\beta}_j\big]_{G^\ell_j}$, $\big[\boldsymbol{\beta}_j\big]_{G^\ell_j}$ and $\big[\boldsymbol{\beta}_j\big]_{I^\ell_j}$ as $\boldsymbol{\beta}_{G^\ell_j}$, $\boldsymbol{\beta}_{J^\ell_j}$ and $\boldsymbol{\beta}_{I^\ell_j}$ respectively when no ambiguity arises.

The key point of the proof is to show that the complement of G^ℓ_j is not too large. To be more specific, we show that $\big|(G^\ell_j)^c\big| \leq 2s^*$ for $\ell \geq 1$. Since $S_j \subset (G^\ell_j)^c$, $(G^\ell_j)^c \leq 2s^*$ implies $\big|(G^\ell_j)^c - S_j\big| \leq s^*$. Note that G^ℓ_j is the set of irrelevant features that are heavily penalized in the ℓ-th iteration of the algorithm, $(G^\ell_j)^c - S$ being a small set indicates that the most of the irrelevant features are heavily penalized in each step. We show that $\big|(G^\ell)^c\big| \leq 2s^*$ for each $\ell \geq 1$ by induction.

For $\ell = 1$, we have $G^1_j = S^c_j$ because $\lambda^{(0)}_{jk} = \lambda$ for all $j, k \in [d]$. Hence $\big|(G^1_j)^c\big| \leq s^*$. Now we assume that $\big|\big(G^\ell_j\big)^c\big| \leq 2s^*$ for some integer ℓ and our goal is to prove that $\big|\big(G^{\ell+1}_j\big)^c\big| \leq 2s^*$. Our proof is based on three technical lemmas. The first lemma shows that the regularization parameter λ in (7) is of the same order as $\|\nabla L_j(\boldsymbol{\beta}^*_j)\|_\infty$.

Lemma 10 *Under Assumptions 2 and 4, there exists a positive constants K such that, it holds with probability at least $1 - (2d)^{-1}$ that*

$$\big\|\nabla L_j(\boldsymbol{\beta}^*_j)\big\|_\infty \leq K\sqrt{\log d/n}, \quad \forall j \in [d]. \tag{22}$$

Proof See §C.1 for a proof. ∎

By this lemma, we conclude that the regularization parameter $\lambda \geq 25\big\|\nabla L_j(\boldsymbol{\beta}^*_j)\big\|_\infty$ with high probability. The following lemma bounds the ℓ_1- and ℓ_2-norms of $\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j$ by the norms of its subvector under the induction assumption that $\big|(G^\ell_j)^c\big| \leq 2s^*$.

Lemma 11 *Letting the index sets S_j, G^ℓ_j, J^ℓ_j and I^ℓ_j be defined as above, we denote $\widetilde{G}^\ell_j := \big(G^\ell_j\big)^c$. Under the assumption that $|G^\ell_j| \leq 2s^*$, we have*

$$\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_2 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_{I^\ell_j} - \boldsymbol{\beta}^*_{I^\ell_j}\big\|_2 \quad \textit{and} \quad \big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_1 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_{\widetilde{G}^\ell_j} - \boldsymbol{\beta}^*_{\widetilde{G}^\ell_j}\big\|_1. \tag{23}$$

Proof See §C.2 for a detailed proof. ∎

The next lemma guarantees that $\widehat{\boldsymbol{\beta}}^{(\ell)}_j$ stays in the ℓ_1-ball centered at $\boldsymbol{\beta}^*_j$ with radius r for $\ell \geq 1$ where r appears in Assumption 4. Moreover, by showing this property of Algorithm 1 , we obtain a crude rate for parameter estimation. We summarized this result in the next lemma.

Lemma 12 *For $\ell \geq 1$ and $j \in [d]$, we denote $\boldsymbol{\lambda}^{(\ell)}_{S_j} := (\lambda^{(\ell)}_{jk}, (j,k) \in S_j)^T$. Assuming that $\big|(G^\ell_j)^c\big| \leq 2s^*$, it holds with probability at least $1 - d^{-1}$ that, for all $j \in [d]$, the estimators $\widehat{\boldsymbol{\beta}}^{(\ell)}_j$ obtained in each iteration of Algorithm 1 satisfy*

$$\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_{I^\ell_j} - \boldsymbol{\beta}^*_{I^\ell_j}\big\|_2 \leq 10\rho_*^{-1}\Big[\big\|\nabla_{\widetilde{G}^\ell_j} L_j(\boldsymbol{\beta}^*_j)\big\|_2 + \big\|\boldsymbol{\lambda}^{(\ell-1)}_{S_j}\big\|_2\Big], \quad \widetilde{G}^\ell_j := (G^\ell_j)^c. \tag{24}$$

This implies the following crude rates of convergence for $\widehat{\boldsymbol{\beta}}^{(\ell)}_j$:

$$\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_2 \leq 24\rho_*^{-1}\sqrt{s^*}\lambda \quad \textit{and} \quad \big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_1 \leq 33\rho_*^{-1}s^*\lambda. \tag{25}$$

Proof See §C.3 for a detailed proof. ∎

Now we show that $\widetilde{G}^{\ell+1}_j = (G^{\ell+1}_j)^c$ satisfies $|\widetilde{G}^{\ell+1}_j| \leq 2s^*$, which concludes our induction. Letting $A := (G^{\ell+1}_j)^c - S_j$, by the definition of $G^{\ell+1}_j$, $(j,k) \in A$ implies that $(j,k) \notin S_j$ and $p'_\lambda\big(\big|\widehat{\beta}^{(\ell)}_{jk}\big|\big) \leq p'_\lambda(c_2\lambda)$. Hence by the concavity of $p_\lambda(\cdot)$, for any $(j,k) \in A$, $\big|\widehat{\beta}^{(\ell)}_{jk}\big| \geq c_2\lambda$. Therefore we have

$$\sqrt{|A|} \leq \big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_A\big\|_2/(c_2\lambda) = \big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_A - \boldsymbol{\beta}^*_A\big\|_2/(c_2\lambda) \leq 24\rho_*^{-1}\sqrt{s^*}/c_2 \leq \sqrt{s^*}, \tag{26}$$

where the first inequality follows from $|A| \leq \sum_{(j,k)\in A}\big|\widehat{\beta}^{(\ell)}_{jk}\big|^2/(c_2\lambda)^2$. Note that (26) implies that $\big|(G^{\ell+1}_j)^c\big| \leq 2s^*$. Therefore by induction, $\big|(G^\ell_j)^c\big| \leq 2s^*$ for any $\ell \geq 1$.

Now we have shown that for $\ell \geq 1$ and $j \in [d]$, $\big|(G^\ell_j)^c\big| \leq 2s^*$ and the crude statistical rates (25) hold. In what follows, we derive the more refined rates (15) and (16).

A refined bound for $\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_2$ and $\big\|\widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j\big\|_1$: For notional simplicity, we let $\boldsymbol{\delta}^{(\ell)} = \widehat{\boldsymbol{\beta}}^{(\ell)}_j - \boldsymbol{\beta}^*_j$ and omit subscript j in S_j, G^ℓ_j, J^ℓ_j and I^ℓ_j. We also denote $\widetilde{G}^\ell := (G^\ell)^c$. We first derive a recursive bound that links $\|\boldsymbol{\delta}^{(\ell)}_{I^\ell}\|_2$ to $\|\boldsymbol{\delta}^{(\ell-1)}_{I^{\ell-1}}\|_2$. Note that by (23), $\|\boldsymbol{\delta}^{(\ell)}\|_1 \leq 2.2\|\boldsymbol{\delta}^{(\ell)}_{\widetilde{G}^\ell}\|_1 \leq 2.2\sqrt{2s^*}\|\boldsymbol{\delta}^{(\ell)}_{\widetilde{G}^\ell}\|_2$. Hence we only need to control $\|\boldsymbol{\delta}^{(\ell)}_{I^\ell}\|_2$ to obtain the statistical rates of convergence for $\widehat{\boldsymbol{\beta}}^{(\ell)}_j$. By triangle inequality,

$$\big\|\nabla_{\widetilde{G}^\ell} L_j(\boldsymbol{\beta}^*_j)\big\|_2 \leq \big\|\nabla_S L_j(\boldsymbol{\beta}^*_j)\big\|_2 + \sqrt{|\widetilde{G}^\ell - S|}\big\|\nabla L_j(\boldsymbol{\beta}^*_j)\big\|_\infty.$$

Since $\lambda > 25\big\|\nabla L_j(\boldsymbol{\beta}^*_j)\big\|_\infty$, (26) implies that

$$\big\|\nabla_{\widetilde{G}^\ell} L_j(\boldsymbol{\beta}^*_j)\big\|_2 \leq \big\|\nabla_S L_j(\boldsymbol{\beta}^*_j)\big\|_2 + \big\|\boldsymbol{\delta}^{(\ell-1)}_A\big\|_2/(25c_2), \tag{27}$$

where $A := (G^\ell)^c - S \subset I^\ell$. Thus (27) can be written as

$$\big\|\nabla_{\widetilde{G}^\ell} L_j(\boldsymbol{\beta}^*_j)\big\|_2 \leq \big\|\nabla_S L_j(\boldsymbol{\beta}^*_j)\big\|_2 + \big\|\boldsymbol{\delta}^{(\ell-1)}_{I^\ell}\big\|_2/(25c_2). \tag{28}$$

Also notice that $\forall \beta_{jk} \in \mathbb{R}$, if $|\beta_{jk} - \beta^*_{jk}| \geq c_2\lambda$,

$$p'_\lambda(|\beta_{jk}|) \leq \lambda \leq |\beta_{jk} - \beta^*_{jk}|/c_2;$$

otherwise we have $|\beta^*_{jk}| - |\beta_{jk}| \leq |\beta_{jk} - \beta^*_{jk}| < c_2\lambda$ and thus $p'_\lambda(|\beta_{jk}|) \leq p'_\lambda\big(|\beta^*_{jk}| - c_2\lambda\big)$ by the concavity of $p_\lambda(\cdot)$. Hence the following inequality always holds:

$$p'_\lambda(|\beta_{jk}|) \leq p'_\lambda\big(|\beta^*_{jk}| - c_2\lambda\big) + |\beta_{jk} - \beta^*_{jk}|/c_2. \tag{29}$$

Applying (29) to $\widehat{\boldsymbol{\beta}}_j^{(\ell-1)}$ we have

$$\big\|\boldsymbol{\lambda}_S^{(\ell-1)}\big\|_2 \leq \Big[\sum_{(j,k)\in S} p'_\lambda\big(|\beta^*_{jk}| - c_2\lambda\big)^2\Big]^{1/2} + \Big[\sum_{(j,k)\in S} |\widehat{\beta}^{(\ell-1)}_{jk} - \beta^*_{jk}|^2\Big]^{1/2}\Big/c_2,$$

which leads to

$$\big\|\boldsymbol{\lambda}_S^{(\ell-1)}\big\|_2 \leq \Big[\sum_{(j,k)\in S} p'_\lambda\big(|\beta^*_{jk}| - c_2\lambda\big)^2\Big]^{1/2} + \big\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\big\|_2/c_2. \tag{30}$$

By (24), (28) and (30) we obtain

$$\big\|\boldsymbol{\delta}_{I^\ell}^{(\ell)}\big\|_2 \leq 10\rho_*^{-1}\big[\big\|\nabla_S L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + \gamma\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\|_2,$$

where $\gamma := 11(c_2\rho_*)^{-1}$ and we define $\Upsilon_j := \big[\sum_{(j,k)\in S} p'_\lambda\big(|\beta^*_{jk}| - c_2\lambda\big)^2\big]^{1/2}$ for notational simplicity. Note that since $c_2 \geq 24\rho_*^{-1}$, we have $\gamma < 1$. By recursion we obtain

$$\big\|\boldsymbol{\delta}_{I^\ell}^{(\ell)}\big\|_2 \leq 10\varrho\big[\big\|\nabla_S L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + \gamma^{\ell-1}\big\|\boldsymbol{\delta}_{I^1}^{(1)}\big\|_2, \tag{31}$$

where $\varrho := \rho_*^{-1} \cdot (1-\gamma)^{-1} = c_2(c_2\rho_* - 11)^{-1}$. Using $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}_{I_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{I_j^\ell}^*\big\|_2$, we can bound $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2$ by

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 22\varrho\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + 2.2\gamma^{\ell-1}\big\|\boldsymbol{\delta}_{I_j^1}^{(1)}\big\|_2.$$

Note that for $\ell = 1$, by (24) we have

$$\big\|\boldsymbol{\delta}_{I_j^1}^{(1)}\big\|_2 \leq 10\rho_*^{-1}\sqrt{s^*}\big[\lambda + \sqrt{2}\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty\big] \leq 11\rho_*^{-1}\sqrt{s^*}\lambda = c_2\gamma\sqrt{s^*}\lambda. \tag{32}$$

then we establish the following bound for $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2$:

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 22\varrho\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + 2.2c_2\sqrt{s^*}\lambda\gamma^\ell. \tag{33}$$

Similarly, by $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 2.2\sqrt{2s^*}\big\|\widehat{\boldsymbol{\beta}}_{I_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{I_j^\ell}^*\big\|_2$, we obtain a bound on $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1$:

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 32\sqrt{s^*}\varrho\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + 2.2\gamma^{\ell-1}\sqrt{2s^*}\big\|\boldsymbol{\delta}_{I_j^1}^{(1)}\big\|_2. \tag{34}$$

By (32) we have $2.2\sqrt{2s^*}\big\|\boldsymbol{\delta}_{I_j^1}^{(1)}\big\|_2 \leq 3.2c_2\gamma s^*\lambda$, then the right-hand side of (34) can be bounded by

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 32\sqrt{s^*}\varrho\big[\big\|\nabla_{S_j} L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \Upsilon_j\big] + 3.2c_2 s^*\lambda\gamma^\ell. \tag{35}$$

Therefore (15) and (16) can be implied by (33) and (35) respectively. Moreover, by Lemma 12, we conclude that the statistical rates (33) and (35) hold for all $j \in [d]$ with probability at least $1 - d^{-1}$. ■

A.2. Proof of Theorem 8

Proof We first remind the reader that, for $1 \leq j \neq k \leq d$, we denote

$$\boldsymbol{\beta}_{j\backslash k} = (\beta_{j1}, \ldots, \beta_{jj-1}, \beta_{jj+1}, \ldots, \beta_{jk-1}, \beta_{jk+1}, \ldots, \beta_{jd})^T \in \mathbb{R}^{d-2},$$

$\boldsymbol{\beta}_{j\vee k} = (\beta_{jk}, \boldsymbol{\beta}_{j\backslash k}, \boldsymbol{\beta}_{k\backslash j})^T \in \mathbb{R}^{2d-3}$ and $\widehat{\boldsymbol{\beta}}'_{j\vee k} = (0, \widehat{\boldsymbol{\beta}}_{j\backslash k}, \widehat{\boldsymbol{\beta}}_{k\backslash j})^T$. In addition, we define $\sigma^2_{jk} = \boldsymbol{\Sigma}^{jk}_{jk,jk} - 2\boldsymbol{\Sigma}^{jk}_{jk,j\backslash k}\mathbf{w}^*_{j,k} - 2\boldsymbol{\Sigma}^{jk}_{jk,k\backslash j}\mathbf{w}^*_{k,j} + \mathbf{w}^{*T}_{j,k}\boldsymbol{\Sigma}^{jk}_{j\backslash k,j\backslash k}\mathbf{w}^*_{j,k} + \mathbf{w}^{*T}_{k,j}\boldsymbol{\Sigma}^{jk}_{k\backslash j,k\backslash j}\mathbf{w}^*_{k,j}$. To prove the theorem our goal is to prove the following two arguments:

$$\lim_{n\to\infty}\max_{j<k}\sqrt{n}\big|\widehat{S}_{jk} - S_{jk}(\boldsymbol{\beta}^*_{j\vee k})\big| = 0 \quad \text{and} \quad \lim_{n\to\infty}\max_{j<k}|\widehat{\sigma}_{jk} - \sigma_{jk}| = 0. \tag{36}$$

Note that by Lemma 14, σ^2_{jk} is the asymptotic variance of $\sqrt{n}/2 \cdot S_{jk}(\boldsymbol{\beta}^*_{j\vee k})$. Thus combining (36) and Slutsky's theorem yields the theorem. By the the expression of $S_{jk}(\boldsymbol{\beta}^*_{j\vee k})$ and $\widehat{S}_{jk}$ in (17) and (18), under null hypothesis, for a fixed pair of nodes j and k, we have $\widehat{S}_{jk} - S_{jk}(\boldsymbol{\beta}^*_{j\vee k}) = I_{1j} + I_{2j} + I_{1k} + I_{2k}$ where I_{1j} and I_{2j} are defined as

$$I_{1j} := \big[\nabla_{jk}L_j(\widehat{\boldsymbol{\beta}}'_j) - \nabla_{jk}L_j(\boldsymbol{\beta}^*_j)\big] - \widehat{\mathbf{w}}^T_{j,k}\big[\nabla_{j\backslash k}L_j(\widehat{\boldsymbol{\beta}}'_j) - \nabla_{j\backslash k}L_j(\boldsymbol{\beta}^*_j)\big] \text{ and}$$
$$I_{2j} := (\mathbf{w}^*_{j,k} - \widehat{\mathbf{w}}_{j,k})^T\nabla_{j\backslash k}L_j(\boldsymbol{\beta}^*_j);$$

whereas I_{1k} and I_{2k} are defined by interchanging j and k in I_{1j} and I_{2j}:

$$I_{1k} := \big[\nabla_{kj}L_k(\widehat{\boldsymbol{\beta}}'_k) - \nabla_{jk}L_k(\boldsymbol{\beta}^*_k)\big] - \widehat{\mathbf{w}}^T_{k,j}\big[\nabla_{k\backslash j}L_k(\widehat{\boldsymbol{\beta}}'_k) - \nabla_{k\backslash j}L_k(\boldsymbol{\beta}^*_k)\big] \text{ and}$$
$$I_{2k} := (\mathbf{w}^*_{k,j} - \widehat{\mathbf{w}}_{k,j})^T\nabla_{k\backslash j}L_j(\boldsymbol{\beta}^*_k).$$

We first bound I_{1j}. Recall that $\widehat{\boldsymbol{\beta}}'_j = (0, \widehat{\boldsymbol{\beta}}_{j\backslash k})^T$. Note that under the null hypothesis, $\beta^*_{jk} = 0$, by the Mean-Value Theorem, there exists a $\widetilde{\boldsymbol{\beta}}_{j\backslash k} \in \mathbb{R}^{d-2}$ in the line segment between $\widehat{\boldsymbol{\beta}}_{j\backslash k}$ and $\boldsymbol{\beta}^*_{j\backslash k}$ such that

$$I_{1j} = \big[\widetilde{\boldsymbol{\Lambda}}_{jk,j\backslash k} - \widehat{\mathbf{w}}^T_{j,k}\widetilde{\boldsymbol{\Lambda}}_{j\backslash k,j\backslash k}\big]\big(\widehat{\boldsymbol{\beta}}_{j\backslash k} - \boldsymbol{\beta}^*_{j\backslash k}\big),$$

where $\widetilde{\boldsymbol{\Lambda}} := \nabla^2 L_j(0, \widetilde{\boldsymbol{\beta}}_{j\backslash k})$. We let $\boldsymbol{\delta} := \widehat{\boldsymbol{\beta}}'_j - \boldsymbol{\beta}^*_j$ and denote $\nabla^2 L_j(\widehat{\boldsymbol{\beta}}'_j)$ and $\nabla^2(\boldsymbol{\beta}^*_j)$ as $\boldsymbol{\Lambda}$ and $\boldsymbol{\Lambda}^*$ respectively. From the definition of Dantzig selector we obtain

$$|I_{1j}| \leq \underbrace{\|\boldsymbol{\Lambda}_{jk,j\backslash k} - \widehat{\mathbf{w}}^T\boldsymbol{\Lambda}_{j\backslash k,j\backslash k}\|_\infty\|\boldsymbol{\delta}_{j\backslash k}\|_1}_{I_{11}} + \underbrace{\|\boldsymbol{\Lambda}_{jk,j\backslash k} - \widetilde{\boldsymbol{\Lambda}}_{jk,j\backslash k}\|_\infty\|\boldsymbol{\delta}_{j\backslash k}\|_1}_{I_{12}}$$
$$+ \underbrace{\|\widehat{\mathbf{w}}^T(\boldsymbol{\Lambda}_{j\backslash k,j\backslash k} - \widetilde{\boldsymbol{\Lambda}}_{j\backslash k,j\backslash k})\boldsymbol{\delta}_{j\backslash k}\|_\infty}_{I_{13}}.$$

Theorem 5 implies that $\|\boldsymbol{\delta}\|_1 \leq Cs^*\lambda$ with probability tending to 1 for some constant $C > 0$. Then by the definition of Dantzig selector, $I_{11} \leq Cs^*\lambda\lambda_D$. with high probability. Moreover, the constant C is the same for all (j,k). By assumption 7, $I_{11} = o(n^{-1/2})$ with probability tending to one.

For term I_{12}, Hölder's inequality implies that

$$I_{12} \leq \|\boldsymbol{\Lambda}_{jk,j\backslash k} - \widetilde{\boldsymbol{\Lambda}}_{jk,j\backslash k}\|_\infty \|\boldsymbol{\delta}_{j\backslash k}\|_1. \tag{37}$$

By Lemma 26 we obtain

$$\|\boldsymbol{\Lambda} - \widetilde{\boldsymbol{\Lambda}}\|_\infty \leq \|\boldsymbol{\Lambda} - \boldsymbol{\Lambda}^*\|_\infty + \|\boldsymbol{\Lambda}^* - \widetilde{\boldsymbol{\Lambda}}\|_\infty \leq 2Cs^*\lambda \log^2 d. \tag{38}$$

Therefore combining (37) and (38) we have

$$I_{12} \leq 2Cs^{*2}\lambda^2 \log^2 d \lesssim s^*\lambda\lambda_D \quad \text{uniformly for } 1 \leq j < k \leq d.$$

Similarly by Hölder's inequality, we have

$$I_{13} \leq \|\widehat{\mathbf{w}}_{j,k}\|_1 \|\boldsymbol{\Lambda} - \widetilde{\boldsymbol{\Lambda}}\|_\infty \|\boldsymbol{\delta}\|_1. \tag{39}$$

Notice that by the optimality of $\widehat{\mathbf{w}}_{j,k}$, $\|\widehat{\mathbf{w}}_{j,k}\|_1 \leq \|\mathbf{w}^*_{j,k}\|_1 \leq w_0$. Combining (39) and (38) we have

$$I_{13} \leq Cw_0 s^{*2}\lambda^2 \log^2 d \lesssim s^*\lambda\lambda_D \quad \text{uniformly for } 1 \leq j < k \leq d.$$

where we use the fact that $\lambda_D \gtrsim \max\{1, w_0\} s^*\lambda \log^2 d$. Therefore we conclude that for all $j \in [d]$, $|I_{1j}| \lesssim s^*\lambda\lambda_D = o_{\mathbb{P}}(n^{-1/2})$. For I_{2j}, Hölder's inequality implies that $|I_{2j}| \leq \|\mathbf{w}^*_{j,k} - \widehat{\mathbf{w}}_{j,k}\|_1 \big\|\nabla L_j(\boldsymbol{\beta}^*_j)\big\|_\infty$. To control $\|\mathbf{w}^*_{j,k} - \widehat{\mathbf{w}}_{j,k}\|_1$, we need to the following lemma to obtain the estimation error of the Dantzig selector $\widehat{\mathbf{w}}_{j,k}$.

Lemma 13 *For $1 \leq j \neq k \leq d$, let $\widehat{\mathbf{w}}_{j,k}$ be the solution of the Dantzig-type optimization problem* (11) *and let $\mathbf{w}^*_{j,k} = \mathbf{H}^j_{jk,j\backslash k}(\mathbf{H}^j_{j\backslash k,j\backslash k})^{-1}$. Under Assumptions 2, 4, 6 and 7, with probability tending to one, we have*

$$\|\widehat{\mathbf{w}}_{j,k} - \mathbf{w}^*_{j,k}\|_1 \leq 37\nu_*^{-1} s_0^\star \lambda_D \quad \textit{for all} \quad 1 \leq j \neq k \leq d.$$

Proof See §D.2 for a detailed proof. ■

Now combining Lemma 13 and Theorem 10 we obtain that

$$|I_{2j}| \leq 37\nu_*^{-1} K_1 s_0^\star \lambda_D \sqrt{\log d/n} \asymp s_0^\star \lambda\lambda_D = o(n^{-1/2}).$$

Therefore we have shown that $I_{1j} + I_{2j} = o(n^{-1/2})$ with high probability. Similarly, we also have $I_{1k} + I_{2k} = o(n^{-1/2})$ with high probability. Moreover, since the bounds for $|I_{1j}|$ and $|I_{2j}|$ is independent of the choice of $(j,k) \in \{(j,k) \colon 1 \leq j \neq k \leq d\}$, we conclude that

$$\sqrt{n}\big[\widehat{S}_{jk} - S_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big)\big] = o_{\mathbb{P}}(1) \quad \text{uniformly for } 1 \leq j < k \leq d.$$

Our next lemma characterizes the limiting distribution of $\nabla L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big)$ and is pivotal for establishing the validity of the composite pairwise score test.

Lemma 14 *For any $\mathbf{b} \in \mathbb{R}^{2d-3}$ with $\|\mathbf{b}\|_2 = 1$ and $\|\mathbf{b}\|_0 \leq \widetilde{s}$, if $\lim_{n\to\infty} \widetilde{s}/n = 0$, we have*

$$\sqrt{n}/2 \cdot \mathbf{b}^T \nabla L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) \rightsquigarrow N\big(0, \mathbf{b}^T \boldsymbol{\Sigma}^{jk} \mathbf{b}\big). \tag{40}$$

By Lemma 14 we obtain

$$\sqrt{n}/2 \cdot S\big(\boldsymbol{\beta}^*_{j\vee k}\big) = \nabla_{jk} L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) - \mathbf{w}^{*T}_{j,k} \nabla_{j\backslash k} L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) - \mathbf{w}^{*T}_{k,j} \nabla_{j\backslash k} L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) \rightsquigarrow N(0, \sigma^2_{jk}),$$

where the asymptotic variance σ^2_{jk} is given by

$$\sigma^2_{jk} = \boldsymbol{\Sigma}^{jk}_{jk,jk} - 2\boldsymbol{\Sigma}^{jk}_{jk,j\backslash k}\mathbf{w}^*_{j,k} - 2\boldsymbol{\Sigma}_{jk,k\backslash j}\mathbf{w}^*_{k,j} + \mathbf{w}^{*T}_{j,k}\boldsymbol{\Sigma}^{jk}_{j\backslash k,j\backslash k}\mathbf{w}^*_{j,k} + \mathbf{w}^{*T}_{k,j}\boldsymbol{\Sigma}^{jk}_{k\backslash j,k\backslash j}\mathbf{w}^*_{k,j}.$$

For a more accurate estimation of $\widehat{S}_{jk} - S_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big)$, we have

$$\sqrt{n}\big|\widehat{S}_{jk} - S_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big)\big| \leq \sqrt{n}\big(|I_1| + |I_2|\big) \lesssim \sqrt{n}(s^* + s_0^*)\lambda\lambda_D. \tag{41}$$

Finally, the following lemma, whose proof is deferred to the supplementary material, shows that $\widehat{\sigma}_{jk}$ is a consistent estimator of σ_{jk}.

Lemma 15 *For $1 \leq j \neq k \leq d$, we denote the asymptotic variance of $\sqrt{n}/2 \cdot S_{jk}(\boldsymbol{\beta}^*_{j\vee k})$ as σ^2_{jk}. Under Assumptions 2, 4, 6 and 7, the estimator $\widehat{\sigma}_{jk}$ satisfies $\lim_{n\to\infty} \max_{j<k} |\widehat{\sigma}_{jk} - \sigma_{jk}| = 0$.*

Proof See §D.3 for a proof. ■

Since $\widehat{\sigma}_{jk}$ is consistent for σ_{jk} by Lemma 15 and σ_{jk} is bounded away from zero by Assumption 6, Slutsky's theorem implies that $\sqrt{n}\widehat{S}_{jk}/(2\widehat{\sigma}_{jk}) \rightsquigarrow N(0,1)$. ■

Appendix B. Additional Estimation Results

We present the additional results of parameter estimation. In §B.1 we verify the sparse eigenvalue condition for Gaussian graphical models, which justifies Assumption 4 in our paper. In §B.2 we derive a more refined statistical rates of convergence for the iterates of Algorithm 1.

B.1. Verify the Sparse Eigenvalue Condition for Gaussian Graphical Models

In this subsection, we verify the sparse eigenvalue condition for Gaussian graphical models. Moreover, we show that such condition holds uniformly over a ℓ_1-ball centered at the true parameter $\boldsymbol{\beta}^*_j$.

Proposition 16 *Suppose $\boldsymbol{X} \sim N(\mathbf{0}, \boldsymbol{\Sigma})$ is a Gaussian graphical model and let $\boldsymbol{\Theta} = \boldsymbol{\Sigma}^{-1}$ be the precision matrix. For all $j \in [d]$, the conditional distribution of X_j given $\boldsymbol{X}_{\backslash j}$ is a normal distribution with mean $\boldsymbol{\beta}^{*T}_j \boldsymbol{X}_{\backslash j}$ and variance $\boldsymbol{\Theta}^{-1}_{jj}$, where $\boldsymbol{\beta}^*_j = \boldsymbol{\Theta}_{j\backslash j}$. Let $L_j(\cdot)$ be the loss function defined in* (6). *We assume that there exist positive constants D, c_λ and C_λ such that $\|\boldsymbol{\Sigma}\|_\infty \leq D$ and $c_\lambda \leq \lambda_{\min}(\boldsymbol{\Sigma}) \leq \lambda_{\max}(\boldsymbol{\Sigma}) \leq C_\lambda$. We let $s^* = \max_{j\in[d]} \|\boldsymbol{\beta}^*_j\|_0$ and also assume that there exists a constant $C_\beta > 0$ such that $\|\boldsymbol{\beta}^*_j\|_2 \leq C_\beta$ for all $j \in [d]$.*

Suppose $r > 0$ is a real number such that $r = \mathcal{O}(1/\sqrt{s^*})$. Then, there exist $\rho_*, \rho^* > 0$ such that for all $j \in [d]$, and $s = 1, \ldots, d-1$,

$$\rho_* \leq \rho_-\big(\mathbb{E}\big[\nabla^2 L_j\big], \boldsymbol{\beta}_j^*; s, r\big) \leq \rho_+\big(\mathbb{E}\big[\nabla^2 L_j\big], \boldsymbol{\beta}_j^*; s, r\big) \leq \rho^*.$$

Proof We prove this lemma in two steps. For any $\boldsymbol{\beta}_j \in \mathbb{R}^{d-1}$ such that $\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\|_1 \leq r$ and any $\mathbf{v} \in \mathbb{R}^{d-1}$ such that $\|\mathbf{v}\|_2 = 1$, we first give a lower bound for $\mathbf{v}^T\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\mathbf{v}$ by truncation. Then we give an upper bound in the second step.

Step (i): Lower Bound of $\mathbf{v}^T\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\mathbf{v}$. We denote $\mathbb{B}_j(r) \coloneqq \big\{\boldsymbol{\beta} \in \mathbb{R}^{d-1}\colon \|\boldsymbol{\beta} - \boldsymbol{\beta}_j^*\|_1 \leq r\big\}$. For two truncation levels $\tau > 0$ and $R > 0$, we denote $\mathcal{A}_{ii'} \coloneqq \big\{|X_{ij}| \leq \tau\big\} \cap \big\{|X_{i'j}| \leq \tau\big\}$, $\mathcal{B}_i \coloneqq \big\{\big|\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j\big| \leq R, \forall \boldsymbol{\beta}_j \in \mathbb{B}_j(r)\big\}$ and $\mathcal{B}_{i'} \coloneqq \big\{\big|\boldsymbol{X}_{i'\backslash j}^T\boldsymbol{\beta}_j^*\big| \leq R, \forall \boldsymbol{\beta}_j \in \mathbb{B}_j(r)\big\}$. The values of R and τ will be determined later. By the definition of $L_j(\cdot)$, for any $\boldsymbol{\beta}_j \in \mathbb{B}_j(r)$ and any $\mathbf{v} \in \mathbb{R}^{d-1}$ with $\|\mathbf{v}\|_2 = 1$, we have

$$\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j)\mathbf{v} \geq \frac{2C_1(R,\tau)}{n(n-1)}\sum_{i<i'}\big(X_{ij} - X_{i'j}\big)^2\big[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2 I(\mathcal{B}_i)T(\mathcal{B}_{i'})I(\mathcal{A}_{ii'}), \quad (42)$$

where $C_1(R,\tau) \coloneqq \exp(-4R\tau)\big[1+\exp(-4R\tau)\big]^{-2}$. For notational simplicity, we denote the right-hand side of (42) as $C_1(R,\tau)\mathbf{v}^T\boldsymbol{\Delta}\mathbf{v}$. By the properties of Gaussian graphical models, the conditional density of X_{ij} given $\mathcal{I}_i \coloneqq \big\{\boldsymbol{X}_{i\backslash j} = \boldsymbol{x}_{i\backslash j}\big\} \cap \mathcal{B}_i$ is

$$p\big(x_{ij}|\mathcal{I}_i\big) = p(\boldsymbol{x}_i|\mathcal{B}_i)\Big/\int_{\mathbb{R}} p(\boldsymbol{x}_i|\mathcal{B}_i)\mathrm{d}x_{ij} = p(x_{ij}|\boldsymbol{x}_{i\backslash j}),$$

where we use the fact that $p(\boldsymbol{x}_i|\mathbb{B}_i) = p(\boldsymbol{x}_i)/\mathbb{P}(\mathcal{B}_i)$ and that $\mathbb{P}(\mathcal{B}_i)$ is a constant. Recall that

$$p(x_{ij}|\boldsymbol{X}_{i\backslash j}) = \sqrt{\boldsymbol{\Theta}_{jj}/(2\pi)}\exp\big[-\boldsymbol{\Theta}_{jj}/2(x_{ij} - \boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j^*)^2\big] \quad \text{where } \boldsymbol{\beta}_j^* = \boldsymbol{\Theta}_{j\backslash j}.$$

Thus the conditional expectation of $(X_{ij} - X_{i'j})^2 I(\mathcal{A}_{ij})$ given $\mathcal{I}_i$ and $\mathcal{I}_{i'}$ is

$$\begin{aligned}
&\mathbb{E}\Big[(X_{ij} - X_{i'j})^2 I(\mathcal{A}_{ii'})\Big|\mathcal{I}_i \cap \mathcal{I}_{i'}\Big] \\
&\quad = \boldsymbol{\Theta}_{jj}/(2\pi)\int_{-\tau}^{\tau}\int_{-\tau}^{\tau}(x_{ij} - x_{ij})^2\exp\Big\{-\boldsymbol{\Theta}_{jj}/2\big[(x_{ij} - \boldsymbol{\beta}_j^T\boldsymbol{x}_{i\backslash j})^2 + (x_{i'j} - \boldsymbol{\beta}_j^T\boldsymbol{x}_{i'\backslash j})^2\big]\Big\}\mathrm{d}x_{ij}\mathrm{d}x_{i'j}.
\end{aligned}$$

Note that on event $\mathcal{I}_i$, $|\boldsymbol{\beta}_j^T\boldsymbol{X}_{i\backslash j}| \leq R$, hence the expression above can be lower-bounded by

$$\begin{aligned}
&\mathbb{E}\Big[(X_{ij} - X_{i'j})^2 I(\mathcal{A}_{ii'})\Big|\mathcal{I}_i \cap \mathcal{I}_{i'}\Big] \\
&\quad \geq \boldsymbol{\Theta}_{jj}/(2\pi)\int_{-\tau}^{\tau}\int_{-\tau}^{\tau}(x_{ij} - x_{i'j})^2\exp\Big\{-\boldsymbol{\Theta}_{jj}/2\big[x_{ij}^2 + x_{i'j}^2 + 2R^2 + 2R(|x_{ij}| + |x_{i'j}|)\big]\Big\}\mathrm{d}x_{ij}\mathrm{d}x_{i'j}.
\end{aligned}$$

The last expression is positive and we denote it as $C_2(R,\tau)$ for simplicity. Thus by the law of total expectation we obtain

$$\mathbf{v}^T\mathbb{E}(\boldsymbol{\Delta})\mathbf{v} = \mathbf{v}^T\mathbb{E}\big[\mathbb{E}(\boldsymbol{\Delta}|\cap_{i=1}^n\mathcal{I}_i)\big]\mathbf{v} \geq C_2(R,\tau)\mathbb{E}\Big\{\big[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2 I(\mathcal{B}_i)I(\mathcal{B}_j)\Big\}.$$

By Cauchy-Schwarz inequality we have

$$\mathbb{E}\Big\{\big[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2\big[1 - I(\mathcal{B}_i)I(\mathcal{B}_{i'})\big]\Big\} \leq \sqrt{\mathbb{E}[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}]^4}\sqrt{\mathbb{P}\big(\mathcal{B}_i^c \cup \mathcal{B}_{i'}^c\big)}. \quad (43)$$

Note that for Gaussian graphical model, the marginal distribution of $\boldsymbol{X}_{\backslash j}$ is $N(\mathbf{0}, \boldsymbol{\Sigma}_{\backslash j\backslash j})$. If we denote $\boldsymbol{\Sigma}_{\backslash j\backslash j}$ as $\boldsymbol{\Sigma}_1$, we have $(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v} \sim N(\mathbf{0}, \sigma_v^2)$, $\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j^* \sim N(\mathbf{0}, \sigma_1^2)$ and $\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta} \sim N(\mathbf{0}, \sigma_2^2)$ where $\sigma_v^2 = 2\mathbf{v}^T\boldsymbol{\Sigma}_1\mathbf{v}$, $\sigma_1^2 = \boldsymbol{\beta}_j^{*T}\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j^*$ and $\sigma_2^2 = \boldsymbol{\beta}_j^T\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j$. Hence we have $\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^4 = 3\sigma_v^4$. Because the maximum eigenvalue of $\boldsymbol{\Sigma}_1$ is upper bounded by C_λ, we have $\sigma_1^2 \leq C_\lambda C_\beta^2$ and $\sigma_v^2 \leq 2C_\lambda$. Note that $\sigma_2^2 - \sigma_1^2 = \boldsymbol{\beta}_j^T\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^{*T}\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j^*$, the following lemma in linear algebra bounds this type of error.

Lemma 17 *Let $\mathbf{M} \in \mathbb{R}^{d\times d}$ be a symmetric matrix and vectors $\mathbf{v}_1$ and $\mathbf{v}_2 \in \mathbb{R}^d$, then*

$$\big|\mathbf{v}_1^T\mathbf{M}\mathbf{v}_1 - \mathbf{v}_2^T\mathbf{M}\mathbf{v}_2\big| \leq \|\mathbf{M}\|_\infty\|\mathbf{v}_1 - \mathbf{v}_2\|_1^2 + 2\|\mathbf{M}\mathbf{v}_2\|_\infty\|\mathbf{v}_1 - \mathbf{v}_2\|_1.$$

Proof Note that $\mathbf{v}_1^T\mathbf{M}\mathbf{v}_1 - \mathbf{v}_2^T\mathbf{M}\mathbf{v}_2 = (\mathbf{v}_1 - \mathbf{v}_2)^T\mathbf{M}(\mathbf{v}_1 - \mathbf{v}_2) + 2\mathbf{v}_2^T\mathbf{M}(\mathbf{v}_1 - \mathbf{v}_2)$, Hölder's inequality implies

$$\begin{aligned}
\big|\mathbf{v}_1^T\mathbf{M}\mathbf{v}_1 - \mathbf{v}_2^T\mathbf{M}\mathbf{v}_2\big| &\leq \big|(\mathbf{v}_1 - \mathbf{v}_2)^T\mathbf{M}(\mathbf{v}_1 - \mathbf{v}_2)\big| + 2\big|\mathbf{v}_2^T\mathbf{M}(\mathbf{v}_1 - \mathbf{v}_2)\big| \\
&\leq \|\mathbf{M}\|_\infty\|\mathbf{v}_1 - \mathbf{v}_2\|_1^2 + 2\|\mathbf{M}\mathbf{v}_2\|_\infty\|\mathbf{v}_1 - \mathbf{v}_2\|_1.
\end{aligned}$$

Hence, we conclude the proof of Lemma 17. ■

By Lemma 17, we have

$$\sigma_2^2 - \sigma_1^2 \leq \|\boldsymbol{\Sigma}_1\|_\infty\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\|_1^2 + 2\|\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j^*\|_\infty\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\|_1. \quad (44)$$

By Hölder's inequality and the relation between ℓ_1-norm and ℓ_2-norm of a vector, we have $\|\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j^*\|_\infty \leq \|\boldsymbol{\Sigma}_1\|_\infty\|\boldsymbol{\beta}_j^*\|_1 \leq \sqrt{s^*}C_\beta D$. Therefore the right-hand side of (44) can be bounded by

$$\sigma_2^2 - \sigma_1^2 \leq r^2 D + 2\sqrt{s^*}rC_\beta D,$$

which shows that σ_2^2 is also bounded because $r = \mathcal{O}(1/\sqrt{s^*})$. In addition, by the bound $1 - \Phi(x) \leq \exp(-x^2/2)/(x\sqrt{2\pi})$ for the standard normal distribution function, we obtain that

$$\begin{aligned}
\mathbb{P}\big(\mathcal{B}_i^c\big) &\leq \mathbb{P}\big(\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j^* > R\big) + \mathbb{P}\big(\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j > R\big) \\
&\leq c\sigma_1\exp\big[-R^2/(2\sigma_1^2)\big]/R + c\sigma_2\exp\big[-R^2/(2\sigma_2^2)\big]/R,
\end{aligned}$$

where the constant $c = 1/\sqrt{2\pi}$. We denote the last expression as $C_3(R)$, then the right-hand side of (43) can be upper-bounded by $\sqrt{3\sigma_v^4}\sqrt{2C_3(R)} \leq 2\sqrt{6C_3(R)}C_\lambda$. Hence we can choose a sufficiently large R such that $2\sqrt{6C_3(R)}C_\lambda = \lambda_{\min}(\boldsymbol{\Sigma})$ and we denote this particular choice of R as R_0.

Now we have

$$\mathbb{E}\Big\{\big[(\boldsymbol{X}_{i\backslash j} - \boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2\big[1 - I(\mathcal{B}_i)I(\mathcal{B}_{i'})\big]\Big\} \leq \lambda_{\min}(\boldsymbol{\Sigma})$$

Note that $\mathbb{E}\big\{[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}]^2\big\}=\sigma_v^2\geq 2\lambda_{\min}(\boldsymbol{\Sigma})$, we obtain that

$$\mathbf{v}^T\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\mathbf{v}\geq C_1(R_0,\tau)C_2(R_0,\tau)\lambda_{\min}(\boldsymbol{\Sigma})\ \text{ for all }\ \tau\in\mathbb{R}.$$

Therefore we conclude that for all $\boldsymbol{\beta}_j\in\mathbb{R}^{d-1}$ such that $\|\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^*\|_1\leq r$,

$$\mathbf{v}^T\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\mathbf{v}\geq\max_{\tau\in\mathbb{R}}\big\{C_1(R_0,\tau)C_2(R_0,\tau)\big\}\lambda_{\min}(\boldsymbol{\Sigma}).\tag{45}$$

Step (ii): Upper Bound of $\mathbf{v}^T\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\mathbf{v}$. For any $\boldsymbol{\beta}_j\in\mathbb{R}^{d-1}$ such that $\|\boldsymbol{\beta}_j-\boldsymbol{\beta}_j^*\|_1\leq r$ and for any $\mathbf{v}\in\mathbb{R}^{d-1}$ with $\|\mathbf{v}\|_2=1$, by the definition of $\nabla^2 L_j(\boldsymbol{\beta}_j)$ we have

$$\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j)\mathbf{v}\leq(X_{ij}-X_{i'j})^2\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2.\tag{46}$$

Notice that conditioning on $\boldsymbol{X}_{i\backslash j}$, $X_{ij}\sim N\big(\boldsymbol{X}_{i\backslash j}^T\boldsymbol{\beta}_j^*,\boldsymbol{\Theta}_{jj}^{-1}\big)$, hence

$$\mathbb{E}\big[(X_{ij}-X_{i'j})^2\big|\boldsymbol{X}_{i\backslash j},\boldsymbol{X}_{i'\backslash j}\big]=\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\big]^2+2\boldsymbol{\Theta}_{jj}^{-1}.\tag{47}$$

Combining (46) and (47) we obtain

$$\begin{aligned}\mathbb{E}\big[\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j)\mathbf{v}\big]&\leq\mathbb{E}\Big\{\mathbb{E}\big[(X_{ij}-X_{i'j})^2\big|\boldsymbol{X}_{i\backslash j},\boldsymbol{X}_{i'\backslash j}\big]\cdot\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2\Big\}\\&\leq 2\boldsymbol{\Theta}_{jj}^{-1}\mathbb{E}\big((\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big)^2+\mathbb{E}\Big\{\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\big]^2\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2\Big\}.\end{aligned}\tag{48}$$

Because $\boldsymbol{X}_{i\backslash j}\sim N(\mathbf{0},\boldsymbol{\Sigma}_1)$ where $\boldsymbol{\Sigma}_1:=\boldsymbol{\Sigma}_{\backslash j,\backslash j}$, and also note that the maximum eigenvalue of $\boldsymbol{\Sigma}_1$ is upper bounded by C_λ, we have

$$\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2=2\mathbf{v}^T\boldsymbol{\Sigma}_1\mathbf{v}\leq 2C_\lambda.$$

Moreover, by inequality $2ab\leq a^2+b^2$ we obtain

$$2\mathbb{E}\Big\{\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\big]^2\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^2\Big\}\leq\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\big]^4+\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^4.$$

Since $(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\sim N(0,\sigma_v^2)$ and $(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\sim N(0,2\sigma_1^2)$ where σ_v^2 and σ_1^2 are defined as $2\mathbf{v}^T\boldsymbol{\Sigma}_1\mathbf{v}$ and $\boldsymbol{\beta}_j^{*T}\boldsymbol{\Sigma}_1\boldsymbol{\beta}_j^*$ respectively, we obtain

$$\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\boldsymbol{\beta}_j^*\big]^4=3\sigma_v^4\leq 12C_\lambda^2\quad\text{and}\quad\mathbb{E}\big[(\boldsymbol{X}_{i\backslash j}-\boldsymbol{X}_{i'\backslash j})^T\mathbf{v}\big]^4=12\sigma_1^4\leq 12C_\lambda C_\beta^2.$$

Therefore we can bound the right-hand side of (48) by

$$\mathbb{E}\big[\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j)\mathbf{v}\big]\leq 4\boldsymbol{\Theta}_{jj}^{-1}C_\lambda+6C_\lambda^2+6C_\lambda C_\beta^2.\tag{49}$$

Combining (45) and (49) we conclude that Proposition 16 holds with

$$\rho_*=\max_{\tau\in\mathbb{R}}\big\{C_1(R_0,\tau)C_2(R_0,\tau)\big\}\lambda_{\min}(\boldsymbol{\Sigma})\ \text{ and }\ \rho^*=4\boldsymbol{\Theta}_{jj}^{-1}C_\lambda+6C_\lambda^2+6C_\lambda C_\beta^2.$$

Therefore, we conclude the proof of Proposition 16. ■

B.2. Refined Statistical Rates of Parameter Estimation

In this subsection we show more refined statistical rates of convergence for the proposed estimators. In specific, we consider the case where $\boldsymbol{\beta}_j^*$ contains nonzero elements with both strong and week magnitudes.

Theorem 18 (Refined statistical rates of convergence) *Under Assumptions 2 and 4, we let K_1 and K_2 be the constants defined in Theorem 10 and also let $\rho_*>0$ and $r>0$ be defined in Assumption 4. For all $j\in[d]$, we define the support of $\boldsymbol{\beta}_j^*$ as $S_j:=\{(j,k)\colon\beta_{jk}^*\neq 0,k\in[d]\}$ and let $s^*=\max_{j\in[d]}\|\boldsymbol{\beta}_j^*\|_0$. The penalty function $p_\lambda(u):[0,+\infty)\to[0,+\infty)$ in (7) satisfies regularity conditions (C.1), (C.2) and (C.3) listed in §3.2 with $c_1=0.91$ and $c_2\geq 24/\rho_*$ for condition (C.3). We set the regularity parameter $\lambda=C\sqrt{\log d/n}$ such that $C\geq 25K_1$. Moreover, we assume that the penalty function $p_\lambda(u)$ satisfies an extra condition (C.4): there exists a constant $c_3>0$ such that $p_\lambda'(u)=0$ for $u\in\big[c_3\lambda,+\infty\big)$. Suppose that the support of $\boldsymbol{\beta}_j^*$ can be partitioned into $S_j=S_{1j}\cup S_{2j}$ where $S_{1j}=\big\{(j,k)\colon|\beta_{jk}^*|\geq(c_2+c_3)\lambda\big\}$ and $S_{2j}=S_j-S_{1j}$. We denote constants $A_1=22\varrho$, $A_2=2.2c_2$, $B_1=32\varrho$, $B_2=3.2c_2$, $\varrho=c_2(c_2\rho_*-11)^{-1}$, $\gamma=11c_2^{-1}\rho_*^{-1}<1$ and $a=1.04$; we let $s_{1j}^*=|S_{1j}|$ and $s_{2j}^*=|S_{2j}|$. With probability at least $1-d^{-1}$, we have the following more refined rates of convergence:*

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)}-\boldsymbol{\beta}_j^*\big\|_2\leq A_1\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2+a\sqrt{s_{2j}^*}\lambda\Big\}+A_2\sqrt{s^*}\lambda\gamma^\ell\quad\textit{and}\tag{50}$$

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)}-\boldsymbol{\beta}_j^*\big\|_1\leq B_1\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2+a\sqrt{s_{2j}^*}\lambda\Big\}+B_2 s^*\lambda\gamma^\ell,\forall j\in[d].\tag{51}$$

Proof Let $S_j=\{(j,k)\colon\beta_{jk}^*\neq 0,k\in[d]\}$ be the support of $\boldsymbol{\beta}_j^*$ and let index set G_j^ℓ, J_j^ℓ and I_j^ℓ be the same as defined in the proof of Theorem 5. For notational simplicity, we omit the subscript j in these index sets which stands for the j-th node of the graph; we simply write them as G^ℓ, J^ℓ and I^ℓ. Moreover, we let $\boldsymbol{\delta}^{(\ell)}=\widehat{\boldsymbol{\beta}}_j^{(\ell)}-\boldsymbol{\beta}_j^*$, it is shown in Lemma 12 that

$$\big\|\boldsymbol{\delta}_{I^\ell}^{(\ell)}\big\|_2\leq 10\rho_*^{-1}\Big(\big\|\nabla_{\widetilde{G}^\ell}L_j(\boldsymbol{\beta}_j^*)\big\|_2+\big\|\boldsymbol{\lambda}_{S_j}^{(\ell-1)}\big\|_2\Big);\ \ \widetilde{G}^\ell=(G^\ell)^c.\tag{52}$$

In the proof of Theorem 5, we show that $|\widetilde{G}^\ell|\leq 2s^*$ for all $j\in[d]$ and $\ell\geq 1$. Because $S_j=S_{1j}\cup S_{2j}$ where $S_{1j}=\big\{(j,k)\colon|\beta_{jk}^*|\geq(c_2+c_3)\lambda\big\}$ and $S_{2j}=S_j-S_{1j}$, then by triangle inequality we have

$$\big\|\nabla_{S_j}L_j(\boldsymbol{\beta}_j^*)\big\|_2\leq\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2+\sqrt{s_{2j}^*}\big\|\nabla_{S_{2j}}L_j(\boldsymbol{\beta}_j^*)\big\|_\infty.$$

Since $\lambda\geq 25\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty$ with high probability, by (28), we further have

$$\big\|\nabla_{\widetilde{G}^\ell}L_j(\boldsymbol{\beta}_j^*)\big\|_2\leq\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2+\sqrt{s_{2j}^*}\lambda/25+\big\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\big\|_2\big/(25c_2).\tag{53}$$

Note that by the definition of S_{1j}, for any $(j,k)\in S_{1j}$, $p_\lambda'\big(|\beta_{jk}|-c_2\lambda\big)\leq p_\lambda'(c_3\lambda)=0$, then we have

$$\Upsilon_j:=\lambda\Big[\sum_{(j,k)\in S_j}p_\lambda'\big(|\beta_{jk}^*|-c_2\lambda\big)^2\Big]^{1/2}=\lambda\Big[\sum_{(j,k)\in S_{2j}}p_\lambda'(|\beta_{jk}^*|-c_2\lambda)^2\Big]^{1/2}\leq\sqrt{s_{2j}^*}\lambda.$$

Therefore (30) is reduced to

$$\big\|\boldsymbol{\lambda}_{S_j}^{(\ell-1)}\big\|_2 \le \Upsilon_j + \big\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\big\|_2/c_2 \le \sqrt{s_{2j}^*}\lambda + \big\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\big\|_2\big/c_2. \tag{54}$$

Combining (52), (53) and (54) we obtain

$$\big\|\boldsymbol{\delta}_{I^\ell}^{(\ell)}\big\|_2 \le 10\rho_*^{-1}\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + 1.04\sqrt{s_{2j}^*}\lambda + 1.04\big\|\boldsymbol{\delta}_{I^{\ell-1}}^{(\ell-1)}\big\|_2/c_2\Big\}.$$

Then by recursion, we obtain the following estimation error:

$$\big\|\boldsymbol{\delta}_{I^\ell}^{(\ell)}\big\|_2 \le 10\varrho\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + 1.04\sqrt{s_{2j}^*}\lambda\Big\} + \gamma^{\ell-1}\big\|\boldsymbol{\delta}_{I^1}^{(1)}\big\|_2,$$

where $\gamma := 11c_2^{-1}\rho_*^{-1}$ and $\varrho := c_2(c_2\rho_* - 11)^{-1}$. Note that we assume $c_2 \ge 24\rho_*^{-1}$, for $k = 1$ by (32) we have

$$2.2\big\|\boldsymbol{\delta}_{I_1}^{(1)}\big\|_2 \le 2.2c_2\gamma\sqrt{s^*}\lambda \ \text{ and } \ 2.2\sqrt{2s^*}\big\|\boldsymbol{\delta}_{I_1}^{(1)}\big\|_2 \le 3.2c_2\gamma s^*\lambda.$$

Therefore, using the original notation, we obtain the refined rates of convergence by (23):

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \le 22\varrho\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + 1.04\sqrt{s_{2j}^*}\lambda\Big\} + 2.2c_2\gamma^\ell\sqrt{s^*}\lambda \ \text{ and}$$
$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \le 32\varrho\sqrt{s^*}\Big\{\big\|\nabla_{S_{1j}}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + 1.04\sqrt{s_{2j}^*}\lambda\Big\} + 3.2c_2\gamma^\ell s^*\lambda,$$

where $s_{2j}^* = |S_{2j}|$. Moreover, it is easy to see that, with probability at least $1 - d^{-1}$, these convergence rates hold for all $j \in [d]$. ■

Appendix C. Proof of the Auxiliary Results for Estimation

In this appendix, we prove the main results for estimation results presented in §4.1. In this appendix, we prove the auxiliary results for estimation. In specific, we give detailed proofs of Lemmas 10, 11, and 12, which are pivotal for the proof of Theorem5. We first prove Lemmas 10, which gives an upper bound for $\|\nabla L_j(\boldsymbol{\beta}_j^*)\|_\infty$.

C.1. Proof of Lemma 10

Proof By definition, $\nabla L_j(\boldsymbol{\beta}_j^*)$ is a centered second-order U-statistic with kernel function $\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*) \in \mathbb{R}^{d-1}$, whose entries are given by

$$\big[\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*)\big]_{jk} = \frac{R_{ii'}^j(\boldsymbol{\beta}_j^*)(X_{ij} - X_{i'j})(X_{ik} - X_{i'k})}{1 + R_{ii'}^j(\boldsymbol{\beta}_j^*)}.$$

By the tail probability bound in (14), for any $i \in [n]$ and $j \in [d]$, we have

$$\begin{aligned}\mathbb{P}\big(|X_{ij}| > x, \forall i \in [n], \forall j \in [d]\big) &\le \sum_{i\in[n], j\in[d]} \mathbb{P}(|X_{ij}| > x) \\ &\le 2\exp(\kappa_m + \kappa_h/2)\exp(-x + \log d + \log n).\end{aligned} \tag{55}$$

By setting $x = C\log d$ for some constant C, we conclude that event $\mathcal{E} := \{|X_{ij}| \le C\log d, \forall i \in [n], \forall j \in [d]\}$ holds with probability at least $1 - (4d)^{-1}$. Following from the same argument as in Ning et al. (2017b), it is easy to show that, conditioning on $\mathcal{E}$, $\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*)$ is also centered. Note that conditioning on event $\mathcal{E}$, $\big\|\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*)\big\|_\infty \le C\log^2 d$ for some generic constant C and for all $i, i' \in [d]$ and $j \in [d]$. The following Bernstein's inequality for U-statistics, presented in Arcones (1995), gives an upper bound for the tail probability of $\nabla L_j(\boldsymbol{\beta}_j^*)$.

Lemma 19 (Bernstein's inequality for U-statistics) *Given n i.i.d. random variables $Z_1, \ldots Z_n$ taking values in a measurable space $(\mathbb{S}, \mathcal{B})$ and a symmetric and measurable kernel function $h\colon \mathbb{S}^m \to R$, we define the U-statistics with kernel h as*

$$U := \binom{n}{m}^{-1} \sum_{i_1 < \ldots < i_m} h(Z_{i_1}, \ldots, Z_{i_m}).$$

Suppose that $\mathbb{E}[h(Z_{i_1}, \ldots, Z_{i_m})] = 0$, $\mathbb{E}\big\{\mathbb{E}[h(Z_{i_1}, \ldots, Z_{i_m}) \mid Z_{i_1}]\big\}^2 = \sigma^2$, and $\|h\|_\infty \le b$ for some positive σ and b. There exists an absolute constant $K(m) > 0$ that only depends on m such that

$$\mathbb{P}(|U| > t) \le 4\exp\big\{-nt^2/[2m^2\sigma^2 + K(m)bt]\big\}, \ \forall t > 0. \tag{56}$$

Note that by (14), the fourth moment of $\boldsymbol{X}$ is bounded, which implies that $\mathbb{E}[\mathbf{h}_{ii'}^j(\boldsymbol{\beta}_j^*)]^2$ is uniformly bounded by an absolute constant for all $j \in [d]$. By Lemma 19, setting $b = C\log^2 d$ in (56) yields that

$$\mathbb{P}\big(\big|\nabla_{jk}L_j(\boldsymbol{\beta}_j^*)\big| > t\big|\mathcal{E}\big) \le 4\exp\Big[-nt^2\big/(C_1 + C_2\log^2 d\cdot t)\Big] \tag{57}$$

for some generic constants C_1 and C_2. Taking a union bound over $\{(j,k)\colon j,k \in [d], k \ne j\}$ we obtain

$$\max_{j\in[d]}\Big\{\mathbb{P}\big(\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty > t\big|\mathcal{E}\big)\Big\} \lesssim d^2\cdot\exp\big[-nt^2\big/(C_1 + C_2\log^2 d\cdot t)\big]. \tag{58}$$

Under Assumption 4 and conditioning on $\mathcal{E}$, by setting $t = K_1\sqrt{\log d/n}$ for a sufficiently large $K_1 > 0$, it holds probability greater than $1 - (4d)^{-1}$ that

$$\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \le K_1\sqrt{\log d/n} \ \ \forall j \in [d].$$

Note that $\mathcal{E}$ holds with probability at least $1 - (4d)^{-1}$, we conclude the proof of Lemma 10. ■

C.2. Proof of Lemma 11

Proof In what follows, for notational simplicity and readability, we omit j in the subscript and ℓ in the superscript by simply writing S_j, G_j^ℓ, J_j^ℓ and I_j^ℓ as S, G, J an I respectively. By the definition of G, $\big\|\boldsymbol{\lambda}_{G^\ell}^{(\ell-1)}\big\|_{\min} \ge p_\lambda'(\theta) \ge 0.91\lambda > 22.75\|\nabla L_j(\boldsymbol{\beta}_j^*)\|_\infty$. We prove this lemma

in two steps. In the **first step** we show that $\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\|_1 \leq 2.2\|\widehat{\boldsymbol{\beta}}_{G^c}^{\ell} - \boldsymbol{\beta}_{G^c}^*\|_1$. Suppose that $\widehat{\boldsymbol{\beta}}_j^{(\ell)}$ is the solution in the ℓ-th iteration and we denote $\nabla_{jk} L_j(\boldsymbol{\beta}_j) = \partial L_j(\boldsymbol{\beta}_j)/\partial \beta_{jk}$, the Karush-Kuhn-Tucker condition implies that

$$\nabla_{jk} L_j(\widehat{\boldsymbol{\beta}}_j^{(\ell)}) + \lambda_{jk}^{(\ell-1)} \mathrm{sign}(\widehat{\beta}_{jk}^{(\ell)}) = 0 \quad if \quad \widehat{\beta}_{jk}^{(\ell)} \neq 0;$$
$$\nabla_{jk} L_j(\widehat{\boldsymbol{\beta}}_j^{(\ell)}) + \lambda_{jk}^{(\ell-1)} \xi_{jk}^{(\ell)} = 0, \quad \xi_{jk}^{(\ell)} \in [-1,1] \quad if \quad \widehat{\beta}_{jk}^{(\ell)} = 0.$$

The above Karush-Kuhn-Tuker condition can be written in a compact form as

$$\nabla L_j(\widehat{\boldsymbol{\beta}}_j^{(\ell)}) + \boldsymbol{\lambda}_j^{(\ell-1)} \circ \boldsymbol{\xi}_j^{(\ell)} = 0, \tag{59}$$

where $\boldsymbol{\xi}_j^{(\ell)} \in \partial \|\widehat{\boldsymbol{\beta}}_j^{(\ell)}\|_1$ and $\boldsymbol{\lambda}_j^{(\ell-1)} = \big(\lambda_{j1}^{(\ell-1)}, \ldots, \lambda_{jj-1}^{(\ell-1)}, \lambda_{jj+1}^{(\ell-1)}, \ldots, \lambda_{jd}^{(\ell-1)}\big)^T \in \mathbb{R}^{d-1}$.

For notational simplicity, we let $\boldsymbol{\delta} = \widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^* \in \mathbb{R}^{d-1}$ and omit the superscript ℓ and subscript j in both $\boldsymbol{\lambda}_j^{(\ell-1)}$ and $\boldsymbol{\xi}_j^{(\ell)}$ by writing them as $\boldsymbol{\lambda}$ and $\boldsymbol{\xi}$. By definition, $I = G^c \cup J$. Note that we denote the support of $\boldsymbol{\beta}_j^*$ as S; we define $H := G^c - S$, then S, H and G is a partition of $\big\{(j,k)\colon k \in [d], k \neq j\big\}$.

By the Mean-Value theorem, there exists an $\alpha \in [0,1]$ such that $\widetilde{\boldsymbol{\beta}}_j := \alpha \boldsymbol{\beta}_j^* + (1-\alpha)\widehat{\boldsymbol{\beta}}_j^{(\ell)} \in \mathbb{R}^{d-1}$ satisfies

$$\nabla L_j(\widehat{\boldsymbol{\beta}}_j) - \nabla L_j(\boldsymbol{\beta}_j^*) = \nabla^2 L_j(\widetilde{\boldsymbol{\beta}}_j)\boldsymbol{\delta}.$$

Then (59) implies that

$$0 \leq \boldsymbol{\delta}^T \nabla^2 L_j(\widetilde{\boldsymbol{\beta}}_j)\boldsymbol{\delta} = -\underbrace{\big\langle \boldsymbol{\delta}, \boldsymbol{\lambda} \circ \boldsymbol{\xi} \big\rangle}_{\text{(i)}} - \underbrace{\big\langle \nabla L_j(\boldsymbol{\beta}_j^*), \boldsymbol{\delta} \big\rangle}_{\text{(ii)}}. \tag{60}$$

For term (ii) in (60), Hölder's inequality implies that

$$\text{(ii)} \geq -\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \|\boldsymbol{\delta}\|_1. \tag{61}$$

For term (i) in (60), recall that we denote $|\mathbf{v}|$ as the vector that takes entrywise absolute value for $\mathbf{v}$. By the fact that $\xi_{jk}^{(\ell)} \widehat{\beta}_{jk}^{(\ell)} = |\widehat{\beta}_{jk}^{(\ell)}|$, we have $\boldsymbol{\xi}_G \circ \boldsymbol{\delta}_G = |\boldsymbol{\delta}_G|$ and $\boldsymbol{\xi}_H \circ \boldsymbol{\delta}_H = |\boldsymbol{\delta}_H|$. Since $\boldsymbol{\delta}_{S^c} = \widehat{\boldsymbol{\beta}}_{S^c}^{(\ell)}$. Hölder's inequality implies that

$$\begin{aligned}\big\langle \boldsymbol{\delta}, \boldsymbol{\lambda} \circ \boldsymbol{\xi} \big\rangle &= \big\langle \boldsymbol{\delta}_S, (\boldsymbol{\lambda} \circ \boldsymbol{\xi})_S \big\rangle + \big\langle |\boldsymbol{\delta}_H|, \boldsymbol{\lambda}_H \big\rangle + \big\langle |\boldsymbol{\delta}_G|, \boldsymbol{\lambda}_G \big\rangle \\ &\geq -\|\boldsymbol{\delta}_S\|_1 \|\boldsymbol{\lambda}_S\|_\infty + \|\boldsymbol{\delta}_G\|_1 \|\boldsymbol{\lambda}_G\|_{\min} + \|\boldsymbol{\delta}_H\|_1 \|\boldsymbol{\lambda}_H\|_{\min}.\end{aligned} \tag{62}$$

Combining (60), (61) and (62) we have

$$-\|\boldsymbol{\delta}_S\|_1 \|\boldsymbol{\lambda}_S\|_\infty + \|\boldsymbol{\delta}_G\|_1 \|\boldsymbol{\lambda}_G\|_{\min} + \|\boldsymbol{\delta}_H\|_1 \|\boldsymbol{\lambda}_H\|_{\min} - \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \|\boldsymbol{\delta}\|_1 \leq 0. \tag{63}$$

By the definition of G, we have $\|\boldsymbol{\lambda}_G\|_{\min} \geq p'_\lambda(c_2\lambda) \geq 0.91\lambda$. Rearranging terms in (63) we have

$$p'_\lambda(c_2\lambda)\|\boldsymbol{\delta}_G\|_1 \leq \|\boldsymbol{\delta}_G\|_1 \|\boldsymbol{\lambda}_G\|_{\min} \leq \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \|\boldsymbol{\delta}\|_1 + \|\boldsymbol{\delta}_S\|_1 \|\boldsymbol{\lambda}_S\|_\infty.$$

Using the decomposability of the ℓ_1-norm, we have

$$\Big[p'_\lambda(c_2\lambda) - \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty\Big]\|\boldsymbol{\delta}_G\|_1 \leq \Big[\|\boldsymbol{\lambda}_S\|_\infty + \|\nabla L_j(\boldsymbol{\beta}_j^*)\|_\infty\Big]\|\boldsymbol{\delta}_{G^c}\|_1 \tag{64}$$

Recall that $\lambda > 25\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty$ and $p'_\lambda(\theta) \geq 0.91\lambda$, (64) implies

$$\big\|\boldsymbol{\delta}_G\big\|_1 \leq \frac{\lambda + \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty}{p'_\lambda(c_2\lambda) - \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty}\|\boldsymbol{\delta}_{G^c}\|_1 \leq 1.2\|\boldsymbol{\delta}_{G^c}\|_1, \tag{65}$$

where we use the fact that

$$\frac{\lambda + \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty}{p'_\lambda(c_2\lambda) - \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty} \leq \frac{\lambda + 0.04\lambda}{0.91\lambda - 0.04\lambda} \leq 1.2.$$

Going back to the original notation, (65) is equivalent to

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}_{\widetilde{G}_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{\widetilde{G}_j^\ell}^*\big\|_1.$$

Now we show in the **second step** that $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}_{I_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{I_j^\ell}^*\big\|_2$. Recall that J is the largest k^* components of $\widehat{\boldsymbol{\beta}}_G^{(\ell)}$ in absolute value where we omit the subscript j and superscript ℓ in the sets G_j^ℓ, J_j^ℓ and I_j^ℓ. By the definition of J we obtain that

$$\|\boldsymbol{\delta}_{I^c}\|_\infty \leq \|\boldsymbol{\delta}_J\|_1/k^* \leq \|\boldsymbol{\delta}_G\|_1/k^*, \quad \text{where} \quad \boldsymbol{\delta} = \widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*.$$

By inequality (65) and the fact that $G^c \subset I$, we further have

$$\|\boldsymbol{\delta}_{I^c}\|_\infty \leq 1.2/k^* \cdot \|\boldsymbol{\delta}_{G^c}\|_1 \leq 1.2/k^* \cdot \|\boldsymbol{\delta}_I\|_1. \tag{66}$$

Then by Hölder' inequality and (66) we obtain that

$$\|\boldsymbol{\delta}_{I^c}\|_2 \leq \big(\|\boldsymbol{\delta}_{I^c}\|_1 \|\boldsymbol{\delta}_{I^c}\|_\infty\big)^{1/2} \leq (1.2/k^*)^{1/2}\big(\|\boldsymbol{\delta}_I\|_1 \|\boldsymbol{\delta}_{I^c}\|_1\big)^{1/2}. \tag{67}$$

By the definition of index sets G and I, we have $I^c \subset G$ and $G^c \subset I$. Then by (65) and (67) we obtain

$$\|\boldsymbol{\delta}_{I^c}\|_2 \leq (1.2/k^*)^{1/2}\big(\|\boldsymbol{\delta}_{G^c}\|_1 \|\boldsymbol{\delta}_G\|_1\big)^{1/2} \leq 1.2\|\boldsymbol{\delta}_{G^c}\|_1/\sqrt{k^*}.$$

By the norm inequality between ℓ_1-norm and ℓ_2-norm, we have

$$\|\boldsymbol{\delta}_{I^c}\|_2 \leq 1.2\|\boldsymbol{\delta}_{G^c}\|_1/\sqrt{k^*} \leq 1.2\sqrt{2s^*/k^*}\|\boldsymbol{\delta}_{G^c}\|_2 \leq 1.2\|\boldsymbol{\delta}_I\|_2,$$

where we use $k^* \geq 2s^*$ and the induction assumption that $|G| \leq 2s^*$. Then triangle inequality for ℓ_2-norm yields that

$$\|\boldsymbol{\delta}\|_2 \leq \|\boldsymbol{\delta}_{I^c}\|_2 + \|\boldsymbol{\delta}_I\|_2 \leq 2.2\|\boldsymbol{\delta}_I\|_2. \tag{68}$$

Note that (65) and (68) are equivalent to

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}_{I_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{I_j^\ell}^*\big\|_2 \ \text{ and } \ \big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 2.2\big\|\widehat{\boldsymbol{\beta}}_{\widetilde{G}_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{\widetilde{G}_j^\ell}^*\big\|_1,$$

where $\widetilde{G}_j^\ell = \big(G_j^{(\ell)}\big)^c$, which concludes the proof. ∎

C.3. Proof of Lemma 12

Proof We first show that $\widehat{\boldsymbol{\beta}}_j^{(\ell)}$ stays in the ℓ_1-ball centered at $\boldsymbol{\beta}_j^*$ with radius $r = C_\rho s^* \sqrt{\log d/n}$, where $C_\rho \geq 33\rho_*^{-1}$. For notational simplicity, we denote $\boldsymbol{\delta} = \widehat{\boldsymbol{\beta}}_j^{(\ell)} - \widehat{\boldsymbol{\beta}}_j$ and write S_j, G_j^ℓ, J_j^ℓ and I_j^ℓ as S, G, J an I respectively. We prove by contradiction. Suppose that $\|\boldsymbol{\delta}\|_1 > r$, then we define $\widetilde{\boldsymbol{\beta}}_j = \boldsymbol{\beta}_j^* + t(\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*) \in \mathbb{R}^{d-1}$ with $t \in [0,1]$ such that $\big\|\widetilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*\big\|_1 \leq r$. Letting $\widetilde{\boldsymbol{\delta}} := \widetilde{\boldsymbol{\beta}}_j - \boldsymbol{\beta}_j^*$, by (68) we obtain

$$\|\widetilde{\boldsymbol{\delta}}\|_2 = t\|\boldsymbol{\delta}\|_2 \leq 2.2t\|\boldsymbol{\delta}_I\|_2 = 2.2\|\widetilde{\boldsymbol{\delta}}_I\|_2. \tag{69}$$

Moreover, by Lemma (11) and the relation between ℓ_1- and ℓ_2-norms we have

$$\|\widetilde{\boldsymbol{\delta}}\|_1 = t\|\boldsymbol{\delta}\|_1 \leq 2.2t\|\boldsymbol{\delta}_{G^c}\|_1 \leq 2.2\sqrt{2s^*}\|\widetilde{\boldsymbol{\delta}}_I\|_2, \tag{70}$$

where we use the fact that $G^c \subset I$ and the induction assumption that $|G^c| \leq 2s^*$. By Mean-Value theorem, there exists a $\gamma \in [0,1]$ such that $\nabla L_j(\widetilde{\boldsymbol{\beta}}_j) - \nabla L_j(\boldsymbol{\beta}_j^*) = \nabla^2 L_j(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}}$, where $\boldsymbol{\beta}_1 := \gamma\boldsymbol{\beta}_j^* + (1-\gamma)\widetilde{\boldsymbol{\beta}}_j \in \mathbb{R}^{d-1}$. In what follows we will derive an upper bound for $\|\widetilde{\boldsymbol{\delta}}_I\|_2$ from $\widetilde{\boldsymbol{\delta}}^T\nabla^2 L_j(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}}$. Before doing that, we present two lemmas. The first one shows that the restricted correlation coefficients defined as follows are closely related to the sparse eigenvalues. This lemma also appear in Zhang (2010) and Zhang et al. (2013) for ℓ_2-loss.

Lemma 20 *(Local sparse eigenvalues and restricted correlation coefficients) Let m be a positive integer and $\mathbf{M}(\cdot)\colon \mathbb{R}^m \to \mathbb{S}^m$ be a mapping from $\mathbb{R}^m$ to the space of $m\times m$ symmetric matrices. We define the s-sparse eigenvalues of $\mathbf{M}(\cdot)$ over the ℓ_1-ball centered at $\mathbf{u}_0 \in \mathbb{R}^m$ with radius r as*

$$\rho_+\big(\mathbf{M}, \mathbf{u}_0; s, r\big) = \sup_{\mathbf{v},\mathbf{u}\in\mathbb{R}^m} \big\{\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v}\colon \|\mathbf{v}\|_0 \leq s, \|\mathbf{v}\|_2 = 1, \|\mathbf{u}-\mathbf{u}_0\|_1 \leq r\big\};$$
$$\rho_-\big(\mathbf{M}, \mathbf{u}_0; s, r\big) = \inf_{\mathbf{v},\mathbf{u}\in\mathbb{R}^m} \big\{\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v}\colon \|\mathbf{v}\|_0 \leq s, \|\mathbf{v}\|_2 = 1, \|\mathbf{u}-\mathbf{u}_0\|_1 \leq r\big\}.$$

In addition, we define the restricted correlation coefficients of $\mathbf{M}$ over the ℓ_1-ball centered at $\mathbf{u}_0$ with radius r as

$$\pi\big(\mathbf{M}, \mathbf{u}_0; s, k, r\big) := \sup_{\mathbf{v},\mathbf{w},\mathbf{u}\in\mathbb{R}^m} \bigg\{\frac{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{w}_J\big\|\mathbf{v}_I\big\|_2}{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big\|\mathbf{w}_J\big\|_\infty}\colon I\cap J = \emptyset, |I|\leq s, |J|\leq k, \big\|\mathbf{u}-\mathbf{u}_0\big\|_1 \leq r\bigg\}.$$

Suppose that the local sparse eigenvalue $\rho_-\big(\mathbf{M}, \mathbf{u}_0; s+k, r\big) > 0$, then we have the following upper bound on the restricted correlation coefficient $\pi(\mathbf{M}, \mathbf{u}_0; s, k)$:

$$\pi\big(\mathbf{M}, \mathbf{u}_0; s, k, r\big) \leq \frac{\sqrt{k}}{2}\sqrt{\rho_+\big(\mathbf{M}, \mathbf{u}_0; k, r\big)\Big/\rho_-\big(\mathbf{M}, \mathbf{u}_0; s+k, r\big) - 1}.$$

Proof See §E.1.1 for a detailed proof. ∎

We denote the restricted correlation coefficients of $\nabla^2 L_j(\cdot)$ over the ℓ_1-ball centered at $\boldsymbol{\beta}_j^*$ with radius r as $\pi_j(s_1, s_2) := \pi\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s_1, s_2, r\big)$ and denote the s-sparse eigenvalues $\rho_-\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ and $\rho_+\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ as $\rho_{j-}(s)$ and $\rho_{j+}(s)$ respectively. Applying Lemma 20 to $\pi_j(2s^*+k^*, k^*)$ we obtain

$$\pi_j(2s^*+k^*, k^*) \leq k^{*1/2}/2 \cdot \sqrt{\rho_{j+}(k^*)/\rho_{j-}(2s^*+2k^*) - 1}. \tag{71}$$

By the law of large numbers, if the sample size n is sufficiently large such that $\nabla^2 L_j$ is close to its expectation $\mathbb{E}\big[\nabla^2 L_j\big]$. When $\boldsymbol{\beta}_j$ is close to $\boldsymbol{\beta}_j^*$, by Assumption 4, we expect that the sparse eigenvalue condition also holds for $\nabla^2 L_j(\boldsymbol{\beta}_j)$ with high probability. The following lemma justifies this intuition.

Lemma 21 *Recall that we define the sparse eigenvalues of $\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]$ in Definition 3. Under Assumptions 2 and 4, if n is sufficiently large such that $\rho_* \gtrsim k^*\lambda\log^2 d$, with probability at least $1-(2d)^{-1}$, for all $j\in[d]$, there exists a constant $C_\rho \geq 33\rho_*^{-1}$ such that*

$$\rho_{j-}^*(2s^*+2k^*) - 0.05\rho_* \leq \rho_{j-}(2s^*+2k^*) < \rho_{j+}(k^*) \leq \rho_{j+}^*(k^*) + 0.05\rho_*, \quad \text{and}$$
$$\rho_{j+}(k^*)/\rho_{j-}(2s^*+2k^*) \leq 1 + 0.27k^*/s^*,$$

where we denote the local sparse eigenvalues $\rho_-\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ and $\rho_+\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ with $r = C_\rho\sqrt{\log d/n}$ as $\rho_{j-}(s)$ and $\rho_{j+}(s)$, respectively.

Proof See §E.1.2 for a detailed proof. ∎

Thus by Lemma 21 we have

$$\pi_j(2s^*+k^*, k^*) \leq 0.5\sqrt{0.27k^{*2}/s^*}. \tag{72}$$

By (65), (72) and $G^c \subset I$ we obtain

$$1 - 2\pi_j(2s^*+k^*, k^*)k^{*-1}\|\widetilde{\boldsymbol{\delta}}_G\|_1/\|\widetilde{\boldsymbol{\delta}}_I\|_2 \geq 1 - 1.2\sqrt{0.54} := \kappa_1, \tag{73}$$

where we denote $\kappa_1 := 1 - 1.2\sqrt{0.54} \geq 0.11$. Now we use the second lemma to get an lower bound of $\widetilde{\boldsymbol{\delta}}^T\nabla^2 L_j(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}}$, which implies an upper bound for $\|\widetilde{\boldsymbol{\delta}}_I\|_2$.

Lemma 22 *Let $\mathbf{M}\colon \mathbb{R}^m \to \mathbb{S}^m$ be a mapping from $\mathbb{R}^m$ to the space of $m\times m$-symmetric matrices. Suppose that the sparse eigenvalue $\rho_-\big(\mathbf{M}, \mathbf{u}_0; s+k, r\big) > 0$, let the restricted correlation coefficients of $\mathbf{M}(\cdot)$ be defined in Lemma 20. We denote the restricted correlation coefficients $\pi\big(\mathbf{M}, \mathbf{u}_0; s, k, r\big)$ and s-sparse eigenvalue $\rho_-\big(\mathbf{M}, \mathbf{u}_0; s, r\big)$ as $\pi(s,k)$ and $\rho_-(s)$ respectively for notational simplicity. For any $\mathbf{v}\in\mathbb{R}^d$, let F be any index set such that $|F^c| \leq s$, let J be the set of indices of the largest k entries of $\mathbf{v}_F$ in absolute value and let $I = F^c \cup J$. For any $\mathbf{u}\in\mathbb{R}^d$ such that $\|\mathbf{u}-\mathbf{u}_0\|_2 \leq r$ and any $\mathbf{v}\in\mathbb{R}^d$ satisfying $1 - 2\pi(s+k,k)\|\mathbf{v}_F\|_1/\|\mathbf{v}_I\|_2 > 0$ we have*

$$\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v} \geq \rho_-(s+k)\big[\|\mathbf{v}_I\|_2 - 2\pi(s+k,k)\|\mathbf{v}_F\|_1/k\big]\|\mathbf{v}_I\|_2.$$

Proof See §E.1.3 for a detailed proof. ■

Now applying Lemma 22 to $\nabla^2 L_j(\cdot)$ with $F=G$, $s=2s^*$ and $k=k^*$ we obtain

$$\widetilde{\boldsymbol{\delta}}^T\nabla^2 L_j(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}} \geq \rho_{j-}(2s^*+k^*)\|\widetilde{\boldsymbol{\delta}}_I\|_2\big[\|\widetilde{\boldsymbol{\delta}}_I\|_2 - 2\pi_j(2s^*+k^*,k^*)/k^*\|\widetilde{\boldsymbol{\delta}}_G\|_1\big]. \tag{74}$$

Then by (73), the right-hand side of (74) can be lower bounded by

$$\widetilde{\boldsymbol{\delta}}^T\nabla^2\ell(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}} \geq \kappa_1\rho_{j-}(2s^*+k^*)\|\widetilde{\boldsymbol{\delta}}_I\|_2^2 \geq 0.95\kappa_1\rho_*\|\widetilde{\boldsymbol{\delta}}_I\|_2^2 = \kappa_2\rho_*\|\widetilde{\boldsymbol{\delta}}_I\|_2^2, \tag{75}$$

where we let $\kappa_2 := 0.95\kappa_1 \geq 0.1$. Now we derive an upper bound for $\widetilde{\boldsymbol{\delta}}^T\nabla^2 L_j(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}}$. We define the symmetric Bregman divergence of $L_j(\boldsymbol{\beta}_j)$ as $D_j(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2) := \big\langle \boldsymbol{\beta}_1-\boldsymbol{\beta}_2, \nabla L_j(\boldsymbol{\beta}_1) - \nabla L_j(\boldsymbol{\beta}_2)\big\rangle$, where $\boldsymbol{\beta}_1,\boldsymbol{\beta}_2 \in \mathbb{R}^{d-1}$. Then by definition, $\widetilde{\boldsymbol{\delta}}^T\nabla^2\ell(\boldsymbol{\beta}_1)\widetilde{\boldsymbol{\delta}} = D_j(\widetilde{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_j^*)$. The following lemma relates $D_j(\widetilde{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_j^*)$ with $D_j(\widehat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_j^*)$.

Lemma 23 *Let $D_j(\boldsymbol{\beta}_1,\boldsymbol{\beta}_2) := \big\langle \boldsymbol{\beta}_1-\boldsymbol{\beta}_2, \nabla L_j(\boldsymbol{\beta}_1) - \nabla L(\boldsymbol{\beta}_2)\big\rangle$, $\boldsymbol{\beta}(t) = \boldsymbol{\beta}_1 + t(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)$, $t\in(0,1)$ be any point on the line segment between $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$. Then we have*

$$D_j(\boldsymbol{\beta}(t),\boldsymbol{\beta}_1) \leq tD_j(\boldsymbol{\beta}_2,\boldsymbol{\beta}_1)$$

Proof See §E.1.4 for a detailed proof. ■

By Lemma 23 and (60),

$$D_j(\widetilde{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_j^*) \leq tD_j(\widehat{\boldsymbol{\beta}}_j,\boldsymbol{\beta}_j^*) \leq \underbrace{-t\big\langle\nabla L_j(\boldsymbol{\beta}_j^*),\boldsymbol{\delta}\big\rangle}_{(i)}\underbrace{-t\big\langle\boldsymbol{\delta},\boldsymbol{\lambda}_j\circ\boldsymbol{\xi}_j\big\rangle}_{(ii)}. \tag{76}$$

For term (i) in (76), by Hölder's inequality we have

$$\begin{aligned}-t\big\langle\nabla L_j(\boldsymbol{\beta}_j^*),\boldsymbol{\delta}\big\rangle &\leq t\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2\|\boldsymbol{\delta}_{G^c}\|_2 + t\big\|\nabla_G L_j(\boldsymbol{\beta}_j^*)\big\|_\infty\|\boldsymbol{\delta}_G\|_1\\ &\leq \big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2\|\widetilde{\boldsymbol{\delta}}_I\|_2 + \big\|\nabla_G L_j(\boldsymbol{\beta}_j^*)\big\|_\infty\|\widetilde{\boldsymbol{\delta}}_G\|_1,\end{aligned} \tag{77}$$

where the inequality follows from $G^c\subset I$. For term (ii) in (76), by (62) and Hölder's inequality we have

$$-t\big\langle\boldsymbol{\delta},\boldsymbol{\lambda}_j\circ\boldsymbol{\xi}_j\big\rangle \leq -\big\langle\boldsymbol{\delta}_S,(\boldsymbol{\lambda}_j\circ\boldsymbol{\xi}_j)_S\big\rangle - \big\langle|\widetilde{\boldsymbol{\delta}}_G|,\boldsymbol{\lambda}_G\big\rangle \leq \|\boldsymbol{\lambda}_S\|_2\|\widetilde{\boldsymbol{\delta}}_I\|_2 - p'_\lambda(c_2\lambda)\|\widetilde{\boldsymbol{\delta}}_G\|_1, \tag{78}$$

where we use the Hölder's inequality and the definition of G. Combining (75),(77) and (78) we obtain that

$$\begin{aligned}\kappa_2\rho_*\big\|\widetilde{\boldsymbol{\delta}}_I\big\|_2^2 &\leq \big(\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \|\boldsymbol{\lambda}_S\|_2\big)\|\widetilde{\boldsymbol{\delta}}_I\|_2 + \big[\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty - p'_\lambda(c_2\lambda)\big]\|\widetilde{\boldsymbol{\delta}}_G\|_1\\ &\leq \big(\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \|\boldsymbol{\lambda}_S\|_2\big)\big\|\widetilde{\boldsymbol{\delta}}_I\big\|_2,\end{aligned}$$

where the second inequality follows from $p'_\lambda(c_2\lambda) > \big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty$. From the inequality above and the induction assumption $|G^c|\leq 2s^*$ we obtain that

$$\big\|\widetilde{\boldsymbol{\delta}}_I\big\|_2 \leq 10\rho_*^{-1}\big(\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \|\boldsymbol{\lambda}_S\|_2\big) \leq 10\rho_*^{-1}\sqrt{s^*}\big(\sqrt{2}\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty + \lambda\big). \tag{79}$$

Thus (70), (79) and the the fact that $25\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \leq \lambda$ imply that

$$\|\widetilde{\boldsymbol{\delta}}\|_1 \leq 22\sqrt{2}\rho_*^{-1}(1+\sqrt{2}/25)s^*\lambda < 33\rho_*^{-1}s^*\lambda \leq r, \tag{80}$$

where the last inequality follows from the definition of λ. Notice that (80) contradicts our assumption that $\|\widetilde{\boldsymbol{\delta}}\|_1 = r$, the reason for this contradiction is because we assume that $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 > r$, hence $\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq r$ and $\widetilde{\boldsymbol{\beta}}_j = \widehat{\boldsymbol{\beta}}_j^{(\ell)}$. This means that $\widehat{\boldsymbol{\beta}}_j^{(\ell)}$ stays in the ℓ_1-ball centered at $\boldsymbol{\beta}_j^*$ with radius r in each iteration.

Moreover, by (68) and (79), we obtain the following upper bound for $\|\boldsymbol{\delta}_I\|_2$:

$$\|\boldsymbol{\delta}\|_2 \leq 22\rho_*^{-1}\big(\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \|\boldsymbol{\lambda}_S\|_2\big) \leq 24\rho_*^{-1}\sqrt{s^*}\lambda,$$

where we use the condition that $\lambda \geq 25\big\|\nabla L_j(\boldsymbol{\beta}_j^*)\big\|_\infty$. In addition, by(65) and (79) we obtain the following bound on $\|\boldsymbol{\delta}\|_1$

$$\|\boldsymbol{\delta}\|_1 \leq 2.2\|\boldsymbol{\delta}_{G^c}\|_1 \leq 22\sqrt{2s^*}\rho_*^{-1}\Big(\big\|\nabla_{G^c}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \|\boldsymbol{\lambda}_S\|_2\Big) \leq 33\rho_*^{-1}s^*\lambda, \tag{81}$$

Therefore going back to the original notations, note that $\kappa_2\geq 0.1$, we establish the following crude rates of convergence for $\ell\geq 1$:

$$\big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_2 \leq 24\rho_*^{-1}\sqrt{s^*}\lambda \;\text{ and }\; \big\|\widehat{\boldsymbol{\beta}}_j^{(\ell)} - \boldsymbol{\beta}_j^*\big\|_1 \leq 33\rho_*^{-1}s^*\lambda. \tag{82}$$

And (79) is equivalent to

$$\big\|\widehat{\boldsymbol{\beta}}_{I_j^\ell}^{(\ell)} - \boldsymbol{\beta}_{I_j^\ell}^*\big\|_2 \leq 10\rho_*^{-1}\Big(\big\|\nabla_{\widetilde{G}_j^\ell}L_j(\boldsymbol{\beta}_j^*)\big\|_2 + \big\|\boldsymbol{\lambda}_{S_j}^{(\ell-1)}\big\|_2\Big),\;\; \widetilde{G}_j^\ell := (G_j^\ell)^c. \tag{83}$$

Note that we use Lemmas 10 and 21, hence (83) and (83) hold with probability at least $1-d^{-1}$ for all $j\in[d]$. ■

Appendix D. Proof of Auxiliary Results for Asymptotic Inference

We prove the auxiliary results for asymptotic inference. More specifically, we first prove Lemma 14, which is pivotal for deriving the limiting distribution of the pairwise score statistic. Then we prove the lemmas presented in the proof of Theorem 8.

D.1. Proof of Lemma 14

Proof Before proving this lemma, we first let $\nabla^2 L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ be the Hessian of $L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ and define $\mathbf{H}^{jk} := \mathbb{E}\big[\nabla^2 L_{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)\big]$. We also define

$$\boldsymbol{\Sigma}^{jk} := \mathbb{E}\big[\mathbf{g}_{jk}(\boldsymbol{X}_i)\mathbf{g}_{jk}(\boldsymbol{X}_i)^T\big] \;\text{ and }\; \boldsymbol{\Theta}^{jk} := \mathbb{E}\big[\mathbf{h}_{ii'}^{jk}(\boldsymbol{\beta}^*)\mathbf{h}_{ii'}^{jk}(\boldsymbol{\beta}^*)^T\big].$$

Under Assumption 2, we first show that there exists a positive constant D such that for any $j,k\in d$, $j\neq k$, $\max\big\{\big\|\boldsymbol{\Sigma}^{jk}\big\|_\infty, \big\|\mathbf{H}^{jk}\big\|_\infty, \big\|\boldsymbol{\Theta}^{jk}\big\|_\infty\big\} \leq D$. The reason is as follows.

Note that Hölder's inequality imply

$$\big\|\mathbf{H}^{jk}\big\|_\infty \lesssim \max_{j\in[d]}\mathbb{E}|X_{ij}-X_{i'j}|^4 \lesssim \max_{j\in[d]}\mathbb{E}|X_j|^4 \;\text{ for any } j,k\in[d], j\neq k.$$

Similarly, for $\mathbf{\Theta}^{jk}$, we also have $\big\|\mathbf{\Theta}^{jk}\big\|_\infty \lesssim \max_{j\in[d]} \mathbb{E}|X_j|^4$. By (14) we have

$$\mathbb{E}|X_j|^4 = \int_0^\infty \mathbb{P}(|X_4|^4 > t)dt \le \int_0^\infty c\exp(-t^{1/4})dt = 24c, \quad c = 2\exp(\kappa_m + \kappa_h/2).$$

Moreover, note that by the law of total variance, the diagonal elements of $\mathbf{\Sigma}^{jk}$ are no larger than the corresponding diagonal elements of $\mathbf{\Theta}^{jk}$; then by Cauchy-Schwarz inequality, $\|\mathbf{\Sigma}^{jk}\|_\infty \le \|\mathbf{\Theta}^{jk}\|_\infty$. Therefore there exists a constant D that does not depend on (s^*, n, d) such that

$$\max\big\{\|\mathbf{H}^{jk}\|_\infty, \|\mathbf{\Sigma}^{jk}\|_\infty, \|\mathbf{\Theta}^{jk}\|_\infty\big\} \le D, \quad 1 \le j < k \le d. \tag{84}$$

Now we are ready to prove the lemma. Recall that $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ is a U-statistic with kernel function $\mathbf{h}^{jk}_{ii'}(\boldsymbol{\beta}_{j\vee k})$. Because $\mathbf{h}^{jk}_{ii'}\big(\boldsymbol{\beta}^*_{j\vee k}\big)$ is centered, the law of total expectation implies that $\mathbb{E}\big[\mathbf{g}_{jk}(\boldsymbol{X}_i)\big] = \mathbf{0}$. Note that the left-hand side of (40) can be written as

$$\begin{aligned}\frac{\sqrt{n}}{2}\mathbf{b}^T\nabla L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) &= \frac{\sqrt{n}}{2}\mathbf{b}^T\mathbf{U}_{jk} + \frac{\sqrt{n}}{2}\mathbf{b}^T\big[\nabla L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) - \mathbf{U}_{jk}\big] \\ &= \underbrace{\frac{1}{\sqrt{n}}\sum_{i=1}^n \mathbf{b}^T\mathbf{g}_{jk}(\boldsymbol{X}_i)}_{I_1} + \underbrace{\frac{\sqrt{n}}{2}\mathbf{b}^T\big[\nabla L_{jk}\big(\boldsymbol{\beta}^*_{j\vee k}\big) - \mathbf{U}_{jk}\big]}_{I_2}.\end{aligned}$$

Notice that I_1 is a weighted sum of i.i.d. random variables with the mean and variance given by

$$\mathbb{E}\big[\mathbf{b}^T\mathbf{g}_{jk}(\boldsymbol{X}_i)\big] = \mathbf{0} \quad \text{and} \quad \mathrm{Var}\big[\mathbf{b}^T\mathbf{g}_{jk}(\boldsymbol{X}_i)\big] = \mathbf{b}^T\mathbf{\Sigma}^{jk}\mathbf{b}.$$

Central limit theorem implies that $I_1 \rightsquigarrow N(0, \mathbf{b}^T\mathbf{\Sigma}^{jk}\mathbf{b})$. In what follows we use $\mathbf{h}_{ii'}$ and $\mathbf{h}_{ii'|i}$ to denote $\mathbf{h}^{jk}_{ii'}\big(\boldsymbol{\beta}^*_{j\vee k}\big)$ and $\mathbb{E}\big[\mathbf{h}^{jk}_{ii'}\big(\boldsymbol{\beta}^*_{j\vee k}\big)\big|\boldsymbol{X}_i\big] = \mathbf{g}_{jk}(\boldsymbol{X}_i)$. Thus we can write I_2 as

$$I_2 = \frac{1}{\sqrt{n}(n-1)}\sum_{i<i'}\mathbf{b}^T\boldsymbol{\chi}_{ii'}, \quad \text{where } \boldsymbol{\chi}_{ii'} = (\mathbf{h}_{ii'} - \mathbf{h}_{ii'|i} - \mathbf{h}_{ii'|i'}).$$

Then $\mathbb{E}(I_2^2)$ can be expanded as

$$\mathbb{E}(I_2^2) = \frac{1}{n(n-1)^2}\sum_{i<i',s<s'}\mathbf{b}^T\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ss'}^T)\mathbf{b}. \tag{85}$$

By the definition of $\boldsymbol{\chi}_{ii'}$, we have

$$\begin{aligned}&\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ss'}^T) = \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ss'}^T) - \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ss'|s}^T) - \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ss'|s'}^T) - \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ss'}^T) \\ &+ \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ss'|s}^T) + \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ss'|s'}^T) - \mathbb{E}(\mathbf{h}_{ii'|i'}\mathbf{h}_{ss'}^T) + \mathbb{E}(\mathbf{h}_{ii'|i'}\mathbf{h}_{ss'|s}^T) + \mathbb{E}(\mathbf{h}_{ii'|i'}\mathbf{h}_{ss'|s'}^T).\end{aligned} \tag{86}$$

Therefore, for $i \ne s, s'$ and $i' \ne s, s'$, law of total expectation implies that $\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ss'}^T) = \mathbf{0}$. Similarly, if exactly one of i, i' is identical to one of s, s', say $i = s$, then (86) becomes

$$\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii''}^T) = \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii''}^T) - \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii''|i}^T) - \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii''}^T) + \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii''|i}^T), \ i \ne i' \ne i''.$$

Note that by the law of total expectation, for each term in (86) we have

$$\mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii''}^T) = \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii''|i}^T) = \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii''}^T) = \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii''|i}^T).$$

Therefore, $\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii''}^T) = \mathbf{0}$. Finally, if $i = s$ and $i' = s'$, by the law of total expectation, (86) can be further reduced to $\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii'}^T) = \mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii'}^T) - \mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii'|i}^T) - \mathbb{E}(\mathbf{h}_{ii'|i'}\mathbf{h}_{ii'|i'}^T) = \mathbf{\Theta}^{jk} - 2\mathbf{\Sigma}^{jk}$. Thus by triangle inequality we have

$$\big\|\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii'}^T)\big\|_\infty \le \big\|\mathbb{E}(\mathbf{h}_{ii'}\mathbf{h}_{ii'}^T)\big\|_\infty + \big\|\mathbb{E}(\mathbf{h}_{ii'|i}\mathbf{h}_{ii'|i}^T)\big\|_\infty + \big\|\mathbb{E}(\mathbf{h}_{ii'|j}\mathbf{h}_{ii'|j}^T)\big\|_\infty \le 3D,$$

where the last inequality follows from Assumption 6. Then equation (85) can be reduced to

$$\mathbb{E}(I_2^2) = \frac{1}{n(n-1)^2}\sum_{i<i',s<s'}\mathbf{b}^T\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ss}^T)\mathbf{b} = \frac{1}{n(n-1)^2}\sum_{i<i'}\mathbf{b}^T\mathbb{E}\big(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii'}^T\big)\mathbf{b}.$$

By Hölder's inequality we obtain

$$\begin{aligned}\mathbb{E}(I_2^2) &\le \frac{1}{2(n-1)}\|\mathbf{b}\|_1\big\|\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii'}^T)\mathbf{b}\big\|_\infty \\ &\le \frac{1}{2(n-1)}\|\mathbf{b}\|_1^2\big\|\mathbb{E}(\boldsymbol{\chi}_{ii'}\boldsymbol{\chi}_{ii'}^T)\big\|_\infty \le \frac{3D}{2(n-1)}\|\mathbf{b}\|_1^2.\end{aligned} \tag{87}$$

Since $\|\mathbf{b}\|_0 \le \widetilde{s}$, by the relationship between ℓ_1-norm and ℓ_2-norm, we can further bound the right-hand side of (87) by $\mathbb{E}(I_2^2) \le 1.5\widetilde{s}D/(n-1) \to 0$, where we use the condition that $\lim_{n\to\infty}\widetilde{s}/n = 0$. Therefore, we conclude the proof of Lemma 14. ■

D.2. Proof of Lemma 13

Proof By the definition of $\mathbf{w}^*_{j,k}$ we have $\mathbf{H}^j_{jk,j\backslash k} = \mathbf{w}^{*T}_{j,k}\mathbf{H}^j_{j\backslash k,j\backslash k}$. We let $\widehat{\boldsymbol{\beta}}'_j = (0, \widehat{\boldsymbol{\beta}}_{j\backslash k})$ and denote $\nabla^2 L_j(\widehat{\boldsymbol{\beta}}'_j)$ and $\nabla^2 L_j(\boldsymbol{\beta}^*_j)$ as $\mathbf{\Lambda}$ and $\mathbf{\Lambda}^*$ respectively. In addition, we write $\mathbf{H}^j, \mathbf{w}^*_{j,k}$ and $\widehat{\mathbf{w}}_{j,k}$ as $\mathbf{H}, \mathbf{w}^*$ and $\widehat{\mathbf{w}}$ respectively for notational simplicity. Triangle inequality implies that

$$\|\mathbf{\Lambda}_{jk,j\backslash k} - \mathbf{w}^{*T}\mathbf{\Lambda}_{j\backslash k,j\backslash k}\|_\infty \le \|\mathbf{H}_{jk,j\backslash k} - \mathbf{\Lambda}_{jk,j\backslash k}\|_\infty + \|\mathbf{w}^{*T}(\mathbf{H}_{j\backslash k,j\backslash k} - \mathbf{\Lambda}_{j\backslash k,j\backslash k})\|_\infty.$$

Hölder's inequality implies that

$$\|\mathbf{\Lambda}_{jk,j\backslash k} - \mathbf{w}^{*T}\mathbf{\Lambda}_{j\backslash k,j\backslash k}\|_\infty \le \|\mathbf{\Lambda} - \mathbf{H}\|_\infty(1 + \|\mathbf{w}^*\|_1). \tag{88}$$

Under null hypothesis, $\beta^*_{jk} = 0$. By Lemma 26, we have $\|\mathbf{\Lambda} - \mathbf{H}\|_\infty \lesssim s^*\lambda\log^2 d$. Then the right-hand side of (88) is bounded by

$$\|\mathbf{\Lambda}_{jk,j\backslash k} - \mathbf{w}^{*T}\mathbf{\Lambda}_{j\backslash k,j\backslash k}\|_\infty \lesssim (w_0 + 1)s^*\lambda\log^2 d.$$

Therefore, by the assumption that $\lambda_D \gtrsim \max\{1, w_0\}s^*\lambda\log^2 d$ we can ensure that $\mathbf{w}^*$ is in the feasible region of the Dantzig selector problem (11), hence we have $\|\widehat{\mathbf{w}}\|_1 \le \|\mathbf{w}^*\|_1 \le w_0$

by the optimality of $\widehat{\mathbf{w}}$. Let J be the support set of $\mathbf{w}^*$, that is, $J := \{(j,\ell)\colon [\mathbf{w}^*_{j,k}]_{j\ell} \neq 0, \ell \in [d], \ell \neq j\}$; the optimality of $\mathbf{w}^*$ is equivalent to $\|\widehat{\mathbf{w}}_{J^c}\|_1 + \|\widehat{\mathbf{w}}_J\|_1 \leq \|\mathbf{w}^*_J\|_1$. By triangle inequality, we have

$$\|\widehat{\mathbf{w}}_{J^c} - \mathbf{w}^*_{J^c}\|_1 = \|\widehat{\mathbf{w}}_{J^c}\|_1 \leq \|\mathbf{w}^*_J\|_1 - \|\widehat{\mathbf{w}}_J\|_1 \leq \|\widehat{\mathbf{w}}_J - \mathbf{w}^*_J\|_1, \tag{89}$$

where $J^c := \{(j,\ell)\colon (j,\ell) \notin J, j \text{ fixed}\}$. Letting $\widehat{\boldsymbol{\omega}} = \widehat{\mathbf{w}} - \mathbf{w}^*$, inequality (89) is equivalent to $\|\widehat{\boldsymbol{\omega}}_{J^c}\|_1 \leq \|\widehat{\boldsymbol{\omega}}_J\|_1$. Moreover, triangle inequality yields that

$$\|\boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}\|_\infty \leq \|\boldsymbol{\Lambda}_{jk,j\setminus k} - \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\mathbf{w}}\|_\infty + \|\boldsymbol{\Lambda}_{jk,j\setminus k} - \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\mathbf{w}^*\|_\infty \leq 2\lambda_D,$$

where the last inequality follows from that both $\mathbf{w}^*$ and $\widehat{\mathbf{w}}$ are feasible for the Dantzig selector problem (11). Then triangle inequality implies that

$$|\widehat{\boldsymbol{\omega}}^T \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}| \leq \underbrace{|\widehat{\boldsymbol{\omega}}_J^T \boldsymbol{\Lambda}_{J, j\setminus k}\widehat{\boldsymbol{\omega}}|}_{A_1} + \underbrace{|\boldsymbol{\omega}_{J^c}^T \boldsymbol{\Lambda}_{J^c, j\setminus k}\widehat{\boldsymbol{\omega}}|}_{A_2}.$$

By Hölder's inequality and inequality between ℓ_1-norm and ℓ_2-norms, we obtain that

$$A_1 \leq 2\lambda_D \|\widehat{\boldsymbol{\omega}}_J\|_1 \leq 2\sqrt{s_0^\star}\lambda_D \|\widehat{\boldsymbol{\omega}}_J\|_2 \quad \text{and} \quad A_2 \leq 2\lambda_D \|\widehat{\boldsymbol{\omega}}_{J^c}\|_1 \leq 2\lambda_D \|\widehat{\boldsymbol{\omega}}_J\|_1 \leq 2\sqrt{s_0^\star}\lambda_D \|\widehat{\boldsymbol{\omega}}_J\|_2.$$

Hence we conclude that $|\widehat{\boldsymbol{\omega}}^T \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}| \leq 4\sqrt{s_0^\star}\lambda_D \|\widehat{\boldsymbol{\omega}}_J\|_2$.

We let J_1 be the set of indices of the largest $k_0^\star$ component of $\widehat{\boldsymbol{\omega}}_{J^c}$ in absolute value and let $I = J_1 \cup J$, then $|I| \leq s_0^\star + k_0^\star$. Under the null hypothesis, $\|\widehat{\boldsymbol{\beta}}'_j - \boldsymbol{\beta}^*_j\|_1 = \|\widehat{\boldsymbol{\beta}}_{j\setminus k} - \boldsymbol{\beta}^*_{j\setminus k}\|_1 \leq 33\rho_*^{-1}s^*\lambda$. We denote the s-sparse eigenvalue of $\nabla^2_{j\setminus k, j\setminus k}L_j(\boldsymbol{\beta}_j)$ over the ℓ_1-ball centered at $\boldsymbol{\beta}^*_j$ with radius r as $\rho'_{j+}(s)$ and $\rho'_{j-}(s)$ respectively and denote the corresponding restricted correlation coefficients as $\pi'_j(s_1, s_2)$. And we denote these quantities of $\nabla^2 L_j(\boldsymbol{\beta}^*_j)$ as $\rho_{j-}(s), \rho_{j+}(s)$ and $\pi_j(s_1, s_2)$. By definition, we immediately have $\rho_{j-}(s) \leq \rho'_{j-}(s) \leq \rho'_{j+}(s) \leq \rho_{j+}(s)$.

By Lemma 22 we have

$$|\widehat{\boldsymbol{\omega}}^T \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}| \geq \rho'_{j-}\big(k^\star + s^\star\big)\big[\|\widehat{\boldsymbol{\omega}}_I\|_2 - 2\pi'_j(s^\star + k_0^\star, s_0^\star)\|\widehat{\boldsymbol{\omega}}_{J^c}\|_1 / k^\star\big]\|\widehat{\boldsymbol{\omega}}_I\|_2. \tag{90}$$

The following lemma relates the sparse eigenvalues of $\nabla^2 L_j(\boldsymbol{\beta}_j)$ to those of $\mathbb{E}\nabla^2 L_j(\boldsymbol{\beta}^*_j)$.

Lemma 24 *Under Assumptions 2, 4 and 7, if n is sufficiently large such that $\rho_* \gtrsim s^*\lambda \log^2 d$, with probability at least $1-(2d)^{-1}$, for all $j \in [d]$, there exists a constant $C_\rho \geq 33\rho_*^{-1}$ such that*

$$\rho^*_{j-}(2s_0^\star + 2k_0^\star) - 0.05\nu_* \leq \rho_{j-}(2s_0^\star + 2k_0^\star) < \rho_{j+}(k_0^\star) \leq \rho^*_{j+}(k_0^\star) + 0.05\nu_*, \quad \text{and}$$
$$\rho_{j+}(k_0^\star)/\rho_{j-}(2s_0^\star + 2k_0^\star) \leq 1 + 0.58k_0^\star/s_0^\star,$$

where we denote the local sparse eigenvalues $\rho_-\big(\nabla^2 L_j, \boldsymbol{\beta}^*_j; s, r\big)$ and $\rho_+\big(\nabla^2 L_j, \boldsymbol{\beta}^*_j; s, r\big)$ with $r = C_\rho\sqrt{\log d/n}$ as $\rho_{j-}(s)$ and $\rho_{j+}(s)$, respectively.

Proof The proof is similar to that of Lemma 3, hence is omitted here. ■

By $\|\widehat{\boldsymbol{\omega}}_{J^c}\|_1 \leq \|\widehat{\boldsymbol{\omega}}_J\|_1 \leq \sqrt{s_0^\star}\|\widehat{\boldsymbol{\omega}}_J\|_2$ and Lemma 24, the right-hand side of (90) can be reduced to

$$|\widehat{\boldsymbol{\omega}}^T \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}| \geq 0.95\nu_*\big(\|\widehat{\boldsymbol{\omega}}_I\|_2 - 2\pi'_j(s_0^\star + k_0^\star, s^\star)\|\widehat{\boldsymbol{\omega}}_J\|_2\sqrt{s_0^\star/k_0^\star})\|\widehat{\boldsymbol{\omega}}_I\|_2. \tag{91}$$

Using Lemma 20 we obtain

$$\begin{aligned} 2\pi'_j(s_0^\star + k_0^\star, k_0^\star)\sqrt{s_0^\star/k_0^\star} &\leq \sqrt{s_0^\star/k_0^\star}\sqrt{\rho'_{j+}(k_0^\star)/\rho'_{j-}(s_0^\star + 2k_0^\star) - 1} \\ &\leq \sqrt{s_0^\star/k_0^\star}\sqrt{\rho_{j+}(k_0^\star)/\rho_{j-}(s_0^\star + 2k_0^\star) - 1} \leq \sqrt{s_0^\star/k_0^\star}\sqrt{0.58k_0^\star/s_0^\star} \leq 0.76. \end{aligned}$$

Thus the right-hand side of (91) can be reduced to

$$|\widehat{\boldsymbol{\omega}}^T \boldsymbol{\Lambda}_{j\setminus k, j\setminus k}\widehat{\boldsymbol{\omega}}| \geq 0.95\nu_*(1 - 0.76\|\widehat{\boldsymbol{\omega}}_J\|_2/\|\widehat{\boldsymbol{\omega}}_I\|_2)\|\widehat{\boldsymbol{\omega}}_I\|_2^2 \geq \nu_*\kappa\|\widehat{\boldsymbol{\omega}}_I\|_2^2, \tag{92}$$

where $\kappa = 0.22$. This inequality holds because $J \subset I$. By (92) we have

$$\nu_*\kappa\|\widehat{\boldsymbol{\omega}}_I\|_2^2 \leq 4\sqrt{s_0^\star}\lambda_d\|\widehat{\boldsymbol{\omega}}_J\|_2 \leq 4\sqrt{s_0^\star}\lambda_d\|\widehat{\boldsymbol{\omega}}_I\|_2, \quad \text{which implies } \|\widehat{\boldsymbol{\omega}}_I\|_2 \leq 4\nu_*^{-1}\kappa^{-1}\sqrt{s_0^\star}\lambda_D.$$

Therefore the estimation error of $\widehat{\mathbf{w}}_{j,k}$ can be bounded by

$$\|\widehat{\boldsymbol{\omega}}\|_1 \leq 2\|\widehat{\boldsymbol{\omega}}_J\|_1 \leq 2\sqrt{s^\star}\|\widehat{\boldsymbol{\omega}}_J\|_2 \leq 8\nu_*^{-1}\kappa^{-1}s_0^\star\lambda_D \leq 37\nu_*^{-1}s_0^\star\lambda_D.$$

Returning to the original notations, we conclude that $\|\widehat{\mathbf{w}}_{j,k} - \mathbf{w}^*_{j,k}\|_1 \leq 37\nu_*^{-1}s_0^\star\lambda_D$ for all (j,k) such that $j, k \in [d]$, $j \neq k$. ■

D.3. Proof of Lemma 15

Proof We only need to show that $\widehat{\sigma}^2_{jk}$ is a consistent estimator of σ^2_{jk}, which is equivalent to showing that $\lim_{n\to\infty}|\widehat{\sigma}^2_{jk} - \sigma^2_{jk}| = 0$. To begin with, triangle inequality implies that

$$\begin{aligned} |\widehat{\sigma}^2_{jk} - \sigma^2_{jk}| \leq{}& \underbrace{\big|\widehat{\boldsymbol{\Sigma}}^{jk}_{jk,jk} - \boldsymbol{\Sigma}^{jk}_{jk,jk}\big|}_{I_1} + 2\underbrace{\big|\widehat{\mathbf{w}}^T_{j,k}\widehat{\boldsymbol{\Sigma}}^{jk}_{j\setminus k,jk} - \mathbf{w}^{*T}_{j,k}\boldsymbol{\Sigma}^{jk}_{j\setminus k,jk}\big|}_{I_{2j}} + \underbrace{\big|\widehat{\mathbf{w}}^T_{j,k}\widehat{\boldsymbol{\Sigma}}^{jk}_{j\setminus k,j\setminus k}\widehat{\mathbf{w}}_{j,k} - \mathbf{w}^{*T}_{j,k}\boldsymbol{\Sigma}^{jk}_{j\setminus k,j\setminus k}\mathbf{w}^*_{j,k}\big|}_{I_{3j}} \\ &+ 2\underbrace{\big|\widehat{\mathbf{w}}^T_{k,j}\widehat{\boldsymbol{\Sigma}}^{jk}_{k\setminus j,jk} - \mathbf{w}^{*T}_{k,j}\boldsymbol{\Sigma}^{jk}_{k\setminus j,jk}\big|}_{I_{2k}} + \underbrace{\big|\widehat{\mathbf{w}}^T_{k,j}\widehat{\boldsymbol{\Sigma}}^{jk}_{k\setminus j,k\setminus j}\widehat{\mathbf{w}}_{k,j} - \mathbf{w}^{*T}_{k,j}\boldsymbol{\Sigma}^{jk}_{k\setminus j,k\setminus j}\mathbf{w}^*_{k,j}\big|}_{I_{3k}}, \end{aligned}$$

where $\widehat{\boldsymbol{\Sigma}}^{jk} = \widehat{\boldsymbol{\Sigma}}^{jk}\big(\widehat{\boldsymbol{\beta}}'_{j\vee k}\big)$ and $\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ is defined as

$$\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k}) = \frac{1}{n}\sum_{i=1}^n \Big\{\frac{1}{n-1}\sum_{i'\neq i}\mathbf{h}^{jk}_{ii'}(\boldsymbol{\beta}_{j\vee k})\Big\}^{\otimes 2}. \tag{93}$$

To prove the consistency of $\widehat{\sigma}^2_{jk}$, we need the following theorem to show that $\widehat{\boldsymbol{\Sigma}}^{jk}$ is a consistent estimator of $\boldsymbol{\Sigma}^{jk}$ in the sense that $\big\|\widehat{\boldsymbol{\Sigma}}^{jk} - \boldsymbol{\Sigma}^{jk}\big\|_\infty$ is negligible.

Lemma 25 *For $1\leq j<k\leq d$, let $\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ be defined as (93). Suppose $\widehat{\boldsymbol{\beta}}_j$ and $\widehat{\boldsymbol{\beta}}_k$ are the estimators of $\boldsymbol{\beta}_j^*$ and $\boldsymbol{\beta}_k^*$ obtained from Algorithm 1 and we denote $\widehat{\boldsymbol{\beta}}_{j\vee k}=(\widehat{\beta}_{jk},\widehat{\boldsymbol{\beta}}_{j\setminus k}^T,\widehat{\boldsymbol{\beta}}_{k\setminus j}^T)^T$. Then $\widehat{\boldsymbol{\Sigma}}^{jk}(\widehat{\boldsymbol{\beta}}_{j\vee k})$ is a consistent estimator of $\boldsymbol{\Sigma}^{jk}$. There exists a constant C_Σ that does not depend on (j,k) such that, with probability tending to one,*

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\widehat{\boldsymbol{\beta}}_{j\vee k})-\boldsymbol{\Sigma}^{jk}\big\|_\infty\leq C_\Sigma s^*\lambda\log^2 d\quad \textit{for } 1\leq j<k\leq d.$$

Proof See §E.2.1 for a detailed proof. ∎

In the rest of the proof, we will omit the superscripts in both $\widehat{\boldsymbol{\Sigma}}^{jk}$ and $\boldsymbol{\Sigma}^{jk}$ for notational simplicity. By Lemma 25,

$$I_1\leq\|\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\|_\infty\leq\mathcal{O}_{\mathbb{P}}\big(s^*\lambda\log^2 d\big).\tag{94}$$

By triangle inequality, we have the following inequality for I_2 :

$$I_{2j}\leq\underbrace{\big|(\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*)^T\big(\widehat{\boldsymbol{\Sigma}}_{j\setminus k,jk}-\boldsymbol{\Sigma}_{j\setminus k,jk}\big)\big|}_{I_{21}}+\underbrace{\big|(\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*)^T\boldsymbol{\Sigma}_{j\setminus k,jk}\big|}_{I_{22}}+\underbrace{\big|\mathbf{w}_{j,k}^{*T}\big(\widehat{\boldsymbol{\Sigma}}_{j\setminus k,jk}-\boldsymbol{\Sigma}_{j\setminus k,jk}\big)\big|}_{I_{23}}.$$

By Hölder's inequality, Lemma 25 and the estimation error of $\widehat{\mathbf{w}}_{j,k}$, we obtain an upper-bound for I_{21} as follows:

$$I_{21}\leq\|\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*\|_1\|\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\|_\infty=\mathcal{O}_{\mathbb{P}}\big(s^*s_0^\star\lambda_D\lambda\log^2 d\big).\tag{95}$$

Similarly, for I_{22}, Hölder's inequality implies that

$$I_{22}\leq\|\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*\|_1\|\boldsymbol{\Sigma}\|_\infty=\mathcal{O}_{\mathbb{P}}\big(s_0^\star\lambda_D D\big),\tag{96}$$

where the constant D appears in (84). For I_{23}, by Hölder's inequality and 25 we obtain

$$I_{23}\leq\|\mathbf{w}_{j,k}^*\|_1\|\widehat{\boldsymbol{\Sigma}}-\boldsymbol{\Sigma}\|_\infty=\mathcal{O}_{\mathbb{P}}\big(w_0s^*\lambda\log^2 d\big).\tag{97}$$

Combining (95), (96) and (97) we have

$$I_{2j}\lesssim(w_0+s_0^\star\lambda_D)s^*\lambda\log^2 d+s_0^\star\lambda_D.\tag{98}$$

For I_{3j}, by triangle inequality we have

$$I_{3j}\leq\underbrace{\big|\widehat{\mathbf{w}}_{j,k}^T\big(\widehat{\boldsymbol{\Sigma}}_{j\setminus k,j\setminus k}-\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\big)\widehat{\mathbf{w}}_{j,k}\big|}_{I_{31}}+\underbrace{\big|\widehat{\mathbf{w}}_{j,k}^T\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^{*T}\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\mathbf{w}_{j,k}^*\big|}_{I_{32}}.$$

For term I_{31}, Hölder's inequality and the optimality of $\widehat{\mathbf{w}}$ implies that

$$I_{31}\leq\|\widehat{\mathbf{w}}_{j,k}\|_1^2\|\widehat{\boldsymbol{\Sigma}}_{j\setminus k,j\setminus k}-\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\|_\infty\leq C_\Sigma w_0^2s^*\lambda\log^2 d.\tag{99}$$

For term I_{32}, Lemma 17 implies that

$$\begin{aligned}I_{32}&\leq\|\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\|_\infty\|\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*\|_1^2+\|\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\mathbf{w}_{j,k}^*\|_\infty\|\widehat{\mathbf{w}}_{j,k}-\mathbf{w}_{j,k}^*\|_1\\&\leq\big(D\omega_0s_0^\star\lambda_D+Ds_0^{\star2}\lambda_D^2\big),\end{aligned}\tag{100}$$

where we use Hölder's inequality $\|\boldsymbol{\Sigma}_{j\setminus k,j\setminus k}\mathbf{w}_{j,k}^*\|_\infty\leq\|\mathbf{w}_{j,k}^*\|_1\|\boldsymbol{\Sigma}\|_\infty\leq Dw_0$. By (99), (100) and $\lambda_D\gtrsim w_0s^*\lambda\log^2 d$, we obtain

$$I_{3j}\lesssim w_0^2s^*\lambda\log^2 d+\big(D\omega_0s_0^\star\lambda_D+Ds_0^{\star2}\lambda_D^2\big).\tag{101}$$

Therefore combining (94), (98) and (101) we obtain $I_1+I_{2j}+I_{3j}=o_{\mathbb{P}}(1)$. We can show similarly that $I_{2k}+I_{3k}=o_{\mathbb{P}}(1)$. Thus $\lim_{n\to\infty}\max_{j<k}\big|\widehat{\sigma}_{jk}^2-\sigma_{jk}^2\big|=0$ with probability converging to one. ∎

Appendix E. Proof of Technical Lemmas

Finally, we prove the technical lemmas in this appendix. Specifically, we prove the lemmas introduced to derive the auxiliary results.

E.1. Proof of Technical Lemmas in §C

In this subsection we prove the technical lemmas we use to prove the auxiliary results of estimation. These lemmas are standard for high-dimensional linear regression, but proving them for our logistic-type loss function needs nontrivial extensions.

E.1.1. Proof of Lemma 20

Proof Let I and J be two index sets with $I\cap J=\emptyset,|I|\leq s,|J|\leq k$, for any $\mathbf{u}\in\mathbb{R}^d$ with $\|\mathbf{u}-\mathbf{u}_0\|_2\leq r$ and any $\mathbf{v},\mathbf{w}\in\mathbb{R}^d$, let $\boldsymbol{\theta}=\mathbf{v}_I+\alpha\mathbf{w}_J$ with some $\alpha\in\mathbb{R}$, then by definition, $\|\boldsymbol{\theta}\|_0\leq s+k$. For notational simplicity, we denote s-sparse eigenvalues $\rho_+\big(\mathbf{M},\mathbf{u}_0;s,r\big)$ and $\rho_-\big(\mathbf{M},\mathbf{u}_0;s,r\big)$ as $\rho_-(s)$ and $\rho_+(s)$ respectively. By definition, we have

$$\rho_-(s+k)\|\boldsymbol{\theta}\|_2^2\leq\boldsymbol{\theta}^T\mathbf{M}(\mathbf{u})\boldsymbol{\theta}=\underbrace{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I}_{A1}+2\alpha\underbrace{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{w}_J}_{A2}+\alpha^2\underbrace{\mathbf{w}_J^T\mathbf{M}(\mathbf{u})\mathbf{w}_J}_{A3}.\tag{102}$$

Since $\|\boldsymbol{\theta}\|_2^2=\|\mathbf{v}_I\|_2^2+\alpha^2\|\mathbf{w}_J\|_2^2$. Rearranging the terms in (102) we have

$$\big[A_3-\rho_-(s+k)\|\mathbf{w}_J\|_2^2\big]\alpha^2+2A_2\alpha+\big[A_1-\rho_-(s+k)\|\mathbf{v}_I\|_2^2\big]\geq0\quad\text{for all }\alpha\in\mathbb{R}.\tag{103}$$

Note that the left-hand side (103) is a univariate quadratic function in α, thus (103) implies that

$$\big[A_1-\rho_-(s+k)\|\mathbf{v}_I\|_2^2\big]\big[A_3-\rho_-(s+k)\|\mathbf{w}_J\|_2^2\big]\geq A_2^2.\tag{104}$$

Therefore by multiplying $4\|\mathbf{v}_I\|_2^\infty/(A_1^2\|\mathbf{w}_J\|_2^2)$ to both sides of (104) we have

$$\frac{4A_2^2\|\mathbf{v}_I\|_2^2}{A_1^2\|\mathbf{w}_J\|_2^2}\leq\frac{4\|\mathbf{v}_I\|_2^2}{A_1\|\mathbf{w}_J\|_2^2}\bigg[\frac{A_1-\rho_-(s+k)\|\mathbf{v}_I\|_2^2}{A_1}\bigg]\big[A_3-\rho_-(s+k)\|\mathbf{w}_J\|_2^2\big].\tag{105}$$

By the inequality of arithmetic and geometric means, we have

$$\frac{\rho_-(s+k)\|\mathbf{v}_I\|_2^2}{A_1}\bigg[\frac{A_1-\rho_-(s+k)\|\mathbf{v}_I\|_2^2}{A_1}\bigg]\leq\frac{1}{4}.$$

Then the right-hand side of (104) can be bounded by

$$\frac{4A_2^2\|\mathbf{v}_I\|_2^2}{A_1^2\|\mathbf{w}_J\|_2^2} \le \frac{A_3-\rho_-(s+k)\|\mathbf{w}_J\|_2^2}{\rho_-(s+k)\|\mathbf{w}_J\|_2^2} \le \frac{\rho_+(k)}{\rho_-(s+k)} - 1,$$

where the last inequality follows from $A_3 \le \rho_+(k)\|\mathbf{w}_J\|_2^2$. Note that by the relationship between ℓ_2- and ℓ_∞ norm, we have $\|\mathbf{w}_J\|_2 \le \sqrt{k}\|\mathbf{w}_J\|_\infty$, which further implies that

$$\frac{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{w}_J\|\mathbf{v}_I\|_2}{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\|\mathbf{w}_J\|_\infty} \le \frac{\sqrt{k}\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{w}_J\|\mathbf{v}_I\|_2}{\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\|\mathbf{w}_J\|_2} = \frac{\sqrt{k}A_2\|\mathbf{v}_I\|_2}{A_1\|\mathbf{w}_J\|_2} \le \frac{\sqrt{k}}{2}\sqrt{\rho_+(k)/\rho_-(s+k)-1}.$$

Taking supremum over $\mathbf{v}, \mathbf{w} \in \mathbb{R}^d$ finally yields Lemma 20. ■

E.1.2. Proof of lemma 21

Proof Under Assumption 4, for any $\boldsymbol{\beta}_j \in \mathbb{R}^{d-1}$ such that $\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\|_2 \le r$ and any $\mathbf{v} \in \mathbb{R}^{d-1}$ with $\|\mathbf{v}\|_0 \le 2s^* + 2k^*$, we denote $\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)$ and $\nabla^2 L_j(\boldsymbol{\beta}_j) - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]$ as $\boldsymbol{\Lambda}_1$ and $\boldsymbol{\Lambda}_2$ respectively. Our goal is to show that both $|\mathbf{v}^T\boldsymbol{\Lambda}_1\mathbf{v}|$ and $|\mathbf{v}^T\boldsymbol{\Lambda}_2\mathbf{v}|$ are negligible. Hölder's inequality implies that $\big|\mathbf{v}^T\boldsymbol{\Lambda}_2\mathbf{v}\big| \le \|\mathbf{v}\|_1\|\boldsymbol{\Lambda}_2\mathbf{v}\|_\infty \le \|\mathbf{v}\|_1^2\|\boldsymbol{\Lambda}_2\|_\infty$. We use the following lemma to control $|\mathbf{v}^\top\boldsymbol{\Lambda}_1\mathbf{v}|$ and $\|\boldsymbol{\Lambda}_2\|_\infty$.

Lemma 26 *We denote $s^* = \max_{j\in[d]}\|\boldsymbol{\beta}_j^*\|_0$. Let $r_1(s^*,n,d) > 0$ be a real number depending on s^*, n, and d that satisfy $\lim_{n\to\infty} r_1(s^*,n,d)\log^2 d = 0$. We define $\mathbb{B}_j(r_1) \coloneqq \big\{\boldsymbol{\beta}_j \in \mathbb{R}^{d-1}\colon \big\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\big\|_1 \le r_1(s^*,n,d)\big\}$ as the ℓ_1-ball centered at $\boldsymbol{\beta}_j^*$ with radius $r_1(s^*,n,d)$. Under Assumptions 2 and 4, there exist absolute constants $C_h, C_r > 0$ such that, with probability at least $1-(2d)^{-1}$, for all $j \in [d]$, $\boldsymbol{\beta}_j \in \mathbb{B}_j(r_1)$ and $\mathbf{v} \in \mathbb{R}^d$, it holds that,*

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*) - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\big\|_\infty \le C_h\sqrt{\log d/n}, \tag{106}$$

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \le C_r r_1(s^*,n,d)\cdot\log^2 d, \tag{107}$$

$$\big|\mathbf{v}^T\big[\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)\mathbf{v}\big]\big| \le C_r r_1(s^*,n,d)\cdot\|\mathbf{v}\|_2^2. \tag{108}$$

Proof See §E.3 for a detailed proof. ■

Lemma 26 implies that $\|\boldsymbol{\Lambda}_2\|_\infty \le C_h\sqrt{\log d/n}$ with probability at least $1-(2d)^{-1}$. By the relation between ℓ_1- and ℓ_2-norms, we have

$$\big|\mathbf{v}^T\boldsymbol{\Lambda}_2\mathbf{v}\big| \le (2s^*+2k^*)\|\mathbf{v}\|_2^2\|\boldsymbol{\Lambda}\|_\infty \le (2s^*+2k^*)C_h\sqrt{\log d/n}.$$

Moreover, setting $r = C_\rho s^*\sqrt{\log d/n}$ with $C_\rho \ge 33\rho_*^{-1}$, we have

$$|\mathbf{v}^T\boldsymbol{\Lambda}_1\mathbf{v}| \le C_r C_\rho\|\mathbf{v}\|_1^2 \le C_r C_\rho(2s^*+2k^*)\sqrt{\log d/n}.$$

By Assumption 4, if n is large enough such that $(2s^*+2k^*)(C_rC_\rho+C_h)\sqrt{\log d/n}\big] \le 0.05\rho_*$, then we have

$$0.95\rho_* \le \rho_{j-}^*(2s^*+2k^*) - 0.05\rho_* \le \rho_{j-}(2s^*+2k^*) < \rho_{j+}(k^*) \le \rho_{j+}^*(k^*) + 0.05\rho_*,$$

where we denote the s-sparse eigenvalues $\rho_-\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ and $\rho_+\big(\nabla^2 L_j, \boldsymbol{\beta}_j^*; s, r\big)$ as $\rho_{j-}(s)$ and $\rho_{j+}(s)$ respectively. Under Assumption 4, $\rho_{j+}^*(k^*)/\rho_{j-}^*(2s^*+2k^*) \le 1 + 0.2k^*/s^*$ and $k^* \ge 2s^*$, simple computation yields that

$$\frac{\rho_{j+}(k^*)}{\rho_{j-}(2s^*+2k^*)} \le \frac{\rho_{j+}^*(k^*)+0.05\rho_*}{\rho_{j-}^*(2s^*+2k^*)-0.05\rho_*} \le \frac{\rho_{j+}^*(k^*)+0.05\rho_{j-}^*(2s^*+2k^*)}{0.95\rho_{j-}^*(2s^*+2k^*)} \le 1 + 0.27k^*/s^*.$$

Thus, we conclude the proof of Lemma 26. ■

E.1.3. Proof of Lemma 22

Proof For $\mathbf{v} = (v_1,\ldots,v_d)^T \in \mathbb{R}^d$, without loss of generality, we assume that $F^c = [s_1]$ where $s_1 = |F^c| \le s$. In addition, we assume that when $j > s_1$, v_j is arranged in descending order of $|v_j|$. That is, we rearrange the components of $\mathbf{v}$ such that $|v_j| \ge |v_{j+1}|$ for all $j \ge s_1$. Let $J_0 = [s_1]$ and $J_i = \{s_1+(i-1)k+1,\ldots,\min(s_1+ik,d)\}$. By definition, we have $J = J_1$ and $I = J_0 \cup J_1$. Moreover, we have $\|\mathbf{v}_{J_i}\|_\infty \le \|\mathbf{v}_{J_{i-1}}\|_1/k$ when $i \ge 2$ because by the definition of J_i, we have $\sum_{i\ge2}\|\mathbf{v}_{J_i}\|_\infty \le \|\mathbf{v}_F\|_1/k$. Note that by the definition of index sets I and J_i, $|J_i| \le k$ and $|I| = k+s_1 \le k+s$. We denote the restricted correlation coefficients $\pi(\mathbf{M},\mathbf{u}_0;s,k,r)$ as $\pi(s,k)$, then by the definition of $\pi(s+k,k)$ we have

$$\big|\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_{J_i}\big| \le \pi(s+k,k)\big[\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big]\|\mathbf{v}_{J_i}\|_\infty/\|\mathbf{v}_I\|_2.$$

Thus we have the following upper bound for $\big|\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_{I^c}\big|$:

$$\begin{aligned}\big|\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_{I^c}\big| &\le \sum_{i\ge2}\big|\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_{J_i}\big| \le \pi(s+k,k)\|\mathbf{v}_I\|_2^{-1}\big[\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big]\sum_{i\ge2}\|\mathbf{v}_{J_i}\|_\infty \\ &\le \pi(s+k,k)\|\mathbf{v}_I\|_2^{-1}\big[\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big]\|\mathbf{v}_F\|_1/k. \end{aligned} \tag{109}$$

Because $\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v} \ge \mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I + 2\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_{I^c}$, by (109) we have

$$\begin{aligned}\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v} &\ge \mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I - 2\pi(s+k,k)\|\mathbf{v}_I\|_2^{-1}\big[\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big]\|\mathbf{v}_F\|_1/k \\ &= \big[\mathbf{v}_I^T\mathbf{M}(\mathbf{u})\mathbf{v}_I\big]\big[1 - 2\pi(s+k,k)\|\mathbf{v}_I\|_2^{-1}\|\mathbf{v}_F\|_1/k\big].\end{aligned}$$

Thus we can bound the right-hand side of the last formula using the sparse eigenvalue condition

$$\mathbf{v}^T\mathbf{M}(\mathbf{u})\mathbf{v} \ge \rho_-(s+k)\big[1 - 2\pi(s+k,k)k^{-1}\|\mathbf{v}_I\|_2^{-1}\|\mathbf{v}_F\|_1\big]\|\mathbf{v}_I\|_2^2, \tag{110}$$

where we denote s-sparse eigenvalue $\rho_-(\mathbf{M},\mathbf{u}_0;s,r)$ as $\rho_-(s+k)$ for the simplicity of notations. Inequality (110) concludes the proof of Lemma 22. ■

E.1.4. Proof of Lemma 23

Proof Let $F(t) = L_j\big(\boldsymbol{\beta}(t)\big) - L_j(\boldsymbol{\beta}_1) - \big\langle\nabla L_j(\boldsymbol{\beta}_1), \boldsymbol{\beta}(t) - \boldsymbol{\beta}_1\big\rangle$. Since the derivative of $L_j\big(\boldsymbol{\beta}(t)\big)$ with respect to t is $\big\langle\nabla L_j\big(\boldsymbol{\beta}(t)\big), \boldsymbol{\beta}_2 - \boldsymbol{\beta}_1\big\rangle$, the derivative of F is given by

$$F'(t) = \big\langle\nabla L_j\big(\boldsymbol{\beta}(t)\big) - \nabla L_j(\boldsymbol{\beta}_1), \boldsymbol{\beta}_2 - \boldsymbol{\beta}_1\big\rangle.$$

Therefore the Bregman divergence $D_j\big(\boldsymbol{\beta}(t),\boldsymbol{\beta}_1\big)$ can be written as

$$D_j\big(\boldsymbol{\beta}(t),\boldsymbol{\beta}_1\big)=\big\langle\nabla L_j[\boldsymbol{\beta}(t)]-\nabla L_j(\boldsymbol{\beta}_1),t(\boldsymbol{\beta}_2-\boldsymbol{\beta}_1)\big\rangle=tF'(t).$$

By definition, it is easy to see that $F'(1)=D_j(\boldsymbol{\beta}_2,\boldsymbol{\beta}_1)$. To derive Lemma 23, it suffices to show that $F(t)$ is convex, which implies that $F'(t)$ is non-decreasing and $D_j\big(\boldsymbol{\beta}(t),\boldsymbol{\beta}_1\big)=tF'(t)\le tF'(1)=tD_j(\boldsymbol{\beta}_2,\boldsymbol{\beta}_1)$.

For $\forall t_1,t_2\in\mathbb{R}_+,t_1+t_2=1,x,y\in(0,1)$, by the linearity of $\boldsymbol{\beta}(t)$, $\boldsymbol{\beta}(t_1x+t_2y)=t_1\boldsymbol{\beta}(x)+t_2\boldsymbol{\beta}(y)$. Then we have

$$\big\langle\nabla L_j(\boldsymbol{\beta}_1),\boldsymbol{\beta}(t_1x+t_2y)-\boldsymbol{\beta}_1\big\rangle=t_1\big\langle\nabla L_j(\boldsymbol{\beta}_1),\boldsymbol{\beta}(x)-\boldsymbol{\beta}_1\big\rangle+t_2\big\langle\nabla L_j(\boldsymbol{\beta}_1),\boldsymbol{\beta}(y)-\boldsymbol{\beta}_1\big\rangle.\quad(111)$$

In addition, by convexity of function $L_j(\cdot)$, we obtain

$$L_j\big(\boldsymbol{\beta}(t_1x+t_2y)\big)\le t_1L_j\big(\boldsymbol{\beta}(x)\big)+t_2L_j\big(\boldsymbol{\beta}(y)\big).\quad(112)$$

Adding (111) and (112) we obtain

$$F\big(t_1x+t_2y\big)\le t_1F(x)+t_2F(y).$$

Therefore $F(t)$ is convex, thus we have $D_j(\boldsymbol{\beta}(t),\boldsymbol{\beta}_1)\le tD_j(\boldsymbol{\beta}_2,\boldsymbol{\beta}_1)$. ■

E.2. Proof of Technical Lemmas in §D

Now we prove the lemmas that supports the auxiliary inferential results. We first prove Lemma 25, which implies that the $\widehat{\sigma}_{jk}^2$ is a consistent estimator of the asymptotic variance of σ_{jk}.

E.2.1. Proof of lemma 25

Proof Recall that we denote $\boldsymbol{\beta}_{j\vee k}=(\beta_{jk},\boldsymbol{\beta}_{j\setminus k},\boldsymbol{\beta}_{k\setminus j})$ and $L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)=L_j(\boldsymbol{\beta}_j)+L_k(\boldsymbol{\beta}_k)$. We denote the kernel function of the second-order U-statistic $\nabla L_{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ as $\mathbf{h}_{ii'}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)$ where the subscripts i,i' indicate that $\mathbf{h}_{ii'}^{jk}(\cdot)$ depends on $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i'}$. We define $\mathbf{V}_{ii'i''}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big):=\mathbf{h}_{ii'}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)\mathbf{h}_{ii'}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)^T$. Then by definition, $\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\setminus k}\big)$can be written as

$$\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)=\frac{1}{n(n-1)^2}\sum_{i=1}^{n}\sum_{i'\neq i,i''\neq i}\mathbf{V}_{ii'i''}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big).$$

Note that $\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)-\boldsymbol{\Sigma}^{jk}=\underbrace{\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}\big)-\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)}_{I_1}+\underbrace{\widehat{\boldsymbol{\Sigma}}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)-\boldsymbol{\Sigma}^{jk}}_{I_2}$.

We first consider I_2. For notational simplicity, we use $\mathbf{h}_{ii'}$ and $\mathbf{h}_{ii'|i}$ to denote $\mathbf{h}_{ij}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)$ and $\mathbf{h}_{ii'|i}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big):=\mathbb{E}\big[\mathbf{h}_{ij}^{jk}\big(\boldsymbol{\beta}_{j\vee k}^*\big)\big|\boldsymbol{X}_i\big]$ respectively. As shown in §D.1, for $i\neq i'\neq i''$,

$$\mathbb{E}\big(\mathbf{h}_{ii'}\mathbf{h}_{ii''}^T\big)=\mathbb{E}\big(\mathbf{h}_{ii'}\mathbf{h}_{ii''}^T\big|\boldsymbol{X}_i\big)=\mathbb{E}\big(\mathbf{h}_{ii'|i}\mathbf{h}_{ii''|i}^T\big)=\boldsymbol{\Sigma}^{jk}\ \text{ and }\ \mathbb{E}\big(\mathbf{h}_{ij}\mathbf{h}_{ij}^T\big)=\boldsymbol{\Theta}^{jk},$$

we can write I_2 as

$$I_2=\frac{n-2}{n-1}\underbrace{\left\{\binom{n}{3}^{-1}\sum_{i<i'<i''}\big[\mathbf{V}_{ii'i''}-\mathbb{E}(\mathbf{V}_{ii'i''})\big]\right\}}_{I_{21}}+\frac{1}{n-1}\underbrace{\left\{\binom{n}{2}^{-1}\sum_{i<i'}\big[\mathbf{V}_{ii'i'}-\mathbb{E}(\mathbf{V}_{ii'i'})\big]\right\}}_{I_{22}}+\frac{1}{n-1}\big(\boldsymbol{\Theta}^{jk}-\boldsymbol{\Sigma}^{jk}\big),$$

where we use $\mathbf{V}_{ii'i''}$ to denote $\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)$. Observing that I_{21} is a centered third order U-statistic, for x large enough such that $x^4\ge\big\|\mathbb{E}\big[\mathbf{V}_{ijk}(\boldsymbol{\beta}_{j\vee k}^*)\big]\big\|_\infty$ and for any $(a,b),(c,d)\in\big\{(p,q)\colon p,q\in\{j,k\}\big\}$ we have

$$\begin{aligned}\mathbb{P}\big(\big[\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k})\big]_{ab,cd}>2x^4\big)&\le\mathbb{P}\big[(X_{ia}-X_{i'a})(X_{ib}-X_{i'b})(X_{ic}-X_{i''c})(X_{id}-X_{i''d})>x^4\big]\\&\le 8\exp(2\kappa_m+\kappa_h)\exp(-x).\end{aligned}$$

Thus there exist constants c_1 and C_1 that does not depend on n or d or (j,k) such that for any $x\in\mathbb{R}$, any $i,i',i''\in[n]$ and any $j,k\in[d]$,

$$\mathbb{P}\big([\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)]_{ab,cd}>x\big)\le C_1\exp(c_1x^{1/4}).\quad(113)$$

This implies that there exists some generic constant C such that $\|\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)\|_\infty\le C\log^4 d$ for all $j,k\in[d]$ and $i,i'\in[n]$ with probability tending to one. Similar to the method we use in §E.3, we define $\mathcal{E}:=\big\{\|\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)\|_\infty\le C\log^4 d,\forall i,i',i''\in[n],j,k\in[d]\big\}$. By Bernstein's inequality for U-statistics (Lemma 19) with $b=C\log^4 d$ in (56), for some generic constants C, it holds with high probability that

$$\binom{n}{2}^{-1}\sum_{i<i'}\big[\mathbf{V}_{ii'i'}-\mathbb{E}(\mathbf{V}_{ii'i'}|\mathcal{E})\big]\le C\sqrt{\log d/n},\quad\forall j,k\in[d],i,i',i''\in[n].\quad(114)$$

Moreover, by (113), we have

$$\begin{aligned}&\mathbb{E}\big\{[\mathbf{V}_{ii'i'}(\boldsymbol{\beta}_{j\vee k}^*)]_{ab,cd}|\mathcal{E}\big\}-\mathbb{E}\big\{[V_{ii'i''}(\boldsymbol{\beta}_{j\vee k}^*)]_{ab,cd}\big\}\\&\le\int_{C\log^4 d}^{\infty}\mathbb{P}\big\{\big|[\mathbf{V}_{ii'i''}^{jk}(\boldsymbol{\beta}_{j\vee k}^*)]_{ab,cd}\big|>x\big\}\le c_1\log^3 d\cdot\exp(-c_2\log d)\end{aligned}\quad(115)$$

for some absolute constant c_1 and c_2. Since (115) holds uniformly, we have

$$\binom{n}{2}^{-1}\sum_{i<i'}\big[\mathbb{E}(\mathbf{V}_{ii'i'}|\mathcal{E})-\mathbb{E}(\mathbf{V}_{ii'i''})\big]\le\log^3 d\cdot\exp(-c_2\log d)\lesssim\sqrt{\log d/n}.\quad(116)$$

Combining (114) and (116) we obtain that

$$\|I_{21}\|_\infty=\mathcal{O}_{\mathbb{P}}\big(\sqrt{\log d/n}\big)\ \text{ uniformly for } 1\le j<k\le n.\quad(117)$$

For the second part I_{22}, noting that it is a U-statistic of order 2, because (113) also holds for $\mathbf{V}_{ii'i''}(\boldsymbol{\beta}_{j\vee k}^*)$, applying the same technique, we have $\|I_{21}\|_\infty=\mathcal{O}_{\mathbb{P}}\big(\sqrt{\log d/n}\big)$ uniformly

for $1 \le j < k \le n$. Combining with (117), we conclude that, for some absolute constant C, we have

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k}) - \boldsymbol{\Sigma}^{jk}\big\|_\infty \le C\sqrt{\log d/n}, \quad \forall 1 \le j < k \le n. \tag{118}$$

Now we turn to I_1. For any $\boldsymbol{\beta}_j, \boldsymbol{\beta}_k \in \mathbb{R}^{d-1}$ such that $\|\boldsymbol{\beta}_j - \boldsymbol{\beta}^*_j\|_1 \le r(s^*, n, d)$ and $\|\boldsymbol{\beta}_k - \boldsymbol{\beta}^*_k\|_1 \le r(s^*, n, d)$, we denote $\omega^j_{ii'} := \exp\big[-(X_{ij} - X_{i'j})(\boldsymbol{\beta}_j - \boldsymbol{\beta}^*_j)^T(\boldsymbol{X}_{i\setminus j} - \boldsymbol{X}_{i'\setminus j})\big]$ and denote $\omega^k_{ii'}$ similarly. Recall that we denote $R^j_{ii'}(\boldsymbol{\beta}_j) = \exp\big[-(x_{ij} - x_{i'j})\boldsymbol{\beta}_j{}^T(\boldsymbol{x}_{i\setminus j} - \boldsymbol{x}_{i'\setminus j})\big]$. Hence by definition we have $R^j_{ii'}(\boldsymbol{\beta}_j) = \omega^j_{ii'}R^j_{ii'}(\boldsymbol{\beta}^*_j)$. As shown in §E.3, we have

$$\min\{1, \omega^j_{ii'}, \omega^k_{ii'}\}\mathbf{h}^{jk}_{ii'}(\boldsymbol{\beta}^*_{j\vee k}) \le \mathbf{h}^{jk}_{ii'}(\boldsymbol{\beta}_{j\vee k}) \le \max\{1, \omega^j_{ii'}, \omega^k_{ii'}\}\mathbf{h}^{jk}_{ii'}(\boldsymbol{\beta}^*_{j\vee k}), \tag{119}$$

where the inequality is taken elementwisely. We denote $b := \max_{i,i'\in[n];j\in[d]} r(s^*, n, d)\big\|(X_{ij} - X_{i'j})(\boldsymbol{X}_{i\setminus j} - \boldsymbol{X}_{i'\setminus j})\big\|_\infty$. Note that when $\|\boldsymbol{\beta}_j - \boldsymbol{\beta}^*_j\|_1 \le r(s^*, n, d)$ and $\|\boldsymbol{\beta}_k - \boldsymbol{\beta}^*_k\|_1 \le r(s^*, n, d)$, we have $\omega^j_{ii'}, \omega^k_{ii'} \in [\exp(-b), \exp(b)]$. Therefore by (119) and the definition of $V^{jk}_{ii'i''}(\boldsymbol{\beta}_{j\setminus k})$, we obtain the following elementwise inequality

$$\exp(-2b)\mathbf{V}^{jk}_{ii'i''}(\boldsymbol{\beta}^*_{j\setminus k}) \le \mathbf{V}^{jk}_{ii'i''}(\boldsymbol{\beta}_{j\setminus k}) \le \exp(2b)\mathbf{V}^{jk}_{ii'i''}(\boldsymbol{\beta}^*_{j\setminus k}),$$

which implies that

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k}) - \widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k})\big\|_\infty \le \max\big\{1 - \exp(-2b), \exp(2b) - 1\big\}\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k})\big\|_\infty. \tag{120}$$

As we show in §E.3, $b \le Cr(s^*, n, d)\log^2 d$ with high probability for some absolute constant $C > 0$. Since $\lim_{n\to\infty} r(s^*, n, d)\log^2 d = 0$, by (120) we have

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k}) - \widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k})\big\|_\infty \lesssim b\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k})\big\|_\infty\big\|_\infty \le b\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k}) - \boldsymbol{\Sigma}^{jk}\big\|_\infty + b\|\boldsymbol{\Sigma}^{jk}\|_\infty.$$

Note that we show $\|I_2\|_\infty = \big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k}) - \boldsymbol{\Sigma}^{jk}\big\|_\infty = \mathcal{O}_{\mathbb{P}}\big(\sqrt{\log d/n}\big)$, which converges to zero asymptotically. Thus we conclude that

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k}) - \widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}^*_{j\vee k})\big\|_\infty = \mathcal{O}_{\mathbb{P}}\big(r(s^*, n, d)\log^2 d\big). \tag{121}$$

Combining (118) and (121), we have the following error bound for $\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k})$:

$$\big\|\widehat{\boldsymbol{\Sigma}}^{jk}(\boldsymbol{\beta}_{j\vee k}) - \boldsymbol{\Sigma}^{jk}\big\|_\infty = \mathcal{O}_{\mathbb{P}}\Big(r(s^*, n, d)\log^2 d + \sqrt{\log d/n}\Big) \quad \text{for all } (j, k). \tag{122}$$

Finally, by the fact that $\max_{j\in[d]}\|\widehat{\boldsymbol{\beta}}_j - \boldsymbol{\beta}^*_j\|_1 \lesssim s^*\lambda$, we conclude the proof of Lemma 25 by setting $r = Cs^*\lambda$. ∎

E.3. Proof of Lemma 26

Now we turn to the last unproven result, namely Lemma 26, which characterizes the perturbation of $\nabla^2 L_j(\boldsymbol{\beta}_j)$.

Proof Note that $\nabla^2 L_j(\boldsymbol{\beta}_j)$ is a second-order U-statistic. Hence $\nabla^2 L_j(\boldsymbol{\beta}_j) - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]$ is a centered U-statistic. We denote its kernel as $\mathbf{T}_{ii'}(\boldsymbol{\beta}_j)$, then

$$\nabla^2 L_j(\boldsymbol{\beta}_j) - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big] = \frac{2}{n(n-1)}\sum_{i<i'}\mathbf{T}_{ii'}(\boldsymbol{\beta}_j).$$

Note that $\big\|\mathbb{E}[\mathbf{T}_{ii'}(\boldsymbol{\beta}_j)]\big\|_\infty$ is bounded for all $\boldsymbol{\beta}_j \in \mathbb{R}^{d-1}$ because

$$\max_{\mathbf{u}\in\mathbb{R}^{d-1}}\big\|\mathbb{E}[T_{ii'}(\boldsymbol{\beta}_j)]\big\|_\infty \lesssim \max_{j\in[d]}\mathbb{E}|X_{ij} - X_{i'j}|^4 \lesssim \max_{j\in[d],i\in[n]}\mathbb{E}|X_{ij}|^4 \le \int_0^\infty c\exp(-t^{1/4})dt = 24c,$$

where $c = 2\exp(\kappa_m + \kappa_h/2)$. Here the last inequality follows from (14). Let $\nabla^2_{jk,j\ell}L_j(\boldsymbol{\beta}_j) = \partial^2 L_j(\boldsymbol{\beta}_j)/\big(\partial\beta_{jk}\partial\beta_{j\ell}\big)$ and let $\big[\boldsymbol{T}_{ii'}(\boldsymbol{\beta}_j)\big]_{k\ell}$ be the corresponding kernel function. That is, $\nabla^2_{jk,j\ell}L_j(\boldsymbol{\beta}_j) = \binom{n}{2}^{-1}\sum_{i<i'}\big[\boldsymbol{T}_{ii'}(\boldsymbol{\beta}_j)\big]_{k\ell}$. For $x > 0$ such that $x^4 > 24c$ and $k, \ell \ne j$, we have

$$\begin{aligned}\mathbb{P}\big\{\big|[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big| > 2x^4\big\} &\le \mathbb{P}\big[(X_{ij} - X_{i'j})^2(X_{ik} - X_{i'k})(X_{i\ell} - X_{i'\ell}) > x^4\big] \\ &\le \mathbb{P}\big(|X_{ij} - X_{i'j}| > x\big) + \mathbb{P}\big(|X_{ik} - X_{i'k}| > x\big) + \mathbb{P}\big(|X_{i\ell} - X_{i'\ell}| > x\big).\end{aligned} \tag{123}$$

As a direct implication of Assumption 2, we have $\mathbb{P}\big(|X_{ij} - X_{ij}| > x\big) \le 2\exp(2\kappa_m + \kappa_k)\exp(-x)$ for all $j \in [d]$. Then we can bound the right-hand side of (123) by

$$\mathbb{P}\big\{\big|[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big| > 2x^4\big\} \le 6\exp(2\kappa_m + \kappa_h)\exp(-x) \quad \text{when } x^4 > 48\exp(\kappa_m + \kappa_h/2).$$

Letting $C_T = \max\Big\{6\exp(2\kappa_m + \kappa_h), \exp\big\{[48\exp(\kappa_m + \kappa_h/2)]^{1/4}\big\}\Big\}$, it holds that

$$\mathbb{P}\big\{\big|[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big| > x\big\} \le C_T\exp(-2^{-1/4}x^{1/4}) \quad \text{for all } x > 0. \tag{124}$$

Thus by a union bound, we conclude that there exists some generic constant C such that $\|\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)\|_\infty \le C\log^4 d$ for all $j \in [d]$ and $i, i' \in [n]$ with probability at least $1 - (8d)^{-1}$. We define an event $\mathcal{E} := \big\{\|\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)\|_\infty \le C\log^4 d, \forall i, i' \in [n], j \in [d]\big\}$. By (124), it is easy to see that $\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)$ is ℓ_2-integrable. By Bernstein's inequality for U-statistics (Lemma 19) with $b = C\log^4 d$ in (56), for some generic constants C_1 and C_2, we obtain that

$$\mathbb{P}\Big(\nabla^2 L_j(\boldsymbol{\beta}_j) - \mathbb{E}_1\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big] > t\big|\mathcal{E}\Big) \le 4\exp\big[-nt^2/(C_1 + C_2\log^4 d\cdot t)\big], \quad \forall j \in [d]. \tag{125}$$

Here we use $\mathbb{E}_1\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]$ to denote $\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big|\mathcal{E}\big]$. Thus under Assumption 4 we obtain that, conditioning on event $\mathcal{E}$,

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j) - \mathbb{E}_1\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\big\|_\infty \le C\sqrt{\log d/n}, \quad \forall j \in [d] \tag{126}$$

with probability at least $1 - (8d)^{-1}$. Moreover, by (124) we obtain that

$$\mathbb{E}\big\{[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big|\mathcal{E}\big\} - \mathbb{E}\big\{[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big\} \le \int_{C\log^4 d}^\infty \mathbb{P}\big\{\big|[\mathbf{T}_{ii'}(\boldsymbol{\beta}^*_j)]_{k\ell}\big| > x\big\} \le c_1\log^3 d\cdot\exp(-c_2\log d)$$

for some absolute constant c_1 and c_2. Therefore we have

$$\big\|\mathbb{E}_1\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big] - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j)\big]\big\|_\infty \lesssim \log^3 d\cdot\exp(-c_2\log d) \lesssim \sqrt{\log d/n}. \tag{127}$$

Combining (126) and (127) we show that, with probability at least $1-(4d)^{-1}$, $\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*) - \mathbb{E}[\nabla^2 L_j(\boldsymbol{\beta}_j^*)]\big\|_\infty \le C_h\sqrt{\log d/n}$ for all $j \in [d]$.

For the second argument (107), let $\boldsymbol{\Delta} = \boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*$ where $\boldsymbol{\beta}_j \in \mathbb{R}^{d-1}$ lies in the ℓ_1-ball centered at $\boldsymbol{\beta}_j^*$ with radius $r_1(s^*,n,d)$, that is, $\big\|\boldsymbol{\beta}_j - \boldsymbol{\beta}_j^*\big\|_1 \le r_1(s^*,n,d)$. By the independence between $\boldsymbol{X}_i$ and $\boldsymbol{X}_{i'}$, Assumption 2 implies that

$$\max\Big\{\log\mathbb{E}\big[\exp(X_{ij}-X_{i'j})\big], \log\mathbb{E}\big[\exp(X_{i'j}-X_{ij})\big]\Big\} \le 2\kappa_m + \kappa_h,$$

which further implies that for any $x > 0$

$$\mathbb{P}\Big(\big|(X_{ij}-X_{i'j})\big| > x\Big) \le 2\exp(2\kappa_m+\kappa_h)\exp(-x), \quad \forall j \in [d].$$

Hence for any $x > 0$ and $j,k \in [d]$, a union bound implies that

$$\begin{aligned}\mathbb{P}\big[\big|(X_{ij}-X_{i'j})(X_{ik}-X_{i'k})\big| > x^2\big] &\le \mathbb{P}\big[\big|(X_{ij}-X_{i'j})\big| > x\big] + \mathbb{P}\big[\big|(X_{ik}-X_{i'k})\big| > x\big] \\ &\le 4\exp(2\kappa_m+\kappa_h)\exp(-x). \end{aligned} \tag{128}$$

Taking a union bound over $1 \le j < k \le d$ and $1 \le i < i' \le n$ we obtain that

$$\mathbb{P}\Big[\max_{i,i'\in[n];j\in[d]} \big\|(X_{ij}-X_{i'j})(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})\big\|_\infty > x^2\Big] \lesssim n^2d^2\exp(-x).$$

If we denote $b := \max_{i,i'\in[n];j\in[d]} r_1(s^*,n,d)\big\|(X_{ij}-X_{i'j})(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})\big\|_\infty$, then we obtain that $b \le Cr_1(s^*,n,d)\log^2 d$ with probability at least $1-(4d)^{-1}$ for some constant $C > 0$. Denoting $\omega_{ii'} := \exp\big\{-(X_{ij}-X_{i'j})\boldsymbol{\Delta}^T(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})\big\}$, by definition,

$$R_{ii'}^j(\boldsymbol{\beta}_j) = \exp\big\{-(X_{ij}-X_{i'j})(\boldsymbol{\Delta}+\boldsymbol{\beta}_j^*)^T(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})\big\} = \omega_{ii'}R_{ii'}^j(\boldsymbol{\beta}_j^*).$$

Thus we can write $\nabla^2 L_j(\boldsymbol{\beta}_j)$ as:

$$\nabla^2 L_j(\boldsymbol{\beta}_j) = \frac{2}{n(n-1)}\sum_{i<i'}\frac{R_{ii'}^j(\boldsymbol{\beta}^*)(X_{ij}-X_{i'j})^2(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})^{\otimes 2}}{\big(1+R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2}\frac{\omega_{ii'}\big(1+R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2}{\big(1+\omega_{ii'}R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2}. \tag{129}$$

If $\omega_{ii'} \ge 1$, then $(\omega_{ii'})^{-2} \le \big(1+R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2/\big(1+\omega_{ii'}R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2 \le 1$; otherwise we have $1 \le \big(1+R_{ii'}^j(\boldsymbol{\beta})\big)^2/\big(1+\omega_{ii'}R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2 \le (\omega_{ii'})^{-2}$. This observation implies

$$\min\big\{\omega_{ii'}, 1/\omega_{ii'}\big\} \le \frac{\omega_{ii'}\big(1+R_{ii'}^j(\boldsymbol{\beta})\big)^2}{\big(1+\omega_{ii'}R_{ii'}^j(\boldsymbol{\beta}^*)\big)^2} \le \max\big\{\omega_{ii'}, 1/\omega_{ii'}\big\}. \tag{130}$$

By the definition of $\omega_{ii'}$, Hölder's inequality implies that $\big|(X_{ij}-X_{i'j})\boldsymbol{\Delta}^T(\boldsymbol{X}_{i\setminus j}-\boldsymbol{X}_{i'\setminus j})\big| \le b$, thus we have

$$\exp(-b) \le \min\big\{\omega_{ii'}1/\omega_{ii'}\big\} \le \max\big\{\omega_{ii'}, 1/\omega_{ii'}\big\} \le \exp(b). \tag{131}$$

Combining (129),(130) and (131) we obtain

$$\exp(-b)\nabla^2 L_j(\boldsymbol{\beta}_j^*) \le \nabla^2 L_j(\boldsymbol{\beta}_j) \le \exp(b)\nabla^2 L_j(\boldsymbol{\beta}_j^*). \tag{132}$$

Then by (132), since $\lim_{n\to\infty} r_1(s^*,n,d)\log^2 d = 0$, we have

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \le \max\big\{1-\exp(-b), \exp(b)-1\big\}\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \lesssim b\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty.$$

Notice that under Assumption 2, as shown in §D.1, we can assume that $\big\|\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\big\|_\infty \le D$ where D appears in (84). By triangle inequality,

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \le \big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*) - \mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\big\|_\infty + \big\|\mathbb{E}\big[\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\big\|_\infty \le D + C_h\sqrt{\log d/n} \le 2D$$

with probability at least $1-(4d)^{-1}$, where the last inequality follows from the fact that $\big(\log^9 d/n\big)^{1/2}$ tends to zero as n goes to infinity. Then we obtain that

$$\big\|\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty \le C_r r_1(s^*,n,d)\log^2 d$$

holds for some absolute constant $C_r > 0$ and uniformly for all $j \in [d]$ and $\boldsymbol{\beta}_j \in \mathbb{B}_j(r_1)$ with probability at least $1-(2d)^{-1}$.

Finally, for the last argument (108), for any $\mathbf{v} \in \mathbb{R}^{d-1}$, by (132) we have

$$\exp(-b)\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j^*)\mathbf{v} \le \mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j)\mathbf{v} \le \exp(b)\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j^*)\mathbf{v}.$$

Thus we have

$$\big|\mathbf{v}^T\big[\nabla^2 L_j(\boldsymbol{\beta}_j) - \nabla^2 L_j(\boldsymbol{\beta}_j^*)\big]\mathbf{v}\big| \lesssim b\big|\mathbf{v}^T\nabla^2 L_j(\boldsymbol{\beta}_j^*)\mathbf{v}\big| \le b\|\mathbf{v}\|_1^2\big\|\nabla^2 L_j(\boldsymbol{\beta}_j^*)\big\|_\infty,$$

which implies (108). ∎

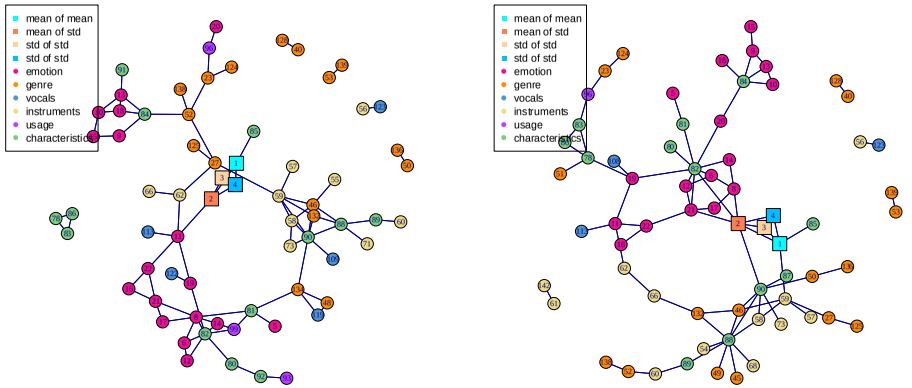
References

Genevera I Allen and Zhandong Liu. A log-linear graphical model for inferring genetic networks from high-throughput sequencing data. In *IEEE International Conference on Bioinformatics and Biomedicine*, 2012.

Miguel A Arcones. A Bernstein-type inequality for U-statistics and U-processes. *Statistics & probability letters*, 22(3):239–247, 1995.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *The Journal of Machine Learning Research*, 9:485–516, 2008.

Alexandre Belloni, Daniel Chen, Victor Chernozhukov, and Christian Hansen. Sparse models and methods for optimal instruments with an application to eminent domain. *Econometrica*, 80(6):2369–2429, 2012.



(a). The asymmetric score test based on $L_k(\boldsymbol{\beta}_k)$. (b). Inconsistent edges of the asymmetric score test

Figure 4: In (a) we plot estimated graph in the `CAL500` dataset inferred by the asymmetric score test based on the loss function $L_k(\boldsymbol{\beta}_k)$ for testing $H_0 \colon \beta^*_{jk} = 0$ for any $1 \le j < k \le d$. We plot the connected components of the estimated graph for illustration. Compared with Figure 3, we observe that the two asymmetric score tests yields different graphs. In (b) we plot the edges that appear in (a) and Figure 3-(a) but not in Figure 3-(b). In other words, we plot the inconsistent edges of these two asymmetric score tests that are discovered by the pairwise score test. Thus, by taking symmetry into consideration, the pairwise score test is able to correct such inconsistency.

Alexandre Belloni, Victor Chernozhukov, and Ying Wei. Honest confidence regions for a regression parameter in logistic regression with a large number of controls. *arXiv:1304.3969*, 2013.

Julian Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 36(2):192–236, 1974.

Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of LASSO and Dantzig selector. *The Annals of Statistics*, 37:1705–1732, 2009.

Jelena Bradic, Jianqing Fan, and Weiwei Wang. Penalized composite quasi-likelihood for ultrahigh dimensional variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(3):325–349, 2011.

Tony Cai, Weidong Liu, and Xi Luo. A constrained ℓ_1-minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association*, 106(494): 594–607, 2011.

Emmanuel Candés, Terence Tao, et al. The Dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6):2313–2351, 2007.

Emmanuel J Candés and Terence Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

Kwun Chuen Gary Chan. Nuisance parameter elimination for proportional likelihood ratio models with nonignorable missingness and random truncation. *Biometrika*, 100(1):269–276, 2012.

Shizhe Chen, Daniela M Witten, and Ali Shojaie. Selection and estimation for mixed graphical models. *Biometrika*, 102(1):47–64, 2015.

Jie Cheng, Elizaveta Levina, Pei Wang, and Ji Zhu. A sparse Ising model with covariates. *Biometrics*, 70(4):943–953, 2014.

Jie Cheng, Tianxi Li, Elizaveta Levina, and Ji Zhu. High-dimensional mixed graphical models. *Journal of Computational and Graphical Statistics*, 26(2):367–378, 2017.

Patrick Danaher, Pei Wang, and Daniela M Witten. The joint graphical LASSO for inverse covariance estimation across multiple classes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(2):373–397, 2014.

Guoqing Diao, Jing Ning, et al. Maximum likelihood estimation for semiparametric density ratio model. *The International Journal of Biostatistics*, 8(1):1–29, 2012.

Mathias Drton and Michael D Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008.

Mathias Drton, Michael D Perlman, et al. Multiple testing and error control in Gaussian graphical model selection. *Statistical Science*, 22(3):430–449, 2007.

David Edwards. *Introduction to Graphical Modelling*. Springer, 2000.

Sasha Epskamp. *Sampling Methods and Distribution Functions for the Ising Model*, 2015. R package.

Jianqing Fan and Jinchi Lv. Nonconcave penalized likelihood with NP-dimensionality. *Information Theory, IEEE Transactions on*, 57(8):5467–5484, 2011.

Jianqing Fan, Lingzhou Xue, Hui Zou, et al. Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics*, 42(3):819–849, 2014.

Jianqing Fan, Han Liu, Yang Ning, and Hui Zou. High dimensional semiparametric latent graphical model for mixed data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 79(2):405–421, 2017.

Jianqing Fan, Han Liu, Qiang Sun, and Tong Zhang. I-LAMM for sparse learning: Simultaneous control of algorithmic complexity and statistical error. *Annals of statistics*, 46 (2):814–841, 2018.

Bernd Fellinghauer, Peter Bühlmann, Martin Ryffel, Michael Von Rhein, and Jan D Reinhardt. Stable graphical model estimation with random forests for discrete, continuous, and mixed variables. *Computational Statistics & Data Analysis*, 64:132–152, 2013.

Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical LASSO. *Biostatistics*, 9(3):432–441, 2008.

Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Applications of the LASSO and grouped LASSO to the estimation of sparse graphical models. Technical report, 2010.

J Ge, X Li, H Jiang, H Liu, T Zhang, M Wang, and T Zhao. *Picasso: A Sparse Learning Library for High Dimensional Data Analysis in R and Python*, 2017. R package.

Quanquan Gu, Yuan Cao, Yang Ning, and Han Liu. Local and global inference for high dimensional gaussian copula graphical models. *arXiv preprint arXiv:1502.02347*, 2015.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint estimation of multiple graphical models. *Biometrika*, 98(1):1–15, 2011.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Graphical models for ordinal data. *Journal of Computational and Graphical Statistics*, 24(1):183–204, 2015.

Holger Höfling and Robert Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *The Journal of Machine Learning Research*, 10:883–906, 2009.

Jana Janková and Sara van de Geer. Confidence intervals for high-dimensional inverse covariance estimation. *Electronic Journal of Statistics*, 9(1):1205–1229, 2015. doi: 10.1214/15-EJS1031.

Adel Javanmard and Andrea Montanari. Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research*, 15(1):2869–2909, 2014.

Clifford Lam and Jianqing Fan. Sparsistency and rates of convergence in large covariance matrix estimation. *Annals of statistics*, 37(6B):4254–4278, 2009.

Steffen L Lauritzen. *Graphical Models*. Oxford University Press, 1996.

Jason D Lee and Trevor J Hastie. Learning the structure of mixed graphical models. *Journal of Computational and Graphical Statistics*, 24(1):230–253, 2015.

Jason D Lee, Dennis L Sun, Yuekai Sun, Jonathan E Taylor, et al. Exact post-selection inference, with application to the LASSO. *The Annals of Statistics*, 44(3):907–927, 2016.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient structure learning of Markov networks using ℓ_1-regularization. In *Advances in neural Information processing systems*, 2006.

Kung-Yee Liang and Jing Qin. Regression analysis under non-standard situations: a pairwise pseudolikelihood approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 62(4):773–786, 2000.

Han Liu, John Lafferty, and Larry Wasserman. The Nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *The Journal of Machine Learning Research*, 10:2295–2328, 2009.

Han Liu, Fang Han, Ming Yuan, John Lafferty, Larry Wasserman, et al. High-dimensional semiparametric Gaussian copula graphical models. *The Annals of Statistics*, 40(4):2293–2326, 2012.

Weidong Liu et al. Gaussian graphical model estimation with false discovery rate control. *The Annals of Statistics*, 41(6):2948–2978, 2013.

Richard Lockhart, Jonathan Taylor, Ryan J Tibshirani, Robert Tibshirani, et al. A significance test for the LASSO. *The Annals of Statistics*, 42(2):413–468, 2014.

Po-Ling Loh and Martin J Wainwright. Regularized M-estimators with nonconvexity: Statistical and algorithmic theory for local optima. *Journal of Machine Learning Research*, 16:559–616, 2015.

Nicolai Meinshausen and Peter Bühlmann. High-dimensional graphs and variable selection with the LASSO. *The Annals of Statistics*, 34(3):1436–1462, 2006.

Karthik Mohan, Palma London, Maryam Fazel, Daniela Witten, and Su-In Lee. Node-based learning of multiple Gaussian graphical models. *The Journal of Machine Learning Research*, 15(1):445–488, 2014.

Sahand N. Negahban, Pradeep Ravikumar, Martin J. Wainwright, and Bin Yu. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. *Statistical Science*, 27(4):538–557, 11 2012.

Matey Neykov, Yang Ning, Jun S Liu, Han Liu, et al. A unified theory of confidence regions and testing for high-dimensional estimating equations. *Statistical Science*, 33(3):427–443, 2018.

Yang Ning and Han Liu. High-dimensional semiparametric bigraphical models. *Biometrika*, 100(3):655–670, 2013.

Yang Ning, Han Liu, et al. A general theory of hypothesis tests and confidence regions for sparse high dimensional models. *The Annals of Statistics*, 45(1):158–195, 2017a.

Yang Ning, Tianqi Zhao, Han Liu, et al. A likelihood ratio framework for high-dimensional semiparametric regression. *The Annals of Statistics*, 45(6):2299–2327, 2017b.

Jie Peng, Pei Wang, Nengfeng Zhou, and Ji Zhu. Partial correlation estimation by joint sparse regression models. *Journal of the American Statistical Association*, 104(486):735–746, 2009.

Garvesh Raskutti, Martin J Wainwright, and Bin Yu. Restricted eigenvalue properties for correlated Gaussian designs. *The Journal of Machine Learning Research*, 11:2241–2259, 2010.

Pradeep Ravikumar, Martin J Wainwright, John D Lafferty, et al. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 38(3): 1287–1319, 2010.

Pradeep Ravikumar, Martin J Wainwright, Garvesh Raskutti, Bin Yu, et al. High-dimensional covariance estimation by minimizing ℓ_1-penalized log-determinant divergence. *Electronic Journal of Statistics*, 5:935–980, 2011.

Zhao Ren, Tingni Sun, Cun-Hui Zhang, Harrison H Zhou, et al. Asymptotic normality and optimalities in estimation of large gaussian graphical models. *The Annals of Statistics*, 43(3):991–1026, 2015.

Adam J Rothman, Peter J Bickel, Elizaveta Levina, and Ji Zhu. Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics*, 2:494–515, 2008.

Xiaotong Shen, Wei Pan, and Yunzhang Zhu. Likelihood-based selection and sharp parameter estimation. *Journal of the American Statistical Association*, 107(497):223–232, 2012.

Tingni Sun and Cun-Hui Zhang. Sparse matrix inversion with scaled LASSO. *The Journal of Machine Learning Research*, 14(1):3385–3418, 2013.

Kean Ming Tan, Palma London, Karthik Mohan, Su-In Lee, Maryam Fazel, and Daniela Witten. Learning graphical models with hubs. *The Journal of Machine Learning Research*, 15(1):3297–3331, 2014.

Kean Ming Tan, Yang Ning, Daniela M Witten, and Han Liu. Replicates in high dimensions, with applications to latent variable graphical models. *Biometrika*, 103(4):761–777, 2016.

Robert Tibshirani. Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. *The Journal of Machine Learning Research*, 12:2411–2414, 2011.

Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):467–476, 2008.

George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 10(5):293–302, 2002.

S. van de Geer, P. Bühlmann, Y. Ritov, and R. Dezeure. On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics*, 42 (3):1166–1202, 06 2014. doi: 10.1214/14-AOS1221.

Aad W Van der Vaart. *Asymptotic Statistics*. Cambridge university press, 2000.

Arend Voorman, Ali Shojaie, and Daniela Witten. Graph estimation with joint additive models. *Biometrika*, 101(1):85–101, 2014.

Zhaoran Wang, Han Liu, and Tong Zhang. Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. *The Annals of Statistics*, 42(6): 2164–2201, 12 2014. doi: 10.1214/14-AOS1238.

Joe Whittaker. *Graphical Models in Applied Multivariate Statistics*. Wiley Publishing, 2009.

Lin Xiao and Tong Zhang. A proximal-gradient homotopy method for the sparse least-squares problem. *SIAM Journal on Optimization*, 23(2):1062–1091, 2013.

Lingzhou Xue, Hui Zou, Tianxi Cai, et al. Nonconcave penalized composite conditional likelihood estimation of sparse Ising models. *The Annals of Statistics*, 40(3):1403–1429, 2012a.

Lingzhou Xue, Hui Zou, et al. Regularized rank-based estimation of high-dimensional nonparanormal graphical models. *The Annals of Statistics*, 40(5):2541–2571, 2012b.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. On graphical models via univariate exponential family distributions. *arXiv preprint arXiv:1301.4183*, 2013a.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, and Zhandong Liu. On Poisson graphical models. *Advances in Neural Information Processing Systems*, pages 1718–1726, 2013b.

Eunho Yang, Pradeep Ravikumar, Genevera I Allen, Yulia Baker, Ying-Wooi Wan, and Zhandong Liu. A general framework for mixed graphical models. *arXiv preprint arXiv:1411.0288*, 2014.

Ming Yuan. High dimensional inverse covariance matrix estimation via linear programming. *Journal of Machine Learning Research*, 11:2261–2286, 2010.

Ming Yuan and Yi Lin. Model selection and estimation in the Gaussian graphical model. *Biometrika*, 94(1):19–35, 2007.

Cun-Hui Zhang and Jian Huang. The sparsity and bias of the LASSO selection in high-dimensional linear regression. *The Annals of Statistics*, pages 1567–1594, 2008.

Cun-Hui Zhang and Stephanie S Zhang. Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 76(1):217–242, 2014.

Tong Zhang. Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research*, 11:1081–1107, 2010.

Tong Zhang et al. Multi-stage convex relaxation for feature selection. *Bernoulli*, 19(5B): 2277–2293, 2013.

Hui Zou and Runze Li. One-step sparse estimates in nonconcave penalized likelihood models. *Annals of statistics*, 36(4):1509, 2008.

Journal of Machine Learning Research 18 (2018) 1-54 Submitted 09/15; Revised 07/18; Published 11/18

Theoretical Analysis of Cross-Validation for Estimating the Risk of the k-Nearest Neighbor Classifier

Alain Celisse CELISSE@MATH.UNIV-LILLE1.FR
Laboratoire de Mathématiques
UMR 8524 CNRS-Université de Lille
Inria – Modal Project-team, Lille
F-59 655 Villeneuve d'Ascq Cedex, France

Tristan Mary-Huard MARYHUAR@AGROPARISTECH.FR
INRA, UMR 0320 / UMR 8120 Génétique Quantitative et Évolution
Le Moulon, F-91190 Gif-sur-Yvette, France
UMR MIA-Paris, AgroParisTech, INRA, Université Paris-Saclay
F-75005, Paris, France

Editor: Hui Zou

Abstract

The present work aims at deriving theoretical guaranties on the behavior of some cross-validation procedures applied to the k-nearest neighbors (kNN) rule in the context of binary classification. Here we focus on the leave-p-out cross-validation (LpO) used to assess the performance of the kNN classifier. Remarkably this LpO estimator can be efficiently computed in this context using closed-form formulas derived by Celisse and Mary-Huard (2011).

We describe a general strategy to derive moment and exponential concentration inequalities for the LpO estimator applied to the kNN classifier. Such results are obtained first by exploiting the connection between the LpO estimator and U-statistics, and second by making an intensive use of the generalized Efron-Stein inequality applied to the L1O estimator. One other important contribution is made by deriving new quantifications of the discrepancy between the LpO estimator and the classification error/risk of the kNN classifier. The optimality of these bounds is discussed by means of several lower bounds as well as simulation experiments.

Keywords: Classification, Cross-validation, Risk estimation

1. Introduction

The k-nearest neighbor (kNN) algorithm (Fix and Hodges, 1951) in binary classification is a popular prediction algorithm based on the idea that the predicted value at a new point is based on a majority vote from the k nearest labeled neighbors of this point. Although quite simple, the kNN classifier has been successfully applied to many difficult classification tasks (Li et al., 2004; Simard et al., 1998; Scheirer and Slaney, 2003). Efficient implementations have been also developed to allow dealing with large datasets (Indyk and Motwani, 1998; Andoni and Indyk, 2006).

The theoretical performances of the kNN classifier have been already extensively investigated. In the context of binary classification preliminary theoretical results date back to Cover and Hart (1967); Cover (1968); Györfi (1981). The kNN classifier has been proved to be (weakly) universally consistent by Stone (1977) as long as $k = k_n \to +\infty$ and $k/n \to 0$ as $n \to +\infty$. For the 1NN classifier, an asymptotic expansion of the error rate has been derived by Psaltis et al. (1994). The same strategy has been successfully applied to the kNN classifier by Snapp and Venkatesh (1998). Hall et al. (2008) study the influence of the parameter k on the risk of the kNN classifier by means of an asymptotic expansion derived from a Poisson or binomial model for the training points. More recently, Cannings et al. (2017) pointed out some limitations suffered by the "classical" kNN classifier and deduced an improved version based on a local choice of k in the semi-supervised context. In contrast to the aforementioned results, the work by Chaudhuri and Dasgupta (2014) focuses on the finite-sample framework. They typically provide upper bounds with high probability on the risk of the kNN classifier where the bounds are not distribution-free. Alternatively in the regression setting, Kulkarni and Posner (1995) derived a strategy leading to a finite-sample bound on the performance of 1NN, which has been extended to the (weighted) kNN rule ($k \geq 1$) by Biau et al. (2010a,b) (see also Berrett et al., 2016, where a weighted kNN estimator is designed for estimating the entropy). We refer interested readers to Biau and Devroye (2016) for an almost thorough presentation of known results on the kNN algorithm in various contexts.

In numerous (if not all) practical applications, computing the cross-validation (CV) estimator (Stone, 1974, 1982) has been among the most popular strategies to evaluate the performance of the kNN classifier (Devroye et al., 1996, Section 24.3). All CV procedures share a common principle which consists in splitting a sample of n points into two disjoint subsets called *training* and *test* sets with respective cardinalities $n - p$ and p, for any $1 \leq p \leq n-1$. The $n-p$ training set data serve to compute a classifier, while its performance is evaluated from the p *left out* data of the test set. For a complete and comprehensive review on cross-validation procedures, we refer the interested reader to Arlot and Celisse (2010).

In the present work, we focus on the leave-p-out (LpO) cross-validation. Among CV procedures, it belongs to exhaustive strategies since it considers (and averages over) all the $\binom{n}{p}$ possible such splittings of $\{1, \ldots, n\}$ into training and test sets. Usually the induced computation time of the LpO is prohibitive, which gives rise to its surrogate called $V-$fold cross-validation (V-FCV) with $V \approx n/p$ (Geisser, 1975). However, Steele (2009); Celisse and Mary-Huard (2011) recently derived closed-form formulas respectively for the bootstrap and the LpO procedures applied to the kNN classifier. Such formulas allow for an efficient computation of the LpO estimator. Moreover since the V-FCV estimator suffers the same bias but a larger variance than the LpO one (Celisse and Robin, 2008; Arlot and Celisse, 2010), LpO (with $p = \lfloor n/V \rfloor$) strictly improves upon V-FCV in the present context.

Although being favored in practice for assessing the risk of the kNN classifier, the use of CV comes with very few theoretical guarantees regarding its performance. Moreover probably for technical reasons, most existing results apply to Hold-out and leave-one-out (L1O), that is LpO with $p = 1$ (Kearns and Ron, 1999). In this paper we rather consider the general LpO procedure (for $1 \leq p \leq n-1$) used to estimate the risk (alternatively the classification error rate) of the kNN classifier. Our main purpose is then to provide distribution-free theoretical guarantees on the behavior of LpO with respect to influential parameters such as p, n, and k. For instance we aim at answering questions such as: "Does

there exist any regime of $p = p(n)$ (with $p(n)$ some function of n) where the LpO estimator is a consistent estimate of the risk of the kNN classifier?", or "Is it possible to describe the convergence rate of the LpO estimator depending on p?"

Contributions. The main contribution of the present work is two-fold: (i) we describe a new general strategy to derive moment and exponential concentration inequalities for the LpO estimator applied to the kNN binary classifier, and (ii) these inequalities serve to derive the convergence rate of the LpO estimator towards the risk of the kNN classifier.

This new strategy relies on several steps. First exploiting the connection between the LpO estimator and U-statistics (Koroljuk and Borovskich, 1994) and the Rosenthal inequality (Ibragimov and Sharakhmetov, 2002), we prove that upper bounding the polynomial moments of the centered LpO estimator reduces to deriving such bounds for the simpler L1O estimator. Second, we derive new upper bounds on the moments of the L1O estimator using the generalized Efron-Stein inequality (Boucheron et al., 2005, 2013, Theorem 15.5). Third, combining the two previous steps provides some insight on the interplay between p/n and k in the concentration rates measured in terms of moments. This finally results in new exponential concentration inequalities for the LpO estimator applying whatever the value of the ratio $p/n \in (0,1)$. In particular while the upper bounds increase with $1 \leq p \leq n/2+1$, it is no longer the case if $p > n/2+1$. We also provide several lower bounds suggesting our upper bounds cannot be improved in some sense in a distribution-free setting.

The remainder of the paper is organized as follows. The connection between the LpO estimator and U-statistics is clarified in Section 2, where we also recall the closed-form formula of the LpO estimator applied to the kNN classifier (Celisse and Mary-Huard, 2011). Order-q moments ($q \geq 2$) of the LpO estimator are then upper bounded in terms of those of the L1O estimator. This step can be applied to any classification algorithm. Section 3 then specifies the previous upper bounds in the case of the kNN classifier, which leads to the main Theorem 3.2 characterizing the concentration behavior of the LpO estimator with respect to p, n, and k in terms of polynomial moments. Deriving exponential concentration inequalities for the LpO estimator is the main concern of Section 4 where we highlight the strength of our strategy by comparing our main inequalities with concentration inequalities derived with less sophisticated tools. Finally Section 5 exploits the previous results to bound the gap between the LpO estimator and the classification error of the kNN classifier. The optimality of these upper bounds is first proved in our distribution-free framework by establishing several new lower bounds matching the upper ones in some specific settings. Second, empirical experiments are also reported which support the above conclusions.

2. U-statistics and LpO estimator

2.1. Statistical framework

Classification We tackle the binary classification problem where the goal is to predict the unknown label $Y \in \{0,1\}$ of an observation $X \in \mathcal{X} \subset \mathbb{R}^d$. The random variable (X,Y) has an *unknown* joint distribution $P_{(X,Y)}$ defined by $P_{(X,Y)}(B) = \mathbb{P}[\,(X,Y) \in B\,]$ for any Borelian set $B \in \mathcal{X} \times \{0,1\}$, where $\mathbb{P}$ denotes a reference probability distribution. In what follows no particular distributional assumption is made regarding X. To predict the label, one aims at building a classifier $\hat{f} : \mathcal{X} \to \{0,1\}$ on the basis of a set of random variables

$\mathcal{D}_n = \{Z_1, \ldots, Z_n\}$ called the training sample, where $Z_i = (X_i, Y_i)$, $1 \leq i \leq n$ represent n copies of (X,Y) drawn independently from $P_{(X,Y)}$. In settings where no confusion is possible, we will replace $\mathcal{D}_n$ by $\mathcal{D}$.

Any strategy to build such a classifier is called a *classification algorithm*, and can be formally defined as a function $\mathcal{A} : \cup_{n \geq 1} \{\mathcal{X} \times \{0,1\}\}^n \to \mathcal{F}$ that maps a training sample $\mathcal{D}_n$ onto the corresponding classifier $\mathcal{A}^{\mathcal{D}_n}(\cdot) = \hat{f} \in \mathcal{F}$, where $\mathcal{F}$ is the set of all measurable functions from $\mathcal{X}$ to $\{0,1\}$. Numerous classifiers have been considered in the literature and it is out of the scope of the present paper to review all of them (see Devroye et al. (1996) for many instances). Here we focus on the k-nearest neighbor rule (kNN) initially proposed by Fix and Hodges (1951) and further studied for instance by Devroye and Wagner (1977); Rogers and Wagner (1978).

The kNN algorithm For $1 \leq k \leq n$, the kNN classification algorithm, denoted by $\mathcal{A}_k$, consists in classifying any new observation x using a *majority vote* decision rule based on the labels of the k closest points to x, denoted by $X_{(1)}(x), \ldots, X_{(k)}(x)$, among the training sample $X_1, \ldots, X_n$. In what follows these k *nearest neighbors* are chosen according to the distance associated with the usual Euclidean norm in $\mathbb{R}^d$. Note that other *adaptive metrics* have been also considered in the literature (see for instance Hastie et al., 2001, Chap. 14). But such examples are out of the scope of the present work, that is our reference distance does not depend on the training sample at hand. Let us also emphasize that possible ties are broken by using the *smallest index* among ties, which is one possible choice for the Stone lemma to hold true (Biau and Devroye, 2016, Lemma 10.6, p.125).

Formally, given $V_k(x) = \left\{1 \leq i \leq n,\ X_i \in \left\{X_{(1)}(x), \ldots, X_{(k)}(x)\right\}\right\}$ the set of indices of the k nearest neighbors of x among $X_1, \ldots, X_n$, the kNN classifier is defined by

$$\mathcal{A}_k(\mathcal{D}_n; x) = \widehat{f}_k(\mathcal{D}_n; x) := \begin{cases} 1 & \text{, if } \frac{1}{k}\sum_{i \in V_k(x)} Y_i = \frac{1}{k}\sum_{i=1}^k Y_{(i)}(x) > 0.5 \\ 0 & \text{, if } \frac{1}{k}\sum_{i=1}^k Y_{(i)}(x) < 0.5 \\ \mathcal{B}(0.5) & \text{, otherwise} \end{cases}, \quad (2.1)$$

where $Y_{(i)}(x)$ is the label of the i-th nearest neighbor of x for $1 \leq i \leq k$, and $\mathcal{B}(0.5)$ denotes a Bernoulli random variable with parameter $1/2$.

Leave-p-out cross-validation For a given sample $\mathcal{D}_n$, the performance of any classifier $\hat{f} = \mathcal{A}^{\mathcal{D}_n}(\cdot)$ (respectively of any classification algorithm $\mathcal{A}$) is assessed by the classification error $L(\hat{f})$ (respectively the risk $R(\hat{f})$) defined by

$$L(\hat{f}) = \mathbb{P}\left(\hat{f}(X) \neq Y \mid \mathcal{D}_n\right), \quad \text{and} \quad R(\hat{f}) = \mathbb{E}\left[\mathbb{P}\left(\hat{f}(X) \neq Y \mid \mathcal{D}_n\right)\right].$$

In this paper we focus on the estimation of $L(\hat{f})$ (and its expectation $R(\hat{f})$) by use of the *Leave-p-Out* (LpO) cross-validation for $1 \leq p \leq n-1$ (Zhang, 1993; Celisse and Robin, 2008). LpO successively considers all possible splits of $\mathcal{D}_n$ into a training set of cardinality $n-p$ and a test set of cardinality p. Denoting by $\mathcal{E}_{n-p}$ the set of all possible subsets of $\{1, \ldots, n\}$ with cardinality $n-p$, any $e \in \mathcal{E}_{n-p}$ defines a split of $\mathcal{D}_n$ into a training sample $\mathcal{D}^e = \{Z_i \mid i \in e\}$ and a test sample $\mathcal{D}^{\bar{e}}$, where $\bar{e} = \{1, \ldots, n\} \setminus e$. For a given classification algorithm $\mathcal{A}$, the final LpO estimator of the performance of $\mathcal{A}^{\mathcal{D}_n}(\cdot) = \widehat{f}$ is the average (over all possible splits) of the classification error estimated on each test set, that is

$$\widehat{R}_p(\mathcal{A}, \mathcal{D}_n) = \binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \left(\frac{1}{p}\sum_{i \in \bar{e}} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i\right\}}\right), \quad (2.2)$$

where $\mathcal{A}^{\mathcal{D}^e}(\cdot)$ is the classifier built from $\mathcal{D}^e$. We refer the reader to Arlot and Celisse (2010) for a detailed description of LpO and other cross-validation procedures. In the sequel, the lengthy notation $\widehat{R}_p(\mathcal{A}, \mathcal{D}_n)$ is replaced by $\widehat{R}_{p,n}$ in settings where no confusion can arise about the algorithm $\mathcal{A}$ or the training sample $\mathcal{D}_n$, and by $\widehat{R}_p(\mathcal{D}_n)$ if the training sample has to be kept in mind.

Exact LpO for the kNN classification algorithm Usually due to its seemingly prohibitive computational cost, LpO is not applied except with $p = 1$ where it reduces to the well known leave-one-out. However in several contexts such as density estimation (Celisse and Robin, 2008; Celisse, 2014) or regression (Celisse, 2008), closed-form formulas have been derived for the LpO estimator when applied with projection and kernel estimators. The kNN classifier is another instance of such estimators for which efficiently computing the LpO estimator is possible. Its computation requires a time complexity that is linear in p as previously established by Celisse and Mary-Huard (2011). Let us briefly recall the main steps leading to the closed-form formula.

1. From Eq. (2.2) the LpO estimator can be expressed as a sum (over the n observations of the complete sample) of probabilities:

$$\begin{aligned}\binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \left(\sum_{i \notin e} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i\right\}} \right) &= \frac{1}{p} \sum_{i=1}^{n} \left[\binom{n}{p}^{-1} \sum_{e \in \mathcal{E}_{n-p}} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i\right\}} \mathbb{1}_{\{i \notin e\}} \right] \\ &= \frac{1}{p} \sum_{i=1}^{n} \mathbb{P}_e(\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i \mid i \notin e) \mathbb{P}_e(i \notin e).\end{aligned}$$

   Here $\mathbb{P}_e$ means that the integration is made with respect to the random variable $e \in \mathcal{E}_{n-p}$, which follows the uniform distribution over the $\binom{n}{p}$ possible subsets in $\mathcal{E}_{n-p}$ with cardinality $n-p$. For instance $\mathbb{P}_e(i \notin e) = p/n$ since it is the proportion of subsamples with cardinality $n-p$ which do not contain a given prescribed index i, which equals $\binom{n-1}{n-p}/\binom{n}{p}$. (See also Lemma D.4 for further examples of such calculations.)

2. For any X_i, let $X_{(1)}, ..., X_{(k+p-1)}, X_{(k+p)}, ..., X_{(n-1)}$ be the ordered sequence of neighbors of X_i. This list depends on X_i, that is $X_{(1)}$ should be noted $X_{(i,1)}$. But this dependency is skipped here for the sake of readability.

   The key in the derivation is to condition with respect to the random variable R_k^i which denotes the rank (in the whole sample $\mathcal{D}_n$) of the k-th neighbor of X_i in the $\mathcal{D}^e$. For instance $R_k^i = j$ means that $X_{(j)}$ is the k-th neighbor of X_i in $\mathcal{D}^e$. Then

$$\mathbb{P}_e(\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i | i \notin e) = \sum_{j=k}^{k+p-1} \mathbb{P}_e(\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i \mid R_k^i = j,\ i \notin e) \mathbb{P}_e(R_k^i = j \mid i \notin e),$$

   where the sum involves p terms since only $X_{(k)}, \ldots, X_{(k+p-1)}$ are candidates for being the k-th neighbor of X_i in at least one training subset e.

3. Observe that the resulting probabilities can be easily computed (see Lemma D.4):

   $\star$ $\mathbb{P}_e(i \notin e) = \frac{p}{n}$
   $\star$ $\mathbb{P}_e(R_k^i = j | i \notin e) = \frac{k}{j} P(U = j - k)$
   $\star$ $\mathbb{P}_e(\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i | V_k^i = j,\ i \notin e) = (1 - Y_j)\left[1 - F_H\left(\frac{k+1}{2}\right)\right] + Y_j\left[1 - F_{H'}\left(\frac{k-1}{2}\right)\right]$,

   with $U \sim \mathcal{H}(j, n-j-1, p-1)$, $H \sim \mathcal{H}(N_i^j, j - N_i^j - 1, k-1)$, and $H' \sim \mathcal{H}(N_i^j - 1, j - N_i^j, k-1)$, where F_H and $F_{H'}$ respectively denote the cumulative distribution functions of H and H', $\mathcal{H}$ denotes the hypergeometric distribution, and N_i^j is the number of 1's among the j nearest neighbors of X_i in $\mathcal{D}_n$.

The computational cost of LpO for the kNN classifier is the same as that of L1O for the $(k + p - 1)$NN classifier whatever p, that is $O(p\,n)$. This contrasts with the usual $\binom{n}{p}$ prohibitive computational complexity seemingly suffered by LpO.

2.2. U-statistics: General bounds on LpO moments

The purpose of the present section is to describe a general strategy allowing to derive new upper bounds on the polynomial moments of the LpO estimator. As a first step of this strategy, we establish the connection between the LpO risk estimator and U-statistics. Second, we exploit this connection to derive new upper bounds on the order-q moments of the LpO estimator for $q \geq 2$. Note that these upper bounds, which relate moments of the LpO estimator to those of the L1O estimator, hold true with any classifier.

Let us start by introducing U-statistics and recalling some of their basic properties that will serve our purposes. For a thorough presentation, we refer to the books by Serfling (1980); Koroljuk and Borovskich (1994). The first step is the definition of a U-statistic of order $m \in \mathbb{N}^*$ as an average over all m-tuples of distinct indices in $\{1, \ldots, n\}$.

Definition 2.1 (Koroljuk and Borovskich (1994)). *Let $h : \mathcal{X}^m \longrightarrow \mathbb{R}$ denote any measurable function where $m \geq 1$ is an integer. Let us further assume h is a symmetric function of its arguments. Then any function $U_n : \mathcal{X}^n \longrightarrow \mathbb{R}$ such that*

$$U_n(x_1, \ldots, x_n) = U_n(h)(x_1, \ldots, x_n) = \binom{n}{m}^{-1} \sum_{1 \leq i_1 < \ldots < i_m \leq n} h(x_{i_1}, \ldots, x_{i_m})$$

where $m \leq n$, is a U-statistic of order m and kernel h.

Before clarifying the connection between LpO and U-statistics, let us introduce the main property of U-statistics our strategy relies on. It consists in representing any U-statistic as an average, over all permutations, of sums of independent variables.

Proposition 2.1 (Eq. (5.5) in Hoeffding (1963)). *With the notation of Definition 2.1, let us define $W : \mathcal{X}^n \longrightarrow \mathbb{R}$ by*

$$W(x_1, \ldots, x_n) = \frac{1}{r} \sum_{j=1}^{r} h\left(x_{(j-1)m+1}, \ldots, x_{jm}\right), \tag{2.3}$$

where $r = \lfloor n/m \rfloor$ denotes the integer part of n/m. Then

$$U_n(x_1, \ldots, x_n) = \frac{1}{n!} \sum_{\sigma} W\left(x_{\sigma(1)}, \ldots, x_{\sigma(n)}\right),$$

where $\sum_\sigma$ denotes the summation over all permutations σ of $\{1, \ldots, n\}$.

We are now in position to state the key remark of the paper. All the developments further exposed in the following result from this connection between the LpO estimator defined by Eq. (2.2) and U-statistics.

Theorem 2.1. *For any classification algorithm $\mathcal{A}$ and any $1 \leq p \leq n-1$ such that a classifier can be computed from $\mathcal{A}$ on $n-p$ training points, the LpO estimator $\widehat{R}_{p,n}$ is a U-statistic of order $m = n-p+1$ with kernel $h_m \colon \mathcal{X}^m \longrightarrow \mathbb{R}$ defined by*

$$h_m(Z_1, \ldots, Z_m) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}_m^{(i)}}(X_i) \neq Y_i\right\}},$$

where $\mathcal{D}_m^{(i)}$ denotes the sample $\mathcal{D}_m = (Z_1, \ldots, Z_m)$ with Z_i withdrawn.

Note for instance that when $\mathcal{A} = \mathcal{A}_k$ denotes the kNN algorithm, the cardinality of $\mathcal{D}_m^{(i)}$ has to satisfy $n-p \geq k$, which implies that $1 \leq p \leq n-k \leq n-1$.

Proof of Theorem 2.1.

From Eq. (2.2), the LpO estimator of the performance of any classification algorithm $\mathcal{A}$ computed from $\mathcal{D}_n$ satisfies

$$\begin{aligned}
\widehat{R}_p(\mathcal{A}, \mathcal{D}_n) = \widehat{R}_{p,n} &= \frac{1}{\binom{n}{p}} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \sum_{i \in \bar{e}} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i\right\}} \\
&= \frac{1}{\binom{n}{p}} \sum_{e \in \mathcal{E}_{n-p}} \frac{1}{p} \sum_{i \in \bar{e}} \left(\sum_{v \in \mathcal{E}_{n-p+1}} \mathbb{1}_{\{v = e \cup \{i\}\}} \right) \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^e}(X_i) \neq Y_i\right\}},
\end{aligned}$$

since there is a unique set of indices v with cardinality $n-p+1$ such that $v = e \cup \{i\}$. Then

$$\widehat{R}_{p,n} = \frac{1}{\binom{n}{p}} \sum_{v \in \mathcal{E}_{n-p+1}} \frac{1}{p} \sum_{i=1}^{n} \left(\sum_{e \in \mathcal{E}_{n-p}} \mathbb{1}_{\{v = e \cup \{i\}\}} \mathbb{1}_{\{i \in \bar{e}\}} \right) \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^{v \setminus \{i\}}}(X_i) \neq Y_i\right\}}.$$

Furthermore for v and i fixed, $\sum_{e \in \mathcal{E}_{n-p}} \mathbb{1}_{\{v = e \cup \{i\}\}} \mathbb{1}_{\{i \in \bar{e}\}} = \mathbb{1}_{\{i \in v\}}$ since there is a unique set of indices e such that $e = v \setminus i$. One gets

$$\begin{aligned}
\widehat{R}_{p,n} &= \frac{1}{p} \frac{1}{\binom{n}{p}} \sum_{v \in \mathcal{E}_{n-p+1}} \sum_{i=1}^{n} \mathbb{1}_{\{i \in v\}} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^{v \setminus \{i\}}}(X_i) \neq Y_i\right\}} \\
&= \frac{1}{\binom{n}{n-p+1}} \sum_{v \in \mathcal{E}_{n-p+1}} \frac{1}{n-p+1} \sum_{i \in v} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}^{v \setminus \{i\}}}(X_i) \neq Y_i\right\}},
\end{aligned}$$

by noticing $p\binom{n}{p} = \frac{pn!}{p!n-p!} = \frac{n!}{p-1!n-p!} = (n-p+1)\binom{n}{n-p+1}$.

□

The kernel h_m is a deterministic and symmetric function of its arguments that does only depend on m. Let us also notice that $h_m(Z_1, \ldots, Z_m)$ reduces to the L1O estimator of the risk of the classifier $\mathcal{A}$ computed from $Z_1, \ldots, Z_m$, that is

$$h_m(Z_1, \ldots, Z_m) = \widehat{R}_1(\mathcal{A}, \mathcal{D}_m) = \widehat{R}_{1,n-p+1}. \tag{2.4}$$

In the context of testing whether two binary classifiers have different error rates, this fact has already been pointed out by Fuchs et al. (2013).

We now derive a general upper bound on the q-th moment ($q \geq 1$) of the LpO estimator that holds true for any classifier (as long as the following expectations are well defined).

Theorem 2.2. *For any classifier $\mathcal{A}$, let $\mathcal{A}^{\mathcal{D}_n}(\cdot)$ and $\mathcal{A}^{\mathcal{D}_m}(\cdot)$ be the corresponding classifiers built from respectively $\mathcal{D}_n$ and $\mathcal{D}_m$, where $m = n-p+1$. Then for every $1 \leq p \leq n-1$ such that a classifier can be computed from $\mathcal{A}$ on $n-p$ training points, and for any $q \geq 1$,*

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] \leq \mathbb{E}\left[\left|\widehat{R}_{1,m} - \mathbb{E}\left[\widehat{R}_{1,m}\right]\right|^q\right]. \tag{2.5}$$

Furthermore as long as $p > n/2+1$, one also gets

- *for $q = 2$*

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^2\right] \leq \frac{\mathbb{E}\left[\left|\widehat{R}_{1,m} - \mathbb{E}\left[\widehat{R}_{1,m}\right]\right|^2\right]}{\left\lfloor \frac{n}{m} \right\rfloor}. \tag{2.6}$$

- *for every $q > 2$*

$$\begin{aligned}
&\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] \leq B(q,\gamma) \times \\
&\max\left\{ 2^q \gamma \left\lfloor \frac{n}{m} \right\rfloor \mathbb{E}\left[\left| \frac{\widehat{R}_{1,m} - \mathbb{E}\left[\widehat{R}_{1,m}\right]}{\left\lfloor \frac{n}{m} \right\rfloor} \right|^q \right], \left(\sqrt{\frac{2\mathrm{Var}\left(\widehat{R}_{1,m}\right)}{\left\lfloor \frac{n}{m} \right\rfloor}} \right)^q \right\},
\end{aligned} \tag{2.7}$$

  where $\gamma > 0$ is a numeric constant and $B(q,\gamma)$ denotes the optimal constant defined in the Rosenthal inequality (Proposition D.2).

The proof is given in Appendix A.1. Eq. (2.5) and (2.6) straightforwardly result from the Jensen inequality applied to the average over all permutations provided in Proposition 2.1. If $p > n/2+1$, the integer part $\lfloor n/m \rfloor$ becomes larger than 1 and Eq. (2.6) becomes better than Eq. (2.5) for $q = 2$. As a consequence of our strategy of proof, the right-hand side of Eq. (2.6) is equal to the classical upper bound on the variance of U-statistics which suggests it cannot be improved without adding further assumptions.

Unlike the above ones, Eq. (2.7) is derived from the Rosenthal inequality, which enables us to upper bound a sum $\left\|\sum_{i=1}^r \xi_i\right\|_q$ of independent and identically centered random variables in terms of $\sum_{i=1}^r \|\xi_i\|_q$ and $\sum_{i=1}^r \mathrm{Var}(\xi_i)$. Let us remark that, for $q = 2$, both terms of the right-hand side of Eq. (2.7) are of the same order as Eq. (2.6) up to constants. Furthermore using the Rosenthal inequality allows taking advantage of the integer part $\lfloor n/m \rfloor$ when $p > n/2+1$ (unlike what we get by using Eq.(2.5) for $q > 2$). In particular it provides a new understanding of the behavior of the LpO estimator when $p/n \to 1$ as highlighted later by Proposition 4.2.

3. New bounds on LpO moments for the kNN classifier

Our goal is now to specify the general upper bounds provided by Theorem 2.2 in the case of the kNN algorithm $\mathcal{A}_k$ ($1 \leq k \leq n$) introduced by (2.1).

Since Theorem 2.2 expresses the moments of the LpO estimator in terms of those of the L1O estimator computed from $\mathcal{D}_m$ (with $m = n-p+1$), the next step consists in focusing on the L1O moments. Deriving upper bounds on the moments of the L1O is achieved using a generalization of the well-known Efron-Stein inequality (see Theorem D.1 for Efron-Stein's inequality and Theorem 15.5 in Boucheron et al. (2013) for its generalization). For the sake of completeness, we first recall a corollary of this generalization that is proved in Section D.1.4 (see Corollary D.1).

Proposition 3.1. *Let $\xi_1, \ldots, \xi_n$ denote n independent Ξ-valued random variables and $\zeta = f(\xi_1, \ldots, \xi_n)$, where $f : \Xi^n \to \mathbb{R}$ is any measurable function. With $\xi'_1, \ldots, \xi'_n$ independent copies of the ξ_is, there exists a universal constant $\kappa \leq 1.271$ such that for any $q \geq 2$,*

$$\|\zeta - \mathbb{E}\zeta\|_q \leq \sqrt{2\kappa q} \sqrt{\left\| \sum_{i=1}^{n} \left(f(\xi_1, \ldots, \xi_i, \ldots, \xi_n) - f(\xi_1, \ldots, \xi'_i, \ldots, \xi_n) \right)^2 \right\|_{q/2}} .$$

Then applying Proposition 3.1 with $\zeta = \widehat{R}_1(\mathcal{A}_k, \mathcal{D}_m) = \widehat{R}_{1,m}$ (L1O estimator computed from $\mathcal{D}_m$ with $m = n-p+1$) and $\Xi = \mathbb{R}^d \times \{0,1\}$ leads to the following Theorem 3.1. It controls the order-q moments of the L1O estimator applied to the kNN classifier.

Theorem 3.1. *For every $1 \leq k \leq n-1$, let $A_k^{\mathcal{D}_m}$ ($m = n-p+1$) denote the kNN classifier learnt from $\mathcal{D}_m$ and $\widehat{R}_{1,m}$ be the corresponding L1O estimator given by Eq. (2.2). Then*

- *for $q = 2$,*

$$\mathbb{E}\left[\left(\widehat{R}_{1,m} - \mathbb{E}\left[\widehat{R}_{1,m} \right] \right)^2 \right] \leq C_1 \frac{k^{3/2}}{m} \ ; \tag{3.1}$$

- *for every $q > 2$,*

$$\mathbb{E}\left[\left| \widehat{R}_{1,m} - \mathbb{E}\left[\widehat{R}_{1,m} \right] \right|^q \right] \ \leq (C_2 \cdot k)^q \left(\tfrac{q}{m} \right)^{q/2}, \tag{3.2}$$

with $C_1 = 2 + 16\gamma_d$ and $C_2 = 4\gamma_d\sqrt{2\kappa}$, where γ_d is a constant (arising from Stone's lemma, see Lemma D.5) that grows exponentially with dimension d, and κ is defined in Proposition 3.1.

Its proof (detailed in Section A.2) relies on Stone's lemma (Lemma D.5). For a given X_i, it proves that the number of points in $\mathcal{D}_n^{(i)}$ having X_i among their k nearest neighbors is not larger than $k\gamma_d$. The dependence of our upper bounds with respect to γ_d (see explicit constants C_1 and C_2) induces their strong deterioration as the dimension d grows since $\gamma_d \approx 4.8^d - 1$. Therefore the larger the dimension d, the larger the required sample size n for the upper bound to be small (at least smaller than 1). Note also that the tie breaking strategy (based on the smallest index in the present work) is chosen so that it ensures Stone's lemma to hold true.

In Eq. (3.1), the easier case $q = 2$ enables to exploit exact calculations of (rather than upper bounds on) the variance of the L1O estimator. Since $\mathbb{E}\left[\widehat{R}_{1,m} \right] = R\left(A_k^{\mathcal{D}_{n-p}} \right)$ (risk of the kNN classifier computed from $\mathcal{D}_{n-p}$), the resulting $k^{3/2}/m$ rate is a strict improvement upon the usual k^2/m that is derived from using the sub-Gaussian exponential concentration inequality proved by Theorem 24.4 in Devroye et al. (1996).

By contrast the larger k^q arising in Eq. (3.2) results from the difficulty to derive a tight upper bound for the expectation of $(\sum_{i=1}^n \mathbb{1}_{\left\{ A_k^{\mathcal{D}_m^{(i)}}(X_i) \neq A_k^{\mathcal{D}_m^{(i,j)}}(X_i) \right\}})^q$ with $q > 2$, where $\mathcal{D}_m^{(i)}$ (resp. $\mathcal{D}_m^{(i,j)}$) denotes the sample $\mathcal{D}_m$ where Z_i has been (resp. Z_i and Z_j have been) removed.

We are now in position to state the main result of this section. It follows from the combination of Theorem 2.2 (connecting moments of the LpO and L1O estimators) and Theorem 3.1 (providing an upper bound on the order-q moments of the L1O).

Theorem 3.2. *For every $p, k \geq 1$ such that $p+k \leq n$, let $\widehat{R}_{p,n}$ denote the LpO risk estimator (see (2.2)) of the kNN classifier $\mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ defined by (2.1). Then there exist (known) constants $C_1, C_2 > 0$ such that for every $1 \leq p \leq n-k$,*

- *for $q = 2$,*

$$\mathbb{E}\left[\left(\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n} \right] \right)^2 \right] \leq C_1 \frac{k^{3/2}}{(n-p+1)} \ ; \tag{3.3}$$

- *for every $q > 2$,*

$$\mathbb{E}\left[\left| \widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n} \right] \right|^q \right] \leq (C_2 k)^q \left(\tfrac{q}{n-p+1} \right)^{q/2}, \tag{3.4}$$

with $C_1 = \frac{128\kappa\gamma_d}{\sqrt{2\pi}}$ and $C_2 = 4\gamma_d\sqrt{2\kappa}$, where γ_d denotes the constant arising from Stone's lemma (Lemma D.5). Furthermore in the particular setting where $n/2 + 1 < p \leq n-k$, then

- *for $q = 2$,*

$$\mathbb{E}\left[\left(\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n} \right] \right)^2 \right] \leq C_1 \frac{k^{3/2}}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor} \ , \tag{3.5}$$

- *for every $q > 2$,*

$$\begin{aligned} &\mathbb{E}\left[\left| \widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n} \right] \right|^q \right] \\ &\leq \left\lfloor \frac{n}{n-p+1} \right\rfloor \Gamma^q \max\left(\frac{k^{3/2}}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor} q, \frac{k^2}{(n-p+1)\left\lfloor \frac{n}{n-p+1} \right\rfloor^2} q^3 \right)^{q/2} \end{aligned} \tag{3.6}$$

where $\Gamma = 2\sqrt{2e}\max\left(\sqrt{2C_1}, 2C_2\right)$.

The straightforward proof is detailed in Section A.3. Let us start by noticing that both upper bounds in Eq. (3.3) and (3.4) deteriorate as p grows. This is no longer the case for Eq. (3.5) and (3.6), which are specifically designed to cover the setup where $p > n/2+1$, that is where $\lfloor n/m \rfloor$ is no longer equal to 1. Therefore unlike Eq. (3.3) and (3.4), these last two inequalities are particularly relevant in the setup where $p/n \to 1$, as $n \to +\infty$, which has been investigated by Shao (1993); Yang (2006, 2007); Celisse (2014). Eq. (3.5) and (3.6) lead to respective convergence rates at worse $k^{3/2}/n$ (for $q = 2$) and k^q/n^{q-1} (for $q > 2$). In particular this last rate becomes approximately equal to $(k/n)^q$ as q gets large.

One can also emphasize that, as a U-statistic of fixed order $m = n - p + 1$, the LpO estimator has a known Gaussian limiting distribution, that is (see Theorem A, Section 5.5.1 Serfling, 1980)

$$\frac{\sqrt{n}}{m}\left(\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right]\right) \xrightarrow[n\to+\infty]{\mathcal{L}} \mathcal{N}\left(0, \sigma_1^2\right),$$

where $\sigma_1^2 = \mathrm{Var}\left[\, g(Z_1)\,\right]$, with $g(z) = E\left[\, h_m(z, Z_2, \ldots, Z_m)\,\right]$. Therefore the upper bound given by Eq. (3.5) is non-improvable in some sense with respect to the interplay between n and p since one recovers the right magnitude for the variance term as long as $m = n-p+1$ is assumed to be constant.

Finally Eq. (3.6) has been derived using a specific version of the Rosenthal inequality (Ibragimov and Sharakhmetov, 2002) stated with the optimal constant and involving a "balancing factor". In particular this balancing factor has allowed us to optimize the relative weight of the two terms between brackets in Eq. (3.6). This leads us to claim that the dependence of the upper bound with respect to q cannot be improved with this line of proof. However we cannot conclude that the term in q^3 cannot be improved using other technical arguments.

4. Exponential concentration inequalities

This section provides exponential concentration inequalities for the LpO estimator applied to the kNN classifier. Our main results heavily rely on the moment inequalities previously derived in Section 3, namely Theorem 3.2. In order to emphasize the gain allowed by this strategy of proof, we start this section by successively proving two exponential inequalities obtained with less sophisticated tools. We then discuss the strength and weakness of each of them to justify the additional refinements we introduce step by step along the section.

A first exponential concentration inequality for $\widehat{R}_p(\mathcal{A}_k, \mathcal{D}_n) = \widehat{R}_{p,n}$ can be derived by use of the bounded difference inequality following the line of proof of Devroye et al. (1996, Theorem 24.4) originally developed for the L1O estimator.

Proposition 4.1. *For any integers $p, k \geq 1$ such that $p + k \leq n$, let $\widehat{R}_{p,n}$ denote the LpO estimator (2.2) of the classification error of the kNN classifier $\mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ defined by (2.1). Then for every $t > 0$,*

$$\mathbb{P}\left(\left|\widehat{R}_{p,n} - \mathbb{E}\left(\widehat{R}_{p,n}\right)\right| > t\right) \leq 2e^{-n\frac{t^2}{8(k+p-1)^2\gamma_d^2}}. \tag{4.1}$$

where γ_d denotes the constant introduced in Stone's lemma (Lemma D.5).

The proof is given in Appendix B.1.

The upper bound of Eq. (4.1) strongly exploits the facts that: (i) for X_j to be one of the k nearest neighbors of X_i in at least one subsample X^e, it requires X_j to be one of the $k+p-1$ nearest neighbors of X_i in the complete sample, and (ii) the number of points for which X_j may be one of the $k+p-1$ nearest neighbors cannot be larger than $(k+p-1)\gamma_d$ by Stone's Lemma (see Lemma D.5).

This reasoning results in a rough upper bound since the denominator in the exponent exhibits a $(k+p-1)^2$ factor where k and p play the same role. The reason is that we do not distinguish between points for which X_j is among or above the k nearest neighbors of X_i in the whole sample (although these two setups lead to highly different probabilities of being among the k nearest neighbors in the training sample). Consequently the dependence of the convergence rate on k and p in Proposition 4.1 can be improved, as confirmed by forthcoming Theorems 4.1 and 4.2.

Based on the previous comments, a sharper quantification of the influence of each neighbor among the $k+p-1$ ones leads to the next result.

Theorem 4.1. *For every $p, k \geq 1$ such that $p + k \leq n$, let $\widehat{R}_{p,n}$ denote the LpO estimator (2.2) of the classification error of the kNN classifier $\mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ defined by (2.1). Then there exists a numeric constant $\square > 0$ such that for every $t > 0$,*

$$\max\left(\mathbb{P}\left(\widehat{R}_{p,n} - \mathbb{E}\left(\widehat{R}_{p,n}\right) > t\right), \mathbb{P}\left(\mathbb{E}\left(\widehat{R}_{p,n}\right) - \widehat{R}_{p,n} > t\right)\right) \leq \exp\left(-n\frac{t^2}{\square k^2\left[1 + (k+p)\frac{p-1}{n-1}\right]}\right),$$

with $\square = 1024e\kappa(1+\gamma_d)$, where γ_d is introduced in Lemma D.5 and $\kappa \leq 1.271$ is a universal constant.

The proof is given in Section B.2.

Unlike Proposition 4.1, taking into account the rank of each neighbor in the whole sample enables us to considerably reduce the weight of p (compared to that of k) in the denominator of the exponent. In particular, letting $p/n \to 0$ as $n \to +\infty$ (with k assumed to be fixed for instance) makes the influence of the $k+p$ factor asymptotically negligible. This would allow for recovering (up to numeric constants) a similar upper bound to that of Devroye et al. (1996, Theorem 24.4), achieved with $p = 1$.

However the upper bound of Theorem 4.1 does not reflect the right dependencies with respect to k and p compared with what has been proved for polynomial moments in Theorem 3.2. In particular it deteriorates as p increases unlike the upper bounds derived for $p > n/2+1$ in Theorem 3.2. This drawback is overcome by the following result, which is our main contribution in the present section.

Theorem 4.2. *For every $p, k \geq 1$ such that $p+k \leq n$, let $\widehat{R}_{p,n}$ denote the LpO estimator of the classification error of the kNN classifier $\hat{f}_k = \mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ defined by (2.1). Then for every $t > 0$,*

$$\max\left(\mathbb{P}\left(\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right] > t\right), \mathbb{P}\left(\mathbb{E}\left[\,\widehat{R}_{p,n}\,\right] - \widehat{R}_{p,n} > t\right)\right) \leq \exp\left(-(n-p+1)\frac{t^2}{\Delta^2 k^2}\right), \tag{4.2}$$

where $\Delta = 4\sqrt{e}\max\left(C_2, \sqrt{C_1}\right)$ *with* $C_1, C_2 > 0$ *defined in Theorem 3.1.*

Furthermore in the particular setting where $p > n/2+1$*, it comes*

$$\max\left(\mathbb{P}\left(\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right] > t\right), \mathbb{P}\left(\mathbb{E}\left[\widehat{R}_{p,n}\right] - \widehat{R}_{p,n} > t\right)\right) \leq e\left\lfloor\frac{n}{n-p+1}\right\rfloor \times$$
$$\exp\left[-\frac{1}{2e}\min\left\{(n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor\frac{t^2}{4\Gamma^2 k^{3/2}}, \left((n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor\right)^2\frac{t^2}{4\Gamma^2 k^2}\right)^{1/3}\right\}\right], \tag{4.3}$$

where Γ *arises in Eq.* (3.6) *and* γ_d *denotes the constant introduced in Stone's lemma (Lemma D.5).*

The proof has been postponed to Appendix B.3. It involves different arguments for deriving the two inequalities (4.2) and (4.3) depending on the range of values of p. Firstly for $p \leq n/2+1$, a simple argument is applied to derive Ineq. (4.2) from the two corresponding moment inequalities of Theorem 3.2 characterizing the sub-Gaussian behavior of the LpO estimator in terms of its even moments (see Lemma D.2). Secondly for $p > n/2+1$, we rather exploit: (i) the appropriate upper bounds on the moments of the LpO estimator given by Theorem 3.2, combined with (ii) Proposition D.1 which establishes exponential concentration inequalities from general moment upper bounds.

In accordance with the conclusions drawn about Theorem 3.2, the upper bound of Eq. (4.2) increases as p grows unlike that of Eq. (4.3). The best concentration rate in Eq. (4.3) is achieved as $p/n \to 1$, whereas Eq. (4.2) turns out to be useless in that setting. However Eq. (4.2) remains strictly better than Theorem 4.1 as long as $p/n \to \delta \in [0,1[$ as $n \to +\infty$. Note also that the constants Γ and γ_d are the same as in Theorem 3.1. Therefore the same comments regarding their dependence with respect to the dimension d apply here.

In order to facilitate the interpretation of the last Ineq. (4.3), we also derive the following proposition (proved in Appendix B.3) which focuses on the description of each deviation term in the particular case where $p > n/2+1$.

Proposition 4.2. *With the same notation as Theorem 4.2, for any* $p, k \geq 1$ *such that* $p+k \leq n$*,* $p > n/2+1$*, and for every* $t > 0$

$$\mathbb{P}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right| > \frac{\sqrt{2e}\Gamma}{\sqrt{n-p+1}}\left(\sqrt{\frac{k^{3/2}}{\left\lfloor\frac{n}{n-p+1}\right\rfloor}t} + 2e\frac{k}{\left\lfloor\frac{n}{n-p+1}\right\rfloor}t^{3/2}\right)\right] \leq \left\lfloor\frac{n}{n-p+1}\right\rfloor e\cdot e^{-t},$$

where $\Gamma > 0$ *is the constant arising from* (3.6).

The present inequality is very similar to the well-known Bernstein inequality (Boucheron et al., 2013, Theorem 2.10) except the second deviation term of order $t^{3/2}$ instead of t (for the Bernstein inequality).

With respect to n, the first deviation term is of order $\approx k^{3/2}/\sqrt{n}$, which is the same as with the Bernstein inequality. The second deviation term is of a somewhat different order, that is $\approx k\sqrt{n-p+1}/n$, as compared with the usual $1/n$ in the Bernstein inequality.

Nevertheless we almost recover the k/n rate by choosing for instance $p \approx n(1-\log n/n)$, which leads to $k\sqrt{\log n}/n$. Therefore varying p allows to interpolate between the $k/\sqrt{n}$ and the k/n rates.

Note also that the dependence of the first (sub-Gaussian) deviation term with respect to k is only $k^{3/2}$, which improves upon the usual k^2 resulting from Ineq. (4.2) in Theorem 4.2 for instance. However this $k^{3/2}$ remains certainly too large for being optimal even if this question remains widely open at this stage in the literature.

More generally one strength of our approach is its versatility. Indeed the two above deviation terms directly result from the two upper bounds on the moments of the L1O established in Theorem 3.1. Therefore any improvement of the latter upper bounds would immediately lead to enhance the present concentration inequality (without changing the proof).

5. Assessing the gap between LpO and classification error

5.1. Upper bounds

First, we derive new upper bounds on different measures of the discrepancy between $\widehat{R}_{p,n} = \widehat{R}_p\left(\mathcal{A}_k, \mathcal{D}_n\right)$ and the classification error $L(\hat{f}_k)$ or the risk $R(\hat{f}_k) = \mathbb{E}\left[L(\hat{f}_k)\right]$. These bounds on the L$p$O estimator are completely new for $p > 1$, some of them being extensions of former ones specifically derived for the L1O estimator applied to the kNN classifier.

Theorem 5.1. *For every* $p, k \geq 1$ *such that* $p \leq \sqrt{k}$ *and* $\sqrt{k}+k \leq n$*, let* $\widehat{R}_{p,n}$ *denote the LpO risk estimator (see* (2.2)*) of the kNN classifier* $\hat{f}_k = \mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ *defined by* (2.1)*. Then,*

$$\left|\mathbb{E}\left[\widehat{R}_{p,n}\right] - R(\hat{f}_k)\right| \leq \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n}\ , \tag{5.1}$$

and

$$\mathbb{E}\left[\left(\widehat{R}_{p,n} - R(\hat{f}_k)\right)^2\right] \leq \frac{128\kappa\gamma_d}{\sqrt{2\pi}}\frac{k^{3/2}}{n-p+1} + \frac{16}{2\pi}\frac{p^2 k}{n^2}\ . \tag{5.2}$$

Moreover,

$$\mathbb{E}\left[\left(\widehat{R}_{p,n} - L(\hat{f}_k)\right)^2\right] \leq \frac{2\sqrt{2}}{\sqrt{\pi}}\frac{(2p+3)\sqrt{k}}{n} + \frac{1}{n}\ . \tag{5.3}$$

In contrast to the results in the previous sections, a new restriction on p arises in Theorem 5.1, that is $p \leq \sqrt{k}$. This comes from the use of Lemma D.6 (proved by Devroye and Wagner (1979b)), which gives an upper bound on the L^1 stability of the kNN classifier when p observations are removed from the training sample $\mathcal{D}_n$. Actually this upper bound only remains meaningful as long as $1 \leq p \leq \sqrt{k}$.

Proof of Theorem 5.1.
Proof of (5.1)**:** With $\hat{f}_k^e = \mathcal{A}_k^{\mathcal{D}^e}$, Lemma D.6 immediately provides

$$\begin{aligned}\left|\mathbb{E}\left[\widehat{R}_{p,n} - L(\hat{f}_k)\right]\right| &= \left|\mathbb{E}\left[L(\hat{f}_k^e)\right] - \mathbb{E}\left[L(\hat{f}_k)\right]\right| \\ &\leq \mathbb{E}\left[\left|\mathbb{1}_{\{\mathcal{A}_k^{\mathcal{D}^e}(X)\neq Y\}} - \mathbb{1}_{\{\mathcal{A}_k^{\mathcal{D}_n}(X)\neq Y\}}\right|\right] \\ &= \mathbb{P}\left(\mathcal{A}_k^{\mathcal{D}^e}(X) \neq \mathcal{A}_k^{\mathcal{D}_n}(X)\right) \leq \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n} \ .\end{aligned}$$

Proof of (5.2)**:** The proof combines the previous upper bound with the one established for the variance of the LpO estimator, that is Eq. (3.3).

$$\begin{aligned}\mathbb{E}\left[\left(\widehat{R}_{p,n} - \mathbb{E}\left[L(\hat{f}_k)\right]\right)^2\right] &= \mathbb{E}\left[\left(\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right)^2\right] + \left(\mathbb{E}\left[\widehat{R}_{p,n}\right] - \mathbb{E}\left[L(\hat{f}_k)\right]\right)^2 \\ &\leq \frac{128\kappa\gamma_d}{\sqrt{2\pi}}\frac{k^{3/2}}{n-p+1} + \left(\frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n}\right)^2 \ ,\end{aligned}$$

which concludes the proof.

The proof of Ineq. (5.3) is more intricate and has been postponed to Appendix C.1. □

Keeping in mind that $\mathbb{E}\left[\widehat{R}_{p,n}\right] = R(\mathcal{A}_k^{\mathcal{D}_{n-p}})$, the right-hand side of Ineq. (5.1) is an upper bound on the bias of the LpO estimator, that is on the difference between the risks of the classifiers built from respectively $n-p$ and n points. Therefore, the fact that this upper bound increases with p is reliable since the classifiers $\mathcal{A}_k^{\mathcal{D}_{n-p+1}}(\cdot)$ and $\mathcal{A}_k^{\mathcal{D}_n}(\cdot)$ can become more and more different from one another as p increases. More precisely, the upper bound in Ineq. (5.1) goes to 0 provided $p\sqrt{k}/n$ does. With the additional restriction $p \leq \sqrt{k}$, this reduces to the usual condition $k/n \to 0$ as $n \to +\infty$ (see Devroye et al., 1996, Chap. 6.6 for instance), which is used to prove the universal consistency of the kNN classifier (Stone, 1977). The monotonicity of this upper upper bound with respect to k can seem somewhat unexpected. One could think that the two classifiers would become more and more "similar" to each other as k increases enough. However it can be proved that, in some sense, this dependence cannot be improved in the present distribution-free framework (see Proposition 5.1 and Figure 1).

Note that an upper bound similar to that of Ineq. (5.2) can be easily derived for any order-q moment ($q \geq 2$) at the price of increasing the constants by using $(a+b)^q \leq 2^{q-1}(a^q + b^q)$, for every $a, b \geq 0$. We also emphasize that Ineq. (5.2) allows us to control the discrepancy between the LpO estimator and the risk of the kNN classifier, that is the expectation of its classification error. Ideally we would have liked to replace the risk $R(\hat{f}_k)$ by the prediction error $L(\hat{f}_k)$. But with our strategy of proof, this would require an additional distribution-free concentration inequality on the prediction error of the kNN classifier. To the best of our knowledge, such a concentration inequality is not available up to now.

Upper bounding the squared difference between the LpO estimator and the prediction error is precisely the purpose of Ineq. (5.3). Proving the latter inequality requires a completely different strategy which can be traced back to an earlier proof by Rogers and Wagner

(1978, see the proof of Theorem 2.1) applying to the L1O estimator. Let us mention that Ineq. (5.3) combined with the Jensen inequality lead to a less accurate upper bound than Ineq. (5.1).

Finally the apparent difference between the upper bounds in Ineq. (5.2) and (5.3) results from the completely different schemes of proof. The first one allows us to derive general upper bounds for all centered moments of the LpO estimator, but exhibits a worse dependence with respect to k. By contrast the second one is exclusively dedicated to upper bounding the mean squared difference between the prediction error and the LpO estimator and leads to a smaller $\sqrt{k}$. However (even if probably not optimal), the upper bound used in Ineq. (5.2) still enables to achieve minimax rates over some Hölder balls as proved by Proposition 5.3.

5.2. Lower bounds

5.2.1. Bias of the L1O estimator

The purpose of the next result is to provide a counter-example highlighting that the upper bound of Eq. (5.1) cannot be improved in some sense. We consider the following discrete setting where $\mathcal{X} = \{0,1\}$ with $\pi_0 = P[X=0]$, and we define $\eta_0 = \mathbb{P}[Y=1 \mid X=0]$ and $\eta_1 = \mathbb{P}[Y=1 \mid X=1]$. In what follows this two-class generative model will be referred to as the discrete setting **DS**.

Note that (i) the 3 parameters π_0, η_0 and η_1 fully describe the joint distribution $P_{(X,Y)}$, and (ii) the distribution of **DS** satisfies the strong margin assumption of Massart and Nédélec (2006) if both η_0 and η_1 are chosen away from $1/2$. However this favourable setting has no particular effect on the forthcoming lower bound except a few simplifications along the calculations.

Proposition 5.1. *Let us consider the **DS** setting with $\pi_0 = 1/2$, $\eta_0 = 0$ and $\eta_1 = 1$, and assume that k is odd. Then there exists a numeric constant $C > 1$ independent of n and k such that, for all $n/2 \leq k \leq n-1$, the kNN classifiers $\mathcal{A}_k^{\mathcal{D}_n}$ and $\mathcal{A}_k^{\mathcal{D}_{n-1}}$ satisfy*

$$\mathbb{E}\left[L\left(\mathcal{A}_k^{\mathcal{D}_n}\right) - L\left(\mathcal{A}_k^{\mathcal{D}_{n-1}}\right)\right] \geq C\frac{\sqrt{k}}{n} \ .$$

The proof of Proposition 5.1 is provided in Appendix C.2. The rate $\sqrt{k}/n$ in the right-hand side of Eq. (5.1) is then achieved under the generative model **DS** for any $k \geq n/2$. As a consequence this rate cannot be improved without any additional assumption, for instance on the distribution of the X_is. See also Figure 1 below and related comments.

Empirical illustration

To further illustrate the result of Proposition 5.1, we simulated data according to **DS**, for different values of n ranging from 100 to 500 and different values of k ranging from 5 to $n-1$.

Figure 1 (a) displays the evolution of the absolute bias $\left|\mathbb{E}\left[L\left(\mathcal{A}_k^{\mathcal{D}_n}\right) - L\left(\mathcal{A}_k^{\mathcal{D}_{n-1}}\right)\right]\right|$ as a function of k, for several values of n (plain curves). The absolute bias is a nondecreasing function of k, as suggested by the upper bound provided in Eq. (5.1) which is also plotted (dashed lines) to ease the comparison. The non-decreasing behavior of the absolute bias is not always restricted to high values of k (w.r.t. n), as illustrated in Figure 1 (b) which

corresponds to **DS** with parameter values $(\pi_0, \eta_0, \eta_1) = (0.2, 0.2, 0.9)$. In particular the non-decreasing behavior of the absolute bias now appears for a range of values of k that are smaller than $n/2$.

Note that a rough idea about the location of the peak, denoted by k_{peak}, can be deduced as follows in the simple case where $\eta_0 = 0$ and $\eta_1 = 1$.

- For the peak to arise, the two classifiers (based on n and respectively $n-1$ observations) have to disagree the most strongly.
- This requires one of the two classifiers – say the first one – to have ties among the k nearest neighbors of each label in at least one of the two cases $X = 0$ or $X = 1$.
- With $\pi_0 < 0.5$, then ties will most likely occur for the case $X = 0$. Therefore the discrepancy between the two classifiers will be the highest at any new observation $x_0 = 0$.
- For the tie situation to arise at x_0, half of its neighbors have to be 1. This only occurs if (i) $k > n_0$ (with n_0 the number of observations such that $X = 0$ in the training set), and (ii) $k_0\eta_0 + k_1\eta_1 = k/2$, where k_0 (resp. k_1) is the number of neighbors of x_0 such that $X = 0$ (resp. $X = 1$).
- Since $k > n_0$, one has $k_0 = n_0$ and the last expression boils down to $k = \dfrac{n_0(\eta_1 - \eta_0)}{\eta_1 - 1/2}$.
- For large values of n, one should have $n_0 \approx n\pi_0$, that is the peak should appear at $k_{peak} \approx \dfrac{n\pi_0(\eta_1 - \eta_0)}{\eta_1 - 1/2}$.

In the setting of Proposition 5.1, this reasoning remarkably yields $k_{peak} \approx n$, while it leads to $k_{peak} \approx 0.4n$ in the setting of Figure 1 (b), which is close to the location of the observed peaks. This also suggests that even smaller values of k_{peak} can arise by tuning the parameter π_0 close to 0. Let us mention that very similar curves have been obtained for a Gaussian mixture model with two disjoint classes (not reported here). On the one hand this empirically illustrates that the $\sqrt{k}/n$ rate is not limited to **DS** (discrete setting). On the other hand, all of this confirms that this rate cannot be improved in the present distribution-free framework.

Let us finally consider Figure 1 (c), which displays the absolute bias as a function of n where $k = \lfloor \text{Coef} \times n \rfloor$ for different values of Coef, where $\lfloor \cdot \rfloor$ denotes the integer part. With this choice of k, Proposition 5.1 implies that the absolute bias should decrease at a $1/\sqrt{n}$ rate, which is supported by the plotted curves. By contrast, panel (d) of Figure 1 illustrates that choosing smaller values of k, that is $k = \lfloor \text{Coef} \times \sqrt{n} \rfloor$, leads to a faster decreasing rate.

5.2.2. Mean squared error

Following an example described by Devroye and Wagner (1979a), we now provide a lower bound on the minimal convergence rate of the mean squared error (see also Devroye et al., 1996, Chap. 24.4, p.415 for a similar argument).

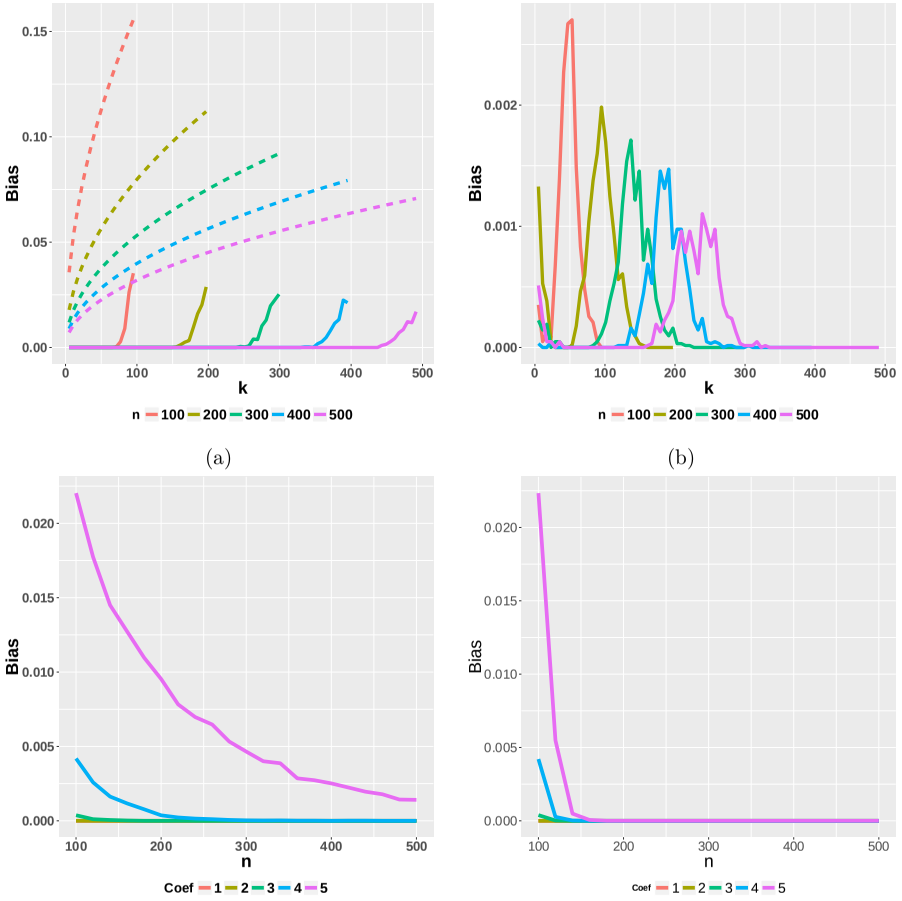


Figure 1: **(a)** Evolution of the absolute value of the bias as a function of k, for different values of n (plain lines). The dashed lines correspond to the upper bound obtained in (5.1). **(b)** Same as previous, except that data were generated according to the **DS** setting with parameters $(\pi_0, \eta_0, \eta_1) = (0.2, 0.2, 0.9)$. Upper bounds are not displayed in order to fit the scale of the absolute bias. **(c)** Evolution of the absolute value of the bias with respect to n, when k is chosen such that $k = \lfloor \text{Coef} \times n \rfloor$ ($\lfloor \cdot \rfloor$ denotes the integer part). The different colors correspond to different values of Coef. **(d)** Same as previous, except that k is chosen such that $k = \lfloor \text{Coef} \times \sqrt{n} \rfloor$.

Proposition 5.2. *Let us assume n is even, and that $P(Y=1 \mid X) = P(Y=1) = 1/2$ is independent of X. Then for $k = n-1$ (k odd), it results*

$$\mathbb{E}\left[\left(\widehat{R}_{1,n} - L(\hat{f}_k)\right)^2\right] = \int_0^1 2t \cdot \mathbb{P}\left[\left|\widehat{R}_{1,n} - L(\hat{f}_k)\right| > t\right] dt \geq \frac{1}{8\sqrt{\pi}} \cdot \frac{1}{\sqrt{n}} \ .$$

From the upper bound of order $\sqrt{k}/n$ provided by Ineq. (5.3) (with $p=1$), choosing $k = n-1$ leads to the same $1/\sqrt{n}$ rate as that of Proposition 5.2. This suggests that, at least for very large values of k, the $\sqrt{k}/n$ rate is of the right order and cannot be improved in the distribution-free framework.

5.3. Minimax rates

Let us conclude this section with a corollary, which provides a finite-sample bound on the gap between $\widehat{R}_{p,n}$ and $R(\hat{f}_k) = \mathbb{E}\left[L(\hat{f}_k)\right]$ with high probability. It is stated under the same restriction on p as the previous Theorem 5.1 it is based on, that is for $p \leq \sqrt{k}$.

Corollary 5.1. *With the notation of Theorems 4.2 and 5.1, let us assume $p, k \geq 1$ with $p \leq \sqrt{k}$, $\sqrt{k} + k \leq n$, and $p \leq n/2 + 1$. Then for every $x > 0$, there exists an event with probability at least $1 - 2e^{-x}$ such that*

$$\left|R(\hat{f}_k) - \widehat{R}_{p,n}\right| \leq \sqrt{\frac{\Delta^2 k^2}{n\left(1 - \frac{p-1}{n}\right)}x} + \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n} \ , \tag{5.4}$$

where $\hat{f}_k = \mathcal{A}_k^{\mathcal{D}_n}(\cdot)$.

Proof of Corollary 5.1. Ineq. (5.4) results from combining the exponential concentration result derived for $\widehat{R}_{p,n}$, namely Ineq. (4.2) (from Theorem 4.2) and the upper bound on the bias, that is Ineq. (5.1).

$$\begin{aligned}\left|R(\hat{f}_k) - \widehat{R}_{p,n}\right| &\leq \left|R(\hat{f}_k) - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right| + \left|\mathbb{E}\left[\widehat{R}_{p,n}\right] - \widehat{R}_{p,n}\right| \\ &\leq \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n} + \sqrt{\frac{\Delta^2 k^2}{n-p+1}x} \ .\end{aligned}$$

□

Note that the right-hand side of Ineq. (5.4) could be used to derive bounds on $R(\hat{f}_k)$ that seem similar to confidence bounds. However we do not recommend doing this in practice for several reasons. On the one hand, Ineq. (5.4) results from the repeated use of concentration inequalities where numeric constants are not optimized at all. This would lead to require a large sample size n for the deviation terms to be small in practice. On the other hand, explicit numeric constants such as Δ^2 in Corollary 5.1 exhibit a dependence on $\gamma_d \approx 4.8^d - 1$, which becomes exponentially large as d increases. Proving that this dependence can be weakened or not remains a completely open question at this stage. Nevertheless one can highlight that, for a given n, increasing d will quickly make the deviation term larger than 1, whereas both $R(\hat{f}_k)$ and $\widehat{R}_{p,n}$ belong to $[0,1]$.

The right-most term of order $\sqrt{k}/n$ in Ineq. (5.4) results from the bias. This is a necessary price to pay which cannot be improved in the present distribution-free framework according to Proposition 5.1. Besides combining the restriction $p \leq \sqrt{k}$ with the usual consistency constraint $k/n = o(1)$ leads to the conclusion that small values of p (w.r.t. n) have almost no effect on the convergence rate of the LpO estimator. Weakening the key restriction $p \leq \sqrt{k}$ would be necessary to potentially nuance this conclusion.

In order to highlight the interest of the above deviation inequality, let us deduce an optimality result in terms of minimax rate over Hölder balls $\mathcal{H}(\tau, \alpha)$ defined by

$$\mathcal{H}(\tau, \alpha) = \left\{g : \ \mathbb{R}^d \mapsto \mathbb{R}, \quad |g(x) - g(y)| \leq \tau \|x - y\|^\alpha\right\},$$

with $\alpha \in]0,1[$ and $\tau > 0$. In the following statement, Corollary 5.1 is used to prove that, uniformly with respect to k, the LpO estimator $\widehat{R}_{p,n}$ and the risk $R(\hat{f}_k)$ of the kNN classifier remain close to each other with high probability.

Proposition 5.3. *With the same notation as Corollary 5.1, for every $C > 1$ and $\theta > 0$, there exists an event of probability at least $1 - 2 \cdot n^{-(C-1)}$ on which, for any $p, k \geq 1$ such that $p \leq \sqrt{k}$, $k + \sqrt{k} \leq n$, and $p \leq n/2 + 1$, the LpO estimator of the kNN classifier satisfies*

$$\begin{aligned}&(1-\theta)\left[R(\hat{f}_k) - L^\star\right] - \frac{\theta^{-1}\Delta^2 C}{4}\frac{k^2\log(n)}{n\left(R(\hat{f}_k) - L^\star\right)} - \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n} \\ &\leq \widehat{R}_p(\mathcal{A}_k, \mathcal{D}_n) - L^\star \leq (1+\theta)\left[R(\hat{f}_k) - L^\star\right] + \frac{\theta^{-1}\Delta^2 C}{4}\frac{k^2\log(n)}{n\left(R(\hat{f}_k) - L^\star\right)} + \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n},\end{aligned} \tag{5.5}$$

where $L^\star$ denotes the classification error of the Bayes classifier.

Furthermore if one assumes the regression function η belongs to a Hölder ball $\mathcal{H}(\tau, \alpha)$ for some $\alpha \in]0, \min(d/4, 1)[$ (recall that $X_i \in \mathbb{R}^d$) and $\tau > 0$, then choosing $k = k^\star = k_0 \cdot n^{\frac{2\alpha}{2\alpha+d}}$ leads to

$$\widehat{R}_p(\mathcal{A}_{k^\star}, \mathcal{D}_n) - L^\star \sim_{n\to+\infty} R(\hat{f}_{k^\star}) - L^\star, \qquad a.s. \ . \tag{5.6}$$

Ineq. (5.5) gives a uniform control (over k) of the gap between the excess risk $R(\hat{f}_k) - L^\star$ and the corresponding LpO estimator $\widehat{R}_p(\hat{f}_k) - L^\star$ with high probability. The decreasing rate (in $n^{-(C-1)}$) of this probability is directly related to the $\log(n)$ factor in the lower and upper bounds. This decreasing rate could be made faster at the price of increasing the exponent of the $\log(n)$ factor. In a similar way the numeric constant θ has no precise meaning and can be chosen as close to 0 as we want, leading to increase one of the other deviation terms by a numeric factor θ^{-1}. For instance one could choose $\theta = 1/\log(n)$, which would replace the $\log(n)$ by a $(\log n)^2$.

The equivalence established by Eq. (5.6) results from knowing that this choice $k = k^\star$ makes the kNN classifier achieve the minimax rate $n^{-\frac{\alpha}{2\alpha+d}}$ over Hölder balls (Yang, 1999). This holds true for $\alpha \in]0,1[$ as long as $d \geq 4$. However if $d < 4$ the minimax rate is only achieved over $]0, d/4[$. This limitation results from the dependence of the deviation terms with respect to k^2 in Eq. (5.5), which is not optimal and should be further improved.

Proof of Proposition 5.3. Let us define $K \leq n$ as the maximum value of k and assume $x_k = C \cdot \log(n)$ (for some constant $C > 1$) for any $1 \leq k \leq K$. Let us also introduce the event

$$\Omega_n = \left\{ \forall 1 \leq k \leq K,\ \left| R(\hat{f}_k) - \widehat{R}_p(\mathcal{A}_k, \mathcal{D}_n) \right| \leq \sqrt{\Delta^2 \frac{k^2}{n} x_k} + \frac{4}{\sqrt{2\pi}} \frac{p\sqrt{k}}{n} \right\}.$$

Then $\mathbb{P}[\,\Omega_n^c\,] \leq \frac{1}{n^{C-1}} \to 0$, as $n \to +\infty$, since a union bound leads to

$$\sum_{k=1}^{K} e^{-x_k} = \sum_{k=1}^{K} e^{-C\cdot\log(n)} = K \cdot e^{-C\cdot\log(n)} \leq e^{-(C-1)\cdot\log(n)} = \frac{1}{n^{C-1}} \ .$$

Furthermore combining (for $a, b > 0$) the inequality $ab \leq a^2\theta^2 + b^2\theta^{-2}/4$ for every $\theta > 0$ with $\sqrt{a+b} \leq \sqrt{a} + \sqrt{b}$, it results that

$$\begin{aligned}\sqrt{\Delta^2 \frac{k^2}{n} x_k} &\leq \theta \left(R(\hat{f}_k) - L^\star \right) + \frac{\theta^{-1}}{4} \Delta^2 \frac{k^2}{n \left(R(\hat{f}_k) - L^\star \right)} x_k \\ &\leq \theta \left(R(\hat{f}_k) - L^\star \right) + \frac{\theta^{-1}}{4} \Delta^2 \frac{k^2}{n \left(R(\hat{f}_k) - L^\star \right)} C \cdot \log(n),\end{aligned}$$

hence Ineq. (5.5).

Let us now prove the next equivalence, namely (5.6), by means of the Borel-Cantelli lemma.

First Yang (1999) combined with Theorem 7 in Chaudhuri and Dasgupta (2014) provide that the minimax rate over the Hölder ball $\mathcal{H}(\tau, \alpha)$ is achieved by the kNN classifier with $k = k^\star$), that is

$$\left(R(\hat{f}_{k^\star}) - L^\star \right) \asymp n^{-\frac{\alpha}{2\alpha+d}},$$

where $a \asymp b$ means there exist numeric constants $l, u > 0$ such that $l \cdot b \leq a \leq u \cdot b$. Moreover it is then easy to check that

- $D_1 := \frac{\theta^{-1}}{4} \Delta^2 \frac{k^{\star 2}}{n\left(R(\hat{f}_{k^\star}) - L^\star \right)} C \cdot \log(n) \asymp \frac{C\theta^{-1}\Delta^2 k_0}{4} \cdot \left(n^{-\frac{d-3\alpha}{2\alpha+d}} \log(n) \right) = o_{n\to+\infty}\left(R(\hat{f}_{k^\star}) - L^\star \right)$,
- $D_2 := \frac{p\sqrt{k^\star}}{n} \leq \frac{k^\star}{n} = k_0 \cdot n^{-\frac{d}{2\alpha+d}} = o_{n\to+\infty}\left(R(\hat{f}_{k^\star}) - L^\star \right)$.

Besides

$$\begin{aligned}\frac{1}{n^{C-1}} \geq \mathbb{P}[\,\Omega_n^c\,] &\geq \mathbb{P}\left[\left| \frac{\left(\widehat{R}_p(\mathcal{A}_{k^\star}, \mathcal{D}_n) - L^\star \right) - \left(R(\hat{f}_{k^\star}) - L^\star \right)}{R(\hat{f}_{k^\star}) - L^\star} \right| > \theta + \frac{D_1 + D_2}{R(\hat{f}_{k^\star}) - L^\star} \right] \\ &= \mathbb{P}\left[\left| \frac{\left(\widehat{R}_p(\mathcal{A}_{k^\star}, \mathcal{D}_n) - L^\star \right)}{R(\hat{f}_{k^\star}) - L^\star} - 1 \right| > \theta + \frac{D_1 + D_2}{R(\hat{f}_{k^\star}) - L^\star} \right].\end{aligned}$$

Let us now choose any $\epsilon > 0$ and introduce the sequence of events $\{A_n(\epsilon)\}_{n\geq 1}$ such that

$$A_n(\epsilon) = \left\{ \left| \frac{\left(\widehat{R}_p(\mathcal{A}_{k^\star}, \mathcal{D}_n) - L^\star \right)}{R(\hat{f}_{k^\star}) - L^\star} - 1 \right| > \epsilon \right\}.$$

Using that D_1 and D_2 are negligible with respect to $R(\hat{f}_{k^\star}) - L^\star$ as $n \to +\infty$, there exists an integer $n_0 = n_0(\epsilon) > 0$ such that, for all $n \geq n_0$ and with $\theta = \epsilon/2$,

$$\theta + \frac{D_1 + D_2}{R(\hat{f}_{k^\star}) - L^\star} \leq \epsilon.$$

Hence

$$\mathbb{P}[\,A_n(\epsilon)\,] \leq \mathbb{P}\left[\left\{ \left| \frac{\left(\widehat{R}_p(\mathcal{A}_{k^\star}, \mathcal{D}_n) - L^\star \right)}{R(\hat{f}_{k^\star}) - L^\star} - 1 \right| > \theta + \frac{D_1 + D_2}{R(\hat{f}_{k^\star}) - L^\star} \right\} \right] \leq \frac{1}{n^{C-1}} \ .$$

Finally, choosing any $C > 2$ leads to $\sum_{n=1}^{+\infty} \mathbb{P}[\,A_n(\epsilon)\,] < +\infty$, which provides the expected conclusion by means of the Borel-Cantelli lemma. □

6. Discussion

The present work provides several new results quantifying the performance of the LpO estimator applied to the kNN classifier. By exploiting the connexion between LpO and U-statistics (Section 2), the polynomial and exponential inequalities derived in Sections 3 and 4 give some new insight on the concentration of the LpO estimator around its expectation for different regimes of p/n. In Section 5, these results serve for instance to conclude to the consistency of the LpO estimator towards the risk (or the classification error rate) of the kNN classifier (Theorem 5.1). They also allow us to establish the asymptotic equivalence between the LpO estimator (shifted by the Bayes risk $L^\star$) and the excess risk over some Hölder class of regression functions (Proposition 5.3).

It is worth mentioning that the upper-bounds derived in Sections 4 and 5 — see for instance Theorem 5.1 — can be minimized by choosing $p = 1$, suggesting that the L1O estimator is optimal in terms of risk estimation when applied to the kNN classification algorithm. This observation corroborates the results of the simulation study presented in Celisse and Mary-Huard (2011), where it is empirically shown that small values of p (and in particular $p = 1$) lead to the best estimation of the risk for any fixed k, whatever the level of noise in the data. The suggested optimality of L1O (for risk estimation) is also consistent with results by Burman (1989) and Celisse (2014), where it is proved that L1O is asymptotically the best cross-validation procedure to perform risk estimation in the context of low-dimensional regression and density estimation respectively.

Alternatively, the LpO estimator can also be used as a data-dependent calibration procedure to tune k, by choosing the value $\hat{k}_p$ which minimizes the LpO estimate. For instance

in classification, LpO can be used to get the value of k leading to the best prediction performance. In this context the value of p (the splitting ratio) leading to the best kNN classifier can be very different from $p = 1$. This is illustrated by the simulation results summarized by Figure 2 in Celisse and Mary-Huard (2011) where p has to be larger than 1 as the noise level becomes strong. This phenomenon is not limited to the kNN classifier, but extends to various estimation/prediction problems (Breiman and Spector, 1992; Arlot and Lerasle, 2012; Celisse, 2014). If we turn now to the question of identifying the best predictor among several candidates, choosing $p = 1$ also leads to poor selection performances as proved by Shao (1993, Eq. (3.8)) with the linear regression model. For the LpO, Shao (1997, Theorem 5) proves the model selection consistency if $p/n \to 1$ and $n - p \to +\infty$ as $n \to +\infty$. For recovering the best predictor among two candidates, Yang (2006, 2007) proved the consistency of CV under conditions relating the optimal splitting ratio p to the convergence rates of the predictors to be compared, and further requiring that $\min(p, n - p) \to +\infty$ as $n \to +\infty$.

Although the focus of the present paper is different, it is worth mentioning that the concentration results established in Section 4 are a significant early step towards deriving theoretical guarantees on LpO as a model selection procedure. Indeed, exponential concentration inequalities have been a key ingredient to assess model selection consistency or model selection efficiency in various contexts (see for instance Celisse (2014) or Arlot and Lerasle (2012) in the density estimation framework). Still theoretically investigating the behavior of $\hat{k}_p$ requires some further dedicated developments. One first step towards such results is to derive a tighter upper bound on the bias between the LpO estimator and the risk. The best known upper bound currently available is derived from Devroye and Wagner (1980, see Lemma D.6 in the present paper). Unfortunately it does not fully capture the true behavior of the LpO estimator with respect to p (at least as p becomes large) and could be improved in particular for $p > \sqrt{k}$ as emphasized in the comments following Theorem 5.1. Another important direction for studying the model selection behavior of the LpO procedure is to prove a concentration inequality for the classification error rate of the kNN classifier around its expectation. While such concentration results have been established for the kNN algorithm in the (fixed-design) regression framework (Arlot and Bach, 2009), deriving similar results in the classification context remains a challenging problem to the best of our knowledge.

Acknowledgments

The authors would like to thank the associate editor and reviewers for their highlightful comments which greatly helped to improve the presentation of the paper. This work was partially funded by the "BeFast" PEPS CNRS.

Appendix A. Proofs of polynomial moment upper bounds

A.1. Proof of Theorem 2.2

The proof relies on Proposition 2.1 that allows to relate the LpO estimator to a sum of independent random variables. In the following, we distinguish between the two settings $q = 2$ (where exact calculations can be carried out), and $q > 2$ where only upper bounds can be derived.

When $q > 2$, our proof deals separately with the cases $p \leq n/2 + 1$ and $p > n/2 + 1$. In the first one, a straightforward use of Jensen's inequality leads to the result. In the second setting, one has to be more cautious when deriving upper bounds. This is done by using the more sophisticated Rosenthal's inequality, namely Proposition D.2.

A.1.1. Exploiting Proposition 2.1

According to the proof of Proposition 2.1, it arises that the LpO estimator can be expressed as a U-statistic since

$$\widehat{R}_{p,n} = \frac{1}{n!} \sum_{\sigma} W\left(Z_{\sigma(1)}, \ldots, Z_{\sigma(n)}\right) \ ,$$

with

$$W\left(Z_1, \ldots, Z_n\right) = \left\lfloor \frac{n}{m} \right\rfloor^{-1} \sum_{a=1}^{\lfloor \frac{n}{m} \rfloor} h_m\left(Z_{(a-1)m+1}, \ldots, Z_{am}\right) \qquad (\text{with } m = n - p + 1)$$

$$\text{and} \quad h_m\left(Z_1, \ldots, Z_m\right) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\left\{\mathcal{A}^{\mathcal{D}_m^{(i)}}(X_i) \neq Y_i\right\}} = \widehat{R}_{1,n-p+1} \ ,$$

where $\mathcal{A}^{\mathcal{D}_m^{(i)}}(.)$ denotes the classifier based on sample $\mathcal{D}_m^{(i)} = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_m)$. Further centering the LpO estimator, it comes

$$\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right] = \frac{1}{n!} \sum_{\sigma} \bar{W}\left(Z_{\sigma(1)}, \ldots, Z_{\sigma(n)}\right),$$

where $\bar{W}(Z_1, \ldots, Z_n) = W(Z_1, \ldots, Z_n) - \mathbb{E}\left[W(Z_1, \ldots, Z_n)\right]$.

Then with $\bar{h}_m(Z_1, \ldots, Z_m) = h_m(Z_1, \ldots, Z_m) - \mathbb{E}\left[h_m(Z_1, \ldots, Z_m)\right]$, one gets

$$\begin{aligned}
\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] &\leq \mathbb{E}\left[\left|\bar{W}\left(Z_1, \ldots, Z_n\right)\right|^q\right] \quad (\text{Jensen}'\text{s inequality}) \\
&= \mathbb{E}\left[\left|\left\lfloor \frac{n}{m} \right\rfloor^{-1} \sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right] \qquad (A.1) \\
&= \left\lfloor \frac{n}{m} \right\rfloor^{-q} \mathbb{E}\left[\left|\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right] .
\end{aligned}$$

A.1.2. The setting $q = 2$

If $q = 2$, then by independence it comes

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] \leq \left\lfloor \frac{n}{m} \right\rfloor^{-2} \mathrm{Var}\left(\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} h_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right)$$
$$= \left\lfloor \frac{n}{m} \right\rfloor^{-2} \sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \mathrm{Var}\left[\, h_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right]$$
$$= \left\lfloor \frac{n}{m} \right\rfloor^{-1} \mathrm{Var}\left(\widehat{R}_1(\mathcal{A}, Z_{1,n-p+1})\right),$$

which leads to the result.

A.1.3. The setting $q > 2$

If $p \leq n/2 + 1$:

A straightforward use of Jensen's inequality from (A.1) provides

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] \leq \left\lfloor \frac{n}{m} \right\rfloor^{-1} \sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \mathbb{E}\left[\left|\bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right]$$
$$= \mathbb{E}\left[\left|\widehat{R}_{1,n-p+1} - \mathbb{E}\left[\widehat{R}_{1,n-p+1}\right]\right|^q\right].$$

If $p > n/2 + 1$: Let us now use Rosenthal's inequality (Proposition D.2) by introducing symmetric random variables $\zeta_1, \ldots, \zeta_{\lfloor n/m \rfloor}$ such that

$$\forall 1 \leq i \leq \lfloor n/m \rfloor, \quad \zeta_i = h_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right) - h_m\left(Z'_{(i-1)m+1}, \ldots, Z'_{im}\right),$$

where $Z'_1, \ldots, Z'_n$ are *i.i.d.* copies of $Z_1, \ldots, Z_n$. Then it comes for every $\gamma > 0$

$$\mathbb{E}\left[\left|\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right] \leq \mathbb{E}\left[\left|\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \zeta_i\right|^q\right],$$

which implies

$$\mathbb{E}\left[\left|\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right] \leq B(q,\gamma) \max\left\{\gamma \sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \mathbb{E}\left[\,|\zeta_i|^q\,\right], \left(\sqrt{\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \mathbb{E}\left[\,\zeta_i^2\,\right]}\right)^q\right\}.$$

Then using for every i that

$$\mathbb{E}\left[\,|\zeta_i|^q\right] \leq 2^q \mathbb{E}\left[\,\left|\bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right],$$

it comes

$$\mathbb{E}\left[\left|\sum_{i=1}^{\lfloor \frac{n}{m} \rfloor} \bar{h}_m\left(Z_{(i-1)m+1}, \ldots, Z_{im}\right)\right|^q\right]$$
$$\leq B(q,\gamma) \max\left(2^q \gamma \left\lfloor \frac{n}{m} \right\rfloor \mathbb{E}\left[\left|\widehat{R}_{1,m} - \mathbb{E}\left[\,\widehat{R}_{1,m}\,\right]\right|^q\right], \left(\sqrt{\left\lfloor \frac{n}{m} \right\rfloor 2\mathrm{Var}\left(\widehat{R}_{1,m}\right)}\right)^q\right).$$

Hence, it results for every $q > 2$

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right]$$
$$\leq B(q,\gamma) \max\left(2^q \gamma \left\lfloor \frac{n}{m} \right\rfloor^{-q+1} \mathbb{E}\left[\left|\widehat{R}_{1,m} - \mathbb{E}\left[\,\widehat{R}_{1,m}\,\right]\right|^q\right], \left\lfloor \frac{n}{m} \right\rfloor^{-q/2} \left(\sqrt{2\mathrm{Var}\left(\widehat{R}_{1,m}\right)}\right)^q\right),$$

which concludes the proof.

A.2. Proof of Theorem 3.1

Our strategy of proof follows several ideas. The first one consists in using Proposition 3.1 which says that, for every $q \geq 2$,

$$\left\|\bar{h}_m(Z_1, \ldots, Z_m)\right\|_q \leq \sqrt{2\kappa q} \sqrt{\left\|\sum_{j=1}^{m} \left(h_m(Z_1, \ldots, Z_m) - h_m(Z_1, \ldots, Z'_j, \ldots, Z_m)\right)^2\right\|_{q/2}},$$

where $h_m(Z_1, \ldots, Z_m) = \widehat{R}_{1,m}$ by Eq. (2.4), and $\bar{h}_m(Z_1, \ldots, Z_m) = h_m(Z_1, \ldots, Z_m) - \mathbb{E}\left[\, h_m(Z_1, \ldots, Z_m)\,\right]$. The second idea consists in deriving upper bounds of

$$\Delta^j h_m = h_m(Z_1, \ldots, Z_j, \ldots, Z_m) - h_m(Z_1, \ldots, Z'_j, \ldots, Z_m)$$

by repeated uses of Stone's lemma, that is Lemma D.5 which upper bounds by $k\gamma_d$ the maximum number of X_is that can have a given X_j among their k nearest neighbors. Finally, for technical reasons we have to distinguish the case $q = 2$ (where we get tighter bounds) and $q > 2$.

A.2.1. Upper bounding $\Delta^j h_m$

For the sake of readability let us now use the notation $\mathcal{D}^{(i)} = \mathcal{D}_m^{(i)}$ (see Theorem 2.1), and let $\mathcal{D}_j^{(i)}$ denote the set $\left(Z_1, \ldots, Z'_j, \ldots, Z_n\right)$ where the i-th coordinate has been removed. Then, $\Delta^j h_m = h_m(Z_1, \ldots, Z_m) - h_m(Z_1, \ldots, Z'_j, \ldots, Z_m)$ is now upper bounded by

$$\left|\Delta^j h_m\right| \leq \frac{1}{m} + \frac{1}{m} \sum_{i \neq j} \left| \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq Y_i\right\}} - \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i) \neq Y_i\right\}} \right|$$
$$\leq \frac{1}{m} + \frac{1}{m} \sum_{i \neq j} \left| \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}} \right|. \tag{A.2}$$

Furthermore, let us introduce for every $1 \leq j \leq n$,

$$A_j = \{1 \leq i \leq m,\ i \neq j,\ j \in V_k(X_i)\} \text{ and } A'_j = \left\{1 \leq i \leq m,\ i \neq j,\ j \in V'_k(X_i)\right\}$$

where $V_k(X_i)$ and $V'_k(X_i)$ denote the indices of the k nearest neighbors of X_i respectively among $X_1, \ldots, X_{j-1}, X_j, X_{j+1}, \ldots, X_m$ and $X_1, ..., X_{j-1}, X'_j, X_{j+1}, \ldots, X_m$. Setting $B_j = A_j \cup A'_j$, one obtains

$$\left|\Delta^j h_m\right| \leq \frac{1}{m} + \frac{1}{m} \sum_{i \in B_j} \left| \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}} \right|. \tag{A.3}$$

From now on, we distinguish between $q = 2$ and $q > 2$ because we will be able to derive a tighter bound for $q = 2$ than for $q > 2$.

A.2.2. Case $q > 2$

From (A.3), Stone's lemma (Lemma D.5) provides

$$\left|\Delta^j h_m\right| \leq \frac{1}{m} + \frac{1}{m} \sum_{i \in B_j} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}} \leq \frac{1}{m} + \frac{2k\gamma_d}{m} \ .$$

Summing over $1 \leq j \leq n$ and applying $(a+b)^q \leq 2^{q-1}\left(a^q + b^q\right)$ $(a, b \geq 0$ and $q \geq 1)$, it comes

$$\sum_j \left(\Delta^j h_m\right)^2 \leq \frac{2}{m}\left(1 + (2k\gamma_d)^2\right) \leq \frac{4}{m}(2k\gamma_d)^2 \ ,$$

hence

$$\left\| \sum_{j=1}^m \left(h_m(Z_1, \ldots, Z_m) - h_m(Z_1, \ldots, Z'_j, \ldots, Z_m)\right)^2 \right\|_{q/2} \leq \frac{4}{m}(2k\gamma_d)^2.$$

This leads for every $q > 2$ to

$$\left\|\bar{h}_m(Z_1, \ldots, Z_m)\right\|_q \leq q^{1/2}\sqrt{2\kappa}\frac{4k\gamma_d}{\sqrt{m}} \ ,$$

which enables to conclude.

A.2.3. Case $q = 2$

It is possible to obtain a slightly better upper bound in the case $q = 2$ with the following reasoning. With the same notation as above and from (A.3), one has

$$\begin{aligned}
\mathbb{E}\left[\left(\Delta^j h_m\right)^2\right] &= \frac{2}{m^2} + \frac{2}{m^2}\mathbb{E}\left[\left(\sum_{i \in B_j} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}}\right)^2\right] \qquad \text{(using } \mathbb{1}_{\{\cdot\}} \leq 1\text{)} \\
&\leq \frac{2}{m^2} + \frac{2}{m^2}\mathbb{E}\left[|B_j| \sum_{i \in B_j} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}}\right].
\end{aligned}$$

Lemma D.5 implies $|B_j| \leq 2k\gamma_d$, which allows to conclude

$$\mathbb{E}\left[\left(\Delta^j h_m\right)^2\right] \leq \frac{2}{m^2} + \frac{4k\gamma_d}{m^2}\mathbb{E}\left[\sum_{i \in B_j} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}}\right].$$

Summing over j and introducing an independent copy of Z_1 denoted by Z_0, one derives

$$\begin{aligned}
&\sum_{j=1}^m \mathbb{E}\left[\left(h_m(Z_1, \ldots, Z_m) - h_m(Z_1, \ldots, Z'_j, \ldots, Z_m)\right)^2\right] \\
&\leq \frac{2}{m} + \frac{4k\gamma_d}{m}\sum_{i=1}^m \mathbb{E}\left[\mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)}}(X_i) \neq \mathcal{A}_k^{\mathcal{D}^{(i)} \cup Z_0}(X_i)\right\}} + \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^{(i)} \cup Z_0}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^{(i)}}(X_i)\right\}}\right] \\
&\leq \frac{2}{m} + 4k\gamma_d \times 2\frac{4\sqrt{k}}{\sqrt{2\pi}m} = \frac{2}{m} + \frac{32\gamma_d}{\sqrt{2\pi}}\frac{k\sqrt{k}}{m} \leq (2 + 16\gamma_d)\frac{k\sqrt{k}}{m} \ ,
\end{aligned} \tag{A.4}$$

where the last but one inequality results from Lemma D.6.

A.3. Proof of Theorem 3.2

The idea is to plug the upper bounds previously derived for the L1O estimator, namely Ineq. (2.5) and (2.6) from Theorem 2.2, in the inequalities proved for the moments of the LpO estimator in Theorem 2.2.

Proof of Ineq. (3.3), (3.4), and (3.5): These inequalities straightforwardly result from the combination of Theorem 2.2 and Ineq. (2.5) and (2.6) from Theorem 3.1.

Proof of Ineq. (3.6): It results from the upper bounds proved in Theorem 3.1 and plugged in Ineq. (2.7) (derived from Rosenthal's inequality with optimized constant γ, namely Proposition D.3).

Then it comes

$$\begin{aligned}
&\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n}\right]\right|^q\right] \leq \left(2\sqrt{2e}\right)^q \times \\
&\max\left\{(\sqrt{q})^q\left(\sqrt{\left\lfloor\frac{n}{n-p+1}\right\rfloor^{-1} 2C_1\sqrt{k}\left(\frac{\sqrt{k}}{\sqrt{n-p+1}}\right)^2}\right)^q, q^q\left\lfloor\frac{n}{n-p+1}\right\rfloor^{-q+1}(2C_2\sqrt{q})^q\left(\frac{k}{\sqrt{n-p+1}}\right)^q\right\} \\
&= \left(2\sqrt{2e}\right)^q \times \\
&\max\left\{(\sqrt{q})^q\left(\sqrt{2C_1\sqrt{k}}\sqrt{\frac{k}{(n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor}}\right)^q, \left(q^{3/2}\right)^q\left\lfloor\frac{n}{n-p+1}\right\rfloor\left(2C_2\frac{k}{\left\lfloor\frac{n}{n-p+1}\right\rfloor\sqrt{n-p+1}}\right)^q\right\} \\
&\leq \left\lfloor\frac{n}{n-p+1}\right\rfloor \max\left\{\left(\lambda_1 q^{1/2}\right)^q, \left(\lambda_2 q^{3/2}\right)^q\right\},
\end{aligned}$$

with

$$\lambda_1 = 2\sqrt{2e}\sqrt{2C_1\sqrt{k}}\sqrt{\frac{k}{(n-p+1)\left\lfloor \frac{n}{n-p+1}\right\rfloor}}, \quad \lambda_2 = 2\sqrt{2e}2C_2\frac{k}{\left\lfloor \frac{n}{n-p+1}\right\rfloor \sqrt{n-p+1}} \ .$$

Finally introducing $\Gamma = 2\sqrt{2e}\max\left(2C_2, \sqrt{2C1}\right)$ provides the result.

Appendix B. Proofs of exponential concentration inequalities

B.1. Proof of Proposition 4.1

The proof relies on two successive ingredients: McDiarmid's inequality (Theorem D.3), and Stone's lemma (Lemma D.5).

First with $\mathcal{D}_n = \mathcal{D}$ and $\mathcal{D}_j = (Z_1, \ldots, Z_{j-1}, Z'_j, Z_{j+1}, \ldots, Z_n)$, let us start by upper bounding $\left|\widehat{R}_p\left(\mathcal{D}_n\right) - \widehat{R}_p\left(\mathcal{D}_j\right)\right|$ for every $1 \leq j \leq n$.

Using Eq. (2.2), one has

$$\begin{aligned}
&\left|\widehat{R}_p\left(\mathcal{D}\right) - \widehat{R}_p\left(\mathcal{D}_j\right)\right| \\
&\leq \frac{1}{p}\sum_{i=1}^{n}\binom{n}{p}^{-1}\sum_{e}\left|\mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i)\neq Y_i\right\}} - \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}_j^e}(X_i)\neq Y_i\right\}}\right|\mathbb{1}_{\{i\notin e\}} \\
&\leq \frac{1}{p}\sum_{i=1}^{n}\binom{n}{p}^{-1}\sum_{e}\mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i)\neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i)\right\}}\mathbb{1}_{\{i\notin e\}} \\
&\leq \frac{1}{p}\sum_{i\neq j}^{n}\binom{n}{p}^{-1}\sum_{e}\left[\mathbb{1}_{\left\{j\in V_k^{\mathcal{D}^e}(X_i)\right\}} + \mathbb{1}_{\left\{j\in V_k^{\mathcal{D}_j^e}(X_i)\right\}}\right]\mathbb{1}_{\{i\notin e\}} + \frac{1}{p}\binom{n}{p}^{-1}\sum_{e}\mathbb{1}_{\{j\notin e\}},
\end{aligned}$$

where $\mathcal{D}_j^e$ denotes the set of random variables among $\mathcal{D}_j$ having indices in e, and $V_k^{\mathcal{D}^e}(X_i)$ (resp. $V_k^{\mathcal{D}_j^e}(X_i)$) denotes the set of indices of the k nearest neighbors of X_i among $\mathcal{D}^e$ (resp. $\mathcal{D}_j^e$).

Second, let us now introduce

$$B_j^{\mathcal{E}_{n-p}} = \bigcup_{e\in\mathcal{E}_{n-p}}\left\{1\leq i\leq n,\ i\notin e\cup\{j\},\ V_k^{\mathcal{D}_j^e}(X_i)\ni j \text{ or } V_k^{\mathcal{D}^e}(X_i)\ni j\right\}.$$

Then Lemma D.5 implies $\mathrm{Card}(B_j^{\mathcal{E}_{n-p}}) \leq 2(k+p-1)\gamma_d$, hence

$$\left|\widehat{R}_p\left(\mathcal{D}_n\right) - \widehat{R}_p\left(\mathcal{D}_j\right)\right| \leq \frac{1}{p}\sum_{i\in B_j^{\mathcal{E}_{n-p}}}\binom{n}{p}^{-1}\sum_{e}2\cdot\mathbb{1}_{\{i\notin e\}} + \frac{1}{n} \leq \frac{4(k+p-1)\gamma_d}{n} + \frac{1}{n} \ .$$

The conclusion results from McDiarmid's inequality (Section D.1.5).

B.2. Proof of Theorem 4.1

In this proof, we use the same notation as in that of Proposition 4.1.

The goal of the proof is to provide a refined version of previous Proposition 4.1 by taking into account the status of each X_j as one of the k nearest neighbors of a given X_i (or not).

To do so, our strategy is to prove a sub-Gaussian concentration inequality by use of Lemma D.2, which requires the control of the even moments of the LpO estimator $\widehat{R}_p$.

Such upper bounds are derived

- First, by using Ineq. (D.4) (generalized Efron-Stein inequality), which amounts to control the q-th moments of the differences

$$\widehat{R}_p(\mathcal{D}) - \widehat{R}_p\left(\mathcal{D}_j\right).$$

- Second, by precisely evaluating the contribution of each neighbor X_i of a given X_j, that is by computing quantities such as $\mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\right]$, where $\mathbb{P}_e[\cdot]$ denotes the probability measure with respect to the uniform random variable e over $\mathcal{E}_{n-p}$, and $V_k^{\mathcal{D}^e}(X_i)$ denotes the indices of the k nearest neighbors of X_i among $X^e = \{X_\ell, \ell \in e\}$.

B.2.1. Upper bounding $\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j)$

For every $1 \le j \le n$, one gets

$$\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j) = \binom{n}{p}^{-1} \sum_e \left\{ \mathbb{1}_{\{j\in\bar{e}\}} \frac{1}{p} \left(\mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_j) \neq Y_j\right\}} - \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X'_j) \neq Y'_j\right\}} \right) \right.$$
$$\left. + \mathbb{1}_{\{j\in e\}} \frac{1}{p} \sum_{i\in\bar{e}} \left(\mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq Y_i\right\}} - \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}_j^e}(X_i) \neq Y_i\right\}} \right) \right\}.$$

Absolute values and Jensen's inequality then provide

$$\begin{aligned}
\left|\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j)\right| &\le \binom{n}{p}^{-1} \sum_e \left\{ \mathbb{1}_{\{j\in\bar{e}\}} \frac{1}{p} + \mathbb{1}_{\{j\in e\}} \frac{1}{p} \sum_{i\in\bar{e}} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i)\right\}} \right\} \\
&\le \frac{1}{n} + \binom{n}{p}^{-1} \sum_e \mathbb{1}_{\{j\in e\}} \frac{1}{p} \sum_{i\in\bar{e}} \mathbb{1}_{\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i)\right\}} \\
&= \frac{1}{n} + \frac{1}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ \mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i) \right].
\end{aligned}$$

where the notation $\mathbb{P}_e$ means the integration is carried out with respect to the random variable $e \in \mathcal{E}_{n-p}$, which follows a discrete uniform distribution over the set $\mathcal{E}_{n-p}$ of all $n-p$ distinct indices among $\{1,\dots,n\}$.

Let us further notice that $\left\{\mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i)\right\} \subset \left\{ j \in V_k^{\mathcal{D}^e}(X_i) \cup V_k^{\mathcal{D}_j^e}(X_i) \right\}$, where $V_k^{\mathcal{D}_j^e}(X_i)$ denotes the set of indices of the k nearest neighbors of X_i among $\mathcal{D}_j^e$ with the notation of the proof of Proposition 4.1. Then it results

$$\begin{aligned}
&\sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ \mathcal{A}_k^{\mathcal{D}^e}(X_i) \neq \mathcal{A}_k^{\mathcal{D}_j^e}(X_i) \right] \\
&\le \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \cup V_k^{\mathcal{D}_j^e}(X_i) \right] \\
&\le \sum_{i=1}^n \left(\mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] + \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \cup V_k^{\mathcal{D}_j^e}(X_i) \right] \right) \\
&\le 2 \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right],
\end{aligned}$$

which leads to

$$\left|\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j)\right| \le \frac{1}{n} + \frac{2}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right].$$

Summing over $1 \le j \le n$ the square of the above quantity, it results

$$\begin{aligned}
\sum_{j=1}^n \left(\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j)\right)^2 &\le \sum_{j=1}^n \left\{ \frac{1}{n} + \frac{2}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 \\
&\le 2 \sum_{j=1}^n \frac{1}{n^2} + 2 \left\{ \frac{2}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 \\
&\le \frac{2}{n} + 8 \sum_{j=1}^n \left\{ \frac{1}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2.
\end{aligned}$$

B.2.2. Evaluating the influence of each neighbor

Further using that

$$\begin{aligned}
&\sum_{j=1}^n \left(\frac{1}{p} \sum_{i=1}^n \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right)^2 \\
&= \sum_{j=1}^n \frac{1}{p^2} \sum_{i=1}^n \left(\mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right)^2 + \\
&\quad \sum_{j=1}^n \frac{1}{p^2} \sum_{1\le i\neq \ell\le n} \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_\ell) \right] \\
&= \quad T1 \quad + \quad T2\ ,
\end{aligned}$$

let us now successively deal with each of these two terms.

Upper bound on $T1$

First, we start by partitioning the sum over j depending on the rank of X_j as a neighbor of X_i in the whole sample $(X_1,\dots,X_n)$. It comes

$$\begin{aligned}
&= \sum_{j=1}^n \sum_{i=1}^n \left\{ \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 \\
&= \sum_{i=1}^n \left(\sum_{j\in V_k(X_i)} \left\{ \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 + \sum_{j\in V_{k+p}(X_i)\setminus V_k(X_i)} \left\{ \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 \right).
\end{aligned}$$

Then Lemma D.4 leads to

$$\begin{aligned}
&\sum_{j\in V_k(X_i)} \left\{ \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 + \sum_{j\in V_{k+p}(X_i)\setminus V_k(X_i)} \left\{ \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \right\}^2 \\
&\le \sum_{j\in V_k(X_i)} \left(\frac{p}{n} \frac{n-p}{n-1} \right)^2 + \sum_{j\in V_{k+p}(X_i)\setminus V_k(X_i)} \mathbb{P}_e\left[j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i) \right] \frac{p}{n} \frac{n-p}{n-1} \\
&= k \left(\frac{p}{n} \frac{n-p}{n-1} \right)^2 + \frac{kp}{n} \frac{p-1}{n-1} \frac{p}{n} \frac{n-p}{n-1} = k \left(\frac{p}{n} \right)^2 \frac{n-p}{n-1}\ ,
\end{aligned}$$

where the upper bound results from $\sum_j a_j^2 \leq (\max_j a_j)\sum_j a_j$, for $a_j \geq 0$. It results

$$T1 = \frac{1}{p^2}\sum_{j=1}^{n}\sum_{i=1}^{n}\left\{\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\right\}^2 \leq \frac{1}{p^2} n\left[k\left(\frac{p}{n}\right)^2 \frac{n-p}{n-1}\right] = \frac{k}{n}\frac{n-p}{n-1}\ .$$

Upper bound on $T2$

Let us now apply the same idea to the second sum, partitioning the sum over j depending on the rank of j as a neighbor of ℓ in the whole sample. Then,

$$\begin{aligned}
T2 &= \frac{1}{p^2}\sum_{j=1}^{n}\sum_{1\leq i\neq \ell\leq n}\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\mathbb{P}_e\left[\, j \in e,\ \ell \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_\ell)\,\right] \\
&\leq \frac{1}{p^2}\sum_{i=1}^{n}\sum_{\ell\neq i}\sum_{j\in V_k(X_\ell)}\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\frac{p}{n}\frac{n-p}{n-1} \\
&\quad + \frac{1}{p^2}\sum_{i=1}^{n}\sum_{\ell\neq i}\sum_{j\in V_{k+p}(X_\ell)\setminus V_k(X_\ell}\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\frac{kp}{n}\frac{p-1}{n-1}\ .
\end{aligned}$$

We then apply Stone's lemma (Lemma D.5) to get

$$\begin{aligned}
&T2 \\
&= \frac{1}{p^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\left(\sum_{\ell\neq i}\mathbb{1}_{j\in V_k(X_\ell)}\frac{p}{n}\frac{n-p}{n-1} + \sum_{\ell\neq i}\mathbb{1}_{j\in V_{k+p}(X_\ell)\setminus V_k(X_\ell}\frac{kp}{n}\frac{p-1}{n-1}\right) \\
&\leq \frac{1}{p^2}\sum_{i=1}^{n}\frac{kp}{n}\left(k\gamma_d\frac{p}{n}\frac{n-p}{n-1} + (k+p)\gamma_d\frac{kp}{n}\frac{p-1}{n-1}\right) = \gamma_d\frac{k^2}{n}\left(\frac{n-p}{n-1} + (k+p)\frac{p-1}{n-1}\right) \\
&= \gamma_d\frac{k^2}{n}\left(1 + (k+p-1)\frac{p-1}{n-1}\right).
\end{aligned}$$

Gathering the upper bounds

The two previous bounds provide

$$\begin{aligned}
\sum_{j=1}^{n}\left\{\frac{1}{p}\sum_{i=1}^{n}\mathbb{P}_e\left[\, j \in e,\ i \in \bar{e},\ j \in V_k^{\mathcal{D}^e}(X_i)\,\right]\right\}^2 &= T1 + T2 \\
&\leq \frac{k}{n}\frac{n-p}{n-1} + \gamma_d\frac{k^2}{n}\left(1 + (k+p-1)\frac{p-1}{n-1}\right),
\end{aligned}$$

which enables to conclude

$$\begin{aligned}
&\sum_{j=1}^{n}\left(\widehat{R}_p(\mathcal{D}) - \widehat{R}_p(\mathcal{D}_j)\right)^2 \\
&\leq \frac{2}{n}\left(1 + 4k + 4k^2\gamma_d\left[1 + (k+p)\frac{p-1}{n-1}\right]\right) \leq \frac{8k^2(1+\gamma_d)}{n}\left[1 + (k+p)\frac{p-1}{n-1}\right].
\end{aligned}$$

B.2.3. Generalized Efron-Stein inequality

Then (D.4) provides for every $q \geq 1$

$$\left\|\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right]\right\|_{2q} \leq 4\sqrt{\kappa q}\sqrt{\frac{8(1+\gamma_d)k^2}{n}\left[1 + (k+p)\frac{p-1}{n-1}\right]}.$$

Hence combined with $q! \geq q^q e^{-q}\sqrt{2\pi q}$, it comes

$$\begin{aligned}
\mathbb{E}\left[\left(\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right]\right)^{2q}\right] &\leq (16\kappa q)^q\left(\frac{8(1+\gamma_d)k^2}{n}\left[1 + (k+p)\frac{p-1}{n-1}\right]\right)^q \\
&\leq q!\left(16e\kappa\frac{8(1+\gamma_d)k^2}{n}\left[1 + (k+p)\frac{p-1}{n-1}\right]\right)^q.
\end{aligned}$$

The conclusion follows from Lemma D.2 with $C = 16e\kappa\frac{8(1+\gamma_d)k^2}{n}\left[1 + (k+p)\frac{p-1}{n-1}\right]$. Then for every $t > 0$,

$$\mathbb{P}\left(\widehat{R}_{p,n} - \mathbb{E}\left(\widehat{R}_{p,n}\right) > t\right) \vee \mathbb{P}\left(\mathbb{E}\left(\widehat{R}_{p,n}\right) - \widehat{R}_{p,n} > t\right) \leq \exp\left(-\frac{nt^2}{1024e\kappa k^2(1+\gamma_d)\left[1 + (k+p)\frac{p-1}{n-1}\right]}\right).$$

B.3. Proof of Theorem 4.2 and Proposition 4.2

B.3.1. Proof of Theorem 4.2

If $p < n/2 + 1$:
In what follows, we exploit a characterization of sub-Gaussian random variables by their $2q$-th moments (Lemma D.2).

From (3.3) and (3.4) applied with $2q$, and further introducing a constant $\Delta = 4\sqrt{e}\max\left(\sqrt{C_1/2}, C_2\right) > 0$, it comes for every $q \geq 1$

$$\mathbb{E}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right]\right|^{2q}\right] \leq \left(\frac{\Delta^2}{16e}\frac{k^2}{n-p+1}\right)^q(2q)^q \leq \left(\frac{\Delta^2}{8}\frac{k^2}{n-p+1}\right)^q q!\ , \qquad \text{(B.1)}$$

with $q^q \leq q!e^q/\sqrt{2\pi q}$. Then Lemma D.2 provides for every $t > 0$

$$\mathbb{P}\left(\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right] > t\right) \vee \mathbb{P}\left(\mathbb{E}\left[\,\widehat{R}_{p,n}\,\right] - \widehat{R}_{p,n} > t\right) \leq \exp\left(-(n-p+1)\frac{t^2}{\Delta^2 k^2}\right).$$

If $p \geq n/2 + 1$:
This part of the proof relies on Proposition D.1 which provides an exponential concentration inequality from upper bounds on the moments of a random variable.

Let us now use (3.3) and (3.6) combined with (D.1), where $C = \left\lfloor\frac{n}{n-p+1}\right\rfloor$, $q_0 = 2$, and $\min_j \alpha_j = 1/2$. This provides for every $t > 0$

$$\begin{aligned}
&\mathbb{P}\left[\left|\widehat{R}_{p,n} - \mathbb{E}\left[\,\widehat{R}_{p,n}\,\right]\right| > t\right] \leq \left\lfloor\frac{n}{n-p+1}\right\rfloor e\times \\
&\exp\left[-\frac{1}{2e}\min\left\{(n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor\frac{t^2}{4\Gamma^2 k\sqrt{k}},\ \left((n-p+1)\left\lfloor\frac{n}{n-p+1}\right\rfloor\right)^2\frac{t^2}{4\Gamma^2 k^2}\right)^{1/3}\right\}\right],
\end{aligned}$$

where Γ arises from Eq. (3.6).

B.3.2. Proof of Proposition 4.2

As in the previous proof, the derivation of the deviation terms results from Proposition D.1.

With the same notation and reasoning as in the previous proof, let us combine (3.3) and (3.6). From (D.2) of Proposition D.1 where $C = \left\lfloor \frac{n}{n-p+1} \right\rfloor$, $q_0 = 2$, and $\min_j \alpha_j = 1/2$, it results for every $t > 0$

$$\mathbb{P}\left[\left| \widehat{R}_{p,n} - \mathbb{E}\left[\widehat{R}_{p,n} \right] \right| > \Gamma \sqrt{\frac{2e}{(n-p+1)}} \left(\sqrt{\frac{k^{3/2}}{\left\lfloor \frac{n}{n-p+1} \right\rfloor} t} + 2e \frac{k}{\left\lfloor \frac{n}{n-p+1} \right\rfloor} t^{3/2} \right) \right] \leq \left\lfloor \frac{n}{n-p+1} \right\rfloor e \cdot e^{-t},$$

where $\Gamma > 0$ is given by Eq. (3.6).

Appendix C. Proofs of deviation upper bounds

C.1. Proof of Ineq. (5.3) in Theorem 5.1

The proof follows the same strategy as that of Theorem 2.1 in Rogers and Wagner (1978).

Along the proof, we will repeatedly use some notation that we briefly introduce here. First, let us define $Z_0 = (X_0, Y_0)$ and $Z_{n+1} = (X_{n+1}, Y_{n+1})$ that are independent copies of Z_1. Second to ease the reading of the proof, we also use several shortcuts: $\widehat{f}_k(X_0) = \mathcal{A}_k^{\mathcal{D}_n}(X_0)$, and $\widehat{f}_k^e(X_0) = \mathcal{A}_k^{\mathcal{D}^e}(X_0)$ for every set of indices $e \in \mathcal{E}_{n-p}$ (with cardinality $n-p$).

Finally along the proof, $e, e' \in \mathcal{E}_{n-p}$ denote two *random variables* which are sets of distinct indices *with discrete uniform distribution over* $\mathcal{E}_{n-p}$. The notation $\mathbb{P}_e$ (resp. $\mathbb{P}_{e,e'}$) means the integration is made with respect to the sample $\mathcal{D}$ and also the random variable e (resp. $\mathcal{D}$ and also the random variables e, e'). $\mathbb{E}_e[\cdot]$ and $\mathbb{E}_{e,e'}[\cdot]$ are teh corresponding expectations. Note that the sample $\mathcal{D}$ and the random variables e, e' are independent from each other, so that computing for instance $\mathbb{P}_e(i \notin e)$ amounts to integrating with respect to the random variable e only.

C.1.1. Main part of the proof

With the notation $L_n = L(\mathcal{A}_k^{\mathcal{D}_n})$, let us start from

$$\mathbb{E}\left[(\widehat{R}_{p,n} - L_n)^2 \right] = \mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n}) \right] + \mathbb{E}\left[L_n^2 \right] - 2\mathbb{E}\left[\widehat{R}_{p,n} L_n \right],$$

let us notice that

$$\mathbb{E}\left[L_n^2 \right] = \mathbb{P}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k(X_{n+1}) \neq Y_{n+1} \right),$$

and

$$\mathbb{E}\left[\widehat{R}_{p,n} L_n \right] = \mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_i) \neq Y_i | \ i \notin e \right) \mathbb{P}_e(i \notin e).$$

It immediately comes

$$\begin{aligned}
&\mathbb{E}\left[(\widehat{R}_{p,n} - L_n)^2 \right] \\
&= \mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n}) \right] - \mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_i) \neq Y_i \mid \ i \notin e \right) \mathbb{P}_e(i \notin e) \qquad \text{(C.1)} \\
&+ \left[\mathbb{P}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k(X_{n+1}) \neq Y_{n+1} \right) - \mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_i) \neq Y_i | \ i \notin e \right) \mathbb{P}_e(i \notin e) \right]. \qquad \text{(C.2)}
\end{aligned}$$

The proof then consists in successively upper bounding the two terms (C.1) and (C.2) of the last equality.

Upper bound of (C.1)

First, we have

$$\begin{aligned}
p^2 \mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n}) \right] &= \sum_{i,j} \mathbb{E}_{e,e'}\left[\mathbb{1}_{\left\{ \widehat{f}_k^e(X_i) \neq Y_i \right\}} \mathbb{1}_{\{i \notin e\}} \mathbb{1}_{\left\{ \widehat{f}_k^{e'}(X_j) \neq Y_j \right\}} \mathbb{1}_{\{j \notin e'\}} \right] \\
&= \sum_{i} \mathbb{E}_{e,e'}\left[\mathbb{1}_{\left\{ \widehat{f}_k^e(X_i) \neq Y_i \right\}} \mathbb{1}_{\{i \notin e\}} \mathbb{1}_{\left\{ \widehat{f}_k^{e'}(X_i) \neq Y_i \right\}} \mathbb{1}_{\{i \notin e'\}} \right] \\
&\quad + \sum_{i \neq j} \mathbb{E}_{e,e'}\left[\mathbb{1}_{\left\{ \widehat{f}_k^e(X_i) \neq Y_i \right\}} \mathbb{1}_{\{i \notin e\}} \mathbb{1}_{\left\{ \widehat{f}_k^{e'}(X_j) \neq Y_j \right\}} \mathbb{1}_{\{j \notin e'\}} \right].
\end{aligned}$$

Let us now introduce the five following events where we emphasize e and e' are random variables with the discrete uniform distribution over $\mathcal{E}_{n-p}$:

$$\begin{aligned} S_i^0 &= \{i \notin e,\ i \notin e'\}, \\ S_{i,j}^1 &= \{i \notin e,\ j \notin e',\ i \notin e',\ j \notin e\}, \qquad S_{i,j}^2 = \{i \notin e,\ j \notin e',\ i \notin e',\ j \in e\}, \\ S_{i,j}^3 &= \{i \notin e,\ j \notin e',\ i \in e',\ j \notin e\}, \qquad S_{i,j}^4 = \{i \notin e,\ j \notin e',\ i \in e',\ j \in e\}. \end{aligned}$$

Then,

$$\begin{aligned} p^2\mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n})\right] &= \sum_i \mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_i) \neq Y_i,\ \widehat{f}_k^{e'}(X_i) \neq Y_i | S_i^0\right)\mathbb{P}_{e,e'}\left(S_i^0\right) \\ &\quad + \sum_{i\neq j}\sum_{\ell=1}^4 \mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_i) \neq Y_i,\ \widehat{f}_k^{e'}(X_i) \neq Y_i | S_{i,j}^\ell\right)\mathbb{P}_{e,e'}\left(S_{i,j}^\ell\right) \\ &= n\mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_1) \neq Y_1,\ \widehat{f}_k^{e'}(X_1) \neq Y_1 | S_1^0\right)\mathbb{P}_{e,e'}\left(S_1^0\right) \\ &\quad + n(n-1)\sum_{\ell=1}^4 \mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_1) \neq Y_1,\ \widehat{f}_k^{e'}(X_2) \neq Y_2 \mid S_{1,2}^\ell\right)\mathbb{P}_{e,e'}\left(S_{1,2}^\ell\right). \end{aligned}$$

Furthermore since

$$\frac{1}{p^2}\left[n\mathbb{P}_{e,e'}\left(S_1^0\right) + n(n-1)\sum_{\ell=1}^4 \mathbb{P}_{e,e'}\left(S_{1,2}^\ell\right)\right] = \frac{1}{p^2}\sum_{i,j}\mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e'\right) = 1,$$

it comes

$$\mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n})\right] - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_1) \neq Y_1\right) = \frac{n}{p^2}A + \frac{n(n-1)}{p^2}B, \quad \text{(C.3)}$$

where

$$\begin{aligned} A &= \left[\mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_1) \neq Y_1,\ \widehat{f}_k^{e'}(X_1) \neq Y_1 \mid S_1^0\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_1) \neq Y_1 \mid S_1^0\right)\right] \\ &\quad \times \mathbb{P}_{e,e'}\left(S_1^0\right), \end{aligned}$$

and

$$\begin{aligned} B &= \sum_{\ell=1}^4 \left[\mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_1) \neq Y_1,\ \widehat{f}_k^{e'}(X_2) \neq Y_2 \mid S_{1,2}^\ell\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_1) \neq Y_1 \mid S_{1,2}^\ell\right)\right] \\ &\quad \times \mathbb{P}_{e,e'}\left(S_{1,2}^\ell\right). \end{aligned}$$

• Upper bound for A:
To upper bound A, simply notice that:

$$A \leq \mathbb{P}_{e,e'}\left(S_i^0\right) \leq \mathbb{P}_{e,e'}\left(i \notin e,\ i \notin e'\right) \leq \left(\frac{p}{n}\right)^2.$$

• Upper bound for B:
To obtain an upper bound for B, one needs to upper bound

$$\mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_1) \neq Y_1,\ \widehat{f}_k^{e'}(X_2) \neq Y_2 \mid S_{1,2}^\ell\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_1) \neq Y_1 \mid S_{1,2}^\ell\right), \quad \text{(C.4)}$$

which depends on ℓ, i.e. on the fact that index 2 belongs or not to the training set e.

- If $2 \notin e$ (i.e. $\ell = 1$ or 3): Then, Lemma C.2 proves

$$\text{(C.4)} \leq \frac{4p\sqrt{k}}{\sqrt{2\pi}n}\ .$$

- If $2 \in e$ (i.e. $\ell = 2$ or 4): Then, Lemma C.3 settles

$$\text{(C.4)} \leq \frac{8\sqrt{k}}{\sqrt{2\pi}(n-p)} + \frac{4p\sqrt{k}}{\sqrt{2\pi}n}\ .$$

Combining the previous bounds and Lemma C.1 leads to

$$\begin{aligned} B &\leq \left(\frac{4p\sqrt{k}}{\sqrt{2\pi}n}\right)\left[\mathbb{P}_{e,e'}\left(S_{1,2}^1\right) + \mathbb{P}_{e,e'}\left(S_{1,2}^3\right)\right] + \left(\frac{8\sqrt{k}}{\sqrt{2\pi}(n-p)} + \frac{4p\sqrt{k}}{\sqrt{2\pi}n}\right)\left[\mathbb{P}_{e,e'}\left(S_{1,2}^2\right) + \mathbb{P}_{e,e'}\left(S_{1,2}^4\right)\right] \\ &\leq \frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{k}\left[\frac{p}{n}\left[\mathbb{P}_{e,e'}\left(S_{1,2}^1\right) + \mathbb{P}_{e,e'}\left(S_{1,2}^3\right)\right] + \left(\frac{2}{n-p} + \frac{p}{n}\right)\left[\mathbb{P}_{e,e'}\left(S_{1,2}^2\right) + \mathbb{P}_{e,e'}\left(S_{1,2}^4\right)\right]\right] \\ &\leq \frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{k}\left[\frac{p}{n}\mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e'\right) + \frac{2}{n-p}\left(\mathbb{P}_{e,e'}\left(S_{1,2}^2\right) + \mathbb{P}_{e,e'}\left(S_{1,2}^4\right)\right)\right] \\ &\leq \frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{k}\left[\frac{p}{n}\left(\frac{p}{n}\right)^2 + \frac{2}{n-p}\left(\frac{(n-p)p^2(p-1)}{n^2(n-1)^2} + \frac{(n-p)^2p^2}{n^2(n-1)^2}\right)\right] \\ &\leq \frac{2\sqrt{2}}{\sqrt{\pi}}\sqrt{k}\left(\frac{p}{n}\right)^2\left[\frac{p}{n} + \frac{2}{n-1}\right]. \end{aligned}$$

Back to Eq. (C.3), one deduces

$$\begin{aligned} \mathbb{E}\left[\widehat{R}_p^2(\mathcal{A}_k^{\mathcal{D}_n})\right] - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_1) \neq Y_1\right) &= \frac{n}{p^2}A + \frac{n(n-1)}{p^2}B \\ &\leq \frac{1}{n} + \frac{2\sqrt{2}}{\sqrt{\pi}}\frac{(p+2)\sqrt{k}}{n}\ . \end{aligned}$$

Upper bound of (C.2) First observe that

$$\mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_i) \neq Y_i \mid i \notin e\right) = \mathbb{P}_{e,e'}\left(\widehat{f}_k^{(-1)}(X_0) \neq Y_0, \widehat{f}_k^e(X_{n+1}) \neq Y_{n+1}\right)$$

where $\widehat{f}_k^{(-1)}$ is built on sample $(X_2, Y_2), ..., (X_{n+1}, Y_{n+1})$. One has

$$\begin{aligned} &\mathbb{P}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k(X_{n+1}) \neq Y_{n+1}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k^e(X_i) \neq Y_i \mid i \notin e\right) \\ =\ &\mathbb{P}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}_k(X_{n+1}) \neq Y_{n+1}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k^{(-1)}(X_0) \neq Y_0, \widehat{f}_k^e(X_{n+1}) \neq Y_{n+1}\right) \\ \leq\ &\mathbb{P}\left(\widehat{f}_k(X_0) \neq \widehat{f}_k^{(-1)}(X_0)\right) + \mathbb{P}_{e,e'}\left(\widehat{f}_k^e(X_{n+1}) \neq \widehat{f}_k(X_{n+1})\right) \\ \leq\ &\frac{4\sqrt{k}}{\sqrt{2\pi}n} + \frac{4p\sqrt{k}}{\sqrt{2\pi}n}\ , \end{aligned}$$

where we used Lemma D.6 again to obtain the last inequality.

Conclusion:

The conclusion simply results from combining bonds (C.1) and (C.2), which leads to

$$\mathbb{E}\left[\left(\widehat{R}_{p,n} - L_n\right)^2\right] \leq \frac{2\sqrt{2}}{\sqrt{\pi}} \frac{(2p+3)\sqrt{k}}{n} + \frac{1}{n} \ .$$

C.1.2. Combinatorial lemmas

All the lemmas of the present section are proved with the notation introduced at the beginning of Section C.1.

Lemma C.1. *For any $1 \leq i \neq j \leq n$,*

$$\mathbb{P}_{e,e'}\left(S^1_{i,j}\right) = \frac{\binom{n-2}{n-p}}{\binom{n}{n-p}} \times \frac{\binom{n-2}{n-p}}{\binom{n}{n-p}}, \qquad \mathbb{P}_{e,e'}\left(S^2_{i,j}\right) = \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \times \frac{\binom{n-p}{n-2}}{\binom{n}{n-p}} \ ,$$
$$\mathbb{P}_{e,e'}\left(S^3_{i,j}\right) = \frac{\binom{n-p}{n-2}}{\binom{n}{n-p}} \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}}, \qquad \mathbb{P}_{e,e'}\left(S^4_{i,j}\right) = \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \times \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \ .$$

Proof of Lemma C.1. Along the proof, we repeatedly exploit the independence of the random variables e and e', which are set of $n-p$ distinct indices with the discrete uniform distribution over $\mathcal{E}_{n-p}$.

Note also that an important ingredient is that the probability of each one of the following events does not depend on the particular choice of the indices (i,j), but only on the fact that $i \neq j$.

$$\begin{aligned}
\mathbb{P}_{e,e'}\left(S^1_{i,j}\right) &= \mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e',\ i \notin e',\ j \notin e\right) \\
&= \mathbb{P}_e\left(i \notin e,\ j \notin e\right)\mathbb{P}_{e'}\left(j \notin e',\ i \notin e'\right) = \frac{\binom{n-2}{n-p}}{\binom{n}{n-p}} \times \frac{\binom{n-2}{n-p}}{\binom{n}{n-p}} \ . \\
\mathbb{P}_{e,e'}\left(S^2_{i,j}\right) &= \mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e',\ i \notin e',\ j \in e\right) \\
&= \mathbb{P}_e\left(i \notin e,\ j \in e\right)\mathbb{P}_{e'}\left(j \notin e',\ i \notin e'\right) = \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \times \frac{\binom{n-p}{n-2}}{\binom{n}{n-p}} \ . \\
\mathbb{P}_{e,e'}\left(S^3_{i,j}\right) &= \mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e',\ i \in e',\ j \notin e\right) \\
&= \mathbb{P}_e\left(i \notin e,\ j \notin e\right)\mathbb{P}_{e'}\left(j \notin e',\ i \in e'\right) = \frac{\binom{n-p}{n-2}}{\binom{n}{n-p}} \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \ . \\
\mathbb{P}_{e,e'}\left(S^4_{i,j}\right) &= \mathbb{P}_{e,e'}\left(i \notin e,\ j \notin e',\ i \in e',\ j \in e\right) \\
&= \mathbb{P}_e\left(i \notin e,\ j \in e\right)\mathbb{P}_{e'}\left(j \notin e',\ i \in e'\right) = \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \times \frac{\binom{n-p-1}{n-2}}{\binom{n}{n-p}} \ .
\end{aligned}$$

□

Lemma C.2. *With the above notation, for $\ell \in \{1,3\}$, it comes*

$$\mathbb{P}_e\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \leq \frac{4p\sqrt{k}}{\sqrt{2\pi}n} \ .$$

Proof of Lemma C.2. First remind that as a test sample element Z_0 cannot belong to either e or e'. Consequently, an exhaustive formulation of

$$\mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) = \mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \ .$$

Then it results

$$\mathbb{P}_e\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \ = \ \mathbb{P}_e\left(\widehat{f}_k^{\,(2)}(X_2) \neq Y_2,\ \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right),$$

where $\widehat{f}_k^{\,(2)}$ is built on sample $(X_0,Y_0),(X_1,Y_1),(X_3,Y_3),...,(X_n,Y_n)$.

Hence Lemma D.6 implies

$$\begin{aligned}
&\mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \\
&= \mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k^{\,(2)}(X_2) \neq Y_2,\ \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \\
&\leq \mathbb{P}_{e,e'}\left(\left\{\widehat{f}^e_k(X_1) \neq Y_1\right\} \triangle \left\{\widehat{f}^e_k(X_1) \neq Y_1\right\} \mid S^\ell_{1,2}\right) + \mathbb{P}_{e,e'}\left(\left\{\widehat{f}_k^{\,(2)}(X_2) \neq Y_2\right\} \triangle \left\{\widehat{f}^{e'}_k(X_2) \neq Y_2\right\} \mid S^\ell_{1,2}\right) \\
&= \mathbb{P}_{e,e'}\left(\widehat{f}_k^{\,(2)}(X_2) \neq \widehat{f}^{e'}_k(X_2) \mid S^\ell_{1,2}\right) \leq \frac{4p\sqrt{k}}{\sqrt{2\pi}n} \ .
\end{aligned}$$

□

Lemma C.3. *With the above notation, for $\ell \in \{2,4\}$, it comes*

$$\begin{aligned}
&\mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \\
&\leq \frac{8\sqrt{k}}{\sqrt{2\pi}(n-p)} + \frac{4p\sqrt{k}}{\sqrt{2\pi}n} \ .
\end{aligned}$$

Proof of Lemma C.3. As for the previous lemma, first notice that

$$\mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) = \mathbb{P}_{e,e'}\left(\widehat{f}_k^{\,(2)}(X_2) \neq Y_2,\ \widehat{f}_k^{\,e_0}(X_1) \neq Y_1 \mid S^\ell_{1,2}\right),$$

where $\widehat{f}_k^{\,e_0}$ is built on sample e with observation (X_2,Y_2) replaced with (X_0,Y_0). Then

$$\begin{aligned}
&\mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k(X_0) \neq Y_0, \widehat{f}^e_k(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \\
&= \mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq Y_1,\ \widehat{f}^{e'}_k(X_2) \neq Y_2 \mid S^\ell_{1,2}\right) - \mathbb{P}_{e,e'}\left(\widehat{f}_k^{\,(2)}(X_2) \neq Y_2,\ \widehat{f}_k^{\,e_0}(X_1) \neq Y_1 \mid S^\ell_{1,2}\right) \\
&\leq \mathbb{P}_{e,e'}\left(\left\{\widehat{f}^e_k(X_1) \neq Y_1\right\} \triangle \left\{\widehat{f}_k^{\,e_0}(X_1) \neq Y_1\right\} \mid S^\ell_{1,2}\right) + \mathbb{P}_{e,e'}\left(\left\{\widehat{f}_k^{\,(2)}(X_2) \neq Y_2\right\} \triangle \left\{\widehat{f}^{e'}_k(X_2) \neq Y_2\right\} \mid S^\ell_{1,2}\right) \\
&= \mathbb{P}_{e,e'}\left(\widehat{f}^e_k(X_1) \neq \widehat{f}_k^{\,e_0}(X_1) \mid S^\ell_{1,2}\right) + \mathbb{P}_{e,e'}\left(\widehat{f}_k^{\,(2)}(X_2) \neq \widehat{f}^{e'}_k(X_2) \mid S^\ell_{1,2}\right) \leq \frac{8\sqrt{k}}{\sqrt{2\pi}(n-p)} + \frac{4p\sqrt{k}}{\sqrt{2\pi}n} \ .
\end{aligned}$$

□

C.2. Proof of Proposition 5.1

The bias of the L1O estimator is equal to

$$
\begin{aligned}
&\mathbb{E}\left[L\left(\mathcal{A}_k^{\mathcal{D}_n}\right) - L\left(\mathcal{A}_k^{\mathcal{D}_{n-1}}\right)\right] \\
&= -2\mathbb{E}\left[(\eta(X)-1/2)\left(\mathbb{E}\left[\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(X) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(X) \mid X_{(k+1)}(X), X\right] \mid X\right]\right)\right] \\
&= -2\mathbb{E}\left[(\eta(X)-1/2)\left(\mathbb{E}\left[\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(X) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(X) \mid X_{(k+1)}(X), X\right] \mid X\right]\right)\right] \\
&= 1/2\left\{\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(0) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(0) \mid X_{(k+1)}(0)=0, X=0\right]\mathbb{P}\left[X_{(k+1)}(0)=0 \mid X=0\right]\right. \\
&\qquad\left. +\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(0) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(0) \mid X_{(k+1)}(0)=1, X=0\right]\mathbb{P}\left[X_{(k+1)}(0)=1 \mid X=0\right]\right\} \\
&- 1/2\left\{\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(1) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(1) \mid X_{(k+1)}(1)=0, X=1\right]\mathbb{P}\left[X_{(k+1)}(1)=0 \mid X=1\right]\right. \\
&\qquad\left. +\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(1) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(1) \mid X_{(k+1)}(1)=1, X=1\right]\mathbb{P}\left[X_{(k+1)}(1)=1 \mid X=1\right]\right\},
\end{aligned}
$$

where $X_{(k+1)}(x)$ denotes the $k+1$-th neighbor of x.

Then, a few remarks lead to simplify the above expression.

- On the one hand it is easy to check that

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(0) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(0) \mid X_{(k+1)}(0)=0, X=0\right] \\
&= \mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(1) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(1) \mid X_{(k+1)}(1)=1, X=1\right] = 0,
\end{aligned}
$$

  since all of the $k+1$ nearest neighbors share the same label.

- On the other hand, let us notice

$$
\begin{aligned}
&\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(0) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(0) \mid X_{(k+1)}(0)=1, X=0\right] \\
&= \mathbb{P}\left[\mathcal{A}_k^{\mathcal{D}_n}(0)=1, \mathcal{A}_k^{\mathcal{D}_{n-1}}(0)=0 \mid X_{(k+1)}(0)=1, X=0\right] \\
&- \mathbb{P}\left[\mathcal{A}_k^{\mathcal{D}_n}(0)=0, \mathcal{A}_k^{\mathcal{D}_{n-1}}(0)=1 \mid X_{(k+1)}(0)=1, X=0\right].
\end{aligned}
$$

  Then knowing $X_{(k+1)}(X)$ and X are not equal implies the only way for $\mathcal{A}_k^{\mathcal{D}_n}$ and $\mathcal{A}_k^{\mathcal{D}_{n-1}}$ to differ is that the numbers of k nearest neighbors of each label are almost equal, that is either equal to $(k-1)/2$ or to $(k+1)/2$ (k is odd by assumption).

  With N_0^1 (respectively $\tilde{N}_0^1$) denoting the number of 1s among th k nearest neighbors of $X=0$ among $X_1,\dots,X_n$ (resp. $X_1,\dots,X_{n-1}$), the proof of Theorem 3 in Chaudhuri and Dasgupta (2014) leads to

$$
\begin{aligned}
&\mathbb{P}\left[\mathcal{A}_k^{\mathcal{D}_n}(0)=1, \mathcal{A}_k^{\mathcal{D}_{n-1}}(0)=0 \mid X_{(k+1)}(0)=1, X=0\right] \\
&= \mathbb{P}\left[n \in V_k(0), N_0^1 = (k+1)/2, \tilde{N}_0^1 = (k-1)/2 \mid X_{(k+1)}(0)=1, X=0\right] \\
&= \frac{k}{n} \times \mathbb{P}\left[\tilde{N}_0^1 = (k-1)/2 \mid N_0^1 = (k+1)/2, X_{(k+1)}(0)=1, X=0\right] \\
&\times \mathbb{P}\left[N_0^1 = (k+1)/2 \mid X_{(k+1)}(0)=1, X=0\right] \\
&= \frac{k}{n} \times \mathbb{P}\left[\mathcal{H}\left(\frac{k+1}{2}, \frac{k-1}{2}; 1\right) = 1\right] \cdot \eta_1 \times \binom{k}{(k+1)/2} \bar{\eta}^{(k+1)/2}\left(1-\bar{\eta}\right)^{(k-1)/2} \\
&= \frac{k+1}{2n} \times \eta_1 \times \binom{k}{(k+1)/2} \bar{\eta}^{(k+1)/2}\left(1-\bar{\eta}\right)^{(k-1)/2},
\end{aligned}
$$

  where $\mathcal{H}(a,b;c)$ denotes a hypergeometric random variable with a successes in a population of cardinality $a+b$, and c draws, and $\bar{\eta} = \pi_0\eta_0 + (1-\pi_0)\eta_1 = 1/2$.

  Following the same reasoning for $\mathbb{P}\left[\mathcal{A}_k^{\mathcal{D}_n}(0)=0, \mathcal{A}_k^{\mathcal{D}_{n-1}}(0)=1 \mid X_{(k+1)}(0)=1, X=0\right]$ and recalling that $\eta_0 = 0$ and $\eta_1 = 1$ by assumption, it results

$$
\mathbb{E}\left[\mathcal{A}_k^{\mathcal{D}_n}(0) - \mathcal{A}_k^{\mathcal{D}_{n-1}}(0) \mid X_{(k+1)}(0)=1, X=0\right] = -\frac{k+1}{2n} \times \binom{k}{(k+1)/2}(1/2)^k.
$$

- Similar calculations applied to $X=1$ finally lead to

$$
\begin{aligned}
\mathbb{E}\left[L\left(\mathcal{A}_k^{\mathcal{D}_n}\right) - L\left(\mathcal{A}_k^{\mathcal{D}_{n-1}}\right)\right] &= \frac{k+1}{2n} \times \binom{k}{(k+1)/2}(1/2)^k \times \mathbb{P}\left[X_{(k+1)}(0)=1 \mid X=0\right] \\
&= \frac{k+1}{2n} \times \binom{k}{(k+1)/2}(1/2)^k \times \mathbb{P}\left[\mathcal{B}(n,1/2) \le k\right].
\end{aligned}
$$

- The conclusion then follows from considering $k \ge n/2$ which entails that $\mathbb{P}\left[\mathcal{B}(n,1/2) \le k\right] \ge 1/2$, and also by noticing that

$$
\frac{k+1}{2n} \times \binom{k}{(k+1)/2}(1/2)^k \ge C\frac{\sqrt{k}}{n},
$$

  where denotes a numeric constant independent of n and k.

Appendix D. Technical results

D.1. Main inequalities

D.1.1. From moment to exponential inequalities

Proposition D.1 (see also Arlot (2007), Lemma 8.10)**.** *Let X denote a real valued random variable, and assume there exist $C \geq 1$, $\lambda_1, \ldots, \lambda_N > 0$, and $\alpha_1, \ldots, \alpha_N > 0$ ($N \in \mathbb{N}^*$) such that for every $q \geq q_0$,*

$$\mathbb{E}\left[\,|X|^q\,\right] \leq C\left(\sum_{i=1}^{N} \lambda_i q^{\alpha_i}\right)^q.$$

Then for every $t > 0$,

$$\mathbb{P}\left[\,|X| > t\,\right] \leq Ce^{q_0 \min_j \alpha_j} e^{-(\min_i \alpha_i)e^{-1}\min_j\left\{\left(\frac{t}{N\lambda_j}\right)^{\frac{1}{\alpha_j}}\right\}}, \tag{D.1}$$

Furthermore for every $x > 0$, it results

$$\mathbb{P}\left[\,|X| > \sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}\right] \leq Ce^{q_0 \min_j \alpha_j} \cdot e^{-x}. \tag{D.2}$$

Proof of Proposition D.1. By use of Markov's inequality applied to $|X|^q$ ($q > 0$), it comes for every $t > 0$

$$\mathbb{P}\left[\,|X| > t\,\right] \leq \mathbb{1}_{q \geq q_0} \frac{\mathbb{E}\left[\,|X|^q\,\right]}{t^q} + \mathbb{1}_{q < q_0} \leq \mathbb{1}_{q \geq q_0} C\left(\frac{\sum_{i=1}^{N} \lambda_i q^{\alpha_i}}{t}\right)^q + \mathbb{1}_{q < q_0}.$$

Now using the upper bound $\sum_{i=1}^{N} \lambda_i q^{\alpha_i} \leq N \max_i \left\{\lambda_i q^{\alpha_i}\right\}$ and choosing the particular value $\tilde{q} = \tilde{q}(t) = e^{-1}\min_j\left\{\left(\frac{t}{N\lambda_j}\right)^{\frac{1}{\alpha_j}}\right\}$, one gets

$$\begin{aligned}\mathbb{P}\left[\,|X| > t\,\right] &\leq \mathbb{1}_{\tilde{q} \geq q_0} C\left(\frac{\max_i\left\{N\lambda_i\left(e^{-\alpha_i}\min_j\left\{\left(\frac{t}{N\lambda_j}\right)^{\frac{1}{\alpha_j}}\right\}\right)^{\alpha_i}\right\}}{t}\right)^{\tilde{q}} + \mathbb{1}_{\tilde{q} < q_0} \\ &\leq \mathbb{1}_{\tilde{q} \geq q_0} Ce^{-(\min_i \alpha_i)\left[e^{-1}\min_j\left\{\left(\frac{t}{N\lambda_j}\right)^{\frac{1}{\alpha_j}}\right\}\right]} + \mathbb{1}_{\tilde{q} < q_0},\end{aligned}$$

which provides (D.1).

Let us now turn to the proof of (D.2). From $t^* = \sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}$ combined with $q^* = \frac{x}{\min_j \alpha_j}$, it arises for every $x > 0$

$$\frac{\sum_{i=1}^{N} \lambda_i (q^*)^{\alpha_i}}{t^*} = \frac{\sum_{i=1}^{N} \lambda_i \left(e^{-1}\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}}{\sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}} \leq \left(\max_k e^{-\alpha_k}\right) \frac{\sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}}{\sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}} = e^{-\min_k \alpha_k}.$$

Then,

$$C\left(\frac{\sum_{i=1}^{N} \lambda_i (q^*)^{\alpha_i}}{t^*}\right)^{q^*} \leq Ce^{-(\min_k \alpha_k)\frac{x}{\min_j \alpha_j}} = Ce^{-x}.$$

Hence,

$$\mathbb{P}\left[\,|X| > \sum_{i=1}^{N} \lambda_i \left(\frac{ex}{\min_j \alpha_j}\right)^{\alpha_i}\right] \leq Ce^{-x}\mathbb{1}_{q^* \geq q_0} + \mathbb{1}_{q^* < q_0} \leq Ce^{q_0 \min_j \alpha_j} \cdot e^{-x},$$

since $e^{q_0 \min_j \alpha_j} \geq 1$ and $-x + q_0 \min_j \alpha_j \geq 0$ if $q < q_0$.

□

D.1.2. Sub-Gaussian random variables

Lemma D.1 (Theorem 2.1 in Boucheron et al. (2013) first part)**.** *Any centered random variable X such that $\mathbb{P}(X > t) \vee \mathbb{P}(-X > t) \leq e^{-t^2/(2\nu)}$ satisfies*

$$\mathbb{E}\left[X^{2q}\right] \leq q!\,(4\nu)^q.$$

for all q in $\mathbb{N}_+$.

Lemma D.2 (Theorem 2.1 in Boucheron et al. (2013) second part)**.** *Any centered random variable X such that*

$$\mathbb{E}\left[X^{2q}\right] \leq q!C^q.$$

for some $C > 0$ and q in $\mathbb{N}_+$ satisfies $\mathbb{P}(X > t) \vee \mathbb{P}(-X > t) \leq e^{-t^2/(2\nu)}$ with $\nu = 4C$.

D.1.3. The Efron-Stein inequality

Theorem D.1 (Efron-Stein's inequality Boucheron et al. (2013), Theorem 3.1)**.** *Let $X_1, \ldots, X_n$ be independent random variables and let $Z = f(X_1, \ldots, X_n)$ be a square-integrable function. Then*

$$\mathrm{Var}(Z) \leq \sum_{i=1}^{n} \mathbb{E}\left[\left(Z - \mathbb{E}\left[Z \mid (X_j)_{j \neq i}\right]\right)^2\right] = \nu.$$

Moreover if $X_1', \ldots, X_n'$ denote independent copies of $X_1, \ldots, X_n$ and if we define for every $1 \leq i \leq n$

$$Z_i' = f\left(X_1, \ldots, X_i', \ldots, X_n\right),$$

then

$$\nu = \frac{1}{2}\sum_{i=1}^{n} \mathbb{E}\left[\left(Z - Z_i'\right)^2\right].$$

D.1.4. Generalized Efron-Stein's inequality

Theorem D.2 (Theorem 15.5 in Boucheron et al. (2013)). *Let $\xi_1, \dots, \xi_n$ be n independent Ξ-valued random variables, $f : \Xi^n \to \mathbb{R}$ denote a measurable function, and define $\zeta = f(\xi_1, \dots, \xi_n)$ and $\zeta_i' = f(\xi_1, \dots, \xi_i', \dots, \xi_n)$, with $\xi_1', \dots, \xi_n'$ independent copies of ξ_i. Furthermore let $V_+ = \mathbb{E}\left[\sum_{i=1}^n \left[(\zeta - \zeta_i')_+\right]^2 \mid \xi_1^n\right]$ and $V_- = \mathbb{E}\left[\sum_{i=1}^n \left[(\zeta - \zeta_i')_-\right]^2 \mid \xi_1^n\right]$. Then there exists a constant $\kappa \leq 1,271$ such that for all q in $[2, +\infty[$,*

$$\left\|(\zeta - \mathbb{E}\zeta)_+\right\|_q \leq \sqrt{2\kappa q \left\|V_+\right\|_{q/2}} \ , \qquad \text{and} \qquad \left\|(\zeta - \mathbb{E}\zeta)_-\right\|_q \leq \sqrt{2\kappa q \left\|V_-\right\|_{q/2}} \ .$$

Corollary D.1. *With the same notation, it comes*

$$\left\|\zeta - \mathbb{E}\zeta\right\|_q \leq \sqrt{2\kappa q}\sqrt{\left\|\sum_{i=1}^n (\zeta - \zeta_i')^2\right\|_{q/2}} \leq 2\sqrt{\kappa q}\sqrt{\left\|\sum_{i=1}^n (\zeta - \mathbb{E}\left[\zeta \mid (\xi_j)_{j\neq i}\right])^2\right\|_{q/2}} \ . \quad \text{(D.3)}$$

Moreover considering $\zeta^{(j)} = f(\xi_1, \dots, \xi_{j-1}, \xi_{j+1}, \dots, \xi_n)$ for every $1 \leq j \leq n$, it results

$$\left\|\zeta - \mathbb{E}\zeta\right\|_q \ \leq 2\sqrt{2\kappa q}\sqrt{\left\|\sum_{i=1}^n \left(\zeta - \zeta^{(j)}\right)^2\right\|_{q/2}} \ . \quad \text{(D.4)}$$

D.1.5. McDiarmid's inequality

Theorem D.3. *Let $X_1, ..., X_n$ be independent random variables taking values in a set A, and assume that $f : A^n \to \mathbb{R}$ satisfies*

$$\sup_{x_1,\dots,x_n,x_i'} \left|f(x_1, ..., x_i, ..., x_n) - f(x_1, ..., x_i', ..., x_n)\right| \leq c_i, \ 1 \leq i \leq n \ .$$

Then for all $\varepsilon > 0$, one has

$$\begin{aligned} \mathbb{P}\left(f(X_1, ..., X_n) - E\left[f(X_1, ..., X_n)\right] \geq \varepsilon\right) &\leq e^{-2\varepsilon^2/\sum_{i=1}^n c_i^2} \\ \mathbb{P}\left(E\left[f(X_1, ..., X_n)\right] - f(X_1, ..., X_n) \geq \varepsilon\right) &\leq e^{-2\varepsilon^2/\sum_{i=1}^n c_i^2} \end{aligned}$$

A proof can be found in Devroye et al. (1996) (see Theorem 9.2).

D.1.6. Rosenthal's inequality

Proposition D.2 (Eq. (20) in Ibragimov and Sharakhmetov (2002)). *Let $X_1, \dots, X_n$ denote independent real random variables with symmetric distributions. Then for every $q > 2$ and $\gamma > 0$,*

$$E\left[\left|\sum_{i=1}^n X_i\right|^q\right] \leq B(q,\gamma)\left\{\gamma \sum_{i=1}^n E\left[\left|X_i\right|^q\right] \vee \left(\sqrt{\sum_{i=1}^n E\left[X_i^2\right]}\right)^q\right\},$$

where $a \vee b = \max(a, b)$ $(a, b \in \mathbb{R})$, and $B(q,\gamma)$ denotes a positive constant only depending on q and γ. Furthermore, the optimal value of $B(q,\gamma)$ is given by

$$\begin{aligned} B^*(q,\gamma) &= 1 + \tfrac{E\left[|N|^q\right]}{\gamma} \qquad , \text{ if } \ 2 < q \leq 4, \\ &= \gamma^{-q/(q-1)} E\left[\left|Z - Z'\right|^q\right] \ , \text{ if } \ 4 < q, \end{aligned}$$

where N denotes a standard Gaussian variable, and Z, Z' are i.i.d. random variables with Poisson distribution $\mathcal{P}\left(\frac{\gamma^{1/(q-1)}}{2}\right)$.

Proposition D.3. *Let $X_1, \dots, X_n$ denote independent real random variables with symmetric distributions. Then for every $q > 2$,*

$$E\left[\left|\sum_{i=1}^n X_i\right|^q\right] \leq \left(2\sqrt{2e}\right)^q \max\left\{q^q \sum_{i=1}^n E\left[\left|X_i\right|^q\right], (\sqrt{q})^q \left(\sqrt{\sum_{i=1}^n E\left[X_i^2\right]}\right)^q\right\}.$$

Proof of Proposition D.3. From Lemma D.3, let us observe

- if $2 < q \leq 4$, choosing $\gamma = 1$ provides
$$B^*(q,\gamma) \leq \left(2\sqrt{2e}\sqrt{q}\right)^q.$$
- if $4 < q$, $\gamma = q^{(q-1)/2}$ leads to
$$B^*(q,\gamma) \leq q^{-q/2}\left(\sqrt{4eq\left(q^{1/2} + q\right)}\right)^q \leq q^{-q/2}\left(\sqrt{8eq}\right)^q = \left(2\sqrt{2e}\sqrt{q}\right)^q.$$

Plugging the previous upper bounds in Rosenthal's inequality (Proposition D.2), it results for every $q > 2$

$$E\left[\left|\sum_{i=1}^n X_i\right|^q\right] \leq \left(2\sqrt{2e}\sqrt{q}\right)^q \max\left\{(\sqrt{q})^q \sum_{i=1}^n E\left[\left|X_i\right|^q\right], \left(\sqrt{\sum_{i=1}^n E\left[X_i^2\right]}\right)^q\right\}.$$

□

Lemma D.3. *With the same notation as Proposition D.2 and for every $\gamma > 0$, it comes*

- *for every $2 < q \leq 4$,*
$$B^*(q,\gamma) \leq 1 + \frac{\left(\sqrt{2e}\sqrt{q}\right)^q}{\gamma},$$
- *for every $4 < q$,*
$$B^*(q,\gamma) \leq \gamma^{-q/(q-1)}\left(\sqrt{4eq\left(\gamma^{1/(q-1)} + q\right)}\right)^q.$$

Proof of Lemma D.3. If $2 < q \leq 4$,

$$B^*(q,\gamma) = 1 + \frac{E\left[\left|N\right|^q\right]}{\gamma} \leq 1 + \frac{\sqrt{2}e\sqrt{q}\left(\frac{q}{e}\right)^{\frac{q}{2}}}{\gamma} \leq 1 + \frac{\sqrt{2e}^q\sqrt{e}^q\left(\frac{q}{e}\right)^{\frac{q}{2}}}{\gamma} = 1 + \frac{\left(\sqrt{2e}\sqrt{q}\right)^q}{\gamma} \ ,$$

by use of Lemma D.9 and $\sqrt{q}^{1/q} \leq \sqrt{e}$ for every $q > 2$.

If $q > 4$,

$$\begin{aligned} B^*(q,\gamma) &= \gamma^{-q/(q-1)} E\left[\left|Z - Z'\right|^q\right] \\ &\leq \gamma^{-q/(q-1)} 2^{q/2+1} e\sqrt{q}\left[\frac{q}{e}\left(\gamma^{1/(q-1)}+q\right)\right]^{q/2} \\ &\leq \gamma^{-q/(q-1)} 2^{q/2}\sqrt{2e}^q\sqrt{e}^q\left[\frac{q}{e}\left(\gamma^{1/(q-1)}+q\right)\right]^{q/2} \\ &\leq \gamma^{-q/(q-1)}\left[4eq\left(\gamma^{1/(q-1)}+q\right)\right]^{q/2} = \gamma^{-q/(q-1)}\left(\sqrt{4eq\left(\gamma^{1/(q-1)}+q\right)}\right)^q , \end{aligned}$$

applying Lemma D.11 with $\lambda = 1/2\gamma^{1/(q-1)}$.

□

D.2. Technical lemmas

D.2.1. Basic computations for resampling applied to the kNN algorithm

Lemma D.4. *For every $1 \leq i \leq n$ and $1 \leq p \leq n$, one has*

$$\mathbb{P}_e\left(i \in \bar{e}\right) = \frac{p}{n}, \tag{D.5}$$

$$\sum_{j=1}^{n} \mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] = \frac{kp}{n}\ , \tag{D.6}$$

$$\sum_{k<\sigma_i(j)\leq k+p} \mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] = \frac{kp}{n}\frac{p-1}{n-1}\ . \tag{D.7}$$

Proof of Lemma D.4. The first equality is straightforward. The second one results from simple calculations as follows.

$$\begin{aligned} \sum_{j=1}^{n} \mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] &= \sum_{j=1}^{n}\binom{n}{p}^{-1}\sum_e \mathbb{1}_{i\in\bar{e}}\mathbb{1}_{j\in V_k^e(X_i)} = \binom{n}{p}^{-1}\sum_e \mathbb{1}_{i\in\bar{e}}\left(\sum_{j=1}^{n}\mathbb{1}_{j\in V_k^e(X_i)}\right) \\ &= \left(\binom{n}{p}^{-1}\sum_e \mathbb{1}_{i\in\bar{e}}\right)k = \frac{p}{n}\,k\ . \end{aligned}$$

For the last equality, let us notice every $j \in V_i$ satisfies

$$\mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] = \mathbb{P}_e\left[\, j \in V_k^e(X_i) \mid i \in \bar{e}\,\right]\mathbb{P}_e\left[\, i \in \bar{e}\,\right] = \frac{n-1}{n-p}\frac{p}{n}\ ,$$

hence

$$\begin{aligned} \sum_{k<\sigma_i(j)\leq k+p} \mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] &= \sum_{j=1}^{n}\mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] - \sum_{\sigma_i(j)\leq k}\mathbb{P}_e\left[\, i \in \bar{e},\ j \in V_k^e(X_i)\,\right] \\ &= k\frac{p}{n} - k\frac{n-1}{n-p}\frac{p}{n} = k\frac{p}{n}\frac{p-1}{n-1}\ . \end{aligned}$$

□

D.2.2. Stone's lemma

Lemma D.5 (Devroye et al. (1996), Corollary 11.1, p. 171)**.** *Given n points $(x_1, ..., x_n)$ in $\mathbb{R}^d$, any of these points belongs to the k nearest neighbors of at most $k\gamma_d$ of the other points, where γ_d increases on d.*

D.2.3. Stability of the kNN classifier when removing p observations

Lemma D.6 (Devroye and Wagner (1979b), Eq. (14))**.** *For every $1 \leq k \leq n$, let $\mathcal{A}_k$ denote k-NN classification algorithm defined by Eq. (2.1), and let $Z_1, \ldots, Z_n$ denote n i.i.d. random variables such that for every $1 \leq i \leq n$, $Z_i = (X_i, Y_i) \sim P$. Then for every $1 \leq p \leq n-k$,*

$$\mathbb{P}\left[\,\mathcal{A}_k(Z_{1,n}; X) \neq \mathcal{A}_k(Z_{1,n-p}; X)\,\right] \leq \frac{4}{\sqrt{2\pi}}\frac{p\sqrt{k}}{n}\ ,$$

where $Z_{1,i} = (Z_1, \ldots, Z_i)$ for every $1 \leq i \leq n$, and $(X, Y) \sim P$ is independent of $Z_{1,n}$.

D.2.4. Exponential concentration inequality for the L1O estimator

Lemma D.7 (Devroye et al. (1996), Theorem 24.4)**.** *For every $1 \leq k \leq n$, let $\mathcal{A}_k$ denote k-NN classification algorithm defined by Eq. (2.1). Let also $\widehat{R}_1(\cdot)$ denote the L1O estimator defined by Eq. (2.2) with $p = 1$. Then for every $\varepsilon > 0$,*

$$\mathbb{P}\left(\left|\widehat{R}_1(\mathcal{A}_k, Z_{1,n}) - \mathbb{E}\left[\,\widehat{R}_1(\mathcal{A}_k, Z_{1,n})\,\right]\right| > \varepsilon\right) \leq 2\exp\left\{-n\frac{\varepsilon^2}{\gamma_d^2 k^2}\right\}.$$

D.2.5. Moment upper bounds for the L1O estimator

Lemma D.8. *For every $1 \leq k \leq n$, let $\mathcal{A}_k$ denote k-NN classification algorithm defined by Eq. (2.1). Let also $\widehat{R}_1(\cdot)$ denote the L1O estimator defined by Eq. (2.2) with $p = 1$. Then for every $q \geq 1$,*

$$\mathbb{E}\left[\left|\widehat{R}_1\left(\mathcal{A}_k, Z_{1,n}\right) - \mathbb{E}\left[\,\widehat{R}_1\left(\mathcal{A}_k, Z_{1,n}\right)\,\right]\right|^{2q}\right] \leq q!\left(2\frac{(k\gamma_d)^2}{n}\right)^q. \tag{D.8}$$

The proof is straightforward from the combination of Lemmas D.1 and D.7.

D.2.6. Upper bound on the optimal constant in the Rosenthal's inequality

Lemma D.9. *Let N denote a real-valued standard Gaussian random variable. Then for every $q > 2$, one has*

$$\mathbb{E}\left[\,|N|^q\,\right] \leq \sqrt{2}e\sqrt{q}\left(\frac{q}{e}\right)^{\frac{q}{2}}.$$

Proof of Lemma D.9. If q is even ($q = 2k > 2$), then

$$\begin{aligned} \mathbb{E}\left[\,|N|^q\,\right] &= 2\int_0^{+\infty} x^q \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}dx = \sqrt{\frac{2}{\pi}}(q-1)\int_0^{+\infty} x^{q-2}e^{-\frac{x^2}{2}}dx \\ &= \sqrt{\frac{2}{\pi}}\frac{(q-1)!}{2^{k-1}(k-1)!} = \sqrt{\frac{2}{\pi}}\frac{q!}{2^{q/2}(q/2)!}\ . \end{aligned}$$

Then using for any positive integer a

$$\sqrt{2\pi a}\left(\frac{a}{e}\right)^a < a! < \sqrt{2e\pi a}\left(\frac{a}{e}\right)^a,$$

it results

$$\frac{q!}{2^{q/2}(q/2)!} < \sqrt{2e}\, e^{-q/2} q^{q/2},$$

which implies

$$\mathbb{E}\left[\,|N|^q\,\right] \leq 2\sqrt{\frac{e}{\pi}}\left(\frac{q}{e}\right)^{q/2} < \sqrt{2}e\sqrt{q}\left(\frac{q}{e}\right)^{\frac{q}{2}} \ .$$

If q is odd ($q = 2k+1 > 2$), then

$$\mathbb{E}\left[\,|N|^q\,\right] = \sqrt{\frac{2}{\pi}}\int_0^{+\infty} x^q e^{-\frac{x^2}{2}} dx = \sqrt{\frac{2}{\pi}}\int_0^{+\infty} \sqrt{2t}^q e^{-t}\frac{dt}{\sqrt{2t}},$$

by setting $x = \sqrt{2t}$. In particular, this implies

$$\mathbb{E}\left[\,|N|^q\,\right] \leq \sqrt{\frac{2}{\pi}}\int_0^{+\infty} (2t)^k\, e^{-t} dt = \sqrt{\frac{2}{\pi}} 2^k k! = \sqrt{\frac{2}{\pi}} 2^{\frac{q-1}{2}}\left(\frac{q-1}{2}\right)! < \sqrt{2}e\sqrt{q}\left(\frac{q}{e}\right)^{\frac{q}{2}}.$$

□

Lemma D.10. *Let S denote a binomial random variable such that $S \sim \mathcal{B}(k, 1/2)$ ($k \in \mathbb{N}^*$). Then for every $q > 3$, it comes*

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] \leq 4\sqrt{e}\sqrt{q}\sqrt{\frac{qk}{2e}}^q \ .$$

Proof of Lemma D.10. Since $S - \mathbb{E}(S)$ is symmetric, it comes

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] = 2\int_0^{+\infty} \mathbb{P}\left[\,S < \mathbb{E}\left[\,S\,\right] - t^{1/q}\,\right] dt = 2q\int_0^{+\infty} \mathbb{P}\left[\,S < \mathbb{E}\left[\,S\,\right] - u\,\right] u^{q-1} du.$$

Using Chernoff's inequality and setting $u = \sqrt{k/2}v$, it results

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] \leq 2q\int_0^{+\infty} u^{q-1} e^{-\frac{u^2}{k}} du = 2q\sqrt{\frac{k}{2}}^q \int_0^{+\infty} v^{q-1} e^{-\frac{v^2}{2}} dv.$$

If q is even, then $q-1 > 2$ is odd and the same calculations as in the proof of Lemma D.9 apply, which leads to

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] \leq 2\sqrt{\frac{k}{2}}^q 2^{q/2}\left(\frac{q}{2}\right)! \leq 2\sqrt{\frac{k}{2}}^q 2^{q/2}\sqrt{\pi e q}\left(\frac{q}{2e}\right)^{q/2} = 2\sqrt{\pi e}\sqrt{q}\sqrt{\frac{qk}{2e}}^q < 4\sqrt{e}\sqrt{q}\sqrt{\frac{qk}{2e}}^q \ .$$

If q is odd, then $q - 1 > 2$ is even and another use of the calculations in the proof of Lemma D.9 provides

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] \leq 2q\sqrt{\frac{k}{2}}^q \frac{(q-1)!}{2^{(q-1)/2}\frac{q-1}{2}!} = 2\sqrt{\frac{k}{2}}^q \frac{q!}{2^{(q-1)/2}\frac{q-1}{2}!}.$$

Let us notice

$$\begin{aligned}\frac{q!}{2^{(q-1)/2}\frac{q-1}{2}!} &\leq \frac{\sqrt{2\pi e q}\left(\frac{q}{e}\right)^q}{2^{(q-1)/2}\sqrt{\pi(q-1)}\left(\frac{q-1}{2e}\right)^{(q-1)/2}} = \sqrt{2e}\sqrt{\frac{q}{q-1}}\frac{\left(\frac{q}{e}\right)^q}{\left(\frac{q-1}{e}\right)^{(q-1)/2}} \\ &= \sqrt{2e}\sqrt{\frac{q}{q-1}}\left(\frac{q}{e}\right)^{(q+1)/2}\left(\frac{q}{q-1}\right)^{(q-1)/2}\end{aligned}$$

and also that

$$\sqrt{\frac{q}{q-1}}\left(\frac{q}{q-1}\right)^{(q-1)/2} \leq \sqrt{2e}.$$

This implies

$$\frac{q!}{2^{(q-1)/2}\frac{q-1}{2}!} \leq 2e\left(\frac{q}{e}\right)^{(q+1)/2} = 2\sqrt{e}\sqrt{q}\left(\frac{q}{e}\right)^{q/2},$$

hence

$$\mathbb{E}\left[\,|S - \mathbb{E}\left[\,S\,\right]|^q\,\right] \leq 2\sqrt{\frac{k}{2}}^q 2\sqrt{e}\sqrt{q}\left(\frac{q}{e}\right)^{q/2} = 4\sqrt{e}\sqrt{q}\sqrt{\frac{qk}{2e}}^q \ .$$

□

Lemma D.11. *Let X, Y be two i.i.d. random variables with Poisson distribution $\mathcal{P}(\lambda)$ ($\lambda > 0$). Then for every $q > 3$, it comes*

$$\mathbb{E}\left[\,|X - Y|^q\,\right] \leq 2^{q/2+1} e\sqrt{q}\left[\,\frac{q}{e}\left(2\lambda + q\right)\,\right]^{q/2}.$$

Proof of Lemma D.11. Let us first remark that

$$\mathbb{E}\left[\,|X - Y|^q\,\right] = \mathbb{E}_N\left[\,\mathbb{E}\left[\,|X - Y|^q \mid N\,\right]\,\right] = 2^q \mathbb{E}_N\left[\,\mathbb{E}\left[\,|X - N/2|^q \mid N\,\right]\,\right],$$

where $N = X + Y$. Furthermore, the conditional distribution of X given $N = X + Y$ is a binomial distribution $\mathcal{B}(N, 1/2)$. Then Lemma D.10 provides that

$$\mathbb{E}\left[\,|X - N/2|^q \mid N\,\right] \leq 4\sqrt{e}\sqrt{q}\sqrt{\frac{qN}{2e}}^q \qquad a.s.,$$

which entails that

$$\mathbb{E}\left[\,|X - Y|^q\,\right] \leq 2^q \mathbb{E}_N\left[\,4\sqrt{e}\sqrt{q}\sqrt{\frac{qN}{2e}}^q\,\right] = 2^{q/2+2}\sqrt{e}\sqrt{q}\sqrt{\frac{q}{e}}^q \mathbb{E}_N\left[\,N^{q/2}\,\right].$$

It only remains to upper bound the last expectation where N is a Poisson random variable $\mathcal{P}(2\lambda)$ (since X, Y are *i.i.d.*):

$$\mathbb{E}_N\left[N^{q/2}\right] \leq \sqrt{\mathbb{E}_N\left[N^q\right]}$$

by Jensen's inequality. Further introducing Touchard polynomials and using a classical upper bound, it comes

$$\begin{aligned}\mathbb{E}_N\left[N^{q/2}\right] &\leq \sqrt{\sum_{i=1}^{q}(2\lambda)^i\frac{1}{2}\binom{q}{i}i^{q-i}} \leq \sqrt{\sum_{i=0}^{q}(2\lambda)^i\frac{1}{2}\binom{q}{i}q^{q-i}} \\ &= \sqrt{\frac{1}{2}\sum_{i=0}^{q}\binom{q}{i}(2\lambda)^i q^{q-i}} = \sqrt{\frac{1}{2}(2\lambda+q)^q} = 2^{\frac{-1}{2}}(2\lambda+q)^{q/2}.\end{aligned}$$

Finally, one concludes

$$\mathbb{E}\left[|X-Y|^q\right] \leq 2^{q/2+2}\sqrt{e}\sqrt{q}\sqrt{\frac{q}{e}}^q 2^{\frac{-1}{2}}(2\lambda+q)^{q/2} < 2^{q/2+1}e\sqrt{q}\left[\frac{q}{e}(2\lambda+q)\right]^{q/2}.$$

□

References

A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 459–468. IEEE, 2006.

S. Arlot. *Resampling and Model Selection*. PhD thesis, University Paris-Sud 11, December 2007. URL `http://tel.archives-ouvertes.fr/tel-00198803/en/`. oai:tel.archives-ouvertes.fr:tel-00198803_v1.

S. Arlot and F. Bach. Data-driven calibration of linear estimators with minimal penalties. *Advances in Neural Information Processing Systems (NIPS)*, 2:46–54, 2009.

S. Arlot and A. Celisse. A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4:40–79, 2010.

S. Arlot and M. Lerasle. Why v= 5 is enough in v-fold cross-validation. *arXiv preprint arXiv:1210.5830*, 2012.

T. B. Berrett, R. J. Samworth, and M. Yuan. Efficient multivariate entropy estimation via k-nearest neighbour distances. *arXiv preprint arXiv:1606.00304*, 2016.

G. Biau and L. Devroye. *Lectures on the nearest neighbor method*. Springer, 2016.

G. Biau, F. Cérou, and A. Guyader. On the rate of convergence of the bagged nearest neighbor estimate. *The Journal of Machine Learning Research*, 11:687–712, 2010a.

G. Biau, F. Cérou, and A. Guyader. Rates of convergence of the functional-nearest neighbor estimate. *Information Theory, IEEE Transactions on*, 56(4):2034–2040, 2010b.

S. Boucheron, O. Bousquet, G. Lugosi, and P. Massart. Moment inequalities for functions of independent random variables. *Ann. Probab.*, 33(2):514–560, 2005. ISSN 0091-1798.

S. Boucheron, G. Lugosi, and P. Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. Oxford University Press, 2013.

L. Breiman and Ph. Spector. Submodel selection and evaluation in regression. the x-random case. *International Statistical Review*, 60(3):291–319, 1992.

P. Burman. Comparative study of Ordinary Cross-Validation, v-Fold Cross-Validation and the repeated Learning-Testing Methods. *Biometrika*, 76(3):503–514, 1989.

T. I. Cannings, T. B. Berrett, and R. J. Samworth. Local nearest neighbour classification with applications to semi-supervised learning. *arXiv preprint arXiv:1704.00642*, 2017.

A. Celisse. *Model selection via cross-validation in density estimation, regression and change-points detection. (In English)*. PhD thesis, University Paris-Sud 11. `http://tel.archives-ouvertes.fr/tel-00346320/en/`., December 2008. URL `http://tel.archives-ouvertes.fr/tel-00346320/en/`.

A. Celisse. Optimal cross-validation in density estimation with the l^2-loss. *The Annals of Statistics*, 42(5):1879–1910, 2014.

A. Celisse and T. Mary-Huard. Exact cross-validation for knn: applications to passive and active learning in classification. *JSFdS*, 152(3), 2011.

A. Celisse and S. Robin. Nonparametric density estimation by exact leave-p-out cross-validation. *Computational Statistics and Data Analysis*, 52(5):2350–2368, 2008.

K. Chaudhuri and S. Dasgupta. Rates of convergence for nearest neighbor classification. In *Advances in Neural Information Processing Systems*, pages 3437–3445, 2014.

T. M. Cover. Rates of convergence for nearest neighbor procedures. In *Proceedings of the Hawaii International Conference on Systems Sciences*, pages 413–415, 1968.

T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, 1967.

L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25:601–604, 1979a.

L. Devroye, L. Györfi, and G. Lugosi. *A Probilistic Theory of Pattern Recognition*. Springer Verlag, 1996.

L. P. Devroye and T. J. Wagner. The strong uniform consistency of nearest neighbor density estimates. *Ann. Statist.*, 5(3):536–540, 1977. ISSN 0090-5364.

L. P. Devroye and T. J. Wagner. Distribution-free inequalities for the deleted and holdout error estimates. *Information Theory, IEEE Transactions on*, 25(2):202–207, 1979b.

L.P. Devroye and T.J. Wagner. Distribution-free consistency results in nonparametric discrimination and regression function estimation. *Ann. Statist.*, 8(2):231–239, 1980.

E. Fix and J. Hodges. *Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques*, chapter Discriminatory analysis- nonparametric discrimination: Consistency principles. IEEE Computer Society Press, Los Alamitos, CA, 1951. Reprint of original work from 1952.

M. Fuchs, R. Hornung, R. De Bin, and A.-L. Boulesteix. A u-statistic estimator for the variance of resampling-based error estimators. Technical report, arXiv, 2013.

S. Geisser. The predictive sample reuse method with applications. *J. Amer. Statist. Assoc.*, 70:320–328, 1975.

L. Györfi. The rate of convergence of k_n-nn regression estimates and classification rules. *IEEE Trans. Commun*, 27(3):362–364, 1981.

P. Hall, B. U. Park, and R. J. Samworth. Choice of neighbor order in nearest-neighbor classification. *The Annals of Statistics*, pages 2135–2152, 2008.

T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer Series in Statistics. Springer-Verlag, New York, 2001. ISBN 0-387-95284-5. Data mining, inference, and prediction.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journ. of the American Statistical Association*, 58(301):13–30, 1963.

R. Ibragimov and S. Sharakhmetov. On extremal problems and best constants in moment inequalities. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 42–56, 2002.

P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 604–613. ACM, 1998.

M. Kearns and D. Ron. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. *Neural Computation*, 11:1427–1453, 1999.

V. S. Koroljuk and Y. V. Borovskich. *Theory of U-statistics*. Springer, 1994.

S. R. Kulkarni and S. E. Posner. Rates of convergence of nearest neighbor estimation under arbitrary sampling. *Information Theory, IEEE Transactions on*, 41(4):1028–1039, 1995.

L. Li, D. M. Umbach, P. Terry, and J. A. Taylor. Application of the ga/knn method to seldi proteomics data. *Bioinformatics*, 20(10):1638–1640, 2004.

P. Massart and E. Nédélec. Risk bounds for statistical learning. *The Annals of Statistics*, pages 2326–2366, 2006.

D. Psaltis, R. R. Snapp, and S. S. Venkatesh. On the finite sample performance of the nearest neighbor classifier. *Information Theory, IEEE Transactions on*, 40(3):820–837, 1994.

W. H. Rogers and T. J. Wagner. A finite sample distribution-free performance bound for local discrimination rules. *Annals of Statistics*, 6(3):506–514, 1978.

E. D. Scheirer and M. Slaney. Multi-feature speech/music discrimination system, May 27 2003. US Patent 6,570,991.

R. J. Serfling. *Approximation Theorems of Mathematical Statistics*. John Wiley & Sons Inc., 1980.

J. Shao. Linear model selection by cross-validation. *J. Amer. Statist. Assoc.*, 88(422): 486–494, 1993. ISSN 0162-1459.

J. Shao. An asymptotic theory for linear model selection. *Statistica Sinica*, 7:221–264, 1997.

P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998.

R. R Snapp and S. S. Venkatesh. Asymptotic expansions of the k nearest neighbor risk. *The Annals of Statistics*, 26(3):850–878, 1998.

B. M. Steele. Exact bootstrap k-nearest neighbor learners. *Machine Learning*, 74(3):235–255, 2009.

C. J. Stone. Consistent nonparametric regression. *The annals of statistics*, pages 595–620, 1977.

C. J. Stone. Optimal global rates of convergence for nonparametric regression. *Ann. Statist.*, 10(4):1040–1053, 1982. ISSN 0090-5364.

M. Stone. Cross-validatory choice and assessment of statistical predictions. *J. Roy. Statist. Soc. Ser. B*, 36:111–147, 1974. ISSN 0035-9246. With discussion by G. A. Barnard, A. C. Atkinson, L. K. Chan, A. P. Dawid, F. Downton, J. Dickey, A. G. Baker, O. Barndorff-Nielsen, D. R. Cox, S. Giesser, D. Hinkley, R. R. Hocking, and A. S. Young, and with a reply by the authors.

Y. Yang. Minimax nonparametric classification. i. rates of convergence. *IEEE Transactions on Information Theory*, 45(7):2271–2284, 1999.

Y. Yang. Comparing learning methods for classification. *Statistica Sinica*, 16(2):635–657, 2006. ISSN 1017-0405.

Y. Yang. Consistency of cross-validation for comparing regression procedures. *The Annals of Statistics*, 35(6):2450–2473, 2007.

P. Zhang. Model selection via multifold cross validation. *Ann. Statist.*, 21(1):299–313, 1993. ISSN 0090-5364.

Journal of Machine Learning Research 19 (2018) 1-31 Submitted 3/17; Revised 7/18; Published 10/18

Maximum Selection and Sorting with Adversarial Comparators

Jayadev Acharya ACHARYA@CORNELL.EDU
School of ECE
Cornell University
Ithaca, NY 14853, USA

Moein Falahatgar MOEIN@UCSD.EDU
ECE Department
UC San Diego
La Jolla, CA 92093, USA

Ashkan Jafarpour ASHKAN.JAFARPOUR@GMAIL.COM
Google
Sunnyvale, CA 94089, USA

Alon Orlitsky ALON@UCSD.EDU
ECE and CSE Departments
UC San Diego
La Jolla, CA 92093, USA

Ananda Theertha Suresh THEERTHA@GOOGLE.COM
Google Research
New York, NY 10011, USA

Editor: Gabor Lugosi

Abstract

We study maximum selection and sorting of n numbers using imperfect pairwise comparators. The imperfect comparator returns the larger of the two inputs if the inputs are more than a given threshold apart and an adversarially-chosen input otherwise. We consider two adversarial models: a non-adaptive adversary that decides on the outcomes in advance and an adaptive adversary that decides on the outcome of each comparison depending on the previous comparisons and outcomes.

Against the non-adaptive adversary, we derive a maximum-selection algorithm that uses at most $2n$ comparisons in expectation and a sorting algorithm that uses at most $2n \ln n$ comparisons in expectation. In the presence of the adaptive adversary, the proposed maximum-selection algorithm uses $\Theta(n \log(1/\epsilon))$ comparisons to output a correct answer with probability at least $1 - \epsilon$, resolving an open problem in Ajtai et al. (2015).

Our study is motivated by a density-estimation problem. Given samples from an unknown distribution, we would like to find a distribution among a known class of n candidate distributions that is close to the underlying distribution in ℓ_1 distance. Scheffe's algorithm, for example, in Devroye and Lugosi (2001) outputs a distribution at an ℓ_1 distance at most 9 times the minimum and runs in time $\Theta(n^2 \log n)$. Using our algorithm, the runtime reduces to $\Theta(n \log n)$.

Keywords: noisy sorting, adversarial comparators, density estimation, Scheffe estimator

1. Introduction

Maximum selection and sorting are fundamental operations with widespread applications in computing, investment, marketing (Aggarwal et al., 2009), decision making (Thurstone, 1927; David, 1963), and sports. These operations are often accomplished via pairwise comparisons between elements, and the goal is to minimize the number of comparisons.

For example, one may find the largest of n elements by first comparing two elements and then successively comparing the larger one to a new element. This simple algorithm takes $n-1$ comparisons, and it is easy to see that $n-1$ comparisons are necessary. Similarly, *merge sort* sorts n elements using less than $n \log n$ comparisons, close to the information theoretic lower bound of $\log n! = n \log n - o(n)$.

However, in many applications, the pairwise comparisons may be imprecise. For example, in comparing two random numbers, such as stock performances, or team strengths, the output of the comparison may vary due to chance. Consequently, a number of researchers have considered maximum selection and sorting with imperfect, or noisy, comparators. The comparators in these models mostly function correctly but occasionally may produce an inaccurate comparison result, where the form of inaccuracy is dictated by the application.

Based on the form of inaccuracy, models can be divided into two categories: probabilistic and adversarial. Probabilistic models can be parametric or non-parametric. One of the simplest parametric probabilistic models was considered in Feige et al. (1994), where the output of each comparator could be wrong with some known probability p. Algorithms applying this model for maximum selection were proposed in Adler et al. (1994) and for ranking in Karp and Kleinberg (2007); Ben-Or and Hassidim (2008); Braverman and Mossel (2008); Braverman et al. (2016).

Another parametric family of probabilistic models, the Bradley-Terry-Luce model (Bradley and Terry, 1952) assumes that if two values x and y are compared, then x is selected as the larger with probability $x/(x+y)$. Observe that the comparison is correct with probability $\max\{x,y\}/(x+y) \geq 1/2$. Algorithms for ranking and estimating values under this and another related model, the Plackett-Luce (Plackett, 1975; Luce, 2005), are proposed, for example, in Negahban et al. (2012); Szörényi et al. (2015). The Mallows model is yet another example of a parametric probabilistic model and is studied in Busa-Fekete et al. (2014).

Non-parametric probabilistic models assume some natural constraints on comparison probabilities, such as Strong Stochastic Transitivity or Stochastic Triangle Inequality. Algorithms for maximum selection and sorting under these models are studied in Falahatgar et al. (2017b,a, 2018); Yue and Joachims (2011) and algorithms for comparison-probability matrix estimation are considered in Shah et al. (2016). This model is also considered for the top-k sorting problem in Chen et al. (2017b,a).

We consider a model where, unlike the probabilistic models, the comparison outcome can be adversarial. If the numbers compared are more than a threshold Δ apart, the comparison is correct, while if they differ by at most Δ, the comparison outcome is arbitrary, and possibly even adversarial.

This model can be partially motivated by physical observations. Measurements are regularly quantized and often adulterated with some measurement noise. Quantities with the same quantized value may, therefore, be incorrectly compared. In psychophysics, the

Weber-Fechner law (Ekman, 1959) stipulates that humans can distinguish between two physical stimuli only when their difference exceeds some threshold (known as *just noticeable difference*). Additionally, in sports, a judge or a home-team advantage may, even adversarially, sway the outcome of a game between two teams of similar strength but not between teams of significantly different strengths. Our main motivation for the model derives from the important problem of density estimation and distribution learning.

1.1. Density estimation via pairwise comparisons

In a typical PAC-learning setup (Valiant, 1984; Kearns et al., 1994), we are given samples from an unknown distribution p_0 in a known distribution class $\mathcal{P}$ and would like to find, with high probability, a distribution $\hat{p} \in \mathcal{P}$ such that $\|\hat{p} - p_0\|_1 < \delta$.

One standard approach proceeds in two steps (Devroye and Lugosi, 2001):

1. Offline, construct a δ-*cover* of $\mathcal{P}$, a finite collection $\mathcal{P}_\delta \subseteq \mathcal{P}$ of distributions such that for any distribution $p \in \mathcal{P}$, there is a distribution $q \in \mathcal{P}_\delta$ such that $\|p - q\|_1 < \delta$.

2. Using the samples from p_0, find a distribution in $\mathcal{P}_\delta$ whose ℓ_1 distance to p_0 is close to the ℓ_1 distance of the distribution in $\mathcal{P}_\delta$ that is closest to p_0.

These two steps output a distribution whose ℓ_1 distance from p_0 is *close* to δ. Surprisingly, for several common distribution classes, such as Gaussian mixtures, the number of samples required by this generic approach matches the information theoretically optimal sample complexity, up to logarithmic factors (Daskalakis and Kamath, 2014; Suresh et al., 2014; Diakonikolas et al., 2016).

The Scheffe Algorithm (Scheffe, 1947; Devroye and Lugosi, 2001) is a popular method for implementing the second step, namely to find a distribution in $\mathcal{P}_\delta$ with a small ℓ_1 distance from p_0. It takes every pair of distributions in $\mathcal{P}_\delta$ and uses the samples from p_0 to decide which of the two distributions is closer to p_0. It then declares the distribution that "wins" the most pairwise closeness comparisons to be the nearly-closest to p_0. As shown in Devroye and Lugosi (2001), the Scheffe algorithm yields, with high probability, a distribution that is at most nine times further from p_0 than the distribution in $\mathcal{P}_\delta$ with the lowest ℓ_1 distance from p_0, plus a diminishing additive term; hence, a distribution that is roughly 9δ away from p_0 is found. Since this algorithm compares every pair of distributions in $\mathcal{P}_\delta$, it uses quadratic in $|\mathcal{P}_\delta|$ comparisons. In Section 6, we use maximum-selection results to derive an algorithm with the same approximation guarantee but linear in $|\mathcal{P}_\delta|$ comparisons.

1.2. Organization

This paper is organized as follows: in Section 2, we define the problem and introduce the notations; in Section 3, we summarize the results; in Section 4, we derive simple bounds and describe the performance of simple algorithms; and, in Section 5, we present our main maximum-selection algorithms. The relation between density estimation problem and our comparison model is discussed in Section 6, and, in Section 7, we discuss sorting with adversarial comparators.

2. Notations and preliminaries

Practical applications call for sorting or selecting the maximum of not just numbers, but, rather, of items with associated values—for example, finding the person with the highest salary, the product with the lowest price, or a sports team with the most *capability* of winning. Associate with each item i a real value x_i and let $\mathcal{X} \stackrel{\text{def}}{=} \{x_1, \ldots, x_n\}$ be the multiset of values. In maximum selection, we use noisy pairwise comparisons to find an index i such that x_i is close to the largest element $x^* \stackrel{\text{def}}{=} \max\{x_1, \ldots, x_n\}$.

Formally, a faulty comparator $\mathcal{C}$ takes two distinct indices i and j and, if $|x_i - x_j| > \Delta$, outputs the index associated with the higher value, while if $|x_i - x_j| \le \Delta$, outputs either i or j, possibly adversarially. Without loss of generality, we assume that $\Delta = 1$. Then,

$$\mathcal{C}(i,j) = \begin{cases} \arg\max\{x_i, x_j\} & \text{if} \quad |x_i - x_j| > 1, \\ i \text{ or } j \text{ (adversarially)} & \text{if} \quad |x_i - x_j| \le 1. \end{cases}$$

It is easier to think just of the numbers, rather than the indices. Therefore, informally we will simply view the comparators as taking two real inputs x_i and x_j, and outputting

$$\mathcal{C}(x_i,x_j) = \begin{cases} \max\{x_i, x_j\} & \text{if} \quad |x_i - x_j| > 1, \\ x_i \text{ or } x_j \text{ (adversarially)} & \text{if} \quad |x_i - x_j| \le 1. \end{cases} \tag{1}$$

We consider two types of adversarial comparators: *non-adaptive* and *adaptive*.

- A *non-adaptive adversarial comparator* has complete knowledge of $\mathcal{X}$ and the algorithm but must fix its outputs for every pair of inputs before the algorithm starts
- An *adaptive adversarial comparator* not only has access to the algorithm and the inputs but is also allowed to adaptively decide the outcomes of the queries taking into account all the previous comparisons made by the algorithm

A non-adaptive comparator can be naturally represented by a directed graph with n nodes representing the n indices. There is an edge from node i to node j if the comparator declares x_i to be larger than x_j, namely, $\mathcal{C}(x_i, x_j) = x_i$. Figure 1 is an example of such a comparator, where, for simplicity, we show only the values 0, 1, 1, 2, and not the indices. Note that, by definition, $\mathcal{C}(2,0) = 2$, but for all the other pairs, the outputs can be decided by the comparator. In this example, the comparator declares the node with value 2 as the "winner" against the right node with value 1 but as the "loser" against the left node, also with value 1. Among the two nodes with value 1, it arbitrarily declares the left one as the winner. An adaptive adversary reveals the edges one-by-one as the algorithm proceeds.

We refer to each comparison as a *query*. The number of queries an algorithm $\mathcal{A}$ makes for $\mathcal{X} = \{x_1, \ldots, x_n\}$ is its *query complexity*, denoted by $Q_n^{\mathcal{A}}$.[1] Our algorithms are randomized, and $Q_n^{\mathcal{A}}$ is a random variable. The *expected query complexity* of $\mathcal{A}$ for the input $\mathcal{X}$ is

$$q_n^{\mathcal{A}} \stackrel{\text{def}}{=} \mathbb{E}[Q_n^{\mathcal{A}}],$$

where the expectation is over the randomness of the algorithm. Note that the expected query complexity is defined for all runs of an algorithm, and it is independent of the success probability.

1. This is a slight abuse of notation suppressing $\mathcal{X}$.

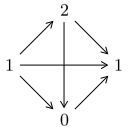


Figure 1: Comparator for four inputs with values $\{0, 1, 1, 2\}$

Let $\mathcal{C}_{\text{non}}(\mathcal{X})$, or simply $\mathcal{C}_{\text{non}}$, be the set of all non-adaptive adversarial comparators, and let $\mathcal{C}_{\text{adpt}}$ be the set of all adaptive adversarial comparators. The *maximum expected query complexity* of $\mathcal{A}$ against non-adaptive adversarial comparators is

$$q_n^{\mathcal{A},\text{non}} \stackrel{\text{def}}{=} \max_{\mathcal{C}\in\mathcal{C}_{\text{non}}} \max_{\mathcal{X}} q_n^{\mathcal{A}}. \tag{2}$$

Similarly, the maximum expected query complexity of $\mathcal{A}$ against adaptive adversarial comparators is

$$q_n^{\mathcal{A},\text{adpt}} \stackrel{\text{def}}{=} \max_{\mathcal{C}\in\mathcal{C}_{\text{adpt}}} \max_{\mathcal{X}} q_n^{\mathcal{A}}.$$

We evaluate an algorithm by how close its output is to x^* (the maximum of $\mathcal{X}$).

Definition 1 *A number x is a* t-approximation *of x^* if $x \geq x^* - t$.*

The *t-approximation error* of an algorithm $\mathcal{A}$ over n inputs is

$$\mathcal{E}_n^{\mathcal{A}}(t) \stackrel{\text{def}}{=} \Pr\left(Y_{\mathcal{A}}(\mathcal{X}) < x^* - t\right),$$

the probability that $\mathcal{A}$'s output $Y_{\mathcal{A}}(\mathcal{X})$ is *not* a t-approximation of x^*. For an algorithm $\mathcal{A}$, the maximum t-approximation error for the worst non-adaptive adversary is

$$\mathcal{E}_n^{\mathcal{A},\text{non}}(t) \stackrel{\text{def}}{=} \max_{\mathcal{C}\in\mathcal{C}_{\text{non}}} \max_{\mathcal{X}} \mathcal{E}_n^{\mathcal{A}}(t),$$

and, similarly, for the adaptive adversary,

$$\mathcal{E}_n^{\mathcal{A},\text{adpt}}(t) \stackrel{\text{def}}{=} \max_{\mathcal{C}\in\mathcal{C}_{\text{adpt}}} \max_{\mathcal{X}} \mathcal{E}_n^{\mathcal{A}}(t).$$

For the non-adaptive adversary, the minimum t-approximation error of any algorithm is

$$\mathcal{E}_n^{\text{non}}(t) \stackrel{\text{def}}{=} \min_{\mathcal{A}} \mathcal{E}_n^{\mathcal{A},\text{non}}(t),$$

and, similarly, for the adaptive adversary,

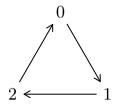
$$\mathcal{E}_n^{\text{adpt}}(t) \stackrel{\text{def}}{=} \min_{\mathcal{A}} \mathcal{E}_n^{\mathcal{A},\text{adpt}}(t).$$

Since adaptive adversarial comparators are stronger than non-adaptive, for all t,

$$\mathcal{E}_n^{\text{adpt}}(t) \geq \mathcal{E}_n^{\text{non}}(t).$$

Example 1 shows that $\mathcal{E}_3^{\text{non}}(t) \geq \frac{1}{3}$ for all $t < 2$.

Example 1 *$\mathcal{E}_3^{\text{non}}(t) \geq \frac{1}{3}$ for all $t < 2$. Consider $\mathcal{X} = \{0, 1, 2\}$ and the following comparators. .*



By symmetry, no algorithm can differentiate between the three inputs. Hence, any algorithm will output 0 with probability $1/3$.

3. Previous and new results

In Section 4.1 we lower bound $\mathcal{E}_n^{\text{non}}(t)$ as a function of t. In Lemma 2, we show that for all $t < 1$ and odd n, $\mathcal{E}_n^{\text{non}}(t) = 1 - 1/n$, namely for some $\mathcal{X}$, approximating the maximum to within less than one is equivalent to guessing a random x_i as the maximum. In Lemma 3, we modify Example 1 and show that for all $t < 2$ and odd n, any algorithm has t-approximation error close to $1/2$ for some input.

We propose a number of algorithms to approximate the maximum. These algorithms have different guarantees in terms of the probability of error, approximation factor, and query complexity.

We first consider two simple algorithms: the complete tournament, denoted COMPL, and the sequential selection, denoted SEQ. Algorithm COMPL compares all the possible input pairs and declares the input with the most wins as the maximum. We show the simple result that COMPL outputs a 2-approximation of x^*. We then consider the algorithm SEQ that compares a pair of inputs, discards the loser, and compares the winner with a new input. We show that even under random selection of the inputs, there exist inputs such that, with high probability, SEQ cannot provide a constant approximation to x^*.

We then consider more advanced algorithms. The knock-out algorithm, at each stage, pairs the inputs at random and keeps the winners of the comparisons for the next stage. We design a slight modification of this algorithm, denoted KO-MOD that achieves a 3-approximation with error probability at most ϵ, even against adaptive adversarial comparators. We note that Ajtai et al. (2015) proposed a different algorithm with similar performance guarantees.

Motivated by quick-sort, we propose a quick-select algorithm Q-SELECT that outputs a 2-approximation with zero error probability. It has an expected query complexity of at most $2n$ against the non-adaptive adversary. However, in Example 2, we see that this algorithm requires $\binom{n}{2}$ queries against the adaptive adversary.

This leaves the question of whether there is a randomized algorithm for 2-approximation of x^* with $\mathcal{O}(n)$ queries against the adaptive adversary. In fact, Ajtai et al. (2015) pose this as an open question. We resolve this problem by designing an algorithm COMB that combines quick-select and knock-out. We prove that COMB outputs a 2-approximation with probability of error, at most, ϵ, using $\mathcal{O}(n \log \frac{1}{\epsilon})$ queries. We summarize the results in Table 1.

We note that while we focus on randomized algorithms, Ajtai et al. (2015) also studied the best possible trade-offs for deterministic algorithms. They designed a deterministic

Algorithm	Notation	Approximation	$q_n^{\mathcal{A},\text{non}}$	$q_n^{\mathcal{A},\text{adpt}}$
complete tournament	COMPL	$\mathcal{E}_n^{\text{COMPL},\text{adpt}}(2) = 0$	$\binom{n}{2}$	
deterministic upper bound (Ajtai et al., 2015)	-	$\mathcal{E}_n^{\mathcal{A},\text{adpt}}(2) = 0$	$\Theta(n^{\frac{3}{2}})$	
deterministic lower bound (Ajtai et al., 2015)	-	$\mathcal{E}_n^{\mathcal{A},\text{adpt}}(2) = 0$	-	$\Omega(n^{\frac{4}{3}})$
sequential	SEQ	$\mathcal{E}_n^{\text{SEQ},\text{non}}\left(\frac{\log n}{\log\log n} - 1\right) \to 1$	$n-1$	
modified knock-out	KO-MOD	$\mathcal{E}_n^{\text{KO-MOD},\text{adpt}}(3) < \epsilon$	$< n + \frac{1}{2}\log^4 n \left\lceil \frac{1}{\epsilon}\ln\frac{1}{\epsilon}\right\rceil^2$	
quick-select	Q-SELECT	$\mathcal{E}_n^{\text{Q-SELECT},\text{adpt}}(2) = 0$	$< 2n$	$\binom{n}{2}$
knock-out and quick-select combination	COMB	$\mathcal{E}_n^{\text{COMB},\text{adpt}}(2) < \epsilon$	$\mathcal{O}\left(n\log\frac{1}{\epsilon}\right)$	

Table 1: Maximum selection algorithms

algorithm for 2-approximation of the maximum using only $\mathcal{O}(n^{3/2})$ queries. Moreover, they prove that no deterministic algorithm with fewer than $\Omega(n^{4/3})$ queries can output a 2-approximation of x^* for the adaptive adversarial model.

4. Simple results

In Lemmas 2 and 3, we prove lower bounds on the error probability of any algorithm that provides a t-approximation of x^* for $t < 1$ and $t < 2$, respectively. We then consider two straightforward algorithms for finding the maximum. One is the complete tournament, where all pairs of inputs are compared, and the other is sequential, where inputs are compared sequentially, and the loser is discarded at each comparison.

4.1. Lower bounds

We show the following two results:

- $\mathcal{E}_n^{\text{non}}(t) = 1 - \frac{1}{n}$ for all $0 \le t < 1$ and odd n
- $\mathcal{E}_n^{\text{non}}(t) \ge \frac{1}{2} - \frac{1}{2n}$ for all $1 \le t < 2$ and odd n

These lower bounds can be applied to n, which is even, by adding an extra input that is smaller than all the other inputs and loses to them.

Lemma 2 *For all $0 \le t < 1$ and odd n,*

$$\mathcal{E}_n^{\text{non}}(t) = 1 - \frac{1}{n}.$$

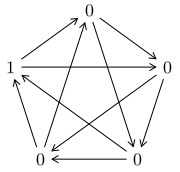


Figure 2: Tournament for Lemma 2 when $n = 5$

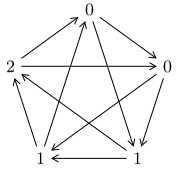


Figure 3: Tournament for Lemma 3 when $n = 5$

Proof Let $(x_1, x_2, \ldots, x_n)$ be an unknown permutation of $(1, \underbrace{0, \ldots, 0}_{n-1})$. Suppose we consider an adversary that ensures each input wins exactly $(n-1)/2$ times. An example is shown in Figure 2 for $n = 5$.

To get a lower bound on the performance of any randomized algorithm, we use Yao's principle. We consider only deterministic algorithms over a uniformly chosen permutation of the inputs, namely only one of the coordinates is 1, and the remaining are less than $1-t$. In this case, if we fix any comparison graph (as in Figure 2), and permute the inputs, the algorithm cannot distinguish between 1 and 0's, and outputs 0 with probability $1 - 1/n$; therefore, $\mathcal{E}_n^{\text{non}}(t) \ge 1 - \frac{1}{n}$. Also, an algorithm that randomly picks an element as the maximum achieves the error $1 - 1/n$; hence, the lemma. ■

Lemma 3 *For all $1 \le t < 2$ and odd n,*

$$\mathcal{E}_n^{\text{non}}(t) \ge \frac{1}{2} - \frac{1}{2n}.$$

Proof Let m be $(n-1)/2$. Let $(x_1, x_2, \ldots, x_n)$ be an unknown permutation of $(2, \underbrace{1, \ldots, 1}_{m}, \underbrace{0, \ldots, 0}_{m})$. Suppose the adversary ensures that 2 loses against all the 1's and, indeed, all inputs have exactly $(n-1)/2$ wins. An example is shown in Figure 3.

Similar to Lemma 2, the inputs are all identical to the algorithm, and, therefore, the algorithm outputs one of the 0's with probability $\frac{m}{n} = \frac{1}{2} - \frac{1}{2n}$. ■

4.2. Two elementary algorithms

In this section, we analyze two well-known maximum selection algorithms, the complete tournament and the sequential selection. We discuss their strengths and weaknesses and show that there is a trade-off between the query complexity and the approximation guarantees of these two algorithms. Another well-known algorithm for maximum selection is the knock-out algorithm, and we discuss a variant of it in Section 5.1.

4.2.1. Complete tournament (round-robin)

As its name evinces, a complete tournament involves a match between every pair of teams. Using this metaphor to competitions, we compare all the $\binom{n}{2}$ input pairs, and the input with maximum wins is declared as the output. If two or more inputs end up with the highest wins, any of them can be declared as the output. This algorithm is formally stated in COMPL.

input: $\mathcal{X}$
    compare all input pairs in $\mathcal{X}$, count the number of times each input wins
output: an input with the maximum number of wins

Algorithm COMPL - Complete tournament

Lemma 4 shows that COMPL gives a 2-approximation against both adversaries. The result, although weaker than the deterministic guarantees of Ajtai et al. (2015), is illustrative and useful in the algorithms proposed later.

Lemma 4 $q_n^{\text{COMPL,adpt}} = \binom{n}{2}$ *and* $\mathcal{E}_n^{\text{COMPL,adpt}}(2) = 0$.

Proof The number of queries is clearly $\binom{n}{2}$. To show $\mathcal{E}_n^{\text{COMPL,adpt}}(2) = 0$, note that if $y < x^* - 2$, then for all z that y wins over, $z \le y + 1 < x^* - 1$, and therefore x^* also beats them. Since x^* wins over y, it wins over more inputs than y, and y cannot be the output of the algorithm. It follows that the input with maximum wins is a 2-approximation of x^*. ■

COMPL is deterministic, and, after $\binom{n}{2}$ queries, it outputs a 2-approximation of x^*. If the comparators are noiseless, we can simply compare the inputs sequentially, discarding the loser at each step, and, thus, requiring only $n-1$ comparisons. This evokes the hope of finding a deterministic algorithm that requires a linear number of comparisons and outputs a 2-approximation of x^*. As mentioned earlier, however, Ajtai et al. (2015) showed it is not achievable, as they proved that any deterministic 2-approximation algorithm requires $\Omega(n^{4/3})$ queries. They also showed a strictly superlinear lower bound on any deterministic constant-approximation algorithm. They designed a deterministic 2-approximation algorithm using $\mathcal{O}(n^{3/2})$ queries.

4.2.2. Sequential selection

Sequential selection first compares a random pair of inputs and, at each successive step, compares the winner of the last comparison with a randomly chosen *new* input. It outputs the final remaining input. This algorithm uses $n-1$ queries.

input: $\mathcal{X}$
    choose a random $y \in \mathcal{X}$ and remove it from $\mathcal{X}$
    while $\mathcal{X}$ is not empty
        choose a random $x \in \mathcal{X}$ and remove it from $\mathcal{X}$
        $y \leftarrow \mathcal{C}(x, y)$
    end while
output: y

Algorithm SEQ - Sequential selection

Lemma 5 shows that even against the non-adaptive adversary, the algorithm cannot output a constant-approximation of x^*.

Lemma 5 *Let* $s = \frac{\log n}{\log\log n}$. *For all* $t < s$,

$$\mathcal{E}_n^{\text{SEQ,non}}(t) \ge 1 - \frac{1}{\log\log n}.$$

Proof Assume that s, $\log n$, and $\log\log n$ are integers and

$$x_i = \begin{cases} s & \text{for } i = 1, \\ s-1 & \text{for } i = 2, \dots, r, \\ s-2 & \text{for } i = r+1, \dots, r^2, \\ \vdots & \\ m & \text{for } i = r^{s-m-1}+1, \dots, r^{s-m}, \\ \vdots & \\ 0 & \text{for } i = r^{s-1}+1, \dots, r^s, \end{cases}$$

where $r = \log n$. Consider the following non-adaptive adversarial comparator:

$$\mathcal{C}(x_i, x_j) = \begin{cases} \max\{x_i, x_j\} & \text{if } |x_i - x_j| > 1, \\ \min\{x_i, x_j\} & \text{if } |x_i - x_j| \le 1. \end{cases} \tag{3}$$

The sequential algorithm takes a random permutation of the inputs. It then starts by comparing the first two elements and then sequentially compares the winner with the next element, and so on. Let L_j be the location in the permutation where input j appears for the last time. The next two observations follow from the construction of inputs and comparators respectively.

Observation 1 *Input* j *appears at least* $(\log n - 1)$ *times that of input* $j+1$.

Observation 2 *For the adversarial comparator defined in* (3), *if* $L_0 > L_1 > \ldots > L_s$, *then no input* j *can survive beyond location* L_{j-1}, *and, therefore,* SEQ *outputs* 0.

As a consequence of Observation 1, in the random permutation of inputs, $L_j > L_{j+1}$ with probability at least $1-\frac{1}{\log n}$. By the union bound, $L_0 > L_1 > \ldots > L_s$ with probability at least,

$$1 - \frac{s}{\log n} = 1 - \frac{1}{\log\log n}.$$

By applying Observation 2, SEQ outputs 0 with probability at least $1 - \frac{1}{\log\log n}$. ■

5. Algorithms

In the previous section, we saw that the complete tournament, COMPL, always outputs a 2-approximation but has quadratic query complexity, while the sequential selection, SEQ, has linear query complexity but a poor approximation guarantee. A natural question to ask is whether there exist algorithms with bounded error and linear query complexity. In this section, we propose algorithms with linear query complexity and approximation guarantees that compete with the best possible, namely, 2-approximation of x^*.

We propose three algorithms with different performance guarantees:

- **Modified knock-out**, described in Section 5.1, has linear query complexity, and, with high probability, outputs a 3-approximation of x^* against both adaptive and non-adaptive adversaries
- **Quick-select**, described in Section 5.2, outputs a 2-approximation to x^* (against both adversaries). It also has a linear expected query complexity against non-adaptive adversarial comparators
- **Knock-out and quick-select combination**, described in Section 5.3, has linear query complexity, and, with high probability, outputs a 2-approximation of x^* even against adaptive adversarial comparators

We now go over these algorithms in detail.

5.1. Modified knock-out

For simplification, in this section, we assume that $\log n$ is an integer. The knock-out algorithm derives its name from knock-out competitions where the tournament is divided into $\log n$ successive rounds. In each round, the inputs are paired at random, and the winners advance to the next round. Therefore, in round i, there are $\frac{n}{2^{i-1}}$ inputs. The winner at the end of $\log n$ rounds is declared as the maximum.

Under our adversarial model, at each round of the knock-out algorithm, the largest remaining input decreases by at most one. Therefore, the knock-out algorithm finds at least $\log n$-approximation of x^*. Analyzing the precise approximation error of knock-out algorithm appears to be difficult. However, simulations suggest that for any large n, for the set consisting of $0.2 \cdot n$ 0's, $\alpha \cdot n$ 1's, $(0.7-\alpha) \cdot n$ 2's, $0.1 \cdot n$ 3's, and a single 4, where $0 < \alpha < 0.7$ is an appropriately chosen parameter, the knock-out algorithm is not able to find a 3-approximation of x^* with positive constant probability. The problem with knock-out algorithm is that if at any of the $\log n$ rounds, many inputs are within 1 from the largest

input at that round, there is a fair chance that the largest input will be eliminated. If this elimination happens in several rounds, we will end up with a number significantly smaller than x^*.

To circumvent the problem of discarding large inputs, we select a specified number of inputs at each round and save them for the very end, thereby ensuring that at every round, if the largest input is eliminated, then an input within 1 from it has been saved. We then perform a complete tournament on these saved inputs. The algorithm is explained in KO-MOD.

input: $\mathcal{X}$
  pair the inputs of $\mathcal{X}$ randomly, let $\mathcal{X}'$ be the winners
output: $\mathcal{X}'$

Algorithm KO-SUB - Subroutine for KO-MOD and COMB

input: $\mathcal{X}, \epsilon$
  $\mathcal{Y} = \emptyset$, $n_1 = \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \cdot \log n \right\rceil$
  while $|\mathcal{X}| > n_1$
    randomly choose n_1 inputs from $\mathcal{X}$ and *copy* them to $\mathcal{Y}$
    $\mathcal{X} \leftarrow$ KO-SUB($\mathcal{X}$)
  end while
output: COMPL($\mathcal{X} \cup \mathcal{Y}$)

Algorithm KO-MOD - Modified knock-out algorithm

In Theorem 6, we show that KO-MOD has a 3-approximation error less than ϵ.

We first explain the algorithm and then state the result. Let $n_1 \stackrel{\text{def}}{=} \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \cdot \log n \right\rceil$. At each round, we add n_1 of the remaining inputs at random to the multiset $\mathcal{Y}$ and run the knock-out subroutine KO-SUB on the multiset $\mathcal{X}$. When $|\mathcal{X}| \le n_1$, we perform a complete tournament on $\mathcal{X} \cup \mathcal{Y}$ and declare the output as the winner. We show that, with probability at least $1-\epsilon$, the final set $\mathcal{Y}$ contains at least one input which is a 1-approximation of x^*. Since the complete tournament outputs a 2-approximation of its maximum input, KO-MOD outputs a 3-approximation of x^* with probability greater than $1-\epsilon$.

Theorem 6 *For $n_1 \ge 2$, we have $q_n^{\text{KO-MOD,adpt}} < n + \frac{1}{2}(\log^4 n) \cdot \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \right\rceil^2$ and $\mathcal{E}_n^{\text{KO-MOD,adpt}}(3) < \epsilon$.*

Proof The number of comparisons made by KO-SUB is at most $\frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \ldots < n$. Observe that KO-SUB is called $m \stackrel{\text{def}}{=} \left\lceil \log \frac{n}{n_1} \right\rceil$ times. Let $\mathcal{X}_i$ be the multiset $\mathcal{X}$ at the start of the ith call to KO-SUB. Let $\mathcal{X}_{m+1}$ and $\mathcal{Y}_{m+1}$ be the multisets $\mathcal{X}$ and $\mathcal{Y}$ right before calling

compl. Then,

$$\begin{aligned}
|\mathcal{X}_{m+1} \cup \mathcal{Y}_{m+1}| &\le |\mathcal{X}_{m+1}| + |\mathcal{Y}_{m+1}| \\
&\le n_1 + \sum_{i=1}^{m} \left(|\mathcal{Y}_{i+1}| - |\mathcal{Y}_i|\right) \\
&\le n_1 + m n_1 \\
&= \left(\left\lceil \log \frac{n}{n_1} \right\rceil + 1\right) \cdot \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \cdot \log n \right\rceil \\
&\le \left(\left\lceil \log \frac{n}{n_1} \right\rceil + 1\right) \cdot \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \right\rceil \lceil \log n \rceil \\
&\le \log^2 n \cdot \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \right\rceil,
\end{aligned}$$

where the last inequality follows as $n_1 \ge 2$ and $\log n$ is an integer. Since the complete tournament is quadratic in the input size, the total number of queries is at most $n + \frac{1}{2}\log^4 n \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \right\rceil^2$.

Next, we bound the error of ko-mod. Let

$$\mathcal{X}^* \stackrel{\text{def}}{=} \{x \in \mathcal{X} : x \ge x^* - 1\}$$

be the multiset of all inputs that are at least $x^* - 1$. For $i \le m+1$, let $\mathcal{X}_i^* = \mathcal{X}_i \cap \mathcal{X}^*$ and $\mathcal{Y}_{m+1}^* = \mathcal{Y}_{m+1} \cap \mathcal{X}^*$. Let $\alpha_i \stackrel{\text{def}}{=} \frac{|\mathcal{X}_i^*|}{|\mathcal{X}_i|}$ and $\alpha = \max\{\alpha_1, \alpha_2, \ldots, \alpha_m\}$. We show that, with high probability, $|\mathcal{X}_{m+1}^* \cup \mathcal{Y}_{m+1}^*| \ge 1$, namely, some input in $\mathcal{X}_{m+1} \cup \mathcal{Y}_{m+1}$ belongs to $\mathcal{X}^*$. In particular, we show that, with probability $1-\epsilon$, for large α, $|\mathcal{Y}_{m+1}^*| > 0$, and for small α, $x^* \in \mathcal{X}_{m+1}$. Observe that

$$\begin{aligned}
\Pr(x^* \notin \mathcal{X}_{m+1}^*) &= \sum_{i=1}^{m} \Pr(x^* \notin \mathcal{X}_{i+1}^* | x^* \in \mathcal{X}_i) \cdot \Pr(x^* \in \mathcal{X}_i) \\
&\le \sum_{i=1}^{m} \Pr(x^* \notin \mathcal{X}_{i+1}^* | x^* \in \mathcal{X}_i) \\
&\stackrel{(a)}{\le} \sum_{i=1}^{m} \frac{|\mathcal{X}_i^*| - 1}{|\mathcal{X}_i| - 1} \\
&\le \sum_{i=1}^{m} \alpha_i \\
&\le \alpha m,
\end{aligned}$$

where (a) follows since at round i, ko-sub randomly pairs the inputs and only inputs in $\mathcal{X}_i^* \backslash \{x^*\}$ are able to eliminate x^*. Next we discuss $\Pr(|\mathcal{Y}_{m+1}^*| = 0)$. At round i, the probability that an input in $\mathcal{X}^*$ is not picked up in $\mathcal{Y}$ is

$$\frac{\binom{|\mathcal{X}_i| - |\mathcal{X}_i^*|}{n_1}}{\binom{|\mathcal{X}_i|}{n_1}} \le \left(1 - \frac{|\mathcal{X}_i^*|}{|\mathcal{X}_i|}\right)^{n_1} = (1-\alpha_i)^{n_1}.$$

Therefore,

$$\begin{aligned}
\Pr(|\mathcal{Y}_{m+1}^*| = 0) &\le \prod_{i=1}^{m} (1-\alpha_i)^{n_1} \\
&\le \min_i (1-\alpha_i)^{n_1} \\
&= (1-\alpha)^{n_1}.
\end{aligned}$$

As a result,

$$\begin{aligned}
\Pr(|\mathcal{X}_{m+1}^* \cup \mathcal{Y}_{m+1}^*| = 0) &= \Pr(|\mathcal{X}_{m+1}^*| = 0 \ \wedge \ |\mathcal{Y}_{m+1}^*| = 0) \\
&\le \Pr(x^* \notin \mathcal{X}_{m+1}^* \ \wedge \ |\mathcal{Y}_{m+1}^*| = 0) \\
&\le \max_\alpha \min\{\Pr(x^* \notin \mathcal{X}_{m+1}^*), \Pr(|\mathcal{Y}_{m+1}^*| = 0)\} \\
&\le \max_\alpha \min\left\{\alpha m, (1-\alpha)^{n_1}\right\} \\
&\stackrel{(a)}{\le} \max\left\{\alpha m, (1-\alpha)^{n_1}\right\}|_{\alpha = \frac{\epsilon}{\log n}} \\
&= \max\left\{\frac{\epsilon m}{\log n}, \left(1 - \frac{\epsilon}{\log n}\right)^{n_1}\right\} \\
&\stackrel{(b)}{<} \epsilon,
\end{aligned}$$

where (a) follows since the first argument of the min increases and the second argument decreases with α. Also, (b) follows since $m \le \log n$ and $n_1 = \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \log n \right\rceil$.

So far, we have shown that with probability $1-\epsilon$, there exists a 1-approximation of x^* in $\mathcal{X}_{m+1} \cup \mathcal{Y}_{m+1}$. From Lemma 4, compl gives a 2-approximation of the maximum input. Consequently, with probability $1-\epsilon$, ko-mod outputs a 3-approximation of x^*. ■

In Appendix A, we show that ko-mod cannot output better than 3-approximation of x^* with constant probability.

5.2. Quick-select

Motivated by quick-sort, we propose a quick-select algorithm q-select that at each round compares all the inputs with a random pivot to provide stronger performance guarantees against the non-adaptive adversary.

input: $\mathcal{X}$
pick a pivot $x_p \in \mathcal{X}$ at random
compare x_p with all other inputs in $\mathcal{X}$
let $\mathcal{Y} \subset \mathcal{X} \backslash \{x_p\}$ be the multiset of inputs that beat x_p
output: if $\mathcal{Y} \neq \emptyset$ output $\mathcal{Y}$ otherwise output $\{x_p\}$

Algorithm qs-sub - Subroutine for q-select and comb

We show that q-select provides a 2-approximation with no error against both the adaptive and non-adaptive adversaries. To show this result, observe that x^* will only be

input: $\mathcal{X}$
  while $|\mathcal{X}| > 1$
    $\mathcal{X} \leftarrow$ QS-SUB($\mathcal{X}$)
  end while
output: the unique input in $\mathcal{X}$

Algorithm Q-SELECT - Quick-select

eliminated if a 1-approximation of x^* is chosen as pivot, and, therefore, only inputs that are 2-approximation of x^* will survive.

Lemma 7 $\mathcal{E}_n^{\text{Q-SELECT,adpt}}(2) = 0$.

Proof If the output is x^*, the lemma holds. Otherwise, x^* is discarded when it was chosen as a pivot or compared with a pivot. Let x_p be the pivot when x^* is discarded; hence, $x_p \geq x^* - 1$. By the algorithm's definition, all the surviving inputs are at least $x_p - 1 \geq x^* - 2$. ■

We now show that the expected query complexity of Q-SELECT against a non-adaptive adversary is at most $2n$. This result follows from the observation that the non-adaptive adversary fixes the comparison graph at the beginning, and hence a random pivot wins against half of the inputs in expectation. This idea is made rigorous in the proof of Lemma 8.

In Example 2 we show an instance for which Q-SELECT requires $\binom{n}{2}$ queries against the adaptive adversary.

Lemma 8 $q_n^{\text{Q-SELECT,non}} < 2n$.

Proof Recall that the non-adaptive adversary can be modeled as a complete directed graph where each node is an input and there is an edge from x to y if $\mathcal{C}(x, y) = x$. Let $\text{in}(x)$ be the in-degree of x in such a graph.

At round i, the algorithm chooses a pivot x_p at random and compares it to all the remaining inputs. By keeping the winners, $\max\{\text{in}(x_p), 1\}$ inputs will remain for the next round. As a result, we have the following recursion for non-adaptive adversaries:

$$\begin{aligned} q_n^{\text{Q-SELECT}} &= \mathbb{E}\left[Q_n^{\text{Q-SELECT}}\right] \\ &= n - 1 + \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}\left[Q_{\text{in}(x_i)}^{\text{Q-SELECT}}\right] \\ &= n - 1 + \frac{1}{n}\sum_{i=1}^{n} q_{\text{in}(x_i)}^{\text{Q-SELECT}}. \end{aligned}$$

By (2),

$$\begin{aligned} q_n^{\text{Q-SELECT,non}} &= \max_{\mathcal{C}\in\mathcal{C}_{\text{non}}} \max_{\mathcal{X}} q_n^{\text{Q-SELECT}} \\ &= \max_{\mathcal{C}\in\mathcal{C}_{\text{non}}} \max_{\mathcal{X}} \left[n - 1 + \frac{1}{n}\sum_{i=1}^{n} q_{\text{in}(x_i)}^{\text{Q-SELECT}}\right] \\ &\leq n - 1 + \frac{1}{n}\sum_{i=1}^{n} \max_{\mathcal{C}\in\mathcal{C}_{\text{non}}} \max_{\mathcal{X}} q_{\text{in}(x_i)}^{\text{Q-SELECT}} \\ &= n - 1 + \frac{1}{n}\sum_{i=1}^{n} q_{\text{in}(x_i)}^{\text{Q-SELECT,non}}, \end{aligned} \tag{4}$$

where the inequality follows as the maximum of sums is at most the sum of maxima. We prove by strong induction that $q_n^{\text{Q-SELECT,non}} \leq 2(n-1)$, which holds for $n = 1$. Suppose it holds for all $n' < n$, then,

$$\begin{aligned} q_n^{\text{Q-SELECT,non}} &\leq n - 1 + \frac{1}{n}\sum_{i=1}^{n} q_{\text{in}(x_i)}^{\text{Q-SELECT,non}} \\ &\leq n - 1 + \frac{1}{n}\sum_{i=1}^{n} 2 \cdot \text{in}(x_i) \\ &= n - 1 + \frac{n(n-1)}{n} \\ &\leq 2(n-1), \end{aligned}$$

where the equality follows since the in-degrees sum to $\frac{n(n-1)}{2}$. ■

Lemma 8 shows that $q_n^{\text{Q-SELECT,non}} < 2n$. Next, we show a naive concentration bound for the query complexity of Q-SELECT. By Markov's inequality, for a non-adaptive adversary,

$$\Pr(Q_n^{\text{Q-SELECT}} > 4n) \leq \frac{1}{2}.$$

Let k be an integer multiple of 4. Now suppose we run Q-SELECT, allowing kn queries. At each $4n$ queries, the Q-SELECT ends with probability $\geq \frac{1}{2}$. Therefore,

$$\Pr(Q_n^{\text{Q-SELECT}} > kn) \leq 2^{-\frac{k}{4}}.$$

This naive bound is exponential in k. The next lemma shows a tighter super-exponential concentration bound on the query complexity of the algorithm beyond its expectation. We defer the proof to appendix B.

Lemma 9 *Let $k' = \max\{e, k/2\}$. For a non-adaptive adversary, $Pr(Q_n^{\text{Q-SELECT}} > kn) \leq e^{-(k-k')\ln k'}$.*

While Q-SELECT has linear expected query complexity under the non-adaptive adversarial model, the following example suggested to us by Nelson (2015) shows that it has a quadratic query complexity against an adaptive adversary.

Example 2 *Let $\mathcal{X} = \{0, 0, \ldots, 0\}$. At each round, the adversary declares the pivot to be smaller than all the other inputs. Consequently, only the pivot is eliminated, and the query complexity is $\binom{n}{2}$.*

5.3. Knock-out and quick-select combination

KO-MOD has the benefit of reducing the number of inputs exponentially at each round and therefore maintaining a linear query-complexity while having only a 3-approximation guarantee. On the other side, Q-SELECT has a 2-approximation guarantee while it may require $\mathcal{O}(n^2)$ queries for some instances of inputs. In COMB we combine the benefits of these algorithms and avoid their shortcomings. By carefully repeating QS-SUB, we try to reduce the number of inputs by a fraction at each round and keep the largest element in the remaining set. If the number of inputs is not reduced by a fraction, most of them must be close to each other. Therefore, repeating the KO-SUB for a sufficient number of times and keeping the inputs with the higher number of wins will guarantee the reduction of the input size without making the approximation error worse. Our final algorithm COMB provides a 2-approximation of x^*, even against the adaptive-adversarial comparator and has linear query complexity. Therefore, an open question of Ajtai et al. (2015) is resolved.

input: $\mathcal{X}, \epsilon$
  $\beta_1 = 9$, $\beta_2 = 25$, $i = 0$
  while $|\mathcal{X}| > 1$
    $i = i + 1$ (i is the round)
    $n_i = |\mathcal{X}|$
    run $\mathcal{X} \leftarrow$ QS-SUB($\mathcal{X}$) for $\lfloor \beta_1 \log \frac{1}{\epsilon} \rfloor$ times
    $\mathcal{X}_i = \mathcal{X}$
    if $|\mathcal{X}_i| > \frac{2}{3} n_i$
      run KO-SUB on *fixed* $\mathcal{X}$ for $\left\lfloor \beta_2 \left(\frac{4}{3}\right)^i \log \frac{1}{\epsilon} \right\rfloor$ times
      if there exists an input with $> \frac{3}{4} \left\lfloor \beta_2 \left(\frac{4}{3}\right)^i \log \frac{1}{\epsilon} \right\rfloor$ wins
        let $\mathcal{X}$ be a multiset of inputs with $> \frac{3}{4} \left\lfloor \beta_2 \left(\frac{4}{3}\right)^i \log \frac{1}{\epsilon} \right\rfloor$ wins
      else
        let $\mathcal{X}$ be an input with highest number of wins
  end while
output: $\mathcal{X}$

Algorithm COMB - Knock-out and quick-select combination

We begin the algorithm's analysis with a few lemmas.

Lemma 10 *At each round $|\mathcal{X}|$ reduces by at least a third, namely, $n_{i+1} \le \frac{2}{3} n_i$.*

Proof If at any round $|\mathcal{X}_i| \le \frac{2}{3} n_i$, then the lemma holds, and the algorithm does not call KO-SUB. On the other hand, if KO-SUB is called, then by Markov's inequality at most two-thirds of the inputs win more than three-fourth of the queries. As a result, at round i, at least one-third of the inputs in $\mathcal{X}$ will be eliminated. ■

Recall that $\mathcal{X}^* = \{x \in \mathcal{X} : x \ge x^* - 1\}$. Lemma 11 shows that choosing inputs inside $\mathcal{X}^*$ as a pivot guarantees a 2-approximation of x^*. The proof is similar to Lemma 7 and is omitted.

Lemma 11 *If $x^* \in \mathcal{X}$, at a call to* QS-SUB *either x^* survives or a pivot from $\mathcal{X}^*$ is chosen where in the latter case, only inputs that are 2-approximation of x^* will survive.*

We showed that at each round, COMB reduces $|\mathcal{X}|$ by at least a third. As a result, the number of inputs decreases exponentially, and the total number of queries is linear in n. We also show that if x^* is eliminated at some round, then, with high probability, the pivot at that round is an input from $\mathcal{X}^*$. Using Lemma 11, this implies that COMB outputs a 2-approximation of x^* with high probability.

Theorem 12 $q_n^{\text{COMB,adpt}} = \mathcal{O}\left(n \log \frac{1}{\epsilon}\right)$ *and* $\mathcal{E}_n^{\text{COMB,adpt}}(2) < \epsilon$.

Proof We start by analyzing the query complexity of COMB. By Lemma 10,

$$n_i \le n \cdot \left(\tfrac{2}{3}\right)^{i-1}.$$

Therefore, the total number of queries at round i is at most

$$n \left(\tfrac{2}{3}\right)^{i-1} \beta_1 \log \tfrac{1}{\epsilon} + \tfrac{n}{2} \left(\tfrac{2}{3}\right)^{i-1} \beta_2 \left(\tfrac{4}{3}\right)^{i} \log \tfrac{1}{\epsilon},$$

where the first term is for calls to QS-SUB, and the second term is for calls to KO-SUB. Adding the query complexity of all the rounds,

$$\begin{aligned} q_n^{\text{COMB,adpt}} &\le n \log \tfrac{1}{\epsilon} \sum_{i=1}^{\infty} \left(\beta_1 \left(\tfrac{2}{3}\right)^{i-1} + \tfrac{2}{3} \beta_2 \left(\tfrac{8}{9}\right)^{i-1} \right) \\ &\le n(3\beta_1 + 6\beta_2) \log \tfrac{1}{\epsilon} \\ &= \mathcal{O}\left(n \log \tfrac{1}{\epsilon}\right). \end{aligned}$$

We now analyze the approximation guarantee of COMB. We show that at least one of the following events happens with probability greater than $1 - \epsilon$.

- COMB outputs x^*.
- An input inside $\mathcal{X}^*$ is chosen as a pivot at some round.

Let $\mathcal{X}_i^* \stackrel{\text{def}}{=} \mathcal{X}_i \cap \mathcal{X}^*$ and $\alpha_i \stackrel{\text{def}}{=} \frac{|\mathcal{X}_i^*|}{|\mathcal{X}_i|}$. We consider the following two cases separately.

- **Case 1** There exists an i such that $|\mathcal{X}_i| > \frac{2}{3} n_i$ and $\alpha_i > \frac{1}{8}$.
- **Case 2** For all i, either $|\mathcal{X}_i| \le \frac{2}{3} n_i$ or $\alpha_i \le \frac{1}{8}$.

First, we consider Case 1. We show that in this case a pivot from $\mathcal{X}^*$ is chosen with probability $> 1 - \epsilon$. Observe that at round i, $|\mathcal{X}|$ starts at $n_i < \frac{3}{2}|\mathcal{X}_i|$ and gradually decreases. On the other hand, in all the $\lfloor \beta_1 \log \frac{1}{\epsilon} \rfloor$ calls to QS-SUB, $|\mathcal{X} \cap \mathcal{X}^*|$ is at least $|\mathcal{X}_i^*| = \alpha_i |\mathcal{X}_i|$. Therefore, in all the calls to QS-SUB at round i,

$$\frac{|\mathcal{X} \cap \mathcal{X}^*|}{|\mathcal{X}|} \ge \frac{\alpha_i |\mathcal{X}_i|}{\frac{3}{2}|\mathcal{X}_i|} = \frac{2}{3} \alpha_i.$$

Let E be the event of not choosing a pivot from $\mathcal{X}^*$ at round i. As a result,

$$\begin{aligned}\Pr(E) &\leq \left(1-\tfrac{2}{3}\alpha_i\right)^{\left\lfloor \beta_1 \log \frac{1}{\epsilon}\right\rfloor} \\ &\leq \left(\tfrac{11}{12}\right)^{8\log\frac{1}{\epsilon}} \\ &< \epsilon.\end{aligned}$$

Therefore, in Case 1, with probability at least $1-\epsilon$, a pivot from $\mathcal{X}^*$ is chosen.

We now consider Case 2. By Lemma 11, during the calls to QS-SUB, either x^* survives or an input from $\mathcal{X}^*$ is chosen as a pivot. Therefore, we may only lose x^* without choosing a pivot from $\mathcal{X}^*$, if at some round i, $|\mathcal{X}_i| > \frac{2}{3}n_i$ and x^* wins less than three-fourth of its queries during the calls to KO-SUB.

Recall that in Case 2, if $|\mathcal{X}_i| > \frac{2}{3}n_i$ then $\alpha_i \leq \frac{1}{8}$. Observe that x^* wins against a random input in $\mathcal{X}_i$ with probability greater than $> 1-\alpha_i$, which is at least seven-eighths. Let E_i' be the event that x^* wins fewer than three-quarters of its queries at round i. By the Chernoff bound,

$$\begin{aligned}\Pr(E_i') &\leq \exp\left(-\left\lfloor \beta_2 \left(\tfrac{4}{3}\right)^i \log\tfrac{1}{\epsilon}\right\rfloor \cdot D\left(\tfrac{3}{4}\|\tfrac{7}{8}\right)\right) \\ &\leq \epsilon^{2\left(\frac{4}{3}\right)^i},\end{aligned}$$

where $D(p||q) \stackrel{\text{def}}{=} p\ln\frac{p}{q} + (1-p)\ln\frac{1-p}{1-q}$ is the Kullback-Leibler distance between Bernoulli distributed random variables with parameters p and q, respectively. Assuming $\epsilon < \frac{1}{2}$, the total probability of missing x^* without choosing a pivot form $\mathcal{X}^*$ is at most

$$\begin{aligned}\sum_{i=1}^{\infty}\Pr(E_i') &\leq \sum_{i=1}^{\infty}\epsilon^{2\left(\frac{4}{3}\right)^i} \\ &< \epsilon.\end{aligned}$$

This shows that with probability $> 1-\epsilon$, either x^* survives or an input inside $\mathcal{X}^*$ is chosen as a pivot. The theorem follows from Lemma 11. ■

6. Application to density estimation

Our study of maximum selection with adversarial comparators was motivated by the following density estimation problem:

Given a *known* set $\mathcal{P}_\delta = \{p_1, \ldots, p_n\}$ of n distributions and k samples from an *unknown* distribution p_0, output a distribution $\hat{p} \in \mathcal{P}_\delta$ such that for a small constant $C > 1$ and with high probability,

$$\|\hat{p} - p_0\|_1 \leq C \cdot \min_{p\in\mathcal{P}_\delta}\|p - p_0\|_1 + o_k(1).$$

This problem was studied in Devroye and Lugosi (2001) who showed that for $n = 2$, the SCHEFFE-TEST, described below in pseudocode, takes k samples and, with probability $1-\varepsilon$,

outputs a distribution $\hat{p} \in \mathcal{P}_\delta$ such that

$$||\hat{p} - p_0||_1 \leq 3 \cdot \min_{p\in\mathcal{P}_\delta}||p - p_0||_1 + \sqrt{\frac{10\log\frac{1}{\varepsilon}}{k}}. \tag{5}$$

input: distributions p_1 and p_2, k *i.i.d.* samples of unknown distribution p_0
let $\mathcal{S} = \{x : p_1(x) > p_2(x)\}$
let $p_1(\mathcal{S})$ and $p_2(\mathcal{S})$ be the probability mass that p_1 and p_2 assign to $\mathcal{S}$
let $\mu_{\mathcal{S}}$ be the frequency of samples in $\mathcal{S}$
output: if $|p_1(\mathcal{S}) - \mu_{\mathcal{S}}| \leq |p_2(\mathcal{S}) - \mu_{\mathcal{S}}|$ output p_1, otherwise output p_2

Algorithm SCHEFFE-TEST- Scheffe test for two distributions

SCHEFFE-TEST provides a factor-3 approximation with high probability. The algorithm, as stated in its pseudocode, requires computing $p_i(\mathcal{S})$ which can be hard since the distributions are not restricted. However, as noted in Suresh et al. (2014), the algorithm can be made to run in time linear in k. Devroye and Lugosi (2001) also extended SCHEFFE-TEST for $n > 2$. Their proposed algorithm for $n > 2$ runs SCHEFFE-TEST for each pair of distributions in $\mathcal{P}_\delta$ and outputs the distribution with the maximum wins, where a distribution is a winner if it is the output of SCHEFFE-TEST. This algorithm is referred to as the Scheffe tournament. They showed that this algorithm finds a distribution $\hat{p} \in \mathcal{P}_\delta$ such that

$$||\hat{p} - p_0||_1 \leq 9\min_{p\in\mathcal{P}_\delta}||p - p_0||_1 + o_k(1),$$

and the running time is clearly $\Theta(n^2k)$—quadratic in the number of distributions.

Mahalanabis and Stefankovic (2008) showed that the optimal coefficients for the Scheffe algorithms are indeed 3 and 9 for $n = 2$ and $n > 2$, respectively. They proposed an algorithm with an improved factor-3 approximation for $n > 2$—still running in time $\Theta(n^2)$, however. They also proposed a linear-time algorithm, but it requires a preprocessing step that runs in time exponential in n.

Scheffe's method has been used recently to obtain nearly sample optimal algorithms for learning Poisson Binomial distributions (Daskalakis et al., 2012), and Gaussian mixtures (Daskalakis and Kamath, 2014; Suresh et al., 2014).

We now describe how our noisy comparison model can be applied to this problem to yield a linear-time algorithm with the same estimation guarantee as the Scheffe tournament. Our algorithm uses the Scheffe test as a subroutine. Given a sufficient number of samples, $k = \Theta(\log n)$, the small term in the RHS of (5) vanishes, and SCHEFFE-TEST outputs

$$\begin{cases} p_i & \text{if } ||p_i - p_0||_1 < \frac{1}{3}||p_j - p_0||_1, \\ p_j & \text{if } ||p_j - p_0||_1 < \frac{1}{3}||p_i - p_0||_1, \\ \text{unknown} & \text{otherwise.}\end{cases}$$

Let $x_i = -\log_3||p_i - p_0||_1$, then analogously to the maximum selection with adversarial noise in (1), SCHEFFE-TEST outputs

$$\begin{cases} \max\{x_i, x_j\} & \text{if } |x_i - x_j| > 1, \\ \text{unknown} & \text{otherwise.}\end{cases}$$

Given a fixed multiset of samples the tournament results are fixed; hence, this setup is identical to the non-adaptive adversarial comparators. In particular, with probability $1-\varepsilon$, our quick-select algorithm can find $\hat{p} \in \mathcal{P}_\delta$ such that

$$||\hat{p} - p_0||_1 \leq 9 \cdot \min_{p \in \mathcal{P}_\delta} ||p - p_0||_1 ,$$

with running time $\Theta(nk)$. Next, we consider the combination of SCHEFFE-TEST and Q-SELECT in greater detail.

Theorem 13 *Combination of* SCHEFFE-TEST *and* Q-SELECT *algorithms, with probability* $1-\varepsilon$*, results in* $\hat{p}$ *such that*

$$||\hat{p} - p_0||_1 \leq 9 \cdot \min_{p \in \mathcal{P}_\delta} ||p - p_0||_1 + 4\sqrt{\frac{10 \log \frac{\binom{n}{2}}{\varepsilon}}{k}}.$$

Proof Let

$$p^* \stackrel{\text{def}}{=} \operatorname*{argmin}_{p \in \mathcal{P}_\delta} ||p - p_0||_1 .$$

Using (5), for each p_i and p_j in $\mathcal{P}_\delta$, with probability $1 - \varepsilon/\binom{n}{2}$, SCHEFFE-TEST outputs $\hat{p}$ such that

$$||\hat{p} - p_0||_1 \leq 3 \cdot \min_{p \in \{p_i, p_j\}} ||p - p_0||_1 + \sqrt{\frac{10 \log \frac{\binom{n}{2}}{\varepsilon}}{k}}. \tag{6}$$

By the union bound (6) holds for all p_i and p_j with probability at least $1-\varepsilon$. Similar to Lemma 7, if p^* is eliminated, then at some round, Q-SELECT has chosen p' as a pivot such that

$$\left|\left|p' - p_0\right|\right|_1 \leq 3 \cdot ||p^* - p_0||_1 + \sqrt{\frac{10 \log \frac{\binom{n}{2}}{\varepsilon}}{k}}.$$

Now after choosing p' as a pivot, for any distribution p'' that survives,

$$\begin{aligned}\left|\left|p'' - p_0\right|\right|_1 &\leq 3 \cdot \left|\left|p' - p_0\right|\right|_1 + \sqrt{\frac{10 \log \frac{\binom{n}{2}}{\varepsilon}}{k}} \\ &\leq 9 \cdot ||p^* - p_0||_1 + 4\sqrt{\frac{10 \log \frac{\binom{n}{2}}{\varepsilon}}{k}}.\end{aligned}$$

■

7. Noisy sorting

7.1. Problem statement

In this section, we consider sorting with noisy comparators. The comparator model is the same as before, and the goal is to approximately sort the inputs in decreasing order.

Consider an Algorithm $\mathcal{A}$ for sorting the inputs. The output of $\mathcal{A}$ is denoted by $\mathbf{Y}_{\mathcal{A}}(\mathcal{X}) \stackrel{\text{def}}{=} (Y_1, Y_2, \ldots, Y_n)$, a particular ordering of the inputs. Similar to the maximum-selection problem, a t-approximation error is

$$\mathcal{E}_n^{\mathcal{A}}(t) \stackrel{\text{def}}{=} \Pr\left(\max_{i,j:i>j} (Y_i - Y_j) > t\right),$$

namely, the probability of Y_i appearing after Y_j in $\mathbf{Y}_{\mathcal{A}}$ while $Y_i - Y_j > t$. Note that our definitions for $\mathcal{E}_n^{\mathcal{A},\text{non}}(t)$, $\mathcal{E}_n^{\mathcal{A},\text{adpt}}(t)$, $q_n^{\mathcal{A},\text{adpt}}$, and $q_n^{\mathcal{A},\text{non}}$ hold the same as before.

In the following, we first revisit the complete tournament with a small modification for the sake of the sorting problem, and we show that, under the adaptive adversarial model, it has zero 2-approximation error and query complexity of $\binom{n}{2}$. We then discuss the quick-sort algorithm Q-SORT and show that it has zero 2-approximation error but with improved query complexity for the non-adaptive adversary. We apply the known bounds for the running time of the general quick-sort algorithm with n distinct inputs to find the query complexity of Q-SORT.

7.2. Complete tournament

The algorithm is similar to COMPL in Section 4.2.1, and we refer to it as COMPL-SORT. The only difference is in the output of the algorithm.

input: $\mathcal{X}$
compare all input pairs in $\mathcal{X}$, count the number of times each input wins
output: output the inputs in the order of their number of wins, breaking the ties randomly

Algorithm COMPL-SORT - Complete tournament

The following lemma—and its proof—is similar to Lemma 4, and, therefore, we skip the proof.

Lemma 14 $q_n^{\text{COMPL-SORT},\text{adpt}} = \binom{n}{2}$ *and* $\mathcal{E}_n^{\text{COMPL-SORT},\text{adpt}}(2) = 0$.

Next, we discuss an algorithm with improved query complexity.

7.3. Quick-sort

Quick-sort is a well-known algorithm and, here, is denoted by Q-SORT. The expected query complexity of quick-sort with noiseless comparisons and distinct inputs is

$$f(n) \stackrel{\text{def}}{=} 2n \ln n - (4 - 2\gamma)n + 2 \ln n + \mathcal{O}(1), \tag{7}$$

where γ is Euler's constant (McDiarmid and Hayward, 1996). Note that $f(n)$ is a convex function of n.

In the rest of this section, we study the error guarantee of quick-sort and its query complexity in the presence of noise. In Lemma 15, we show that the error guarantee of quick-sort for our noise model is the same as the complete tournament, namely, it can sort

the inputs with zero 2-approximation error. Next, in Lemma 16, we show that the expected query complexity of quick-sort with non-adaptive adversarial noise is at most its expected query complexity in the noiseless model.

Lemma 15 $\mathcal{E}_n^{\text{Q-SORT,adpt}}(2) = 0$.

Proof The proof is by contradiction. Suppose $x_i > x_j + 2$, but x_j appears before x_i in the output of quick-sort algorithm. Then there must have been a pivot x_p such that $\mathcal{C}(x_i, x_p) = x_p$ while $\mathcal{C}(x_j, x_p) = x_j$. Since $x_i > x_j + 2$ no such a pivot exists. ■

The quick-sort algorithm chooses a pivot randomly to divide the set of inputs into smaller-size sets. The optimal pivot for noiseless quick-sort is known to be the median of the inputs to balance the size of the remained sets. In fact, it is easy to show that if we choose the median of the inputs as the pivot, the query complexity of quick-sort reduces to less than $n \log n$. Observe that in a non-adaptive adversarial model, the probability of having balanced sets after choosing pivot increases. As a result, in Lemma 16, we show that the expected query complexity of quick-sort in the presence of noise is upper bounded by $f(n)$.

Lemma 16 $q_n^{\text{Q-SORT,non}} = f(n)$ *and is achieved when the queries are noiseless and the inputs are distinct.*

Proof Let $\text{in}(x)$ and $\text{out}(x)$ be the in-degree and out-degree of node x in the complete tournament respectively. For the noiseless comparator with distinct inputs, the in-degrees and out-degrees of inputs are permutation of $(0, 1, \ldots, n-1)$. We show that

$$\operatorname*{argmax}_{\mathcal{C} \in \mathcal{C}_{\text{non}}} \max_{\mathcal{X}} q_n^{\text{Q-SORT}},$$

is a comparator whose complete tournament in-degrees and out-degrees are permutations of $(0, 1, \ldots, n-1)$. For notational simplicity let $q_n = q_n^{\text{Q-SORT,non}}$. We have the following recursion for quick-sort similar to (4):

$$q_n \le n - 1 + \frac{1}{n} \sum_{i=1}^{n} q_{\text{out}(x_i)} + q_{\text{in}(x_i)}. \tag{8}$$

By induction, we show that the solution to (8) is bounded above by $f(n)$, a convex function of n. The induction holds for $n = 0, 1$, and 2. Now suppose the induction holds for all $i < n$. Since $f(n)$ is a convex function of n and $\sum_i \text{in}(x_i) = \sum_i \text{out}(x_i) = \frac{n(n-1)}{2}$, the right hand side of (8) is maximized when the in-degrees and out-degrees take their extreme values, namely, when they are permutation of $(0, 1, \ldots, n-1)$. Plugging in these values, (8) is equivalent to,

$$\begin{aligned} q_n &\le n - 1 + \frac{1}{n} \sum_{i=1}^{n} f(\text{in}(x_i)) + f(\text{out}(x_i)) \\ &\le n - 1 + \frac{1}{n} \sum_{i=1}^{n} f(i-1) + f(n-i), \end{aligned}$$

where the solution to this recursion is $f(n)$, given in (7). Hence q_n is bounded above by $f(n)$, and the equality happens when the in-degrees and out-degrees are permutations of $(0, 1, \ldots, n-1)$. ■

Knuth (1998); Hennequin (1989); McDiarmid and Hayward (1992) show different concentration bounds for quick-sort. In particular, McDiarmid and Hayward (1992) show that the probability of the quick-sort algorithm requiring more comparisons than $(1+\epsilon)$ times its expected query complexity is $n^{-2\epsilon \ln \ln n + \mathcal{O}(\ln \ln \ln n)}$. Observe that for the non-adaptive adversarial model, the chance of a random pivot cutting the set of inputs into balanced sets increases. As a result, one can show that the analysis in McDiarmid and Hayward (1992) follows automatically. In particular, Lemmas 2.1 and 2.2 in McDiarmid and Hayward (1992), which are the basis of their analysis, are valid for our non-adaptive adversarial model. Therefore, their tight concentration bound for quick-sort algorithm can be applied to our non-adaptive adversarial model.

Acknowledgments

The authors would like to thank the editor and anonymous reviewers for their constructive comments, and Jelani Nelson for introducing the authors to the adaptive adversarial model. This work was supported in part by NSF grants CIF-1564355 and CIF-1619448. Part of this work was done while J.Acharya was at MIT, supported by MIT-Shell initiative.

Appendix A. For all $t < 3$, ko-mod cannot output a t-approximation

Example 3 shows that the modified knock-out algorithm cannot achieve better than 3-approximation of x^*.

Example 3 *Suppose $n-2$ is multiple of 3 and n is a large number. Let $\mathcal{X}$ be a random permutation of*

$$\{3, \underbrace{2, 2, \ldots, 2}_{\frac{n-2}{3}}, \underbrace{1, 1, \ldots, 1}_{\frac{n-2}{3}}, \underbrace{0, 0, \ldots, 0}_{\frac{n-2}{3}}, 0^*\}.$$

This multiset consists of an input with value zero but specified with 0^* since this input is going to behave differently from other 0s. Let the adversarial comparator be such that all 0s, except 0^*, and all 2s lose to all 1s, and 3 loses to all 2s. If two inputs of the same value get paired, one of them wins randomly (except in the case of 0^*). By the properties of comparator, it is obvious that any 2 will defeat all zeros, including 0^*. In order to prove our main claim, we make the following arguments and show that each of them happens with high probability:

- *Pr(input with value 3 is not present in the final multiset)*$> \frac{3}{10}$
- *Pr(input 0^* is present in the final multiset)*$> \frac{1}{3}$
- *With high probability, the fraction of 1s in the final multiset is close to 1*

Before proving each argument, we show why satisfying all the above statements are sufficient to prove our claim. Consider the final multiset; with high probability, it mainly consists of 1s, and there are a small number of 0s and 2s. Moreover, with probability greater than $\frac{1}{3} \times \frac{3}{10}$, input with value 3 has been removed before reaching the final multiset, and 0^* has survived to reach the final multiset. Therefore, if we run algorithm COMPL *on the final multiset, the input 0^* will have the most wins and be declared as the output. Hence for all $t < 3$, we have $\mathcal{E}_n^{\text{KO-MOD,non}}(t) >$ constant. Note that we did not try to optimize this constant.*

Now we show why each of the arguments above is true. Note that the reasoning made here is in expectation and assuming n is sufficiently large. However, the concentration bounds for all these claims are straightforward and thus omitted.

Lemma 17 *With high probability, the fraction of 1s in the final multiset is close to 1, and the fraction of 0s and 2s are very small.*

Proof *We calculate the expected number of 0s, 1s, and 2s at each step. Let $f_i(j)$ be the fraction of j's at the end of step i. After each step, we lose an input with value 1 if and only if they are paired with each other. As a result, we have the following recursion:*

$$f_{i+1}(1) = 2 \cdot f_i(1) \left(\tfrac{f_i(1)}{2} + 1 - f_i(1) \right),$$

where the factor 2 on the RHS of the recursion above is due to the fact that at each step we are reducing the number of inputs to half. Starting with $f_0(1) = 1/3$, we get the set of values $\{1/3, 5/9, 65/81, 6305/6561 \sim 0.96, ...\}$ for $f_i(1)$s. We can see that the ratio is approaching 1 very fast. More precisely, the fraction of 0s is decreasing quadratically since their only chance of survival is to get paired among themselves. As a result, after a couple of steps, the fraction of zeros is extremely small, and, henceforth, the only chance of survival for 2s becomes getting paired among themselves. Additionally their fraction is going to decrease quadratically afterward. As a result, more samples of 1s will be in the final $\mathcal{Y}$ with high probability. ■

Lemma 18 *Pr(input with value 3 is not present in the final multiset)$> \frac{3}{10}$.*

Proof *The input with value 3 is going to be removed when it is compared against one of the 2s. There is a slight chance of it surviving if it is chosen randomly for being in the output. Thus, the probability of input 3 being removed from the multiset in the first round is*

$$Pr(\text{input 3 is being removed in the first round}) = \frac{n-2}{3n}\left(1 - \frac{n_1}{n}\right) > \frac{3}{10},$$

where $n_1 = \left\lceil \frac{1}{\epsilon} \ln \frac{1}{\epsilon} \log n \right\rceil$. ■

Lemma 19 *Pr(input 0^* is present in the final multiset)$> \frac{1}{3}$.*

Proof *Similar to the argument made in the proof of Lemma 17, we have the following recursion for $f_i(2)$.*

$$f_{i+1}(2) = 2 \cdot f_i(2) \left(\tfrac{f_i(2)}{2} + 1 - f_i(2) - f_i(1) \right)$$

Thus, we have $f_0(2) = 1/3$, $f_1(2) = 1/3$, $f_2(2) = 5/27$, $f_3(2) = 85/2187$. As we stated in the proof of Lemma 17, the expected fraction of 2s is decreasing quadratically and

$$Pr(0^* \text{surviving}) = (1 - \tfrac{1}{3})(1 - \tfrac{1}{3})(1 - \tfrac{5}{27})(1 - \tfrac{85}{2187}) \cdots > \frac{1}{3},$$

proving the lemma. ■

Appendix B. Proof of Lemma 9

Abbreviate $Q_n^{\text{Q-SELECT}}$ by Q_n. As in the Chernoff bound proof, for all $\lambda > 0$,

$$\Pr(Q_n > kn) \le \frac{\mathbb{E}[e^{\lambda Q_n}]}{e^{k\lambda n}}. \tag{9}$$

Let $\lambda = \frac{1}{n} \ln k'$ and $\Phi(i) \stackrel{\text{def}}{=} \mathbb{E}[e^{\lambda Q_i}]$. We prove by induction that $\Phi(i) \le e^{k'\lambda i}$. The induction holds for $i = 0$. Similar to (4), we have the following recursion for $\Phi(n)$:

$$\begin{aligned}\Phi(n) &\le \frac{e^{\lambda(n-1)}}{n} \sum_{j=1}^{n} \Phi(\text{in}(x_j)) \\ &\le \frac{e^{\lambda n}}{n} \sum_{j=1}^{n} \Phi(\text{in}(x_j)).\end{aligned}$$

Since $\text{in}(x_j) < n$, using induction,

$$\frac{e^{\lambda n}}{n} \sum_{j=1}^{n} \Phi(\text{in}(x_j)) \le \frac{e^{\lambda n}}{n} \sum_{j=1}^{n} e^{k'\lambda \text{in}(x_j)}. \tag{10}$$

Observe that $e^{k'\lambda \text{in}(x_j)}$ is a convex function of $\text{in}(x_j)$, and $\sum_{j=1}^{n} \text{in}(x_j) = \frac{n(n-1)}{2}$. As a result, the RHS of (10) is maximized when the in-degrees take their extreme values, namely, any permutation of $(0, 1, \ldots, n-1)$. Therefore,

$$\begin{aligned}\frac{e^{\lambda n}}{n} \sum_{j=1}^{n} e^{k'\lambda \text{in}(x_j)} &\le \frac{e^{\lambda n}}{n} \sum_{j=0}^{n-1} e^{k'\lambda j} \\ &= \frac{e^{\lambda n}}{n} \frac{e^{k'\lambda n} - 1}{e^{k'\lambda} - 1}.\end{aligned}$$

Combining the above equations,

$$\Phi(n) \le \frac{e^{\lambda n}}{n} \frac{e^{k'\lambda n} - 1}{e^{k'\lambda} - 1}.$$

Similarly, by induction on $1 \le i < n$,

$$\Phi(i) \le \frac{e^{\lambda i}}{i} \frac{e^{k'\lambda i} - 1}{e^{k'\lambda} - 1}.$$

In Lemma 20 we show that for $1 \le i \le n$,

$$\frac{e^{\lambda i}}{i}\frac{e^{k'\lambda i}-1}{e^{k'\lambda}-1} \le e^{k'\lambda i}. \tag{11}$$

Therefore, $\Phi(i) \le e^{k'\lambda i}$ for $1 \le i \le n$, and, in particular, $\Phi(n) \le e^{k'\lambda n}$. Substituting $\mathbb{E}[e^{\lambda Q_n}] = \Phi(n)$ in (9),

$$\begin{aligned}\Pr(Q_n > kn) &\le \frac{e^{k'\lambda n}}{e^{k\lambda n}} \\ &= \frac{e^{k'\ln k'}}{e^{k\ln k'}} \\ &= e^{-(k-k')\ln k'}.\end{aligned}$$

This proves the lemma.
We now prove (11). Let $k' = \max\{e, \frac{k}{2}\}$ and $\lambda = \frac{1}{n}\ln k'$.

Lemma 20 *For all* $1 \le i \le n$, $\frac{e^{\lambda i}}{i}\frac{e^{k'\lambda i}-1}{e^{k'\lambda}-1} \le e^{k'\lambda i}$.

Proof It suffices to show that for all $0 < t \le n$,

$$f(t) \stackrel{\text{def}}{=} \frac{e^{\lambda t}}{t}\frac{1-e^{-k'\lambda t}}{e^{k'\lambda}-1} < 1.$$

Observe that

$$\lim_{t\to 0} f(t) = \frac{k'\lambda}{e^{k'\lambda}-1} \le 1.$$

On the other hand,

$$\begin{aligned}f(n) &= \frac{e^{\lambda n}}{n}\frac{1-e^{-k'\lambda n}}{e^{k'\lambda}-1} \\ &\le \frac{k'}{n}\frac{1}{e^{k'\ln k'/n}-1} \\ &\le \frac{k'}{n}\frac{n}{k'\ln k'} \\ &\le 1.\end{aligned}$$

Next, we show that $f(t)$ is convex. One can show that,

$$\ln\frac{1-e^{-u}}{u},$$

is a convex function of u. As a result,

$$\ln\frac{1-e^{-k'\lambda t}}{t},$$

is a convex function of t. Observe that $\ln e^{\lambda t}$ is also convex. Therefore,

$$\ln\frac{1-e^{-k'\lambda t}}{t} + \ln e^{\lambda t},$$

is convex. As a result, logarithm of $f(t)$ is convex, and, therefore, $f(t)$ is convex.

We showed that $f(t)$ is convex, $f(t \to 0) \le 1$, and $f(n) \le 1$. Therefore, for all $0 < t \le n$, $f(t) \le 1$. ■

References

Jayadev Acharya, Ashkan Jafarpour, Alon Orlitksy, and Ananda Theertha Suresh. Sorting with adversarial comparators and application to density estimation. In *Proceedings of the 2014 IEEE International Symposium on Information Theory (ISIT)*, 2014.

Micah Adler, Peter Gemmell, Mor Harchol-Balter, Richard M. Karp, and Claire Kenyon. Selection in the presence of noise: The design of playoff systems. In *Proceedings of the Fifth Annual ACM-SIAM Symposium on Discrete Algorithms. 23-25 January 1994, Arlington, Virginia*, pages 564–572, 1994.

Gagan Aggarwal, S Muthukrishnan, Dávid Pál, and Martin Pál. General auction mechanism for search advertising. In *Proceedings of the 18th international conference on World wide web*, pages 241–250. ACM, 2009.

Miklós Ajtai, Vitaly Feldman, Avinatan Hassidim, and Jelani Nelson. Sorting and selection with imprecise comparisons. *ACM Trans. Algorithms*, 12(2):19:1–19:19, November 2015. ISSN 1549-6325.

Michael Ben-Or and Avinatan Hassidim. The bayesian learner is optimal for noisy binary search (and pretty good for quantum as well). In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 221–230. IEEE, 2008.

Ralph Allan Bradley and Milton E Terry. Rank analysis of incomplete block designs the method of paired comparisons. *Biometrika*, 39(3-4):324–345, 1952.

Mark Braverman and Elchanan Mossel. Noisy sorting without resampling. In *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms. January 20-22, 2008, San Francisco, California, USA*, pages 268–276, 2008.

Mark Braverman, Jieming Mao, and S Matthew Weinberg. Parallel algorithms for select and partition with noisy comparisons. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 851–862. ACM, 2016.

Róbert Busa-Fekete, Eyke Hüllermeier, and Balázs Szörényi. Preference-based rank elicitation using statistical models: The case of mallows. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pages 1071–1079, 2014.

Xi Chen, Sivakanth Gopi, Jieming Mao, and Jon Schneider. Competitive analysis of the top-k ranking problem. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1245–1264. SIAM, 2017a.

Xi Chen, Yuanzhi Li, and Jieming Mao. An instance optimal algorithm for top-k ranking under the multinomial logit model. *arXiv preprint arXiv:1707.08238*, 2017b.

Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of gaussians. In *Proceedings of the 27th Annual Conference on Learning Theory (COLT)*, 2014.

Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A Servedio. Learning poisson binomial distributions. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 709–728, 2012.

Herbert Aron David. *The method of paired comparisons*, volume 12. Defence Technical Information Center Document, 1963.

Luc Devroye and Gabor Lugosi. *Combinatorial Methods in Density Estimation*. Springer - verlag, New York, 2001.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. *arXiv preprint arXiv:1604.06443*, 2016.

GÖSta Ekman. Weber's law and related functions. *The Journal of Psychology*, 47(2): 343–352, 1959.

Moein Falahatgar, Yi Hao, Alon Orlitsky, Venkatadheeraj Pichapati, and Vaishakh Ravindrakumar. Maxing and ranking with few assumptions. In *Advances in Neural Information Processing Systems*, pages 7060–7070, 2017a.

Moein Falahatgar, Alon Orlitsky, Venkatadheeraj Pichapati, and Ananda Theertha Suresh. Maximum selection and ranking under noisy comparisons. In *International Conference on Machine Learning*, pages 1088–1096, 2017b.

Moein Falahatgar, Ayush Jain, Alon Orlitsky, Venkatadheeraj Pichapati, and Vaishakh Ravindrakumar. The limits of maxing, ranking, and preference learning. In *International Conference on Machine Learning*, pages 1426–1435, 2018.

Uriel Feige, Prabhakar Raghavan, David Peleg, and Eli Upfal. Computing with noisy information. *SIAM Journal on Computing*, 23(5):1001–1018, 1994.

Pascal Hennequin. Combinatorial analysis of quicksort algorithm. *Informatique théorique et applications*, 23(3):317–333, 1989.

Richard M. Karp and Robert Kleinberg. Noisy binary search and its applications. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007, New Orleans, Louisiana, USA, January 7-9, 2007*, pages 881–890, 2007.

Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 273–282, 1994.

Donald Ervin Knuth. *The art of computer programming: sorting and searching*, volume 3. Pearson Education, 1998.

R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

Satyaki Mahalanabis and Daniel Stefankovic. Density estimation in linear time. In *Proceedings of the 21st Annual Conference on Learning Theory (COLT)*, pages 503–512, 2008.

Colin McDiarmid and Ryan Hayward. Strong concentration for quicksort. In *Proceedings of the Third Annual ACM/SIGACT-SIAM Symposium on Discrete Algorithms, 27-29 January 1992, Orlando, Florida*, pages 414–421, 1992.

Colin McDiarmid and Ryan B Hayward. Large deviations for quicksort. *journal of algorithms*, 21(3):476–507, 1996.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States*, pages 2483–2491, 2012.

Jelani Nelson. Personal communication. 2015.

Robin L Plackett. The analysis of permutations. *Applied Statistics*, pages 193–202, 1975.

Henry Scheffe. A useful convergence theorem for probability distributions. In *The Annals of Mathematical Statistics*, volume 18, pages 434–438, 1947.

Nihar Shah, Sivaraman Balakrishnan, Aditya Guntuboyina, and Martin Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pages 11–20, 2016.

Ananda Theertha Suresh, Alon Orlitsky, Jayadev Acharya, and Ashkan Jafarpour. Near-optimal-sample estimators for spherical gaussian mixtures. In *Advances in Neural Information Processing Systems*, pages 1395–1403, 2014.

Balázs Szörényi, Róbert Busa-Fekete, Adil Paul, and Eyke Hüllermeier. Online rank elicitation for plackett-luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems*, pages 604–612, 2015.

Louis L Thurstone. A law of comparative judgment. *Psychological review*, 34(4):273, 1927.

Leslie G. Valiant. A theory of the learnable. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 436–445, 1984.

Yisong Yue and Thorsten Joachims. Beat the mean bandit. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 241–248, 2011.

Journal of Machine Learning Research 19 (2018) 1-29 Submitted 3/17; Revised 8/18; Published 11/18

A New and Flexible Approach to the Analysis of Paired Comparison Data

Ivo F. D. Oliveira IVODAVID@GMAIL.COM
Department of Science, Engineering and Technology
UFVJM - Federal University of the Valleys of Jequitinhonha and Mucuri
Teofilo Otoni, Minas Gerais, Brazil

Nir Ailon NAILON@CS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technionlogy
Haifa, Israel

Ori Davidov DAVIDOV@STAT.HAIFA.AC.IL
Department of Statistics
University of Haifa
Haifa, Israel

Editor: Nicolas Vayatis

Abstract

We consider the situation where I items are ranked by paired comparisons. It is usually assumed that the probability that item i is preferred over item j is $p_{ij} = F(\mu_i - \mu_j)$ where F is a symmetric distribution function, which we refer to as the comparison function, and μ_i and μ_j are the merits or scores of the compared items. This modelling framework, which is ubiquitous in the paired comparison literature, strongly depends on the assumption that the comparison function F is known. In practice, however, this assumption is often unrealistic and may result in poor fit and erroneous inferences. This limitation has motivated us to relax the assumption that F is fully known and simultaneously estimate the merits of the objects and the underlying comparison function. Our formulation yields a flexible semi-definite programming problem that we use as a refinement step for estimating the paired comparison probability matrix. We provide a detailed sensitivity analysis and, as a result, we establish the consistency of the resulting estimators and provide bounds on the estimation and approximation errors. Some statistical properties of the resulting estimators as well as model selection criteria are investigated. Finally, using a large data-set of computer chess matches, we estimate the comparison function and find that the model used by the International Chess Federation does not seem to apply to computer chess.

Keywords: linear stochastic transitivity, statistical ranking, semi-definite programming, model selection, sensitivity analysis, chess

1. Introduction

There are many situations in which a preference or a ranking among a set of items is desired. Ranking methods are widely used in settings such as product testing (Cremonesi et al., 2010), the evaluation of political candidates (Saari, 1995; Pacuit, 2012), psychometrics (Regenwetter et al., 2011), machine learning (Ailon, 2012; Shah et al., 2015a) and sports (Herbrich et al., 2007; Govan, 2008). An ordering of a set of items can be inferred from different types of data including scores (Balinski and Laraki, 2010) and ranked lists (Marden, 1996). A ranked list may be complete, i.e., all items are compared and ranked, or partial or incomplete, when only a subset of items is compared and ranked. In particular, paired comparison data is obtained if all comparisons involve only two items (David, 1988). Here we focus on paired comparisons with a binary outcomes.

Given a set of I items, also called objects, or players, let Y_{ijk}, $1 \leq i, j \leq I$, be independent binary random variables where $Y_{ijk} = 1$ if item i is preferred over item j on their k^{th} comparison and $Y_{ijk} = 0$ otherwise. The probability of this event is denoted by p_{ij}, i.e., $p_{ij} = \mathbb{P}(Y_{ijk} = 1)$. We assume that m_{ij} comparisons were observed between each pair of items and we let $Y_{ij} = \sum_{k=1}^{m_{ij}} Y_{ijk}$ denote the number of times item i was preferred over item j. Further let $\boldsymbol{P} = [p_{ij}]$ denote the $I \times I$ underlying probability matrix.

Typically, it is assumed that

$$p_{ij} = F(\mu_i - \mu_j) \tag{1}$$

where $\mu_i, \mu_j \in \mathbb{R}$ are the merits (also called skills, scores or ratings) of items i and j respectively, and $F : \mathbb{R} \to [0, 1]$ is a known, strictly increasing, comparison function, i.e., a symmetric absolutely continuous distribution function. We assume that the merits are fixed and unknown. In some situations the merits may vary according to an "effort" (Jia et al., 2013), or depend on covariates (Herbrich et al., 2007; Allison and Christakis, 1994). Model (1) implies that $p_{ij} + p_{ji} = 1$ and imposes a form of stochastic transitivity (Morrison, 1963; Regenwetter et al., 2011) which is known as linear stochastic transitivity (LST). Models satisfying (1) will be referred to as LST models. Various LST models have been proposed, these differ in the choice of the comparison function F. In particular, two canonical LST models have been widely studied; the Thurstonian model (Thurstone, 1927) and the (Zarmelo) Bradley-Terry-Luce model (Zermelo, 1928; Bradley and Terry, 1952). The Bradley-Terry-Luce model (BTL, henceforth) assumes that F is a standard logistic distribution whereas Thurstone's model assumes F is a standard normal distribution. There are literally thousands of studies which employ these models and their variants. Other, albeit less popular, LST models are also studied in literature, e.g., the Threshold model which employs the Laplace distribution and is used for modelling animal behavior (Yellott, 1970), and the locally linear model (Batchelder et al., 1992) which employs a uniform on $[-1, 1]$ distribution.

The assumption that F is known has been recognized as rather unrealistic (Morrison, 1963; David, 1988; Regenwetter and Davis-Stober, 2008; Hwang, 2009; Shah et al., 2015b; Heckel et al., 2016). This has motivated several authors to resort to the use of less restrictive transitivity relations. A variety of stochastic transitivity relations have been explored in the literature (Morrison, 1963; Regenwetter et al., 2011; Oliveira et al., 2018), the weakest of which is known as weak stochastic transitivity. Under weak stochastic transitivity if $p_{ij} \geq 1/2$ and $p_{jk} \geq 1/2$ then $p_{ik} \geq 1/2$. A stronger form of stochastic transitivity, referred to as strong stochastic transitivity (SST), postulates that if $p_{ij} \geq 1/2$ and $p_{jk} \geq 1/2$ then $p_{ik} \geq \max\{p_{ij}, p_{jk}\}$. Of course model (1) satisfies both of the relations. Various authors have developed methods for analyzing data under these less restrictive assumptions (deCani, 1969; Regenwetter and Davis-Stober, 2008; Chatterjee and Mukherjee, 2016; Shah et al., 2015b). It turns out that the estimation procedures associated with these less restrictive

transitivity relations are in general, NP-hard. Since these models provide less structure, they may not be adequate when the comparison graph is sparse and most importantly may provide less predictive power. The SST model, for example, cannot guarantee that stronger players have higher chances than weaker players in knockout tournaments, whereas BTL can (Chung and Hwang, 1978; Israel, 1981; Adler et al., 2017; Baek et al., 2013). Thus we propose a different, potentially more powerful approach, within the LST framework, in which the assumption that the comparison function F is known is relaxed. The proposed methodology is flexible, tractable and retains the desirable computational and statistical characteristics of LST models.

A natural and widely used approach for estimating the model parameters in (1) is by least squares (LS). LS is often the method of choice due to its (relative) computational simplicity. Thus if (1) holds then the LS estimators solve the following optimization problem:

$$\hat{\boldsymbol{\mu}} \in \operatorname{argmin} \sum_{i \neq j} w_{ij} (\hat{\Delta}_{ij} - (\mu_i - \mu_j))^2, \tag{2}$$

where, typically, $\hat{\Delta}_{ij} \equiv F^{-1}(\hat{p}_{ij})$ and $\hat{p}_{ij}$ is an estimator of the probability p_{ij}. For now we assume that $\hat{p}_{ij}$ is bounded away from 0 and 1 and $\hat{p}_{ij} + \hat{p}_{ji} = 1$ so $\hat{\Delta}_{ij}$ is well defined. The weights w_{ij} are given and are either proportional to the variance of the estimated $\hat{\Delta}_{ij}$, or the number of comparisons between items i and j. Notice that (2) admits multiple solutions, for if $\boldsymbol{\mu}^*$ is a solution to problem (2) then so is $\boldsymbol{\mu}^* + v\mathbf{1}$ for any $v \in \mathbb{R}$. A unique solution exists if the comparison graph is connected and an additional linear constraint such as $\sum_i \mu_i = 0$ is enforced (Tsukida and Gupta, 2011).

When all w_{ij} in (2) are equal, then the solution is of the form $\mu_i^* = \kappa \sum_j \hat{\Delta}_{ij}$ for some $\kappa > 0$. For this reason the LS method is sometimes referred to as the row—sum procedure (Huber, 1963). There are other well known ranking methods which can be viewed as row—sum procedures with varying definitions of $\hat{\Delta}_{ij}$. For example, the Copland Method, popular within the voting literature (Levin and Nalebuff, 1995; Favardin et al., 2002), is a row sum procedure in which $\hat{p}_{ij}$ is defined as $\hat{p}_{ij} \equiv (\sum_k Y_{ijk})/I$ and $F(x) = 1/2 + x$ for $x \in [-1/2, 1/2]$ where Y_{ijk} equals one if the kth voter prefers candidate i above candidate j and zero otherwise. The Borda Count can be also shown to be a row—sum method. Another LS variant, known as Massey Ratings, is widely used in the rating of sport teams in college football, basketball, hockey, and baseball, see Chapter 4 of Massey (2017). For more on the LS literature, refer to Hodge rank in Jiang et al. (2010).

Our Contribution. We will weaken the assumption that the comparison function F is known and assume only that it belongs to, a new, large family of parametric functions. Our parametric set, can be understood as an interior approximation, with arbitrary precision, to the full set of comparison functions. We then simultaneously estimate the merits as well as the comparison function F by generalizing the LS approach. Estimating the probabilities p_{ij} is now an easy consequence. We show that this can be done efficiently both computationally and statistically. In particular, we develop a procedure that takes as input an estimate of the probability matrix and returns an estimate of the comparison function F, the merit vector $\boldsymbol{\mu}$ and a refinement of the original estimate of the probability matrix. Estimation reduces to a semi-definite programming problem with a tractable solution. We provide a thorough sensitivity analysis and derive statistical properties such as convergence and concentration bounds on the refined estimator and the estimated function. By applying our methodology to a large data-set of computer chess matches, we verify that the ubiquitous (Zarmelo) Bradley-Terry-Luce model may be inappropriate for computer chess.

2. Formulation and Estimation

Formulation: Least Squares Estimation Over Polynomial Families. First, we may generalize problem (2) by rewriting it in matrix notation in the following way

$$\hat{\boldsymbol{\mu}} \in \operatorname{argmin}_{\boldsymbol{\mu}} ||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}||_{\boldsymbol{W}}. \tag{3}$$

Here $\Delta\boldsymbol{\mu}$ is an $I \times I$ matrix whose ij^{th} element is $\mu_i - \mu_j$ and $F^{-1}(\hat{\boldsymbol{P}})$ is a matrix with the same dimensions whose ij^{th} element is $F^{-1}(\hat{p}_{ij})$, if $w_{ij} > 0$ and 0 otherwise. Unless specified otherwise, in this paper the norm $||\cdot||_{\boldsymbol{W}}$ will be the weighted Frobenius (semi-)norm, with pre-specified weights and $||\cdot||$ will be its unweighted counterpart. With a slight abuse of notation we will refer to the frobenius norm as the $\mathcal{L}_2$ norm. The minimization in (3) can also be formulated with respect to the sum of the absolute values of the elements, which we refer to as the $\mathcal{L}_1$ norm, or maximum value of the elements of a matrix, which we refer to as the $\mathcal{L}_\infty$ norm. The mechanics involved in solving (3) are norm dependent. If one views Δ as an operator from $\mathbb{R}^I \to \mathbb{R}^{I \times I}$ then $\Delta\hat{\boldsymbol{\mu}}$ is the projection of $F^{-1}(\hat{\boldsymbol{P}})$ on the image set of the operator Δ. Finally note that the least squares procedure takes an estimator $\hat{\boldsymbol{P}}$ and produces a refined estimator denoted by $\boldsymbol{P}^* = F(\Delta\hat{\boldsymbol{\mu}})$.

The assumption that the comparison function F is known is relaxed and instead it is assumed that $F \in \mathcal{F}$ where $\mathcal{F}$, where:

A$_1$ *(Parametric Assumption): The family $\mathcal{F}$ indexed by $\boldsymbol{\beta} \in \mathbb{R}^{D+1}$ consists of all distribution functions whose inverse, i.e., its quantile function, may be written as a polynomial, of the form*

$$F_{\boldsymbol{\beta}}^{-1}(p) = \beta_0 + \beta_1 p + ... + \beta_D p^D \text{ where } p \in [0, 1/2]. \tag{4}$$

Equation (4) defines a quantile regression model (Takeuchi et al., 2006; Su, 2015). By the LST condition $F_{\boldsymbol{\beta}}$ is symmetric, i.e., $F_{\boldsymbol{\beta}}(x) + F_{\boldsymbol{\beta}}(-x) = 1$ so $F_{\boldsymbol{\beta}}^{-1}(p) + F_{\boldsymbol{\beta}}^{-1}(1-p) = 0$. It immediately follows that $F_{\boldsymbol{\beta}}^{-1}(p)$ is also a polynomial when $p \in [1/2, 1]$. Also, (4) implies that the support of $F_{\boldsymbol{\beta}}$ is the finite interval $[\beta_0, -\beta_0]$, where $\beta_0 < 0$. It is further assumed that:

A$_2$ *(Lipschitz Assumption): For all $F_{\boldsymbol{\beta}} \in \mathcal{F}$, $F_{\boldsymbol{\beta}}$ is L-Lipschitz and $||\boldsymbol{\beta}||_\infty \leq U$* for some constants L and U.

We note that each fixed value of (D, U, L) generates a parametric family of distributions $\mathcal{F}(D, U, L)$; which we denote for convenience by $\mathcal{F}$. This is a new, non-standard, rich family of distributions, in which the quantile function, not the density, is parametrized. Figure 1 shows that the Bradley-Terry-Luce model can be can be approximated by a low degree polynomial over the interval $p \in [0.01, 0.99]$. Furthernote, that by increasing D, U and L we can approximate any quantile function with arbitrary precision.

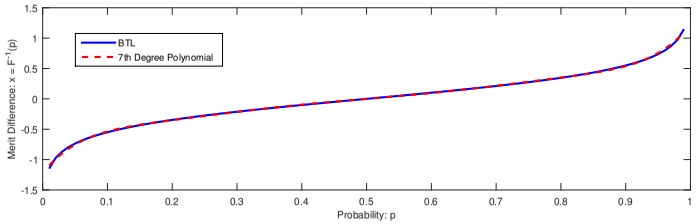


Figure 1: An approximation of the Bradley-Terry-Luce quantile function over the interval $p \in [0.01, 0.99]$ by a function $F \in \mathcal{F}$ for $D = 7$.

It is known that F and G are equivalent comparison functions iff $F(x) = G(\kappa x)$ for some positive κ, see Yellott (1977), and therefore F^{-1} is equivalent to G^{-1} iff $\kappa F^{-1}(p) = G^{-1}(p)$. A valid linear constraint on the coefficient vector $\boldsymbol{\beta}$ is thus imposed to ensure identifiability. For example, fixing the support of F, which amounts to fixing $\beta_0 < 0$, is sufficient. Another natural choice is to fix the derivative of F at 0, which amounts to fixing $\beta_1 > 0$. A rescaling argument shows that the resulting inferences do not depend on the chosen constraint. As noted earlier, identifiablity requires that the merits satisfy a constraint. Henceforth it will be assumed that:

A_3 *(Scaling Assumption): The merits and the comparison function are scaled to satisfy*

$$\sum_i \mu_i = 0 \quad \textit{and} \quad \beta_0 = -1. \tag{5}$$

Thus, if F belongs to $\mathcal{F}$ we may estimate $(\boldsymbol{\mu}, \boldsymbol{\beta})$ by solving the following optimization problem:

$$(\hat{\boldsymbol{\mu}}, \hat{\boldsymbol{\beta}}) \in \operatorname{argmin}_{\boldsymbol{\mu} \in \mathbb{R}, F_{\boldsymbol{\beta}} \in \mathcal{F}} ||F_{\boldsymbol{\beta}}^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}||_{\boldsymbol{W}}. \tag{6}$$

Although (6) is a least squares problem it is non-standard as "both sides", i.e., the "predictor" and the "response" in the regression equation, are associated with unknown parameters which are estimated simultaneously. In the following subsection we will study problem (6) under assumptions A_1 to A_3.

Solution via Semidefinite Programming. Our first concern is to characterize the set of feasible solutions for $(\boldsymbol{\mu}, \boldsymbol{\beta})$. As noted earlier the quantile function (4) satisfies $F^{-1}(p) = -F^{-1}(1-p)$ so we need only consider constraints generated by values $p \in [0, 1/2]$. In this interval $F_{\boldsymbol{\beta}}^{-1}(p)$ is increasing hence its derivative, which is a polynomial of degree $D-1$, is non-negative. Furthermore, the Lipschitz continuity constraint on F implies that F^{-1} is strictly monotone with derivative greater or equal to $1/L$. Thus,

$$\beta_1 + ... + D\beta_D p^{(D-1)} - \frac{1}{L} \geq 0 \text{ for all } p \in [0, 1/2] \tag{7}$$

In addition $F(0) = 1/2$, so $F_{\boldsymbol{\beta}}^{-1}(1/2) = 0$ and $||\boldsymbol{\beta}||_\infty \leq U$ therefore

$$\sum_{i=0}^{D} \left(\frac{1}{2}\right)^i \beta_i = 0 \text{ and } -U \leq \beta_i \leq U \text{ for all } i. \tag{8}$$

Minimizing $||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}||_{\boldsymbol{W}}$ subject to (7) and (8) yields a semi-infinite programming problem (SIP) (Mutapcic and Boyd, 2009; Stein, 2012), i.e., an optimization problem with an infinite number of, in this case linear, constraints.

There are a number of methods for solving SIPs. One natural approach is discretization, which in our case means replacing (7) by N constraints of the form $\beta_1 + \beta_2 p_j ... + D\beta_D p_j^{(D-1)} - 1/L \geq 0$ where $0 < p_1 < p_2 < ... < p_N < 1/2$ for some finite N. This yields a simple quadratic programming problem. From a practitioner's point of view, discretization may be a method of choice due to its simplicity. This is specifically true when $||\cdot||_{\boldsymbol{W}}$ is either the weighted or unweighted $\mathcal{L}_1$ or $\mathcal{L}_\infty$ norms, since in these cases discretization results in a simple linear program. However, the resulting estimate of F is not guaranteed to be strictly monotone (although this can be overcome, see section 3.2 of Mutapcic and Boyd 2009) and more importantly in the worst case the solution may not be polynomially computable. These issues may be overcome by noting that the constraints in (7)-(8) are equivalent to a combination of linear constraints and positive semi-definite cone constraints, see Parrilo (2016) for further details. Thus our optimization problem is (also) a semi-definite optimization problem (SDP). There is a large literature on SDPs (Nemirovski and Todd, 2009) and in particular it is known that SDPs can be solved by interior point methods (Vandenberghe and Boyd, 1996) in polynomial time. Therefore we may formally rewrite (6) as:

Theorem 1 *Given $U, L \geq 0$, $D = 2d+1$ with $d \in \mathbb{N}$, a symmetric weight matrix $\boldsymbol{W}$ and an estimator $\hat{\boldsymbol{P}}$, then problem (6) is equivalent to:*

$$\begin{array}{ll} \textit{minimize} & \sum_{(i,j)\in\mathcal{S}} w_{ij}(\beta_0 + ... + \beta_D \hat{p}_{ij}^D + \mu_j - \mu_i)^2 \\ \textit{subject to} & \\ & \beta_i = \frac{1}{i}(\frac{1}{2}t_{i-2} - t_{i-3} + s_{i-1}) \textit{ for } i = 2, ..., D, \\ & \beta_1 = s_0 + \frac{1}{L}, \quad \sum_{k=0}^{D} \left(\frac{1}{2}\right)^k \beta_k = 0, \quad ||\boldsymbol{\beta}||_\infty \leq U, \\ & \\ & s_i = \sum_{j+k=i} Q_{jk}^0 \textit{ for } i = 0, ..., D-1, \quad \boldsymbol{Q^0} \in \mathbb{S}_+^{d+1}, \\ & t_i = \sum_{j+k=i} Q_{jk}^1 \textit{ for } i = 0, ..., D-3, \quad \boldsymbol{Q^1} \in \mathbb{S}_+^{d}. \end{array} \tag{9}$$

where $\mathcal{S} = \{(i,j) \mid p_{ij} \leq .5 \textit{ or } p_{ij} = .5 \textit{ and } i < j\}$, and $t_{D-1} = t_{D-2} = t_{-1} = t_{-2} = 0$, $\mathbb{S}_+^k$ is the set of $k \times k$ symmetric positive semi-definite matrices, and the rows and columns of $\boldsymbol{Q^0}$ and $\boldsymbol{Q^1}$ are indexed by 0 to d and 0 to $d-1$ respectively.

For brevity we present here only the case when D is odd, a similar characterization holds for D even. We note that analogues of Theorem 1 could also be formulated for the $\mathcal{L}_1$ and $\mathcal{L}_\infty$ norms and their weighted versions in which case the constraints in (9) would remain unaltered whereas the objective function would be as defined by the corresponding norm.

Note that (9) admits a unique solution when the objective function is positive definite. Lemma 2 below provides an example in a simple but important case. In large samples the solution to (9) is uniquely determined when the system of equations $\beta_1 p_{ij} + ... + \beta_D p_{ij}^D - (\mu_i - \mu_j) = -\beta_0$ for $(i,j) \in \mathcal{S}$ is of full rank. This condition is met when (i) there are at least $I + D - 1$ connected pairs $(i,j) \in \mathcal{S}$; which (ii) the coefficients appearing in the linear equations, which are derived from the comparison probabilities p_{ij}, are sufficiently diverse, otherwise the resulting equations would not be linearly independent. Thus we assume that:

A$_4$ *(Connectivity & Diversity Assumption): The comparison graph has at least $I + D - 1$ edges. These edges correspond to a set of linearly independent equations of the form $\beta_1 p_{ij} + ... + \beta_D p_{ij}^D - (\mu_i - \mu_j) = -\beta_0$.*

If we label these equations $(ij)_1, ..., (ij)_{D+I-1}$ then together with the constraint $\sum \mu_i = 0$ we may write the resulting system of equations (with a slight abuse of notation) as:

$$\boldsymbol{A} \begin{pmatrix} \boldsymbol{\beta} \\ \boldsymbol{\mu} \end{pmatrix} = \begin{pmatrix} -\boldsymbol{\beta_0} \\ 0 \end{pmatrix} \tag{10}$$

where the k'th row of $\boldsymbol{A}$ is $\boldsymbol{A}_k = (p_{(ij)_k} \quad ... \quad p_{(ij)_k}^D \quad -\boldsymbol{e}_{(ij)_k})$ for $k = 1, ..., I + D - 1$ and $\boldsymbol{A}_{I+D} = (\boldsymbol{0}^T \quad ... \quad \boldsymbol{1}^T)$ where $\boldsymbol{e}_{ij} \in \mathbb{R}^I$ is defined by $\boldsymbol{e}_{ij} \equiv \boldsymbol{e}_i - \boldsymbol{e}_j$, where $\boldsymbol{e}_i$ is the standard basis. The condition number of $\boldsymbol{A}$ plays a role in the quality of our estimators.

Define $\boldsymbol{p} = (1,\ p,\ ...\ ,\ p^D)^T$ and note that if all the weights are equal then

$$\mu_i = \frac{1}{I} \sum_k F_{\boldsymbol{\beta}}^{-1}(p_{ik}) = \frac{1}{I} \sum_{(i,k) \in \mathcal{S}} F_{\boldsymbol{\beta}}^{-1}(p_{ik}) - \frac{1}{I} \sum_{(i,k) \notin \mathcal{S}} F_{\boldsymbol{\beta}}^{-1}(p_{ki}) \tag{11}$$

and thus we may eliminate $\boldsymbol{\mu}$ from (9) by means of equation (11). This considerably reduces the size of the SDP at hand when the number of items I is larger than D. Algorithm PolyRank (displayed below) takes advantage of this. Furthermore:

Lemma 2 *When all weights w_{ij} are equal, then, the objective function of problem (9) is equivalent to minimizing $\boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta}$, where*

$$\boldsymbol{M} \equiv \sum_{(ij) \in \mathcal{S}} \boldsymbol{v}_{(ij)} \boldsymbol{v}_{(ij)}^T \tag{12}$$

and $\boldsymbol{v}_{(ij)} \equiv -I \hat{\boldsymbol{p}}_{ij} + \sum_{(ik) \in \mathcal{S}} \hat{\boldsymbol{p}}_{ik} - \sum_{(ik) \notin \mathcal{S}} \hat{\boldsymbol{p}}_{ki} - \sum_{(jk) \in \mathcal{S}} \hat{\boldsymbol{p}}_{jk} + \sum_{(jk) \notin \mathcal{S}} \hat{\boldsymbol{p}}_{kj}$.

Hence the estimators can be efficiently calculated in three steps:

***Algorithm:* PolyRank**

Input: $\hat{\boldsymbol{P}} \in [0,1]^{I \times I}$ $D \in \mathbb{N}$ *an odd number and* $U, L \geq 0$.

1. *Preprocessing: Calculate* $\boldsymbol{M}$ *as in equation (12);*
2. *Functional Estimation:* $\hat{\boldsymbol{\beta}} \equiv$ *argmin* $\{\boldsymbol{\beta}^T \boldsymbol{M} \boldsymbol{\beta}$ *subject to* (9) *and* (5)$\}$;
3. *Merit Estimation:* $\hat{\mu}_i = \frac{1}{I} \sum_{(i,k) \in \mathcal{S}} F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{p}_{ik}) - \frac{1}{I} \sum_{(i,k) \notin \mathcal{S}} F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{p}_{ki})$;

Output: $\hat{\boldsymbol{\beta}} \in \mathbb{R}^{D+1}$, $\hat{\boldsymbol{\mu}} \in \mathbb{R}^n$ *and* $\boldsymbol{P}^* \equiv F_{\hat{\boldsymbol{\beta}}}(\Delta \hat{\boldsymbol{\mu}})$

Step 1 may be performed with no more than $O(I^3 D + I^2 D^2)$ operations, Step 2 with no more than $O(D^2 \sqrt{D})$ operations and Step 3 with no more than $O(I^2 D)$ operations. Thus, the overall computational complexity of solving problem (6) is no more than $O(I^3 D + I^2 D^2 + D^2 \sqrt{D})$. Notice also that Steps 1 and 3 can be done in a distributed fashion. When the weights are not all equal the merits cannot be written as in (11) and therefore Algorithm PolyRank as stated above cannot be used, in that case we solve (9) directly. Nevertheless we will refer to all versions of our estimation procedure as PolyRank. In our experience, problem (6) with any norm (weighted or unweighted) can be tackled successfully with a generic convex optimization solver on a desktop computer for problems of moderate size (e.g. with $D \leq 10$ and $I \leq 120$) in at most 2 or 3 seconds. Using the three step procedure (with the same generic solver) allows easy scaling up to problems where $D \leq 20$ and $I \leq 10000$. If (6) is treated as a SIP and solved via discretization, then significant reduction in computation time is observed at the cost of loosening the guarantee of optimality.

Remark 1 Notice that if the machine precision is ϵ and if D is such that $\epsilon > (1/2)^D$ then the last terms of the polynomial F^{-1} are rounded to zero. Therefore for standard 32 bit floating point arithmetic one should choose D at most 22, similarly for a 64 bit arithmetic D should not surpass 44. $O(V^2 \sqrt{V})$

Remark 2 In theory F can be recovered from F^{-1} exaclty via Lagrange Inversion Theorem. Numerically though, calculating $F_{\boldsymbol{\beta}}(\mu_i - \mu_j)$ reduces to a polynomial root-finding problem. Although root-finding is an ill-conditioned problem for general polynomials (Trefethen 2011), it may be solved via binary search (with linear convergence in the worst case) or via Newton steps (with possible quadratic convergence).

Remark 3 When the weights w_{ij} are not all equal, or $||\cdot||_{\boldsymbol{W}}$ represents the $\mathcal{L}_1$ or $\mathcal{L}_\infty$ norms, then a full SDP with $V = I + D$ variables must be solved. In these cases the simplifying row-sum structure is absent and the worst case bounds are well known, and of the order $O(V^2 \sqrt{V})$, see the general SDP literature (Vandenberghe and Boyd, 1996).

3. Sensitivity Analysis

The goal of this section is to investigate the sensitivity of PolyRank with respect to the input matrix $\hat{\boldsymbol{P}}$. There are several reasons for developing thorough, non-stochastic, sensitivity bounds. Firstly, the analysis serves to clarify the mechanics of PolyRank providing bounds that apply to any choice of $\hat{\boldsymbol{P}}$. Secondly, using the sensitivity bounds statistical properties such as consistency of the refined estimator are easily derived. A third motivation is that different estimators $\hat{\boldsymbol{P}}$ have been investigated in the literature, e.g., Rajkumar and Agarwal (2014); Chatterjee and Mukherjee (2016); Shah et al. (2015b), and since PolyRank may be applied to any of them, the respective bounds on the refined estimator are universal and apply to any $\hat{\boldsymbol{P}}$. Sensitivity analysis is carried out under three increasingly general settings. First, we provide a benchmark by studying the LS method with known F. Then, we consider PolyRank in the case where the model is correctly specified, i.e., $F \in \mathcal{F}$. This is also called the realizable case. Finally, we consider agnostic cases, that is, situations where the model may be misspecified in some way. Three examples of misspecifications are analyzed.

For simplicity we first focus on the unweighted $\mathcal{L}_2$ norm, extensions to the respective weighted versions are similarly obtained.

3.1. Known Comparison Function

Here the function F is assumed to be a known L-Lipschitz continuous function with a U-Lipschitz inverse, i.e., it is bilipschitz. A common assumption in the literature, cf. Shah et al. (2015a,b), is that the probabilities in (1) are bounded away from 0 and 1, i.e., $p_{ij} \in [\epsilon, 1-\epsilon]$, for some $\epsilon > 0$. Over this domain the Bradley-Terry-Luce, Thurstone, Threshold and Locally Linear models are all bilipschitz.

Theorem 3 *Let F be a known L-Lipschitz continuous function with a $4U$-Lipschitz continuous inverse (over their respective domains). Let $\hat{\boldsymbol{\mu}}$ be as in (2) and $\boldsymbol{P^*} = F(\Delta\hat{\boldsymbol{\mu}})$. Then,*

$$||\boldsymbol{P^*} - \boldsymbol{P}|| \leq 4LU||\hat{\boldsymbol{P}} - \boldsymbol{P}||, \tag{13}$$

and

$$||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}|| \leq \frac{4U}{\sqrt{2I}}||\hat{\boldsymbol{P}} - \boldsymbol{P}||. \tag{14}$$

If, additionally, it is assumed that $\hat{\boldsymbol{P}}$ obeys strong stochastic transitivity, then the estimators are order preserving, i.e.,

$$\hat{\mu}_i < \hat{\mu}_j \iff \hat{p}_{ij} < \hat{p}_{ji}. \tag{15}$$

By construction the constant $4LU \geq 1$ and so (13) guarantees that $||\boldsymbol{P^*} - \boldsymbol{P}||$ will be at most a constant times $||\hat{\boldsymbol{P}} - \boldsymbol{P}||$. Although it may be possible to improve the constant in (13), its value can never be less than 1, for if not, one could generate a converging sequence $\boldsymbol{P_1}^*, \boldsymbol{P_2}^*, ...$ by recursively applying the LS refinement to any initial (blind) guess of $\hat{\boldsymbol{P}}$. This argument holds for any refinement procedure, including PolyRank. Also, by construction the LS refinement defines $\boldsymbol{P}^* = F(\Delta\hat{\boldsymbol{\mu}})$ and thus $\min_{\boldsymbol{\mu}} ||F^{-1}(\boldsymbol{P}^*) - \Delta\boldsymbol{\mu}|| = 0$ for $\boldsymbol{\mu} = \hat{\boldsymbol{\mu}}$ and so no improvement will be obtained via recursive LS type refinements. The bound in (13) is a "worst case" bound and on average we often observe that $||\boldsymbol{P^*} - \boldsymbol{P}||$ is indeed smaller that $||\hat{\boldsymbol{P}} - \boldsymbol{P}||$. For other norms refer to the appendix.

The benchmarks provided by Theorem 3 will be used for comparison with the more general cases tackled by PolyRank. As will be shown, inequality (13) also holds when F is unknown (with different constant factors); similarly, the order preservation is maintained in all the settings considered.

3.2. Realizable Case

Under the hypothesis of realizability, i.e., when the model is correctly specified, we have:

Theorem 4 *Let $\boldsymbol{P^*} = F_{\hat{\boldsymbol{\beta}}}(\Delta\hat{\boldsymbol{\mu}})$ where $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\mu}}$ are estimated using PolyRank. Then,*

$$||\boldsymbol{P^*} - \boldsymbol{P}|| \leq (1 + 4LU)||\hat{\boldsymbol{P}} - \boldsymbol{P}||, \tag{16}$$

and

$$\left|\left|\begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} \end{pmatrix}\right|\right| \leq K_1||\hat{\boldsymbol{P}} - \boldsymbol{P}||, \tag{17}$$

as well as,

$$\max_{x\in[-1,1]} |F_{\hat{\boldsymbol{\beta}}}(x) - F_{\boldsymbol{\beta}}(x)| \leq K_2||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\infty, \tag{18}$$

where $K_1 \leq U(1+4LU)\sqrt{D(I+D)}||\boldsymbol{A}^{-1}||$ and $K_2 \leq 16LDU^2 \max_i \sum_j |A^{-1}_{ij}|$. If, additionally, it is assumed that $\hat{\boldsymbol{P}}$ obeys strong stochastic transitivity, then the estimators are order preserving.

Notice that the constants in (13) and (16) depend solely on the product of the Lipschitz constants of F and F^{-1}. Moreover the constant in (16) does not depend on D nor on the condition number of A . In contrast, the constants in inequalities (14) and (17) do depend on the dimensions of the problem. Equation (18) guarantees the convergence of $F_{\hat{\boldsymbol{\beta}}}$ to the true function F with respect to the Chebyshev distance, thus, one can eventually recover F with arbitrary precision.

3.3. Agnostic Cases

We will now investigate the properties of PolyRank under several types of misspecification. First, we investigate the effect of misspecifying the degree of the polynomial (4). Then, we provide results analogous to those provided by Theorem 4 by replacing the assumption that $F \in \mathcal{F}$ by the assumption that the true F is an analytic function. Finally, we drop the assumption that $\boldsymbol{P}$ satisfies the LST hypothesis all together and verify that we can still derive, albeit, weaker sensitivity bounds and rank consistency properties if strong stochastic transitivity is assumed.

Theorem 5 *Assume the true model satisfies (4), however the fitted model was of degree $D' \leq D-1$. Then for any $\boldsymbol{\beta'}$ of dimension $D' \leq D-1$, a lower bound on the approximation error, in the Chebyshev norm, is:*

$$\frac{1}{2U}\frac{|\beta_D|}{8^D} \leq \max_\alpha |F_{\boldsymbol{\beta'}}(\alpha) - F_{\boldsymbol{\beta}}(\alpha)| \tag{19}$$

The lower bound (19) shows that the Chebyshev distance between the true function and the estimated function cannot be arbitrarily minimized when the degree of the fitted polynomial is under-specified. The lower-bound, though, decreases with the value of D at an exponential rate.

Theorem 6 *Let $\boldsymbol{P}^* = F_{\hat{\boldsymbol{\beta}}}(\Delta\hat{\boldsymbol{\mu}})$ where $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\mu}}$ are estimated with* PolyRank. *Assume that the true probability matrix $\boldsymbol{P} = F(\Delta\boldsymbol{\mu})$ for some $\boldsymbol{\mu}$ and some unknown L-Lipschitz continuous function F with an analytic inverse function F^{-1} whose coefficients are upper-bounded by U. Then for the estimated probability matrix we have:*

$$||\boldsymbol{P}^* - \boldsymbol{P}|| \leq (1 + 4LU)||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + \frac{1}{2^D} LUI \tag{20}$$

If additionally it is assumed that $\hat{\boldsymbol{P}}$ obeys strong stochastic transitivity, then the estimators are order preserving.

Equation (20) shows that the error of $\boldsymbol{P}^*$ can be controlled under a broad class of analytic functions. The first term is controlled by increasing the precision of $\hat{\boldsymbol{P}}$ and the second term is controlled by choosing larger values for D.

In the following Theorem we will assume no particular structure on $\boldsymbol{P}$, i.e. the probability matrix $\boldsymbol{P}$ need not be consistent with any stochastic transitivity model.

Theorem 7 *Let $\boldsymbol{P}^* = F_{\hat{\boldsymbol{\beta}}}(\Delta\hat{\boldsymbol{\mu}})$ where $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\mu}}$ are estimated with* PolyRank. *Then for the estimated probability matrix we have:*

$$||\boldsymbol{P}^* - \boldsymbol{P}|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + L||F^{-1}_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}||, \tag{21}$$

If additionally it is assumed that $\hat{\boldsymbol{P}}$ obeys strong stochastic transitivity, then the estimators are order preserving.

An immediate consequence of order preservation is that if $\boldsymbol{P}$ is in the interior of the strong stochastic transitivity set then PolyRank is order-consistent for any consistent estimator $\hat{\boldsymbol{P}}$, i.e., when $\hat{\boldsymbol{P}} \to \boldsymbol{P}$ then the vector $\hat{\boldsymbol{\mu}}$ will correctly recover the underlying order among the items. The error bound in equation (21), though, cannot be controlled as in equation (20), this is so because the second term can be as big as κI^2 for some positive κ even when $\hat{\boldsymbol{P}}$ satisfies strong stochastic transitivity (Shah et al., 2015b).

4. Convergence and Concentration

In this subsection we assume that the model is correctly specified and investigate some properties of the estimators obtained by PolyRank. We start with the case where the comparisons graph is fixed and the number of comparisons per pair, i.e., the m_{ij}'s is allowed to increase. Similar conditions have been considered in literature (Rajkumar and Agarwal, 2014; Shah et al., 2015a).

It is well known that the topology of the comparison graph plays an important role in the quality of the estimators (Shah et al., 2015a; Massey, 2017; Colley, 2002). In particular Shah et al. (2015a) show that, the mean squared errors of the estimated merits from a

standard Bradley-Terry-Luce model are proportional to the second eigenvalue of the graph Laplacian. This eigenvalue, referred to as the algebraic connectivity of the graph, measures how "well" the graph is connected (Chung, 1994). In our concentration bounds the number of edges in the comparison graph and the condition number associated with (10) will play a similar role. Let $n = \sum_{i,j} m_{ij}$ be the number of paired comparisons.

Theorem 8 *Let $\hat{\boldsymbol{\beta}}_n$ and $\hat{\boldsymbol{\mu}}_n$ be estimated using* PolyRank *with $m_{ij} \equiv w_{ij}n$. Let $\hat{p}_{ij}$ be the usual MLEs. Then for large enough n there are constants K_1 and K_2 such that,*

$$\mathbb{P}\left(\left|\left|\begin{pmatrix} \hat{\boldsymbol{\beta}}_n - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}}_n - \boldsymbol{\mu} \end{pmatrix}\right|\right| \geq \epsilon\right) \leq K_1 \exp\left(-nK_2\epsilon^2\right), \tag{22}$$

where K_1 and K_2 are discussed bellow.

Theorem 8 shows that the estimators $\hat{\boldsymbol{\beta}}_n$ and $\hat{\boldsymbol{\mu}}_n$ converge at an exponential rate and are therefore strongly consistent. Theorem 8 also implies an exponential convergence of $\boldsymbol{P}^*$ to $\boldsymbol{P}$ as well as of $F_{\hat{\boldsymbol{\beta}}}(x)$ to $F_{\boldsymbol{\beta}}(x)$ in the Chebyschev distance. Theorem 8 is proved by first establishing sensitivity bounds for the weighted norm. These are analogues of Theorem 4 and are of the form

$$\left|\left|\begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} \end{pmatrix}\right|\right| \leq K||\hat{\boldsymbol{P}} - \boldsymbol{P}||_{\boldsymbol{W}}. \tag{23}$$

The constants in (22) are $K_1 = 2|E|$ where $|E|$ is the number of edges in the comparison graph, and $K_2 = 2/(|E|(1+4LU)^2U^2D(I+D)||\boldsymbol{A}_{\boldsymbol{W}}^{-1}||^2)$, where $\boldsymbol{A}_{\boldsymbol{W}}^{-1}$ is defined as in equation (10) with the appropriate modifications for weights. Clearly, $I + D - 1 \leq |E| \leq (I^2 - I)/2$. Of course, smaller values of $|E|$ will provide tighter bounds in equation (22). The value of $||\boldsymbol{A}_{\boldsymbol{W}}^{-1}||$ is a function of, among other things, the topology of the comparison graph. Unfortunately, the condition number of $A_{\boldsymbol{W}}$ is difficult to analyze as it contains a $(I+D-1)\times D$ Vandermonde submatrix which can range from 1 (the best possible condition number) to exponential on the dimensions of the matrix (Pan, 2015). As a rule of thumb Vandermonde matrices are well conditioned when the points $p_{(ij)_1}, ..., p_{(ij)_{D+I-1}}$ are (approximately) spaced over Chebyshev points. Matrix $\boldsymbol{A}_{\boldsymbol{W}}$ also contains a standard $(I + D - 1) \times I$ submatrix of pairings and thus we conjecture that smaller values of the second eigenvalue of the graph Laplacian matrix should also provide tighter estimation bounds.

4.1. Round robin

We now turn our attention to round robin tournaments (Chatterjee and Mukherjee, 2016; Shah et al., 2015b; Simons and Yao, 1999), in which each pair of items is compared m times. If the number of items I is fixed and if $m \to \infty$ then we can use the results described above. A more interesting situations arises when $m = 1$ but the number of items $I \to \infty$. As pointed out earlier, in this setting if $\hat{p}_{ij} \propto Y_{ij}$ then the LS estimator $\hat{\mu}_i$ will be proportional to its Copeland Score (the number of times an item was preferred). Recent papers addressing this setting are by Chatterjee and Mukherjee (2016) and Shah et al. (2015b). In particular they assume strong stochastic transitivity and construct an estimator $\hat{\boldsymbol{P}}_{\text{ISO}}$ which is shown to satisfy:

$$\sup \frac{1}{I^2}\mathbb{E}||\hat{\boldsymbol{P}}_{\text{ISO}} - \boldsymbol{P}||_2^2 \leq C\sqrt{\frac{\log I}{I}}, \tag{24}$$

where the supremum is taken over all matrices that satisfy strong stochastic transitivity. They also show that if the true model was LST then, under some regularity conditions, the upper bound in (24) can be tightened to $O(1/I)$ up to log factors (Shah et al., 2015b). Their estimator is calculated in two steps: (i) first, they sort the items according to their Copeland Score; (ii) then, they perform a two dimensional isotonic regression on the matrix $\boldsymbol{Y}$ assuming the order obtained in (i).

The resulting estimator has two drawbacks when the true model is LST. First, the estimator may be infeasible, i.e., $\hat{\boldsymbol{P}}_{\text{ISO}}$ may not be LST. Our experience indicates that this is frequently the case. In addition the resulting estimator does not fully exploit the benefits of an LST model since the estimated probability matrix is not a Functional of a merit vector and the comparison function. These deficiencies, however, can be addressed by applying PolyRank to their estimator. A trivial consequence of equation (16) is that the refined estimator $\boldsymbol{P}^*$ retains the optimal risk bounds of $\hat{\boldsymbol{P}}_{\text{ISO}}$ and by construction is feasible. We state the full result for completeness:

Theorem 9 *Let $\boldsymbol{P}^*$ be the refinement of $\hat{\boldsymbol{P}}_{ISO}$ using* PolyRank*, where $\hat{\boldsymbol{P}}_{ISO}$ is the estimator of Chatterjee and Mukherjee (2016), then:*

$$\sup \frac{1}{I^2}\mathbb{E}||\boldsymbol{P}^* - \boldsymbol{P}||_2^2 \leq K\frac{\log^2 I}{I}, \tag{25}$$

for some constant K that does not depend on neither I nor D (nor the condition number of A) and the supremum is taken over the set of probability matrices consistent with functions $F \in \mathcal{F}$. This is optimal up to log factors.

5. Numerical Experiments and An Illustrative Example

In the following we describe four experiments performed to further test and investigate PolyRank. Each simulation is performed 1000 times and we report and discuss the average performance under the specified conditions.

Experiment 1: In this experiment we compare the empirical performance of the estimator of $\boldsymbol{P}$ when using PolyRank with a low degree polynomial with its performance given the correct comparison function. Specifically, this is done by generating $I = 20$ items with merits μ_i sampled uniformly from $[0, 10]$. A total of 50 pairs, selected randomly, were compared assuming a Bradley-Terry-Luce (BTL) model. We refine the estimator $\hat{p}_{ij} = (Y_{ij}+1)/(m_{ij}+2)$ with PolyRank using $D = 5$. We also compute the LS estimated with the known F. Figure 2 shows the average distance $||\boldsymbol{P}^* - \boldsymbol{P}||_2$. As expected, the LS method with the correct comparison function performs best, PolyRank performs almost as well and both substantially outperform the initial estimates. This is consistent with our expectations because the BTL model, despite not belonging to the class of functions $\mathcal{F}$, can be well approximated by this class within the range of choice probabilities generated.

Experiment 2 In this experiment we investigate the empirical performance of PolyRank in the round-robin setting when the number of items is increasing. Specifically, we generate a sequence of round-robin tournaments with an increasing number of items. The data is generated assuming model (4) with $D = 5$. The matrix $\boldsymbol{P}$ is estimated using the isotonic

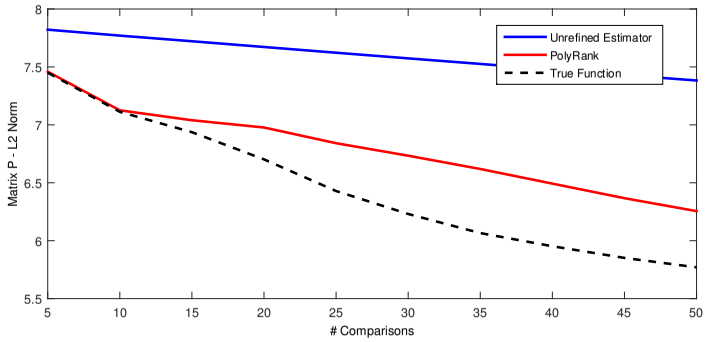


Figure 2: Comparison of refined estimators with low sampling.

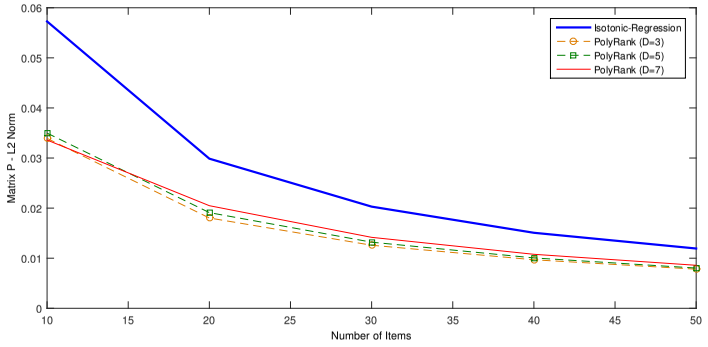


Figure 3: Refined estimators for round-robin tournaments.

regression estimator of Chatterjee and Mukherjee (2016) and refined using PolyRank with $D \in \{3, 5, 7\}$. Figures 3 and 4 display, respectively, the average of $||\boldsymbol{P}^* - \boldsymbol{P}||^2/I^2$ and the average of $||(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}, \hat{\boldsymbol{\mu}} - \boldsymbol{\mu})||^2/(I + D)$ for $I = 10, 20, 30, 40$ and 50.

Figure 3 shows four curves, all of which decrease with I. The top curve is the risk for the unrefined isotonic-regression based estimator. The estimators refined by PolyRank, which correspond to the lower 4 curves always do better. Notice that over-fitting, i.e., $D = 7$, which corresponds to the second curve from the top, usually results in higher estimation error with no change in the approximation error when compared to $D = 3, 5$. The second curve from the bottom corresponds to the true model. The bottom curve is obtained when $D = 3$, i.e., under under-fitting, and results in the lowest risk. Although this result is somewhat surprising it has been documented also in the context of other models (Claeskens

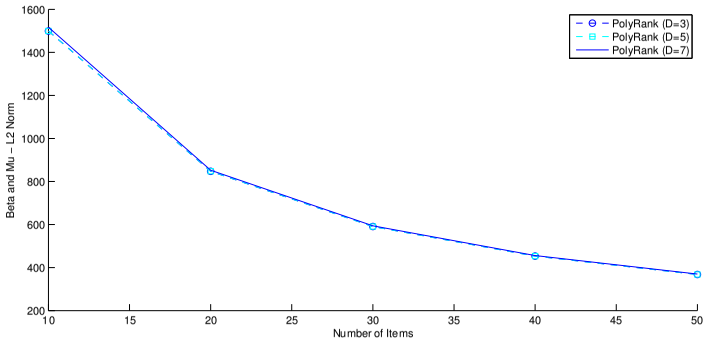


Figure 4: Estimated parameters for round-robin tournaments.

and Hjort, 2006, Chapter 5). This indicates that lower degree polynomial often perform well in practice. In Figure 4 we see that the average error of the estimated parameters decreases as a function of I.

Experiment 3: In this experiment we investigate the performance of PolyRank in the round-robin setting with a fixed number of items and a increasing number of comparisons. we generated a sequence of round-robin tournaments with $I = 10$ and an increasing number of matches. The data is generated assuming model (4) with $D = 3$. The matrix $\boldsymbol{P}$ is estimated using the standard frequency estimator for $\hat{p}_{ij}$ and is refined using PolyRank (with $D = 3$). Figure 5 displays the average of $||\boldsymbol{P}^* - \boldsymbol{P}||^2/I^2$, of $||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||^2/I$ and of $||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}||^2$ for $m_{ij} = 1$ to 5 for all pairs (i, j) and Figure 6 shows the sequence of estimated functions.

The three decreasing curves of Figure 5 show that the variance of the estimators decreases with the amount of paired comparisons. Figure 6 shows that the estimated comparison function converges to its true value. These results and those of Experiment 2 are consistent with Theorems 8 and 9.

Experiment 4: In practice the degree D of the polynomial (4) may not be known in advance. If we choose D to be too small then we may not fully capture the geometry of F, while if D is too large there is a danger of over-fitting and possible numerical problems. In this experiment we investigate the use of some well known model selection criteria (Claeskens and Hjort, 2006) for choosing D. In particular, we test the empirical performance of the Bayesian Information Criterion (BIC) and two variants of the Akaike Information Criterion (AIC) and contrast these with the performance of (leave-one-out) cross-validation. The classical AIC criteria is

$$\mathcal{AIC}(D) \equiv 2(I + D) + n \log(l(D)) \tag{26}$$

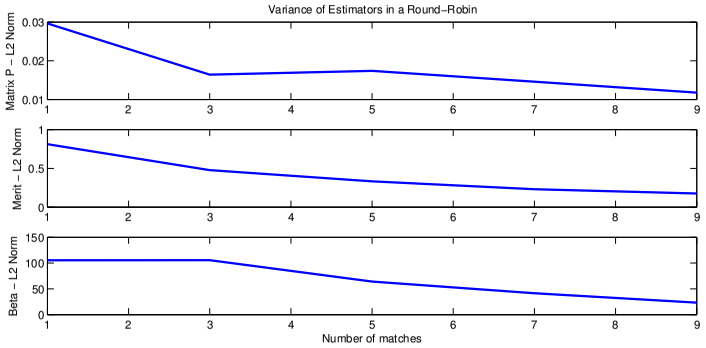


Figure 5: Variance of estimators in a round-robin with increasing number of matches between each pair.

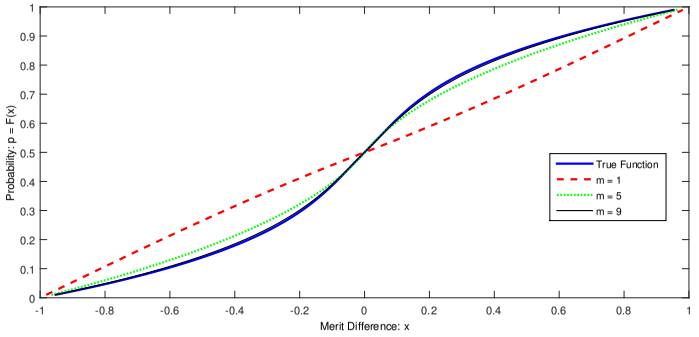


Figure 6: Recovered function graph.

	CV	AICc	AIC	BIC	CV	AICc	AIC	BIC
$D=1$	100%	100%	100%	100%	22%	30.3%	24.4%	37.4%
$D=3$	0%	0%	0%	0%	63.7%	67.3%	69.8%	61.6%
$D=5$	0%	0%	0%	0%	11.8%	1.5%	4.9%	0.1%
$D=7$	0%	0%	0%	0%	2.4%	0.8%	0.8%	0.8%
$D=9$	0%	0%	0%	0%	0.1%	0.1%	0.1%	0.1%
	$m_{ij}=1$				$m_{ij}=3$			

Table 1: Model Selection ($I=6$, $D=5$)

	CV	AICc	AIC	BIC	CV	AICc	AIC	BIC
$D=1$	100%	100%	100%	100%	7.1%	2.8%	2.6%	6.8%
$D=3$	0%	0%	0%	0%	87.4%	89%	85.2%	92.5%
$D=5$	0%	0%	0%	0%	5.3%	8.2%	12.2%	0.7%
$D=7$	0%	0%	0%	0%	.2%	0%	0%	0%
$D=9$	0%	0%	0%	0%	0%	0%	0%	0%
	$m_{ij}=1$				$m_{ij}=3$			

Table 2: Model Selection ($I=12$, $D=5$)

where $l(D)$ is the least-squares loss function, given in (6) and evaluated at the estimated parameters, $I+D$ is the number of parameters in the model and n is the sample-size. The corrected AIC (AICc) is

$$\mathcal{AIC}_c(D) \equiv \mathcal{AIC}(D) + \frac{2(I+D+1)(I+D+2)}{n-I-D-2} \tag{27}$$

and is designed to correct for small sample-sizes. The BIC method penalizes more the number of parameters and is defined by

$$\mathcal{BIC}(D) \equiv (I+D)\log(n) + n\log(l(D)). \tag{28}$$

Tables 1 and 2 compares the AIC, the AICc and the BIC methods in a round-robin data generated with $I=6$ and $I=12$ objects and with $m_{ij}=1$ for all ij, as well as $m_{ij}=3$. The data was generated monotone polynomials of degree 5 randomly selected with uniform coefficients and projected to the monotone cone. We display the frequency in which the methods correctly identify the degree of the polynomial as opposed to overfit/underfit.

Tables 1 and 2 show the empirical performance of the model selection criteria as a function of number of items I and the number of paired comparisons m_{ij}. For low values of m_{ij} all criteria select the lowest degree polynomial, i.e., $D=1$. When the number of comparisons increases the procedures tend to select larger values of D. In general the performance of the procedures are comparable. However, AIC and Cross validation do seem to (slightly) outperform the other methods. Cross validation is significantly more demanding computationally than AIC and thus as a rule of thumb we recommend the use of the AIC method when no further information is available.

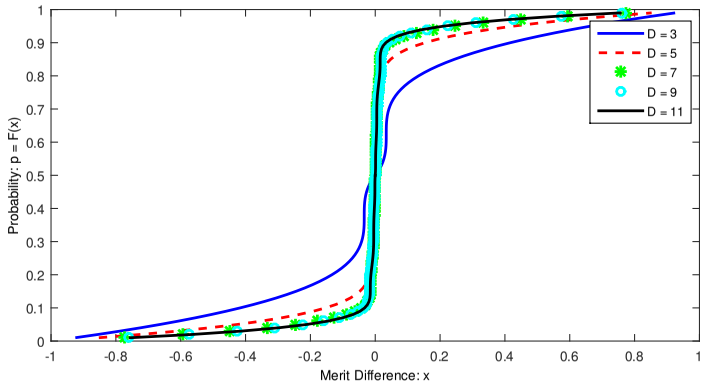


Figure 7: The effect of increasing the dimension D in estimating function F.

5.1. Illustrative Example

In this subsection we illustrate the use of PolyRank on a computer-chess data set[1]. The data set comprises of matches between 186 free single-CPU chess-engines. Each chess-engine played (roughly) 32 matches against 40 opponents. We use PolyRank to estimate model parameters from observed matches that resulted in a victory or defeat; ties are ignored. Figure 7 shows the estimated comparison function for various values of D when for the first 100 chess-engines. As is observed the estimated function $F_{\hat{\beta}}$ seems to stabilize for D greater than or equal to 7. Figure 8 shows the estimated function when the dimension $D=7$ is fixed and the number of chess-engines I is gradually increased. For I greater than or equal to 60 our estimated function seems to stabilize. Finally, Figure 9 compares the best fit function recovered by PolyRank to the family of BTL models described by $F_{\mathrm{BTL}}(x) = 1/(1+\exp(-\kappa x))$ for various values of $\kappa > 0$. Somewhat surprisingly, it seems that the family of BTL functions does not provide a good fit.

To illustrate this point consider three players i, j and k such that $p_{ij} = p_{jk} = \alpha > 0.5$ and $p_{ik} = \beta$, then, from (1) we have that $\beta = F(2F^{-1}(\alpha))$. For low values of α (say $\alpha = 0.55$) the BTL model and the polynomila model estimated by PolyRank virtually agree on the value of β (BTL: $\beta \approx 0.6$; Polynomial: $\beta \approx 0.59$); for intermediate values of α (say, $\alpha = 0.7$) the models begin to diverge $\beta \approx 0.84$; Polynomial: $\beta \approx 0.77$) and for large values of α (say $\alpha = 0.9$) this divergence is even more extreme (BTL: $\beta \approx 0.99$; Polynomial: $\beta \approx 0.92$). The estimated polynomial model seems to be more agreeable with the the data at hand since very few chess engines had a (near) perfect win against any opponent. It will be interesting to investigate whether our findings hold for human chess as well.

1. Publicly available at `http://kirill-kryukov.com/chess/kcec/games.html`.

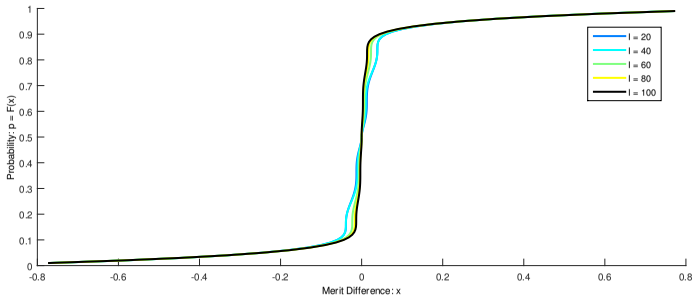


Figure 8: The effect of increasing the amount of data in estimating function F.

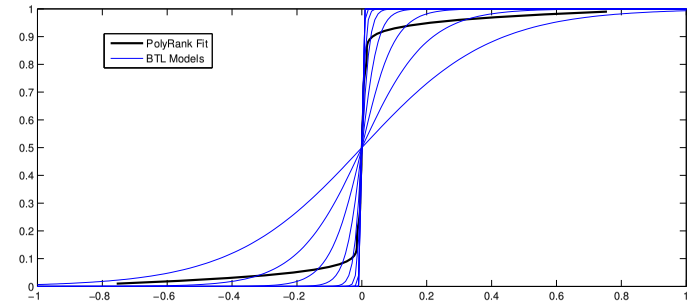


Figure 9: Bradley-Terry-Luce models compared to the best fit comparison function.

6. Summary and Discussion

In this paper we propose a new method for analyzing paired comparison data. Our main contribution is to relax the assumption that the comparison function is known in advance. Instead, we assume that the inverse of the comparison function is a D'th degree polynomial and the comparison function has a bounded support. We show that estimation reduces to a tractable SDP that simultaneously recovers the merit vector and the underlying comparison function F from an initial estimator $\hat{\boldsymbol{P}}$. We refer to this new methodology as PolyRank. We provide non-stochastic as well as stochastic guarantees for our estimators. This includes a thorough sensitivity analysis and additionally convergence and concentration bounds. Our simulation study demonstrates that the method works well in practice. Finally, we investigate a large data set of computer chess matches and provide evidence that the comparison function used for calculating chess ratings for almost nine decades seems to be inadequate, at least for computer chess engines.

Our work shows that PolyRank can be used whenever the existing methods, which assume that the comparison function is known, are used. The only additional requirement is that the comparison graph must have at least $I+D-1$ edges, a condition which is almost always satisfied in practice. Thus, PolyRank provides a flexible and principled alternative to the existing methods for ranking and rating which are based on paired comparisons. Our analysis, however, is just a starting point and many open research problems remain. It is clear that PolyRank can be extended in various directions; these can be grouped into several domains including: (i) modelling issues; (ii) computational/numerical issues; (iii) statistical and inferential problems of varied types.

Modelling. We have assumed that F^{-1} is given by a polynomial. Many other models, in which the polynomial in (4) is replaced by some other set of basis functions, are possible. Monotone splines provide a class of such functions (Ramsay, 1988). One other interesting possibility, with more of a statistical flavor, is to write

$$F^{-1}(p) = \sum_{i=1}^{D} \beta_i K_i(p)$$

where $K_i(p)$ are themselves quantile functions of symmetric random variables. This equation can be viewed as a mixture model on the quantile scale. The family $\{K_i\}$ is then chosen by the investigator; the symmetrized beta family of distributions seems like a suitable family to explore. Another, important issue is the incorporation of covariates, such as time or a "home advantage", as well as many others in the model. This, again, can be done in several ways. The merits can be modeled as regression functions or alternatively one can incorporate the covariates directly into the comparison function. Other issues which deserve attention are the modelling of ties and the comparison of more than two items a time.

Computations and Numerics. Compared with traditional methods, where the comparison function is given, PolyRank has higher computational complexity and may suffer from numerical instability. In part, the numerical issues are related to our decision to model the inverse of the comparison function as a polynomial. This in turn entails that the normal equations are associated with Vandermonde matrices. A known way to circumvent

this problem is to use a different basis for solving equation (10), this amounts to choosing a different basis for the polynomial regression, such as Chebyshev Polynomials. Another possibility is the use of a different loss function which is less sensitive to numerical issues. Developing a method to uncouple the estimation of F and of $\boldsymbol{\mu}$ as we provided for the unweighted $\mathcal{L}_2$ norm might provide further insight in this direction. One other, future objective, is to extend the practical reach of PolyRank to larger values of D and I while at the same time increasing computational and numerical efficiency. There may be several ways of doing so. One approach is hand crafting a solver for the SDP at hand. Another possibility is developing an online distributed version of PolyRank in which the function and the merits are updated after each pairwise comparison is observed.

Statistics. The current paper leaves many statistical issues unresolved. For example, we did not provide any results on the asymptotic distributions of our estimated parameters. We believe, however, that normal limits are obtained provided $(\boldsymbol{\mu}, \boldsymbol{\beta})$ are in the interior of the parameter space. It is also clear that employing the one step method we can obtain a fully efficient estimator (Fan and Chen, 1999). Other issues of interest are limit theorems for the case where $I \to \infty$ and when paired comparisons are made adaptively. In the adaptive set up one may exploit the fact that function F can be recovered up to arbitrary precision by using a small subset of the items in order to reduce the overall query complexity of the paired comparison experiment.

Acknowledgments

This paper was written when the first author was a graduate student at the Israel Institute of Technology. The research leading to these results has received funding from the European Research Council under European Union's Horizon 2020 Program, ERC Grant agreement no. 682203 "SpeedInfTradeoff" and the research of Ori Davidov was partially supported by the Israeli Science Foundation Grants No. 1256/13 and 457/17.

Appendix A. Proof of Theorems

The following contains the proofs of our main results.

A.1. Proof of Theorem 1

Proof The constraint $F(0) = 1/2$ is equivalent to $F^{-1}(1/2) = \sum_{i=0}^{D} \beta_i (1/2)^i = 0$, which is the last equality constraint in (9). Also, F is increasing iff $F^{-1}(p)$ is increasing and for our set of polynomials this is equivalent to $(F^{-1}(p))' = \beta_1 + 2 \cdot \beta_2 p + ... + D \cdot \beta_D p^{D-1} \geq 0$ for every $p \in [0, 1/2]$. In addition, F is L-Lipschitz continuous and so $|F'(x)| \leq L$; which combined with the monotonicity constraint is equivalent to the constraint $(F^{-1}(p))' \geq 1/L$. By Theorem 6 of (Parrilo, 2016) we have that $(F^{-1}(p))' - 1/L = s(x) + x(1/2 - x)t(x)$ where $s(x)$ and $t(x)$ are sum of squares polynomial functions of degree at most $2d$ and $2d-2$ respectively. Now by Lemma 4 of Parrilo (2016) there exists $\boldsymbol{Q^0} \in \mathbb{S}_+^{d+1}$ and $\boldsymbol{Q^1} \in \mathbb{S}_+^{d}$ such that the coefficients of the polynomials $s(x)$ and $t(x)$ are $s_i = \sum_{j+k=i} Q^0_{jk}$ and $t_i = \sum_{j+k=i} Q^1_{jk}$. By combining these conditions on the polynomial $(F^{-1}(p))' - 1/L$ and the constraint $||\boldsymbol{\beta}||_\infty \leq U$ we obtain the desired result. ■

A.2. Proof of Theorem 3

Proof A little algebra show that

$$||\Delta\hat{\boldsymbol{\mu}} - \Delta\boldsymbol{\mu}||_2^2 = 2n||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||_2^2; \tag{29}$$

this is so because

$$||\Delta\hat{\boldsymbol{\mu}} - \Delta\boldsymbol{\mu}||_2^2 = \sum_{ij}(\hat{\mu}_i - \hat{\mu}_j - \mu_i + \mu_j)^2 = 2n||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||_2^2 - 2(\sum_i \hat{\mu}_i - \sum_i \mu_i)^2$$

where the last term is zero by construction. It follows that

$$||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||_2 = \frac{1}{\sqrt{2n}}||\Delta\hat{\boldsymbol{\mu}} - \Delta\boldsymbol{\mu}||_2 \leq \frac{1}{\sqrt{2n}}||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}||_2$$

$$= \frac{1}{\sqrt{2n}}||F^{-1}(\hat{\boldsymbol{P}}) - F^{-1}(\boldsymbol{P})||_2 \leq \frac{4U}{\sqrt{2n}}||\hat{\boldsymbol{P}} - \boldsymbol{P}||_2.$$

The first inequality is a conscequence of the convex projection theorem and the first equality follows from (29). Equation (13) is derived from

$$||\boldsymbol{P}^* - \boldsymbol{P}||_2^2 = ||F(\Delta\hat{\boldsymbol{\mu}}) - F(\Delta\boldsymbol{\mu})||_2^2 \leq L^2||\Delta\hat{\boldsymbol{\mu}} - \Delta\boldsymbol{\mu}||_2^2 = L^2 2n||\hat{\boldsymbol{\mu}} - \boldsymbol{\mu}||_2^2 \leq L^2(4U)^2||\hat{\boldsymbol{P}} - \boldsymbol{P}||_2^2.$$

Where the last equality is an application of equation (29) and the last inequality an application of (14). If $\hat{\boldsymbol{P}}$ is assumed to obey strong stochastic transitivity, then $\hat{p}_{ij} < 1/2$ implies that $\hat{p}_{ik} \leq \hat{p}_{jk}$ for all k and $\hat{p}_{ik} < \hat{p}_{jk}$ for at least some k. Thus, the identity $\hat{\mu}_i - \hat{\mu}_j = (1/n)\sum_k (F^{-1}(\hat{p}_{ik}) - F^{-1}(\hat{p}_{jk})) < 0$ together with the fact that F^{-1} is strictly monotone assures that the strong stochastic transitivity order is the same as the order of the estimated merits. ■

Remark Theorem 3, applies to other norms with the proper modifications. Assume that the estimator $\boldsymbol{P}^*$ is obtained via (3) under the norm or semi-norm $||\cdot||_\#$. Let $L_\#$ and $4U_\#$ be the Lipschitz constants of F and F^{-1} associated with $||\cdot||_\#$, then

$$||\boldsymbol{P}^* - \boldsymbol{P}||_\# \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}^*||_\# + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# \leq L_\#||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}||_\# + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# \leq$$

$$L_\#||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}||_\# + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# \leq 4L_\#U_\#||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# = (1+4L_\#U_\#)||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\#,$$

as in (13).

A.3. Proof of Theorem 4

Proof Equation (16) is a conscequence of

$$||\boldsymbol{P}^* - \boldsymbol{P}|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}^*|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| \leq L||F^{-1}_{\hat{\boldsymbol{\beta}}}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||$$

$$\leq L||F^{-1}_{\boldsymbol{\beta}}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| \leq 4LU||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}||;$$

where the last inequality stems from the fact that $F^{-1}_{\boldsymbol{\beta}}(p)$ is $4U$-Lipschitz continuous for every $F_{\boldsymbol{\beta}}(p) \in \mathcal{F}$ and the previous inequality is a consequence of the optimality of PolyRank. In order to prove (17), consider a set of linear equations $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ and a perturbed version $(\boldsymbol{A}+\Delta\boldsymbol{A})(\boldsymbol{x}+\Delta\boldsymbol{x}) = b$ where both $\boldsymbol{A}$ and $\boldsymbol{A}+\Delta\boldsymbol{A}$ are non-singular square matrices. Under these conditions one can show that $\Delta\boldsymbol{x} = \boldsymbol{A}^{-1}\Delta\boldsymbol{A}(\boldsymbol{x}+\Delta\boldsymbol{x})$; thus:

$$\begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} \end{pmatrix} = \boldsymbol{A}(\boldsymbol{P})^{-1}[\boldsymbol{A}(\boldsymbol{P}^*) - \boldsymbol{A}(\boldsymbol{P})] \begin{pmatrix} \hat{\boldsymbol{\beta}} \\ \hat{\boldsymbol{\mu}} \end{pmatrix}. \tag{30}$$

Notice that $||\hat{\boldsymbol{\beta}}||_\infty \leq U$ and also $||\hat{\boldsymbol{\mu}}||_\infty \leq U$. The second claim is true for if $\hat{\mu}_j \geq U$ for some j then $|\hat{\mu}_i - \hat{\mu}_j| = |F^{-1}_{\hat{\boldsymbol{\beta}}}(\hat{p}^*_{ij})| \leq |F^{-1}_{\hat{\boldsymbol{\beta}}}(0)| = |\beta_0| \leq U$ which then implies that $\hat{\mu}_i \geq 0$ for every i and so $\sum_i \hat{\mu}_i \geq U > 0$ which violates the constraint $\sum_i \hat{\mu}_i = 0$; therefore we must have $\hat{\mu}_j < U$ for every j (the analogous argument is valid for $\hat{\mu}_j > -U$). Using the equivalence between norms find:

$$\left|\left|\begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} \end{pmatrix}\right|\right| \leq U\sqrt{I+D}||\boldsymbol{A}(\boldsymbol{P})^{-1}||||\boldsymbol{A}(\boldsymbol{P}^*) - \boldsymbol{A}(\boldsymbol{P})||.$$

Now notice that

$$||\boldsymbol{A}(\boldsymbol{P}^*) - \boldsymbol{A}(\boldsymbol{P})||^2 = \sum_{k=1}^{D+I-1}\sum_{n=1}^{D}(p^{*n}_{(ij)_k} - p^n_{(ij)_k})^2 \leq \sum_{k=1}^{D+I-1} D(p^*_{(ij)_k} - p_{(ij)_k})^2 \leq D||\boldsymbol{P}^* - \boldsymbol{P}||^2$$

and therefore:

$$\left|\left|\begin{pmatrix} \hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu} \end{pmatrix}\right|\right| \leq U(4LU+1)\sqrt{D(I+D)}||\boldsymbol{A}(\boldsymbol{P})^{-1}||||\hat{\boldsymbol{P}} - \boldsymbol{P}||,$$

which completes the proof of (17). Now we will prove equation (18). A bit of algebra shows that for every $F_{\boldsymbol{\beta}} \in \mathcal{F}$ we have that $\max_{\alpha\in[0,1]}|F^{-1}_{\boldsymbol{\beta}}(\alpha) - F^{-1}_{\hat{\boldsymbol{\beta}}}(\alpha)| \leq 2||\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}||_\infty$; combining

this with (30), (17) and the lipschitz continuity of $F_{\boldsymbol{\beta}}$ we find the desired result. Finally, to prove order preservation, recognize that problem (9) can be solved by minimizing in $\boldsymbol{\mu}$ and in $\boldsymbol{\beta}$ separately. By minimizing on $\boldsymbol{\mu}$ we find the same closed form solution as the least squares refinement procedure, namely $\mu_i = (1/I)\sum_j F^{-1}_{\boldsymbol{\beta}}(\hat{p}_{ij})$. The proof follows by the same arguments as in Theorem 3. ■

A.4. Proof of Theorem 5

Proof We will first prove that $\max_{\alpha\in[0,1]}|F^{-1}_{\hat{\boldsymbol{\beta}}}(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)| \geq 2|\beta_D|/8^D$. In the following proof $\mathcal{P}_{D-1}$ is the set of polynomials of degree less than or equal to $D-1$.

$$\max_{\alpha\in[0,1]}|F^{-1}_{\hat{\boldsymbol{\beta}}}(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)| = \max_{\alpha\in[0,1/2]}|F^{-1}_{\hat{\boldsymbol{\beta}}}(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)|$$

$$\geq \min_{G\in\mathcal{F}_{D'}}\max_{\alpha\in[0,1/2]}|G(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)| \geq \min_{G^{-1}\in\mathcal{P}_{D-1}}\max_{\alpha\in[0,1/2]}|G^{-1}(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)|$$

$$= \min_{F^{-1}_{\tilde{\boldsymbol{\beta}}}\in\mathcal{P}_{D-1}}\max_{\alpha\in[0,1/2]}|F^{-1}_{\tilde{\boldsymbol{\beta}}}(\alpha) + \beta_D\alpha^D|$$

$$= |\beta_D|\min_{F^{-1}_{\tilde{\boldsymbol{\beta}}}\in\mathcal{P}_{D-1}}\max_{\alpha\in[0,1/2]}|F^{-1}_{\tilde{\boldsymbol{\beta}}}(\alpha) + \alpha^D|$$

$$= |\beta_D|\min_{F^{-1}_{\tilde{\boldsymbol{\beta}}}\in\mathcal{P}_{D-1}}\max_{\alpha\in[-1,1]}|F^{-1}_{\tilde{\boldsymbol{\beta}}}((\alpha+1)/4) + ((\alpha+1)/4)^D|$$

$$\geq \frac{|\beta_D|}{4^D}\min_{F^{-1}_{\tilde{\boldsymbol{\beta}}}\in\mathcal{P}_{D-1}}\max_{\alpha\in[-1,1]}|F^{-1}_{\tilde{\boldsymbol{\beta}}}(\alpha) + \alpha^D|$$

$$= \frac{|\beta_D|}{4^D}\frac{1}{2^{D-1}}$$

The last equality is a defining property of Chebyshev polynomials (Mason and Handscomb, 2002). Now, notice that the $4U$-Lipschitz continuity of $F^{-1}_{\boldsymbol{\beta}}$ is equivalent to $|F_{\boldsymbol{\beta}}(y) - F_{\boldsymbol{\beta}}(x)| \geq (1/4U)|y-x|$; combining this with $\max_{\alpha\in[0,1]}|F^{-1}_{\hat{\boldsymbol{\beta}}}(\alpha) - F^{-1}_{\boldsymbol{\beta}}(\alpha)| \geq 2|\beta_D|/8^D$ we obtain equation (19). ■

A.5. Proof of Theorem 6

Proof Let $\hat{l}(D)$ be the empirical loss of a D dimensional fit provided by PolyRank, then:

$$\hat{l}(D) = \min_{\beta\in\mathbb{R}^D,\mu\in\mathbb{R}^I}\sqrt{\sum_{(ij)\in\mathcal{S}}(\beta_0 + ... + \beta_D\hat{p}^D_{ij} - \mu_i + \mu_j)^2}$$

where the minimum is taken over the sets specified by PolyRank for some triplet (D, U, L). It is also true that:

$$\hat{l}(D) = \min_{\beta \in \mathbb{R}^{D+1}, \mu \in \mathbb{R}^I} \sqrt{\sum_{(ij)\in\mathcal{S}} (\beta_0 + ... + \beta_D \hat{p}_{ij}^D - \mu_i + \mu_j)^2}$$

for the set specified by the triplet $(D+1, U, L)$, and so

$$\hat{l}(D) \leq \min_{\beta \in \mathbb{R}^{D+1}, \mu \in \mathbb{R}^I} \sqrt{\sum_{(ij)\in\mathcal{S}} (\beta_0 + ... + \beta_D \hat{p}_{ij}^D + \beta_{D+1}\hat{p}_{ij}^{D+1} - \mu_i + \mu_j)^2} + \sqrt{\sum_{(ij)\in\mathcal{S}} (\beta_{D+1}\hat{p}_{ij}^{D+1})^2}$$

$$\leq \min_{\beta \in \mathbb{R}^{D+1}, \mu \in \mathbb{R}^I} \sqrt{\sum_{(ij)\in\mathcal{S}} (\beta_0 + ... + \beta_D \hat{p}_{ij}^D + \beta_{D+1}\hat{p}_{ij}^{D+1} - \mu_i + \mu_j)^2} + U\sqrt{\sum_{(ij)\in\mathcal{S}} ((1/2)^{D+1})^2}$$

$$\leq \hat{l}(D+1) + U\,(1/2)^{D+1}\sqrt{|\mathcal{S}|}.$$

We have shown that $\hat{l}(D) \leq \hat{l}(D+1) + U\sqrt{|\mathcal{S}|}/2^{D+1}$ which implies that:

$$\hat{l}(D) \leq \hat{l}(D+K) + U\sqrt{|\mathcal{S}|}\sum_{i=D+1}^{D+K}\frac{1}{2^i} = \hat{l}(D+K) + \frac{U\sqrt{|\mathcal{S}|}}{2^D}\sum_{i=1}^{K}\frac{1}{2^i}.$$

Therefore taking the limit of $K \to \infty$ we find that $\hat{l}(D) \leq \hat{l}(\infty) + U\sqrt{|\mathcal{S}|}/2^D$; combining this with the optimality of the estimated parameters we obtain

$$||F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}|| \leq ||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}|| + \frac{U\sqrt{|\mathcal{S}|}}{2^D}. \tag{31}$$

To complete the proof of (20) notice that

$$||\boldsymbol{P}^* - \boldsymbol{P}|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}^*|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + L||F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}||$$

$$\leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + L||F^{-1}(\hat{\boldsymbol{P}}) - \Delta\boldsymbol{\mu}|| + \frac{LU\sqrt{|\mathcal{S}|}}{2^D} \leq (1+4LU)||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + \frac{1}{2^D}LUI.$$

■

Remark One could equivalently prove that PolyRank defined with the $\mathcal{L}_1$ norm satisfies $||\boldsymbol{P}^* - \boldsymbol{P}||_1 \leq (1+4LU)||\hat{\boldsymbol{P}} - \boldsymbol{P}||_1 + (1/2^D)LUI^2$ for analytic functions with bounded coefficients and with the $\mathcal{L}_\infty$ norm one finds that $||\boldsymbol{P}^* - \boldsymbol{P}||_\infty \leq (1+4LU)||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\infty + (1/2^D)LU$. We provide a sketch of the proof for a generic norm $||\cdot||_\#$. Again, we take $\mathcal{L}_\#$ to be the Lipschitz constant of F associated to $||\cdot||_\#$ and $4U_\#$ the Lipschitz constant of F^{-1} associated to $||\cdot||_\#$, then, we find that

$$||\boldsymbol{P}^* - \boldsymbol{P}||_\# \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# + ||\hat{\boldsymbol{P}} - \boldsymbol{P}^*||_\# \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# + L_\#||F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}||_\#;$$

then an inequality similar to (31) is obtained for the norm $||\cdot||_\#$. Norm equivalence guarantees that this can be done up to constant factors. Then, combining the two inequalities one finds that for some $k_1 \geq 1$ and some $k_2 \geq 0$ the following inequality holds:

$$||\boldsymbol{P}^* - \boldsymbol{P}||_\# \leq k_1||\hat{\boldsymbol{P}} - \boldsymbol{P}||_\# + \frac{k_2}{s^D}.$$

This completes the proof.

A.6. Proof of Theorem 7

Proof Equation (21) is a conscequence of the L-Lipschitz continuity of functions in $\mathcal{F}$:

$$||\boldsymbol{P}^* - \boldsymbol{P}|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + ||\hat{\boldsymbol{P}} - \boldsymbol{P}^*|| \leq ||\hat{\boldsymbol{P}} - \boldsymbol{P}|| + L||F_{\hat{\boldsymbol{\beta}}}^{-1}(\hat{\boldsymbol{P}}) - \Delta\hat{\boldsymbol{\mu}}||.$$

■

A.7. Proof of Theorem 8

We will use of the following lemma whose proof is virtually identical to that of Theorem 4:

Lemma 10 *Let $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\mu}}$ be estimated using* PolyRank. *Then,*

$$\left|\left|\begin{pmatrix}\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\end{pmatrix}\right|\right| \leq K||\hat{\boldsymbol{P}} - \boldsymbol{P}||_{\boldsymbol{W}}, \tag{32}$$

where $K \leq U(1+4LU)\sqrt{D(I+D)}||\boldsymbol{A}_{\boldsymbol{W}}^{-1}||$ and $\boldsymbol{A}_{\boldsymbol{W}}$ is defined as in 10 with each line multiplied by its respective weight.

Proof Note that:

$$\mathbb{P}\left\{\left|\left|\begin{pmatrix}\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\end{pmatrix}\right|\right| \geq \epsilon\right\} = \mathbb{P}\left\{\left|\left|\begin{pmatrix}\hat{\boldsymbol{\beta}} - \boldsymbol{\beta} \\ \hat{\boldsymbol{\mu}} - \boldsymbol{\mu}\end{pmatrix}\right|\right|^2 \geq \epsilon^2\right\} \leq \mathbb{P}\left\{K^2\sum w_{ij}|\hat{p}_{ij} - p_{ij}|^2 \geq \epsilon^2\right\}$$

$$\leq \mathbb{P}\left\{|E|\max_{ij}\{w_{ij}|\hat{p}_{ij} - p_{ij}|^2\} \geq \frac{\epsilon^2}{K^2}\right\} \leq \sum_{ij}\mathbb{P}\left\{w_{ij}|\hat{p}_{ij} - p_{ij}|^2 \geq \frac{\epsilon^2}{|E|K^2}\right\}$$

$$= \sum_{ij}\mathbb{P}\left\{|\hat{p}_{ij} - p_{ij}| \geq \frac{\epsilon}{K\sqrt{w_{ij}|E|}}\right\} \leq 2\sum_{ij}\exp\left\{-2m_{ij}\left(\frac{\epsilon}{K\sqrt{w_{ij}|E|}}\right)^2\right\}.$$

Where the last inequality follows from Hoeffding's bound; thus, taking $m_{ij} = nw_{ij}$ we have:

$$= 2\sum_{ij}\exp\left\{-2nw_{ij}\left(\frac{\epsilon}{K\sqrt{w_{ij}|E|}}\right)^2\right\} \leq 2|E|\exp\left\{-2n\frac{\epsilon^2}{K^2|E|}\right\},$$

which completes our proof. ■

References

I. Adler, Y. Cao, R. Karp, E.A. Pekoz, and S.M. Ross. Random knockout tournaments. *Operations Research*, 2017.

N. Ailon. An active learning algorithm for ranking from pairwise preferences with almost optimal query complexity. *Journal of Machine Learning Research*, 2012.

P.D. Allison and N.A. Christakis. Logit models for sets of ranked items. *Sociological Methodology*, 1994.

S.K. Baek, I.G. Yi, H.J. Park, and B.J. Kim. Universal statistics of the knockout tournament. *Nature, Scientific Reports*, 2013.

M. Balinski and R. Laraki. *Majority Judgment, Measuring, Ranking, and Electing*. The MIT Press, 2010.

W.H. Batchelder, N.J. Bershal, and R.S. Simpson. Dynamic paired-comparison scaling. *Journal of Mathematical Psychology*, 1992.

R.A. Bradley and M.E. Terry. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 1952.

S. Chatterjee and S. Mukherjee. On estimation in tournaments and graphs under monotonicity constraints. *arXiv:1603.04556 [math.ST]*, 2016.

F.R.K. Chung. Spectral graph theory. *AMS and CBMS*, 1994.

F.R.K. Chung and F.K. Hwang. Do stronger players win more knockout tournaments? *Journal of the American Statistical Association*, 1978.

G. Claeskens and N.L. Hjort. Model selection and model averaging. *Cambridge Series in Statistical and Probabilistic Mathematics*, 2006.

W.N. Colley. Colley's bias free college football ranking method: The colley matrix explained, 2002. URL `http://www.colleyrankings.com/matrate.pdf`.

P. Cremonesi, Y. Koren, and R. Turrin. Performance of recommender algorithms on top-n recommendation tasks. *Proceedings of the fourth ACM conference on recommender systems*, 2010.

H.A. David. *The method of paired comparisons*. Hodder Arnold, 1988.

J.S. deCani. Maximum likelihood paired comparison ranking by linear programming. *Biometrika*, page 537, 1969.

J. Fan and J. Chen. One-step local quasi-likelihood estimation. *Journal of the Royal Statistical Society*, 1999.

P. Favardin, D. Lepelley, and J. Serais. Borda rule, copeland method and strategic manipulation. *Rev.Econ.Design*, 2002.

A.Y. Govan. Ranking theory with application to popular sports, 2008.

R. Heckel, N.B. Shah, K. Ramchandran, and M.J. Wainwright. Active ranking from pairwise comparisons and when parametric assumptions don't help. *arxiv:1606.08842v2 [cs.LG]*, 2016.

R. Herbrich, T. Minka, and T. Graepel. Trueskill tm: A bayesian skill rating system. *Advances in Neural Information Processing Systems, MIT Press*, 2007.

P.J. Huber. Pairwise comparison and ranking: optimum properties of the row sum procedure. *Annals of Mathematical Statistics*, 1963.

S.H. Hwang. Contest success functions: Theory and evidence. *Economics Department Working Paper Series. 11.*, 2009. URL `http://scholarworks.umass.edu/econ_workingpaper/11`.

R.B. Israel. Stronger players need not win more knockout tournaments. *Journal of the American Statistical Association*, 1981.

H. Jia, S. Skaperdas, and S. Vaidya. Contest functions: Theoretical foundations and issues in estimation. *International Journal of Industrial Organization*, 2013.

X. Jiang, L.H. Lim, Y. Yao, and Y. Ye. Statistical ranking and combinatorial hodge theory. *Mathematical Programming*, 2010.

J. Levin and B. Nalebuff. An introduction to vote-counting schemes. *Journal of Economic Perspectives*, 1995.

J.I. Marden. *Analyzing and Modeling Rank Data*. Chapman and Hall/CRC, 1996.

J.C. Mason and D.C. Handscomb. *Chebyshev Polynomials*. CRC Press, 2002.

K. Massey. Massey ratings, 2017. URL `https://www.masseyratings.com/theory/massey97.pdf`.

H.W. Morrison. Testable conditions for triads of paired comparison choices. *Psychometrika*, 1963.

A. Mutapcic and S. Boyd. Cutting-set methods for robust convex optimization with pessimizing oracles. *Optimization Methods & Software*, 2009.

A.S. Nemirovski and M.J. Todd. Interior-point methods for optimization. *Acta Numerica*, 2009.

I.F.D. Oliveira, S. Zehavi, and O. Davidov. Stochastic transitivity: Axioms and models. *Journal of Mathematical Psychology*, 2018.

E. Pacuit. Voting methods. *In The Stanford Encyclopedia of Philosophy*, 2012.

V. Pan. How bad are vandermonde matrices? *SIAM Journal on Matrix Analysis and Applications*, 2015.

P.A. Parrilo. Algebraic techniques and semidefinite optimization, 2016. URL `http://stellar.mit.edu/S/course/6/sp10/6.256/courseMaterial/topics/topic2/lectureNotes/lecture-10/lecture-10.pdf`.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. *Proceedings of JMLR*, 2014.

J.O. Ramsay. Monotone regression splines in action. *Statistical Science*, 1988.

M. Regenwetter and C.P. Davis-Stober. There are many models of transitive preference: a tutorial review and current perspective. *Decision Modeling and Behavior in Complex and Uncertain Environments*, 2008.

M. Regenwetter, J. Dana, and C.P. Davis-Stober. Transitivity of preferences. *Psychological Review*, 2011.

D.G. Saari. Basic geometry of voting. *Springer-Verlag Berlin Heidelberg*, 1995.

N.B. Shah, S. Balakrishnan, J. Bradley, A. Parekh, K. Ramchandran, and M.J. Wainwright. Estimation from pairwise comparisons: sharp minimax bounds with topology dependence. *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, 2015a.

N.B. Shah, S. Balakrishnan, and A. Guntuboyina. Stochastically transitive models for pairwise comparisons: statistical and computational issues. *arXiv:1510.05610 [stat.ML]*, 2015b.

G. Simons and Y.C. Yao. Asymptotics when the number of parameters tends to infinity in the bradley-terry model for paired comparisons. *The Annals of Statistics*, 1999.

O. Stein. How to solve a semi-infinite optimization problem. *European Journal of Operational Research*, 2012.

S. Su. Fleximble parametric quantile regression model. *Stat Compt*, 2015.

I. Takeuchi, Q.V. Le, T.D. Sears, and A.J. Smola. Nonparametric quantile estimation. *Journal of Machine Learning Research*, 2006.

L.L. Thurstone. A law of comparative judgment. *Psychology Review*, 1927.

K. Tsukida and M.R. Gupta. How to analyze paired comparison data, 2011. URL `http://www.dtic.mil/dtic/tr/fulltext/u2/a543806.pdf`.

L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 1996.

J.I. Yellott. The relationship between thurstone's, luce's and dawkins' models for paired comparisons. *Annual Meeting of Mathematical Psychology*, 1970.

J.I. Yellott. The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential. *Journal of Mathematical Psychology*, 1977.

E. Zermelo. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 1928.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 7/17; Revised 2/18; Published 10/18

Simple Classification Using Binary Data

Deanna Needell DEANNA@MATH.UCLA.EDU
Department of Mathematics
520 Portola Plaza, University of California, Los Angeles, CA 90095

Rayan Saab RSAAB@UCSD.EDU
Department of Mathematics
9500 Gilman Drive, University of California, La Jolla, CA 92093

Tina Woolf TINA.WOOLF@CGU.EDU
Institute of Mathematical Sciences
150 E. 10th Street, Claremont Graduate University, Claremont CA 91711

Editor: David Wipf

Abstract

Binary, or one-bit, representations of data arise naturally in many applications, and are appealing in both hardware implementations and algorithm design. In this work, we study the problem of data classification from binary data obtained from the sign pattern of low-dimensional projections and propose a framework with low computation and resource costs. We illustrate the utility of the proposed approach through stylized and realistic numerical experiments, and provide a theoretical analysis for a simple case. We hope that our framework and analysis will serve as a foundation for studying similar types of approaches.

Keywords: binary measurements, one-bit representations, classification

1. Introduction

Our focus is on data classification problems in which only a *binary* representation of the data is available. Such binary representations may arise under a variety of circumstances. In some cases, they may arise naturally due to compressive acquisition. For example, distributed systems may have bandwidth and energy constraints that necessitate extremely coarse quantization of the measurements (Fang et al., 2014). A binary data representation can also be particularly appealing in hardware implementations because it is inexpensive to compute and promotes a fast hardware device (Jacques et al., 2013b; Laska et al., 2011); such benefits have contributed to the success, for example, of 1-bit Sigma-Delta converters (Aziz et al., 1996; Candy and Temes, 1962). Alternatively, binary, heavily quantized, or compressed representations may be part of the classification algorithm design in the interest of data compression and speed (Boufounos and Baraniuk, 2008; Hunter et al., 2010; Calderbank et al., 2009; Davenport et al., 2010; Gupta et al., 2010; Hahn et al., 2014). The goal of this paper is to present a framework for performing learning inferences, such as classification, from highly quantized data representations—we focus on the extreme case of 1-bit (binary) representations. Let us begin with the mathematical formulation of this problem.

Problem Formulation. Let $\{x_i\}_{i=1}^p \subset \mathbb{R}^n$ be a point cloud represented via a matrix

$$X = [x_1\ x_2\ \cdots\ x_p] \in \mathbb{R}^{n\times p}.$$

Moreover, let $A : \mathbb{R}^n \to \mathbb{R}^m$ be a linear map, and denote by $\text{sign} : \mathbb{R} \to \mathbb{R}$ the sign operator given by

$$\text{sign}(a) = \begin{cases} 1 & a \geq 0 \\ -1 & a < 0. \end{cases}$$

Without risk of confusion, we overload the above notation so the sign operator can apply to matrices (entrywise). In particular, for an m by p matrix M, and $(i,j) \in [m] \times [p]$, we define $\text{sign}(M)$ as the $m \times p$ matrix with entries

$$(\text{sign}(M))_{i,j} := \text{sign}(M_{i,j}).$$

We consider the setting where a classification algorithm has access to training data of the form $Q = \text{sign}(AX)$, along with a vector of associated labels $b = (b_1, \cdots, b_p) \in \{1, \ldots, G\}^p$, indicating the membership of each x_i to exactly one of G classes. Here, A is an m by n matrix. The rows of A define *hyperplanes* in $\mathbb{R}^n$ and the binary sign information tells us which side of the hyperplane each data point lies on. Throughout, we will primarily take A to have independent identically distributed standard Gaussian entries (though experimental results are also included for structured matrices). Given Q and b, we wish to train an algorithm that can be used to classify new signals, available only in a similar binary form via the matrix A, for which the label is unknown.

1.1. Contribution

Our contribution is a *framework* for classifying data into a given number of classes using only a binary representation (obtained as the sign pattern from low-dimensional projections, as described above) of the data. This framework serves several purposes: (i) it provides mathematical tools that can be used for classification in applications where data is already captured in a simple binary representation, (ii) demonstrates that for general problems, classification can be done effectively using low-dimensional measurements, (iii) suggests an approach to use these measurements for classification using low computation, (iv) provides a simple technique for classification that can be mathematically analyzed. We believe this framework can be extended and utilized to build novel algorithmic approaches for many types of learning problems. In this work, we present one method for classification using training data, illustrate its promise on synthetic and real data, and provide a theoretical analysis of the proposed approach in the simple setting of two-dimensional signals and two possible classes. Under mild assumptions, we derive an explicit lower bound on the probability that a new data point gets classified correctly. This analysis serves as a foundation for analyzing the method in more complicated settings, and a framework for studying similar types of approaches.

1.2. Organization

We proceed next in Section 1.3 with a brief overview of related work. Then, in Section 2 we propose a two-stage method for classifying data into a given number of classes using only a binary representation of the data. The first stage of the method performs training on data with known class membership, and the second stage is used for classifying new data points with a priori unknown class membership. Next, in Section 3 we demonstrate the potential of the proposed approach on both synthetically generated data as well as real datasets with application to handwritten digit recognition and facial recognition. Finally, in Section 4 we provide a theoretical analysis of the proposed approach in the simple setting of two-dimensional signals and two classes. We conclude in Section 5 with some discussion and future directions.

1.3. Prior Work

There is a large body of work on several areas related to the subject of this paper, ranging from classification to compressed sensing, hashing, quantization, and deep learning. Due to the popularity and impact of each of these research areas, any review of prior work that we provide here must necessarily be non-exhaustive. Thus, in what follows, we briefly discuss related prior work, highlighting connections to our work but also stressing the distinctions.

Support vector machines (SVM) (Christianini and Shawe-Taylor, 2000; Hearst et al., 1998; Joachims, 1998; Steinwart and Christmann, 2008) have become popular in machine learning, and are often used for classification. Provided a training set of data points and known labels, the SVM problem is to construct the optimal hyperplane (or hyperplanes) separating the data (if the data is linearly separable) or maximizing the geometric margin between the classes (if the data is not linearly separable). Although loosely related (in the sense that at a high level we utilize hyperplanes to separate the data), the approach taken in this paper is fundamentally different than in SVM. Instead of searching for the *optimal* separating hyperplane, our proposed algorithm uses many, randomly selected hyperplanes (via the rows of the matrix A), and uses the relationship between these hyperplanes and the training data to construct a classification procedure that operates on information between the same hyperplanes and the data to be classified.

The process of transforming high-dimensional data points into low-dimensional spaces has been studied extensively in related contexts. For example, the pioneering Johnson-Lindenstrauss Lemma states that any set of p points in high dimensional Euclidean space can be (linearly) embedded into $O(\epsilon^{-2}\log(p))$ dimensions, without distorting the distance between any two points by more than a small factor, namely ϵ (Johnson and Lindenstrauss, 1982). Since the original work of Johnson and Lindenstrauss, much work on Johnson-Lindenstrauss embeddings (often motivated by signal processing and data analysis applications) has focused on randomized embeddings where the matrix associated with the linear embedding is drawn from an appropriate random distribution. Such random embeddings include those based on Gaussian and other subgaussian random variables as well as those that admit fast implementations, usually based on the fast Fourier transform (Ailon and Chazelle, 2006; Achlioptas, 2003; Dasgupta and Gupta, 2003).

Another important line of related work is *compressed sensing*, in which it has been demonstrated that far fewer linear measurements than dictated by traditional Nyquist sampling can be used to represent high-dimensional data (Candès et al., 2006b,a; Donoho, 2006). For a signal $x \in \mathbb{R}^n$, one obtains $m < n$ measurements of the form $y = Ax$ (or noisy measurements $y = Ax + z$ for $z \in \mathbb{R}^m$), where $A \in \mathbb{R}^{m\times n}$, and the goal is to recover the signal x. By assuming the signal x is s-sparse, meaning that $\|x\|_0 = |\text{supp}(x)| = s \ll n$, the recovery problem becomes well-posed under certain conditions on A. Indeed, there is now a vast literature describing recovery results and algorithms when A, say, is a random matrix drawn from appropriate distributions (including those where the entries of A are independent Gaussian random variables). The relationship between Johnson-Lindenstrauss embeddings and compressed sensing is deep and bi-directional; matrices that yield Johnson-Lindenstrauss embeddings make excellent compressed sensing matrices (Baraniuk et al., 2006) and conversely, compressed sensing matrices (with minor modifications) yield Johnson-Lindenstrauss embeddings (Krahmer and Ward, 2011). Some initial work on performing inference tasks like classification from compressed sensing data shows promising results (Boufounos and Baraniuk, 2008; Hunter et al., 2010; Calderbank et al., 2009; Davenport et al., 2010; Gupta et al., 2010; Hahn et al., 2014).

To allow processing on digital computers, compressive measurements must often be *quantized*, or mapped to discrete values from some finite set. The extreme quantization setting where only the sign bit is acquired is known as *one-bit compressed sensing* and was introduced recently (Boufounos and Baraniuk, 2008). In this framework, the measurements now take the form $y = \text{sign}(Ax)$, and the objective is still to recover the signal x. Several methods have since been developed to recover the signal x (up to normalization) from such simple one-bit measurements (Plan and Vershynin, 2013a,b; Gopi et al., 2013; Jacques et al., 2013b; Yan et al., 2012; Jacques et al., 2013a). Although the data we consider in this paper takes a similar form, the overall goal is different; rather than signal *reconstruction*, our interest is data *classification*.

More recently, there has been growing interest in binary embeddings (embeddings into the binary cube (Plan and Vershynin, 2014; Yu et al., 2014; Gong et al., 2013; Yi et al., 2015; Choromanska et al., 2016; Dirksen and Stollenwerk, 2016), where it has been observed that using certain linear projections and then applying the sign operator as a nonlinear map largely preserves information about the angular distance between vectors provided one takes sufficiently many measurements. Indeed, the measurement operators used for binary embeddings are Johnson-Lindenstrauss embeddings and thus also similar to those used in compressed sensing, so they again range from random Gaussian and subgaussian matrices to those admitting fast linear transformations, such as random circulant matrices (Dirksen and Stollenwerk, 2016), although there are limitations to such embeddings for subgaussian but non-Gaussian matrices (Plan and Vershynin, 2014, 2013a). Although we consider a similar binary measurement process, we are not necessarily concerned with geometry preservation in the low-dimensional space, but rather the ability to still perform data classification.

Deep Learning is an area of machine learning based on learning data representations using multiple levels of abstraction, or layers. Each of these layers is essentially a function whose parameters are learned, and the full network is thus a composition of such functions. Algorithms for such deep neural networks have recently obtained state of the art results for classification. Their success has been due to the availability of large training data sets coupled with advancements in computing power and the development of new techniques (Krizhevsky et al., 2012; Simonyan and Zisserman, 2014; Szegedy et al., 2015; Russakovsky

et al., 2015). Randomization in neural networks has again been shown to give computational advantages and even so-called "shallow" networks with randomization and random initializations of deep neural networks have been shown to obtain results close to deep networks requiring heavy optimization (Rahimi and Recht, 2009; Giryes et al., 2016). Deep neural networks have also been extended to binary data, where the net represents a set of Boolean functions that maps all binary inputs to the outputs (Kim and Smaragdis, 2016; Courbariaux et al., 2015, 2016). Other types of quantizations have been proposed to reduce multiplications in both the input and hidden layers (Lin et al., 2015; Marchesi et al., 1993; Simard and Graf, 1994; Burge et al., 1999; Rastegari et al., 2016; Hubara et al., 2016). We will use randomized non-linear measurements but consider deep learning and neural networks as motivational to our multi-level algorithm design. Indeed, we are not tuning parameters nor doing any optimization as is typically done in deep learning, nor do our levels necessarily possess the structure typical in deep learning "architectures"; this makes our approach potentially simpler and easier to work with.

Using randomized non-linearities and simpler optimizations appears in several other works (Rahimi and Recht, 2009; Ozuysal et al., 2010). The latter work most closely resembles our approach in that the authors propose a "score function" using binary tests in the training phase, and then classifies new data based on the maximization of a class probability function. The perspective of this prior approach however is Bayesian rather than geometric, the score functions do not include any balancing terms as ours will below, the measurements are taken as "binary tests" using components of the data vectors (rather than our compressed sensing style projections), and the approach does not utilize a multi-level approach as ours does. We believe our geometric framework not only lends itself to easily obtained binary data but also a simpler method and analysis.

2. The Proposed Classification Algorithm

The training phase of our algorithm is detailed in Algorithm 1. Here, the method may take the binary data Q as input directly, or the training data $Q = \text{sign}(AX)$ may be computed as a one-time pre-processing step. For arbitrary matrices A, this step of course may incur a computational cost on the order of mnp. In Section 3, we also include experiments using structured matrices that have a fast multiply, reducing this cost to a logarithmic dependence on the dimension n. Then, the training algorithm proceeds in L "levels". In the ℓ-th level, m index sets $\Lambda_{\ell,i} \subset [m]$, $|\Lambda_{\ell,i}| = \ell$, $i = 1, ..., m$, are randomly selected, so that all elements of $\Lambda_{\ell,i}$ are unique, and $\Lambda_{\ell,i} \neq \Lambda_{\ell,j}$ for $i \neq j$. This is achieved by selecting the multi-set of $\Lambda_{\ell,i}$'s uniformly at random from a set of cardinality $\binom{\binom{m}{\ell}}{m}$. During the i-th "iteration" of the ℓ-th level, the rows of Q indexed by $\Lambda_{\ell,i}$ are used to form the $\ell \times p$ submatrix of Q, the columns of which define the sign patterns $\{\pm 1\}^\ell$ observed by the training data. For example, at the first level the possible sign patterns are 1 and -1, describing which side of the selected hyperplane the training data points lie on; at the second level the possible sign patters are $\begin{bmatrix}1\\1\end{bmatrix}$, $\begin{bmatrix}1\\-1\end{bmatrix}$, $\begin{bmatrix}-1\\1\end{bmatrix}$, $\begin{bmatrix}-1\\-1\end{bmatrix}$, describing which side of the two selected hyperplanes the training data points lie on, and so on for the subsequent levels. At each level, there are at most 2^ℓ possible sign patterns. Let $t = t(\ell) \in \{0, 1, 2, \dots\}$ denote the sign pattern *index* at level ℓ, where $0 \leq t \leq 2^\ell - 1$. Then, the binary (i.e., base 2) representation of each

$t = (t_\ell \dots t_2 t_1)_{\text{bin}} := \sum_{k=1}^{\ell} t_k 2^{k-1}$ is in one-to-one correspondence with the binary sign pattern it represents, up to the identification of $\{0, 1\}$ with the images $\{-1, 1\}$ of the sign operator. For example, at level $\ell = 2$ the sign pattern index $t = 2 = (10)_{\text{bin}}$ corresponds to the sign pattern $\begin{bmatrix}1\\-1\end{bmatrix}$.

For the t-th sign pattern and g-th class, a *membership index* parameter $r(\ell, i, t, g)$ that uses knowledge of the number of training points in class g having the t-th sign pattern, is calculated for every $\Lambda_{\ell,i}$. Larger values of $r(\ell, i, t, g)$ suggest that the t-th sign pattern is more heavily dominated by class g; thus, if a signal with unknown label corresponds to the t-th sign pattern, we will be more likely to classify it into the g-th class. In this paper, we use the following choice for the membership index parameter $r(\ell, i, t, g)$, which we found to work well experimentally. Below, $P_{g|t} = P_{g|t}(\Lambda_{\ell,i})$ denotes the number of training points from the g-th class with the t-th sign pattern at the i-th set selection in the ℓ-th level:

$$r(\ell, i, t, g) = \frac{P_{g|t}}{\sum_{j=1}^{G} P_{j|t}} \frac{\sum_{j=1}^{G} |P_{g|t} - P_{j|t}|}{\sum_{j=1}^{G} P_{j|t}}. \tag{1}$$

Let us briefly explain the intuition for this formula. The first fraction in (1) indicates the proportion of training points in class g out of all points with sign pattern t (at the ℓ-th level and i-th iteration). The second fraction in (1) is a balancing term that gives more weight to group g when that group is much different in size than the others with the same sign pattern. If $P_{j|t}$ is the same for all classes $j = 1, \dots, G$, then $r(\ell, i, t, g) = 0$ for all g, and thus no class is given extra weight for the given sign pattern, set selection, and level. If $P_{g|t}$ is nonzero and $P_{j|t} = 0$ for all other classes, then $r(\ell, i, t, g) = G - 1$ and $r(\ell, i, t, j) = 0$ for all $j \neq g$, so that class g receives the largest weight. It is certainly possible that a large number of the sign pattern indices t will have $P_{g|t} = 0$ for all groups (i.e., not all binary sign patterns are observed from the training data), in which case $r(\ell, i, t, g) = 0$.

Remark 1 *Note that in practice the membership index value need not be stored for all 2^ℓ possible sign pattern indices, but rather only for the unique sign patterns that are actually observed by the training data. In this case, the unique sign patterns at each level ℓ and iteration i must be input to the classification phase of the algorithm (Algorithm 2).*

Algorithm 1 Training

input: training labels b, number of classes G, number of levels L, binary training data Q (or raw training data X and fixed matrix A)
if raw data: Compute $Q = \text{sign}(AX)$
for ℓ from 1 to L, i from 1 to m **do**
  select: Randomly select $\Lambda_{\ell,i} \subset [m]$, $|\Lambda_{\ell,i}| = \ell$
  for t from 0 to $2^\ell - 1$, g from 1 to G **do**
    compute: Compute $r(\ell, i, t, g)$ by (1)
  end for
end for

Once the algorithm has been trained, we can use it to classify new signals. Suppose $x \in \mathbb{R}^n$ is a new signal for which the class is unknown, and we have available the quantized

measurements $q = \text{sign}(Ax)$. Then Algorithm 2 is used for the classification of x into one of the G classes. Notice that the number of levels L, the learned membership index values $r(\ell, i, t, g)$, and the set selections $\Lambda_{\ell,i}$ at each iteration of each level are all available from Algorithm 1. First, the decision vector $\tilde{r}$ is initialized to the zero vector in $\mathbb{R}^G$. Then for each level ℓ and set selection i, the sign pattern, and hence the binary base 2 representation, can be determined using q and $\Lambda_{\ell,i}$. Thus, the corresponding sign pattern index $t^\star = t^\star(\ell, i) \in \{0, 1, 2, \dots\}$ such that $0 \leq t^\star \leq 2^\ell - 1$ is identified. For each class g, $\tilde{r}(g)$ is updated via $\tilde{r}(g) \leftarrow \tilde{r}(g) + r(\ell, i, t^\star, g)$. Finally, after scaling $\tilde{r}$ with respect to the number of levels and measurements, the largest entry of $\tilde{r}$ identifies how the estimated label $\widehat{b}_x$ of x is set. This scaling of course does not actually affect the outcome of classification, we use it simply to ensure the quantity does not become unbounded for large problem sizes. We note here that especially for large m, the bulk of the classification will come from the higher levels (in fact the last level) due to the geometry of the algorithm. However, we choose to write the testing phase using all levels since the lower levels are cheap to compute with, may still contribute to classification accuracy especially for small m, and can be used naturally in other settings such as hierarchical classification and detection (see remarks in Section 5).

Algorithm 2 Classification

input: binary data q, number of classes G, number of levels L, learned parameters $r(\ell, i, t, g)$ and $\Lambda_{\ell,i}$ from Algorithm 1

initialize: $\tilde{r}(g) = 0$ for $g = 1, \dots, G$.
for ℓ from 1 to L, i from 1 to m **do**
  identify: Identify the sign pattern index $t^\star$ using q and $\Lambda_{\ell,i}$
  for g from 1 to G **do**
    update: $\tilde{r}(g) = \tilde{r}(g) + r(\ell, i, t^\star, g)$
  end for
end for
scale: Set $\tilde{r}(g) = \frac{\tilde{r}(g)}{Lm}$ for $g = 1, \dots, G$
classify: $\widehat{b}_x = \text{argmax}_{g \in \{1,\dots,G\}} \tilde{r}(g)$

3. Experimental Results

In this section, we provide experimental results of Algorithms 1 and 2 for synthetically generated datasets, handwritten digit recognition using the MNIST dataset, and facial recognition using the extended YaleB database. We note that for the synthetic data, we typically use Gaussian clouds, but note that since our algorithms use hyperplanes to classify data, the results on these type of datasets would be identical to any with the same radial distribution around the origin. We use Gaussian clouds simply because they are easy to visualize and allow for various geometries. Of course, our methods require no particular structure other than being centered around the origin, which can be done as a pre-processing step (and the framework could clearly be extended to remove this property in future work). The real data like the hand-written digits and faces clearly have more complicated geometries and

are harder to visualize. We include both types of data to fully characterize our method's performance.

We also remark here that we purposefully choose not to compare to other related methods like SVM for several reasons. First, if the data happens to be linearly separable it is clear that SVM will outperform or match our approach since it is designed precisely for such data. In the interesting case when the data is not linearly separable, our method will clearly outperform SVM since SVM will fail. To use SVM in this case, one needs an appropriate kernel, and identifying such a kernel is highly non-trivial without understanding the data's geometry, and precisely what our method avoids having to do.

Unless otherwise specified, the matrix A is taken to have i.i.d. standard Gaussian entries. Also, we assume the data is centered. To ensure this, a pre-processing step on the raw data is performed to account for the fact that the data may not be centered around the origin. That is, given the original training data matrix X, we calculate $\mu = \frac{1}{p}\sum_{i=1}^{p} x_i$. Then for each column x_i of X, we set $x_i \leftarrow x_i - \mu$. The testing data is adjusted similarly by μ. Note that this assumption can be overcome in future work by using *dithers*—that is, hyperplane dither values may be learned so that $Q = \text{sign}(AX + \tau)$, where $\tau \in \mathbb{R}^m$—or even with random dithers, as motivated by quantizer results (Baraniuk et al., 2017; Cambareri et al., 2017).

3.1. Classification of Synthetic Datasets

In our first stylized experiment, we consider three classes of Gaussian clouds in $\mathbb{R}^2$ (i.e., $n = 2$); see Figure 1 for an example training and testing data setup. For each choice of $m \in \{5, 7, 9, 11, 13, 15, 17, 19\}$ and $p \in \{75, 150, 225\}$ with equally sized training data sets for each class (that is, each class is tested with either 25, 50, or 75 training points), we execute Algorithms 1 and 2 with a single level and 30 trials of generating A. We perform classification of 50 test points per group, and report the average correct classification rate (ACCR) over all trials. Note that the ACCR is simply defined as the number of correctly classified testing points divided by the total number of testing points (where the correct class is known either from the generated distribution or the real label for real world data), and then averaged over the trials of generating A. We choose this metric since it captures both false negatives and positives, and since in all experiments we have access to the correct labels. The right plot of Figure 1 shows that $m \geq 15$ results in nearly perfect classification.

Next, we present a suite of experiments where we again construct the classes as Gaussian clouds in $\mathbb{R}^2$, but utilize various types of data geometries. In each case, we set the number of training data points for each class to be 25, 50, and 75. In Figure 2, we have two classes forming a total of six Gaussian clouds, and execute Algorithms 1 and 2 using four levels and $m \in \{10, 30, 50, 70, 90, 110, 130\}$. The classification accuracy increases for larger m, with nearly perfect classification for the largest values of m selected. A similar experiment is shown in Figure 3, where we have two classes forming a total of eight Gaussian clouds, and execute the proposed algorithm using five levels.

In the next two experiments, we display the classification results of Algorithms 1 and 2 when using $m \in \{10, 30, 50, 70, 90\}$ and one through four levels, and see that adding levels can be beneficial for more complicated data geometries. In Figure 4, we have three classes forming a total of eight Gaussian clouds. We see that from both $L = 1$ to $L = 2$ and $L = 2$

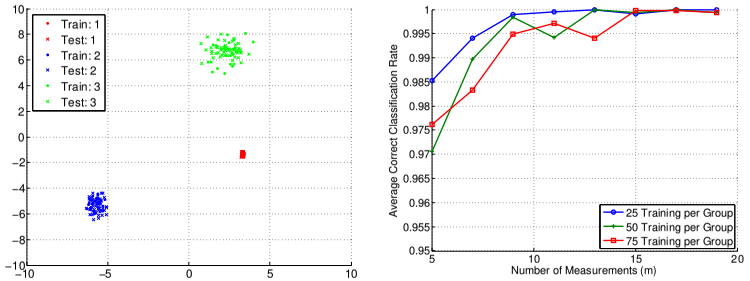


Figure 1: Synthetic classification experiment with three Gaussian clouds ($G = 3$), $L = 1$, $n = 2$, 50 test points per group, and 30 trials of randomly generating A. (Left) Example training and testing data setup. (Right) Average correct classification rate versus m and for the indicated number of training points per class.

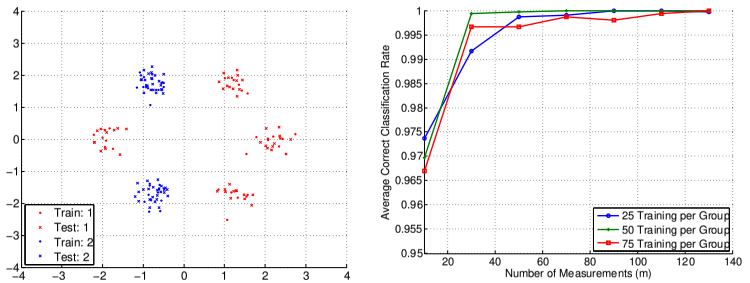


Figure 2: Synthetic classification experiment with six Gaussian clouds and two classes ($G = 2$), $L = 4$, $n = 2$, 50 test points per group, and 30 trials of randomly generating A. (Left) Example training and testing data setup. (Right) Average correct classification rate versus m and for the indicated number of training points per class.

to $L = 3$, there are huge gains in classification accuracy. In Figure 5, we have four classes forming a total of eight Gaussian clouds. Again, from both $L = 1$ to $L = 2$ and $L = 2$ to $L = 3$ we see large improvements in classification accuracy, yet still better classification with $L = 4$. We note here that in this case it also appears that more training data does not improve the performance (and perhaps even slightly decreases accuracy); this is of course unexpected in practice, but we believe this happens here only because of the construction of the Gaussian clouds—more training data leads to more outliers in each cloud, making the sets harder to separate.

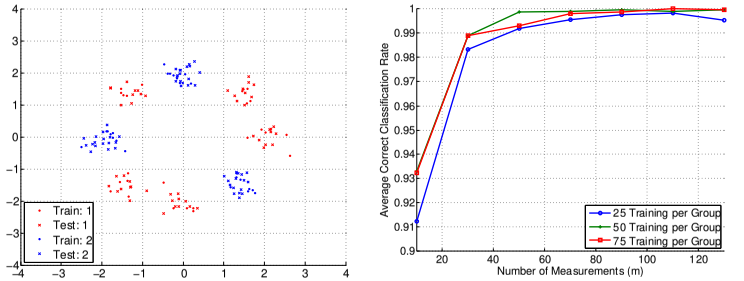
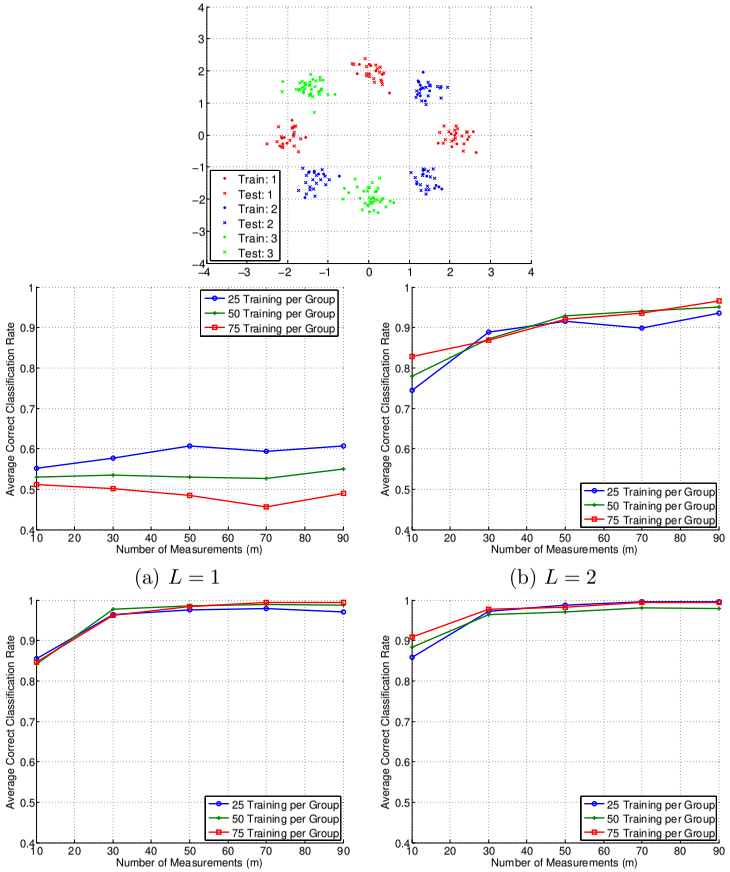


Figure 3: Synthetic classification experiment with eight Gaussian clouds and two classes ($G = 2$), $L = 5$, $n = 2$, 50 test points per group, and 30 trials of randomly generating A. (Left) Example training and testing data setup. (Right) Average correct classification rate versus m and for the indicated number of training points per class.

3.2. Handwritten Digit Classification

In this section, we apply Algorithms 1 and 2 to the MNIST (LeCun, 2018) dataset, which is a benchmark dataset of images of handwritten digits, each with 28×28 pixels. In total, the dataset has $60,000$ training examples and $10,000$ testing examples.

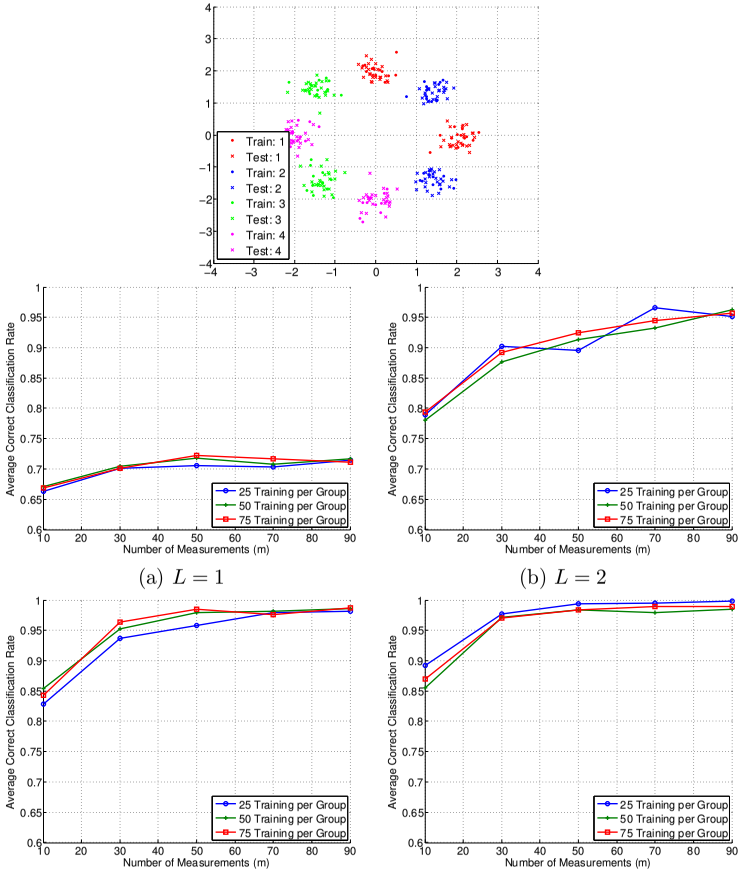
First, we apply Algorithms 1 and 2 when considering only two digit classes. Figure 6 shows the correct classification rate for the digits "0" versus "1". We set $m \in \{10, 30, 50, 70, 90, 110\}$, $p \in \{50, 100, 150\}$ with equally sized training data sets for each class, and classify 50 images per digit class. Notice that the algorithm is performing very well for small m in comparison to $n = 28 \times 28 = 784$ and only a single level. Figure 7 shows the results of a similar setup for the digits "0" and "5". In this experiment, we increased to four levels and achieve classification accuracy around 90% at the high end of m values tested. This indicates that the digits "0" and "5" are more likely to be mixed up than "0" and "1", which is understandable due to the more similar digit shape between "0" and "5". In Figure 7, we include the classification performance when the matrix A is constructed using the two-dimensional Discrete Cosine Transform (DCT) in addition to our typical Gaussian matrix A (note one could similarly use the Discrete Fourier Transform instead of the DCT but that requires re-defining the sign function on complex values). Specifically, to construct A from the $n \times n$ two-dimensional DCT, we select m rows uniformly at random



(a) $L = 1$ (b) $L = 2$

(c) $L = 3$ (d) $L = 4$

Figure 4: Synthetic classification experiment with eight Gaussian clouds and three classes ($G = 3$), $L = 1, \ldots, 4$, $n = 2$, 50 test points per group, and 30 trials of randomly generating A. (Top) Example training and testing data setup. Average correct classification rate versus m and for the indicated number of training points per class for: (middle left) $L = 1$, (middle right) $L = 2$, (bottom left) $L = 3$, (bottom right) $L = 4$.



(a) $L = 1$ (b) $L = 2$

(c) $L = 3$ (d) $L = 4$

Figure 5: Synthetic classification experiment with eight Gaussian clouds and four classes ($G = 4$), $L = 1, \ldots, 4$, $n = 2$, 50 test points per group, and 30 trials of randomly generating A. (Top) Example training and testing data setup. Average correct classification rate versus m and for the indicated number of training points per class for: (middle left) $L = 1$, (middle right) $L = 2$, (bottom left) $L = 3$, (bottom right) $L = 4$.

and then apply a random sign (i.e., multiply by +1 or -1) to the columns. We include these two results to illustrate that there is not much difference when using the DCT and Gaussian constructions of A, though we expect analyzing the DCT case to be more challenging and limit the theoretical analysis in this paper to the Gaussian setting. The advantage of using a structured matrix like the DCT is of course the reduction in computation cost in acquiring the measurements.

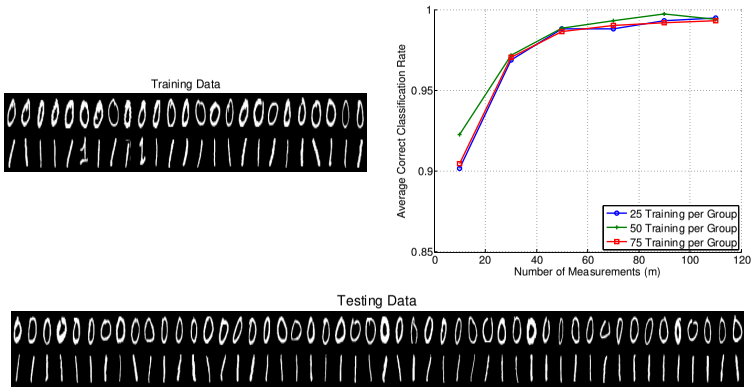


Figure 6: Classification experiment using the handwritten "0" and "1" digit images from the MNIST dataset, $L = 1$, $n = 28 \times 28 = 784$, 50 test points per group, and 30 trials of randomly generating A. (Top left) Training data images when $p = 50$. (Top right) Average correct classification rate versus m and for the indicated number of training points per class. (Bottom) Testing data images.

Next, we apply Algorithms 1 and 2 to the MNIST dataset with all ten digits. We utilize $1,000$, $3,000$, and $5,000$ training points per digit class, and perform classification with 800 test images per class. The classification results using 18 levels and $m \in \{100, 200, 400, 600, 800\}$ are shown in Figure 8, where it can be seen that with $5,000$ training points per class, above 90% classification accuracy is achieved for $m \geq 200$. We also see that larger training sets result in slightly improved classification.

3.3. Facial Recognition

Our last experiment considers facial recognition using the extended YaleB dataset (Cai et al., 2007b,a, 2006; He et al., 2005). This dataset includes 32×32 images of 38 individuals with roughly 64 near-frontal images under different illuminations per individual. We select four individuals from the dataset, and randomly select images with different illuminations to be included in the training and testing sets (note that the same illumination was included for *each* individual in the training and testing data). We execute Algorithms 1 and 2 using four levels with $m \in \{10, 50, 100, 150, 200, 250, 300\}$, $p \in \{20, 40, 60\}$ with equally sized training data sets for each class, and classify 30 images per class. The results are displayed

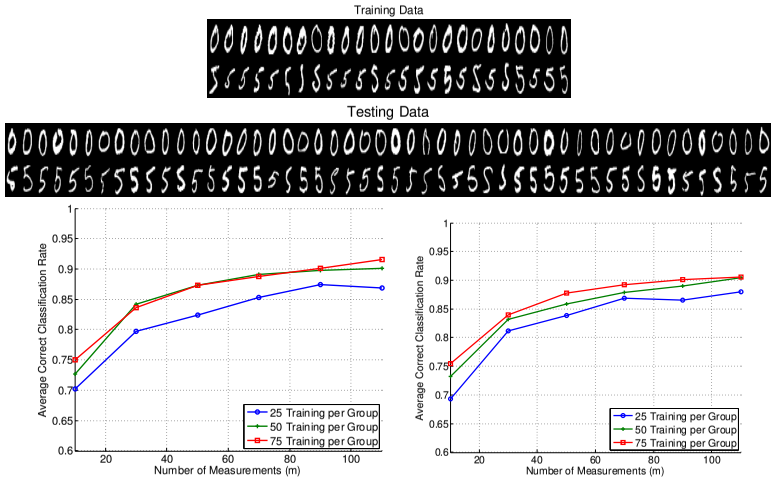


Figure 7: Classification experiment using the handwritten "0" and "5" digit images from the MNIST dataset, $L = 4$, $n = 28 \times 28 = 784$, 50 test points per group, and 30 trials of randomly generating A. (Top) Training data images when $p = 50$. (Middle) Testing data images. Average correct classification rate versus m and for the indicated number of training points per class (bottom left) when using a Gaussian matrix A and (bottom right) when using a DCT matrix A.

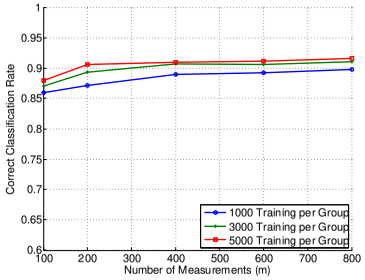


Figure 8: Correct classification rate versus m when using all ten (0-9) handwritten digits from the MNIST dataset, $L = 18$, $n = 28 \times 28 = 784$, 1,000, 3,000, and 5,000 training points per group, 800 test points per group (8,000 total), and a single instance of randomly generating A.

in Figure 9. Above 90% correct classification is achieved for $m \geq 150$ when using the largest training set.

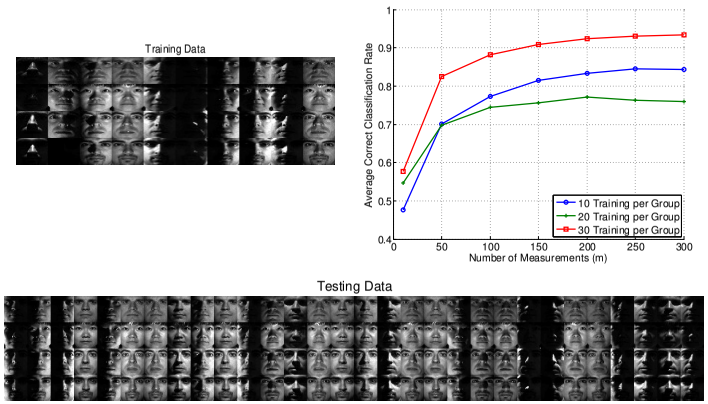


Figure 9: Classification experiment using four individuals from the extended YaleB dataset, $L = 4$, $n = 32 \times 32 = 1024$, 30 test points per group, and 30 trials of randomly generating A. (Top left) Training data images when $p = 20$. (Top right) Average correct classification rate versus m and for the indicated number of training points per class. (Bottom) Testing data images.

4. Theoretical Analysis for a Simple Case

4.1. Main Results

We now provide a theoretical analysis of Algorithms 1 and 2 in which we make a series of simplifying assumptions to make the development more tractable. We focus on the setting where the signals are two-dimensional, belonging to one of two classes, and consider a single level (i.e., $\ell = 1$, $n = 2$, and $G = 2$). Moreover, we assume the true classes G_1 and G_2 to be two disjoint *cones* in $\mathbb{R}^2$ and assume that regions of the same angular measure have the same number (or density) of training points. Of course, the problem of non-uniform densities relates to complicated geometries that may dictate the number of training points required for accurate classification (especially when many levels are needed) and is a great direction for future work. However, we believe analyzing this simpler setup will provide a foundation for a more generalized analysis in future work.

Let A_1 denote the angular measure of G_1, defined by

$$A_1 = \max_{x_1, x_2 \in G_1} \angle(x_1, x_2),$$

where $\angle(x_1, x_2)$ denotes the angle between the vectors x_1 and x_2; define A_2 similarly for G_2. Also, define

$$A_{12} = \min_{x_1 \in G_1, x_2 \in G_2} \angle(x_1, x_2)$$

as the angle between classes G_1 and G_2. Suppose that the test point $x \in G_1$, and that we classify x using m random hyperplanes. For simplicity, we assume that the hyperplanes can intersect the cones, but only intersect *one* cone at a time. This means we are imposing the condition $A_{12} + A_1 + A_2 \leq \pi$. See Figure 10 for a visualization of the setup for the analysis. Notice that A_1 is partitioned into two disjoint pieces, θ_1 and θ_2, where $A_1 = \theta_1 + \theta_2$. The angles θ_1 and θ_2 are determined by the location of x within G_1.

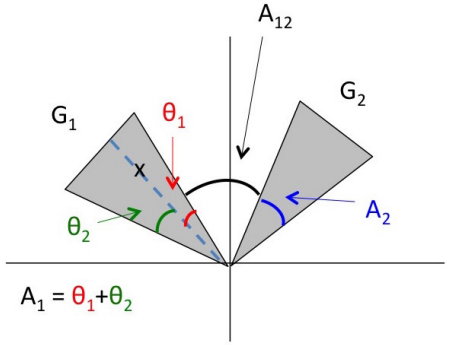


Figure 10: Visualization of the analysis setup for two classes of two dimensions. If a hyperplane intersects the θ_1 region of G_1, then x is not on the same side of the hyperplane as G_2. If a hyperplane intersects the θ_2 region of G_1, then x is on the same side of the hyperplane as G_2. That is, θ_1 and θ_2 are determined by the position of x within G_1, and $\theta_1 + \theta_2 = A_1$.

The membership index parameter (1) is still used; however, now we have angles instead of numbers of training points. That is,

$$r(\ell, i, t, g) = \frac{A_{g|t}}{\sum_{j=1}^{G} A_{j|t}} \frac{\sum_{j=1}^{G} |A_{g|t} - A_{j|t}|}{\sum_{j=1}^{G} A_{j|t}}, \tag{2}$$

where $A_{g|t} = A_{g|t}(\Lambda_{\ell,i})$ denotes the angle of the part of class g with the t-th sign pattern index at the i-th set selection in the ℓ-th level. Throughout, let $t_i^\star$ denote the sign pattern index of the test point x with the i-th hyperplane at the first level, $\ell = 1$; i.e. $t_i^\star = t_{\Lambda_{\ell,i}}^\star$ with the identification $\Lambda_{\ell,i} = \{i\}$ (since $\ell = 1$ implies a single hyperplane is used). Letting $\widehat{b}_x$ denote the classification label for x after running the proposed algorithm, Theorem 2 describes the probability that x gets classified correctly with $\widehat{b}_x = 1$. Note that for simplicity,

in Theorem 2 we assume the classes G_1 and G_2 are of the same size (i.e., $A_1 = A_2$) and the test point x lies in the middle of class G_1 (i.e., $\theta_1 = \theta_2$). These assumptions are for convenience and clarity of presentation only (note that (3) is already quite cumbersome), but the proof follows analogously (albeit without easy simplifications) for the general case; for convenience we leave the computations in Table 1 in general form and do not utilize the assumption $\theta_1 = \theta_2$ until the end of the proof. We first state a technical result in Theorem 2, and include two corollaries below that illustrate its usefulness.

Theorem 2 *Let the classes G_1 and G_2 be two cones in $\mathbb{R}^2$ defined by angular measures A_1 and A_2, respectively, and suppose regions of the same angular measure have the same density of training points. Suppose $A_1 = A_2$, $\theta_1 = \theta_2$, and $A_{12} + A_1 + A_2 \leq \pi$. Then, the probability that a data point $x \in G_1$ gets classified in class G_1 by Algorithms 1 and 2 using a single level and a measurement matrix $A \in \mathbb{R}^{m\times 2}$ with independent standard Gaussian entries is bounded as follows,*

$$\mathbb{P}[\widehat{b}_x = 1] \geq 1 - \sum_{\substack{j+k_{1,1}+k_{1,2}+k_2+k=m,\ k_{1,2}\geq 9(j+k_{1,1})}}^{m}\sum_{k_{1,1}=0}^{m}\sum_{k_{1,2}=0}^{m}\sum_{k_2=0}^{m}\sum_{k=0}^{m} \binom{m}{j,k_{1,1},k_{1,2},k_2,k} \left(\frac{A_{12}}{\pi}\right)^{j} \left(\frac{A_1}{2\pi}\right)^{k_{1,1}+k_{1,2}} \times \left(\frac{A_1}{\pi}\right)^{k_2} \left(\frac{\pi - 2A_1 - A_{12}}{\pi}\right)^{k}. \tag{3}$$

Figure 11 displays the classification probability bound of Theorem 2 compared to the (simulated) true value of $\mathbb{P}[\widehat{b}_x = 1]$. Here, $A_1 = A_2 = 15°$, $\theta_1 = \theta_2 = 7.5°$, and A_{12} and m are varied. Most importantly, notice that in all cases, the classification probability is approaching 1 with increasing m. Also, the result from Theorem 2 behaves similarly as the simulated true probability, especially as m and A_{12} increase.

The following two corollaries provide asymptotic results for situations where $\mathbb{P}[\widehat{b}_x = 1]$ tends to 1 when $m \to \infty$. Corollary 3 provides this result whenever A_{12} is at least as large as both A_1 and $\pi - 2A_1 - A_{12}$, and Corollary 4 provides this result for certain combinations of A_1 and A_{12}. These results of course should match intuition, since as m grows large, our hyperplanes essentially chop up the space into finer and finer wedges. Below, the dependence on the constants on A_1, A_{12} is explicit in the proofs.

Corollary 3 *Consider the setup of Theorem 2. Suppose $A_{12} \geq A_1$ and $2A_{12} \geq \pi - 2A_1$. Then $\mathbb{P}[\widehat{b}_x = 1] \to 1$ as $m \to \infty$. In fact, the probability converges to 1 exponentially, i.e. $\mathbb{P}[\widehat{b}_x = 1] \geq 1 - Ce^{-cm}$ for positive constants c and C that may depend on A_1, A_{12}.*

Corollary 4 *Consider the setup of Theorem 2. Suppose $A_1 + A_{12} > 0.58\pi$ and $A_{12} + \frac{3}{4}A_1 \leq \frac{\pi}{2}$. Then $\mathbb{P}[\widehat{b}_x = 1] \to 1$ as $m \to \infty$. In fact, the probability converges to 1 exponentially, i.e. $\mathbb{P}[\widehat{b}_x = 1] \geq 1 - Ce^{-cm}$ for positive constants c and C that may depend on A_1, A_{12}.*

4.2. Proof of Main Results

4.2.1. Proof of Theorem 2

Proof Using our setup, we have five possibilities for any given hyperplane: (i) the hyperplane completely separates the two classes, i.e., the cones associated with the two classes

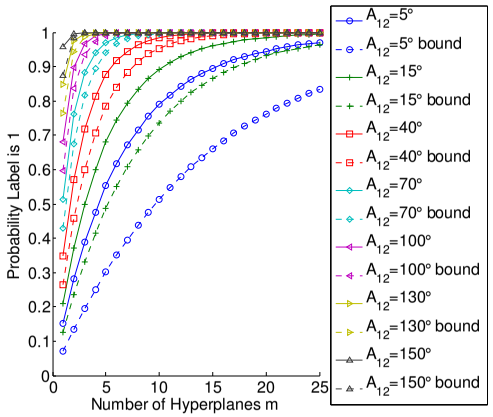


Figure 11: $\mathbb{P}[\widehat{b}_x = 1]$ versus the number of hyperplanes m when A_{12} is varied (see legend), $A_1 = A_2 = 15°$, and $\theta_1 = \theta_2 = 7.5°$. The solid lines indicate the probability (5) with the multinomial probability given by (6) and the conditional probability (9) simulated over 1000 trials of the uniform random variables. The dashed lines indicate the result (3) provided in Theorem 2.

fall on either side of the hyperplane, (ii) the hyperplane completely does not separate the two classes, i.e., the cones fall on the same side of the hyperplane, (iii) the hyperplane cuts through G_2, (iv) the hyperplane cuts through G_1 via θ_1, or (v) the hyperplane cuts through G_1 via θ_2. Using this observation, we can now define the event

$$E(j, k_{1,1}, k_{1,2}, k_2) \tag{4}$$

whereby from among the m total hyperplanes, j hyperplanes separate the cones, $k_{1,1}$ hyperplanes cut G_1 in θ_1, $k_{1,2}$ hyperplanes cut G_1 in θ_2, and k_2 hyperplanes cut G_2. See Table 1 for an easy reference of these quantities. Note that we must distinguish between hyperplanes that cut through θ_1 and those that cut through θ_2; $k_{1,1}$ hyperplanes cut G_1 and land within θ_1 so that x is *not* on the same side of the hyperplane as G_2 whereas $k_{1,2}$ hyperplanes cut G_1 and land within θ_2 so that x *is* on the same side of the hyperplane as G_2. These orientations will affect the computation of the membership index. Using the above definition of (4), we use the law of total probability to get a handle on $\mathbb{P}[\widehat{b}_x = 1]$, the probability that the test point x gets classified correctly, as follows,

$$\begin{aligned}\mathbb{P}[\widehat{b}_x = 1] &= \mathbb{P}\left[\sum_{i=1}^{m} r(\ell, i, t_i^\star, 1) > \sum_{i=1}^{m} r(\ell, i, t_i^\star, 2)\right] \\ &= \sum_{\substack{j,k_{1,1},k_{1,2},k_2 \\ j+k_{1,1}+k_{1,2}+k_2\leq m}} \mathbb{P}\left[\sum_{i=1}^{m} r(\ell, i, t_i^\star, 1) > \sum_{i=1}^{m} r(\ell, i, t_i^\star, 2)\,|E(j, k_{1,1}, k_{1,2}, k_2)\right]\end{aligned}$$

$$\times \mathbb{P}\left[E(j,k_{1,1},k_{1,2},k_2)\right]. \tag{5}$$

The latter probability in (5) is similar to the probability density of a multinomial random variable:

$$\begin{aligned}&\mathbb{P}\left[E(j,k_{1,1},k_{1,2},k_2)\right]\\ &= \binom{m}{j,k_{1,1},k_{1,2},k_2,m-j-k_{1,1}-k_{1,2}-k_2}\left(\frac{A_{12}}{\pi}\right)^j\left(\frac{\theta_1}{\pi}\right)^{k_{1,1}}\left(\frac{\theta_2}{\pi}\right)^{k_{1,2}}\\ &\quad\times\left(\frac{A_2}{\pi}\right)^{k_2}\left(\frac{\pi-A_1-A_2-A_{12}}{\pi}\right)^{m-j-k_{1,1}-k_{1,2}-k_2},\end{aligned} \tag{6}$$

where $\binom{n}{k_1,k_2,\ldots,k_m} = \frac{n!}{k_1!k_2!\cdots k_m!}$.

To evaluate the conditional probability in (5), we must determine the value of $r(\ell,i,t_i^\star,g)$, for $g=1,2$, given the hyperplane cutting pattern event. Table 1 summarizes the possible cases. In the cases where the hyperplane cuts through either G_1 or G_2, we model the location of the hyperplane within the class by a random variable defined on the interval $[0,1]$, with no assumed distribution. We let u, u', u_h, $u'_h \in [0,1]$ (for an index h) denote independent copies of such random variables.

Hyperplane Case	Number in event (4)	Class g	Value of $r(\ell,i,t_i^\star,g)$ (see (2))
(i) separates	j	1	1
		2	0
(ii) does not separate	$m-j-k_2-k_{1,1}-k_{1,2}$	1	$\frac{A_1\lvert A_1-A_2\rvert}{(A_1+A_2)^2}$
		2	$\frac{A_2\lvert A_1-A_2\rvert}{(A_1+A_2)^2}$
(iii) cuts G_2	k_2	1	$\frac{A_1\lvert A_1-A_2u'\rvert}{(A_1+A_2u')^2}$
		2	$\frac{A_2u'\lvert A_1-A_2u'\rvert}{(A_1+A_2u')^2}$
(iv) cuts G_1, θ_1	$k_{1,1}$	1	1
		2	0
(v) cuts G_1, θ_2	$k_{1,2}$	1	$\frac{(\theta_1+\theta_2u)\lvert\theta_1+\theta_2u-A_2\rvert}{(\theta_1+\theta_2u+A_2)^2}$
		2	$\frac{A_2\lvert\theta_1+\theta_2u-A_2\rvert}{(\theta_1+\theta_2u+A_2)^2}$

Table 1: Summary of (2) when up to one cone can be cut per hyperplane, where u,u' are independent random variables defined over the interval $[0,1]$.

Using the computations given in Table 1 and assuming j hyperplanes separate (i.e. condition (i) described above), $k_{1,1}$ hyperplanes cut G_1 in θ_1 (condition (iv) above), $k_{1,2}$ hyperplanes cut G_1 in θ_2 (condition (v) above), k_2 hyperplanes cut G_2 (condition (iii) above), and $m-j-k_{1,1}-k_{1,2}-k_2$ hyperplanes do not separate (condition (ii) above), we compute the membership index parameters defined in (2) as:

$$\sum_{i=1}^m r(\ell,i,t_i^\star,1) = j + (m-j-k_{1,1}-k_{1,2}-k_2)\frac{A_1|A_1-A_2|}{(A_1+A_2)^2} + k_{1,1}$$

$$\begin{aligned}&+\sum_{h=1}^{k_{1,2}}\frac{(\theta_1+\theta_2u_h)|\theta_1+\theta_2u_h-A_2|}{(\theta_1+\theta_2u_h+A_2)^2}+\sum_{h=1}^{k_2}\frac{A_1|A_1-A_2u'_h|}{(A_1+A_2u'_h)^2}\\ &= j+k_{1,1}+\sum_{h=1}^{k_{1,2}}\frac{(\theta_1+\theta_2u_h)|\theta_1+\theta_2u_h-A_1|}{(\theta_1+\theta_2u_h+A_1)^2}+\sum_{h=1}^{k_2}\frac{A_1|A_1-A_1u'_h|}{(A_1+A_1u'_h)^2}\end{aligned} \tag{7}$$

and

$$\begin{aligned}\sum_{i=1}^m r(\ell,i,t_i^\star,2) &= (m-j-k_{1,1}-k_{1,2}-k_2)\frac{A_2|A_1-A_2|}{(A_1+A_2)^2}\\ &\quad+\sum_{h=1}^{k_{1,2}}\frac{A_2|\theta_1+\theta_2u_h-A_2|}{(\theta_1+\theta_2u_h+A_2)^2}+\sum_{h=1}^{k_2}\frac{A_2u'_h|A_1-A_2u'_h|}{(A_1+A_2u'_h)^2}\\ &= \sum_{h=1}^{k_{1,2}}\frac{A_1|\theta_1+\theta_2u_h-A_1|}{(\theta_1+\theta_2u_h+A_1)^2}+\sum_{h=1}^{k_2}\frac{A_1u'_h|A_1-A_1u'_h|}{(A_1+A_1u'_h)^2},\end{aligned} \tag{8}$$

where in both cases we have simplified using the assumption $A_1=A_2$. Thus, the conditional probability in (5), can be expressed as:

$$\mathbb{P}\left[j+k_{1,1}+\sum_{h=1}^{k_{1,2}}\frac{|\theta_1+\theta_2u_h-A_1|(\theta_1+\theta_2u_h-A_1)}{(\theta_1+\theta_2u_h+A_1)^2}+\sum_{h=1}^{k_2}\frac{|A_1-A_1u'_h|(A_1-A_1u'_h)}{(A_1+A_1u'_h)^2}>0\right], \tag{9}$$

where it is implied that this probably is conditioned on the hyperplane configuration as in (5). Once the probability (9) is known, we can calculate the full classification probability (5).

Since by assumption, $\theta_1+\theta_2=A_1$, we have $\theta_1+\theta_2u-A_1\le 0$ and $A_1-A_1u'\ge 0$. Thus, (9) simplifies to

$$\mathbb{P}\left[j+k_{1,1}-\sum_{h=1}^{k_{1,2}}\frac{(\theta_1+\theta_2u_h-A_1)^2}{(\theta_1+\theta_2u_h+A_1)^2}+\sum_{h=1}^{k_2}\frac{(A_1-A_1u'_h)^2}{(A_1+A_1u'_h)^2}>0\right] \;\ge\; \mathbb{P}[\gamma>\beta] \tag{10}$$

where

$$\beta = k_{1,2}\left(\frac{\theta_2}{A_1+\theta_1}\right)^2 \quad\text{and}\quad \gamma = j+k_{1,1}.$$

To obtain the inequality in (10), we used the fact that

$$j+k_{1,1}-\sum_{h=1}^{k_{1,2}}\frac{(\theta_1-A_1)^2}{(\theta_1+\theta_2u_h+A_1)^2}+\sum_{h=1}^{k_2}\frac{(A_1-A_1u'_h)^2}{(A_1+A_1u'_h)^2}\ge j+k_{1,1}-\sum_{h=1}^{k_{1,2}}\frac{(\theta_1-A_1)^2}{(\theta_1+A_1)^2}+0=\gamma-\beta.$$

By conditioning on $\gamma>\beta$, the probability of interest (5) reduces to (note the bounds on the summation indices):

$$\mathbb{P}[\widehat{b}_x=1] = \sum_{\substack{j,k_{1,1},k_{1,2},k_2\\ j+k_{1,1}+k_{1,2}+k_2\le m}}\mathbb{P}\left[\sum_{i=1}^m r(\ell,i,t_i^\star,1)>\sum_{i=1}^m r(\ell,i,t_i^\star,2)\;|E(j,k_{1,1},k_{1,2},k_2)\right]$$

$$\times \mathbb{P}\left[E(j,k_{1,1},k_{1,2},k_2)\right] \tag{11}$$

$$\geq \sum_{\substack{j,k_{1,1},k_{1,2},k_2 \\ j+k_{1,1}+k_{1,2}+k_2\leq m, \\ \beta-\gamma<0}} \binom{m}{j,k_{1,1},k_{1,2},k_2,m-j-k_{1,1}-k_{1,2}-k_2} \left(\frac{A_{12}}{\pi}\right)^j \left(\frac{\theta_1}{\pi}\right)^{k_{1,1}} \times \left(\frac{\theta_2}{\pi}\right)^{k_{1,2}} \left(\frac{A_2}{\pi}\right)^{k_2} \left(\frac{\pi-A_1-A_2-A_{12}}{\pi}\right)^{m-j-k_{1,1}-k_{1,2}-k_2}. \tag{12}$$

The condition $\beta-\gamma<0$ is equivalent to $k_{1,2}(\frac{\theta_2}{A_1+\theta_1})^2-(j+k_{1,1})<0$, which implies $k_{1,2}(\frac{\theta_2}{A_1+\theta_1})^2<j+k_{1,1}$. Assuming $\theta_1=\theta_2$ simplifies this condition to depend *only* on the hyperplane configuration (and not A_1, θ_1, and θ_2) since $\frac{\theta_2}{A_1+\theta_1}=\frac{\theta_2}{3\theta_2}=\frac{1}{3}$. Thus, the condition $\beta-\gamma<0$ reduces to the condition $k_{1,2}<9(j+k_{1,1})$ and (12) then simplifies to

$$\sum_{\substack{j+k_{1,1}+k_{1,2}+k_2\leq m, \\ k_{1,2}<9(j+k_{1,1})}} \binom{m}{j,k_{1,1},k_{1,2},k_2,m-j-k_{1,1}-k_{1,2}-k_2} \left(\frac{A_{12}}{\pi}\right)^j \left(\frac{\theta_1}{\pi}\right)^{k_{1,1}+k_{1,2}} \times \left(\frac{A_2}{\pi}\right)^{k_2} \left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^{m-j-k_{1,1}-k_{1,2}-k_2} \tag{13}$$

$$= \sum_{\substack{j+k_{1,1}+k_{1,2}+k_2+k=m, \\ k_{1,2}<9(j+k_{1,1})}} \binom{m}{j,k_{1,1},k_{1,2},k_2,k} \left(\frac{A_{12}}{\pi}\right)^j \left(\frac{\theta_1}{\pi}\right)^{k_{1,1}+k_{1,2}} \left(\frac{A_2}{\pi}\right)^{k_2} \left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^{k}, \tag{14}$$

$$= \sum_{\substack{j+k_{1,1}+k_{1,2}+k_2+k=m, \\ k_{1,2}<9(j+k_{1,1})}} \binom{m}{j,k_{1,1},k_{1,2},k_2,k} \left(\frac{A_{12}}{\pi}\right)^j \left(\frac{A_1}{2\pi}\right)^{k_{1,1}+k_{1,2}} \left(\frac{A_1}{\pi}\right)^{k_2} \left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^{k}, \tag{15}$$

where we have introduced k to denote the number of hyperplanes that do not separate nor cut through either of the groups, and simplified using the assumptions that $\theta_1=\frac{A_1}{2}$ and $A_1=A_2$.

Note that if we did not have the condition $k_{1,2}<9(j+k_{1,1})$ in the sum (15) (that is, if we summed over all terms), the quantity would sum to 1 (this can easily be seen by the Multinomial Theorem). Finally, this means (15) is equivalent to (3), thereby completing the proof. ■

4.2.2. Proof of Corollary 3

Proof We can bound (3) from below by bounding the excluded terms in the sum (i.e., those that satisfy $k_{1,2}\geq 9(j+k_{1,1})$) from above. One approach to this would be to count the number of terms satisfying $k_{1,2}\geq 9(j+k_{1,1})$ and bound them by their maximum. Using basic combinatorics (see the appendix, Section A.1), that the number of terms satisfying $k_{1,2}\geq 9(j+k_{1,1})$ is given by

$$W_1=\frac{1}{12}\left(\left\lfloor\frac{m}{10}\right\rfloor+1\right)\left(\left\lfloor\frac{m}{10}\right\rfloor+2\right)\left(150\left\lfloor\frac{m}{10}\right\rfloor^2-10(4m+1)\left\lfloor\frac{m}{10}\right\rfloor+3(m^2+3m+2)\right)\sim m^4. \tag{16}$$

Then, the quantity (3) can be bounded below by

$$1-W_1\max\left(\binom{m}{j,k_{1,1},k_{1,2},k_2,k}\left(\frac{A_{12}}{\pi}\right)^j\left(\frac{A_1}{2\pi}\right)^{k_{1,1}+k_{1,2}}\left(\frac{A_1}{\pi}\right)^{k_2}\left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^k\right)=$$
$$1-W_1\max\left(\binom{m}{j,k_{1,1},k_{1,2},k_2,k}\left(\frac{1}{2}\right)^{k_{1,1}+k_{1,2}}\left(\frac{A_{12}}{\pi}\right)^j\left(\frac{A_1}{\pi}\right)^{k_{1,1}+k_{1,2}+k_2}\left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^k\right), \tag{17}$$

where the maximum is taken over all $j,k_{1,1},k_{1,2},k_2,k=0,\ldots,m$ such that $k_{1,2}\geq 9(j+k_{1,1})$. Ignoring the constraint $k_{1,2}\geq 9(j+k_{1,1})$, we can upper bound the multinomial coefficient using the trivial upper bound of 5^m:

$$\binom{m}{j,k_{1,1},k_{1,2},k_2,k}\leq 5^m. \tag{18}$$

Since we are assuming A_{12} is larger than A_1 and $\pi-2A_1-A_{12}$ (from the assumption that $2A_{12}\geq\pi-2A_1$), the strategy is to take j to be as large as possible while satisfying $k_{1,2}\geq 9j$ and $j+k_{1,2}=m$. Since $k_{1,2}\geq 9j$, we have $j+9j\leq m$ which implies $j\leq\frac{m}{10}$. So, we take $j=\frac{m}{10}$, $k_{1,2}=\frac{9m}{10}$, and $k_{1,1}=k_2=k=0$. Then

$$\left(\frac{1}{2}\right)^{k_{1,1}+k_{1,2}}\left(\frac{A_{12}}{\pi}\right)^j\left(\frac{A_1}{\pi}\right)^{k_{1,1}+k_{1,2}+k_2}\left(\frac{\pi-2A_1-A_{12}}{\pi}\right)^k \tag{19}$$
$$\leq\left(\frac{1}{2}\right)^{9m/10}\left(\frac{A_{12}}{\pi}\right)^{m/10}\left(\frac{A_1}{\pi}\right)^{9m/10}$$
$$=\left(\frac{1}{2^9}\frac{A_{12}}{\pi}\left(\frac{A_1}{\pi}\right)^9\right)^{m/10}. \tag{20}$$

Combining (17) with the bounds given in (18) and (20), we have

$$\geq 1-W_1 5^m\left(\frac{1}{2^9}\frac{A_{12}}{\pi}\left(\frac{A_1}{\pi}\right)^9\right)^{m/10}$$
$$\sim 1-m^4 5^m\left(\frac{1}{2^9}\frac{A_{12}}{\pi}\left(\frac{A_1}{\pi}\right)^9\right)^{m/10}$$
$$=1-m^2\left(5^{10}\frac{1}{2^9}\frac{A_{12}}{\pi}\left(\frac{A_1}{\pi}\right)^9\right)^{m/10}. \tag{21}$$

For the above to tend to 1 as $m \to \infty$, we need $\frac{5^{10}}{2^9}\frac{A_{12}}{\pi}\left(\frac{A_1}{\pi}\right)^9 < 1$. This is equivalent to $A_{12}\left(\frac{A_1}{2}\right)^9 < \frac{\pi^{10}}{5^{10}}$, which implies $A_{12}\theta_1^9 < \left(\frac{\pi}{5}\right)^{10} = \frac{\pi}{5}\left(\frac{\pi}{5}\right)^9$. Note that if $\theta_1 = \frac{\pi}{5}$, then $A_1 = A_2 = 2\theta_1 = \frac{2\pi}{5}$. Then A_{12} could be at most $\frac{\pi}{5}$. But, this can't be because we have assumed $A_{12} \geq A_1$. Thus, we must have $\theta_1 < \frac{\pi}{5}$. In fact, $\theta_1 = \frac{\pi}{6}$ is the largest possible, in which case $A_{12} = A_1 = A_2 = \frac{\pi}{3}$. If $\theta_1 = \frac{\pi}{6}$, then $A_{12}\theta_1^9 < \frac{\pi}{5}\left(\frac{\pi}{5}\right)^9$ becomes $A_{12} < \frac{\pi}{5}\left(\frac{6}{5}\right)^9 \approx 3.24$. Therefore, since we are already assuming $A_{12} + 2A_1 \leq \pi$, this is essentially no further restriction on A_{12}, and the same would be true for all $\theta_1 \leq \frac{\pi}{6}$. This completes the proof. ■

4.2.3. Proof of Corollary 4

Proof Consider (3) and set $j' = j + k_{1,1}$ and $r = k_2 + k$. Then we view (3) as a probability equivalent to

$$1 - \sum_{\substack{j'=0}}^{2m} \sum_{k_{1,2}=0}^{m} \sum_{r=0}^{2m} \binom{m}{k_{1,2}, j', r} \left(\frac{A_{12} + \frac{A_1}{2}}{\pi}\right)^{j'} \left(\frac{A_1}{2\pi}\right)^{k_{1,2}} \left(\frac{\pi - A_1 - A_{12}}{\pi}\right)^{r}. \quad (22)$$
$$j'+k_{1,2}+r=m,\ k_{1,2}\geq 9j'$$

Note that multinomial coefficients are maximized when the parameters all attain the same value. Thus, the multinomial term above is maximized when $k_{1,2}$, j' and r are all as close to one another as possible. Thus, given the additional constraint that $k_{1,2} \geq 9j'$, the multinomial term is maximized when $k_{1,2} = \frac{9m}{19}$, $j' = \frac{m}{19}$, and $r = \frac{9m}{19}$ (possibly with ceilings/floors as necessary if m is not a multiple of 19), (see the appendix, Section A.2, for a quick explanation), which means

$$\binom{m}{k_{1,2}, j', r} \leq \frac{m!}{(\frac{9m}{19})!(\frac{m}{19})!(\frac{9m}{19})!} \quad (23)$$

$$\sim \frac{\sqrt{2\pi m}(\frac{m}{e})^m}{2\pi\frac{9m}{19}(\frac{9m}{19e})^{18m/19}\sqrt{2\pi\frac{m}{19}}(\frac{m}{19e})^{m/19}} \quad (24)$$

$$= \frac{19\sqrt{19}}{18\pi m}\left((\frac{19}{9})^{18/19}19^{1/19}\right)^m$$

$$\approx \frac{19\sqrt{19}}{18\pi m}2.37^m, \quad (25)$$

where (24) follows from Stirling's approximation for the factorial (and we use the notation $\sim$ to denote asymptotic equivalence, i.e. that two quantities have a ratio that tends to 1 as the parameter size grows).

Now assume $A_{12} + \frac{3}{4}A_1 \leq \frac{\pi}{2}$, which implies $\pi - A_1 - A_{12} \geq A_{12} + \frac{A_1}{2}$. Note also that $\pi - A_1 - A_{12} \geq A_1$ since it is assumed that $\pi - 2A_1 - A_{12} \geq 0$. Therefore, we can lower bound (22) by

$$1 - W_2\frac{19\sqrt{19}}{18\pi m}2.37^m\left(\frac{\pi - A_1 - A_{12}}{\pi}\right)^m, \quad (26)$$

where W_2 is the number of terms in the summation in (22), and is given by

$$W_2 = \frac{1}{6}\left(\left\lfloor\frac{m}{10}\right\rfloor + 1\right)\left(100\left\lfloor\frac{m}{10}\right\rfloor^2 + (5 - 30m)\left\lfloor\frac{m}{10}\right\rfloor + 3(m^2 + 3m + 2)\right) \sim m^3. \quad (27)$$

Thus, (26) goes to 1 as $m \to \infty$ when $2.37\left(\frac{\pi - A_1 - A_{12}}{\pi}\right) < 1$, which holds if $A_1 + A_{12} > 0.58\pi$. ■

5. Discussion and Conclusion

In this work, we have presented a supervised classification algorithm that operates on binary, or one-bit, data. Along with encouraging numerical experiments, we have also included a theoretical analysis for a simple case. We believe our framework and analysis approach is relevant to analyzing similar, multi-level-type algorithms. Future directions of this work include the use of dithers for more complicated data geometries, identifying settings where real-valued measurements may be worth the additional complexity, analyzing geometries with non-uniform densities of data, as well as a generalized theory for high dimensional data belonging to many classes and utilizing multiple levels within the algorithm. In addition, we believe the framework will extend nicely into other applications such as hierarchical clustering and classification as well as detection problems. In particular, the membership function scores themselves can provide information about the classes and/or data points that can then be utilized for detection, structured classification, false negative rates, and so on. We believe this framework will naturally extend to these types of settings and provide both simplistic algorithmic approaches as well as the ability for mathematical rigor.

Acknowledgments

DN acknowledges support from the Alfred P. Sloan Foundation, NSF CAREER DMS #1348721 and NSF BIGDATA DMS #1740325. RS acknowledges support from the NSF under DMS-1517204. The authors would like to thank the reviewers for their suggestions, questions, and comments which significantly improved the manuscript.

Appendix A. Elementary Computations

A.1. Derivation of (16)

Suppose we have M objects that must be divided into 5 boxes (for us, the boxes are the 5 different types of hyperplanes). Let n_i denote the number of objects put into box i. Recall that in general, M objects can be divided into k boxes $\binom{M+k-1}{k-1}$ ways.

How many arrangements satisfy $n_1 \geq 9(n_2+n_3)$? To simplify, let n denote the total number of objects in boxes 2 and 3 (that is, $n = n_2 + n_3$). Then, we want to know how many arrangements satisfy $n_1 \geq 9n$?

If $n = 0$, then $n_1 \geq 9n$ is satisfied no matter how many objects are in box 1. So, this reduces to the number of ways to arrange M objects into 3 boxes, which is given by $\binom{M+2}{2}$.

Suppose $n = 1$. For $n_1 \geq 9n$ to be true, we must at least reserve 9 objects in box 1. Then $M-10$ objects remain to be placed in 3 boxes, which can be done in $\binom{(M-10)+2}{2}$ ways. But, there are 2 ways for $n = 1$, either $n_2 = 1$ or $n_3 = 1$, so we must multiply this by 2. Thus, $\binom{(M-10)+2}{2} \times 2$ arrangements satisfy $n_1 \geq 9n$.

Continuing in this way, in general for a given n, there are $\binom{M-10n+2}{2} \times (n+1)$ arrangements that satisfy $n_1 \geq 9n$. There are $n+1$ ways to arrange the objects in boxes 2 and 3, and $\binom{M-10n+2}{2}$ ways to arrange the remaining objects after $9n$ have been reserved in box 1.

Therefore, the total number of arrangements that satisfy $n_1 \geq 9n$ is given by

$$\sum_{n=0}^{\lfloor \frac{M}{10} \rfloor} \binom{M-10n+2}{2} \times (n+1). \tag{28}$$

To see the upper limit of the sum above, note that we must have $M - 10n + 2 \geq 2$, which means $n \leq \frac{M}{10}$. Since n must be an integer, we take $n \leq \lfloor \frac{M}{10} \rfloor$. After some heavy algebra (i.e. using software!), one can express this sum as:

$$W = \frac{1}{12}\left(\left\lfloor \frac{M}{10} \right\rfloor + 1\right)\left(\left\lfloor \frac{M}{10} \right\rfloor + 2\right)\left(150\left\lfloor \frac{M}{10} \right\rfloor^2 - 10(4M+1)\left\lfloor \frac{M}{10} \right\rfloor + 3(M^2+3M+2)\right) \tag{29}$$

$$\sim M^4. \tag{30}$$

A.2. Derivation of (23)

Suppose we want to maximize (over the choices of a, b, c) a trinomial $\frac{m!}{a!b!c!}$ subject to $a + b + c = m$ and $a > 9b$. Since m is fixed, this is equivalent to choosing a, b, c so as to minimize $a!b!c!$ subject to these constraints. First, fix c and consider optimizing a and b subject to $a + b = m - c =: k$ and $a > 9b$ in order to minimize $a!b!$. For convenience, suppose k is a multiple of 10. We claim the optimal choice is to set $a = 9b$ (i.e. $a = \frac{9}{10}k$ and $b = \frac{1}{10}k$). Write $a = 9b + x$ where x must be some non-negative integer in order to satisfy the constraint. We then wish to compare $(9b)!b!$ to $(9b+x)!(b-x)!$, since the sum of a and b must be fixed. One readily observes that:

$$(9b+x)!(b-x)! = \frac{(9b+x)(9b+x-1)\cdots(9b+1)}{b(b-1)\cdots(b-x+1)} \cdot (9b)!b! \geq \frac{9b \cdot 9b \cdots 9b}{b \cdot b \cdots b} \cdot (9b)!b! = 9^x \cdot (9b)!b!.$$

Thus, we only increase the product $a!b!$ when $a > 9b$, so the optimal choice is when $a = 9b$. This holds for any choice of c. A similar argument shows that optimizing b and c subject to $9b + b + c = m$ to minimize $(9b)!b!c!$ results in the choice that $c = 9b$. Therefore, one desires that $a = c = 9b$ and $a + b + c = m$, which means $a = c = \frac{9}{19}m$ and $b = \frac{1}{19}m$.

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-lindenstrauss with binary coins. *J. Comput. Syst. Sci.*, 66(4):671–687, 2003.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proc. 38th annual ACM Symposium on Theory of computing*, pages 557–563. ACM, 2006.

Pervez M Aziz, Henrik V Sorensen, and J Vn der Spiegel. An overview of sigma-delta converters. *IEEE Signal Proc. Mag.*, 13(1):61–84, 1996.

Richard Baraniuk, Mark Davenport, Ronald DeVore, and Michael Wakin. The Johnson-Lindenstrauss lemma meets compressed sensing. *preprint*, 2006.

Richard Baraniuk, Simon Foucart, Deanna Needell, Yaniv Plan, and Mary Wootters. Exponential decay of reconstruction error from binary measurements of sparse signals. *IEEE Trans. Info. Theory*, 63(6):3368–3385, 2017.

Petros Boufounos and Richard Baraniuk. 1-bit compressive sensing. In *Proc. IEEE Conf. Inform. Science and Systems (CISS)*, Princeton, NJ, March 2008.

Peter S. Burge, Max R. van Daalen, Barry J. P. Rising, and John S. Shawe-Taylor. Pulsed neural networks. In *Stochastic Bit-stream Neural Networks*, pages 337–352. MIT Press, Cambridge, MA, 1999.

Deng Cai, Xiaofei He, Jiawei Han, and Hong-Jiang Zhang. Orthogonal Laplacianfaces for face recognition. *IEEE Trans. Image Proc.*, 15(11):3608–3614, 2006.

Deng Cai, Xiaofei He, and Jiawei Han. Spectral regression for efficient regularized subspace learning. In *Proc. Int. Conf. Computer Vision (ICCV'07)*, 2007a.

Deng Cai, Xiaofei He, Yuxiao Hu, Jiawei Han, and Thomas Huang. Learning a spatially smooth subspace for face recognition. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR'07)*, 2007b.

Robert Calderbank, Sina Jafarpour, and Robert Schapire. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. *preprint*, 2009.

Valerio Cambareri, Chunlei Xu, and Laurent Jacques. The rare eclipse problem on tiles: Quantised embeddings of disjoint convex sets. *arXiv preprint arXiv:1702.04664*, 2017.

Emmanuel Candès, Justin Romberg, and Terrence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Trans. Inform. Theory*, 52(2):489–509, 2006a.

Emmanuel Candès, Justin Romberg, and Terrence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Comm. Pure Appl. Math.*, 59(8):1207–1223, 2006b.

James C Candy and Gabor C Temes. *Oversampling delta-sigma data converters: theory, design, and simulation*. University of Texas Press, 1962.

Anna Choromanska, Krzysztof Choromanski, Mariusz Bojarski, Tony Jebara, Sanjiv Kumar, and Yann LeCun. Binary embeddings with structured hashed projections. In *Proc. 33rd International Conference on Machine Learning*, pages 344–353, 2016.

Nello Christianini and John Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, Cambridge, England, 2000.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*, pages 3123–3131, 2015.

Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of johnson and lindenstrauss. *Random Structures & Algorithms*, 22(1):60–65, 2003.

Mark A Davenport, Petros T Boufounos, Michael B Wakin, and Richard G Baraniuk. Signal processing with compressive measurements. *IEEE J. Sel. Topics Signal Process.*, 4(2): 445–460, 2010.

Sjoerd Dirksen and Alexander Stollenwerk. Fast binary embeddings with gaussian circulant matrices: improved bounds. *arXiv preprint arXiv:1608.06498*, 2016.

D. Donoho. Compressed sensing. *IEEE Trans. Inform. Theory*, 52(4):1289–1306, 2006.

Jun Fang, Yanning Shen, Hongbin Li, and Zhi Ren. Sparse signal recovery from one-bit quantized data: An iterative reweighted algorithm. *Signal Processing*, 102:201–206, 2014.

Raja Giryes, Guillermo Sapiro, and Alexander M Bronstein. Deep neural networks with random gaussian weights: a universal classification strategy? *IEEE Trans. Signal Process.*, 64(13):3444–3457, 2016.

Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Machine Intell.*, 35(12):2916–2929, 2013.

Sivakant Gopi, Praneeth Netrapalli, Prateek Jain, and Aditya V Nori. One-bit compressed sensing: Provable support and vector recovery. In *ICML (3)*, pages 154–162, 2013.

Ankit Gupta, Robert Nowak, and Benjamin Recht. Sample complexity for 1-bit compressed sensing and sparse classification. In *Proc. IEEE Int. Symp. Inform. Theory (ISIT)*, pages 1553–1557. IEEE, 2010.

Jurgen Hahn, Simon Rosenkranz, and Abdelhak M Zoubir. Adaptive compressed classification for hyperspectral imagery. In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Processing (ICASSP)*, pages 1020–1024. IEEE, 2014.

Xiaofei He, Shuicheng Yan, Yuxiao Hu, Partha Niyogi, and Hong-Jiang Zhang. Face recognition using Laplacianfaces. *IEEE Trans. Pattern Anal. Machine Intell.*, 27(3):328–340, 2005.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Syst. Appl.*, 13(4):18–28, 1998.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. Quantized neural networks: Training neural networks with low precision weights and activations. *arXiv preprint arXiv:1609.07061*, 2016.

Blake Hunter, Thomas Strohmer, Theodore E Simos, George Psihoyios, and Ch Tsitouras. Compressive spectral clustering. In *AIP Conference Proceedings*, volume 1281, pages 1720–1722. AIP, 2010.

Laurent Jacques, Kévin Degraux, and Christophe De Vleeschouwer. Quantized iterative hard thresholding: Bridging 1-bit and high-resolution quantized compressed sensing. *arXiv preprint arXiv:1305.1786*, 2013a.

Laurent Jacques, Jason Laska, Petros Boufounos, and Richard Baraniuk. Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors. *IEEE Trans. Inform. Theory*, 59(4):2082–2102, 2013b.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142, 1998.

William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. In *Proc. Conf. Modern Anal. and Prob.*, New Haven, CT, June 1982.

Minje Kim and Paris Smaragdis. Bitwise neural networks. *arXiv preprint arXiv:1601.06071*, 2016.

Felix Krahmer and Rachel Ward. New and improved Johnson–Lindenstrauss embeddings via the restricted isometry property. *SIAM J. Math. Anal.*, 43(3):1269–1281, 2011.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 1097–1105, 2012.

Jason N Laska, Zaiwen Wen, Wotao Yin, and Richard G Baraniuk. Trust, but verify: Fast and accurate signal recovery from 1-bit compressive measurements. *IEEE Trans. Signal Processing*, 59(11):5289–5301, 2011.

Yann LeCun. The MNIST database of handwritten digits, 2018. `http://yann.lecun.com/exdb/mnist/`.

Zhouhan Lin, Matthieu Courbariaux, Roland Memisevic, and Yoshua Bengio. Neural networks with few multiplications. *arXiv preprint arXiv:1510.03009*, 2015.

Michele Marchesi, Gianni Orlandi, Francesco Piazza, and Aurelio Uncini. Fast neural networks without multipliers. *IEEE Trans. Neural Networks*, 4(1):53–62, 1993.

Mustafa Ozuysal, Michael Calonder, Vincent Lepetit, and Pascal Fua. Fast keypoint recognition using random ferns. *IEEE Trans. Pattern Anal. Machine Intell.*, 32(3):448–461, 2010.

Yaniv Plan and Roman Vershynin. One-bit compressed sensing by linear programming. *Commun. Pur. Appl. Math.*, 66(8):1275–1297, 2013a.

Yaniv Plan and Roman Vershynin. Robust 1-bit compressed sensing and sparse logistic regression: A convex programming approach. *IEEE Trans. Inform. Theory*, 59(1):482–494, 2013b.

Yaniv Plan and Roman Vershynin. Dimension reduction by random hyperplane tessellations. *Discrete & Computational Geometry*, 51(2):438–461, 2014.

Ali Rahimi and Benjamin Recht. Weighted sums of random kitchen sinks: Replacing minimization with randomization in learning. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 1313–1320, 2009.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pages 525–542. Springer, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *Int. J. Computer Vision*, 115(3):211–252, 2015.

Patrice Y Simard and Hans Peter Graf. Backpropagation without multiplication. In *Proc. Adv. in Neural Processing Systems (NIPS)*, pages 232–239, 1994.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pages 1–9, 2015.

Ming Yan, Yi Yang, and Stanley Osher. Robust 1-bit compressive sensing using adaptive outlier pursuit. *IEEE Trans. Signal Processing*, 60(7):3868–3875, 2012.

Xinyang Yi, Constantine Caravans, and Eric Price. Binary embedding: Fundamental limits and fast algorithm. In *Int. Conf. on Machine Learning*, pages 2162–2170, 2015.

Felix X. Yu, Sanjiv Kumar, Yunchao Gong, and Shih-Fu Chang. Circulant binary embedding. In *Int. Conf. on machine learning*, volume 6, page 7, 2014.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 7/17; Revised 9/18; Published 10/18

Hinge-Minimax Learner for the Ensemble of Hyperplanes

Dolev Raviv DOLEV.RAVIV@GMAIL.COM
Department of Computer Science
University of haifa
Haifa, 31905, Israel

Tamir Hazan TAMIR.HAZAN@TECHNION.AC.IL
Faculty of Industrial Engineering and Management
Technion - Israel Institute of Technology
Haifa, 32000, Israel

Margarita Osadchy RITA@CS.HAIFA.AC.IL
Department of Computer Science
University of haifa
Haifa, 31905, Israel

Editor: David Sontag

Abstract

In this work we consider non-linear classifiers that comprise intersections of hyperplanes. We learn these classifiers by minimizing the "minimax" bound over the negative training examples and the hinge type loss of the positive training examples. These classifiers fit typical real-life datasets that consist of a small number of positive data points and a large number of negative data points. Such an approach is computationally appealing since the majority of training examples (belonging to the negative class) are represented by the statistics of their distribution, which is used in a single constraint on the empirical risk, as opposed to SVM, in which the number of variables is equal to the size of the training set. We first focus on intersection of K hyperplanes, for which we provide empirical risk bounds. We show that these bounds are dimensionally independent and decay as $K/\sqrt{m}$ for m samples. We then extend the K-hyperplane mixed risk to the latent mixed risk for training a union of C K-hyperplane models, which can form an arbitrary complex, piecewise linear boundaries. We propose efficient algorithms for training the proposed models. Finally, we show how to combine hinge-minimax training with deep architectures and extend it to multi-class settings using transfer learning. The empirical evaluation of the proposed models shows their advantage over the existing methods in a small training labeled data regime.

Keywords: Minimiax, Imbalanced Classification, Intersection of K Hyperplanes, Transfer Learning

1. Introduction

Many real-life binary classification problems involve imbalanced classes, for example object detection in vision and fraud detection in security. In such problems it is easy to collect background data (the negative class), while data representing the target class (the positive class) is rare or hard (expensive) to obtain. The majority of existing classifiers (e.g., SVM, Neural Networks, including deep ones) assume balanced training sets and when trained on imbalanced sets show degraded classification performance or require a long and tedious bootstrapping process of mining negative examples (e.g. Malisiewicz et al. (2011); Girshick et al. (2014)) out of millions.

When there are (infinitely) many training examples, instead of minimizing the average sample loss, it is more computationally appealing to apply minimax setting (Lanckriet et al. (2003); Honorio and Jaakkola (2014)), which upper bounds the expected risk of a classifier assuming only the knowledge of mean and covariance of the data distribution. The "minimax" bound provides an upper bound for *every* distribution with a given mean and covariance. Applying the minimax learning (Lanckriet et al. (2003)) to the negative class (the majority class) allows to avoid bootstrapping procedure and makes learning more efficient, as it replaces loss evaluation on all negative samples with a single "minimax" bound.

Due to the assumption that the positive class is rare, we cannot apply the minimax learning to the positive class, as it completely relies on the mean and covariance of the data. Estimating the covariance matrix in high-dimensional space from a small number of positive training samples is problematic. Alternatively, we can use the hinge loss (Vapnik (2000); Zhang (2002); Bartlett and Mendelson (2003); Bousquet et al. (2004); Kakade et al. (2008)) for the positive class as it is computationally appealing when there are fairly small number of training samples.

We suggest to combine the hinge-like loss for the positive samples with the "minimax" bound applied to the statistics of the negative samples to enjoy the best of both worlds and we call these classifiers *Hinge-Minimax* classifiers.

The idea of combining "minimax" bound for the negative class and svm-like formulation for the positive samples was introduced in Osadchy et al. (2012, 2016) for computing linear and kernel classifiers. Kernel classifiers could be quite slow, as they require evaluating kernel on many support vectors. In this work we focus on more efficient non-linear classifiers – ensembles of hyperplanes. We first consider an intersection of hyperplanes and then extended it to more general ensembles of hyperplanes.

Previous algorithms for intersection of hyperplanes are computationally costly when considering large sets of negative data points (Klivans and Sherstov (2009); Daniely et al. (2014)). To deal with this computational difficulty we use the mixed risk. Namely, we extend the "minimax" bound to deal with intersection of hyperplanes over (infinitely many) negative examples. For the positive samples, we define a K-hyperplane hinge loss. We derive an empirical mixed-risk bound, that uses the Rademacher complexities to bound the risk of the positive class and vector Bernstein's inequalities to bound the risk associated with the negative class. Note that we treat the positive and negative samples differently because of the computational gain such separation provides.

Recently, Honorio and Jaakkola (2014) derived a generalization bound for the minimax setting using PAC-Bayesian approach, which bounds the expected loss with respect to a posterior distribution over all possible classifiers. Our work differs as we use stronger assumptions - that the norm of the data points is bounded by a constant, an assumption that is natural in many applications.[1] Thus we are able to avoid the PAC-Bayesian approach that considers generalization bounds over randomized predictors.

Intersection of positive half-spaces is a convex set. We generalize the mixed risk for a non-convex classifier. We learn an ensemble of K-hyperplane models, that can form arbitrary, piece-wise linear boundaries. We propose a training algorithm that minimizes this risk by simultaneously discovering the convex components in the positive class and building K-hyperplane models to separate

1. Input normalization is a standard procedure, applied for faster learning.

each component from the negative class. The learning is done by alternating between finding the best partition of the data into hidden components and updating the model over this partition. We call this novel classifier the *Latent Hinge Minimax* (LHM) classifier, as it discovers the latent structure in the data and employs the Hinge-Minimax paradigm.

We show that the LHM model has an equivalent Neural Networks (NN) architecture. This allows us 1) to use deep learning features via transfer learning and 2) to extend the proposed model to the multi-class setting. For the multi-class problems, we build one-against-all classifiers for all classes and combine them in a single model by mapping class specific LHM models to a multi-class NN with a matching architecture. We then use the cross-entropy loss to adjust the weights in the resulting *LHM-NN* combination.

We show that using LHM-NN in the transfer learning settings has significant benefits compared to NN (standard settings), in both classification accuracy and training efficiency. The improved accuracy stems from the ability of LHM model to learn from unlabeled data. The fast convergence of the LHM-NN (just a handful of epochs) is due to a very good initialization of the upper layers with class specific LHM classifiers. Note that class specific LHM models can be trained in parallel. Moreover, adding a new class to LHM-NN is fast and easy: train a classifier for the new class, map it to the corresponding LHM-NN architecture and run a very fast fine-tuning. Similarly to Kuzborskij et al. (2013), which considered the transfer learning for the $n+1$ category from a fully trained n-category classifier, we use only a handful of training samples for tuning it. In contrast to Kuzborskij et al. (2013), we do not restrict the new classifier to belong to the span of the previously learned n classifiers. This allows us greater flexibility in adding a new, non-related class to the multi-class model.

We performed empirical evaluation of the proposed models: the K-hyperplane, the LHM, and the multi-class models. In all cases, the proposed models outperformed their counterparts in small and imbalanced training data regime.

The rest of the paper is organized as follows. Section 2 introduces the K hyperplane Hinge-Minimax classifier (KHHM). Specifically, we extend the "minimax" bound for the intersection of K positive half-spaces in Section 2.1.1. Then in Section 2.2, we propose a novel training algorithm for the KHHM classifier. We prove the uniform generalization bounds for the KHHM model in Section 2.3. Section 3 focuses on a non-convex generalization of the KHHM model – LHM classifier. We introduce the latent mixed risk in Section 3.1, the training algorithm in Section 3.2, and we prove a uniform generalization bound for the LHM classifier with a fixed assignment of the positive labeled training set in Section 3.3. Section 4 discusses the mapping of the LHM classifier to a neural network. Section 5 reports the experiments with KHHM, LHM, and LHM-NN classifiers. Section 6 discusses the efficiency of the proposed models and Section 7 concludes the paper.

2. K-Hyperplane Hinge-Minimax Classifier

In the following, for simplicity we assume that for a linear classifier which predicts $y = sign(w^T x)$, $b = 0$ (or absorbed by w).

Definition 1 *Let w_i, $i = 1,..,K$ denote K hyperplanes. Let W be a matrix with w_i as its ith column. We define K-hyperplane classifier $f_W(x)$ as an intersection of these K half-spaces:*

$$f_W(x) = \begin{cases} 1 & \text{if } W^\top x \geq \vec{0} \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

where $\vec{0}$ denotes a vector of zeros.

2.1. Mixed Risk for K-Hyperplane Model

We are interested in a classification problem in which the positive class corresponds to a single concept and the negative class is its complement and we refer to it as a *background*. We assume that the sample of the positive class is relatively small while the negative sample is very large (it can be represented by an unlabeled data as well, thus is easy to collect). Due to the specifics of the problem we propose a mixed risk for the non-linear classifier in eq. 1. We first define its parts in Definitions 2 and 3 and then define the mixed risk in Definition 4.

Definition 2 *Let $(x, y) \sim D$ be a joint distribution of samples $x \in \mathbb{R}^n$ and labels $y \in \{-1, 1\}$. We define the* hinge risk *of $f_W(x)$ as follows,*

$$L_D^H(W) = \mathbb{E}_D \left[\ell(W, x, y)\mathbb{1}[y = 1] + 0 \cdot \mathbb{1}[y = -1]\right] \tag{2}$$

where $\ell(W, x, y) = \max_{j\in\{1,..,K\}}\{\max\{0, 1 - yw_j^\top x\}\}$.

Definition 3 *Under the assumptions of Definition 2, let D_{neg} be a marginal distribution of samples from a ball of radius C over the negative labels with mean μ and covariance matrix Σ. Let $\Omega(\mu, \Sigma)$ be a family of all distributions with mean μ and covariance matrix Σ. We assume that $D_{neg} \in \Omega(\mu, \Sigma)$.*

We define the background risk[2] *of $f_W(x)$ as follows,*

$$L^B_{\mu,\Sigma}(W) = \left[\sup_{Z\in\Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0})\right] \mathbb{1}[y = -1] + 0 \cdot \mathbb{1}[y = 1] \tag{3}$$

We derive the expression for the background risk in the next section.

According to the Definitions 2 and 3, the hinge risk is defined over the samples of the positive class only and the background risk is defined over the distribution of the negative class only. Thus we can sum the two to form the mixed risk over D.

Definition 4 *Under the assumptions of Definitions 2 and 3, we define the mixed risk for the K-hyperplane classifier as:*

$$L_D^{HB}(W) = L_D^H(W) + L^B_{\mu,\Sigma}(W), \tag{4}$$

2.1.1. THE EXPECTED RISK OF THE NEGATIVE CLASS

The extension of Theorem 3.1 from Marshall and Olkin (1960) to a nonzero mean variable (as shown in Marshall and Olkin (1960) Eq. 7.7–7.8) states that for a random vector x with mean M and covariance Γ,

$$\sup_{x\sim(M,\Gamma)} \Pr(x \in S) = \frac{1}{1 + d^2},$$

with $d^2 = \inf_{x\in S}(x - M)^\top \Gamma^{-1}(x - M)$, where S is a given convex set.

The intersection of K hyperplanes is a convex set, thus we can bound the probability of a negative sample falling into the intersection of K hyperplanes using the above result and derive the expression for $L^B_{\mu,\Sigma}(W)$ as shown below in Theorem 1.

2. the name "background" is chosen to emphasize the fact that the negative class is the majority class, while the positive class is rare.

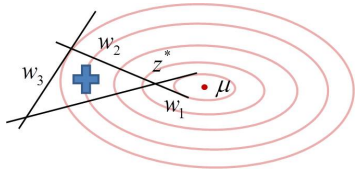


Figure 1: A 2D illustrative example of Theorem 1. z^* is the closest point to the mean of the negative distribution.

Theorem 1 *For any finite number of hyperplanes w_j,*

$$\sup_{Z\in\Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1+d^2}$$

with $d^2 = \mu^\top U(U^\top\Sigma U)^{-1}U^\top\mu$, where U is a subset of columns of W that satisfy $w^\top z^* = 0$, where $z^* = \arg\min_z (z-\mu)^\top\Sigma^{-1}(z-\mu)$.

Before we proceed with the proof, let us consider the following 2D toy example to gain some intuition into Theorem 1 (see Figure 2.1.1). Assume that the positive class lies inside an intersection of three hyperplanes $W = [w_1, w_2, w_3]$ and the negative class is described by the normal distribution with mean μ and covariance Σ. d^2 is a square distance between the mean of the negative distribution and the point on the boundary of the positive region of the classifier, which is closest to μ. In the example, depicted in Figure 2.1.1, the closest point is denoted by z^* and it's an intersection of w_1 and w_2, thus $U = [w_1, w_2]$.

Proof Let $z \sim D_{neg} \in \Omega(\mu,\Sigma)$ be a sample from the negative class. $W^\top z \geq \vec{0}$ defines a convex set, thus we can apply the result due to Marshall and Olkin (1960) to obtain:

$$\sup_{Z\in\Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1+d^2},$$

with $d^2 = \inf_{W^\top z\geq\vec{0}}(z-\mu)^\top\Sigma^{-1}(z-\mu)$, where the supremum is taken over all distributions in $\Omega(\mu,\Sigma)$.

Next, we want to derive a closed-form expression for d^2. We seek the solution for the primal problem

$$\min_z (z-\mu)^\top\Sigma^{-1}(z-\mu)$$

s.t. $w_i^\top z \geq 0$ for $i = 1,..,K$. We construct the Lagrangian:

$$L(z,\lambda_i) = (z-\mu)^\top\Sigma^{-1}(z-\mu) + \sum_i \lambda_i w_i^\top z, \quad \lambda_i \geq 0.$$

The optimality condition:

$$\frac{\partial L}{\partial z} = 2\Sigma^{-1}z - 2\Sigma^{-1}\mu + \sum_i \lambda_i w_i = 0,$$

gives us $z^* = \mu - \frac{1}{2}\sum_i \lambda_i\Sigma w_i$. The Lagrange dual function is as follows,

$$L(z^*,\lambda) = \left(\frac{1}{2}\sum_i \lambda_i\Sigma w_i\right)^\top \Sigma^{-1}\left(\frac{1}{2}\sum_j \lambda_j\Sigma w_j\right) + \sum_i \lambda_i w_i^\top\left(\mu - \frac{1}{2}\sum_j \lambda_j\Sigma w_j\right) \quad (5)$$

The optimality conditions are:

$$\frac{\partial L(z^*,\lambda)}{\partial\lambda_t} = -\frac{1}{2}\sum_i \lambda_i w_t^\top\Sigma w_i + w_t^\top\mu = 0$$

for t such that $\lambda_t > 0$.
The function is optimized at

$$\lambda^* = 2(U\Sigma U)^{-1}U^\top\mu, \quad (6)$$

where U is formed by a subset of columns of W for which $\lambda_t > 0$, and thus $w_t^\top z^* = 0$.

For the last step we substitute the optimal λ, given in eq. 6 into the dual function in eq. 5 and after simple algebraic manipulations we get:

$$d^2 = \max_{\lambda\geq 0}(L(z^*,\lambda^*)) = \mu^\top U(U^\top\Sigma U)^{-1}U^\top\mu$$

■

Given the result of Theorem 1, we can express the background part of the mixed risk of the K-hyperplane classifier as follows,

$$L^B_{\mu,\Sigma}(W) = \sup_{Z\in\Omega(\mu,\Sigma)} Pr_{z\sim Z}(W^\top z \geq \vec{0}) = \frac{1}{1+\mu^\top U(U^\top\Sigma U)^{-1}U^\top\mu}$$

2.2. K-Hyperplane Hinge-Minimax (KHHM) Training

We aim to minimize the mixed risk in eq. 4. To this end, we minimize the empirical risk

$$L^{HB}_S = L^B_S(W) + L^H_S(W) \quad (7)$$

regularized by the sum of L_2 norms of the K hyperplanes: $\frac{C}{2}\sum_i \|w_i\|^2 + L^{HB}_S$. This empirical risk is a non convex and non smooth function, hence a gradient based optimization of it is difficult. However, Osadchy et al. (2016) showed an algorithm for approximating this problem for a single hyperplane $\mathbf{w}$ using the following convex formulation:

$$\begin{aligned} &\underset{\mathbf{w}}{\text{minimize}} \quad \lambda\|\mathbf{w}\|^2 + \sum_i \max\{0, 1-\mathbf{w}^\top x_i\} \\ &\text{subject to} \quad \gamma\sqrt{\mathbf{w}^\top\Sigma\mathbf{w}} + \mathbf{w}^\top\mu \leq 0. \end{aligned} \quad (8)$$

where $\gamma = \sqrt{\frac{1-\delta}{\delta}}$. This formulation minimizes the regularized hinge loss on the positive samples while constraining the probability of "background" data misclassified by the classifier $\mathbf{w}$ to be less than a small threshold δ.

We approximate the solution to the problem in eq. 7 by finding K hyperplanes W, which minimize the regularized hinge loss $L_S^H(W)$ on the positive samples while constraining the probability of "background" data misclassified by the intersection of these hyperplanes to be less than a small threshold ϵ.

We propose an algorithm (Algorithm 1) that computes the hyperplane iteratively, each hyperplane at a time using Osadchy et al. (2016). Note that since the algorithm in Osadchy et al. (2016) constraints the supremum over all distributions with given μ and Σ, it constraints an upper bound on the true distribution of the negative class. However, for a Gaussian distribution the bound is tight.

The Algorithm 1 starts by training K hyperplanes in a greedy manner and then iteratively adjusts each hyperplane to further reduce the loss.

Algorithm 1 KHHM Training

Input: $\{x_i\}, i = 1,..,m^+$ a set of positive examples; $\{z_i\}, i = 1,..,m^-$ a set of negative examples.

Initialization:

Estimate μ and Σ using $\{z_i\}_i^{m^-}$. Find w_1 using Osadchy et al. (2016) with μ, Σ.
for t=2 to k **do**
Estimate μ_t and Σ_t using $\{z_i \mid w_j^\top z_i > 0, j = 1,..,t-1\}$
Find w_t using Osadchy et al. (2016) with μ_t and Σ_t.
end for

Training:

for t=$1, 2, 3...$ **do**
Let P_t be the probability $Pr(W^\top z > 0)$ in iteration t
if $(P_{t-1} - P_t > \epsilon)$
Estimate μ_t and Σ_t using $\{z_i \mid w_j^\top z_i > 0, j = 1,..,K; j \neq t\}$.
Find w_t using Osadchy et al. (2016) with μ_t and Σ_t.
else
Output W (K hyperplanes)
end if
end for

Lemma 2 *Algorithm 1 minimizes the regularized hinge loss $L_S^H(W)$ on the positive samples while keeping $Pr(W^\top z \geq \vec{0}) \leq \epsilon$ (for a small ϵ).*

Proof Let Z denote the distribution of the negative class and $z \sim Z$ denote a sample from this distribution. Let S^t denote the part of negative class that falls inside the intersection of $K-1$ hyperplanes (w_t is not included):

$$S^t = \{z | w_i^\top z \geq 0, \forall i \in \{1,...,K\} \setminus \{t\}\}.$$

In step i, Algorithm 1 finds w_{t_i} that minimizes the hinge loss (which is always positive) of w_{t_i} over positive labels and constrains $Pr(w_{t_i}^\top z \geq 0) | z \in S^{t_i}) \leq \delta$, while keeping the rest of the hyperplanes fixed.

The empirical risk of the intersection of K hyperplanes over positive labels is the maximum over K of hinge losses. Thus, the hinge loss of W is decreased at the iterations, in which the hyperplane with the maximal loss is updated, and it remains unchanged otherwise. Consequently, Algorithm 1 minimizes the hinge loss of W.

We can write

$$Pr(W^\top z \geq 0) = Pr(w_{t_i}^\top z \geq 0) | z \in S^{t_i}) Pr(z \in S^{t_i}) + 0 \cdot Pr(z \notin S^{t_i}).$$

$Pr(z \in S^{t_i}) = a$ which is constant (does not depend on w_{t_i}) and $Pr(w_{t_i}^\top z \geq 0) | z \in S^{t_i}) \leq \delta$. Thus, $Pr(W^\top z \geq 0) \leq a\delta$. Setting $\epsilon = a\delta$ concludes the proof. ■

2.3. Generalization Bound for KHHM Model

In the following we bound the mixed risk of the KHHM classifier by its finite sample. We show that the discrepancy between the risk $L_D^{HB}(W)$ and its empirical estimation $L_S^{HB}(W)$ decays at the rate of $O(K\sqrt{\frac{\log(1/\delta)}{m}})$ where δ is the confidence over the samples of the training data and m is the training data size. The main difficulty in deriving a generalization bound arises from mixing the hinge risk for the positive examples and the background risk for the negative examples. We approach this problem by deriving the uniform generalization bounds separately for the positive and negative classes.

2.3.1. UNIFORM GENERALIZATION BOUND FOR THE EMPIRICAL BACKGROUND RISK

Recall that D_{neg} is the distribution of the negative data points, and μ and Σ are its mean and covariance respectively. Let $\hat{\mu}$ and $\hat{\Sigma}$ be the mean and covariance estimates from the training data points that are associated with negative labels. We bound the background risk by its training sample estimation. The generalization bound is dominated by the discrepancy

$$\Delta = L_{\mu,\Sigma}^B(w) - L_{\hat{\mu},\hat{\Sigma}}^B(w)$$

To provide uniform generalization bound to the background risk, we show that the discrepancy Δ decreases when the size of the training sample increases. Therefore we represent the discrepancy with $\|\hat{\mu} - \mu\|$ and $\|\hat{\Sigma} - \Sigma\|$ that decrease as a function of the training sample.

Let U denote a subset of columns of W that satisfy $w^\top z^* = 0$, where $z^* = \arg\min_z (z-\mu)^\top \Sigma^{-1}(z-\mu)$. We make two additional assumptions on U: First, the number of hyperplanes K_U comprising U is smaller than the dimension of the features and second, the hyperplanes in U are linearly independent. Both assumptions hold in practice. The number of hypeplanes must be small to make the classifier computationally efficient and $K_U \leq K$ (while the dimension of the feature space is usually large). The same reason justifies the independence assumption, as linearly dependent hyperplanes are redundant and do not contribute to the classifier, thus should be removed/avoided.

Using the result of Theorem 1, we can write the discrepancy Δ as follows:

$$\Delta = \frac{1}{1 + \mu^\top U(U^\top \Sigma U)^{-1} U^\top \mu} - \frac{1}{1 + \hat{\mu}^\top U(U^\top \hat{\Sigma} U)^{-1} U^\top \hat{\mu}}$$

By noting that the denominator of both terms is greater than 1 we can upper bound Δ by omitting the denominator. Then,

$$\Delta \leq \hat{\mu}U(U^\top\hat{\Sigma}U)^{-1}U^\top\hat{\mu} - \mu^\top U(U^\top\Sigma U)^{-1}U^\top\mu$$

Next, we denote $A \triangleq U(U^\top\Sigma U)^{-1}U^\top$ and $\hat{A} \triangleq U(U^\top\hat{\Sigma}U)^{-1}U^\top$. By adding and subtracting $\mu^\top A\hat{\mu}$ and rearranging the terms, we obtain

$$\Delta \leq \mu^\top A(\hat{\mu}-\mu) + \hat{\mu}^\top(\hat{A}-A)\hat{\mu} + (\hat{\mu}-\mu)^\top A\hat{\mu}. \tag{9}$$

Denote

$$\Delta_1 \triangleq \mu^\top A(\hat{\mu}-\mu) + (\hat{\mu}-\mu)^\top A\hat{\mu}$$

and

$$\Delta_2 \triangleq \hat{\mu}^\top(\hat{A}-A)\hat{\mu}.$$

Going back to the original notations, we obtain

$$\Delta_1 = \mu^\top U(U^\top\Sigma U)^{-1}U^\top(\hat{\mu}-\mu) + (\hat{\mu}-\mu)^\top U(U^\top\Sigma U)^{-1}U^\top\hat{\mu} \tag{10}$$

$$\Delta_2 = \hat{\mu}^\top(U(U^\top\hat{\Sigma}U)^{-1}U^\top - U(U^\top\Sigma U)^{-1}U^\top)\hat{\mu} \tag{11}$$

In the following, let $\|\cdot\|_F$ denote the Frobenius norm of a matrix (the ℓ_2- norm of matrix vectorized form).

Lemma 3 *Assume $x \sim D_{neg}$ is a distribution over data points x with negative labels such that $\|x\| \leq C$ holds with probability 1. Denote by μ its mean and by Σ its covariance. Let S_1 denote a training sample of size m_1 and let $\hat{\mu} = \frac{1}{m_1}\sum_{x\in S_1} x$ be its sampled mean and $\hat{\Sigma} = \frac{1}{m_1}\sum_{x\in S_1}(x-\hat{\mu})(x-\hat{\mu})^\top$ be its sampled covariance. Define Δ_1 as in eq. 10, where matrix U has K_U linearly independent columns ($K_U \leq n$) . Assume that the minimal eigenvalues of $\Sigma, \hat{\Sigma}$ are lower bounded by $\alpha = \lambda_{min}(\Sigma), \hat{\alpha} = \lambda_{min}(\hat{\Sigma})$, respectively. Then, with probability at least $1-\delta$ over the draws of the training set S_1, the following holds uniformly for all W*

$$\Delta_1 \leq \frac{2C}{\alpha}\sqrt{\frac{32C^4(\log(1/\delta)+1/4)}{m_1}}$$

Proof First, we show the upper bound

$$\|U(U^\top\Sigma U)^{-1}U^\top\| \leq \frac{1}{\alpha} \tag{12}$$

Following the assumption that the columns of U are linearly independent and that their number is smaller than the dimension of the feature space, we can derive that $U = \sum_{i=1}^{K_U} s_i v_i t_i^\top$, where $v_1, ..., v_{K_U}$ are left singular vectors, $t_1, ..., t_{K_U}$ are right singular vectors, and $s_1, ..., s_{K_U}$ are singular values of U which are all non-zero. Let V be $n \times K_U$ matrix of left singular vectors $v_1, ..., v_{K_U}$, T be a $K_U \times K_U$ matrix of right singular vectors $t_1, ..., t_{K_U}$, and S be a $K_U \times K_U$ diagonal matrix of singular values $s_1, ..., s_{K_U}$, then

$$U(U^\top\Sigma U)^{-1}U^\top = VST^\top(TSV^\top\Sigma VST^\top)^{-1}TSV^\top = V(V^\top\Sigma V)^{-1}V^\top \tag{13}$$

$$\begin{aligned}\|V(V^\top\Sigma V)^{-1}V^\top\|^2 &= \max_{\|\beta\|\leq 1}\|V(V^\top\Sigma V)^{-1}V^\top\beta\|^2 \\ &= \max_{\|\beta\|\leq 1}(\beta^\top V(V^\top\Sigma V)^{-1}V^\top V(V^\top\Sigma V)^{-1}V^\top\beta) \\ &= \max_{\|\beta\|\leq 1}((V^\top\beta)^\top(V^\top\Sigma V)^{-2}(V^\top\beta)) \leq \|V^\top\Sigma V)^{-1}\|^2\end{aligned} \tag{14}$$

Since V is a projection operator, the last inequality in Eq. 14 is due to $\|V\beta\| \leq \|\beta\| \leq 1$.

$$\|V^\top\Sigma V)^{-1}\|^2 = \frac{1}{\lambda_{min}(V^\top\Sigma V)^2} \leq \frac{1}{\lambda_{min}(\Sigma)^2} \tag{15}$$

The last inequality in Eq. 15 follows from Poincaré Separation Theorem (Bellman (1970)).

Combining equations 13, 14, and 15, we obtain

$$\|U(U^\top\Sigma U)^{-1}U\| = \|V(V^\top\Sigma V)^{-1}V^\top\| \leq \|(V^\top\Sigma V)^{-1}\| \leq \frac{1}{\alpha}$$

Then applying the above upper bound and the Cauchy-Schwarz inequality, we obtain:

$$\Delta_1 \leq \frac{(\|\hat{\mu}\|+\|\mu\|)\|\hat{\mu}-\mu\|}{\alpha}.$$

Since $\|x\| \leq C$, then $\|\hat{\mu}\| + \|\mu\| \leq 2C$ and we get:

$$\Delta_1 \leq \frac{2C}{\alpha}\|\mu-\hat{\mu}\|.$$

Finally, we use the Bernstein inequality for vectors (cf. Candes and Plan (2011) Theorem 2.6) which states that for vectors $v_1, ..., v_{m_1}$ with $Ev_k = 0$ and $\|v_k\| \leq B$ and $\sum_k E\|v_k\|^2 \leq \sigma^2$ it holds that for all $0 \leq t \leq \sigma^2/B$ that $P[\|\sum_k v_k\| \geq t] \leq \exp(-\frac{t^2}{8\sigma^2}+\frac{1}{4})$. Here $\|v\|^2 = \sum_i v_i^2$.

To fit the Bernstein inequality for vectors to our setting, we set $v_k = \frac{1}{m_1}(x_k - E_{x\sim D_{neg}}x)$ for any $x_k \in S_1$. To see that the conditions of Candes and Plan (2011) Theorem 2.6 hold, we note that since $\|x\| \leq C$ then $\|v_k\| \leq 2C/m_1 \stackrel{def}{=} B$ and therefore $\|v_k\|^2 \leq \frac{4C^2}{m_1^2}$ and $\sum_{k=1}^{m_1} E\|v_k\|^2 \leq \frac{4C^2}{m_1} \stackrel{def}{=} \sigma^2$. Consequently it holds for $0 \leq t \leq 2C$ that

$$P[\|\hat{\mu}-\mu\| \geq t] \leq \exp\left(-\frac{m_1 t^2}{32C^2}+\frac{1}{4}\right).$$

The result follows when setting $\delta = \exp(-\frac{m_1 t^2}{32C^2}+\frac{1}{4})$, or equivalently $t = \sqrt{\frac{32C^2(\log(1/\delta)+1/4)}{m_1}}$. ■

We turn to handle the second term Δ_2 of the discrepancy.

Lemma 4 *Under the conditions of Lemma 3, define Δ_2 as in eq. 11. Then, with probability at least $1-\delta$ over the draws of the training set S_1, the following holds uniformly for all W*

$$\Delta_2 \leq \frac{C^2}{\alpha\hat{\alpha}}\left(2C\|\hat{\mu}-\mu\| + \sqrt{\frac{32C^4(\log(1/\delta)+1/4)}{m_1}}\right)$$

Proof

$$\begin{aligned}\Delta_2 &= \hat{\mu}^\top U\left((U^\top\hat{\Sigma}U)^{-1} - (U^\top\Sigma U)^{-1}\right)U^\top\hat{\mu} \\ &\leq \left|\hat{\mu}^\top U(U^\top\hat{\Sigma}U)^{-1}(U^\top\Sigma U - U^\top\hat{\Sigma}U)(U^\top\Sigma U)^{-1}U^\top\hat{\mu}\right| \\ &= \left|\hat{\mu}^\top U(U^\top\hat{\Sigma}U)^{-1}U^\top(\Sigma-\hat{\Sigma})U(U^\top\Sigma U)^{-1}U^\top\hat{\mu}\right|\end{aligned} \tag{16}$$

Applying the Cauchy-Schwarz inequality and the upper bound in eq. 12, we obtain:

$$\Delta_2 \leq \frac{\|\hat{\mu}\|^2\|\hat{\Sigma}-\Sigma\|}{\alpha\hat{\alpha}}.$$

We now consider $\|\Sigma-\hat{\Sigma}\|$:

$$\begin{aligned}\hat{\Sigma} &= \frac{1}{m_1}\sum_{k=1}^{m_1}(x_k-\hat{\mu})(x_k-\hat{\mu})^\top = \frac{1}{m_1}\sum_{k=1}^{m_1}x_k x_k - \hat{\mu}\hat{\mu}^\top \\ \Sigma &= E_{x\sim D_{neg}}(x-\mu)(x-\mu)^\top = E_{x\sim D_{neg}}xx^\top - \mu\mu^\top \\ \|\Sigma-\hat{\Sigma}\| \leq \|\hat{\Sigma}-\Sigma\|_F &\leq \|\frac{1}{m_1}\sum_{k=1}^{m_1}x_k x_k^\top - E_{x\sim D_{neg}}xx^\top\|_F + \|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F\end{aligned}$$

For the second component in the bound: $\hat{\mu}\hat{\mu}^\top - \mu\mu^\top = \hat{\mu}(\hat{\mu}-\mu)^\top + (\hat{\mu}-\mu)\mu^\top$ therefore $\|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F \leq \|\hat{\mu}(\hat{\mu}-\mu)^\top\|_F + \|(\hat{\mu}-\mu)\mu^\top\|_F \leq 2C\|\hat{\mu}-\mu\|$.

For the first component in the bound, we use the Bernstein inequality for vectors (Candes and Plan (2011) Theorem 2.6) in the same manner it is applied in Lemma 3. We set v_k to be the vectorization of $\frac{1}{m_1}x_k x_k^\top - E_{x\sim D_{neg}} xx^\top$ for any $x_k \in S_1$. To see that the conditions of Candes and Plan (2011) in Theorem 2.6 hold, we note that since $\|x\| \leq C$ then $\|v_k\| \leq \frac{2C^2}{m_1} \stackrel{def}{=} B$ and therefore $\|v_k\|^2 \leq \frac{4C^4}{m_1^2}$ and $\sum_{k=1}^{m_1} E\|v_k\|^2 \leq \frac{4C^4}{m_1} \stackrel{def}{=} \sigma^2$. Consequently it holds for $0 \leq t \leq 2C^2$ that

$$P\Big[\Big\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\Big\| \geq t\Big] \leq \exp(-\frac{m_1 t^2}{32C^4} + \frac{1}{4})$$

The result follows when setting $\delta = \exp(-\frac{m_1 t^2}{32C^4} + \frac{1}{4})$ or equivalently $t = \sqrt{\frac{32C^4(\log(1/\delta)+1/4)}{m_1}}$. ■

The upper bound for Δ relies on the lower bound on the minimal eigenvalue of covariance $\Sigma = E_{x\sim D_{neg}}(x-\mu)(x-\mu)^\top$ and of the sampled covariance $\hat{\Sigma} = \frac{1}{m_1}\sum_{x\in S_1}(x-\hat{\mu})(x-\hat{\mu})^\top$. While the minimal eigenvalue of Σ, say $\alpha = \lambda_{min}(\Sigma)$, can be set to be away from zero, the minimal eigenvalue of the sampled covariance $\hat{\alpha} = \lambda_{min}(\hat{\Sigma})$ is a random variable. We show that $\hat{\alpha}$ is close to α with high probability:

Lemma 5 *Assume the conditions of Lemma 3 hold. Assume that the minimal eigenvalue of $\alpha = \lambda_{min}(\Sigma)$ is positive. Let $\hat{\alpha} = \lambda_{min}(\hat{\Sigma})$ be the minimal eigenvalue of the random covariance matrix $\hat{\Sigma}$. Then $\hat{\alpha} \geq \alpha/2$ with probability at least $1 - 2\exp(-\frac{m_1\alpha^2}{32\cdot 36C^4} + \frac{1}{4})$ over the draws of the training set S_1.*

Proof Using Cauchy-Schwartz inequality we obtain

$$\|U^\top\Sigma U\| - \|U^\top\hat{\Sigma}U\| \leq \Big|\|U^\top\Sigma U\| - \|U^\top\hat{\Sigma}U\|\Big| \leq \|U^\top\Sigma U - U^\top\hat{\Sigma}U\| \leq \|\Sigma-\hat{\Sigma}\|\|U\|^2.$$

Therefore,

$$\|U^\top\hat{\Sigma}U\| \geq \|U^\top\Sigma U\| - \|\Sigma-\hat{\Sigma}\|\|U\|^2.$$

Following Lemma 4 we note that

$$\|\hat{\Sigma}-\Sigma\| \leq \|\hat{\Sigma}-\Sigma\|_F \leq \|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\|_F + \|\hat{\mu}\hat{\mu}^\top - \mu\mu^\top\|_F$$

We use Bernstein inequality for vectors with $t = \alpha/6$ to bound

$$P[\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\| \geq \alpha/6] \leq \exp(-m_1\alpha^2/(36\cdot 32) + 1/4)$$

for any $\alpha/6 \leq 4m_1$. We also use Bernstein inequality for vectors with $t = \alpha/(2C\cdot 3)$ to bound

$$P[\|\hat{\mu}-\mu\| \geq \alpha/(2C\cdot 3)] \leq \exp\left(-\frac{m_1\alpha^2}{36\cdot 32C^4} + \frac{1}{4}\right).$$

Thus with error probability of $2\exp\left(-\frac{m_1\alpha^2}{36\cdot 32C^4} + \frac{1}{4}\right)$ there hold $\|\frac{1}{m_1}\sum_{k=1}^{m_1} x_k x_k^\top - E_{x\sim D_{neg}} xx^\top\| \leq \alpha/6$ and $2C\|\hat{\mu}-\mu\| \leq \alpha/3$. In particular, the sum of both is upper bounded by $\alpha/2$, hence $\|\hat{\Sigma}-\Sigma\| \leq \alpha/2$, resulting in $\|U^\top\hat{\Sigma}U\| \geq (\alpha-\alpha/2)\|U\|^2$. Therefore $\hat{\alpha}\|U\|^2 \geq \alpha/2\|U\|^2$. ■

Bounds on the discrepancy between the expected and the empirical background risks that are uniform for any U guarantee generalization. The above lemmas suggest that the penalty of observing a finite sample space decreases as $1/m_1$. This is summarized in the following theorem.

Theorem 6 *Under the conditions of Lemma 3, for $\alpha \geq \sqrt{\frac{1152(\log(2/\delta)+1/4)}{m_1}}$ with probability at least $1-3\delta$ over m_1 the i.i.d. samples from D_{neg} the following holds uniformly for all W with K linearly independent hyperplanes:*

$$L^B_{\mu,\Sigma}(U) \leq L^B_{\hat{\mu},\hat{\Sigma}}(U) + \Big(\frac{2C^2}{\alpha} + \frac{6C^4}{\alpha^2}\Big)\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}}.$$

Proof Following Eq. 9, we note that $L^B_{\mu,\Sigma}(U) - L^B_{\hat{\mu},\hat{\Sigma}}(U) \leq \Delta_1 + \Delta_2$, for Δ_1, Δ_2 defined in Lemmas 3, 4. The proof applies the bounds on Δ_1, Δ_2, where each of these bounds holds with an error probability at most δ. The proof is concluded by bounding $\hat{\alpha} \geq \alpha/2$ with an error probability at most δ. Formally, with error probability at most 3δ:

$$\begin{aligned}\Delta_1 &\leq \frac{2C^2}{\alpha}\sqrt{\frac{32\log(1/\delta)+1/4}{m_1}}\\ \Delta_2 &\leq \frac{3C^4}{\alpha\hat{\alpha}}\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}} \leq \frac{6C^4}{\alpha^2}\sqrt{\frac{32(\log(1/\delta)+1/4)}{m_1}}\end{aligned}$$

■

The generalization guarantees of the background risk penalize a finite sample size m by $\sqrt{1/m_1}$. It decays to zero when the number of the negative labels in the training sample tends to infinity. In our setting, we assume that $m \approx m_1$, thus we get favorable guarantees with respect to the training size.

2.3.2. UNIFORM GENERALIZATION BOUND FOR THE EMPIRICAL RISK OF THE HINGE-LOSS

We derive a uniform generalization bound for the expected risk over the positive examples using Rademacher complexity. The Rademacher complexity of a bounded set $A \subset R^K$ is

$$R(A) = \frac{1}{m}E_\sigma\left[\max_{a\in A}\sum_{i=1}^{m}\sigma_i a_i\right],$$

while $\sigma_i \in \{-1,+1\}$ are i.i.d. and equally probable random variables.

Let $\mathcal{F}$ denote a family of functions:

$$\mathcal{F} \triangleq \{(x,y) \to \ell(W,x,y) : W = [w_1,...,w_k], w_j \in \mathbb{R}^d, \forall j\}.$$

Let $S = \{(x_1,y_1),...,(x_m,y_m)\}$ be a training sample. Let $\mathcal{F}\circ S$ be the set of all possible evaluations a function $f \in \mathcal{F}$ can achieve on a sample S:

$$\mathcal{F}\circ S = \{f(x_1,y_1),...,f(x_m,y_m)\}.$$

The Rademacher complexity of $\mathcal{F}$ with respect to S is defined as follows:

$$R(\mathcal{F}\circ S) \triangleq \frac{1}{m}\mathbb{E}_{\sigma\sim\{\pm1\}^m}\left[\sup_{f\in\mathcal{F}}\sum_{i=1}^{m}\sigma_i f(x_i,y_i)\right]$$

Since L_D^H is zero for $y=-1$, we consider only the positive subset of S, denoted as, $S_2 \triangleq \{(x_i,1),...(x_{m_2},1)\}$.

Theorem 7 [3] *Consider a $K-$hyperplanes loss function*

$$\ell(W,x,y) = \max_{j\in\{1,...,K\}}\{\max\{0,1-yw_j^\top x\}\}$$

3. Theorem 4 in the ICML'15 version of the paper had a typo. Here we present a corrected version of the theorem with a detailed proof.

for which each hyperplane satisfies $\|w_j\| \leq 1$ *and each data point satisfies* $\|x\| \leq 1$. *Then,*

$$R(\mathcal{F} \circ S_2) \leq \frac{K}{\sqrt{m_2}},$$

where m_2 *is the number of positive examples.*

Proof

$$m_2 R(\mathcal{F} \circ S_2) =$$

$$\mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\max_{\substack{\|w_1\| \leq 1 \\ \ldots \\ \|w_k\| \leq 1}} \sum_{i=1}^{m} \sigma_i \max_{j \in \{1,\ldots,K\}} \{\max(0, 1 - w_j^\top x_i)\} \right] \leq$$

$$\mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\max_{\substack{\|w_1\| \leq 1 \\ \ldots \\ \|w_k\| \leq 1}} \sum_{i=1}^{m} \sigma_i \sum_{j=1}^{K} \max(0, 1 - w_j^\top x_i) \right] \leq$$

$$\sum_{j=1}^{K} \mathbb{E}_{\sigma \sim \{\pm 1\}^m} \left[\sum_{i=1}^{m} \sigma_i \max(0, 1 - w_j^\top x_i) \right] \leq K\sqrt{m_2}$$

The bound follows from the Contraction Lemma (Ledoux and Talagrand, 1991), applied to $\max(0, 1 - w^\top x)$, which is 1-Lipschitz function.
Hence, $R(\mathcal{F} \circ S_2) \leq \frac{K}{\sqrt{m_2}}$. ■

Next we provide the uniform generalization bound for the empirical risk of the maximum over hinge losses.

Theorem 8 *Let* $L_D^H(W) = \mathbb{E}_D\left[\ell(W, x, y)\mathbb{1}[y = 1] + 0 \cdot \mathbb{1}[y = -1]\right]$ *be the expected risk, and let* $L_S^H(W) = \frac{1}{m_2}\sum_{i=1}^{m} \ell(W, x_i, 1)$ *be the empirical risk over a positive label training sample of size* m_2. *Then, for any* $\delta \in (0, 1]$ *with probability at least* $1 - \delta$ *over the i.i.d. sample of size* m_2 *it holds simultaneously for all* $\|w_1\|, ..., \|w_k\| \leq 1$ *that whenever* $\|x\| \leq 1$*:*

$$L_D^H(W) \leq L_S^H(W) + \frac{2K}{\sqrt{m_2}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}$$

Proof By noting that $|\ell(W, x, y)| \leq 2K$ and that a maximum over positive numbers is upper bounded by their sum, the result follows immediately, from Bartlett and Mendelson (2003). ■

3. Latent Hinge-Minimax Classifier

The intersection of K positive half-spaces forms a convex set. For non-convex sets, KHHM will produce many false positives (as show in Figure 2 left). To accommodate classes that form non-convex or disjoint sets, we propose a non-convex classifier, which is an ensemble of KHHM models that we call the Latent Hinge-Minimax (LHM) classifier. Specifically, we define the LHM classifier as a union of intersections of positive half-spaces. We assume that each intersection is composed of K hyperplanes: $W^i = [w_1^i, ..., w_K^i]$ and there are C components in the union (see Figure 2 right). Let $W_{LHM} \triangleq (W^1, \ldots, W^C)$ denote the LHM model.

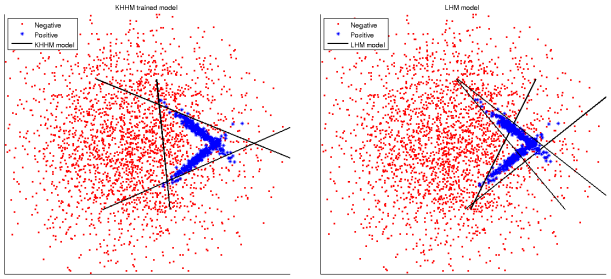


Figure 2: Schematic comparison of KHHM (left) and LHM (right) classifiers on a non-convex positive class. The LHM classifier iteratively discovers a partition of the positive set into convex components and builds KHHM model for each convex component.

3.1. Expected Latent Mixed Risk

Similarly to K-hyperplane model, the latent mixed risk is also composed of the hinge and background parts. However, we extend the risk in Eq. 4 to contain multiple components $Q^i \triangleq \{x \in R^n | {W^i}^T x \geq \vec{0}\}$ and a latent variable $\varphi(x) = i$ ($i \in \{1, \ldots, C\}$) which assigns each positive sample $x \in \mathbb{R}^n$ to one of the C components.

$$L_D(W_{LHM}; \varphi) = L^B_{\mu,\Sigma}(W_{LHM}) + L^H_D(W_{LHM}; \varphi), \tag{17}$$

The background part of the latent mixed risk bounds the probability of the negative class in all components Q^i:

$$L^B_{\mu,\Sigma}(W_{LHM}) = \sum_{i=1}^{C} L^B_{\mu,\Sigma}(W^i) = \sum_{i=1}^{C} \sup_{z \sim Z(\mu,\Sigma)} \Pr(z \in Q^i) \tag{18}$$

The hinge part of the latent mixed risk aggregates the K-hyperplane hinge risk over C components:

$$L^H_D(W_{LHM}; \varphi) = \mathbb{E}_{(x,y)\in D}\left[\sum_{i=1}^{C} \ell(W^i; x, y)\mathbb{1}\left[\varphi(x) = i\right]\right] \tag{19}$$

where

$$\ell(W; x, y) = \max_{j \in \{1..K\}} \{\max\{0, \alpha - y w_j^T x\}\}$$

is the modified K-hyperplane hinge loss (in Eq. 2). We replaced 1 with α to accommodate comparison between different norms of the hyperplanes.

3.2. Empirical Latent Mixed Risk

Recall that S is a training sample of size m, where $S_2 = \{(x, y) \in S : y = 1\}$, and $S_1 = \{(x, y) \in S : y = -1\}$ are the positive and negative training sets correspondingly, m_2 is the size of S_2 and m_1 be the size of S_1. We define the empirical risk over S as follows:

$$L^{HB}_S(W_{LHM}, \varphi) = L^H_{S_2}(W_{LHM}, \varphi) + L^B_{S_1}(W_{LHM}) \tag{20}$$

Both parts of the empirical latent mixed risk are aggregated over C latent components. Specifically, the background part of the risk is defined as a sum of background empirical risks of the model's components:

$$L^B_{S_1}(W_{LHM}) = \sum_{i=1}^{C} L^B_{\hat{\mu},\hat{\Sigma}}(W^i) \tag{21}$$

where $L^B_{\hat{\mu},\hat{\Sigma}}(W^i) = \sup_{Z\in\Omega(\hat{\mu},\hat{\Sigma})} Pr_{z\sim Z}(z \in Q^i)$ and $\hat{\mu}, \hat{\Sigma}$ are the empirical mean and covariance matrix, estimated from the negative training sample S_1.

Let $X^i \triangleq \{x : \varphi(x) = i\}$ define a subset of positive samples. The hinge part of the empirical risk is defined as the follows:

$$L^H_{S_2}(W_{LHM}, \varphi) = \sum_{i=1}^{C} L^H_{S_2}(W^i) \tag{22}$$

where $L^H_{S_2}(W^i) = \sum_{x\in X^i} \ell(W^i, x, 1)$.

Using the above notations, we can define a component empirical risk, as

$$L^{HB}(W^i) = L^H_{S_2}(W^i) + L^B_{\hat{\mu},\hat{\Sigma}}(W^i) \tag{23}$$

Next, we formalize the loss function for a positive sample which we use in the training algorithm in Section 3.3. Each sample with positive label encounters a loss only in a single latent component, specified by its latent variable $\varphi(x)$. The hinge part of this loss is $\ell(W^i, x, 1)$. The background empirical risk of a component $L^M_{\hat{\mu},\hat{\Sigma}}(W^i)$ depends on $\hat{\mu}, \hat{\Sigma}$ and W^i. W^i depends on the latent assignment of the positive samples. Thus the optimal assignment should minimize also the background part of the risk. We implement this by dividing the empirical risk $L^M_{\hat{\mu},\hat{\Sigma}}(W^i)$ equally among the positive samples with $\varphi(x) = i$. Hence the sample loss (for positive samples) is defined as follows:

$$L(W^{\varphi(x)}; x, 1, \varphi(x)) = \ell(W^{\varphi(x)}, x, 1) + \frac{1}{|S^i|} L^M_{\hat{\mu},\hat{\Sigma}}(W^{\varphi(x)}) \tag{24}$$

where $|S^i|$ is the number of samples with $\varphi(x) = i$.

3.3. LHM Training Algorithm

The training aims to minimize the empirical risk in Eq. 20 over the parameters W_{LHM} and the hidden variables φ. We propose an iterative algorithm, which reaches fast convergence and shows good results in practice. The algorithm iterates between two steps: First, given an assignment it produces a model W_{LHM}, second, it updates the latent variables $\varphi(x), \forall(x, y) \in S_2$ to better represent the latent structure of the data.

The **first** step updates the LHM model W^t_{LHM} in iteration t given the latent variables φ from iteration $t-1$. Namely, for each hidden component $i = 1, ..., C$, we find the hyperplanes W^i separating the training samples in S^i from D_{neg} by minimizing the empirical risk in Eq. 23. This risk is minimized by the training algorithm proposed in Algorithm 1.

The **second** step updates the latent variable assignment, given the current W^t_{LHM}. For each positive sample, it finds the best component w.r.t. the risk in Eq 20. Specifically, the hinge risk for x is simply $\ell(W^i, x, 1)$. The background part of the assignment function for $x \notin Q^i$ should consider the probability that this point adds when it is included in the component i (as shown in Figure 3, left). For $x \in Q^i$, the background part should consider the amount of probability released when the component shrinks as a result of change in the assignment of x (as shown in Figure 3, right). The optimal assignment should take both cases into consideration for all components. To define the assignment function we introduce the following notations.

W^{def} is a *deflated model* derived from W^i by parallel translation of the hyperplane closest to x such that $w^T_* x + b_* = 0$. W^{inf} is an *inflated model* derived from W^i by parallel translation of the hyperplanes for

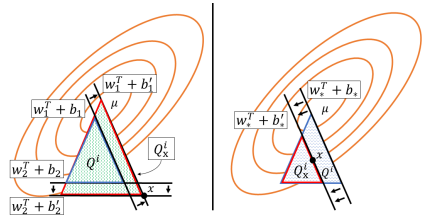


Figure 3: The orange ellipses represent the negative distribution $Z(\hat{\mu}, \hat{\Sigma})$, the red triangles corresponds to Q^i_x and the blue ones to Q^i. **Left**: w^i_1, w^i_2 are moved to pass through x, causing the probability Q^i to increase. **Right**:w^i_* is moved to pass through x, causing the probability Q^i to decrease.

which $w^T_k x + b_k < 0$, until they intersect in x, namely, $w^T_k x + b'_k = 0$. W^i_x is a surrogate model, defined as follows,

$$W^i_x \triangleq \begin{cases} W^{def} & \text{if } x \in Q^i \\ W^{inf} & \text{if } x \notin Q^i \end{cases}$$

and $Q^i_x \triangleq \{x : {W^i_x}^T x \geq \vec{0}\}$. Using the above notations, we define the assignment function as follows,

$$\varphi(x) = \underset{i\in\{1..C\}}{\operatorname{argmin}} Pr_{z\sim Z}(z \in Q^i_x) + \lambda\ell(W^i; x, 1) \tag{25}$$

where λ is a balancing parameter. The full training algorithm is summarized in Algorithm 2.

Lemma 9 *Algorithm 2 reduces the empirical risk $L^{HB}_S(W_{LHM}; \varphi)$ in each iteration.*

Proof Since latent mixed risk is a sum of risks over the latent components (Eq. 20), it is minimized by minimizing the empirical risk of each component. In step (5) of the Algorithm 2, we train $W^{i,t}$ model for each latent component $i = 1, ..., C$ using Algorithm 1 (Section 2.2). It is easy to see that $L^{HB}_S(W^i) = L^{HB}_S(W)$ (in Eq. 7), thus step (5) of the Algorithm 2 minimizes the component's risk in Eq. 23.

It is now left to show that the assignment φ^t in iteration t, will cause the reduction in the empirical risk in iteration $t+1$. Since the empirical risk is aggregated over positive samples, it is enough to prove the claim for a single sample. We consider two cases:

1. The assignment of sample x does not change, formally $\varphi^t(x) = \varphi^{t+1}(x)$.
In this case $L(W^{t+1}_{LHM}; \varphi^{t+1}(x))$ will only be affected by the $W^{i,t+1}$ training, thus

$$L\left(W^t_{LHM}; \varphi^t(x)\right) \geq L\left(W^{t+1}_{LHM}; \varphi^{t+1}(x)\right) \tag{26}$$

2. The assignment of sample x is changed. Formally, in iteration t: $\varphi^t(x) = i$ and in iteration $t+1$, exists $j \neq i$, such that

$$\varphi^{t+1}(x) = j = \underset{k\in\{1..C\}}{\operatorname{argmin}} L^B_{S_1}\left(W^{k,t}_x\right) + \lambda L^H_{S_2}\left(W^{k,t}; x\right). \tag{27}$$

Since $x \in Q^i$, reassigning it to a different component will cause the $Pr_{z\sim Z}(z \in Q^i_x)$ to decrease (or stay the same), thus

$$L^B_{S_1}(W^{i,t}_x) - L^B_{S_2}(W^{i,t}) \leq 0. \tag{28}$$

Algorithm 2 LHM Training. T is the threshold on the empirical risk change.

Input: C, K, S_1, S_2,T
Initialization:
$t \leftarrow 1$
$L(W_{LHM}^{t=0}; \varphi^{t=0}) \leftarrow \infty$
$\varphi^t \leftarrow Init(S_2, C)$

Training:
while $L(W_{LHM}^t; \varphi^t) - L(W_{LHM}^{t-1}; \varphi^{t-1}) \geq T$ **do**
{Model Step}
for i=1 to C do
$W^{i,t}$ =*KHHM-training*(S_1,X^i) {in Algorithm 1}
end for
{Assignment Step}
for$(x, y) \in S_2$
$\varphi^{t+1}(x)$ as defined in Eq. 25
end for
$t \leftarrow t + 1$
end while
Output: W_{LHM}, φ

Hence, the sample loss in component i is larger than the sample loss in the deflated component:

$$L\left(W^{i,t}; x\right) \geq L_{S_1}^B\left(W_x^{i,t}\right) + \lambda L_{S_2}^H\left(W^{i,t}; x\right). \tag{29}$$

At the same time, j is the optimal assignment, thus

$$L_{S_1}^B\left(W_x^{i,t}\right) + \lambda L_{S_2}^H\left(W^{i,t}; x\right) \geq L_{S_1}^B\left(W_x^{j,t}\right) + \lambda L_{S_2}^H\left(W^{j,t}; x\right). \tag{30}$$

Since $W_x^{j,t}$ is a naive inflation of $W^{j,t}$ to include x, the solution $W^{j,t+1}$, provided by KHHM training, would have lower (or same) empirical risk, thus

$$L_{S_1}^B\left(W_x^{j,t}\right) \geq L_{S_1}^B(W^{j,t+1}). \tag{31}$$

In iteration $t+1$, x is included in X^j for training the j'th latent component, consequently

$$L_{S_2}^H\left(W^{j,t}; x\right) \geq L_{S_2}^H(W^{j,t+1}; x). \tag{32}$$

(as we assume that $x \in X^j$ leads to $x \in Q^{j,t+1}$). Finally, by combining the inequalities in Eq. 29–32, we obtain:

$$L\left(W^{i,t}; x\right) \geq L\left(W^{j,t+1}; x\right). \tag{33}$$

■

3.4. Generalization Bound for LHM Model with Fixed Assignment

For a fixed $\varphi(x), \forall(x, y) \in S_2$, we can derive a uniform generalization bound for the union of the K-hyperplane models. Similarly to KHHM model (in Section 2.3), we derive the uniform generalization bounds separately for the positive and negative classes. We start with the positive class, for which we use the hinge part of the latent mixed risk.

Theorem 10 *Let φ^* denote a fixed assignment of the positive training samples to components. Let $L_D^H(W_{LHM}; \varphi^*) = \mathbb{E}_{(x,y)\in D}\left[\sum_{i=1}^C \ell(W^i; x, y)\mathbb{1}\left[\varphi^*(x) = i\right]\right]$ be the expected risk, and let $L_{S_2}^H(W_{LHM}, \varphi^*) = \sum_{i=1}^C L_{S_2}^H(W^i)$ be the empirical risk over a positive label training sample of size m_2 for a fixed assignment φ^* of the positive samples to C components. Then, for any $\delta \in (0, 1]$ with probability at least $1 - \delta$ over the i.i.d. sample of size m_2 it holds simultaneously for all $\|w_i\| \leq 1$ $(i = 1, ..., C \cdot K)$ that whenever $\|x\| \leq 1$:*

$$L_D^H(W_{LHM}; \varphi^*) \leq L_{S_2}^H(W_{LHM}, \varphi^*) + \sqrt{\frac{\log 1/\delta}{2m_2}} L_{S_2}^H(W_{LHM}, \varphi^*) + \max_{i\in\{1,...,C\}}\left(\frac{2K}{\sqrt{m^i}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}\right)$$

Proof Let $p_i = E_{(x,1)\sim D}\Big[\mathbb{1}[\varphi^*(x) = i]\Big]$, and let $\frac{m_i}{m^+}$ be its estimated mean. Then,

$$L_D^H(W_{LHM}; \varphi^*) - L_{S_2}^H(W_{LHM}, \varphi^*) = \sum_{i=1}^C\left(p_i L_D^H(W^i) - \frac{m_i}{m_2} L_{S^+}^H(W^i)\right)$$
$$= \sum_{i=1}^C\left[p_i\left(L_D^H(W^i) - L_{S_2}^H(W^i)\right)\right] + \sum_{i=1}^C\left[(p_i - \frac{m_i}{m^+})L_{S_2}^H(W^i)\right] \tag{34}$$

We can bound the discrepancy between the expected and empirical risks in a component using Theorem 8. Hence, the first term in Eq. 34 is upper bounded by $\max_{i\in\{1,...,C\}}\left(\frac{2K}{\sqrt{m^i}} + 8K\sqrt{\frac{2\log(2/\delta)}{m_2}}\right)$. We can upper bound $(p_i - \frac{m_i}{m_2})$ using the Hoeffding inequality. Rearranging the terms and noting that $\sum_{i=1}^C L_{S_2}^H(W^i) = L_{S_2}^H(W_{LHM}, \varphi^*)$ conclude the proof. ■

We formulate the uniform generalization bound for the negative class below.

Theorem 11 *Suppose that D is a distribution over $X \times Y$ such that $Y = \{-1, +1\}$ and $X = \{x : \|x\| \leq G\}$. Let $L_{\mu,\Sigma}^B(W_{LHM}; \varphi^*)$ be the background risk over the negative labels, where μ, Σ are the mean and covariance of the marginal distribution of x over the negative labels and the positive labeled samples have a fixed assignment φ^*. Consider a training sample S of size m, m_1 of which have negative label and let*

$$L_{\hat{\mu},\hat{\Sigma}}^B(W_{LHM}; \varphi^*) = \sum_{i=1}^C L_{\hat{\mu},\hat{\Sigma}}^B(W^i)$$

be the empirical background risk over the negative labels ($\hat{\mu}, \hat{\Sigma}$ are the empirical mean and covariance estimation of D_{neg}). With probability at least $1 - 3\delta$ over m_1 the i.i.d. samples from D_{neg} the following holds uniformly for all W including C components, each with K independent hyperplanes:

$$L_{\mu,\Sigma}^B(W_{LHM}; \varphi^*) \leq L_{\hat{\mu},\hat{\Sigma}}^B(W_{LHM}; \varphi^*) + C\Big(\frac{2G^2}{\alpha} + \frac{6G^4}{\alpha^2}\Big)\sqrt{\frac{32(\log(1/\delta) + 1/4)}{m_1}}.$$

The proof is straightforward as the assignment φ^* does not affect negative samples, and thus the bound is a simple summation of bounds for each component which is derived using Theorem 6.

4. Mapping LHM Classifier to a Neural Network

Deep Neural Networks and Convolutional Neural Networks (CNN) in particular have shown impressive results in a variety of domains, including images, speech, text, etc. CNN enables learning very good features for these domains, but requires a lot of labeled training samples. One way to reduce the number of examples

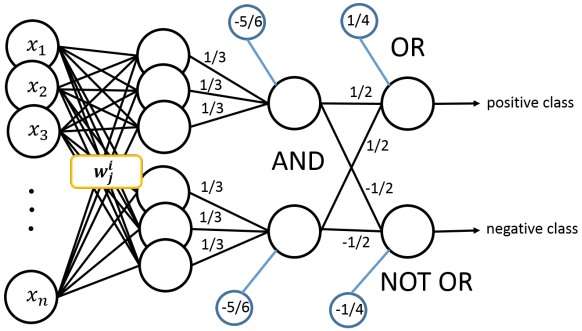


Figure 4: An example of NN equivalent to LHM for two components and three hyperplanes in each.

needed for training of a specific classification task is to pre-train a CNN on a different classification problem in a similar domain and then change the last layer of the CNN to fit the target classification problem and fine-tune the network on a smaller training set associated with the target problem. This approach is referred to as transfer learning.

When the target classification problem is less similar to that used to train CNN features, the classification accuracy of the fine-tuned network could be quite poor. This is because the high-level representation of the original and the target networks are different. One way to approach this problem is by employing a non-linear classifier, such as LHM classifier, on CNN features. In order to fine-tune the feature layers with the LHM classifier, we need to combine them in a single architecture. To this end we propose to map LHM Classifier to a Neural Network and stack it on top of the pre-trained convolutional layers. This enables end-to-end training of feature extraction and classifier. As we show below, mapping of LHM classifier to NN also allows extending it to multi-class problems.

4.1. Binary NN

A union of the intersections of positive half spaces can be implemented by a NN with three hidden layers. The first fully connected hidden layer has $K \times C$ neurons with a sigmoid activation, where K is the number of hyperplanes in an intersection and C is the number of components. The second hidden layer has C nodes with a sigmoid activation, connected only to the neurons associated with hyperplanes forming the corresponding intersection. The weights on these connections and the biases are fixed and mimic **AND** operation, namely, all weights of this layer are equal to $1/K$ and the biases are equal to $-1 + 1/(2K)$. The last hidden layer has two neurons, which are fully connected to the previous layer with the fixed weights and biases, one of which mimics **OR** operation, and the other **NOT OR**. Namely, the neuron, corresponding to OR has weights equal to $1/C$ and the bias of $-1/(2C)$. The neuron corresponding to NOT OR has weights equal to $-1/C$ and the bias of $1/(2C)$. The network has two outputs, one for the positive class (with label 1) and one for the negative class (with label 0). An example of such network for $C = 2$ and $K = 3$ is depicted in Figure 4.

4.2. Multi-Class NN

For a multi-class setting, we suggest to train LHM model for each class using an additional unlabeled data for estimating the statistics of the negative class. We then map these models to a multi-class NN with the following architecture. The first hidden layer is a fully connected layer with $C \times K$ neurons per class,

$C \times K \times G$ neurons in total, where G is the number of classes. These are equivalent to $C \times K \times G$ hyperplanes in the LHM model. For each hidden component, all hyperplanes in the intersection are connected to their corresponding node in the **AND** layer (as detailed in Section 4.1). The **AND** layer comprises $C \times G$ neurons. The next layer is a fully connected layer, comprising G nodes. The weights on the connections to the C components of the corresponding class are initialized with 1's, and the weights on the remaining connections are initialized with very small values from a Gaussian distribution. The network has G outputs and is trained using the cross-entropy loss.

To provide an end-to-end training, one can consider stacking the feature extraction layers of CNN (up to fully connected layers) with one of the above networks.

5. Experiments

We start by evaluating the simple K-hyperplane model (Section 5.1) and then move to a more general LHM model (Section 5.2). Both models are tested on synthetic and real data. Then we show how the hinge-minimax training can be combined with a CNN (Section 5.3) for approaching problems that require more powerful features but do not have large enough data to train a deep model from scratch. We demonstrate this for both binary and multi-class settings.

5.1. K-Hyperplane Hinge-Minimax Classifier

To test the proposed KHHM classifier, we ran experiments in three different scenarios: synthetic 2D data, letter recognition, and large scale scene classification.

During classification, the K-hyperplane classifier incurs only K times the computational complexity of a linear classifier (just K inner products), hence its "natural competitors" are linear classifiers, and we choose linear SVM for the benchmark. We have also compared the hinge-minimax classifiers to kernel SVM and ensemble-based methods, which incur far longer running times (this is especially true for kernel SVM). The classification rates of the hinge-minimax classifier in all our experiments were comparable to ensemble classifiers which required 100-170 basic classifiers in order to reach similar performance. In experiments with high-dimensional data, the KHHM classifiers preformed as well as kernel SVM.

The SVM classifiers were trained using C-SVC in LIBSVM [4]. We used the CVX optimization package [5] to find a single hyperplane in Algorithm 1. The ensemble classifiers were trained using the Matlab Statistic toolbox.

5.1.1. SYNTHETIC DATA EXAMPLE

We construct the KHHM classifier for 2D data to illustrate Algorithm 1. We samples 5000 data points from two highly overlapping Gaussians (see Figure 5) with varying ratio of positive (shown in red) and negative (shown in blue) examples. Each class was equally partitioned into training, validation, and test sets. We estimated the mean and covariance from the training data and tuned the parameters (C and γ) and the bias using the validation set. Table 1 shows the AUC for the different ratios of positive and negative examples using an intersection of 5 hyperplanes. These results demonstrate the robustness of the algorithm to imbalanced sets.

Positive fraction	0.01	0.1	0.2	0.3	0.4	0.5
AUC	94.68	94.91	95.07	94.96	94.89	95.83

Table 1: AUC for different size partitions of positive and negative classes

4. *http://www.csie.ntu.edu.tw/cjlin/libsvm/*
5. *http://cvxr.com/cvx/download/*

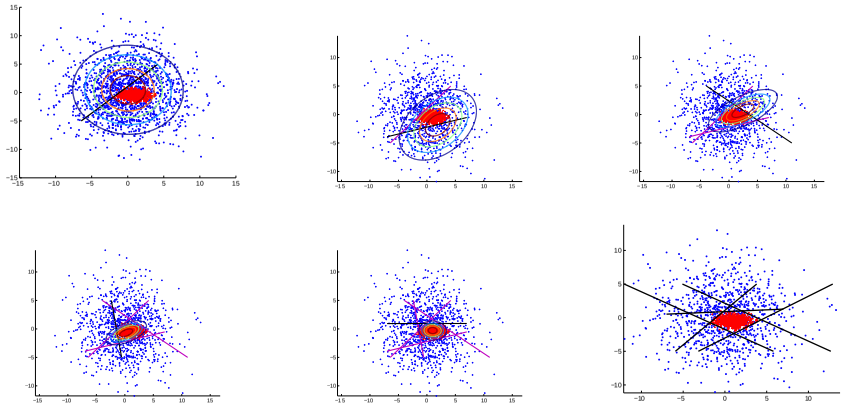


Figure 5: Illustration of KHHM classifier construction on a toy example.The first 5 figures show the greedy initial step. The last figure shows the final classifier after 25 iterations. The contour lines show the covariance matrix of the negative distribution inside the intersection of hyperplane, which is used to find the optimal hyperplane, depicted in black.

The first five plots in Figure 5 show the result of the initial greedy step for the first, second, third, forth, and fifth hyperplanes respectively. The contour lines in Figure 5 illustrate the covariance of the negative distribution inside the intersection, which is used to find the optimal separation hyperplane, depicted in black. The last plot in Figure 5 shows the final classifier after 25 iterations. It illustrates that the approximation algorithm succeeds in separating the positive set from the background, and that the refinement iterations improve the separation boundary.

5.1.2. LETTER RECOGNITION

The following tests were performed on a data set of letters from the UCI Machine Learning Repository (Murphy and Aha (1994)), which includes 16-dimensional feature vectors for the 26 letters in the English alphabet. The letter images were based on 20 different fonts and each letter within these 20 fonts was randomly distorted to produce 20,000 samples. For each letter, we used 100 samples for training, 250 for validation, and the rest for test (about 400 samples per letter). The parameters of all methods have been chosen using the validation set. Since the test set includes 25 times more negatives than positives, which leads to about 96% classification rate by just classifying all inputs as negative, we used EER as a more faithful measure of performance. Table 2 shows the classification rate at EER , averaged over 26 letters, and the average classification times of the tested classifiers.

The KHHM classifiers improve over the linear SVM for all K, and for $K > 1$ outperforms Adaboost with much shorter classification time. For this data set, kernel SVM outperform all methods. However, the KHHM classifier with $K = 4$ comes fairly close to the performance of the kernel SVM, while its classification time is three magnitudes faster.

Method	Classification rate at EER	Classification time
KHHM $K = 1$	89.32	5.6e-07
KHHM $K = 2$	92.98	1.4e-06
KHHM $K = 3$	93.93	1.5e-06
KHHM $K = 4$	94.48	1.7e-06
Linear SVM	84.87	4.6e-07
RBF kernel SVM	96.47	1.7e-03
AdaBoost	92.26	1.0e-03

Table 2: Letter experiments. K corresponds to the number of hyperplanes used in the hinge-minimax classifier. The times are in sec. AdaBoost uses 100 decision trees.

Method	AUC	classification time
KHHM $K = 1$	88.89	9.8e-05
KHHM $K = 2$	90.99	1.34e-04
Linear SVM	88.20	8.6e-05
RBF kernel SVM	90.77	23.97
RUSBoost	90.76	0.08

Table 3: Scene classification with 300 dim. features. The classification time of RBF kernel SVM is very high, since it chooses about 15,000 SVs from 19850 training examples. The RUSBoost uses 100 decision trees.

5.1.3. LARGE SCALE SCENE RECOGNITION

In this test we used 397 scene categories of the SUN data base, which have at least 100 images per category (Xiao et al. (2010)). We represent the images as BOW of dense HOG features with 300 words. We downloaded the features from the SUN web page[6], containing spatial pyramid of BOWs, and used the bottom layer (the details of the feature extraction can be found in Xiao et al. (2010)). The data is divided into 50 training and 50 test images in 10 folds. Training one-against-all classifiers for 397 categories with 50 training samples per category uses very unbalanced training sets. Thus we defined different weights for positive and negative samples in SVM training and we used RUSBoost (Seiffert et al. (2008)) as an ensemble method (it is designed for skewed data and performed significantly better than AdaBoost on this data set). Note that the KHHM classifier naturally handles imbalanced sets. KHHM classifier with more than two hyperplanes didn't improve the performance. Table 5.1.3 shows the average AUC of the tested methods and their running times.

Using a pyramid of BOWs with the histogram intersection kernel improves over the RBF kernel applied to the bottom layer of the pyramid, but then the dimension of the feature vector increases to 6300. The AUC of the KHHM classifier with $K = 2$ is 92.99% and of histogram kernel is 92.85%. Figure 6 shows the ROCs of the first three categories produced by the KHHM classifier and the histogram intersection kernel SVM classifiers.

6. $http://vision.cs.princeton.edu/projects/2010/SUN/$

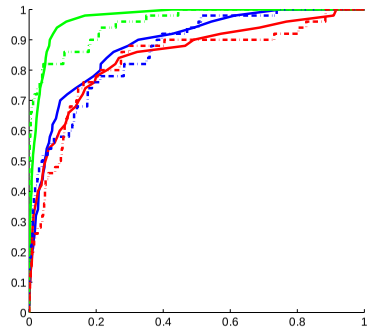


Figure 6: ROCs of the first three categories of the SUN data set, represented by a spatial pyramid of BOWs, obtained from dense HOG. The solid lines correspond to the hinge-minimax classifier, dotted lines correspond to the histogram intersection kernel SVM.

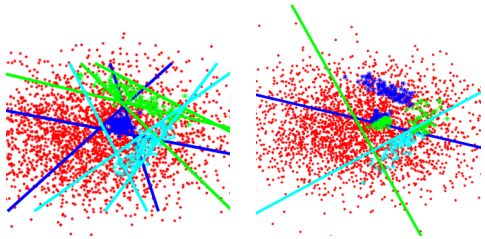


Figure 7: A qualitative comparison of the latent hinge minimax classifier (on the left) to the union of LDA classifiers (on the right).

5.2. Latent Hinge Minimax Classifier

We first show a 2D toy example (Section 5.2.1) to illustrate the ability of the LHM classifier to discover the hidden components in the positive class and to separate each of them from the negative class using a K-hyperplane model.

Then, we compare LHM model to alternative ensembles of hyperplanes (in shallow architectures) on the PASCAL-VOC 2007 dataset (Everingham et al. (2010)) (Section 5.2.2), and show its advantage over those methods and its robustness to the choice of the number of latent components.

5.2.1. SYNTHETIC DATA

A simpler alternative to the LHM model is a two-step algorithm which first finds the structure of the target class by applying some kind of unsupervised learning (e.g, k-means clustering) and then builds a model

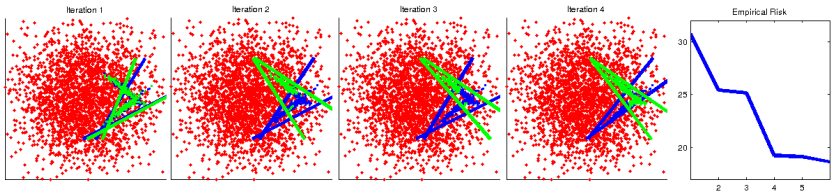


Figure 8: First four iterations of the LHM training on toy example and the corresponding loss convergence.

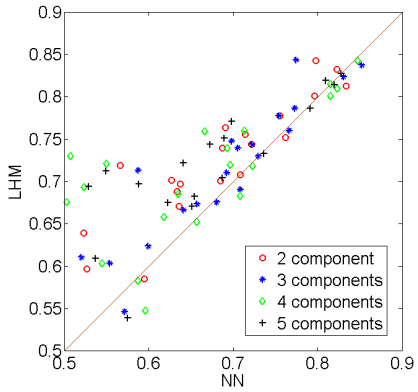


Figure 9: Comparison of the LHM classifier to the equivalent NN for a varying number of hidden components (from 2 to 5) on PASCAL VOC 2007. The points above the diagonal line show the advantage of LHM classifier.

for each component. Such a simple approach was employed in Hariharan et al. (2012) with LDA classifier (Hastie et al. (2001)) trained per cluster. Unless the clusters are very small[7], it relies heavily on the results of the clustering. If an initial clustering is incorrect (as in Figure 7, right), LDA (or any other convex classifier) cannot separate the resulting components from the background without including many false positives. The LHM training finds the underlying structure of the data and the model iteratively, improving both (Figure 7, left). Furthermore, LHM is quite robust to the initial assignment. Figure 8 shows a few iterations and the corresponding loss convergence when the initial assignment of the positive samples to components is chosen at random. Note the LHM training discovers the underlying structure in a 3-4 iterations.

7. as in time consuming exemplar-based approach(Malisiewicz et al. (2011))

LHM	Union of LDAs	NN	KHHM
71.48%	65.17%	67.19%	69.45%

Table 4: The table reports the accuracy at the EER point averaged over 20 classes and different hidden partitions (except for KHHM) on PASCAL VOC-2007 classification task using 80-dimensional HOG features.

5.2.2. ENSEMBLES OF HYPERPLANES

Next, we compared the LHM classifier to alternative ensembles of linear classifiers on PASCAL VOC 2007 dataset (Everingham et al. (2010)). To compare the raw performance of the classifiers we designed the experiment to separate the contribution of the classifier from that of features and the detection system (which usually involves various engineering steps that obscure the actual contribution of the classifier). To this end we used simple features, such as Dalal-Triggs variant of the HOG features (Dalal and Triggs (2005)) with a fixed number of cells (thus keeping the classification problem difficult), and we compared the classification accuracy on the bounding boxes of 20 VOC object categories in test images (instead of running a full detection system).

LDA Union (as a baseline model): We applied k-means clustering on whitened features to find the partition. We then learned an LDA classifier for each cluster in that partition. We varied the number of clusters from 2 to 5.

NN with an architecture equivalent to LHM: We used the model described in Section 4.1 with $K = 2$ and $H = 2, .., 5$, but the weights were initialized at random.

KHHM model This is essentially an LHM model with a single component, thus it is theoretically inferior to LHM. However, we ran this experiment to test the benefits of modeling the hidden structure of the positive class. We varied the number of hyperplanes from 2 to 5.

LHM model: We set the number of hyperplanes in each component to 2 and varied the number of components from 2 to 5. An initial assignment to the components was done using k-means with the Euclidian distance.

All ensembles were trained in one-against-all manner. Similarly to (Hariharan et al. (2012); Osadchy et al. (2012)), we learned the background mean and covariance using bounding boxes from all classes and used them to represent the negative class in LDA union, KHHM, and LHM training. We tested all ensemble classifiers on all bounding boxes from the test set. Table 4 summarizes the accuracy at the EER points of all ensembles averaged over classes and different parameters. It shows that LHM model outperforms all other classifiers. Figure 9 compares LHM to NN on 20 categories (as one-against-all binary classifiers) for varying number of hidden components. The plot shows that LHM outperforms NN independently of the number of components.

5.3. Hinge-Minimax Training in Deep Architecture

In the following experiments, we show that LHM classifier can be combined with CNN via transfer learning. Specifically, we test the LHM classifier on top of the pre-trained CNN feature extraction in imbalanced binary problems and in multi-class tasks with a small number of labeled examples.

We explore the following transfer learning settings. The first setting refers to the **best case** scenario in which the source and the target classification tasks operate on the *same* set of features but differ in the classification problem. The second setting refers to the **worst case** scenario for the transfer learning where the source and the target classification problems *share very little similarity*. The "worst case" scenario is very common in practice, as many classification tasks do not have a large, comprehensive training set (such as ImageNet (Deng et al. (2009)) in object recognition) to be used in transfer learning. No good solution currently exists for such problems.

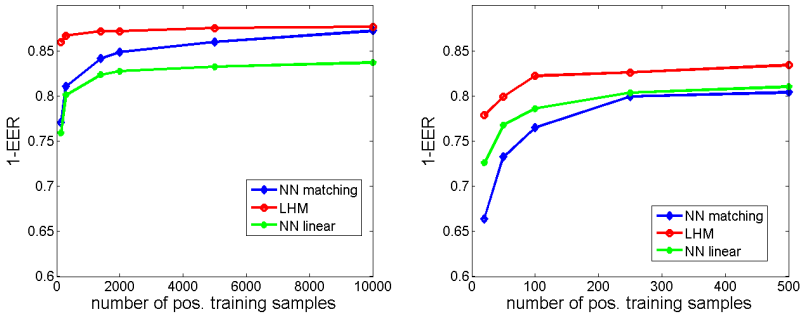


Figure 10: Binary imbalanced classification: left – the "best-case" transfer learning setting, right – the "worst-case" transfer learning setting.

We used the CIFAR-10, composed of 10 categories (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck) as the source problem. Specifically, we trained the LeNet model implemented in MatConvNet Vedaldi and Lenc (2015) on CIFAR-10. Then we removed the last fully-connected layer and the soft-max and used this trimmed network as a feature extractor which converts images to a 64-dimensional feature vectors.

For the best case transfer learning, we defined a new set of classes by coupling i and $i + 5$ indexes of CIFAR-10 classes. CNN trained on CIFAR-10 maps individual classes to linearly separable sub-spaces, thus using pairs of classes as a target classification problem makes it non-linear. Consequently, we get a new classification problem over the same space of features.

For the worst case transfer learning, we picked a subset of 5 classes (train, bottle, cattle, forest, and sweet peppers) from the CIFAR-100, which do not overlap (in their visual appearance) with the CIFAR-10 categories, to be the target classification task. CIFAR-10 data set is not rich enough to enable learning of features that can be used for an arbitrary category, thus we believe that such setting is especially difficult.

We tested the LHM binary and multi-class classifiers in the best and the worst case transfer learning scenarios and compared their performance to two baselines. One is an NN with a single fully connected layer and the cross-entropy loss (NN linear) and the other is the NN with the architecture matching the LHM model (NN matching). We repeated each experiment 50 times over different random subsets of training samples and random initialization of NN and averaged the results.

5.3.1. BINARY IMBALANCED SETTING

The "Best Case" Transfer Learning: We trained binary classifiers for pairs of classes from CIFAR-10 using imbalanced training sets, in which the negative class included all samples from all other classes (40,000 examples) and the positive class included a varying number of samples (140, 300, 600, 1400, 2000, 5000-all). This resulted in imbalance ratios from 1:256 to 1:4.

LHM model was trained with 2 hidden components and 3 hyperplanes per component. The matching NN mimicked the configuration of LHM model, but the weights were allowed to change in training. Figure 10-left shows the 1-EER (averaged over 5 classification problems) of the LHM classifier and the two NN baselines as a function of the positive training sample size.

The "Worst Case" Transfer Learning for Binary Imbalanced Problems: Since the number of samples per class in CIFAR-100 is significantly smaller, this experiment tests the robustness to imbalanced training data

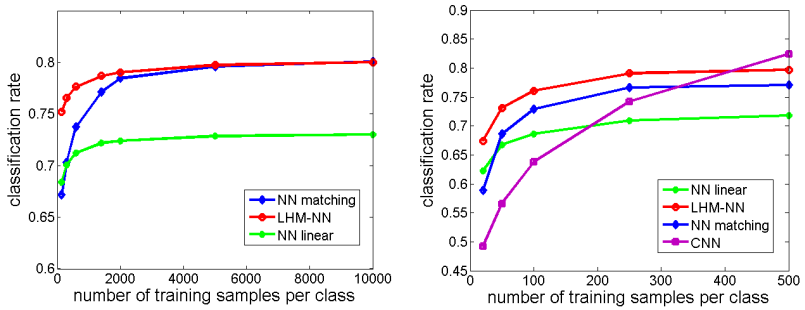


Figure 11: Multi-class classification: left – the "best-case" transfer learning setting, right – "worst-case" transfer learning setting.

and to a small number of examples. We varied the size of the positive training set between 20, 50, 100, 250, 500(all) samples and we used all 2,000 samples of other classes as the negative training set. We compared the LHM model trained with 2 hidden components and 2 hyperplanes per component to NN baselines. Figure 10-right shows the 1-EER of the classifiers averaged over 5 classification problems as a function of the positive training set size.

5.3.2. MULTI-CLASS SETTING

The "Best Case" Transfer Learning: We mapped the LHM binary classifiers trained for 5 pairs of categories to a multi-class NN as described in Section 4.2. We fine-tuned the weights with a very fast training (just a handful of epochs, while training from scratch requires two orders of magnitude more training epochs). Figure 11-left shows the accuracy of the LHM models mapped to a multi-class NN (LHM-NN) with the two baseline NNs as a function of the size of the training set.

The "Worst Case" Transfer Learning for Multi-Class Problems: We mapped the LHM binary classifiers trained for the 5 categories from CIFAR-100 (using CIFAR-10 features) to a multi-class NN and fine-tuned the weights with a small number of epochs.

To test the complexity of the transfer learning problem we also trained a CNN (LeNet model implemented in MatConvNet (Vedaldi and Lenc (2015))) on the target problem. We hoped that due to the small size of the target classification problem, 500 training examples per class would yield relatively good accuracy. Figure 11-right compares the accuracy of LHM-NN, two baseline NNs, and CNN (trained from scratch) as a function of the training sample size. It shows that CNN trained on the target problem is indeed the best as it succeeds to learn features specific for the task, but its accuracy drops very abruptly when the number of training samples becomes smaller. This suggests that when the number of training examples is small, using transfer learning even in a such difficult setting is a better solution than training a CNN from scratch.

The results in Figures 10 and 11 show that the NN models either heavily overfit when the number of training samples is small (NN matching) or they are not expressive enough when the number of training samples increases (NN linear). LHM classifiers are expressive enough to learn from a large set of examples and are more robust to overfitting when the number of examples is small.

6. Training Efficiency

Another advantage of LHM-NN is its training efficiency. A class-specific LHM model converges in 5-10 iterations. Its training time primarily depends on the number of positive samples and the dimension. The negative samples are used to estimate the mean and covariance of the background. The initial estimation (which involves a large number of samples) is done only once and used for all classes. Since the probability of the negative class is evaluated inside the positive region using false positives, the number of which drops very fast, the estimation time of the mean and covariance during the training is negligible. Training of a binary classifier per class is independent of other classes, thus their training can be done in parallel. Finally, the fine-tuning of the multi-class network after mapping is very fast, due to the initialization of all layers (using supervized learning): feature extraction layers with pre-trained CNN and classifier's layers with LHM models.

The LHM-NN is also beneficial for the problems in which classes are dynamically added or removed from the classification task. Adding a class requires training a single binary classifier and fast fine-tuning; removing a class requires only fine-tuning.

7. Conclusions and Future Work

We proposed an efficient method for learning an intersection of finite number of hyperplanes which combines the hinge-risk (for the small number of positive data) with the background risk, based on the "minimax bound" (for a large number of negative data points) and derived a generalization bound for the mixed risk. We showed that the proposed classifier yields results comparable to the popular non-linear classifiers, but at much lower (order of magnitude) computational cost of classification.

We generalized this model to a non-convex classifier (Latent Hinge-Minimax classifier), which discovers the hidden components in the positive class and separates them from the negative class with the intersections of positive half spaces. The main advantage of this classifier is its ability to incorporate unlabeled data in training which improves the robustness to imbalanced sets.

We showed that for multi-class tasks, class-specific LHM models can be mapped to a multi-class NN with matching architecture requiring only a few iterations of fine-tuning. Finally, the showed that LHM architecture can be integrated with CNN features via transfer learning. The entire training procedure is very efficient. Our experiments showed that such classifiers are much more robust to the number of labeled training samples than the equivalent NNs.

This work cay be extended in various directions. A lower Rademacher complexity was shown for the k-fold maxima of hyperplanes in Kontorovich (2018). We plan to extend this result to the maximum over hinge losses and improve the generalization bound for the positive sample. Structured output learning have had an impact on machine vision and can be applied to this framework while improving the multiclass procedure. Another direction is devising a unified probabilistic framework to include both the hinge-loss and background-loss. Also, extensions of Marshall-Olkin theorem to non-convex sets might have a significant impact on robust deep learning methods.

References

Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.

R. E. Bellman. *Introduction to Matrix Analysis 2nd ed.* New York: McGraw-Hilla, 1970.

Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Advanced Lectures on Machine Learning*, pages 169–207. Springer, 2004.

Emmanuel J Candes and Yaniv Plan. A probabilistic and ripless theory of compressed sensing. *Information Theory, IEEE Transactions on*, 57(11):7235–7254, 2011.

Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *CVPR*, pages 886–893, 2005.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. From average case complexity to improper learning complexity. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 441–448, 2014.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010.

Ross B. Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition,CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014.

Bharath Hariharan, Jitendra Malik, and Deva Ramanan. Discriminative decorrelation for clustering and classification. In *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part IV*, pages 459–472, 2012.

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer New York Inc., 2001.

Jean Honorio and Tommi Jaakkola. {Tight Bounds for the Expected Risk of Linear Classifiers and PAC-Bayes Finite-Sample Guarantees}. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 384–392, 2014.

Sham M Kakade, Karthik Sridharan, and Ambuj Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in neural information processing systems*, pages 793–800, 2008.

Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2 – 12, 2009.

Aryeh Kontorovich. Rademacher complexity of k-fold maxima of hyperplanes. 2018. URL `https://www.cs.bgu.ac.il/~karyeh/rademacher-max-hyperplane.pdf`.

Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From N to N+1: multiclass transfer incremental learning. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, pages 3358–3365, 2013.

Gert R.G. Lanckriet, Laurent El Ghaoui, Chiranjib Bhattacharyya, and Michael I. Jordan. A robust minimax approach to classification. *J. Mach. Learn. Res.*, 3:555–582, 2003.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. A Series of Modern Surveys in Mathematics Series. Springer, 1991. ISBN 9783540520139.

Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011*, pages 89–96, 2011.

Albert W. Marshall and Ingram Olkin. Multivariate chebyshev inequalities. *Ann. Math. Statist.*, 31(4):1001–1014, 1960.

P. Murphy and D. Aha. Uci repository of machine learning databases. *Tech. rep., U. California, Dept. of Information and Computer Science*, 1994.

M. Osadchy, D. Keren, and B. Fadida-Specktor. Hybrid classifiers for object classification with a rich background. In *ECCV (5)*, pages 284–297, 2012.

Margarita Osadchy, Daniel Keren, and Dolev Raviv. Recognition using hybrid classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 38(4):759–771, 2016.

Chris Seiffert, Taghi M. Khoshgoftaar, Jason Van Hulse, and Amri Napolitano. Rusboost: Improving classification performance when training data is skewed. In *ICPR*, pages 1–4, 2008.

Vladimir Vapnik. *The nature of statistical learning theory*. Springer Science & Business Media, 2000.

Andrea Vedaldi and Karel Lenc. MatConvNet: Convolutional Neural Networks for MATLAB. In *Proc. of the 23rd Annual ACM Conference on Multimedia Conference, Brisbane*, pages 689–692, 2015. doi: 10.1145/2733373.2807412.

J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *CVPR*, pages 3485–3492, 2010.

Tong Zhang. Covering number bounds of certain regularized linear function classes. *The Journal of Machine Learning Research*, 2:527–550, 2002.

Journal of Machine Learning Research 19 (2018) 1-28 Submitted 9/17; Revised 9/18; Published 10/18

Short-term Sparse Portfolio Optimization Based on Alternating Direction Method of Multipliers

Zhao-Rong Lai LAIZHR@JNU.EDU.CN
Department of Mathematics
Jinan University
Guangzhou 510632, China

Pei-Yi Yang YANGPY23@MAIL2.SYSU.EDU.CN
School of Mathematics
Sun Yat-Sen University
Guangzhou 510275, China

Liangda Fang FANGLD@JNU.EDU.CN
Xiaotian Wu WXIAOTIAN@JNU.EDU.CN
Department of Computer Science
Jinan University
Guangzhou 510632, China

Editor: Qiaozhu Mei

Abstract

We propose a short-term sparse portfolio optimization (SSPO) system based on alternating direction method of multipliers (ADMM). Although some existing strategies have also exploited sparsity, they either constrain the quantity of the portfolio change or aim at the long-term portfolio optimization. Very few of them are dedicated to constructing sparse portfolios for the short-term portfolio optimization, which will be complemented by the proposed SSPO. SSPO concentrates wealth on a small proportion of assets that have good increasing potential according to some empirical financial principles, so as to maximize the cumulative wealth for the whole investment. We also propose a solving algorithm based on ADMM to handle the ℓ^1-regularization term and the self-financing constraint simultaneously. As a significant improvement in the proposed ADMM, we have proven that its augmented Lagrangian has a saddle point, which is the foundation of the iterative formulae of ADMM but is seldom addressed by other sparsity strategies. Extensive experiments on 5 benchmark data sets from real-world stock markets show that SSPO outperforms other state-of-the-art systems in thorough evaluations, withstands reasonable transaction costs and runs fast. Thus it is suitable for real-world financial environments.

Keywords: short-term portfolio optimization, sparse portfolio, alternating direction method of multipliers

1. Introduction

Portfolio optimization (PO) via machine learning systems has been catching more and more attention recently (Li and Hoi, 2012; Huang et al., 2013; Li et al., 2015; Li and Hoi, 2014; He et al., 2015; Yang et al., 2015; Huang et al., 2016; Li et al., 2016; Mahdavi-Damghani et al., 2017). It aims to invest in a group of financial assets with instructions by the machine, based on some financial principles and optimization strategies. Since it requires a large amount of quantitative calculation, machine learning methods are in high demand to reduce mistakes and biases made by human in real-world investment.

PO originates from the mean-variance theory (Markowitz, 1952), which seeks to minimize the volatility of a portfolio given a certain expected return level. From then on, various theories based on such a framework are proposed in the language of probability and statistics (Sharpe, 1964; Fama, 1965; Lintner, 1965; Fama, 1970; Treynor and Black, 1973). The equally-weighted portfolio is a popular theory that can achieve good performance if the individual risks of the assets are similar, while the risk parity theory is efficient in diversifying the portfolio risk if the individual risks are significantly different (Maillard et al., 2010). Instead of treating PO as a static problem, the stochastic portfolio theory (Fernholz, 2002) further extends it to a dynamic problem under some strict statistical assumptions. Besides, some special objectives can be achieved by adding specific prior knowledge into the dynamic model, like the rank-dependent portfolios. However, all the above methods are rather theoretical and their conclusions are based on very strict statistical assumptions that are far from real-world situations. On the other hand, machine learning strategies do not require such strict assumptions, thus they are more productive and effective in a well-defined technical sense (Das et al., 2013; Shen et al., 2014; Ho et al., 2015).

The attempt to construct sparse portfolios via machine learning starts in the long-term PO. In general, a long-term PO changes its portfolio once for a week or a month (or even a year). It focuses on the choice of assets rather than the timing of trading (buying or selling). A kind of sparse and stable Markowitz portfolios (SSMP) is proposed by Brodie et al. (2009). It adds an ℓ^1-regularization (Candès and Plan, 2009) term of the portfolio vector to the traditional Markowitz model, which regulates the amount of short-position in the portfolio. The regularized symmetric tail average (STA) minimization is proposed by Still and Kondor (2010), which exploits the Karush-Kuhn-Tucker (KKT) conditions (Bertsekas, 1999) to make the portfolio sparse. Ho et al. (2015) also construct sparse mean-variance portfolios with weighted elastic net penalization (Zou and Hastie, 2005). Shen et al. (2014) propose the doubly regularized portfolio allocation to control the quantity of portfolio change. Empirical or simulated experiments indicate that they show some advantages in the long-term PO.

However, few systems construct sparse portfolios for the short-term and online PO, which changes its portfolio once for a day (or even shorter). Since the short-term PO is more general than the long-term PO (e.g., one can change the portfolio only on the first day of each month), the short-term PO considers both of the choice of assets and the timing of trading. While the objective of long-term PO is to minimize the quadratic risk (Markowitz, 1952; Brodie et al., 2009; Ho et al., 2015; Shen et al., 2014), the objective of short-term PO is to maximize the increasing factor on each day and ultimately the cumulative wealth (Cover, 1991; Agarwal et al., 2006; Li et al., 2011; Das et al., 2013; Li et al., 2015; Huang et al., 2016; Li et al., 2016).

Alternating direction method of multipliers (ADMM) (Boyd et al., 2010) is an efficient and reliable approach for solving ℓ^1-regularization problems in many applications (Yin et al., 2015; Fang et al., 2016). However, it is nontrivial to implement ADMM (and other approaches) on the short-term sparse PO especially when both the ℓ^1-regularization term and the self-financing constraint are present (Shen et al., 2014), thus there are few

public solving schemes. Shen et al. (2014) point out that this problem is difficult and turn to a commercial software for solution. Ho et al. (2015) include the ℓ^1-regularization term but exclude the self-financing constraint. Although Das et al. (2013) successfully set up an ADMM algorithm, they do not prove that its augmented Lagrangian has a saddle point, which is the foundation of the iterative formulae of ADMM. In fact, the saddle point proof is also absent in other applications (Mohsin et al., 2015; Zhan et al., 2016) than PO. Besides, not all designs of ADMM have saddle points.

To address the above-mentioned problems, we propose a novel Short-term Sparse PO (SSPO) system based on ADMM. It well fits the machine learning framework for the short-term PO and is free of the strict assumptions of the stochastic portfolio theory. Its novelty falls into several aspects.

- Most state-of-the-art short-term PO systems focus only on adopting empirical financial principles (like the mean reversion) (Li et al., 2012, 2013, 2015; Huang et al., 2016), but SSPO also exploits the intrinsic sparse structure of portfolio by using an ℓ^1-regularization term and a self-financing constraint simultaneously, which lacks public solving schemes.

- Most previous sparsity models minimize the square error (Brodie et al., 2009; Still and Kondor, 2010; Lai et al., 2015; Zhan et al., 2016) or the quadratic risk (Brodie et al., 2009; Ho et al., 2015), but SSPO maximizes an increasing factor, which is quite different from the former in formulation.

- Previous sparsity models for short-term PO constrain the quantity of the portfolio change to form "lazy updates" (Das et al., 2013; Shen et al., 2014), which are defensive strategies and not really sparse PO systems. But SSPO actively updates the sparse portfolio according to the changing increasing potential of different assets as time passes, which is an aggressive strategy.

- We prove that the augmented Lagrangian of SSPO has a saddle point, which is the foundation of the iterative formulae of ADMM but is seldom addressed by other sparsity models.

The rest of the paper is organized as follows. Section 2 presents some preliminaries and related works. Section 3 illustrates the whole SSPO system and its solving algorithm. Section 4 presents extensive experimental results to evaluate SSPO. Finally, Section 5 draws conclusions.

2. Preliminaries and Related Works

2.1. Problem Setting of Short-term Portfolio Optimization

We present the framework of short-term PO via machine learning, which is taken as baseline by many previous researches (Cover, 1991; Agarwal et al., 2006; Li et al., 2015; Huang et al., 2016; Li et al., 2016). Suppose there are d assets in a financial market, their prices are collected as a vector $\mathbf{p}_t \in \mathbb{R}^d_+, t = 0, 1, 2, \cdots$, where $\mathbb{R}^d_+$ represents the d-dimensional nonnegative real space. Note that $\mathbf{p}_t$ may change as time t goes, forming a

sequence. We can evaluate the performance of assets by the price relative $\mathbf{x}_t \triangleq \frac{\mathbf{p}_t}{\mathbf{p}_{t-1}}$, where the division is performed element-wise.

A portfolio is a vector lying on the d-dimensional simplex $\mathbf{b}_t \in \Delta_d := \{\mathbf{b} \in \mathbb{R}^d_+ : \sum_{i=1}^d \mathbf{b}^{(i)} = 1\}$ with assumptions of non-short-selling and self-financing (without borrowing money and full re-investment). It represents the proportion of the total wealth invested in different assets at the beginning of the t-th period.

At the end of the t-th period, the cumulative wealth S_t evolves by an increasing factor of $\mathbf{b}_t^\top \mathbf{x}_t$: $S_t = S_{t-1} \cdot (\mathbf{b}_t^\top \mathbf{x}_t)$. Suppose the whole investment lasts n periods and the initial wealth is $S_0 = 1$, then the final cumulative wealth is $S_n = \prod_{t=1}^n (\mathbf{b}_t^\top \mathbf{x}_t)$. The objective of short-term PO is to maximize S_n by maximizing $\mathbf{b}_t^\top \mathbf{x}_t$ on each period (Cover, 1991; Agarwal et al., 2006; Li et al., 2011; Das et al., 2013; Li et al., 2015; Huang et al., 2016; Li et al., 2016):

$$\hat{S}_n = \max_{\{\mathbf{b}_t \in \Delta_d\}_{t=1}^n} \prod_{t=1}^n (\mathbf{b}_t^\top \mathbf{x}_t). \tag{1}$$

Note that no statistical assumptions regarding the movement of asset prices are required in the framework.

2.2. Related Works on Short-term Portfolio Optimization

Different systems or strategies suggest different principles of optimizing $\mathbf{b}_t$ as time passes. In general, short-term PO systems need to flexibly react to the rapid change of financial environments, thus they have to exploit some principles of empirical financial studies (Jegadeesh, 1990, 1991; Li and Hoi, 2014; Li et al., 2016) and investing behaviors (Kahneman and Tversky, 1979; Bondt and Thaler, 1985; Jegadeesh and Titman, 1993) to make future price predictions, rather than follow strict statistical assumptions that may only take effect in the long run (Das et al., 2013). For example, some state-of-the-art short-term PO systems adopt the mean reversion principle in finance (Jegadeesh, 1990, 1991; Li et al., 2012, 2013, 2015; Huang et al., 2016), which indicates that the future price of an asset will reverse to some kind of its historical mean.

2.2.1. Two Trivial Systems

A simple but widely used system is the Uniformly Buy-And-Hold (UBAH) strategy (Li and Hoi, 2014), which disperses the wealth equally in all the assets at the very beginning and remains unchanged: $\hat{S}_n^{UBAH} = \frac{1}{d} \sum_{i=1}^d \prod_{t=1}^n \mathbf{x}_t^{(i)}$. It is taken by most financial markets as their market strategies.

On the contrary, the Beststock (BS) strategy (Li and Hoi, 2014) allocates all the wealth to the best performing asset in the whole investment: $\hat{S}_n^{BS} = \max_{\mathbf{b} \in \Delta_d} \mathbf{b}^\top (\bigotimes_{t=1}^n \mathbf{x}_t)$, where $\bigotimes$ is the element-wise product operator. It is a hindsight strategy that cannot be implemented in reality.

2.2.2. Systems Based on Correlation

Anticor (Borodin et al., 2004) is a mean-reversion system, selling previous good performing assets and buying the anti-correlated bad performing ones. It measures the similarity

of different assets by the cross-window correlation $cor_{i,j} = \frac{(\mathbf{lx}^{(i)} - \bar{lx}^{(i)})^\top (\mathbf{ly}^{(j)} - \bar{ly}^{(j)})}{\|\mathbf{lx}^{(i)} - \bar{lx}^{(i)}\| \|\mathbf{ly}^{(j)} - \bar{ly}^{(j)}\|}$ and an anti-correlation $A_i = |cor_{i,i}|$ if $cor_{i,i} < 0$; else $A_i = 0$, where $\|\cdot\|$ is the Euclidean norm, $\mathbf{lx}$ and $\mathbf{ly}$ are logarithmic returns in two successive windows.

CORN (Li et al., 2011) further combines correlation with pattern-matching (Györfi et al., 2006) and searches for historical correlation-similar patterns

$$\mathcal{C}_{t+1}(w, \rho) = \left\{ w < k < t : \frac{cov(\mathbf{X}_{k-w}^{k-1}, \mathbf{X}_{t-w+1}^{t})}{std(\mathbf{X}_{k-w}^{k-1}) std(\mathbf{X}_{t-w+1}^{t})} \geqslant \rho \right\},$$

where $\mathbf{X}_{k-w}^{k-1}$ denotes the price relative vectors in the time window $[k-w, k-1]$.

Mahdavi-Damghani et al. (2017) consider PO in the context of cointelated pairs, a typical model for pairs trading. It is designed to signify a hybrid method between the cointegration and the correlation models. There are two approaches to addressed this problem: Stochastic Differential Equation and Band-wise Gaussian Mixture, which give similar results but the latter keeps the methodology simpler and more adaptable to the regime change.

2.2.3. Systems Based on Average Indices

OLMAR (Li et al., 2015) exploits the popular financial tool of moving average (MA) to predict future asset prices. It is a defensive and moderate strategy with a neutral risk appetite, so as to avoid over-estimating or under-estimating:

$$\hat{\mathbf{x}}_{t+1}(w) = \frac{MA_t(w)}{\mathbf{p}_t} = \frac{\sum_{k=0}^{w-1} \mathbf{p}_{t-k}}{w \mathbf{p}_t} = \frac{1}{w} \left(\mathbf{1} + \frac{\mathbf{1}}{\mathbf{x}_t} + \cdots + \frac{\mathbf{1}}{\bigotimes_{k=0}^{w-2} \mathbf{x}_{t-k}} \right), \tag{2}$$

where $\mathbf{1}$ is a vector with elements of 1, whose dimension can be inferred from the context, and w is the window size. Its portfolio update scheme is the same as the below-mentioned RMR strategy (Huang et al., 2016).

RMR also makes moderate price predictions, but substitutes ℓ^1-median (Vardi and Zhang, 2000) for MA, since ℓ^1-median is more robust to noise and outliers:

$$\hat{\mathbf{p}}_{t+1} = \operatorname*{argmin}_{\mathbf{p} \in \mathbb{R}_+^d} \sum_{k=0}^{w-1} \|\mathbf{p}_{t-k} - \mathbf{p}\|, \quad \hat{\mathbf{x}}_{t+1} = \frac{\hat{\mathbf{p}}_{t+1}}{\mathbf{p}_t}. \tag{3}$$

Both RMR and OLMAR use the following optimization model to update portfolio:

$$\mathbf{b}_{t+1} = \operatorname*{argmin}_{\mathbf{b}} \frac{1}{2} \|\mathbf{b} - \hat{\mathbf{b}}_t\|^2 \quad \text{s.t. } \mathbf{b}^\top \hat{\mathbf{x}}_{t+1} \geqslant \epsilon > 0, \tag{4}$$

$$\hat{\mathbf{b}}_{t+1} = \operatorname*{argmin}_{\mathbf{b} \in \Delta_d} \|\mathbf{b} - \mathbf{b}_{t+1}\|^2. \tag{5}$$

(5) is a normalization to the simplex (Duchi et al., 2008). This optimization model does not impose sparsity constraints on the portfolio, and it tries to approach the current portfolio $\hat{\mathbf{b}}_t$ with a prospective growth $\mathbf{b}^\top \hat{\mathbf{x}}_{t+1} \geqslant \epsilon$.

2.3. Related Works on Sparsity Constraints

Although sparsity models have been proposed for PO, they either aim at the long-term PO or constrain the quantity of the portfolio change. Very few of them aim to construct sparse portfolios for the short-term PO.

2.3.1. Systems for Long-term Portfolio Optimization

Sparse and Stable Markowitz Portfolio (SSMP) (Brodie et al., 2009) uses the following regularization model

$$\mathbf{b}_{SSMP} = \operatorname*{argmin}_{\mathbf{b}} \|\epsilon \mathbf{1}_{(n)} - \mathcal{R}\mathbf{b}\|^2 + \tau \|\mathbf{b}\|_1 \quad \text{s.t. } \mathbf{b}^\top \boldsymbol{\mu} = \epsilon, \mathbf{b}^\top \mathbf{1} = 1, \tag{6}$$

where ϵ is a predefined prospective growth rate, $\mathbf{1}_{(n)}$ is an n-dimensional vector of 1, $\mathcal{R}$ is an $n \times d$-dimensional asset return matrix, $\boldsymbol{\mu}$ is the expectation of asset returns, $\|\cdot\|_1$ denotes the ℓ^1-Norm, and τ is the regularization strength. $\mathcal{R}\mathbf{b}$ contains the actual portfolio returns of n samples, while $\mathbf{b}^\top \boldsymbol{\mu}$ is the expectation of the portfolio return. (6) constrains $\mathbf{b}^\top \boldsymbol{\mu}$ to a predefined level ϵ, and tries to minimize the square error (i.e., the quadratic risk in the context of quantitative finance) of the actual portfolio returns to their expectation. Besides, it relaxes the nonnegativity constraint of $\mathbf{b}$ and retains the self-financing constraint $\mathbf{b}^\top \mathbf{1} = 1$. With ℓ^1-regularization, $\mathbf{b}$ is forced to be sparse and the short position is limited (Brodie et al., 2009).

Weighted Elastic Net Penalized Portfolio (WENPP) (Ho et al., 2015) adds an elastic net penalization (Zou and Hastie, 2005) but removes the self-financing constraint $\mathbf{b}^\top \mathbf{1} = 1$:

$$\mathbf{b}_{WENPP} = \operatorname*{argmin}_{\mathbf{b}} \mathbf{b}^\top \hat{\boldsymbol{\Sigma}} \mathbf{b} - \mathbf{b}^\top \hat{\boldsymbol{\mu}} + \sum_{i=1}^{d} \tau_i |\mathbf{b}^{(i)}| + \sum_{i=1}^{d} \iota_i |\mathbf{b}^{(i)}|^2, \tag{7}$$

where $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ are the estimated covariance and expectation of asset returns, respectively.

SSMP and WENPP reflect that the objective of long-term PO is to minimize the quadratic risk $\|\epsilon \mathbf{1}_{(n)} - \mathcal{R}\mathbf{b}\|^2$ or $\mathbf{b}^\top \hat{\boldsymbol{\Sigma}} \mathbf{b}$, which is quite different from the problem setting (1) of short-term PO. Besides, $\hat{\boldsymbol{\Sigma}}$ and $\hat{\boldsymbol{\mu}}$ have to be estimated in a single static period, which is not adaptive to the rapid changing financial environments in the short-term PO (Das et al., 2013).

2.3.2. Systems for Lazy Updates

Online Lazy Update (OLU) (Das et al., 2013) is a sparsity model for the short-term PO. However, the sparsity is on the change of portfolio, not on the portfolio itself:

$$\hat{\mathbf{b}}_{t+1} = \operatorname*{argmin}_{\mathbf{b} \in \Delta_d} -\eta \log(\mathbf{b}^\top \mathbf{x}_t) + \tau \|\mathbf{b} - \hat{\mathbf{b}}_t\|_1 + \frac{1}{2} \|\mathbf{b} - \hat{\mathbf{b}}_t\|^2. \tag{8}$$

Hence it is not really a sparse PO. Besides, it assumes that the current price relative $\mathbf{x}_t$ will be replicated in the next day, which may lack supports from empirical financial studies. Moreover, although it has established an ADMM solver, it has not given a saddle point proof for its augmented Lagrangian, which is the foundation of the iterative formulae of ADMM. Note that not all ADMM designs have saddle points.

Doubly Regularized Portfolio (DRP) (Shen et al., 2014) also employs the lazy update strategy in its model:

$$\hat{\mathbf{b}}_t = \operatorname*{argmin}_{\mathbf{b}} \mathbf{b}^\top \hat{\mathbf{\Sigma}}_t \mathbf{b} + \tau_1 \|\mathbf{b}\|_1 + \tau_2 \|\mathbf{b} - \hat{\mathbf{b}}_{t-}\|^2 \quad \text{s.t. } \mathbf{b}^\top \mathbf{1} = 1, \tag{9}$$

where $\hat{\mathbf{b}}_{t-}$ denotes the re-normalized portfolio before the rebalancing at time t, and $\hat{\mathbf{\Sigma}}_t$ denotes the estimated covariance at time t. DRP is also for the long-term PO in both model setting (minimizing the quadratic risk) and its experimental evaluation (weekly or monthly data). It updates the covariance $\hat{\mathbf{\Sigma}}_t$ based on the newest price information, instead of using a single static $\hat{\mathbf{\Sigma}}$ as in (7). The regularization term $\tau_2 \|\mathbf{b} - \hat{\mathbf{b}}_{t-}\|^2$ is introduced to control the change of portfolio. DRP does not propose a solver for its model, but turns to a commercial software toolbox (Shen et al., 2014).

3. Short-term Sparse Portfolio Optimization

To make a summary of Section 2, there are few sparse portfolio methods for the short-term PO that have reliable public solving schemes, especially when both the ℓ^1-regularization term and the self-financing constraint are present. Besides, most state-of-the-art short-term PO systems focus only on exploiting empirical financial principles and pay little attention to constructing sparse, concentrated and effective portfolios.

To fill this gap, we propose a novel Short-term Sparse PO (SSPO) system based on ADMM. We also prove that its augmented Lagrangian has a saddle point, which is the foundation of the iterative formulae of ADMM but is seldom addressed by other sparsity models. SSPO actively rebalances the sparse portfolio according to some empirical financial principles in order to maximize the cumulative wealth. To establish SSPO, we follow the 3 conventional steps of short-term PO system design (Borodin et al., 2004; Li et al., 2011; Li and Hoi, 2014; Li et al., 2015; Huang et al., 2016): price information processing, sparse portfolio model setup, and solving algorithm design.

3.1. Price Information

A sparse portfolio should concentrate on only a few assets that have good increasing potential, which is more adaptive to an aggressive strategy than a defensive one. Hence SSPO exploits the irrational investing behaviors indicated by some empirical financial studies on stock price overreactions (Bondt and Thaler, 1985; Kahneman and Tversky, 1979; Shiller, 2003), instead of the mean reversion principle (Jegadeesh, 1991). For example, investors are usually irrational to keep on buying the assets rising in value, which further pushes up the asset prices and postpones the reversion (Jegadeesh and Titman, 1993; Shiller, 2000).

To evaluate the increasing potential of an asset, the highest price in a recent time window with size w is observed:

$$\mathbf{p}_{MAX}^{(i)} = \max_{0 \leqslant k \leqslant w-1} \mathbf{p}_{t-k}^{(i)}, \quad i = 1, 2, \cdots, d. \tag{10}$$

$\mathbf{p}_{MAX}^{(i)}$ plays an important role in real-world investment and portfolio management, and it is an indispensable indicator in nearly every stock analysis software. Since most investors

gain profits by the growth of asset price, they consider $\mathbf{p}_{MAX}^{(i)}$ as a potential level that the future price can probably reach.

Next, the relative distance from the current price vector $\mathbf{p}_t$ to $\mathbf{p}_{MAX}$ implies the increasing potential of the assets. Thus we define a generalized logarithmic return as follows:

$$\mathbf{R}_t = 1.1 \log\left(\frac{\mathbf{p}_{MAX}}{\mathbf{p}_t}\right) + 1, \tag{11}$$

where $\log\left(\frac{\mathbf{p}_{MAX}}{\mathbf{p}_t}\right)$ is the logarithmic return (Borodin et al., 2004), $\mathbf{R}_t$ is a linear transform of $\log\left(\frac{\mathbf{p}_{MAX}}{\mathbf{p}_t}\right)$. $\frac{\mathbf{p}_{MAX}}{\mathbf{p}_t} \geqslant \mathbf{1}$ and the inequality dominates each element. If $\frac{\mathbf{p}_{MAX}^{(i)}}{\mathbf{p}_t^{(i)}} = 1$, then $\mathbf{R}_t^{(i)} = 1$. The coefficient 1.1 is a slight adjustment of the shape of linear transform. We use $\mathbf{R}_t$ as the price information input for the SSPO system.

3.2. Sparse Portfolio Model

We consider the following objective to set up a sparse portfolio model: first, we should maximize $\mathbf{b}^\top \mathbf{R}_t$, which is the increasing potential of the whole portfolio. Denote $\boldsymbol{\varphi}_t = -\mathbf{R}_t$, then we can change the maximization to a minimization. Second, we adopt an ℓ^1-regularization term and a self-financing constraint simultaneously to concentrate the portfolio on a few assets. It leads to the following model

$$\mathbf{b}_{t+1} = \min_{\mathbf{b}} \mathbf{b}^\top \boldsymbol{\varphi}_t + \lambda \|\mathbf{b}\|_1, \quad \text{s.t. } \mathbf{1}^\top \mathbf{b} = 1, \tag{12}$$

where $\lambda > 0$ controls the regularization strength.

Different from previous compressed sensing models (Brodie et al., 2009; Mohsin et al., 2015; Zhan et al., 2016) that minimize a square error, model (12) minimizes $\mathbf{b}^\top \boldsymbol{\varphi}_t$. It is also different from (7) and (9) which minimize a quadratic risk for the long-term PO. Besides, (12) has a self-financing constraint $\mathbf{1}^\top \mathbf{b} = 1$ which is absent in (7), and (9) is solved by a commercial software (Shen et al., 2014). Thus there are few existing public algorithms to solve (12) and we have to design a new algorithm. Last, $\mathbf{b}_{t+1}$ can be projected onto the simplex to form an eligible portfolio, as instructed by Duchi et al. (2008); Li et al. (2015); Huang et al. (2016).

Based on the ADMM criterion, we introduce an auxiliary vector $\mathbf{g} \in \mathbb{R}^d$ to approach $\mathbf{b}$, and turn to minimize $\lambda \|\mathbf{g}\|_1$. We also introduce a dual variable ρ for the constraint $\mathbf{1}^\top \mathbf{b} = 1$, and transform the constraint into a penalty function. By this way, the constrained model (12) is changed to the optimization of an unconstrained augmented Lagrangian

$$L(\mathbf{b}, \mathbf{g}, \rho) = \mathbf{b}^\top \boldsymbol{\varphi}_t + \frac{\lambda}{2\gamma} \|\mathbf{b} - \mathbf{g}\|^2 + \lambda \|\mathbf{g}\|_1 + \frac{\eta}{2} (\mathbf{1}^\top \mathbf{b} - 1)^2 + \rho (\mathbf{1}^\top \mathbf{b} - 1), \tag{13}$$

where $\gamma > 0$ controls the approximation of $\mathbf{g}$ to $\mathbf{b}$, and $\eta > 0$ controls the penalty strength when $\mathbf{1}^\top \mathbf{b} \neq 1$. Generally speaking, if $\gamma \to 0$, then the penalty $\frac{\lambda}{2\gamma} \|\mathbf{b} - \mathbf{g}\|^2$ forces $\mathbf{g} \to \mathbf{b}$. Further analysis of γ will be given later in the optimization steps.

Not all formulations of ADMM have saddle points. Few methods take the bother to figure out and prove the existence of saddle points (Das et al., 2013; Mohsin et al., 2015; Zhan et al., 2016). But we can prove that the augmented Lagrangian (13) has a saddle point, which makes the iterative formulae (25)$\sim$(27) of ADMM appropriate.

3.3. Solving Algorithm

3.3.1. The Existence of Saddle Point

Our algorithm originates from the existence of a saddle point $(\mathbf{b}^*, \mathbf{g}^*, \rho^*)$ for the Lagrangian (13) such that

$$L(\mathbf{b}^*, \mathbf{g}^*, \rho) \leqslant L(\mathbf{b}^*, \mathbf{g}^*, \rho^*) \leqslant L(\mathbf{b}, \mathbf{g}, \rho^*), \quad \forall\, \mathbf{b}, \mathbf{g}, \rho. \tag{14}$$

We now prove that this saddle point really exists. First, for any given ρ, the following equation holds:

$$\min_{\mathbf{b},\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho) = \min_{\mathbf{b}} \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho). \tag{15}$$

It can be seen that

$$\min_{\mathbf{b},\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho) \leqslant \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho) \leqslant L(\mathbf{b}, \mathbf{g}, \rho). \tag{16}$$

Taking $\min_{\mathbf{b},\mathbf{g}}$ in all 3 terms of the above inequalities, we find that they are restricted to equalities, since the first term equals the last term. Besides, $\min_{\mathbf{b},\mathbf{g}} \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho) = \min_{\mathbf{b}} \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho)$, thus we deduce equation (15).

Next, we examine the following functions

$$l(\mathbf{b}, \rho) = \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho) = \mathbf{b}^\top \boldsymbol{\varphi}_t + \lambda H(\mathbf{b}) + \frac{\eta}{2}(\mathbf{1}^\top \mathbf{b} - 1)^2 + \rho(\mathbf{1}^\top \mathbf{b} - 1), \tag{17}$$

$$H(\mathbf{b}) \triangleq \min_{\mathbf{g}} \frac{1}{2\gamma}\|\mathbf{b} - \mathbf{g}\|^2 + \|\mathbf{g}\|_1 = \sum_{i=1}^{d} h(\mathbf{b}^{(i)}), \tag{18}$$

where $h(\mathbf{b}^{(i)})$ is the Huber function (Boyd et al., 2010)

$$h(\mathbf{b}^{(i)}) = \min_{\mathbf{g}^{(i)}} \frac{1}{2\gamma}(\mathbf{b}^{(i)} - \mathbf{g}^{(i)})^2 + |\mathbf{g}^{(i)}| = \begin{cases} \frac{|\mathbf{b}^{(i)}|^2}{2\gamma} & \text{if } |\mathbf{b}^{(i)}| \leqslant \gamma \\ |\mathbf{b}^{(i)}| - \frac{\gamma}{2} & \text{if } |\mathbf{b}^{(i)}| > \gamma \end{cases}. \tag{19}$$

It is a continuous function, which only needs to be verified at $|\mathbf{b}^{(i)}| = \gamma$ and it is obvious. Moreover, it is decreasing when $\mathbf{b}^{(i)} \leqslant 0$ and increasing when $\mathbf{b}^{(i)} > 0$, indicating that it is also a strictly convex function.

Suppose we have 2 vectors $\mathbf{b} \neq \mathbf{c}$. For any $0 < \theta < 1$,

$$\begin{aligned} H(\theta\mathbf{b} + (1-\theta)\mathbf{c}) &= \sum_{i=1}^{d} h(\theta\mathbf{b}^{(i)} + (1-\theta)\mathbf{c}^{(i)}) \\ &< \theta \sum_{i=1}^{d} h(\mathbf{b}^{(i)}) + (1-\theta) \sum_{i=1}^{d} h(\mathbf{c}^{(i)}) = \theta H(\mathbf{b}) + (1-\theta) H(\mathbf{c}). \end{aligned}$$

Hence $H(\mathbf{b})$ is also a strictly convex function.

Looking back to (17), $l(\mathbf{b}, \rho)$ is the Lagrangian of the following problem

$$\min_{\mathbf{b}} \mathbf{b}^\top \boldsymbol{\varphi}_t + \lambda H(\mathbf{b}) + \frac{\eta}{2}(\mathbf{1}^\top \mathbf{b} - 1)^2, \text{ s.t. } \mathbf{1}^\top \mathbf{b} = 1. \tag{20}$$

We have proven that $H(\mathbf{b})$ is strictly convex. It is apparent that $\mathbf{b}^\top \boldsymbol{\varphi}_t$ and $\frac{\eta}{2}(\mathbf{1}^\top \mathbf{b} - 1)^2$ are also convex on $\mathbf{b}$. Hence the whole function to be minimized in (20) is strictly convex. By Slater's theorem (Boyd and Vandenberghe, 2004), strong duality holds and there exists a saddle point $(\mathbf{b}^*, \rho^*)$ such that

$$l(\mathbf{b}^*, \rho) \leqslant l(\mathbf{b}^*, \rho^*) \leqslant l(\mathbf{b}, \rho^*), \quad \forall\, \mathbf{b}, \rho. \tag{21}$$

The last step is to find $\mathbf{g}^*$. From (18) we know that $\mathbf{g}$ is determined by $\mathbf{b}$. Specifically, $\mathbf{g}^*$ is the minimizer of $H(\mathbf{b}^*)$ in (18), which is a soft shrinkage (Tibshirani, 1996)

$$\mathbf{g}^* = \text{sign}(\mathbf{b}^*) \otimes (\text{abs}(\mathbf{b}^*) - \gamma\mathbf{1})^+, \tag{22}$$

where $(\cdot)^+$ denotes the positive part of a vector, which maps all the negative elements to 0 and retains all the nonnegative ones. $\text{abs}(\cdot)$, $\text{sign}(\cdot)$ are the absolute value function and the sign function implemented on each element of a vector, respectively. The soft shrinkage operator shrinks each element of $\mathbf{b}^*$ towards 0 by a step size of γ.

Therefore, from the first inequality of (21) we have

$$L(\mathbf{b}^*, \mathbf{g}^*, \rho) = l(\mathbf{b}^*, \rho) \leqslant l(\mathbf{b}^*, \rho^*) = L(\mathbf{b}^*, \mathbf{g}^*, \rho^*). \tag{23}$$

Furthermore, by (15), (17) and the second inequality of (21) we have

$$L(\mathbf{b}^*, \mathbf{g}^*, \rho^*) = l(\mathbf{b}^*, \rho^*) = \min_{\mathbf{b}} l(\mathbf{b}, \rho^*) = \min_{\mathbf{b}} \min_{\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho^*) = \min_{\mathbf{b},\mathbf{g}} L(\mathbf{b}, \mathbf{g}, \rho^*). \tag{24}$$

Combining (23) and (24), we prove that $(\mathbf{b}^*, \mathbf{g}^*, \rho^*)$ is a saddle point satisfying (14).

3.3.2. The Algorithm Based on ADMM

Based on the saddle point inequalities (14) and the ADMM criterion, the Lagrangian (13) should be minimized by $\mathbf{b}$,$\mathbf{g}$ and maximized by ρ, which can be formulated in 3 iterative steps

$$\mathbf{b}_{(o+1)} = \operatorname*{argmin}_{\mathbf{b}} L(\mathbf{b}, \mathbf{g}_{(o)}, \rho_{(o)}), \tag{25}$$

$$\mathbf{g}_{(o+1)} = \operatorname*{argmin}_{\mathbf{g}} L(\mathbf{b}_{(o+1)}, \mathbf{g}, \rho_{(o)}), \tag{26}$$

$$\rho_{(o+1)} = \rho_{(o)} + \eta(\mathbf{1}^\top \mathbf{b}_{(o+1)} - 1). \tag{27}$$

To solve (25), we can leave out all the constant terms that do not include $\mathbf{b}$ in (13), which leads to

$$\begin{aligned} \mathbf{b}_{(o+1)} &= \operatorname*{argmin}_{\mathbf{b}} \mathbf{b}^\top \boldsymbol{\varphi}_t + \frac{\lambda}{2\gamma}\mathbf{b}^\top \mathbf{I}\mathbf{b} - \frac{\lambda}{\gamma}\mathbf{b}^\top \mathbf{g}_{(o)} + \frac{\eta}{2}\mathbf{b}^\top \mathbf{1}\mathbf{1}^\top \mathbf{b} - (\eta - \rho_{(o)})\mathbf{b}^\top \mathbf{1} \\ &= \operatorname*{argmin}_{\mathbf{b}} \frac{1}{2}\mathbf{b}^\top \left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)\mathbf{b} + \mathbf{b}^\top \left[\boldsymbol{\varphi}_t - \frac{\lambda}{\gamma}\mathbf{g}_{(o)} - (\eta - \rho_{(o)})\mathbf{1}\right], \end{aligned} \tag{28}$$

where $\mathbf{I}$ is an identity matrix whose dimension can be inferred from the context.

Proposition 1 *There is a unique minimum of (28):*

$$\mathbf{b}_{(o+1)} = \left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)^{-1}\left[\frac{\lambda}{\gamma}\mathbf{g}_{(o)} + (\eta - \rho_{(o)})\mathbf{1} - \boldsymbol{\varphi}_t\right]. \tag{29}$$

Proof We analyze the quadratic form $\left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)$ in (28). Apparently $\frac{\lambda}{\gamma}\mathbf{I}$ is positive definite. It can be shown that $\eta\mathbf{1}\mathbf{1}^\top$ is positive semidefinite:

$$\forall \mathbf{b},\ \eta\mathbf{b}^\top\mathbf{1}\mathbf{1}^\top\mathbf{b} = \eta(\mathbf{b}^\top\mathbf{1})(\mathbf{1}^\top\mathbf{b}) = \eta(\mathbf{b}^\top\mathbf{1})^2 \geqslant 0.$$

Thus the whole quadratic form $\left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)$ is positive definite.

We take the gradient of (28) with respect to $\mathbf{b}$ and let it be a zero vector

$$\left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)\mathbf{b} + \boldsymbol{\varphi}_t - \frac{\lambda}{\gamma}\mathbf{g}_{(o)} - (\eta - \rho_{(o)})\mathbf{1} = \mathbf{0}. \tag{30}$$

Since $\left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)$ is positive definite, the function of (28) is convex and the solution of (30) is the unique minimum

$$\mathbf{b}_{(o+1)} = \left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)^{-1}\left[\frac{\lambda}{\gamma}\mathbf{g}_{(o)} + (\eta - \rho_{(o)})\mathbf{1} - \boldsymbol{\varphi}_t\right].$$

■

Next, we turn to solve (26), which has already been addressed in Section 3.3.1. By excluding all the terms unrelated to $\mathbf{g}$ in (13), we have a Huber vector function

$$H(\mathbf{b}_{(o+1)}) \quad = \quad \min_{\mathbf{g}} \frac{1}{2\gamma}\|\mathbf{b}_{(o+1)} - \mathbf{g}\|^2 + \|\mathbf{g}\|_1. \tag{31}$$

Now we can see that γ balances between the approximation of $\mathbf{g}$ to $\mathbf{b}_{(o+1)}$ and the sparsity of $\mathbf{g}$. When $\gamma \to 0$, $\mathbf{g} \to \mathbf{b}_{(o+1)}$ and the sparse regularization $\|\mathbf{g}\|_1$ will be weaken, vice versa. The minimizer of (31) is a soft shrinkage:

$$\mathbf{g}_{(o+1)} \quad = \quad \text{sign}(\mathbf{b}_{(o+1)}) \otimes (\text{abs}(\mathbf{b}_{(o+1)}) - \gamma\mathbf{1})^+. \tag{32}$$

Next, we implement (27) as a dual ascent step for (13), and turn to the next iteration. We repeat (29)(32)(27) until the equality tolerance $|\mathbf{1}^\top\mathbf{b}_{(o)} - 1| < \epsilon$ or the maximum iteration is reached. Last, $\mathbf{b}_{(o)}$ should be normalized to be a real portfolio output for the next trading period (Duchi et al., 2008):

$$\hat{\mathbf{b}}_{t+1} \quad = \quad \underset{\mathbf{b}\in\Delta_d}{\text{argmin}} \|\mathbf{b} - \zeta\mathbf{b}_{(o)}\|^2, \tag{33}$$

where $\zeta > 0$ is a scale parameter.

We summarize the whole SSPO as Algorithm 1.

ALGORITHM 1: Short-term Sparse Portfolio Optimization (SSPO)

Input: Asset prices in the recent time window $\{\mathbf{p}_{t-k}\}_{k=0}^{w-1}$, the current portfolio $\hat{\mathbf{b}}_t$, parameters w, λ, γ, η, ζ. Set the equality tolerance $\epsilon = 10^{-4}$ and the maximum iteration$= 10^4$.

1. $\mathbf{p}_{MAX}^{(i)} = \max_{0\leqslant k\leqslant w-1} \mathbf{p}_{t-k}^{(i)}$, $i = 1, 2, \cdots, d$.
2. $\boldsymbol{\varphi}_t = -1.1\log\left(\frac{\mathbf{p}_{MAX}}{\mathbf{p}_t}\right) - 1$.
3. Initialize: $o = 1$, $\mathbf{b}_{(1)} = \mathbf{g}_{(1)} = \hat{\mathbf{b}}_t$, $\rho_{(1)} = 0$.

repeat

4. $\mathbf{b}_{(o+1)} = \left(\frac{\lambda}{\gamma}\mathbf{I} + \eta\mathbf{1}\mathbf{1}^\top\right)^{-1}\left[\frac{\lambda}{\gamma}\mathbf{g}_{(o)} + (\eta - \rho_{(o)})\mathbf{1} - \boldsymbol{\varphi}_t\right]$.
5. $\mathbf{g}_{(o+1)} = \text{sign}(\mathbf{b}_{(o+1)}) \otimes (\text{abs}(\mathbf{b}_{(o+1)}) - \gamma\mathbf{1})^+$.
6. $\rho_{(o+1)} = \rho_{(o)} + \eta(\mathbf{1}^\top\mathbf{b}_{(o+1)} - 1)$.
7. $o = o + 1$.

until $|\mathbf{1}^\top\mathbf{b}_{(o)} - 1| < \epsilon$ or $o > Max\_Iter$

8. Normalize: $\hat{\mathbf{b}}_{t+1} = \text{argmin}_{\mathbf{b}\in\Delta_d} \|\mathbf{b} - \zeta\mathbf{b}_{(o)}\|^2$.

Output: The next portfolio $\hat{\mathbf{b}}_{t+1}$.

3.3.3. Sparsity of the Portfolio

We show that our algorithm really produces sparse portfolios. We compute the corresponding portfolio $\mathbf{b}_{t+1}$ by Algorithm 1 without normalization in Step 8 (to verify that the core procedure of the algorithm produces sparse portfolios) and test its sparsity. For a portfolio, we consider the weights which are no larger than 10% of the maximum weight as small weights. Then the sparsity of a portfolio is the proportion of the small weights in all the non-maximum weights:

$$spar = \frac{\#\{small\ weights\}}{d - 1}. \tag{34}$$

We compute the average sparsity of the portfolios of SSPO on each of the 5 experimental data sets NYSE(O) (Cover, 1991), NYSE(N) (Li et al., 2013), DJIA (Borodin et al., 2004), SP500 (Borodin et al., 2004) and TSE (Borodin et al., 2004), and show the results in Table 1. The parameters of SSPO are set in Section 4.1. SSPO achieves high sparsities that are near to or greater than 90% on all the data sets. It indicates that SSPO can produce sparse portfolios. Moreover, these sparse portfolios also achieve good investing performance, which is supported by the experimental results in Section 4.

NYSE(O)	NYSE(N)	DJIA	SP500	TSE
92.91%	89.06%	91.91%	91.36%	94.50%

Table 1: Average sparsity of the portfolios of SSPO on 5 benchmark data sets.

To visualize the sparsity, we randomly choose 6 portfolios (the 2-nd,33-rd,450-th,276-th,127-th and 98-th investing days) of SSPO on the DJIA data set and show them in Figure 1. We can see that the portfolios are sparse and concentrate on only a few assets. For

example, in the first figure, the portfolio weight of the 29-th asset is much larger than other weights. In the second figure, the portfolio weights of the 18-th and the 26-th assets are much larger than other weights. Portfolios on other investing days are also sparse but we need not exhaustively show them all.

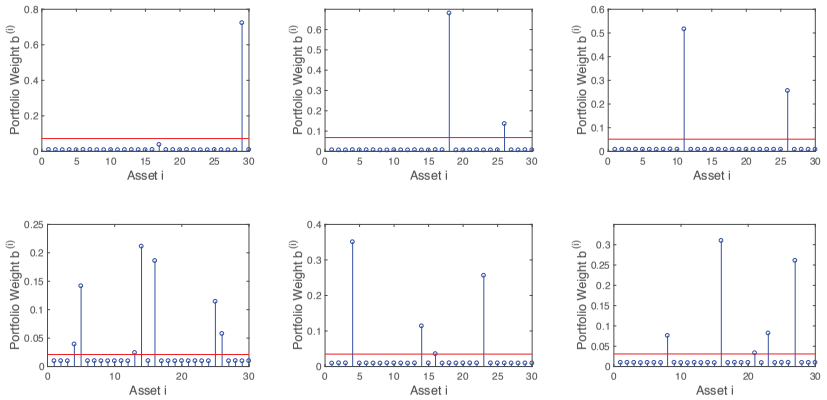


Figure 1: Portfolios produced by SSPO on 6 random investing days (2-nd,33-rd,450-th,276-th,127-th,98-th) of the DJIA data set. The red horizontal line separates small weights from large weights. The portfolios are sparse and concentrate on only a few assets. For example, in the first figure, the portfolio weight of the 29-th asset dominates others. In the second figure, the portfolio weights of the 18-th and the 26-th assets dominate others.

4. Experimental Results

We conduct extensive experiments on 5 benchmark data sets from real-world stock markets: NYSE(O) (Cover, 1991), NYSE(N) (Li et al., 2013), DJIA (Borodin et al., 2004), SP500 (Borodin et al., 2004), and TSE (Borodin et al., 2004). They consist of daily price relatives from New York Stock Exchange, Dow Jones Industrial Average, Standard & Pool 500 and Toronto Stock Exchange, covering a wide range of assets and time spans. Their information is shown in Table 2. To be consistent across the experiments, all the involved returns and increasing factors in this section are daily but not annualized.

We also show the box plots of asset price relatives on the 5 benchmark data sets in Figure 2. Note that if Asset i does not change in price at day (period) t, then $\mathbf{x}_t^{(i)} = \frac{\mathbf{p}_t^{(i)}}{\mathbf{p}_{t-1}^{(i)}} = 1$, which is the standard level of a price relative. The figure shows that all the benchmark data sets have a representative asset pool with diverse characteristics of return and volatility. Hence

Data Set	Region	Time	Days	Stocks
NYSE(O)	US	3/7/1962 ∼ 31/12/1984	5651	36
NYSE(N)	US	1/1/1985 ∼ 30/6/2010	6431	23
DJIA	US	14/1/2001 ∼ 14/1/2003	507	30
SP500	US	2/1/1998 ∼ 31/1/2003	1276	25
TSE	CA	4/1/1994 ∼ 31/12/1998	1259	88

Table 2: Information of 5 benchmark data sets from real-world stock markets.

they are reliable to test the performance of different PO systems in real-world financial environments.

For comparison, 7 state-of-the-art short-term PO systems are evaluated: ONS (Agarwal et al., 2006), CORN, Anticor, PAMR (Li et al., 2012), CWMR (Li et al., 2013), OLMAR and RMR, as well as 2 trivial ones: Beststock and Market. The parameters for these systems are set by the defaults in their original papers and previous experiments (Li and Hoi, 2014; Li et al., 2016, 2015; Huang et al., 2016): ONS: $\eta = 0, \beta = 1, \gamma = \frac{1}{8}$; CORN: $w = 5, P = 1, \rho = 0.1$; Anticor: $w = 5$; PAMR and CWMR: $\epsilon = 0.5$; OLMAR: $w = 5, \epsilon = 10$; RMR: $w = 5, \epsilon = 5$. The parameters for SSPO are set as: $w = 5, \lambda = 0.5, \gamma = 0.01, \eta = 0.005, \zeta = 500$. Note that the window size $w = 5$ is the same with all these systems, to be consistent.

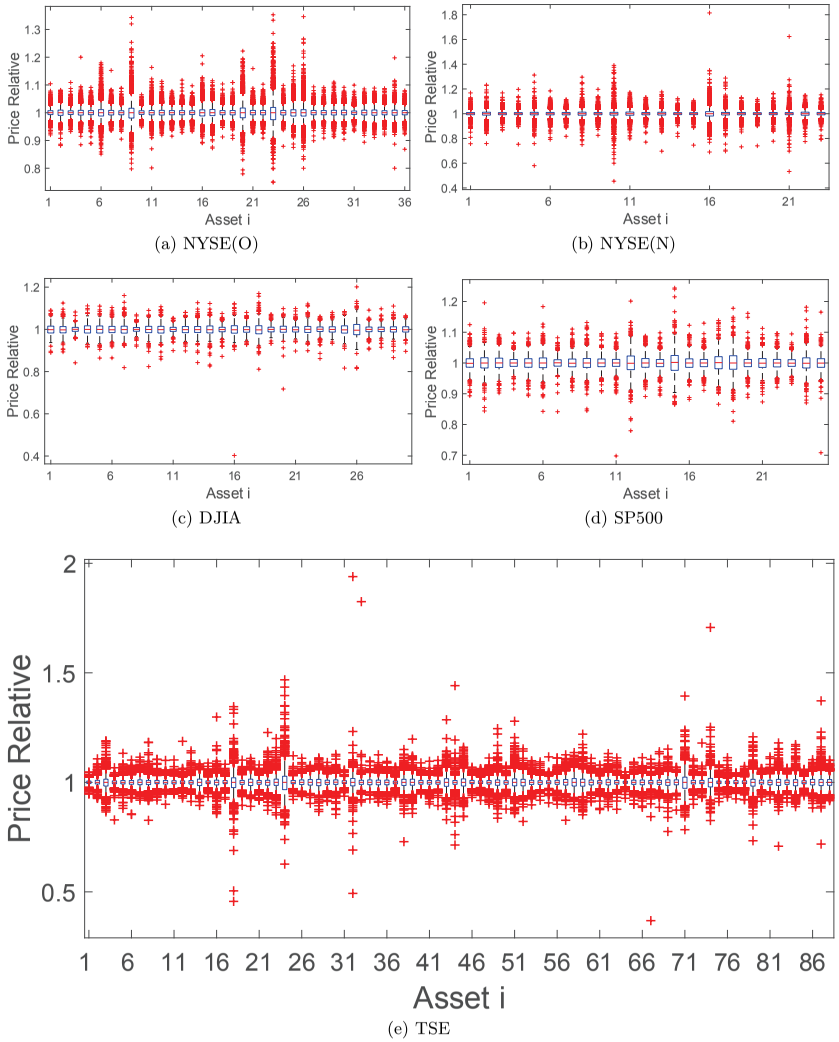
Evaluation protocols mainly fall into 8 indicators: 1. Cumulative wealth (CW): the main score to evaluate investing performance; 2. Mean Excess Return (MER) (Jegadeesh, 1990): the average excess performance of a system compared with the market; 3. α Factor (Lintner, 1965): MER excluding the market risk; 4. Order statistics: the rank of a system return in all the asset returns; 5. Sharpe Ratio (Sharpe, 1966): risk-adjusted average return; 6. Information Ratio (Treynor and Black, 1973): risk-adjusted MER; 7. Transaction costs; 8. Running times. Indicators 1∼3 are investing performance measurements, Indicators 4∼6 are risk metrics, and Indicators 7,8 evaluate the applicability to real-world financial environments. SSPO achieves state-of-the-art results in all these indicators.

4.1. Parameter Setting

We first conduct experiments of the final cumulative wealth (CW) to empirically set the parameters for SSPO, which is similar to Borodin et al. (2004); Li et al. (2011, 2015); Huang et al. (2013, 2016). The final CW is the cumulative product of the actual increasing factors $\{\hat{\mathbf{b}}_t^\top \mathbf{x}_t\}_{t=1}^n$ with the initial wealth $S_0 = 1$.

First, $w = 5$ is a common window size in real-world stock trading, especially in short-term investment. It is also consistent with the compared state-of-the-art methods. Hence it is adopted for SSPO. Second, we fix $w = 5$, $\gamma = 0.01$, $\eta = 0.005$, $\zeta = 500$ and change λ in $0.4 \sim 0.65$. The results in Figure 3 show that $\lambda = 0.5$ leads to good performance in general and thus is set for SSPO. To better show the geometric scaling effect of the daily geometric increasing factor on the final CW, the power form is used to label the y-axis. The base is the daily geometric increasing factor while the exponent is the total number of trading days in the data set.

At each time, we change one parameter and fix other parameters, and the results are shown in Figure 4, Figure 5, and Figure 6. They indicate that $\gamma = 0.01$, $\eta = 0.005$, and $\zeta = 500$ are suitable parameters for SSPO.

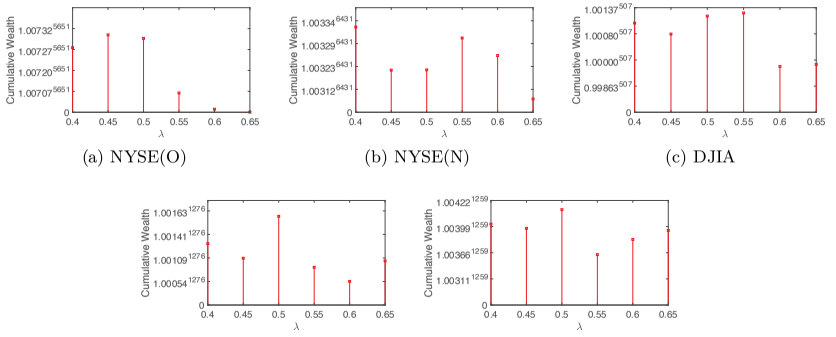


(a) NYSE(O) (b) NYSE(N)

(c) DJIA (d) SP500

(e) TSE

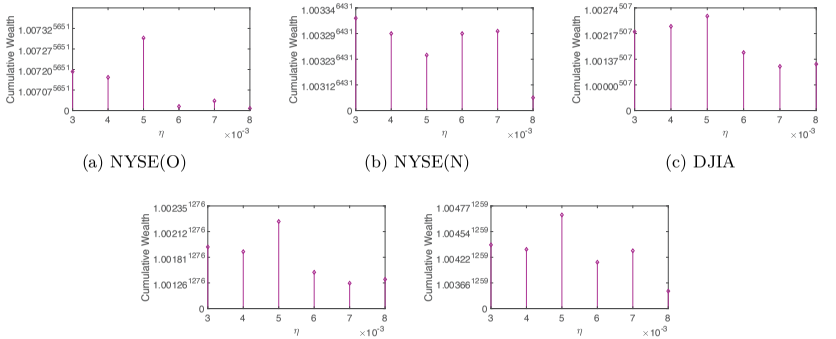
Figure 2: Box plots of asset price relatives on 5 benchmark data sets.



(a) NYSE(O) (b) NYSE(N) (c) DJIA

(d) SP500 (e) TSE

Figure 3: Final cumulative wealths of SSPO with respect to λ on 5 benchmark data sets (fix $w = 5$, $\gamma = 0.01$, $\eta = 0.005$, $\zeta = 500$). The power form is used to label the y-axis to better show the geometric scaling effect of the daily geometric increasing factor on the final cumulative wealth. The base is the daily geometric increasing factor while the exponent is the total number of trading days in the data set.



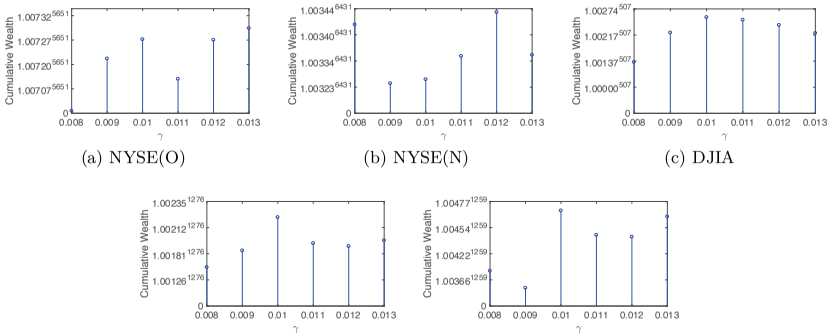
(a) NYSE(O) (b) NYSE(N) (c) DJIA

(d) SP500 (e) TSE

Figure 4: Final cumulative wealths of SSPO with respect to η on 5 benchmark data sets (fix $w = 5$, $\gamma = 0.01$, $\lambda = 0.5$, $\zeta = 500$).

4.2. Cumulative Wealth

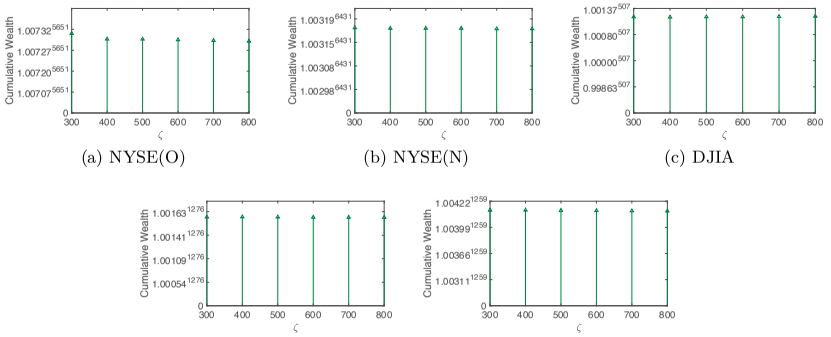
We show the final CWs of different PO systems on 5 benchmark data sets in Table 3. The approximate power form is used to make an intuitive sense of the geometric scaling effect



(a) NYSE(O) (b) NYSE(N) (c) DJIA

(d) SP500 (e) TSE

Figure 5: Final cumulative wealths of SSPO with respect to γ on 5 benchmark data sets (fix $w = 5$, $\eta = 0.005$, $\lambda = 0.5$, $\zeta = 500$).



(a) NYSE(O) (b) NYSE(N) (c) DJIA

(d) SP500 (e) TSE

Figure 6: Final cumulative wealths of SSPO with respect to ζ on 5 benchmark data sets (fix $w = 5$, $\eta = 0.005$, $\lambda = 0.5$, $\gamma = 0.01$).

of the daily geometric increasing factor on the final CW. The base is the daily geometric increasing factor while the exponent is the total number of trading days in the data set. SSPO achieves the best performance on all the data sets against other state-of-the-art systems. For instance, SSPO achieves a much higher CW (3.68) than OLMAR (2.54) and RMR (2.67) on DJIA which is difficult to handle (Huang et al., 2016). On NYSE(O), NYSE(N) and TSE, SSPO achieves CW= $1.06E{+}18$, $1.62E{+}9$ and 364.94, respectively,

which are much higher than OLMAR ($7.21E{+}16$, $4.14E{+}8$ and 58.51) and RMR ($1.64E{+}17$, $3.25E{+}8$ and 181.34). Therefore, SSPO is an efficient short-term PO system for diverse real-world stock markets. We also plot the CW evolution paths for different PO systems on DJIA in Figure 7. On most days, the SSPO plot is over other systems, and it climbs up significantly when there are good opportunities.

	NYSE(O)	NYSE(N)	DJIA	SP500	TSE
Market	$14.50 \approx 1.00047^{5651}$	$18.06 \approx 1.00045^{6431}$	$0.76 \approx 0.99947^{507}$	$1.34 \approx 1.00023^{1276}$	$1.61 \approx 1.00038^{1259}$
Beststock	$54.14 \approx 1.00071^{5651}$	$83.51 \approx 1.00069^{6431}$	$1.19 \approx 1.00034^{507}$	$3.78 \approx 1.00104^{1276}$	$6.28 \approx 1.00146^{1259}$
ONS	$109.19 \approx 1.00083^{5651}$	$21.59 \approx 1.00048^{6431}$	$1.53 \approx 1.00084^{507}$	$3.34 \approx 1.00095^{1276}$	$1.62 \approx 1.00038^{1259}$
CORN	$8.09E{+}11 \approx 1.00486^{5651}$	$2.33E{+}5 \approx 1.00192^{6431}$	$0.78 \approx 0.99951^{507}$	$5.29 \approx 1.00131^{1276}$	$9.80 \approx 1.00018^{1259}$
Anticor	$2.04E{+}7 \approx 1.00298^{5651}$	$2.11E{+}5 \approx 1.00191^{6431}$	$1.63 \approx 1.00096^{507}$	$5.61 \approx 1.00135^{1276}$	$28.68 \approx 1.00267^{1259}$
PAMR	$5.14E{+}15 \approx 1.00642^{5651}$	$1.25E{+}6 \approx 1.00219^{6431}$	$0.68 \approx 0.99924^{507}$	$5.09 \approx 1.00128^{1276}$	$264.86 \approx 1.00444^{1259}$
CWMR	$6.49E{+}15 \approx 1.00646^{5651}$	$1.41E{+}6 \approx 1.00220^{6431}$	$0.69 \approx 0.99926^{507}$	$5.95 \approx 1.00140^{1276}$	$332.62 \approx 1.00462^{1259}$
OLMAR	$7.21E{+}16 \approx 1.00689^{5651}$	$4.14E{+}8 \approx 1.00309^{6431}$	$2.54 \approx 1.00184^{507}$	$15.94 \approx 1.00217^{1276}$	$58.51 \approx 1.00324^{1259}$
RMR	$1.64E{+}17 \approx 1.00704^{5651}$	$3.25E{+}8 \approx 1.00305^{6431}$	$2.67 \approx 1.00194^{507}$	$8.28 \approx 1.00166^{1276}$	$181.34 \approx 1.00414^{1259}$
SSPO	$\mathbf{1.06E{+}18 \approx 1.00737^{5651}}$	$\mathbf{1.62E{+}9 \approx 1.00330^{6431}}$	$\mathbf{3.68 \approx 1.00257^{507}}$	$\mathbf{16.97 \approx 1.00222^{1276}}$	$\mathbf{364.94 \approx 1.00470^{1259}}$

Table 3: Final cumulative wealths of portfolio optimization systems on 5 benchmark data sets. The approximate power form is used to make an intuitive sense of the geometric scaling effect of the daily geometric increasing factor on the final CW. The base is the daily geometric increasing factor while the exponent is the total number of trading days in the data set.

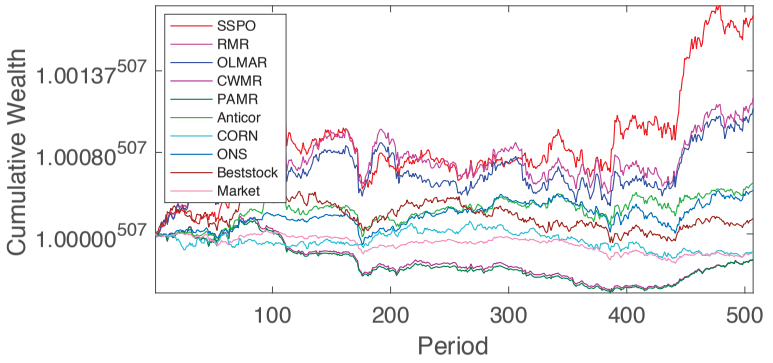


Figure 7: Cumulative wealth evolution paths of portfolio optimization systems on DJIA. The length of one period is one day.

4.3. Mean Excess Return

At the end of each day, we want to know what proportion of total wealth gained or lost on this day. This concept can be represented by the term "return": $r_t = \hat{\mathbf{b}}_t^\top \mathbf{x}_t - 1$.

Mean Excess Return (MER) (Jegadeesh, 1990) is the average daily excess return of a system compared with the market in the long run:

$$MER = \bar{r}_s - \bar{r}_m = \frac{1}{n}\sum_{t=1}^{n}(r_{s,t} - r_{m,t}), \tag{35}$$

where $r_{s,t}$ and $r_{m,t}$ are daily returns of a PO system and the market on the t-th day, respectively. It is more worthy of implementing a PO system with a higher MER. Even a small difference of MER leads to a large gap of CW in the long run, due to the geometric scaling effect.

We present the MERs for different PO systems in Table 4. SSPO outperforms other state-of-the-art systems on all the data sets. For instance, the MERs of SSPO (0.0036 and 0.0060) are much higher than those of OLMAR (0.0028 and 0.0045) and RMR (0.0029 and 0.0053) on DJIA and TSE, respectively. This is the reason why SSPO outperforms other systems in cumulative wealth in the long run.

System	NYSE(O)	NYSE(N)	DJIA	SP500	TSE
Beststock	0.0003	0.0003	0.0011	0.0012	0.0016
ONS	0.0004	0.0003	0.0015	0.0007	0.0002
CORN	0.0049	0.0017	0.0002	0.0014	0.0020
Anticor	0.0026	0.0016	0.0016	0.0013	0.0027
PAMR	0.0064	0.0021	0.0001	0.0014	0.0050
CWMR	0.0064	0.0022	0.0001	0.0015	0.0053
OLMAR	0.0070	0.0032	0.0028	0.0024	0.0045
RMR	0.0071	0.0032	0.0029	0.0019	0.0053
SSPO	**0.0076**	**0.0035**	**0.0036**	**0.0025**	**0.0060**

Table 4: Mean excess returns of portfolio optimization systems on 5 benchmark data sets.

4.4. α Factor

According to the Capital Asset Pricing Model (CAPM) (Sharpe, 1964), the expected return of a system can be decomposed into 2 parts: the market return component, and the intrinsic excess return usually called the α Factor in the finance industry (Lintner, 1965). The former is determined by the market environment, which cannot be improved by any active investing strategy or PO system, while the latter can be improved by a good PO system. The α Factor can be represented as follows:

$$E(r_s) = \beta E(r_m) + \alpha, \tag{36}$$

$$\hat{\beta} = \frac{\hat{c}(r_s, r_m)}{\hat{\sigma}^2(r_m)}, \quad \hat{\alpha} = \bar{r}_s - \hat{\beta}\bar{r}_m, \tag{37}$$

where $E(\cdot)$ denotes the mathematical expectation, $\hat{c}(\cdot,\cdot)$ and $\hat{\sigma}(\cdot)$ are the sample covariance and the sample standard deviation (STD) computed on n trading days, respectively. In finance, STD is a common tool to measure risk (volatility).

The α Factors of different PO systems are given in Table 5. SSPO outperforms other PO systems on all the data sets, indicating that SSPO achieves high intrinsic excess returns despite the market volatility. Particularly on DJIA data set which is difficult to handle, SSPO achieves $\alpha = 0.0037$, which is much higher than OLMAR (0.0029) and RMR (0.0030).

In the real-world portfolio management, it is common to perform a right-tailed t-test to see whether α is significantly > 0. If so, it indicates that the good performance of the system is not due to luck (Grinold and Kahn, 1999; Li et al., 2015; Huang et al., 2013, 2016). According to the results in Table 5, SSPO achieves significantly better performance than the market at a high confidence level of 99% (with all p-values < 0.01).

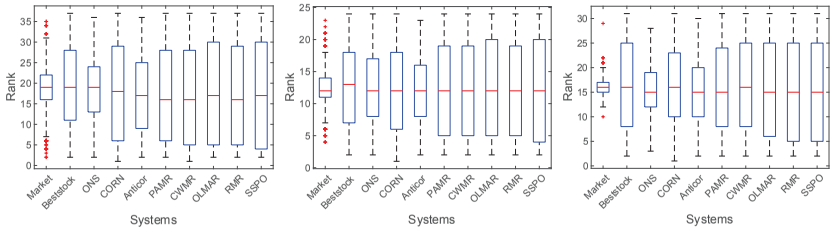
System	NYSE(O)		NYSE(N)		DJIA		SP500		TSE	
	α	p-Value	α	p-Value	α	p-Value	α	p-Value	α	p-Value
Beststock	0.0003	0.0195	0.0004	0.0176	0.0012	0.0838	0.0011	0.0593	0.0014	0.0606
ONS	0.0005	< 0.0001	0.0003	0.1257	0.0016	< 0.0001	0.0008	0.0040	0.0002	0.3514
CORN	0.0048	< 0.0001	0.0016	< 0.0001	0.0002	0.4077	0.0013	0.0190	0.0018	0.0278
Anticor	0.0026	< 0.0001	0.0016	< 0.0001	0.0017	0.0010	0.0012	0.0021	0.0025	< 0.0001
PAMR	0.0063	< 0.0001	0.0021	< 0.0001	0.0002	0.4275	0.0013	0.0259	0.0048	< 0.0001
CWMR	0.0063	< 0.0001	0.0021	< 0.0001	0.0002	0.4168	0.0014	0.0170	0.0051	< 0.0001
OLMAR	0.0069	< 0.0001	0.0031	< 0.0001	0.0029	0.0065	0.0023	0.0017	0.0042	0.0050
RMR	0.0070	< 0.0001	0.0031	< 0.0001	0.0030	0.0054	0.0018	0.0102	0.0051	< 0.0001
SSPO	**0.0074**	< 0.0001	**0.0034**	< 0.0001	**0.0037**	0.0009	**0.0024**	0.0019	**0.0058**	< 0.0001

Table 5: α factors (with p-values of t-tests) of portfolio optimization systems on 5 benchmark data sets.

4.5. Order Statistics

One may worry about that a PO system just simply selects the lucky assets with the highest growth. To check this point, we can compare the system return $r_{s,t} = \hat{\mathbf{b}}_t^\top \mathbf{x}_t - 1$ with all the asset returns $r_t^{(i)} = \mathbf{x}_t^{(i)} - 1$, $i = 1, \cdots, d$ (d is the number of assets in a data set). By sorting the $(d+1)$ returns $\{r_{s,t}\} \bigcup \{r_t^{(i)}\}_{i=1}^d$ in the descending order, we can obtain the rank of $r_{s,t}$. If a PO system always chooses the lucky assets, then $r_{s,t}$ will always rank very high in the investment.

We give a summary description of the rank of $r_{s,t}$ in the whole investment (n days) on each data set for different PO systems, shown in Figure 8 and Table 6. It can be seen that the ranks for SSPO cover a wide range instead of concentrating on the highest ranks. Besides, the mean rank and the median rank of SSPO are close to the middle rank $(d+1)/2$ on each data set, indicating that SSPO is at the average return level of all the asset returns. Furthermore, the rank results of SSPO are similar to other state-of-the-art short-term PO systems, which means that SSPO has a similar return level to the related systems. To summarize, SSPO does not simply select the lucky assets with the highest growth and its returns are credible. Note that this rank test is different from the rank-dependent portfolio (Fernholz, 2002) that takes the rank information as prior knowledge and exploits it to formulate PO models.



(a) NYSE(O) (b) NYSE(N) (c) DJIA

(d) SP500 (e) TSE

Figure 8: Box plots of the ranks for the system returns $r_{s,t}$ on 5 benchmark data sets. The ranks for SSPO cover a wide range instead of concentrating on the highest ranks.

System	NYSE(O): 37		NYSE(N): 24		DJIA: 31		SP500: 26		TSE: 89	
	Median	Mean	Median	Mean	Median	Mean	Median	Mean	Median	Mean
Market	19	18.63	12	12.25	16	16.12	13	13.34	43	42.88
Beststock	19	19.34	13	12.76	16	16.32	14	14.05	52	47.09
ONS	19	18.56	12	12.42	15	15.35	13	13.01	45	44.83
CORN	18	18.08	12	12.10	16	16.04	13	13.26	46	44.87
Anticor	17	17.25	12	11.72	15	15.32	13	12.98	39	41.77
PAMR	16	17.27	12	12.22	15	16.05	13	13.22	35	40.11
CWMR	16	17.22	12	12.19	16	16.09	13	13.18	35	40.01
OLMAR	17	17.56	12	12.26	15	15.45	12	13.31	43	43.12
RMR	16	17.40	12	12.26	15	15.34	13	13.40	40	42.18
SSPO	17	17.71	12	12.25	15	15.25	12	13.33	44	43.03

Table 6: Medians and means of the ranks for the system returns $r_{s,t}$ on 5 benchmark data sets. The value of $(d+1)$ is also given behind each data set name. The mean rank and the median rank of SSPO are close to the middle rank $(d+1)/2$ on each data set, which indicates that SSPO does not simply select the lucky assets with the highest growth.

4.6. Sharpe Ratio

In general, when an investor pursues high return, he/she should be ready to undertake high risk. Thus he/she has to balance between return and risk all the time. Sharpe Ratio (SR) (Sharpe, 1966) is such a measurement to meet this demand based on CAPM:

$$SR = \frac{\bar{r}_s - r_f}{\hat{\sigma}(r_s)}, \tag{38}$$

where r_f is the daily return of some risk-free asset, which is not considered in this paper. Hence we set $r_f = 0$ to make SR a daily risk-adjusted return.

We compute and present the daily SRs of different PO systems in Table 7. SSPO achieves the highest SR on NYSE(N) and DJIA, and is close to the best system on other data sets. Besides, SSPO is competitive to OLMAR and RMR. For example, SSPO achieves $SR = 0.0791$ on SP500 compared with RMR (0.0659), and achieves $SR = 0.1054$ on TSE compared with OLMAR (0.0820). It indicates that SSPO has a good ability in balancing between return and risk on the premise of good investing achievement with high CWs.

System	NYSE(O)		NYSE(N)		DJIA		SP500		TSE	
	SR	IR	SR	IR	SR	IR	SR	IR	SR	IR
Market	0.0549	Null	0.0458	Null	−0.0273	Null	0.0224	Null	0.0491	Null
Beststock	0.0536	0.0241	0.0472	0.0225	0.0253	0.0560	0.0485	0.0468	0.0579	0.0490
ONS	0.0789	0.0424	0.0313	0.0129	0.0504	**0.1574**	0.0701	0.0582	0.0281	0.0093
CORN	0.1635	0.1570	0.0968	0.0859	−0.0097	0.0108	0.0581	0.0618	0.0675	0.0595
Anticor	0.1720	0.1743	0.0973	0.0899	0.0535	0.1255	0.0682	0.0834	0.1061	0.0993
PAMR	0.2149	0.2121	0.0864	0.0763	−0.0115	0.0036	0.0568	0.0573	**0.1182**	**0.1131**
CWMR	**0.2169**	**0.2143**	0.0858	0.0758	−0.0106	0.0047	0.0606	0.0623	0.1179	0.1129
OLMAR	0.2102	0.2071	0.1038	0.0958	0.0731	0.1058	**0.0806**	**0.0848**	0.0820	0.0769
RMR	0.2153	0.2123	0.1033	0.0953	0.0763	0.1092	0.0659	0.0678	0.0981	0.0932
SSPO	**0.2073**	**0.2041**	**0.1060**	**0.0979**	**0.0919**	**0.1304**	**0.0791**	**0.0840**	**0.1054**	**0.1009**

Table 7: Daily Sharpe Ratios (SR) and daily Information Ratios (IR) of portfolio optimization systems on 5 benchmark data sets.

4.7. Information Ratio

Different from SR, Information Ratio (IR) (Treynor and Black, 1973) directly measures the daily risk-adjusted excess return of a system compared with the market, which can be seen as a combination of MER and SR. It is also worth reference for the concern with risk.

$$IR = \frac{(\bar{r}_s - \bar{r}_m)}{\hat{\sigma}(r_s - r_m)}. \tag{39}$$

We compute the daily IRs of different PO systems in Table 7. SSPO achieves the highest IR on NYSE(N) and is close to the best system on other data sets. Moreover, SSPO is competitive to OLMAR and RMR. For example, SSPO achieves $IR = 0.0840, 0.1009$ on SP500 and TSE, respectively, while RMR achieves $IR = 0.0678$ on SP500 and OLMAR achieves $IR = 0.0769$ on TSE. It shows the robustness of SSPO to risk on the premise of good investing performance.

4.8. Transaction Costs

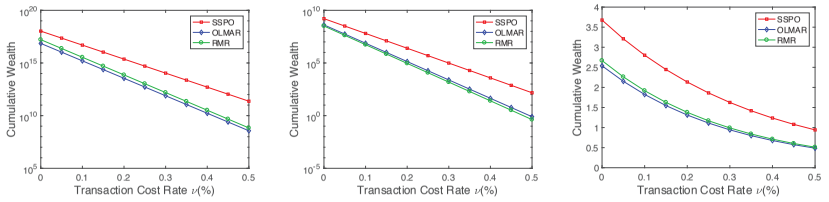
In practice, transaction cost is an important issue in PO. Suppose we have to pay at a transaction cost rate $\nu \in (0,1)$ to update the portfolio. Then according to the proportional transaction cost model (Blum and Kalai, 1999; Li et al., 2015; Huang et al., 2016), the cumulative wealth at the beginning of the t-th day is

$$S_n^\nu = S_0 \prod_{t=1}^{n} [(\hat{\mathbf{b}}_t^\top \mathbf{x}_t) \cdot (1 - \frac{\nu}{2} \sum_{i=1}^{d} |\hat{\mathbf{b}}_t^{(i)} - \tilde{\mathbf{b}}_{t-1}^{(i)}|)], \tag{40}$$

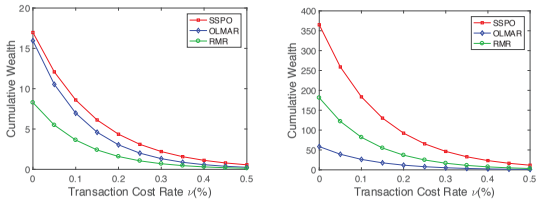
$$\tilde{\mathbf{b}}_{t-1}^{(i)} = \frac{\hat{\mathbf{b}}_{t-1}^{(i)} \cdot \mathbf{x}_{t-1}^{(i)}}{\hat{\mathbf{b}}_{t-1}^\top \mathbf{x}_{t-1}}, \tag{41}$$

where $\tilde{\mathbf{b}}_{t-1}^{(i)}$ denotes the adjusted portfolio of Asset i at the end of the $(t-1)$-th day and $\tilde{\mathbf{b}}_0$ is set as $[0, \cdots, 0]^\top$. $\frac{\nu}{2} \sum_{i=1}^{d} |\hat{\mathbf{b}}_t^{(i)} - \tilde{\mathbf{b}}_{t-1}^{(i)}|$ is the proportional transaction cost when we change the adjusted portfolio $\tilde{\mathbf{b}}_{t-1}$ to the next portfolio $\hat{\mathbf{b}}_t$.

To test the effectiveness of SSPO with consideration of transaction cost, we conduct experiments of cumulative wealth with $\nu = 0 \sim 0.5\%$, where $\nu = 0.5\%$ is a rather high cost rate for stock transactions. The results shown in Figure 9 indicate that SSPO outperforms the two state-of-the-art systems OLMAR and RMR on all the data sets and thus it is applicable to real-world financial environments.



(a) NYSE(O) (b) NYSE(N) (c) DJIA



(d) SP500 (e) TSE

Figure 9: Cumulative wealths of portfolio optimization systems with respect to the transaction cost rate ν on 5 benchmark data sets.

4.9. Running Times

We use a computer with an AMD A10-7800 CPU and an 8GB DDR3 1600MHz memory card to run SSPO in the experiments, which shows that it is sufficiently fast for large-scale and time-limited trading environments such as High-Frequency Trading (HFT) (Aldridge, 2013). The average running times (in seconds) of SSPO for one trade on different data sets are: NYSE(O) (0.0576s), NYSE(N) (0.0450s), DJIA (0.0455s), SP500 (0.0449s), and TSE (0.1190s). Hence SSPO has good computational efficiency besides significant investing advantage.

5. Conclusions

We present a novel short-term sparse portfolio optimization (SSPO) system to concentrate wealth on a few assets with good increasing potential according to some empirical financial principles. Few short-term PO systems construct sparse portfolios, and most existing sparsity systems are either lazy updates or for the long-term PO. These problems motivate the design of SSPO. We further propose an ADMM algorithm for SSPO, and prove that its augmented Lagrangian has a saddle point. This is the foundation of the iterative formulae of ADMM but is seldom addressed before.

We conduct extensive experiments on 5 benchmark data sets with diverse real-world stock data, which shows that SSPO outperforms other state-of-the-art short-term PO systems with all the investing performance measurements (cumulative wealth, mean excess return and α Factor) on all the benchmark data sets. The order statistics of the SSPO returns also indicate that SSPO does not simply select the lucky assets with the highest growth and its returns are credible. SSPO is also competitive to other systems on the risk metrics SR and IR, and shows robustness in balancing between return and risk. Furthermore, SSPO can withstand reasonable transaction costs and runs fast, thus it is applicable to real-world financial environments including High-Frequency Trading. Therefore, SSPO is an effective and robust system that is worth further investigations. In the future, we will continue to establish more complex SSPO systems, so as to improve the investing performance and the robustness to risk.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grants 61703182, 61603152, 61602211, in part by the Talent Introduction Foundation of Jinan University under Grants 88016653, 88016534, in part by the Fundamental Research Funds for the Central Universities under Grants 21617347, 21617404, in part by the Fundamental Research Funds for the Center for Mathematical Finance in Guangdong Province under Grant 50411628, in part by the Science and Technology Program of Guangzhou, China under Grant 201707010259, and in part by the Guangxi Key Laboratory of Trusted Software (No. kx201606).

References

A. Agarwal, E. Hazan, S. Kale, and R. E. Schapire. Algorithms for portfolio management based on the Newton method. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2006.

I. Aldridge. *High-Frequency Trading: A Practical Guide to Algorithmic Strategies and Trading Systems*. Wiley, Hoboken, NJ, 2 edition, Apr. 2013.

D. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1999.

A. Blum and A. Kalai. Universal portfolios with and without transaction costs. *Machine Learning*, 35(3):193–205, 1999.

W. F. M. D. Bondt and R. Thaler. Does the stock market overreact? *Journal of Finance*, 40(3):793–805, Jul. 1985.

A. Borodin, R. El-Yaniv, and V. Gogan. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 21(1):579–594, Jan. 2004.

S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2010.

J. Brodie, I. Daubechies, C. D. Giannone, and I. Loris. Sparse and stable Markowitz portfolios. *Proceedings of the National Academy of Sciences of the United States of America*, 106(30):12267–12272, Jul. 2009.

E. J. Candès and Y. Plan. Near-ideal model selection by ℓ_1 minimization. *Annals of Statistics*, 37(5A):2145–2177, 2009.

T. M. Cover. Universal portfolios. *Mathematical Finance*, 1(1):1–29, Jan. 1991.

P. Das, N. Johnson, and A. Banerjee. Online lazy updates for portfolio selection with transaction costs. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, pages 202–208, 2013.

J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra. Efficient projections onto the ℓ^1-ball for learning in high dimensions. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2008.

E. F. Fama. The behavior of stock-market prices. *Journal of Business*, 38(1):34–105, Jan. 1965.

E. F. Fama. Efficient capital markets: A review of theory and empirical work. *Journal of Finance*, 25(2):383–417, May 1970.

X. Fang, Y. Xu, X. Li, Z. Lai, and W. K. Wong. Robust semi-supervised subspace clustering via non-negative low-rank representation. *IEEE Transactions on Cybernetics*, 46(8):1828–1838, Aug. 2016.

R. Fernholz. *Stochastic Portfolio Theory*. Springer Verlag, 2002.

R. Grinold and R. Kahn. *Active Portfolio Management: A Quantitative Approach for Producing Superior Returns and Controlling Risk*. McGraw-Hill, New York, 1999.

L. Györfi, G. Lugosi, and F. Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, Apr. 2006.

J. He, Q. G. Wang, P. Cheng, J. Chen, and Y. Sun. Multi-period mean-variance portfolio optimization with high-order coupled asset dynamics. *IEEE Transactions on Automatic Control*, 60(5):1320–1335, May 2015.

M. Ho, Z. Sun, and J. Xin. Weighted elastic net penalized mean-variance portfolio design and computation. *SIAM Journal on Financial Mathematics*, 6(1), 2015.

D. Huang, J. Zhou, B. Li, S. C. H. Hoi, and S. Zhou. Robust median reversion strategy for on-line portfolio selection. In *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2006–2012, 2013.

D. Huang, J. Zhou, B. Li, S. C. H. Hoi, and S. Zhou. Robust median reversion strategy for online portfolio selection. *IEEE Transactions on Knowledge and Data Engineering*, 28 (9):2480–2493, Sep. 2016.

N. Jegadeesh. Evidence of predictable behavior of security returns. *Journal of Finance*, 45 (3):881–898, Jul. 1990.

N. Jegadeesh. Seasonality in stock price mean reversion: Evidence from the U.S. and the U.K. *Journal of Finance*, 46(4):1427–1444, Sep. 1991.

N. Jegadeesh and S. Titman. Returns to buying winners and selling losers: Implications for stock market efficiency. *Journal of Finance*, 48(1):65–91, Mar. 1993.

D. Kahneman and A. Tversky. Prospect theory: An analysis of decision under risk. *Econometrica*, 47(2):263–292, Mar. 1979.

Z. R. Lai, D. Q. Dai, C. X. Ren, and K. K. Huang. Discriminative and compact coding for robust face recognition. *IEEE Transactions on Cybernetics*, 45(9):1900–1912, Sep. 2015.

B. Li and S. C. H. Hoi. On-line portfolio selection with moving average reversion. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

B. Li and S. C. H. Hoi. Online portfolio selection: A survey. *ACM Computing Surveys (CSUR)*, 46(3):35:1–35:36, 2014.

B. Li, S. C. H. Hoi, and V. Gopalkrishnan. CORN: Correlation-driven nonparametric learning approach for portfolio selection. *ACM Transactions on Intelligent Systems and Technology*, 2(3), Apr. 2011. Article No.21.

B. Li, P. Zhao, S. C. H. Hoi, and V. Gopalkrishnan. PAMR: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.

B. Li, S. C. H. Hoi, P. Zhao, and V. Gopalkrishnan. Confidence weighted mean reversion strategy for online portfolio selection. *ACM Transactions on Knowledge Discovery from Data*, 7(1), Mar. 2013. Article 4.

B. Li, S. C. H. Hoi, D. Sahoo, and Z. Y. Liu. Moving average reversion strategy for on-line portfolio selection. *Artificial Intelligence*, 222:104–123, 2015.

B. Li, D. Sahoo, and S. C. H. Hoi. OLPS: a toolbox for on-line portfolio selection. *Journal of Machine Learning Research*, 17(1):1242–1246, 2016.

J. Lintner. The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets. *Review of Economics and Statistics*, 47(1):13–37, Feb. 1965.

B. Mahdavi-Damghani, K. Mustafayeva, S. Roberts, and C. Buescu. Portfolio optimization in the context of cointelated pairs: Stochastic differential equation vs. machine learning approach. *Social Science Electronic Publishing*, 2017.

S. Maillard, T. Roncalli, and J. Teiletche. On the properties of equally-weighted risk contributions portfolios. *Social Science Electronic Publishing*, 36(4):60–70, 2010.

H. M. Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, Mar. 1952.

Y. Q. Mohsin, G. Ongie, and M. Jacob. Iterative shrinkage algorithm for patch-smoothness regularized medical image recovery. *IEEE Transactions on Medical Imaging*, 34(12): 2417–2428, Dec. 2015.

W. F. Sharpe. Capital asset prices: A theory of market equilibrium under conditions of risk. *Journal of Finance*, 19(3):425–442, Sep. 1964.

W. F. Sharpe. Mutual fund performance. *Journal of Business*, 39(1):119–138, Jan. 1966.

W. Shen, J. Wang, and S. Ma. Doubly regularized portfolio with risk minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2014.

R. J. Shiller. *Irrational Exuberance*. Princeton University Press, Princeton, NJ, 2000.

R. J. Shiller. From efficient markets theory to behavioral finance. *Journal of Economic Perspectives*, 17(1):83–104, 2003.

S. Still and I. Kondor. Regularizing portfolio optimization. *New Journal of Physics*, 12: 1–14, 2010.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society*, 58(1):267–288, 1996.

J. L. Treynor and F. Black. How to use security analysis to improve portfolio selection. *Journal of Business*, 46(1):66–86, Jan. 1973.

Y. Vardi and C. H. Zhang. The multivariate ℓ^1-median and associated data depth. *Proceedings of the National Academy of Sciences of the United States of America*, 97(4): 1423–1426, Feb. 2000.

L. Yang, R. Couillet, and M. R. McKay. A robust statistics approach to minimum variance portfolio optimization. *IEEE Transactions on Signal Processing*, 63(24):6684–6697, Dec. 2015.

P. Yin, Y. Lou, Q. He, and J. Xin. Minimization of ℓ_{1-2} for compressed sensing. *SIAM Journal on Scientific Computing*, 37(1):A536–A563, 2015.

Z. Zhan, J. F. Cai, D. Guo, Y. Liu, Z. Chen, and X. Qu. Fast multiclass dictionaries learning with geometrical directions in MRI reconstruction. *IEEE Transactions on Biomedical Engineering*, 63(9):1850–1861, Sep. 2016.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 9/17; Revised 8/18; Published 10/18

Scaling up Data Augmentation MCMC via Calibration

Leo L. Duan LI.DUAN@UFL.EDU
Department of Statistics
University of Florida
Gainesville, FL

James E. Johndrow JOHNDROW@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA

David B. Dunson DUNSON@DUKE.EDU
Department of Statistical Science
Duke University
Durham, NC

Editor: Ryan Adams

Abstract

There has been considerable interest in making Bayesian inference more scalable. In big data settings, most of the focus has been on reducing the computing time per iteration rather than reducing the number of iterations needed in Markov chain Monte Carlo (MCMC). This article considers data augmentation MCMC (DA-MCMC), a widely used technique. DA-MCMC samples tend to become highly autocorrelated in large samples, due to a mis-calibration problem in which conditional posterior distributions given augmented data are too concentrated. This makes it necessary to collect very long MCMC paths to obtain acceptably low MC error. To combat this inefficiency, we propose a family of calibrated data augmentation algorithms, which appropriately adjust the variance of conditional posterior distributions. A Metropolis-Hastings step is used to eliminate bias in the stationary distribution of the resulting sampler. Compared to existing alternatives, this approach can dramatically reduce MC error by reducing autocorrelation and increasing the effective number of DA-MCMC samples per unit of computing time. The approach is simple and applicable to a broad variety of existing data augmentation algorithms. We focus on three popular generalized linear models: probit, logistic and Poisson log-linear. Dramatic gains in computational efficiency are shown in applications.

Keywords: Bayesian Probit, Biased subsampling, Big n, Data augmentation, Log-linear model, Logistic regression, Maximal correlation, Polya-Gamma

1. Introduction

With the deluge of data in many modern application areas, there is pressing need for scalable computational algorithms for inference from such data, including uncertainty quantification (UQ). Somewhat surprisingly, even as the volume of data increases, uncertainty often remains sizable. Examples in which this phenomenon occurs include financial fraud detection (Ngai et al., 2011), disease mapping (Wakefield, 2007) and online click-through tracking (Wang et al., 2010). Bayesian approaches provide a useful paradigm for quantifying uncertainty in these and other settings.

The standard approach to Bayesian posterior computation is Markov chain Monte Carlo (MCMC) and related sampling algorithms. However, conventional MCMC algorithms often scale poorly in problem size and complexity. Due to its sequential nature, the computational cost of MCMC is the product of two factors: the evaluation cost at each sampling iteration and the total number of iterations needed to obtain an acceptably low Monte Carlo (MC) error. While a substantial literature has developed focusing on decreasing computational cost per iteration in "big data" (large sample) settings (Minsker et al. (2017); Maclaurin and Adams (2015); Srivastava et al. (2015); Conrad et al. (2016) among others), there has been less focus on reducing the required number of MCMC iterations. This contrasts with a historical focus in the statistics and probability literature on improving mixing and convergence of MCMC in more traditional small to moderate sample size problems, and suggests the opportunity for improved performance in big data settings through a renewed focus on improving mixing.

A major concern in applying MCMC algorithms in big data problems is that the level of autocorrelation in the MCMC path may increase with the size of the data. Markov chains with high autocorrelation have low *effective sample size (ESS)* per unit computational time, which we refer to informally as the *slow mixing* problem. The ESS compares the asymptotic variance of the MCMC time averaging estimate to a gold standard Monte Carlo algorithm that collects independent samples. For example, if the number of effective samples in 1,000 MCMC iterations is only 10, then the MCMC algorithm will need to be run 100 times as long as an ordinary MC algorithm to obtain the same MC error for time averages. Such a scenario is not unusual in big data problems, leading MCMC algorithms to face a *double burden*, with the time per iteration increasing and it becoming necessary to collect more iterations as sample size increases.

This double burden has led many members of the machine learning community to abandon MCMC in favor of more easily scalable alternatives, such as variational approximations. Unfortunately, these approaches typically lack theoretical guarantees and often badly underestimate posterior uncertainty. Hence, there has been substantial interest in recent years in designing scalable MCMC algorithms. The focus of this paper is a popular and broad class of Data Augmentation (DA)-MCMC algorithms. DA-MCMC algorithms are used routinely in many classes of models, with the algorithms of Albert and Chib (1993) for probit models and Polson et al. (2013) for logistic models being particularly popular. Our focus is on improving the performance of such algorithms in big data settings in which poor scalability occurs both because of high cost per iteration and deterioration of mixing as sample size increases. We focus here on the slow mixing problem.

Johndrow et al. (2018) demonstrate that popular DA-MCMC algorithms have small effective sample sizes in large data settings involving imbalanced data. For example, data may be binary with a high proportion of zeros. A key insight is that this problem results from a discrepancy in the rates at which Gibbs step sizes and the width of the high-probability region of the posterior converge to zero as n increases. In particular, the conditional posterior given the augmented data may simply be too concentrated relative to the marginal posterior, with this problem amplified as the data sample size increases. There is a rich literature on methods for accelerating mixing in DA-MCMC algorithms using tricks ranging from reparameterization to parameter-expansion (Liu and Wu, 1999; Meng and Van Dyk, 1999; Papaspiliopoulos et al., 2007). However, we find that such approaches fail to address the miscalibration problem and have no impact on the worsening mixing rate with increasing data sample size n.

This article proposes a general new class of algorithms that addresses the miscalibration of step sizes in DA. The idea underlying these *calibrated* DA (CDA) algorithms is to introduce auxiliary parameters that change the variance of full conditional distributions for one or more parameters. These auxiliary parameters can adapt with the data sample size n to correct the typical step sizes of the CDA algorithm to match the rate at which the high probability region of the posterior contracts as n increases. In general, the invariant measure of CDA-MCMC – which typically does exist and is unique – differs from the posterior of interest. Thus, CDA-MCMC is a computationally more efficient perturbation of the original Markov chain, and the bias can be eliminated using Metropolis-Hastings. Compared to other adaptive Metropolis-Hastings algorithms, which often require carefully chosen multivariate proposals and complicated adaptation with multiple chains (Tran et al., 2016), CDA-MCMC only requires a simple modification to Gibbs sampling steps to generate proposals. We show the auxiliary parameters can be efficiently adapted for each type of data augmentation via minimizing the difference between Fisher information of conditional and marginal distributions.

2. Calibrated Data Augmentation

Data augmentation Gibbs samplers alternate between sampling latent data z from their conditional posterior distribution given model parameters θ and observed data y, and sampling parameters θ given z and y; either of these steps can be further broken down into a series of full conditional sampling steps but we focus for simplicity on algorithms of the form:

$$\begin{aligned} z \mid \theta, y &\sim \pi(z; \theta, y) \\ \theta \mid z, y &\sim f(\theta; z, y), \end{aligned} \tag{1}$$

where f belongs to a location-scale family, such as the Gaussian. Popular data augmentation algorithms are designed so that both of these sampling steps can be conducted easily and efficiently; e.g., sampling the latent data for each subject independently and then drawing θ simultaneously (or at least in blocks) from a multivariate Gaussian or other standard distribution. This effectively avoids the need for tuning, which is a major issue for Metropolis-Hastings algorithms, particularly when θ is high-dimensional. Data augmentation algorithms are particularly common for generalized linear models (GLMs), with

$\mathbb{E}(y_i \mid x_i, \theta) = g^{-1}(x_i\theta)$ and a conditionally Gaussian prior distribution chosen for θ. We focus in particular on Poisson log-linear, binomial logistic, and binomial probit as motivating examples.

Consider a Markov kernel $K((\theta, z); \cdot)$ with invariant measure Π and update rule of the form (1), and a Markov chain (θ_t, z_t) on a state space $\Theta \times \mathcal{Z}$ evolving according to K. We will abuse notation in writing $\Pi(d\theta) = \int_{z\in\mathcal{Z}} \Pi(d\theta, dz)$. The lag-1 autocorrelation for a function $h : \Theta \to \mathbb{R}$ at stationarity can be expressed as the Bayesian fraction of missing information (Papaspiliopoulos et al. (2007), Rubin (2004), Liu (1994b))

$$\gamma_g = 1 - \frac{\mathbb{E}[\text{var}(h(\theta) \mid z)]}{\text{var}(h(\theta))}, \tag{2}$$

where the integrals in the numerator are with respect to $\Pi(d\theta, dz)$ and in the denominator with respect to $\Pi(d\theta)$. Let

$$L_2(\Pi) = \left\{ h : \Theta \to \mathbb{R}, \int_{\theta\in\Theta} \{h(\theta)\}^2 \Pi(d\theta) < \infty \right\}$$

be the set of real-valued, Π square-integrable functions. The *maximal autocorrelation*

$$\gamma = \sup_{h\in L_2(\Pi)} \gamma_h = 1 - \inf_{h\in L_2(\Pi)} \frac{\mathbb{E}[\text{var}(h(\theta) \mid z)]}{\text{var}(h(\theta))}$$

is equal to the geometric convergence rate of the data augmentation Gibbs sampler (Liu (1994b)). For $h(\theta) = \theta_j$ a coordinate projection, the numerator of the last term of (2) is, informally, the average squared step size for the augmentation algorithm at stationarity in direction j, while the denominator is the squared width of the bulk of the posterior in direction j. Consequently, γ will be close to 1 whenever the average step size at stationarity is small relative to the width of the bulk of the posterior.

The purpose of CDA is to introduce additional parameters that allow us to control the step size relative to the posterior width – roughly speaking, the ratio in (2) – with greater flexibility than reparametrization or parameter expansion. The flexibility gains are achieved by allowing the invariant measure to change as a result of the introduced parameters. The additional parameters, which we denote (r, b), correspond to a collection of reparametrizations, each of which defines a proper (but distinct) likelihood $L_{r,b}(\theta; y)$, and for which there exists a Gibbs update rule of the form (1). In general, r is a vector of scale parameters that are tuned to increase $\mathbb{E}[\text{var}(h(\theta) \mid z)]\{\text{var}(h(\theta))\}^{-1}$ – usually for coordinate projections $h(\theta) = \theta_j$ – although the exact way in which they enter the likelihood and corresponding Gibbs update depend on the application; b are location parameters that shift the high posterior region of $L_{r,b}(\theta; y)$ to better approximate $L(\theta; y)$. The reparametrization also has the property that $L_{1,0}(\theta; y) = L(\theta; y)$, the original likelihood. The resulting Gibbs sampler, which we refer to as CDA Gibbs, has θ-marginal invariant measure $\Pi_{r,b}(\theta; y) \propto L_{r,b}(\theta; y)\Pi^0(\theta)$, where $\Pi^0(\theta)$ is the prior. Ultimately, we are interested in $\Pi_{1,0}(\theta; y)$, so we use CDA Gibbs as an efficient proposal for Metropolis-Hastings. That is, we propose θ^* from $q(\theta^*; \theta)$ with

$$q(\theta^*; \theta) = \int_{z\in\mathcal{Z}} \pi_{r,b}(z; \theta, y) f_{r,b}(\theta^*; z, y) dz, \tag{3}$$

where $\pi_{r,b}$ and $f_{r,b}$ denote the conditional densities of z and θ in the Gibbs sampler with invariant measure $\Pi_{r,b}$. By tuning (r, b) during an adaptation phase to reduce the autocorrelations and increase the Metropolis-Hastings acceptance rate, we can obtain a computationally efficient algorithm. Tuning is facilitated by the fact that the MH acceptance ratios using this proposal kernel have a convenient form, which is a nice feature of using Gibbs to generate MH proposals.

Remark 1 *The CDA MH acceptance ratio is given by*

$$\begin{aligned}\alpha(\theta, \theta^*) &= \min\left\{1, \frac{L(\theta^*; y)\Pi^0(\theta^*)q(\theta; \theta^*)}{L(\theta; y)\Pi^0(\theta)q(\theta^*; \theta)}\right\} \\ &= \min\left\{1, \frac{L(\theta^*; y)L_{r,b}(\theta; y)}{L(\theta; y)L_{r,b}(\theta^*; y)}\right\}. \end{aligned} \tag{4}$$

A general strategy for tuning is given in Section 4.

We give a basic convergence guarantee that holds for CDA MH under weak assumptions on $L_{r,b}$, which is based on Roberts and Smith (1994). Basically, one needs $\Pi(\cdot) \ll \Pi_{r,b}(\cdot)$ for all r, b, where for two probability measures μ, ν, $\mu(\cdot) \ll \nu(\cdot)$ means μ is absolutely continuous with respect to ν.

Remark 2 (Ergodicity) *Assume that $\Pi(d\theta)$ and $\Pi_{r,b}(d\theta)$ have densities with respect to Lebesgue measure on $\mathbb{R}^p$, and that $K_{r,b}((\theta, z); (\theta', z')) > 0 \,\forall\, ((\theta, z), (\theta', z')) \in (\Theta \times \mathcal{Z}) \times (\Theta \times \mathcal{Z})$. Then,*

- *For fixed r, b, CDA Gibbs is ergodic with invariant measure $\Pi_{r,b}(d\theta, dz)$.*
- *A Metropolis-Hastings algorithm with proposal kernel $q_{r,b}(\theta'; \theta)$ as defined in (3) with fixed r, b is ergodic with invariant measure $\Pi(d\theta)$.*

Proofs are located in the Appendix.

2.1. Initial Example: Probit with Intercept Only

To illustrate the CDA algorithm, we first present a toy example of the probit regression with intercept only.

$$y_i \sim \text{Bernoulli}(p_i), \quad p_i = \Phi(\theta) \quad i = 1, \ldots, n$$

and improper prior $\Pi^0(\theta) \propto 1$. The data augmentation algorithm (Tanner and Wong, 1987; Albert and Chib, 1993) is based on the following integral

$$L(y_i; \theta) = \begin{cases} \int_0^\infty f(z_i; \theta, 1)dz_i & \text{if } y_i = 1 \\ \int_{-\infty}^0 f(z_i; \theta, 1)dz_i & \text{if } y_i = 0 \end{cases} \qquad i = 1, \ldots, n$$

where $f(z; \mu, \sigma^2)$ is the density for normal distribution $\text{No}(\mu, \sigma^2)$.

This leads to the update rule

$$z_i \mid \theta, y_i \sim \begin{cases} \text{No}_{[0,\infty)}(\theta, 1) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(\theta, 1) & \text{if } y_i = 0 \end{cases} \qquad i = 1, \ldots, n$$

$$\theta \mid z, y \sim \text{No}\left(n^{-1}\sum_i z_i, n^{-1}\right),$$

where the subscript in $\text{No}_{[a,b]}(\mu, \sigma^2)$ denotes the truncation to the interval $[a, b]$. Johndrow et al. (2018) show that when $\sum_i y_i = 1$, $\text{var}(\theta_t \mid \theta_{t-1})$ is approximately $n^{-1}\log n$, while the width of the high probability region of the posterior is order $(\log n)^{-1}$, leading to slow mixing.

We introduce a scale parameter r_i in the update for z_i, and adjust the conditional mean by a location parameter b_i. This is equivalent to changing the scale of $z_i \mid \theta, y_i$ from 1 to r_i and the mean from θ to $\theta + b_i$. These adjustments yield

$$\begin{aligned}\Pr(y_i = 1|\theta, r_i, b_i) &= \int_0^\infty \frac{1}{\sqrt{2\pi r_i}} \exp\left(-\frac{(z_i - \theta - b_i)^2}{2r_i^2}\right) dz_i \\ &= \Phi\left(\frac{\theta + b_i}{\sqrt{r_i}}\right). \end{aligned} \tag{5}$$

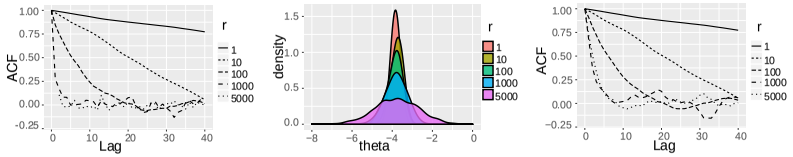
In this simple example, we set the tuning parameters to be all the same: $r_i = r_0$ and $b_i = b_0$ over $i = 1, \ldots, n$, with r_0 and b_0 two scalars. This leads to the modified data augmentation algorithm

$$\begin{aligned} z_i \mid \theta, y_i &\sim \begin{cases} \text{No}_{[0,\infty)}(\theta + b_0, r_0) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(\theta + b_0, r_0) & \text{if } y_i = 0 \end{cases} \qquad i = 1, \ldots, n \\ \theta \mid z, y &\sim \text{No}\left(n^{-1}\sum_i (z_i - b_0), n^{-1}r_0\right). \end{aligned} \tag{6}$$

To achieve step sizes consistent with the width of the high posterior probability region, we need $n^{-1}r_0 \approx (\log n)^{-1}$, so $r_0 \approx n/\log n$. To preserve the original target, we use (6) to generate an MH proposal θ^*. By Remark 1, the MH acceptance probability is given by (4) with $L_{r,b}(\theta; y_i) = \Phi\big((\theta + b_0){r_0}^{-1/2}\big)^{y_i}\Phi\big(-(\theta + b_0){r_0}^{-1/2}\big)^{(1-y_i)}$ and $L(\theta; y_i) = L_{1,0}(\theta; y_i)$. Setting $r_0 = 1$ and $b_0 = 0$ leads to acceptance rate of 1, which corresponds to the original Gibbs sampler.

To illustrate, we consider $\sum_i y_i = 1$ and $n = 10^4$. Letting $r_0 = n/\log n$, we then choose b_0 to increase the acceptance rate in the MH step. In this simple example, it is easy to compute a "good" value of b_0, since $b_0 = -3.7(\sqrt{r} - 1)$ results in $\Pr(y_i = 1) = \Phi(-3.7) = n^{-1}\sum_i y_i \approx 10^{-4}$ in the proposal distribution, centering the proposals near the MLE for p_i.

We perform computation for these data with different values of r_0 ranging from $r_0 = 1$ to $r_0 = 5,000$, with $r_0 = 1,000 \approx n/\log n$ corresponding to the theoretically optimal value. Figure 1(a) plots autocorrelation functions (ACFs) for these different samplers without MH adjustment. Autocorrelation is very high even at lag 40 for $r_0 = 1$, while increasing r_0 leads to dramatic improvements in mixing. There are no further gains in increasing r_0 from the theoretically optimal value of $r_0 = 1,000$ to $r_0 = 5,000$. Figure 1(b) shows kernel-smoothed density estimates of the posterior of θ without MH adjustment for different values of r_0 and based on long chains to minimize the impact of Monte Carlo error; the posteriors are all centered on the same values but with variance increasing somewhat with r_0. With MH adjustment such differences are removed; the MH step has acceptance probability close to one for $r_0 = 10$ and $r_0 = 100$, about 0.6 for $r_0 = 1,000$, and 0.2 for $r_0 = 5,000$.



(a) ACF for CDA without MH adjustment. (b) Posterior density estimates without MH adjustment. (c) ACF for CDA with MH adjustment

Figure 1: Autocorrelation functions (ACFs) and kernel-smoothed density estimates for different CDA samplers in intercept-only probit model.

We also study a common hierarchical Gaussian example in appendix C.

3. Specific Algorithms

In this section, we describe CDA algorithms for general probit and logistic regression.

3.1. Probit Regression

Consider the probit regression:

$$y_i \sim \text{Bernoulli}(p_i), \quad p_i = \Phi(x_i\theta) \quad i = 1, \ldots, n$$

with improper prior $\Pi^0(\theta) \propto 1$. The data augmentation sampler (Tanner and Wong, 1987; Albert and Chib, 1993) has the update rule

$$\begin{aligned} z_i \mid \theta, x_i, y_i &\sim \begin{cases} \text{No}_{[0,\infty)}(x_i\theta, 1) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(x_i\theta, 1) & \text{if } y_i = 0 \end{cases} \quad i = 1, \ldots, n \\ \theta \mid z, x, y &\sim \text{No}((X'X)^{-1}X'z, (X'X)^{-1}). \end{aligned}$$

Liu and Wu (1999) and Meng and Van Dyk (1999), among others, previously studied this algorithm and proposed to rescale θ through parameter expansion. However, this modification does not impact the conditional variance of θ and thus does not directly increase typical step sizes.

Our approach is fundamentally different, since we directly adjust the conditional variance. Similar to the intercept only model, we modify $\text{var}(\theta|z)$ by changing the scale of each z_i. This yields the update rule

$$\begin{aligned} z_i \mid \theta, x_i, y_i &\sim \begin{cases} \text{No}_{[0,\infty)}(x_i\theta + b_i, r_i) & \text{if } y_i = 1 \\ \text{No}_{(-\infty,0]}(x_i\theta + b_i, r_i) & \text{if } y_i = 0 \end{cases} \quad i = 1, \ldots, n \\ \theta \mid z, X &\sim \text{No}((X'R^{-1}X)^{-1}X'R^{-1}(z-b), (X'R^{-1}X)^{-1}), \end{aligned} \tag{7}$$

where $R = \text{diag}(r_1, \ldots, r_n)$, $b = (b_1, \ldots, b_n)'$. Under the Bernoulli likelihood, we have

$$\Pr(y_i = 1|\theta, x_i, r_i, b_i) = \int_0^\infty \frac{1}{\sqrt{2\pi r_i}} \exp\left(-\frac{(z_i - x_i\theta - b_i)^2}{2r_i}\right) dz_i$$

$$= \Phi\left(\frac{x_i\theta + b_i}{\sqrt{r_i}}\right). \tag{8}$$

For fixed $r = (r_1, \ldots, r_n)$ and $b = (b_1, \ldots, b_n)$, (8) defines a proper Bernoulli likelihood for y_i conditional on parameters, and therefore the transition kernel $Q_{r,b}((\theta, z); \cdot)$ defined by the Gibbs update rule in (7) would have a unique invariant measure for fixed r, b, which we denote $\Pi_{r,b}(\theta, z \mid y)$.

For insight into the relationship between r and step size, consider the θ-marginal autocovariance in a Gibbs sampler evolving according to $K_{r,b}$

$$\begin{aligned} &\text{cov}_{r,b}(\theta_t \mid \theta_{t-1}, X, z, y) \\ =&(X'R^{-1}X)^{-1} + (X'R^{-1}X)^{-1}X'R^{-1}\text{cov}(z - b|R)R^{-1}X(X'R^{-1}X)^{-1} \\ \geq&(X'R^{-1}X)^{-1}. \end{aligned}$$

In the special case where $r_i = r_0$ for all i, we have

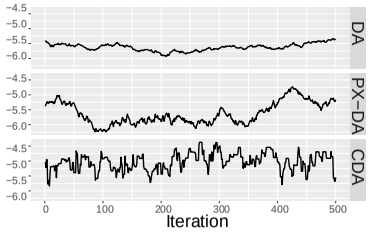
$$\text{cov}_{r,b}(\theta_t \mid \theta_{t-1}, X, z, y) \geq r_0(X'X)^{-1},$$

so that all of the conditional variances are increased by at least a factor of r_0. This holds uniformly over the entire state space, so it follows that

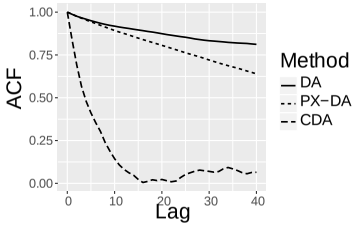
$$\mathbb{E}_{\Pi_{r,b}}[\text{var}(\theta_j \mid z)] \geq r_0\mathbb{E}_{\Pi}[\text{var}(\theta_j \mid z)].$$

The key to CDA is to choose r, b to make $\mathbb{E}_{\Pi_{r,b}}[\text{var}(\theta_j \mid z)]$ close to $\text{var}_{\Pi_{r,b}}(\theta_j \mid z)$, while additionally maximizing the MH acceptance probability. We defer the details of tuning algorithm for r, b to the next section.

For illustration, we consider a simulation study for probit regression with an intercept and two predictors $x_{i,1}, x_{i,2} \sim \text{No}(1, 1)$, with $\theta = (-5, 1, -1)'$, generating $\sum_i y_i \approx 20$ among $n = 10,000$. The Albert and Chib (1993) DA algorithm mixes slowly (Figure 2(a) and 2(b)). We also show the results of the parameter expansion algorithm (PX-DA) proposed by Liu and Wu (1999). PX-DA only mildly reduces the correlation, as it does not solve the small step size problem. After tuning, CDA reaches a satisfactory acceptance rate of 0.6 and leads to dramatically better mixing.



(a) Traceplot for the original DA, parameter expanded DA and CDA algorithms.



(b) ACF for original DA, parameter expanded DA and CDA algorithms.

Figure 2: Panel (a) demonstrates in traceplot and panel (b) in autocorrelation the substantial improvement in CDA by correcting the variance mis-match in probit regression with rare event data, compared with the original (Albert and Chib, 1993) and parameter-expanded methods (Liu and Wu, 1999).

3.2. Logistic Regression

In the second example, we focus on the logistic regression model with

$$y_i \sim \text{Bernoulli}(p_i), \quad p_i = \frac{\exp(x_i\theta)}{1+\exp(x_i\theta)} \quad i = 1, \ldots, n \tag{9}$$

and improper prior $\Pi^0(\theta) \propto 1$. For this model, Polson et al. (2013) proposed Polya-Gamma data augmentation:

$$\begin{aligned} z_i &\sim \text{PG}(1, |x_i\theta|) \quad i = 1, \ldots, n, \\ \theta &\sim \text{No}\left((X'ZX)^{-1}X'(y-0.5), (X'ZX)^{-1}\right), \end{aligned}$$

where $Z = \text{diag}(z_1, \ldots, z_n)$. This algorithm relies on expressing the logistic regression likelihood as

$$L(x_i\theta; y_i) = \int_0^\infty \exp\{x_i\theta(y_i - 1/2)\} \exp\left\{-\frac{z_i(x_i\theta)^2}{2}\right\} \text{PG}(z_i \mid 1, 0) dz_i,$$

where $\text{PG}(a_1, a_2)$ denotes the density of the Polya-Gamma distribution with parameters a_1, a_2, with $\mathbb{E}z_i = a_1/(2a_2)\tanh(a_2/2)$.

Since our goal is to increase the conditional variance $(X'ZX)^{-1}$, we can achieve this stochastically by reducing the mean $\mathbb{E}z_i$. We replace $\text{PG}(z_i \mid 1, 0)$ with $\text{PG}(z_i \mid r_i, 0)$ in the step for updating the latent data. Adding the location term b_i to the linear predictor $\eta_i = x_i\theta$ leads to

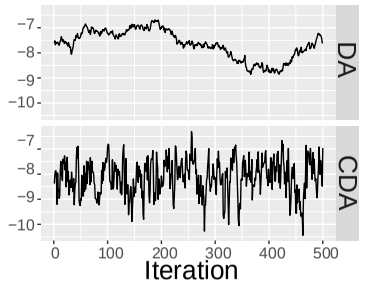
$$\begin{aligned} L_{r,b}(x_i\theta; y_i) &= \int_0^\infty \exp\{(x_i\theta + b_i)(y_i - r_i/2)\} \exp\left\{-\frac{z_i(x_i\theta+b_i)^2}{2}\right\} \text{PG}(z_i \mid r_i, 0) dz_i \\ &= \frac{\exp\{(x_i\theta + b_i)y_i\}}{\{1+\exp(x_i\theta+b_i)\}^{r_i}}, \end{aligned} \tag{10}$$

and the update rule for the CDA Gibbs sampler is then

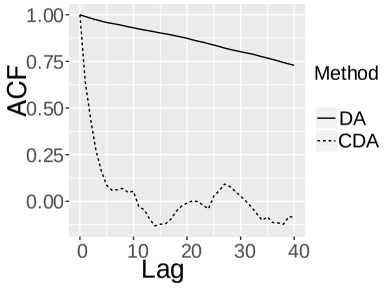
$$\begin{aligned} z_i &\sim \text{PG}(r_i, |x_i\theta + b_i|) \quad i = 1, \ldots, n, \\ \theta' &\sim \text{No}\left((X'ZX)^{-1}X'(y - r/2 - Zb), (X'ZX)^{-1}\right), \end{aligned}$$

where $r = (r_1, \ldots, r_n)'$ and $b = (b_1, \ldots, b_n)'$. We again defer the tuning details for r and b to the next section.

For illustration, we use a two-parameter intercept-slope model with $x_{i1} \overset{iid}{\sim} \text{No}(0, 1)$ and $\theta = (-8, 1)'$. With $n = 10^5$, we obtain rare outcome data with $\sum y_i \approx 50$. In CDA, after tuning, it reaches an acceptance rate of 0.8. Shown in Figure 3, DA mixes slowly, exhibiting strong autocorrelation even at lag 40, while CDA has dramatically better mixing.



(a) Traceplots for DA and CDA.



(b) ACF for DA and CDA.

Figure 3: Panel (a) demonstrates in traceplot and panel (b) in autocorrelation the substantial improvement of CDA in logistic regression with rare event data, compared with the original DA (Polson et al., 2013).

4. Automatic Tuning of Calibration Parameters

As illustrated in the previous subsection, efficiency of CDA is dependent on good choices of the calibration parameters r and b. We propose a simple and efficient algorithm for calculating "good" values of these parameters utilizing the Fisher information and empirical MH acceptance rate. Although our choice of calibration parameters relies on large sample approximations, we find that this calibration approach also works well for modest sample size.

Our goal is to adjust the conditional variance under calibration of (r, b) to approximately match the marginal variance under the exact target distribution, while maintaining a reasonable MH acceptance rate.

To approximate the marginal variance, we use the inverse of the observed Fisher information (Efron and Hinkley, 1978):

$$\begin{aligned}&\mathrm{var}(\theta \mid y) \approx \mathcal{I}^{-1}(\hat{\theta})\\ &\left(\mathcal{I}(\hat{\theta})\right)_{i,j} = \left(\frac{\partial}{\partial \theta_i} \log L(\theta; y)\right)\left(\frac{\partial}{\partial \theta_j} \log L(\theta; y)\right)\Big|_{\theta=\hat{\theta}}\end{aligned}$$

for $i = 1, \ldots, p$, $j = 1, \ldots, p$, where $\hat{\theta}$ is the Maximum a Posteriori (MAP) estimate of θ.

Recall that the CDA proposal has density

$$q(\theta^*; \theta) = \int f_{r,b}(\theta^*; z, y)\pi_{r,b}(z; \theta, y)dz,$$

and the conditional variance has lower bound $\mathrm{var}(\theta^* \mid \theta) \geq \mathbb{E}_{z|\theta}\mathrm{var}(\theta^* \mid z)$. We use the inverse of the observed Fisher information to approximate $\mathrm{var}(\theta^* \mid z)$ via

$$\begin{aligned}&\mathbb{E}_{z|\theta}\mathrm{var}(\theta^* \mid z) \approx \mathbb{E}_{z|\theta}\mathcal{I}^{-1}(\hat{\theta}; r, b, z) \approx \mathcal{I}^{-1}(\hat{\theta}; r, b, \tilde{z}(\hat{\theta}))\\ &\left(\mathcal{I}(\hat{\theta}; r, b, z)\right)_{i,j} = \left(\frac{\partial}{\partial \theta^*_i} \log f_{r,b}(\theta^*; y, z)\right)\left(\frac{\partial}{\partial \theta^*_j} \log f_{r,b}(\theta^*; y, z)\right)\Big|_{\theta^*=\hat{\theta}}.\end{aligned}$$

Since $\mathbb{E}_{z|\theta}\mathcal{I}^{-1}(\hat{\theta}; r, b, z)$ is often intractable or cumbersome to compute, we instead use the second approximation, the Fisher information evaluated at $\tilde{z}(\hat{\theta})$, the conditional mean or mode of $\pi_{r,b}(z; \hat{\theta}, y)$. The choice between mean and mode depends on which has a closed-form expression.

One can now adjust r, b to reduce the distance

$$d_1(r, b) = \mathrm{Dist}\left[\mathcal{I}^{-1}(\hat{\theta}), \mathcal{I}^{-1}(\hat{\theta}; r, b, \tilde{z}(\hat{\theta}))\right], \tag{11}$$

where $\mathrm{Dist}(M_1, M_2)$ is a distance between two matrices, such as $\|M_1 - M_2\|_F$ or $\|M_1^{-1} - M_2^{-1}\|_F$, the Frobenius norm of the difference.

However, the increase in proposal variance only results in an increase in the variance of the Metropolis-Hastings transition density so long as the acceptance probability is not substantially depressed (relative to the DA Gibbs sampler, which has acceptance probability one). Therefore, one also needs to adjust (r, b) both to optimize the acceptance rate $\alpha(\theta, \theta^*)$ and the proposal variance. Considering the average acceptance rate (on the negative-log scale), with expectation over proposal density $q(\theta^*; \theta)$ and posterior $\pi(\theta \mid y)$

$$\begin{aligned}&\mathbb{E}_{\theta|y}\mathbb{E}_{\theta^*|\theta}\big[-\log \alpha(\theta, \theta^*)\big]\\ =&\mathbb{E}_{\theta|y}\mathbb{E}_{\theta^*|z}\mathbb{E}_{z|\theta} \max\left[-\log \frac{L(\theta^*; y)}{L_{r,b}(\theta^*; y)} + \log \frac{L(\theta; y)}{L_{r,b}(\theta; y)}, 0\right].\end{aligned}$$

To provide tractable computation, we again use the functions evaluated at the conditional mean or mode to approximate the three expectations. This yields

$$d_2(r, b) = \max\left[-\log \frac{L(\tilde{\theta}^*(\tilde{z}(\hat{\theta})); y)}{L_{r,b}(\tilde{\theta}^*(\tilde{z}(\hat{\theta})); y)} + \log \frac{L(\hat{\theta}; y)}{L_{r,b}(\hat{\theta}; y)}, 0\right], \tag{12}$$

with $\hat{\theta}$ the MAP of θ, $\tilde{z}(\theta)$ the mean or mode of $\pi_{r,b}(z; \theta, y)$, $\tilde{\theta}^*$ the mean or mode of $f_{r,b}(\theta^*; z, y)$.

Combining (11) and (12), this yields the optimal tuning parameters under those two criteria:

$$(\hat{r}, \hat{b}) = \min_{r,b}\big[d_1(r, b) + \lambda d_2(r, b)\big]. \tag{13}$$

The optional parameter $\lambda > 0$ allows for differential weighting of the acceptance rate and variance, although the default $\lambda = 1$ works well for all of our applications.

To facilitate automatic tuning in generic cases, we exploit the automatic differentiation and optimization software, TensorFlow, to compute the Fisher information and optimize for $(\hat{r}, \hat{b})$. One only needs to provide the densities $L_{r,b}(\theta; y)$, $f_{r,b}(\theta^*; z, y)$, $\pi_{r,b}(z; \theta, y)$ and two conditional estimators $\tilde{z}(\theta)$ and $\tilde{\theta}^*(z)$.

We now provide the tuning details for probit and logistic regression. The likelihood and update densities $L_{r,b}(\theta; y)$, $f_{r,b}(\theta^*; z, y)$, $\pi_{r,b}(z; \theta, y)$ are already given, we present the conditional estimators. For probit regression, the two conditional modes for $\pi_{r,b}(z; \theta, y)$, $\tilde{\theta}^*$ and $f_{r,b}(\theta^*; z, y)$ are available in closed form, viz

$$\begin{aligned}\tilde{z}_i(\theta) =&\begin{cases} x_i\theta + b_i & \text{if } (y_i - 0.5)(x_i\theta + b_i) > 0\\ 0 & \text{otherwise}\end{cases} \quad \text{for } i = 1, \ldots, n,\\ \tilde{\theta}^*(z) =&(X'R^{-1}X)^{-1}X'R^{-1}(z - b).\end{aligned}$$

For logistic regression, the conditional means for $\pi_{r,b}(z; \theta, y)$, $\tilde{\theta}^*$ and $f_{r,b}(\theta^*; z, y)$ all have closed-form expressions given by

$$\begin{aligned}\tilde{z}_i(\theta) =&\frac{r_i}{2|x_i\theta_i + b_i|}\tanh(\frac{|x_i\theta_i + b_i|}{2}) \quad i = 1, \ldots, n,\\ \tilde{\theta}^*(z) =&(X'ZX)^{-1}X'(y - r/2 - Zb).\end{aligned}$$

5. Geometric convergence rates for CDA-MH and CDA-Gibbs

Although Remark 2 gives a basic guarantee of convergence of the usual time-averaging estimators commonly used in MCMC, the goal of CDA-MH is to improve upon the convergence rate of the usual DA Gibbs. Motivation for CDA is provided by the results of Johndrow et al. (2018), which studied the special case of intercept-only logistic and probit regression when $y = 1$ and $n \to \infty$ so that the data grow increasingly imbalanced as the sample size increases. Johndrow et al. (2018) showed that in this setting, the spectral gap of DA converges to zero at least as fast as $n^{-1/2}(\log n)^k$ for $k \leq 5.5$, while random-walk Metropolis has spectral gap order $(\log n)^3$ or larger. This suggests the superiority of Metropolis algorithms in the large sample imbalanced data setting. However, to implement Metropolis effectively with moderate to large numbers of covariates, one needs an efficient way to construct proposals, which is the goal of CDA.

We now give a result on the convergence rates of CDA and CDA-MH for imbalanced intercept-only logistic regression. The result shows that the spectral gap is larger than that for DA (as a function of n), and comparable to MH with optimally tuned proposals when y grows no faster than $\log n$. While this is a special case, we note that the result in Johndrow et al. (2018) is given only for fixed $y = 1$, and thus our result is more general. The difficulty

of obtaining quantitative estimates of the rate at which the spectral gap converges to zero as n grows is underscored by the length and complexity of the arguments in Johndrow et al. (2018).

Consider intercept-only logistic regression from (9) with $x_i = 1$ for $i = 1, \ldots, n$ and prior $\theta \sim \text{No}(0, \sigma^2)$. As all p_i's are the same, we use a single scalar $r_i = r$ and $b_i = b$ for all i. With r, b fixed, the update rule for CDA-Gibbs is

$$\pi_{r,b}(z \mid \theta) = \text{PG}(nr, |\theta + b|)$$
$$f_{r,b}(\theta' \mid z) = \text{No}(m, \Lambda)$$

where $\Lambda = (z + 1/\sigma^2)^{-1}$, $m = \Lambda a - b$ and $a = \sum_i y_i - nr/2 + b/\sigma^2$.

Theorem 3 *Consider intercept-only logistic regression with n observations. Then*

1. *CDA-Gibbs is uniformly ergodic*

2. *CDA-MH is uniformly ergodic*

3. *If $\sum_i y_i = o(\log n)$, there exist choices for r, b such that CDA-MH has spectral gap*

$$\kappa = \mathcal{O}(n^{-\frac{2.5+2\log 2}{\sigma^2}}).$$

Thus, for $\sigma^2 > 5 + 4\log 2 \approx 7.77$, the spectral gap of CDA-MH goes to zero more slowly with increasing n than DA-Gibbs. Moreover, if we choose the prior $\sigma^2 = \log n$, the spectral gap of CDA-MH is independent of n. It follows that CDA-MH mixes rapidly as $n \to \infty$ in the large-sample imbalanced setting, unlike DA-Gibbs, which has spectral gap converging to zero at rate $n^{-1/2}$ or faster (ignoring logarithmic factors).

To show that the result is borne out empirically, we conduct simulations as in Johndrow et al. (2018), with fixed $\sum_i y_i = 1$ and increasing n from 10^1 to a massive 10^{14}. Figure 4 compares the effective sample size per $1,000$ steps using DA and CDA. The deterioration of DA shows up as early as $n = 10^2$; its slow-down becomes critical at $n = 10^4$ with effective sample size close to 0. CDA performs exceptionally well, even at $n = 10^{14}$ (we stop at 10^{14} as $1/n$ reaches the limit of floating point accuracy).

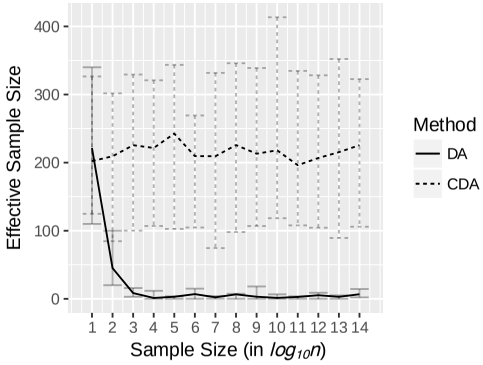


Figure 4: Effective sample size (with 95% pointwise confidence interval) per $1,000$ steps with different sample size n from 10 to 10^{14}, using logistic regression model with intercept.

6. Simulation Study

In this section, we compare the performance of CDA against popular alternative algorithms.

6.1. Comparison with Downsampling Algorithm

As motivated in the introduction, two factors are necessary for MCMC to be practically useful: a low computing cost in each iteration and a high effective sample size within a small number of iterations.

One potential issue for data augmentation in general is the large number of latent variables to sample in each iteration. A common strategy is to avoid sampling latent variables for every observation by approximating the Markov transition kernel using subsamples (Korattikara et al., 2014; Quiroz et al., 2018; Bardenet et al., 2017). Unlike other alternative algorithms we consider here, this changes the invariant measure. Finding a suitable subsample size while controlling the approximation error is challenging and usually problem-specific (Johndrow et al., 2017; Rudolf et al., 2018), and we do not consider it here. Instead, our goal is to show sub-sampling alone does not address the low ESS of DA; whereas one can trivially combine our proposed CDA strategy with subsampling to scale DA-MCMC up to enormous data sample sizes. We illustrate this strategy here.

We consider the same two-parameter intercept-slope model in logistic regression as described above, except we now vary data sample size from $n = 10^5$ to 10^8. We simulate Bernoulli outcomes $y_i \sim \text{Bernoulli}\big((1 + \exp(-x_i\theta))^{-1}\big)$ with $x_i = (1, w_i)$ for $w_i \overset{iid}{\sim} \text{No}(0, 1)$ and $\theta = (-\theta_0, 1)'$. We vary θ_0 to obtain $\sum y_i \approx 10$ for each n. We utilize the minibatch Polya-Gamma algorithm described by Johndrow et al. (2017), and apply CDA to calibrate

the variance discrepancy. Since y is highly imbalanced, we apply biased sampling by including all data with $y_i = 1$, while sub-sampling 1% of data with $y_i = 0$.

Denoting the set of all data with $y_i = 1$ as V_1 and a random subset with $y_i = 0$ as V_0, we adjust the likelihood contribution from $y_i = 0$ via a power of $(n - |V_1|)/|V_0|$ to compensate for the downsampling, leading to an approximate likelihood

$$L(\theta; y) = \prod_{i \in V_1} \frac{\exp(x_i\theta)}{1 + \exp(x_i\theta)} \left(\prod_{i \in V_0} \frac{1}{1 + \exp(x_i\theta)} \right)^{\frac{n-|V_1|}{|V_0|}}.$$

The number of latent variables is reduced to $n_0 \equiv |V_0| + |V_1|$; since n_0 is still large, slow mixing remains a problem and calibration is needed. The algorithmic details are presented in the appendix.

Figure 5 compares the performance of the two approximating algorithms, one combining CDA and sub-sampling, and one using sub-sampling alone. Clearly, sub-sampling alone still results in very small effective sample size, while using CDA and sub-sampling together can produce excellent computational performance.

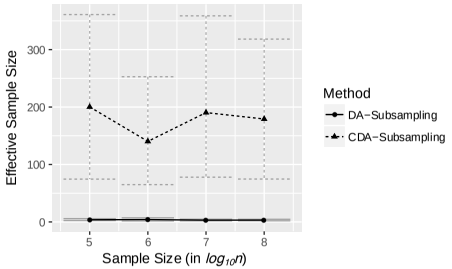


Figure 5: Comparing the performance of CDA and DA, coupled with sub-sampling approximation to reduce the number of sampled latent variables.

6.2. Comparison with Independence Metropolis-Hastings

In the CDA-MH algorithm, we utilize the marginal quantities $\hat{\theta}$ and $\mathcal{I}(\hat{\theta})$ to tune the (r, b) parameters. We compare the performance of the CDA proposal against alternative MH proposal with access to the same information. Specifically, we analyze MH using independent multivariate t proposals with mean $\hat{\theta}$ and variance $\mathcal{I}^{-1}(\hat{\theta})$. We show that this algorithm has very low acceptance rate relative to CDA-MH.

The general form of the MH acceptance rate is given by

$$\alpha(\theta, \theta^*) = \min\left\{1, \frac{L(\theta^*; y)q(\theta; \theta^*)}{q(\theta^*; \theta)L(\theta; y)} \frac{\Pi^0(\theta^*)}{\Pi^0(\theta)}\right\}.$$

Assuming the prior $\Pi^0(\theta)$ has negligible impact when n is large, the key to a high acceptance rate is to have $L(\theta; y)/q(\theta; \theta^*)$ close to a constant in the high posterior density region of the

parameter space. However, for computational convenience, one often has to use a proposal that is easy to sample. The density mis-match between $L(\theta; y)$ and $q(\theta; \theta^*)$ can cause the ratio to decrease rapidly moving away from the posterior mode of θ, resulting in a high rejection rate.

To illustrate, we consider the independent multivariate t-distribution proposal for logistic regression:

$$q(\theta^*, \theta) = q(\theta) = t_\nu\left\{\theta; \hat{\theta}, (\nu - 2)\nu^{-1}\mathcal{I}^{-1}(\hat{\theta})\right\},$$

where $\nu > 2$ and $\mathcal{I}(\hat{\theta}) = X^T \mathrm{diag}[\exp(x_i\hat{\theta})\{1 + \exp(x_i\hat{\theta})\}^{-2}]X$. The second parameter is set to have $\mathrm{var}(\theta) = \mathcal{I}^{-1}(\hat{\theta})$ exactly. We choose $\nu = 3$ to induce a tail heavier than the target likelihood, which is a necessary condition for geometric ergodicity of MH with independent proposals (Mengersen et al., 1996).

The density ratio is

$$\begin{aligned}\frac{L(\theta; y)}{q(\theta)} &= c_1 \left\{\prod_i \frac{\exp(y_i x_i \theta)}{1 + \exp(x_i\theta)}\right\} \left\{1 + \frac{1}{\nu - 2}(\theta - \hat{\theta})^{\mathrm{T}}\mathcal{I}(\hat{\theta})(\theta - \hat{\theta})\right\}^{(\nu+p)/2} \\ &= c_1 \frac{\exp(\sum_i y_i x_i \theta)}{\prod_i\{1 + \exp(x_i\theta)\}} \left[1 + \sum_i \frac{1}{\nu - 2}(x_i\theta - x_i\hat{\theta})^2 \exp(x_i\hat{\theta})\{1 + \exp(x_i\hat{\theta})\}^{-2}\right]^{(\nu+p)/2}, \end{aligned} \tag{14}$$

where c_1 denotes the constant part. We give an approximation of the acceptance ratio.

We focus on the case $\sum y_i \ll n$, where the mixing is slow for DA-Gibbs. This results in $\exp(x_i\hat{\theta}) \approx 0$ for most i. Assuming the high posterior density region is a neighborhood $\{\theta : |x_i\theta - x_i\hat{\theta}| < \eta \quad \text{for all } i\}$, where η is a bounded constant, the second term in (14) is close to a constant, while the first term is approximately equal to its numerator. The acceptance ratio is thus approximately

$$\frac{L(\theta^*; y)q(\theta)}{q(\theta^*)L(\theta; y)} \approx \exp\left\{\sum_i y_i x_i(\theta^* - \theta)\right\},$$

which decreases exponentially away from the current state.

In contrast, since the CDA proposal density is similar to the target, with calibration the density ratio can be made close to a constant in the neighborhood of the mode. Consider the density ratio in the logistic CDA proposal:

$$\frac{L(\theta; y)}{L_{r,b}(\theta; y)} = c_2 \prod_i \frac{\{1 + \exp(x_i\theta + b_i)\}^{r_i}}{1 + \exp(x_i\theta)},$$

where c_2 is a constant. Minimizing the Fisher information distance gives approximately $r_i \approx \exp(x_i\hat{\theta})$ and $b_i \approx -x_i\hat{\theta}$, so the density ratio is approximately c_2. Thus the acceptance ratio

$$\frac{L(\theta^*; y)L_{r,b}(\theta; y)}{L_{r,b}(\theta^*; y)L(\theta; y)} \approx 1.$$

We compare the performance of MH algorithms with t_3 and CDA proposals, using the two-parameter intercept-slope example described in Section 6.1. Figure 6 shows the acceptance ratio at different intercept values θ_0, which is approximately the average of $x_i\theta$.

The acceptance rate drops rapidly to 0 for the t_3 proposal, and is close to one for the CDA proposal.

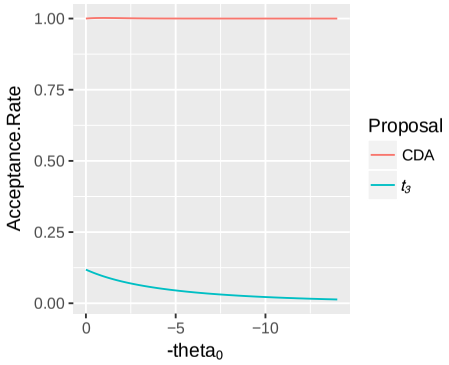


Figure 6: Comparing the acceptance ratios using the multivariate t-distribution and CDA proposals in logistic regression, with variance fixed at the inverse Fisher information. CDA has a much higher acceptance ratio than the multivariate t proposal.

7. Data Applications

7.1. Bernoulli Latent Factor Model with Group Intercepts for Network Modeling

We now apply CDA to accelerate estimation of group intercepts in a latent factor model. The dataset is a large sparse network from the Human Connectome Project (Marcus et al., 2011). The network data under consideration is an adjacency matrix representing the connectivity among $V = 1015$ macroscopic regions of one human brain. The matrix $\{A_{ij}\}_{(i,j)\in\{1\dots V\}^2}$ is binary and symmetric. For $i \neq j$, $A_{ij} = 1$ if regions i and j are connected, $A_{ij} = 0$ otherwise; A_{ii} are missing as self-connections are ignored. Therefore, there are effectively $n = V(V-1)/2 = 514,605$ observed binary outcomes.

There are 507 regions located in the left ($\mathcal{L}$) and 508 in the right hemisphere ($\mathcal{R}$). There are many more connections within each hemisphere ($\sum A_{i\in\mathcal{L},j\in\mathcal{L},i>j} = 2,280$, $\sum A_{i\in\mathcal{R},j\in\mathcal{R},i>j} = 2,443$), than across hemispheres ($\sum A_{i\in\mathcal{L},j\in\mathcal{R}} = 23$). To quantify this phenomenon, we use two intercepts β_0 and β_1 to represent the within- and across-hemisphere fixed effects

within the following Bernoulli probit latent factor model

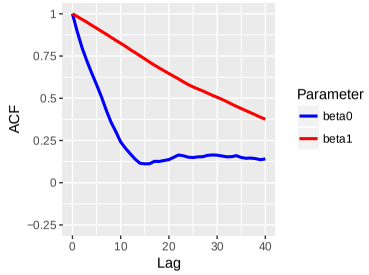
$$
\begin{aligned}
&A_{ij} \sim \text{Bernoulli}(p_{ij}), \qquad p_{ij} = \Phi(\psi_{ij}),\\
&\psi_{ij} = \sum_{r=1}^{d} u_{ir} v_r u_{jr} + \beta_0 w_{ij} + \beta_1 (1 - w_{ij}) \quad \text{for } i = 2\dots V, j < i,\\
&\pi(U) \propto 1, \qquad U^{\mathrm{T}} U = I_d,\\
&v_r \sim \text{No}_{(0,\infty)}(0, \sigma^2) \qquad \text{for } r = 1, \dots, d,\\
&\beta_0 \sim \text{No}(0, 100), \quad \beta_1 \sim \text{No}(0, 100), \quad \sigma^2 \sim \text{Inverse-Gamma}(2, 1),
\end{aligned}
$$

where $w_{ij} = 0$ if $i \in \mathcal{L}$ and $j \in \mathcal{R}$, otherwise $w_{ij} = 1$; $U = \{u_{ir}\}$ is a V-by-d matrix of latent factors. Following Hoff (2009), we assign U a uniform prior on Stiefel manifold $\mathbb{S}^{V\times d} = \{U : U^{\mathrm{T}}U = I_d\}$, and set the latent dimension at $d = 10$. The latent variable updates in the probit data augmentation algorithm are given by

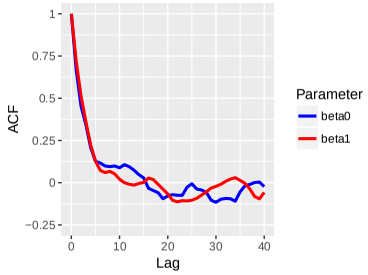
$$
\begin{aligned}
&z_{ij} \sim \begin{cases} \text{No}_{(0,\infty)}(\psi_{ij}, 1) & \text{if } A_{ij} = 1 \\ \text{No}_{(-\infty,0)}(\psi_{ij}, 1) & \text{if } A_{ij} = 0 \end{cases} \qquad \text{for } i = 2\dots V, j < i,\\
&z_{ji} = z_{ij}.
\end{aligned}
$$

Because the connection data are highly imbalanced – fewer than 5,000 connections out of a possible 514,605 – the intercepts β_0 and β_1 mix slowly in an ordinary DA Gibbs algorithm (Figure 7(a)). Without using DA, efficient MH proposals are difficult to develop due to the restriction that $U \in \mathbb{S}^{V\times d}$. The DA-Gibbs relies on the full conditional distribution

$$
U \mid \beta_0, \beta_1, Z \sim \text{Bingham}(\{z_{ij} - \beta_0 w_{ij} - \beta_1(1 - w_{ij})\}, \text{diag}\{v_r/2\}).
$$



(a) ACFs of the parameters β_0 and β_1 using DA.



(b) ACFs of the parameters β_0 and β_1 using CDA.

Figure 7: ACFs show the mixing performance of β_0 and β_1 in modeling average sparsity in network connectivity of a brain.

We use CDA to calibrate the updates of β_0 and β_1, while keeping the other Gibbs sampling steps unchanged, i.e.

$$z^*_{ij} \sim \begin{cases} \mathrm{No}_{(0,\infty)}(\psi_{ij} + b_{ij}, r_{ij}) & \text{if } A_{ij} = 1 \\ \mathrm{No}_{(-\infty,0)}(\psi_{ij} + b_{ij}, r_{ij}) & \text{if } A_{ij} = 0 \end{cases} \quad \text{for } i = 2 \dots V, j < i,$$

$$\beta^*_0 \sim \mathrm{No}\Big([\sum_{j<i} \frac{w_{ij}}{r_{ij}}]^{-1} \sum_{j<i} [\frac{w_{ij}}{r_{ij}} (z^*_{ij} - b_{ij} - \sum_{r=1}^{d} u_{ir} v_r u_{jr})], [\sum_{j<i} \frac{w_{ij}}{r_{ij}}]^{-1} \Big),$$

$$\beta^*_1 \sim \mathrm{No}\Big([\sum_{j<i} \frac{1-w_{ij}}{r_{ij}}]^{-1} \sum_{j<i} [\frac{1-w_{ij}}{r_{ij}} (z^*_{ij} - b_{ij} - \sum_{r=1}^{d} u_{ir} v_r u_{jr})], [\sum_{j<i} \frac{1-w_{ij}}{r_{ij}}]^{-1} \Big).$$

Then β^*_0 and β^*_1 are accepted via MH step with calibrated conditional density

$$L_{r,b}(\beta_0, \beta_1 \mid U, V, A) = \prod_{j<i} \Phi(\psi_{ij})^{A_{ij}} [1 - \Phi(\psi_{ij})]^{(1-A_{ij})}$$

$$\psi_{ij} = r_{ij}^{-1} [\sum_{r=1}^{d} u_{ir} v_r u_{jr} + \beta_0 w_{ij} + \beta_1 (1 - w_{ij}) + b_{ij}]$$

The tuning parameters are optimized using the approach described in Section 4, except with $x_i\theta$ replaced by $\sum_{r=1}^{d} u_{ir} v_r u_{jr} + \beta_0 w_{ij} + \beta_1 (1 - w_{ij})$.

	DA	CDA
β_0	-2.09 (-2.10, -2.08)	-2.09 (-2.10, -2.08)
β_1	-3.68 (-3.72, -3.64)	-3.75 (-3.86, -3.66)
Fitted AUC	90.5%	92.1%
T_{eff}/T	0.008	0.142
Avg Computing Time / T	2.0 sec	2.0 sec
Avg Computing Time / T_{eff}	251 sec	14.1 sec

Table 1: Parameter estimates and computing speed of DA and CDA in Bernoulli latent factor modeling of a brain network.

We run DA for 30,000 steps and CDA for 2,000 steps, so that they have approximately the same effective sample size (calculated with the `CODA` package in `R`). Both algorithms are initialized at the MAP estimates. CDA leads to significant reduction in autocorrelation (Figure 7(b)) and about 18 times lower computing time per effective sample size. We also compare the in-sample fitted AUCs, computed based on A_{ij} and the posterior mean of p_{ij}. The CDA estimates clearly have a better fit to the data.

7.2. Poisson Log-Normal Model for Web Traffic Prediction

As a second application, we apply CDA to an online browsing activity dataset obtained from a computational advertising company. The dataset contains a two-way table of visit count by users who browsed one of 96 websites belonging to clients of the computational advertising agency, and one of the $n = 59,792$ high-traffic sites during the same browsing

session. We refer to visiting more than one site during the same session as co-browsing. For each of the client websites, it is of commercial interest to identify the high-traffic sites with relatively high co-browsing rates, so that ads can be more effectively placed. In computational advertising, it is also valuable to understand the co-browsing behavior and predict the traffic pattern of users.

We consider a Poisson regression model for co-browsing. We use the co-browsing count of a single client website as the outcome y_i and the log of one plus the co-browsing count of the other 95 websites as the predictors, i.e. $x_{ij} = \log(x^*_{ij} + 1)$ for $i = 1, \dots, 59792$ and $j = 1, \dots, 95$, where x^* is the raw co-browsing count for high-traffic site i and client site j. A Gaussian random effect is included to account for over-dispersion relative to the Poisson distribution, leading to a Poisson log-normal regression model:

$$\begin{aligned} y_i \sim \text{Poisson}\left(\exp(x_i\beta + \tau_i)\right), \quad \tau_i \overset{iid}{\sim} \mathrm{No}(\tau_0, \nu^2), \quad i = 1 \dots n \\ \beta \sim \mathrm{No}(0, I\sigma^2_\beta), \quad \tau_0 \sim \mathrm{No}(0, \sigma^2_\tau) \quad \nu^2 \sim \pi(\nu^2). \end{aligned}$$

We assign a weakly informative prior for β and τ_0 with $\sigma^2_\beta = \sigma^2_\tau = 100$. For the over-dispersion parameter ν^2, we assign a non-informative flat prior on $(0, \infty)$.

When β and τ are sampled separately, the random effects $\tau = \{\tau_1, \dots, \tau_n\}$ mix slowly. Instead, we sample β and τ jointly. Letting $\tilde{X}$ be the $n \times (n+p)$ matrix given by $\tilde{X} = [I_n\, X]$, and $\eta_i = x_i\beta + \tau_i$ the linear predictor, $\theta = \{\tau, \beta\}'$ can be sampled jointly in a block. An explanation of improved mixing with blocked sampling can be found in Liu (1994a).

We now focus on the mixing behavior of data augmentation. We first review data augmentation for the Poisson log-normal model. Zhou et al. (2012) proposed to treat Poisson(η_i) as the limit of the negative binomial $\mathrm{NB}\big(\lambda, \eta_i/(\lambda + \eta_i)\big)$ with $\lambda \to \infty$, and used moderate $\lambda = 1,000$ for approximation. The limit relationship, omitting constants, is given by

$$L(\eta_i; y_i) = \frac{\exp(y_i\eta_i)}{\exp\{\exp(\eta_i)\}} = \lim_{\lambda\to\infty} \frac{\exp(y_i\eta_i)}{\{1 + \exp(\eta_i)/\lambda\}^\lambda}. \tag{15}$$

With finite λ approximation, the posterior can be sampled using Polya-Gamma data augmentation

$$\begin{aligned} z_i \mid \eta_i \sim \mathrm{PG}(\lambda, \eta_i - \log\lambda) \quad i = 1 \dots n \\ \theta \mid z, y \sim \mathrm{No}\Big(\Big(\tilde{X}' Z \tilde{X} + \begin{bmatrix} 1/\nu^2 \cdot I_n & 0 \\ 0 & 1/\sigma^2_\beta \cdot I_p \end{bmatrix} \Big)^{-1} \\ \Big\{ \tilde{X}'\big(y - \lambda/2 + Z\log\lambda\big) + \begin{bmatrix} \tau_0/\nu^2 1_n \\ 0_p \end{bmatrix} \Big\}, \\ \Big(\tilde{X}' Z \tilde{X} + \begin{bmatrix} 1/\nu^2 \cdot I_n & 0 \\ 0 & 1/\sigma^2_\beta \cdot I_p \end{bmatrix} \Big)^{-1} \Big), \end{aligned}$$

where $Z = \mathrm{diag}\{z_1, \dots, z_n\}$, $1_n = \{1, \dots, 1\}'$ and $0_p = \{0, \dots 0\}'$.

However, this approximation-based data augmentation is inherently problematic. For example, setting $\lambda = 1,000$ leads to large approximation error. As in (15), the approximating denominator has $(1 + \exp(\eta_i)/\lambda)^\lambda = \exp\{\exp(\eta_i) + \mathcal{O}(\exp(2\eta_i)/\lambda)\}$; for moderately large $\eta_i \approx 10$, λ needs to be at least 10^9 to make $\exp(2\eta_i)/\lambda$ close to 0. This large error

cannot be corrected with an additional MH step, since the acceptance rate would be too low. On the other hand, it is not practical to use a large λ in a Gibbs sampler, as it would create extremely large z_i (associated with small conditional covariance for θ), resulting in slow mixing.

We use CDA to circumvent this issue. We first choose a very large λ (10^9) to control the approximation error, then use a small fractional r_i multiplying to λ for calibration. This leads to a proposal likelihood similar to the logistic CDA:

$$L_{r,b}(x_i\theta; y_i) = \frac{\exp(\eta_i - \log\lambda + b_i)^{y_i}}{\{1+\exp(\eta_i - \log\lambda + b_i)\}^{r_i\lambda}},$$

with $r_i \geq (y_i - 1)/\lambda + \epsilon$ for proper likelihood, and proposal update rule:

$$\begin{aligned} z_i &\sim \mathrm{PG}(r_i\lambda, \eta_i - \log\lambda + b_i) \quad i = 1 \dots n \\ \theta^* &\sim \mathrm{No}\Big(\Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2 \cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1} \\ &\quad \Big\{\tilde{X}'\big(y - r\lambda/2 + Z\log(\lambda - b)\big) + \begin{bmatrix} \tau_0/\nu^2 1_n \\ 0_p \end{bmatrix}\Big\}, \\ &\quad \Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2 \cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1}\Big) \end{aligned}$$

Letting $\eta_i^* = \tilde{X}\theta^*$, the proposal is accepted with probability (based on the Poisson density and the approximation $L_{r,b}(x_i\theta; y_i)$):

$$\min\left\{1, \prod_i \frac{\exp\{\exp(\eta_i)\}}{\exp\{\exp(\eta_i^*)\}} \frac{\{1+\exp(\eta_i^* - \log\lambda + b_i)\}^{r_i\lambda}}{\{1+\exp(\eta_i - \log\lambda + b_i)\}^{r_i\lambda}}\right\}.$$

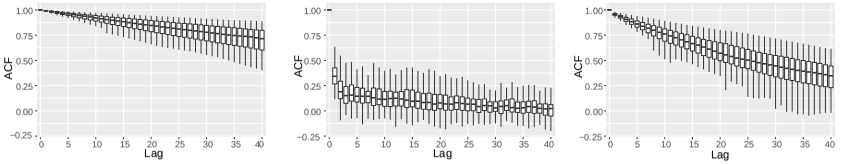
The tuning parameters are then optimized as described in Section 4, using

$$\begin{aligned} \tilde{z}_i(\theta) =& \frac{\lambda r_i}{2|\eta_i - \log\lambda + b_i|} \tanh\left(\frac{|\eta_i - \log\lambda + b_i|}{2}\right) \quad i = 1, \dots, n, \\ \tilde{\theta}^*(z) =& \Big(\tilde{X}'Z\tilde{X} + \begin{bmatrix} 1/\nu^2 \cdot I_n & 0 \\ 0 & 1/\sigma_\beta^2 \cdot I_p \end{bmatrix}\Big)^{-1} \\ & \Big\{\tilde{X}'\big(y - r\lambda/2 + Z\log(\lambda - b)\big) + \begin{bmatrix} \tau_0/\nu^2 1_n \\ 0_p \end{bmatrix}\Big\}. \end{aligned}$$

After θ is updated, the other parameters can be sampled via $\tau_0 \sim \mathrm{No}\big((n/\nu^2 + 1/\sigma_\tau^2)^{-1} \sum_i \tau_i/\nu^2, (n/\nu^2 + 1/\sigma_\tau^2)^{-1}\big)$ and $\nu^2 \sim \text{Inverse-Gamma}(n/2 - 1, \sum_i(\tau_i - \tau_0)^2/2)$.

We ran the ordinary DA algorithm with $\lambda = 1,000$, CDA with $\lambda = 10^9$ and Hamiltonian Monte Carlo with No-U-Turn sampler under the default tuning setting (as implemented in `STAN 2.17`). All algorithms are initialized at the MAP. We ran DA for $200,000$ steps, CDA for $2,000$ steps and HMC for $20,000$ steps so that they have approximately the same effective sample size. For CDA, we used the first $1,000$ steps for adapting r and b. Figure 8 shows empirical autocorrelations for DA, CDA and HMC. Even with small $\lambda = 1,000$ in DA, all of the parameters mix poorly; HMC seemed to be affected by the presence of random

effects, and most of parameters remain highly correlated within 40 lags; CDA substantially improves the mixing. Table 2 compares all three algorithms. CDA has the most efficient computing time per effective sample, and is about $30 - 300$ times more efficient than the other two algorithms.



(a) Autocorrelation of the parameters from DA.

(b) Autocorrelation of the parameters from CDA.

(c) Autocorrelation of the parameters from HMC.

Figure 8: CDA significantly improves the mixing of the parameters in the Poisson log-normal.

To evaluate predictive performance, we use another co-browsing count table for the same high traffic and client sites, collected during a different time period. We use the high traffic co-browsing count $x_{ij}^{\dagger *}$ and their log transform $x_{ij}^{\dagger} = \log(x_{ij}^{\dagger *} + 1)$ for the $j = 1, \dots, 95$ clients to predict the count for the client of interest $y_i^{\dagger}$, over the high traffic site $i = 1, \dots, 59792$. We predict using $\hat{y}_i^{\dagger} = \mathbb{E}_{\beta,\tau|y,x} y_i^{\dagger} = \mathbb{E}_{\beta,\tau|y,x} \exp(x_i^{\dagger}\beta + \tau_i)$ on the client site. The expectation is approximated using the MCMC sample path for $\beta, \tau \mid y, x$ obtained using training set $\{y, x\}$, as discussed above. Cross-validation root-mean-squared error $\big(\sum_i (\hat{y}_i^{\dagger} - y_i^{\dagger})^2/n\big)^{1/2}$ between the prediction and actual count $y_i^{\dagger}$'s is computed. As shown in Table 2, slow mixing in DA and HMC cause poor estimation of the parameters and high prediction error, while CDA has significantly lower error.

	DA	CDA	HMC
$\sum \beta_j/95$	0.072 (0.071, 0.075)	-0.041 (-0.042, -0.038)	-0.010 (-0.042, -0.037)
$\sum \beta_j^2/95$	0.0034 (0.0033, 0.0035)	0.231 (0.219 0.244)	0.232 (0.216 0.244)
$\sum \tau_i/n$	-0.405 (-0.642, -0.155)	-1.292 (-2.351, -0.446)	-1.297 (-2.354, -0.451)
$\sum \tau_i^2/n$	1.126 (0.968, 1.339)	3.608 (0.696, 7.928)	3.589 (0.678, 8.011)
Prediction RMSE	33.21	8.52	13.18
T_{eff}/T	0.0037 (0.0011 0.0096)	0.3348 (0.0279, 0.699)	0.0173 (0.0065, 0.0655)
Avg Comp. Time / T	1.3 sec	1.3 sec	56 sec
Avg Comp. Time / T_{eff}	346.4 sec	11.5 sec	3240.6 sec

Table 2: Parameter estimates, prediction error and computing speed of the DA, CDA and HMC in Poisson regression model.

8. Discussion

Data augmentation (DA) is a technique routinely used to enable implementation of simple Gibbs samplers, avoiding the need for expensive and complex tuning of Metropolis-Hastings algorithms. Despite the convenience, DA mixes slowly when the conditional posterior variance given the augmented data is substantially smaller than the marginal variance. When the data sample size is massive, this problem arises when the rates of convergence of the augmented and marginal posterior differ. There is a rich literature on strategies for improving mixing rates of Gibbs samplers, with centered or non-centered re-parameterizations (Papaspiliopoulos et al., 2007) and parameter-expansion (Liu and Wu, 1999) leading to some improvements. However, existing approaches do not solve large sample mixing problems because they do not address the fundamental rate mismatch issue.

To tackle this problem, we propose to calibrate data augmentation by directly adjusting the conditional variance (which is associated with step size). CDA adds a small cost for likelihood evaluation, which is often negligible when compared to the random number generation required at each iteration of DA. In this article, we demonstrate that calibration is generally applicable when $\theta \mid z$ belongs to a location-scale family. We expect it to be also useful outside of location-scale families, but have not pursued that here.

As both CDA and HMC involve MH steps, we draw some further comparison between the two. Both methods rely on finding a good proposal by searching a region far from the current state. One key difference lies in the computing efficiency. Although HMC is more generally applicable beyond data augmentation, it is computationally intensive since Hamiltonian dynamics often requires multiple numeric steps. CDA only requires one step of calibrated Gibbs sampling, which is often much more efficient, leveraging on existing data augmentation algorithms. The idea of using an auxiliary Gibbs chain to generate MH proposals seems generally promising (Tran et al., 2016), yet has received little attention in the literature.

In this work, we focused on cases when the sample size n is large, with the parameter dimension p moderate. One limitation of CDA-MH is that when p grows, in order to maintain a reasonable acceptance rate, the range to increase the conditional variance has to decrease. This is a common problem for general MH algorithms. Therefore, solutions to high dimensionality require further study.

Appendix A. Proof of Remark 1

Proof Since $q_{r,b}(\theta;\theta')$ is the θ marginal of a Gibbs transition kernel, and Gibbs is reversible on its margins, we have

$$q(\theta;\theta^*)\Pi_{r,b}(\theta^*) = q(\theta^*;\theta)\Pi_{r,b}(\theta),$$

and so

$$\begin{aligned}\frac{L(\theta^*;y)\Pi^0(\theta^*)q(\theta;\theta^*)}{L(\theta;y)\Pi^0(\theta)q(\theta^*;\theta)} &= \frac{L(\theta^*;y)\Pi^0(\theta^*)L_{r,b}(\theta;y)\Pi^0(\theta)}{L(\theta;y)\Pi^0(\theta)L_{r,b}(\theta^*;y)\Pi^0(\theta^*)}\\ &= \frac{L(\theta^*;y)L_{r,b}(\theta;y)}{L(\theta;y)L_{r,b}(\theta^*;y)}.\end{aligned}$$

■

Appendix B. Proof of Remark 2

Proof For any r, b, the conditionals $\Pi_{r,b}(z \mid \theta)$ and $\Pi_{r,b}(\theta \mid z)$ are well-defined for all $z \in \mathcal{Z}, \theta \in \Theta$, and therefore the Gibbs transition kernel $K_{r,b}((\theta, z);\cdot)$ and corresponding marginal kernels $Q_{r,b}(\theta;\cdot)$ are well-defined. Moreover, for any $(z,\theta) \in \mathcal{Z} \times \Theta$, we have $\mathbb{P}[(\theta', z') \in A \mid (\theta, z)] > 0$ by assumption. Thus $K_{r,b}$ is aperiodic and $\Pi_{r,b}$-irreducible (see the discussion following Corollary 1 in Roberts and Smith (1994)).

$Q_{r,b}(\theta';\theta)$ is aperiodic and $\Pi_{r,b}(\theta)$-irreducible, since it is the θ marginal transition kernel induced by $K_{r,b}((\theta, z);\cdot)$. Thus, it is also $\Pi(\theta)$-irreducible so long as $\Pi \gg \Pi_{r,b}$, where for two measure μ, ν, $\mu \gg \nu$ indicates absolute continuity. Since $\Pi, \Pi_{r,b}$ have densities with respect to Lebesgue measure, $\Pi_{r,b}$-irreducibility implies Π irreducibility. Moreover, $q(\theta;\theta') > 0$ for all $\theta \in \Theta$. Thus, by Theorem 3 of Roberts and Smith (1994), CDA MH is Π-irreducible and aperiodic. ■

Appendix C. Toy example: Hierarchical Normal

To demonstrate the effects of r, b, we use a toy example commonly used in the data augmentation literature (Liu and Wu, 1999). Consider a marginal Normal model

$$y_i \sim \mathrm{No}(\theta, \sigma^2 + 1) \qquad i = 1, \ldots, n$$

with σ^2 known and improper prior $\pi(\theta) \propto 1$. This can be considered as a hierarchical model

$$y_i \sim \mathrm{No}(z_i, \sigma^2), \quad z_i \sim \mathrm{No}(\theta, 1), \qquad i = 1, \ldots, n, \tag{16}$$

where $z = \{z_1, \ldots, z_n\}$ are augmented data. The standard data augmentation algorithm has the update rule

$$\begin{aligned} z_i \mid y, \theta &\sim \mathrm{No}\left(\frac{y_i\sigma^{-2} + \theta}{\sigma^{-2} + 1}, \frac{1}{\sigma^{-2} + 1}\right) \qquad i = 1, \ldots, n\\ \theta \mid z &\sim \mathrm{No}(n^{-1}\sum_i z_i, n^{-1}).\end{aligned}$$

Thanks to the simple form, it is straightforward to compute the marginal variance of θ, $\mathrm{var}(\theta \mid y) = n^{-1}(1 + \sigma^2)$. Clearly, this is larger than the conditional variance $\mathbb{E}_z\mathrm{var}(\theta' \mid z) = n^{-1}$, when σ^2 is large.

To be able to adjust the conditional variance, we consider an alternative hierarchical model

$$y_i \sim \mathrm{No}(z_i, \sigma^2), \quad z_i \sim \mathrm{No}(\theta + b_0, r_0), \qquad i = 1, \ldots, n,$$

with update rule

$$\begin{aligned} z_i \mid y, \theta &\sim \text{No}\left(\frac{y_i\sigma^{-2} + (\theta + b_0)r_0^{-1}}{\sigma^{-2} + r_0^{-1}}, \frac{1}{\sigma^{-2} + r_0^{-1}}\right) \qquad i = 1, \ldots, n \\ \theta^* \mid z &\sim \text{No}(n^{-1}\sum_i z_i - b_0, n^{-1}r_0). \end{aligned} \tag{17}$$

To correct the deviation caused by the alternative model, we treat θ^* as a proposal to the target model (16), using M-H as in Remark 1 with $L_{r,b}(\theta; y) = (r_0 + \sigma^2)^{-1/2}\phi[(r_0 + \sigma^2)^{-1/2}(y_i - \theta - b_0)]$ and ϕ the standard normal density. We can choose r_0 so that the proposal variance equals to the target marginal variance $\text{var}_{r,b}(\theta' \mid z) = \text{var}(\theta \mid y)$; this yields

$$r_0 = 1 + \sigma^2.$$

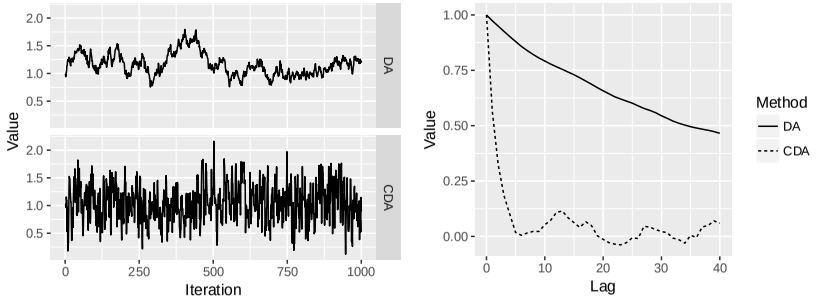
Note the proposal mean has

$$\mathbb{E}(\theta^* \mid \theta) = \mathbb{E}_{z|\theta}\mathbb{E}_{\theta^*|z}(\theta^* \mid z) = \theta + \frac{n^{-1}\sum_i y_i r_0}{\sigma^2 + r_0} - \frac{(b_0 + \theta)r_0}{\sigma^2 + r_0}$$

Intuitively, one way to improve the acceptance rate is to have the proposal centered at the current θ in the high posterior density region. That is, $\mathbb{E}(\theta^* \mid \theta) \approx \theta$ for θ near the MAP $\hat{\theta} = n^{-1}\sum_i y_i$. This yields one choice for b_0

$$\frac{n^{-1}\sum_i y_i r_0}{\sigma^2 + r_0} - \frac{(b_0 + \hat{\theta})r_0}{\sigma^2 + r_0} = 0 \qquad \Rightarrow b_0 = 0$$

We use $\sigma^2 = 100$, $\theta = 1$ to simulate $n = 1000$ data. Figure 9 compares the mixing performance, in terms of traceplots and autocorrelation plots (ACF) for the original DA and calibrated DA. Each algorithm was initiated at the MAP $\hat{\theta} = n^{-1}\sum_i y_i$. CDA significantly improves the mixing performance, with acceptance rate approximately 0.9.



(a) Traceplot for DA and CDA.

(b) ACF for DA and CDA.

Figure 9: Trace and autocorrelation plots for DA and CDA in hierarchical normal model.

Note in this special example, instead of relying on $\text{var}_{r,b}(\theta^* \mid z)$, one could directly adjust $\text{var}_{r,b}(\theta^* \mid \theta) = [r_0^2 + 2r_0\sigma^2]/[n(r_0 + \sigma_0^2)]$ to match $\text{var}(\theta \mid y)$. However, in general non-Gaussian cases, $\text{var}_{r,b}(\theta^* \mid \theta)$ is intractable, so we expect adjusting $\text{var}(\theta^* \mid z)$ to be more useful.

Appendix D. Calibrated Polya-Gamma Algorithm with Sub-sampling

Adapting based on Johndrow et al. (2017), we first randomly sample a subset of indices V of size $|V|$. This algorithm generates proposals from

$$\begin{aligned} &V = V_1 \cup V_0, \quad V_1 = \{i \in \{1, \ldots, n\} : y_i = 1\}, \quad V_0 \sim \text{Subset}(|V|, \{i \in \{1, \ldots, n\} : y_i = 0\}) \\ &z_i \sim \text{PG}(k_i r_i, |x_i\theta + b_i|) \quad i \in V, \\ &\theta^* \sim \text{No}\left((X_V' Z_V X_V)^{-1} X_V'(y_V - k_V r_V/2 - Z_V b_V), (X_V' Z_V X_V)^{-1}\right), \end{aligned}$$

where subscript $._V$ indicates the sub-matrix or sub-vector corresponding to the sub-sample; $k_i = 1$ if $y_i = 1$, and $k_i = (n - |V_1|)/|V_0|$. We accept θ^* in an MH step using calibrated likelihood

$$L_{r,b}(\theta; y) = \prod_{i \in V_1} \frac{\exp(x_i\theta + b_i)}{\{1 + \exp(x_i\theta + b_i)\}^{r_i}} \Big(\prod_{i \in V_0} \frac{1}{\{1 + \exp(x_i\theta + b_i)\}^{r_i}}\Big)^{\frac{n - |V_1|}{|V_0|}},$$

with target approximate likelihood $L_{1,0}(\theta; y)$.

Appendix E. Proof of Theorem 1

Proof Let Q be the proposal kernel for CDA-MH, which is identically the transition kernel for CDA-Gibbs, and let $\mathcal{P}$ be the Markov transition semigroup of CDA-MH.

Both have densities with respect to Lebesgue measure given by

$$\begin{aligned} q(\theta, \theta') &= \int_{\mathcal{Z}} f_{r,b}(\theta' \mid z, y)\pi_{r,b}(z \mid \theta, y)dz \\ p(\theta, \theta') &= \alpha(\theta, \theta')q(\theta, \theta') + \delta_\theta(\theta')\left(1 - \int \alpha(\theta, \tilde{\theta})q(\theta, \tilde{\theta})d\tilde{\theta}\right), \end{aligned}$$

respectively.

We seek a constant $c > 0$ and a density g such that

$$\inf_{\theta \in \Theta} p(\theta, \theta') > cg(\theta')$$

We proceed in the following steps:

1. Show that there exists a constant $c_1 > 0$ and a density g such that $\int_\Theta g(\theta)d\theta = 1$ for which

$$\inf_{\theta \in \Theta} q(\theta, \theta') \geq c_1 g(\theta');$$

conclude that CDA-Gibbs is uniformly ergodic.

2. Show that there exists $S \subset \Theta$ and a constant $c_2 > 0$ such that

$$\inf_{\theta\in\Theta,\theta'\in S} \alpha(\theta,\theta') > c_2.$$

3. Combine 1 and 2 to show $p(\theta,\theta') \geq \kappa g_S(\theta')$, where $\kappa = c_1c_2c_3$ with

$$c_3 = \int_S g(\theta)d\theta, \quad g_S(\theta) = c_3^{-1}g(\theta)\mathbf{1}\{\theta \in S\}$$

the restriction of g to S. Conclude that CDA-MH is uniformly ergodic with spectral gap κ.

4. Find values (r_0, b_0, S_0) of the tuning parameters r, b, S so that κ goes to zero slowly as $n \to \infty$.

1. Show that $q(\theta,\theta') \geq c_1 g(\theta')$. First we bound $\pi_{r,b}(\theta' \mid z)$ by a constant times a function depending on z

$$\begin{aligned}
\pi_{r,b}(\theta' \mid z) &= (2\pi)^{-1/2}(z+1/\sigma^2)^{1/2}\exp\left[-\frac{1}{2}(\theta'-m)(z+1/\sigma^2)(\theta'-m)\right]\\
&=(2\pi)^{-1/2}(z+1/\sigma^2)^{1/2}\exp\left[-\frac{1}{2}\left\{(\theta'+b)(z+1/\sigma^2)(\theta'+b)-2a(\theta'+b)+\frac{a^2}{z+1/\sigma^2}\right\}\right]\\
&>(2\pi)^{-1/2}(1/\sigma^2)^{1/2}\exp\left[-\frac{1}{2}\left\{(\theta'+b)(z+1/\sigma^2)(\theta'+b)-2a(\theta'+b)+\frac{a^2}{1/\sigma^2}\right\}\right]\\
&=(2\pi)^{-1/2}\sigma^{-1}\exp\left[-\frac{1}{2}\left\{(\theta'+b)^2/\sigma^2-2a(\theta'+b)+a^2\sigma^2\right\}\right]\exp\left[-\frac{1}{2}\left\{(\theta'+b)^2 z\right\}\right]
\end{aligned} \tag{18}$$

in which the inequality holds since $z > 0$.

Using the Laplace transform of $\omega \sim \mathrm{PG}(\alpha,\beta)$

$$\mathbb{E}[\exp(-\omega t)] = \frac{\cosh^\alpha(\beta/2)}{\cosh^\alpha(\sqrt{(\beta^2/2+t)/2})},$$

we proceed to bound the expectation of (18) with respect to z

$$\begin{aligned}
\int_0^\infty \exp\left[-\frac{1}{2}\left\{(\theta'+b)^2 z\right\}\right]\pi_{r,b}(z \mid \theta)dz &= \cosh^{nr}(\frac{|\theta+b|}{2})\cosh^{-nr}(\frac{\sqrt{((\theta+b)^2+(\theta'+b)^2)}}{2})\\
&\geq \cosh^{nr}(\frac{|\theta+b|}{2})\cosh^{-nr}(\frac{(|\theta+b|+|\theta'+b|)}{2})\\
&\geq 2^{-nr}\cosh^{nr}(\frac{|\theta+b|}{2})\cosh^{-nr}(\frac{|\theta+b|}{2})\cosh^{-nr}(\frac{|\theta'+b|}{2})\\
&= 2^{-nr}\cosh^{-nr}(\frac{|\theta'+b|}{2})\\
&\geq 2^{-nr}\exp\left[-\frac{nr|\theta'+b|}{2}\right]
\end{aligned}$$

$$\geq 2^{-nr}\exp\left[-\frac{nr[(\theta'+b)^2+1]}{4}\right]$$

where the first inequality uses $a^2+b^2 \leq (|a|+|b|)^2$; the second uses Lemma 3.2 of Choi and Hobert (2013); the third uses the property of cosh; and the fourth uses $|a| \leq (1+a^2)/2$. We combine to obtain $q(\theta,\theta') > c_1 g(\theta')$, viz

$$\begin{aligned}
q(\theta,\theta') &= \int_0^\infty \pi_{r,b}(\theta' \mid z, y_{1:n})\pi_{r,b}(z \mid \theta, y_{1:n})dz\\
&>(2\pi)^{-1/2}\sigma^{-1}\exp\left[-\frac{1}{2}\left\{(\theta'+b)^2/\sigma^2-2a(\theta'+b)+a^2\sigma^2\right\}\right]2^{-nr}\exp\left[-\frac{nr[(\theta'+b)^2+1]}{4}\right]\\
&=(2\pi)^{-1/2}\sigma^{-1}2^{-nr}\exp\left[-\frac{1}{2}\left\{(\theta'+b)^2(\frac{1}{\sigma^2}+\frac{nr}{2})-2a(\theta'+b)\right\}\right]\exp\left[-\frac{nr}{4}-\frac{a^2\sigma^2}{2}\right]\\
&=\sigma^{-1}2^{-nr}(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1/2}\exp\left[-\frac{nr}{4}-\frac{a^2\sigma^2}{2}\right]\exp\left[\frac{1}{2}a^2(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}\right]\\
&\quad(2\pi)^{-1/2}(\frac{1}{\sigma^2}+\frac{nr}{2})^{1/2}\exp\left[-\frac{1}{2}\left(\theta'+b-(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}a\right)^2(\frac{1}{\sigma^2}+\frac{nr}{2})\right]\\
&=c_1 g(\theta')
\end{aligned}$$

where

$$\begin{aligned}
c_1 &= \sigma^{-1}2^{-nr}(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1/2}\exp\left[-\frac{nr}{4}-\frac{a^2\sigma^2}{2}\right]\exp\left[\frac{1}{2}a^2(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}\right]\\
g(\theta') &= \mathrm{No}\left[\theta' \mid (\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}a-b, (\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}\right].
\end{aligned}$$

This completes the first part.

2. Show that $\inf_{\theta\in\Theta,\theta'\in S}\alpha(\theta,\theta') > c_2$ for some set S.

The acceptance ratio is

$$\alpha(\theta,\theta') = \min\left(\left[\frac{\alpha_0(\theta')}{\alpha_0(\theta)}\right]^n, 1\right), \qquad \alpha_0(x) = \frac{\{1+\exp(x+b)\}^r}{1+\exp(x)}$$

Differentiating with respect to x we obtain

$$\frac{\partial\alpha_0(x)}{\partial x} = \frac{e^x\left(e^{b+x}+1\right)^{r-1}\left((r-1)e^b e^x+e^b r-1\right)}{\left(e^x+1\right)^2},$$

Assuming that $r < 1$ and $e^b r > 1$, since

$$\frac{e^x\left(e^{b+x}+1\right)^{r-1}}{\left(e^x+1\right)^2} > 0$$

and there is only one root on $(-\infty,\infty)$ for

$$(r-1)e^b e^x + e^b r - 1 = 0 \quad\Longrightarrow\quad x = \log(\frac{e^b r-1}{1-r}) - b \equiv \hat{\theta}$$

$$(r-1)e^b e^x + e^b r - 1 < 0 \quad \Longrightarrow \quad x > \hat{\theta}$$
$$(r-1)e^b e^x + e^b r - 1 > 0 \quad \Longrightarrow \quad x < \hat{\theta}$$

Therefore, $\hat{\theta}$ is the unique mode of α_0, and α_0 is (1) monotonically increasing for $x < \hat{\theta}$ and monotonically decreasing for $x > \hat{\theta}$.

For convenience, we write $b = -\log(r) + \xi$ with $\xi > 0$, so that $1 + \exp(\hat{\theta} + b) = (e^\xi - r)/(1 - r)$. Now set $S = (s_1, s_2)$. We now show that $\alpha(\theta, \theta') > c_2$ for $\theta' \in S$. We proceed in two cases.

1. **Case 1**: $\theta \le \hat{\theta}$. We have three subcases

   (a) If $\theta < \theta' \le \hat{\theta}$, then $\alpha_0(\theta') \ge \alpha_0(\theta)$, $\alpha(\theta, \theta') = 1$.

   (b) If $s_1 < \theta' \le \theta \le \hat{\theta}$ then

$$\begin{aligned}\alpha(\theta,\theta') =& \left(\frac{1+\exp(\theta)}{1+\exp(\theta')}\right)^n \left(\frac{1+\exp(\theta'+b)}{1+\exp(\theta+b)}\right)^{rn} \\ \ge& 1 \times \quad \left(\frac{1}{1+\exp(\hat{\theta}+b)}\right)^{rn} = \left(\frac{1-r}{e^\xi - r}\right)^{rn}\end{aligned}$$

   (c) If $\theta \le \hat{\theta} < \theta' < s_2$,

$$\begin{aligned}\alpha(\theta,\theta') =& \left(\frac{1+\exp(\theta)}{1+\exp(\theta')}\right)^n \left(\frac{1+\exp(\theta'+b)}{1+\exp(\theta+b)}\right)^{rn} \\ \ge& \left(\frac{1}{1+\exp(\theta')}\right)^n \times 1 \ge \left(\frac{1}{1+\exp(s_2)}\right)^n\end{aligned}$$

   where we used that $\theta' > \theta$ so the second term is bounded below by 1, and that $1 + e^\theta > 1$. If $s_2 \le \hat{\theta}$, then $\theta' < \hat{\theta}$, we only need to consider the condition (a) and (b).

2. **Case 2:** $\theta > \hat{\theta}$,

   (a) If $\hat{\theta} < \theta' \le \theta$, then $\alpha_0(\theta') \ge \alpha_0(\theta)$, $\alpha(\theta, \theta') = 1$.

   (b) If $s_1 < \theta' \le \hat{\theta} < \theta$, because α_0 is monotone nondecreasing on $(-\infty, \hat{\theta})$, we have:

$$\alpha_0(\theta') = \frac{\{1+\exp(\theta'+b)\}^r}{1+\exp(\theta')} \ge \lim_{\theta' \to -\infty} \alpha_0(\theta') = 1,$$

   Further, because α_0 is monotone nonincreasing on $(\hat{\theta}, \infty)$ we have

$$\begin{aligned}\frac{1}{\alpha_0(\theta)} &= \frac{1+\exp(\theta)}{\{1+\exp(\theta+b)\}^r} \ge \frac{1}{\alpha_0(\hat{\theta})} \\ \alpha(\theta,\theta') &= \alpha_0(\theta')\frac{1}{\alpha_0(\theta)}\end{aligned}$$

$$\begin{aligned}&\ge 1 \times \frac{1}{\alpha_0(\hat{\theta})} = \frac{\{1+\exp(\hat{\theta})\}^n}{\{1+\exp(\hat{\theta}+b)\}^{rn}} \\ &\ge \frac{1}{\{1+\exp(\hat{\theta}+b)\}^{rn}} = \left(\frac{1-r}{e^\xi - r}\right)^{rn}\end{aligned}$$

   (c) If $\hat{\theta} < \theta < \theta' < s_2$,

$$\begin{aligned}\alpha(\theta,\theta') &= \left(\frac{1+\exp(\theta)}{1+\exp(\theta')}\right)^n \left(\frac{1+\exp(\theta'+b)}{1+\exp(\theta+b)}\right)^{rn} \\ &\ge \left(\frac{1}{1+\exp(\theta')}\right)^n \times 1 = \left(\frac{1}{1+\exp(s_2)}\right)^n\end{aligned}$$

If $s_2 \le \hat{\theta}$, then $\theta' < \hat{\theta}$ and we only need to consider the condition (b).

Combining (1) and (2), even when $s_2 \le \hat{\theta}$, the lower bound still has:

$$\left(\frac{1-r}{e^\xi - r}\right)^{rn} \ge \min\left\{\left(\frac{1-r}{e^\xi - r}\right)^{rn}, \left(\frac{1}{1+\exp(s_2)}\right)^n\right\}$$

Therefore we have the common lower bound:

$$\alpha(\theta,\theta') \ge c_2, \quad c_2 = \min\left\{\left(\frac{1-r}{e^\xi - r}\right)^{rn}, \left(\frac{1}{1+\exp(s_2)}\right)^n\right\}$$

for $\theta' \in (s_1, s_2)$. Since this does not depend on s_1, we take $s_1 = -\infty$.

3. Combine to show $p(\theta, \theta') \ge c_1 c_2 c_3 g_S(\theta')$

Since

$$\begin{aligned}p(\theta,\theta') &= \alpha(\theta,\theta')q(\theta,\theta') + \delta_\theta(\theta')\left(1 - \int \alpha(\theta,\tilde{\theta})q(\theta,\tilde{\theta})d\tilde{\theta}\right), \\ &\ge \alpha(\theta,\theta')q(\theta,\theta'),\end{aligned}$$

parts (1) and (2) establish the bound

$$\begin{aligned}\inf_{\theta\in\Theta} p(\theta,\theta') &\ge c_1 c_2 g(\theta')\mathbf{1}\{\theta' \in S\} \\ &= c_1 c_2 c_3 c_3^{-1} g(\theta')\mathbf{1}\{\theta' \in S\},\end{aligned}$$

where

$$c_3 = \int g(\theta')\mathbf{1}\{\theta' \in S\},$$

so that $g_S(\theta') = c_3^{-1} g(\theta')\mathbf{1}\{\theta' \in S\}$ is a density. Specifically we have

$$g_S(\theta') = c_3^{-1}\left(\frac{1}{\sigma^2} + \frac{nr}{2}\right)^{1/2} \phi\left(\frac{\theta' - \left((\frac{1}{\sigma^2} + \frac{nr}{2})^{-1}a - b\right)}{(\frac{1}{\sigma^2} + \frac{nr}{2})^{-1/2}}\right)\mathbf{1}\{\theta' \in S\}$$

for $\phi(\cdot)$ the standard Gaussian density, where

$$c_3 = \Phi\left\{\left(\frac{1}{\sigma^2}+\frac{nr}{2}\right)^{1/2}\left[s_2-(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1}a+b\right]\right\}$$
$$= \Phi\left[\left(\frac{1}{\sigma^2}+\frac{rn}{2}\right)^{1/2}(s_2+b)-\left(\frac{1}{\sigma^2}+\frac{rn}{2}\right)^{-1/2}a\right].$$

It follows that $\mathcal{P}$ is uniformly ergodic with spectral gap at least $\kappa = c_1c_2c_3$.

4. Tune constants so that $\kappa \to 0$ slowly as $n \to \infty$

We now may choose r, b, S in such a way as to minimize the rate at which the spectal gap goes to zero, subject to the constraints on r, b from part (2) and

$$\begin{aligned}\kappa(r,b,S) &= c_1c_2c_3\\ &= \sigma^{-1}2^{-nr}(\frac{1}{\sigma^2}+\frac{nr}{2})^{-1/2}\exp\left[-\frac{nr}{4}-\frac{a^2\sigma^2}{2}\right]\exp\left[\frac{1}{2}a^2\left(\frac{1}{\sigma^2}+\frac{nr}{2}\right)^{-1}\right]\\ &\times\min\left\{\left(\frac{1-r}{e^\xi-r}\right)^{rn},\left(\frac{1}{1+\exp(s_2)}\right)^n\right\}\\ &\times\Phi\left[\left(\frac{1}{\sigma^2}+\frac{rn}{2}\right)^{1/2}(s_2+b)-\left(\frac{1}{\sigma^2}+\frac{rn}{2}\right)^{-1/2}a\right]\end{aligned}$$

First, we note that because $b = \xi - \log r$, tuning of r, ξ is equivalent to tuning of r, b, so we elect to do the former.

First, to reduce the effect of n, we set $r = w/n$, with $0 < w < n$. Noting $\exp(-a^2\sigma^2/2)$ decreases rapidly in a and recalling $a = \sum_i y_i - nr/2 + b/\sigma^2$ and $b = \xi - \log r$, we solve for w to make $a = 0$

$$\sum_i y_i - w/2 + (-\log(w)+\log(n)+\xi)/\sigma^2 = 0$$
$$\frac{\log w}{\sigma^2}+\frac{w}{2} = \sum_i y_i + \frac{\log n}{\sigma^2}+\frac{\xi}{\sigma^2}$$

assuming $\sum y_i + \xi/\sigma^2 = o(\log(n))$, we have

$$w = 2\log(n)/\sigma^2 + o(\log(n)).$$

Second, we make c_3 a constant independent of y, n, by choosing s_2 such that

$$(\frac{1}{\sigma^2}+\frac{rn}{2})^{1/2}(s_2+b)-(\frac{1}{\sigma^2}+\frac{rn}{2})^{-1/2}a = 0$$
$$(\frac{1}{\sigma^2}+\frac{rn}{2})^{1/2}(s_2+b) = 0$$
$$s_2 = -b = \log(w)-\log(n)-\xi.$$

which yields $c_3 = 0.5$.

Third, choose ξ so that

$$\xi \le \log\left\{\left(1-\frac{w}{n}\right)e+\frac{w}{n}\right\} \Longrightarrow \log\left(\frac{e^\xi - w/n}{1-w/n}\right) \le 1$$

meaning

$$\left(\frac{1-r}{e^\xi-r}\right)^{rn} = \left(\frac{1-w/n}{e^\xi-w/n}\right)^w = \exp\left(-w\log\left(\frac{e^\xi-w/n}{1-w/n}\right)\right) \ge e^{-w}$$

and

$$\left(\frac{1}{1+\exp(s_2)}\right)^n = \left(\frac{1}{1+we^{-\xi}/n}\right)^n \ge \exp(-we^{-\xi}) \ge e^{-w}$$

We have

$$c_2 = \min\left\{\left(\frac{1-r}{e^\xi-r}\right)^{rn},\left(\frac{1}{1+\exp(s_2)}\right)^n\right\} \ge \exp(-w).$$

Combining results and choosing $r = r_0 = w/n$, $b = b_0 = -\log(w)+\log(n)+\xi$, $S = S_0 = (-\infty, \log(w)-\log(n)-\xi)$, with $(w,\xi) : \sum_i y_i - w/2 + (-\log(w)+\log(n)+\xi)/\sigma^2 = 0, \xi \le \log[(1-w/n)e + w/n]$, we have

$$\begin{aligned}\kappa(r_0,b_0,S_0) =& \sigma^{-1}2^{-w-1}(\frac{1}{\sigma^2}+\frac{w}{2})^{-1/2}\exp\left[-\frac{w}{4}\right]\exp(-w)\\ =& \mathcal{O}\Big(\exp\big[-(\frac{5}{4}+\log 2)w\big]\Big)\\ =& \mathcal{O}\left(n^{-\frac{5/2+2\log 2}{\sigma^2}}\right).\end{aligned}$$

■

References

James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

Rémi Bardenet, Arnaud Doucet, and Chris Holmes. On Markov chain Monte Carlo methods for tall data. *The Journal of Machine Learning Research*, 18(1):1515–1557, 2017.

Patrick R Conrad, Youssef M Marzouk, Natesh S Pillai, and Aaron Smith. Accelerating asymptotically exact MCMC for computationally intensive models via local approximations. *Journal of the American Statistical Association*, 111(516):1591–1607, 2016.

Bradley Efron and David V Hinkley. Assessing the accuracy of the maximum likelihood estimator: observed versus expected Fisher information. *Biometrika*, 65(3):457–483, 1978.

Peter D Hoff. Simulation of the matrix Bingham–von Mises–Fisher distribution, with applications to multivariate and relational data. *Journal of Computational and Graphical Statistics*, 18(2):438–456, 2009.

James E Johndrow, Jonathan C Mattingly, Sayan Mukherjee, and David B Dunson. Optimal approximating Markov chains for Bayesian inference. *arXiv preprint arXiv:1508.03387*, 2017.

James E Johndrow, Aaron Smith, Natesh Pillai, and David B Dunson. MCMC for imbalanced categorical data. *Journal of the American Statistical Association*, (in press):1–44, 2018.

Anoop Korattikara, Yutian Chen, and Max Welling. Austerity in MCMC land: Cutting the Metropolis-Hastings budget. In *International Conference on Machine Learning*, pages 181–189, 2014.

Jun S Liu. The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem. *Journal of the American Statistical Association*, 89(427): 958–966, 1994a.

Jun S Liu. The fraction of missing information and convergence rate for data augmentation. *Computing Science and Statistics*, pages 490–490, 1994b.

Jun S Liu and Ying Nian Wu. Parameter expansion for data augmentation. *Journal of the American Statistical Association*, 94(448):1264–1274, 1999.

Dougal Maclaurin and Ryan P Adams. Firefly Monte Carlo: exact MCMC with subsets of data. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, pages 543–552, 2015.

Daniel Marcus, John Harwell, Timothy Olsen, Michael Hodge, Matthew Glasser, Fred Prior, Mark Jenkinson, Timothy Laumann, Sandra Curtiss, and David Van Essen. Informatics and data mining tools and strategies for the human connectome project. *Frontiers in neuroinformatics*, 5:4, 2011.

Xiao-Li Meng and David A Van Dyk. Seeking efficient data augmentation schemes via conditional and marginal augmentation. *Biometrika*, 86(2):301–320, 1999.

Kerrie L Mengersen, Richard L Tweedie, et al. Rates of convergence of the Hastings and Metropolis algorithms. *The Annals of Statistics*, 24(1):101–121, 1996.

Stanislav Minsker, Sanvesh Srivastava, Lizhen Lin, and David B Dunson. Robust and scalable Bayes via a median of subset posterior measures. *The Journal of Machine Learning Research*, 18(1):4488–4527, 2017.

EWT Ngai, Yong Hu, YH Wong, Yijun Chen, and Xin Sun. The application of data mining techniques in financial fraud detection: a classification framework and an academic review of literature. *Decision Support Systems*, 50(3):559–569, 2011.

Omiros Papaspiliopoulos, Gareth O Roberts, and Martin Sköld. A general framework for the parametrization of hierarchical models. *Statistical Science*, pages 59–73, 2007.

Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using Pólya–Gamma latent variables. *Journal of the American Statistical Association*, 108(504):1339–1349, 2013.

Matias Quiroz, Robert Kohn, Mattias Villani, and Minh-Ngoc Tran. Speeding up MCMC by efficient data subsampling. *Journal of the American Statistical Association*, (in press): 1–35, 2018.

Gareth O Roberts and Adrian FM Smith. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and Their Applications*, 49(2):207–216, 1994.

Donald B Rubin. *Multiple imputation for nonresponse in surveys*, volume 81. John Wiley & Sons, 2004.

Daniel Rudolf, Nikolaus Schweizer, et al. Perturbation theory for Markov chains via Wasserstein distance. *Bernoulli*, 24(4A):2610–2639, 2018.

Sanvesh Srivastava, Volkan Cevher, Quoc Tran-Dinh, and David B Dunson. WASP: scalable Bayes via Barycenters of subset posteriors. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics*, pages 912–920, 2015.

Martin A Tanner and Wing Hung Wong. The calculation of posterior distributions by data augmentation. *Journal of the American Statistical Association*, 82(398):528–540, 1987.

Minh-Ngoc Tran, Michael K Pitt, and Robert Kohn. Adaptive Metropolis–Hastings sampling using reversible dependent mixture proposals. *Statistics and Computing*, 26(1-2): 361–381, 2016.

Jon Wakefield. Disease mapping and spatial regression with count data. *Biostatistics*, 8(2): 158–183, 2007.

Xuerui Wang, Wei Li, Ying Cui, Ruofei Zhang, and Jianchang Mao. Click-through rate estimation for rare events in online advertising. *Online Multimedia Advertising: Techniques and Technologies*, pages 1–12, 2010.

Mingyuan Zhou, Lingbo Li, David B Dunson, and Lawrence Carin. Lognormal and Gamma mixed negative Binomial regression. In *Proceedings of the International Conference on Machine Learning*, volume 2012, page 1343, 2012.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 11/17; Revised 8/18; Published 10/18

Extrapolating Expected Accuracies for Large Multi-Class Problems

Charles Zheng CHARLES.Y.ZHENG@GMAIL.COM
Section on Functional Imaging Methods
National Institute of Mental Health
Bethesda, MD

Rakesh Achanta RAKESHA@STANFORD.EDU
Department of Statistics
Stanford University
Palo Alto, CA

Yuval Benjamini YUVAL.BENJAMINI@MAIL.HUJI.AC.IL
Department of Statistics
The Hebrew University of Jerusalem,
Jerusalem, Israel

Editor: Christoph Lampert

Abstract

The difficulty of multi-class classification generally increases with the number of classes. Using data for a small set of the classes, can we predict how well the classifier scales as the number of classes increases? We propose a framework for studying this question, assuming that classes in both sets are sampled from the same population and that the classifier is based on independently learned scoring functions. Under this framework, we can express the classification accuracy on a set of k classes as the $(k-1)$st moment of a discriminability function; the discriminability function itself does not depend on k. We leverage this result to develop a non-parametric regression estimator for the discriminability function, which can extrapolate accuracy results to larger unobserved sets. We also formalize an alternative approach that extrapolates accuracy separately for each class, and identify tradeoffs between the two methods. We show that both methods can accurately predict classifier performance on label sets up to ten times the size of the original set, both in simulations as well as in realistic face recognition or character recognition tasks.

Keywords: Multi-class problems, face recognition, object recognition, transfer learning, nonparametric models

1. Introduction

Many machine learning tasks are interested in recognizing or identifying an individual instance within a large set of possible candidates. These problems are usually modeled as multi-class classification problems, with a large and possibly complex label set. Leading examples include detecting the speaker from his voice patterns (Togneri and Pullella, 2011), identifying the author from her written text (Stamatatos et al., 2014), or labeling the object category from its image (Duygulu et al., 2002; Deng et al., 2010; Oquab et al., 2014). In all these examples, the algorithm observes an input x, and uses the classifier function h to guess the label y from a large label set $\mathcal{S}$.

There are multiple practical challenges in developing classifiers for large label sets. Collecting high quality training data is perhaps the main obstacle, as the costs scale with the number of classes. It can be more affordable to first collect data for a small set of classes, even if the long-term goal is to generalize to a larger set. Furthermore, classifier development can be accelerated by training first on fewer classes, as each training cycle may require substantially less resources. Indeed, due to interest in how small-set performance generalizes to larger sets, such comparisons can found in the literature (Oquab et al., 2014; Griffin et al., 2007). A natural question is: how does changing the size of the label set affect the classification accuracy?

We consider a pair of classification problems on finite label sets: a source task with label set $\mathcal{S}_{k_1}$ of size k_1, and a target task with a larger label set $\mathcal{S}_{k_2}$ of size $k_2 > k_1$. For each label set $\mathcal{S}_k$, one constructs the classification rule $h^{(k)} : \mathcal{X} \to \mathcal{S}_k$. Supposing that in each task, the test example (X^*, Y^*) has a joint distribution, define the generalization accuracy for label set $\mathcal{S}_k$ as

$$\mathrm{GA}_k = \Pr[h^{(k)}(X^*) = Y^*]. \tag{1}$$

The problem of *performance extrapolation* is the following: using data from only the source task $\mathcal{S}_{k_1}$, predict the accuracy for a target task with a larger unobserved label set $\mathcal{S}_{k_2}$.

A natural use case for performance extrapolation would be in the deployment of a facial recognition system. Suppose a system was developed in the lab on a database of k_1 individuals. Clients would like to deploy this system on a new larger set of k_2 individuals. Performance extrapolation could allow the lab to predict how well the algorithm will perform on the clients' problem, accounting for the difference in label set size.

Extrapolation should be possible when the source and target classifications belong to the same problem domain. In many cases, the set of categories $\mathcal{S}$ is to some degree a random or arbitrary selection out of a larger, perhaps infinite, set of potential categories $\mathcal{Y}$. Yet any specific experiment uses a fixed finite set. For example, categories in the classical Caltech-256 image recognition data set (Griffin et al., 2007) were assembled by aggregating keywords proposed by students and then collecting matching images from the web. The arbitrary nature of the label set is even more apparent in biometric applications (face recognition, authorship, fingerprint identification) where the labels correspond to human individuals (Togneri and Pullella, 2011; Stamatatos et al., 2014). In all these cases, the number of the labels used to define a concrete data set is therefore an experimental choice rather than a property of the domain. Despite the arbitrary nature of these choices, such data sets are viewed as representing the larger problem of recognition within the given domain, in the sense that success on such a data set should inform performance on similar problems.

In this paper, we assume that both $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$ are samples consisting of independent and identically distributed (i.i.d.) labels from a population (or prior distribution) π, which is defined on the label space $\mathcal{Y}$. We have no constraints on the dependence between $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$: for example, $\mathcal{S}_{k_1}$ may be independent of $\mathcal{S}_{k_2}$, or alternatively $\mathcal{S}_{k_1}$ may be a subsample of $\mathcal{S}_{k_2}$ (the latter case is used in our experiments in Section 5). The sampling assumption is an approximate characterization of the label selection process, which is often at least partially manual. Nevertheless, it provides the exact properties we need without having

to derive specialized metrics of how similar $\mathcal{S}_{k_2}$ is to $\mathcal{S}_{k_1}$. This simplifies the theory, and demonstrates that in a very general setting, extrapolation is possible. We also make the assumption that the classifiers train a model independently for each class. This convenient property allows us to characterize the accuracy of the classifier by selectively conditioning on one class at a time.

Since we assume the label set is random, the generalization accuracy of a given classifier becomes a random variable. Performance extrapolation then becomes the problem of estimating the average generalization accuracy AGA_k of an i.i.d. label set $\mathcal{S}_k$ of size k. Roughly speaking, the achievable accuracy of a classification problem depends on how well the labels can be 'separated' based on the training data– that is, how different the empirical distributions of the training data look at the points where new test instances are drawn. The condition of i.i.d. sampling of labels ensures that the separation of labels in a random set $\mathcal{S}_{k_2}$ can be inferred by looking at the empirical separation in $\mathcal{S}_{k_1}$, and therefore that some estimate of the average accuracy on $\mathcal{S}_{k_2}$ can be obtained.

Our paper presents several main contributions related to extrapolation within this framework. First, we present a theoretical formula describing how average accuracy for smaller k is linked to average accuracy for label set of size $K > k$. We show that accuracy at any size depends on a discriminability function D, which is determined by properties of the data distribution and the classifier but does not depend on k. Second, we propose an estimation procedure that allows extrapolation of the observed average accuracy curve from k_1-class data to a larger number of classes, based on the theoretical formula. Under certain conditions, the estimation method has the property of being an unbiased estimator of the average accuracy. Third, we formalize an alternative approach (proposed by Kay et al. (2008)) that extrapolates accuracy separately for each class, and discuss tradeoffs between the two methods.

The paper is organized as follows. In the rest of this section, we discuss related work. The framework of randomized classification is introduced in Section 2, and there we also introduce a toy example which is revisited throughout the paper. Section 3 develops our theory of extrapolation, and in Section 3.3 we suggest an estimation method. We evaluate our method using simulations in Section 4. In Section 5, we demonstrate our method on a facial recognition problem, as well as an optical character recognition problem. In Section 6 we discuss modeling choices and limitations of our theory, as well as potential extensions.

1.1. Related Work

Linking performance between two different but related classification tasks can be considered an instance of transfer learning (Pan and Yang, 2010). Under Pan and Yang's terminology, our setup is an example of multi-task learning, because the source task has labeled data, which is used to predict performance on a target task that also has labeled data. Applied examples of transfer learning from one label set to another include Oquab et al. (2014), Donahue et al. (2014), Sharif Razavian et al. (2014). However, there is little theory for predicting the behavior of the learned classifier on a new label set. Instead, most research of classification for large label sets deal with the computational challenges of jointly optimizing the many parameters required for these models for specific classification algorithms (Crammer and Singer, 2001; Lee et al., 2004; Weston and Watkins, 1999). Gupta et al. (2014) presents a method for estimating the accuracy of a classifier which can be used to improve performance for general classifiers, but doesn't apply for different set sizes.

The theoretical framework we adopt is one where there exists a family of classification problems with increasing number of classes. This framework can be traced back to Shannon (1948), who considered the error rate of a random codebook, which is a special case of randomized classification. More recently, a number of authors have considered the problem of high-dimensional feature selection for multi-class classification with a large number of classes (Pan et al., 2016; Abramovich and Pensky, 2015; Davis et al., 2011). All of these works assume specific distributional models for classification compared to our more general setup. However, we do not deal with the problem of feature selection.

Perhaps the most similar method that deals with extrapolation of classification error to a larger number of classes can be found in Kay et al. (2008). They trained a classifier for identifying the observed stimulus from a functional MRI scan of brain activity, and were interested in its performance on larger stimuli sets. They proposed an extrapolation algorithm, based on per-class kernel density estimation, as a heuristic with little theoretical discussion. In Section 3.5, we formalize their method within our framework, and implement two variations of their algorithm. We present simulation results and theoretical arguments to compare the algorithm to the regression approach we propose.

2. Randomized Classification

The randomized classification model we study has the following features. We assume that there exists an infinite, perhaps continuous, label space $\mathcal{Y}$ and an example space $\mathcal{X} \subseteq \mathbb{R}^p$. In the subsequent theory we assume that $\mathcal{Y}$ is continuous solely for the sake of mathematical convenience, so that we can discuss probability integrals on the space without the use of measure-theoretic notation. However, the theory would also approximately describe the case of $\mathcal{Y}$ is a sufficiently large discrete space, as long as the probability mass of the largest atom is suitably small.

We assume there exists a prior distribution π on the label space $\mathcal{Y}$, and that for each label $y \in \mathcal{Y}$, there exists a distribution of examples F_y. In other words, for an example-label pair (X, Y), the conditional distribution of X given $Y = y$ is given by F_y.

A random classification task can be generated as follows. The label set $\mathcal{S} = \{Y^{(1)}, \ldots, Y^{(k)}\}$ is generated by drawing labels $Y^{(1)}, \ldots, Y^{(k)}$ i.i.d. from π. Here we assume the number of labels k to be deterministic. For each label, we sample a training set and a test set. The test set is obtained by sampling r observations $X_\ell^{(i)}$ i.i.d. from $F_{Y^{(i)}}$ for $\ell = 1, \ldots, r$. For now, we can also assume that the training set is obtained by sampling r_{train} observations $X_{\ell,train}^{(i)}$ i.i.d. from $F_{Y^{(i)}}$ for $\ell = 1, \ldots, r_{train}$ and $i = 1, \ldots, k$. However, later we will relax these assumptions on the sampling of the training sets, so that we can accommodate classes have differing or stochastically determined number of instances, as long as the number of training instances for different labels are conditionally independent.

Recalling the face recognition example, $\mathcal{Y}$ is the space of all people, π is some distribution for sampling over people, X is a photo of a person's face, and F_Y is the conditional distribution of photos for person Y. The goal of classification is to label the photo X with the correct person Y.

We assume that the classifier $h(x)$ works by assigning a score to each label $y^{(i)} \in \mathcal{S}$, then choosing the label with the highest score. That is, there exist real-valued *score functions* $m_{y^{(i)}}(x)$ for each label $y^{(i)} \in \mathcal{S}$. Here we used the lower-case notation $y^{(i)}$ for the labels, treating them as fixed for now. Since the classifier is allowed to depend on the training data, it is convenient to view it (and its associated score functions) as random. We write $H(x)$ when we wish to work with the classifier as a random function, and likewise $M_y(x)$ to denote the score functions whenever they are considered as random. Since the classifier works by choosing the label with the highest score, the classifier is correct for a given test instance x^* with true label y^* whenever $m_{y^*}(x^*) = \max_j m_{y^{(j)}}(x^*)$, assuming that there are no ties.

For a fixed instance of the classification task with labels $\mathcal{S} = \{y^{(i)}\}_{i=1}^k$ and associated score functions $\{m_{y^{(i)}}\}_{i=1}^k$, recall the definition of the k-class generalization error (1). We will use the assumption that the test labels are uniformly distributed[1] over $\mathcal{S}$, which makes $\mathrm{GA}_k(h, \mathcal{S})$ a *balanced accuracy*, which equally weights the accuracies of the individual classes. Assuming that there are no ties, it can be written in terms of score functions as

$$\mathrm{GA}_k(h, \mathcal{S}) = \frac{1}{k} \sum_{i=1}^{k} \Pr[m_{y^{(i)}}(X^{(i)}) = \max_j m_{y^{(j)}}(X^{(i)})],$$

where $X^{(i)} \sim F_{y^{(i)}}$ for $i = 1, \ldots, k$.

However, it is often appropriate to model the labels $\{Y^{(i)}\}_{i=1}^k$ as a random sample from a distribution. Examples for such *randomized classification* problems include face recognition, where faces are drawn from a larger population, and large multi-class problems where only an arbitrary subset of labels have been collected.

Given our assumption that k is fixed, a natural target for prediction extrapolation is the expected value of the generalization accuracy $\mathrm{GA}_k(h, \mathcal{S})$ over the distribution of label sets. We call this the k-class *average generalization accuracy* of the classifier, denoted AGA_k, and formally defined as

$$\begin{aligned}
\mathrm{AGA}_k &= \mathbf{E}[\mathrm{GA}_k(H, \mathcal{S}_k)] \\
&= \frac{1}{k} \sum_{i=1}^{k} \Pr[M_{Y^{(i)}}(X^{(i)}) = \max_j M_{Y^{(j)}}(X^{(i)})] \\
&= \Pr[M_Y(X) > \max_{j=1}^{k-1} M_{Y^{(j)}}(X)]
\end{aligned}$$

where $Y^{(1)}, \ldots, Y^{(k)} \overset{iid}{\sim} \pi$, and where (X, Y) is an independent draw with the same joint distribution as its superscripted counterparts $(X^{(i)}, Y^{(i)})$. The last line follows from noting that all k summands in the previous line are identical, as each $Y^{(i)}$ is drawn from the same distribution π. The definition of average generalization accuracy is illustrated in Figure 1.

Note that in this framework, the role of the training and test sets is different than how they are usually used machine learning. Our goal is to predict the accuracy achieved on another (random) label set. Therefore, both the training and test data may be used

1. In Section 6, we discuss extensions of our framework that can accommodate the case that the test labels are not uniformly drawn from $\mathcal{S}$, i.e. that one has a non-uniform prior distribution over test labels.

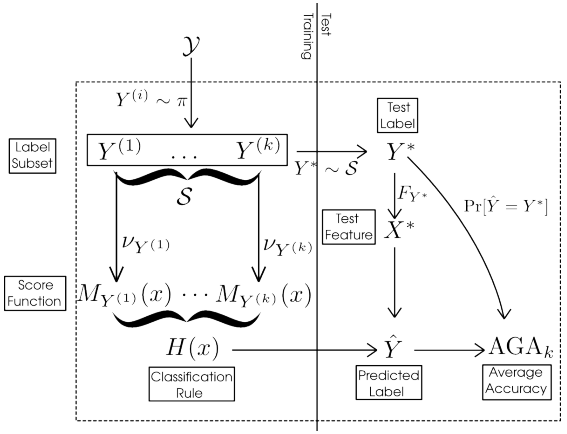


Figure 1: **Average generalization accuracy:** A diagram of the random quantities underlying the average generalization accuracy for k labels (AGA_k). At the training stage (left), a set of k labels $\mathcal{S}$ is sampled from the prior π, and score functions are trained from examples for these classes. At the test stage (right), one true class Y^* is sampled uniformly from $\mathcal{S}$, as well as a test example X^*. AGA_k measures the expected accuracy over these random variables.

in this estimation. Our approach, to be described in Section 2.3, uses the training data exclusively to construct classifiers on label subsets, and test data exclusively to estimate the distribution of favorability over test examples.

2.1. Marginal Classifier

The theoretical analysis of the average generalization accuracy is made much simpler if we can assume that the learning of the scoring functions $M_{Y^{(i)}}$ occurs independently for each labels–that is, there is no information shared between classes. For example, if there exists some function g such that

$$M_{y^{(i)}}(x) = g(x; y^{(i)}, (X^{(i)}_{1,train}, ..., X^{(i)}_{r_{train},train})),$$

the H is a marginal classifier since $M_{y^{(i)}}(x)$ only depends on the label $y^{(i)}$ and the class training set $X^{(i)}_{j,train}$.

In our analysis however, we shall relax the assumption that the classifier $H(x)$ is based on a training set[2]. Instead, it is sufficient that the score functions $\{M_{Y^{(i)}}\}_{i=1}^k$ associated

2. Due to this abstraction, our framework can accommodate scenarios where the classes have differing or stochastically determined number of instances, as long as the number of training instances for different labels are conditionally independent.

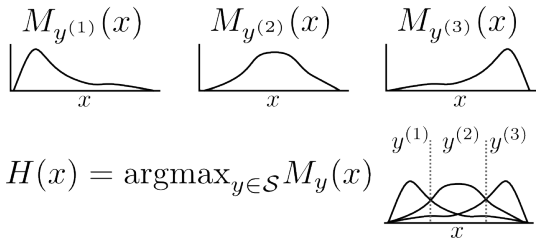


Figure 2: **Classification rule:** Top: Score functions for three classes in a one-dimensional example space. Bottom: The classification rule chooses between $y^{(1)}, y^{(2)}$ or $y^{(3)}$ by choosing the maximal score function.

with the random label set $\{Y^{(i)}\}_{i=1}^{k}$ are independent of the test instances. Under this formalism, we define a marginal classifier as follows.

Definition 1 *The classifier $H(x)$ is called a* marginal classifier *if and only if $M_{Y^{(i)}}$ are independent of both $Y^{(j)}$ and $M_{Y^{(j)}}$ for $j \neq i$.*

In marginal classifiers, classes "compete" only through selecting the highest score, but not in constructing the score functions. Therefore, each M_y can be considered to have been independently drawn from a distribution ν_y. The operation of a marginal classifier is illustrated in Figure 2.

For marginal classifiers, we can prove especially strong results about the accuracy of the classifier under i.i.d. sampling assumptions. And as we will see in the following section, many well-known types of classifiers satisfy the marginal property.

2.2. Examples of Marginal Classifiers

Estimated Bayes classifiers are primary examples of marginal classifiers. By this, we mean classifiers which output the class that maximizes the posterior probability for a class label according to Bayes' rule, but substituting in estimated distributions for the unknown true distributions. Let $\hat{f}_y$ be a density estimate of the example distribution under label y obtained from the empirical distribution $\hat{F}_y$, and let $\pi(y)$ be the prior distribution over labels. Then, we can use the estimated density to produce the score functions:

$$M_y^{EB}(x) = \log(\hat{f}_y(x)) + \log(\pi(y)).$$

The resulting empirical approximation for the Bayes classifier would be

$$H^{EB}(x) = \text{argmax}_{Y \in \mathcal{S}}(M_Y^{EB}(x)).$$

Both Quadratic Discriminant Analysis (QDA) and naïve Bayes classifiers can be seen as specific instances of an estimated Bayes classifier. For QDA, the score function is given by

$$m_y^{QDA}(x) = -(x - \mu(\hat{F}_y))^T \Sigma(\hat{F}_y)^{-1}(x - \mu(\hat{F}_y)) - \log\det(\Sigma(\hat{F}_y)),$$

where $\mu(F) = \int y dF(y)$ and $\Sigma(F) = \int (y - \mu(F))(y - \mu(F))^T dF(y)$. Hence, QDA is the special case of the estimated Bayes classifier when $\hat{f}_y$ is obtained as the multivariate Gaussian density with mean and covariance parameters estimated from the data.

In Naïve Bayes, the score function is

$$m_y^{NB}(x) = \sum_{j=1}^{p} \log \hat{f}_{y,j}(x),$$

where $\hat{f}_{y,j}$ is a density estimate for the j-th component of $\hat{F}_y$. Hence, Naïve Bayes is the estimated Bayes classifier when $\hat{f}_y$ is obtained as the product of estimated componentwise marginal distributions of $p(x_i|y)$.

For some classifiers, M_y is a deterministic function of y (and therefore ν_y is degenerate). A prime example is when there exist fixed or pre-trained embeddings $g, \tilde{g}$ that map both labels y and examples x into R^p. Then

$$M_y^{embed} = -\|g(y) - \tilde{g}(x)\|_2. \tag{2}$$

This would be the case when features from publicly available embeddings (e.g. word embedding vectors) are used for classification; see our example in Section 5. See also Pereira et al. (2018) for an example using word embedding vectors. Note that if the embeddings g or $\tilde{g}$ are informed by the specific set of classes in the experiment, this would no longer be a marginal classifier.

There are many classifiers which do not satisfy the marginal property, such as multinomial logistic regression, multilayer neural networks, decision trees, and k-nearest neighbors.

2.3. Estimation of Average Accuracy

Before tackling extrapolation, it will be useful for us to discuss the simpler task of generalizing accuracy results when the target set is *not* larger than the source set. This allows us to introduce several concepts and notations that are used in the harder problem of generalizing to a larger set. We will also illustrate all of these concept in a toy example following this section, which we shall revisit once more while tackling the problem of extrapolation.

Suppose we have training and test data for a classification task with k_1 classes. That is, we have a label set $\mathcal{S}_{k_1} = \{y^{(i)}\}_{i=1}^{k_1}$, and we assume that the training data has been used to obtain its associated set of score functions $M_{y^{(i)}}$. The test set, composed of $(x_1^{(i)}, \ldots, x_r^{(i)})$ for $i = 1, \ldots, k_1$ is also available to be used for this estimation task. What would be the predicted accuracy for a new randomly sampled set of $k_2 \leq k_1$ labels?

Note that AGA_{k_2} is the expected value of the accuracy on the new set of k_2 labels. Therefore, any unbiased estimator of AGA_{k_2} will be an unbiased predictor for the accuracy on the new set.

Let us start with the case $k_2 = k_1 = k$. For each test observation $x_\ell^{(i)}$, define the ranks of the candidate classes $j = 1, \ldots, k$ by

$$R_\ell^{i,j} = \sum_{s=1}^{k} I\{m_\ell^{(i,j)} \geq m_\ell^{(i,s)}\}.$$

where we have defined $m_\ell^{(i,j)} = m_{y^{(j)}}(x_\ell^{(i)})$. The test accuracy is the fraction of observations for which the correct class also has the highest rank

$$\mathrm{TA}_k = \frac{1}{rk}\sum_{i=1}^{k}\sum_{\ell=1}^{r} I\{R_\ell^{i,i} = k\}. \tag{3}$$

Taking expectations over both the test set and the random labels, the expected value of the test accuracy is AGA_k. Therefore, in this special case, TA_k provides an unbiased estimator for AGA_{k_2}.

Next, let us consider the case where $k_2 < k_1$. Consider label set $\mathcal{S}_{k_2}$ obtained by sampling k_2 labels uniformly without replacement from $\mathcal{S}_{k_1}$. Since $\mathcal{S}_{k_2}$ is unconditionally an i.i.d. sample from the population of labels π, the test accuracy of $\mathcal{S}_{k_2}$ is an unbiased estimator of AGA_{k_2}. However, we can get a better unbiased estimate of AGA_{k_2} by averaging over all the possible subsamples $\mathcal{S}_{k_2} \subset \mathcal{S}_{k_1}$. This defines the average test accuracy over subsampled tasks, ATA_{k_2}.

Remark. Naïvely, computing ATA_{k_2} requires us to train and evaluate $\binom{k_1}{k_2}$ classification rules. However, for marginal classifiers, retraining the classifier is not necessary. The rank $R_\ell^{i,i}$ of the correct label i for $x_\ell^{(i)}$, allows us to determine how many subsets $\mathcal{S}_2$ will result in a correct classification. Specifically, there are $R_\ell^{i,i} - 1 \le k_2$ labels with a lower score than the correct label i. Therefore, as long as one of the classes in $\mathcal{S}_2$ is i, and the other $k_2 - 1$ labels are from the set of $R_\ell^{i,i} - 1$ labels with lower score than i, the classification of $x_j^{(i)}$ will be correct. This implies that there are $\binom{R_\ell^{i,i}-1}{k_2-1}$ such subsets $\mathcal{S}_2$ where $x_\ell^{(i)}$ is classified correctly, and therefore the average test accuracy for all $\binom{k_1}{k_2}$ subsets $\mathcal{S}_2$ is

$$\mathrm{ATA}_{k_2} = \frac{1}{\binom{k_1}{k_2}}\frac{1}{rk_2}\sum_{i=1}^{k_1}\sum_{\ell=1}^{r}\binom{R_\ell^{i,i}-1}{k_2-1}. \tag{4}$$

2.4. Toy Example: Bivariate Normal

Let us illustrate these ideas using a toy example. Let (Y, X) have a bivariate normal joint distribution,

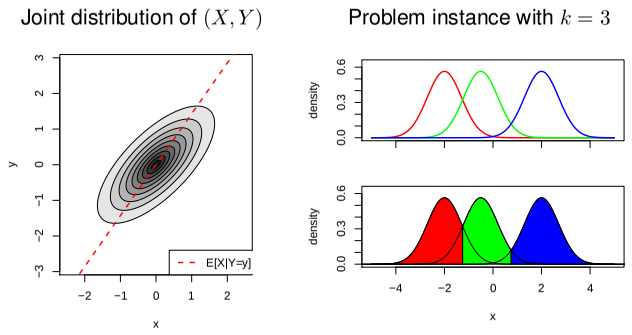
$$(Y, X) \sim N\left(\begin{pmatrix}0\\0\end{pmatrix}, \begin{pmatrix}1 & \rho\\ \rho & 1\end{pmatrix}\right),$$

as illustrated in Figure 3(a). Therefore, for a given randomly drawn label Y, the conditional distribution of X for that label is univariate normal with mean ρY and variance $1 - \rho^2$,

$$X|Y = y \sim N(\rho y, 1 - \rho^2).$$

Supposing we draw $k = 3$ labels $\{y^{(1)}, y^{(2)}, y^{(3)}\}$, the classification problem will be to assign a test instance X^* to the correct label. The test instance X^* would be drawn with equal probability from one of three conditional distributions $X|Y = y^{(i)}$, as illustrated in Figure 3(b, top). The Bayes rule assigns X^* to the class with the highest density $p(x|y^{(i)})$, as illustrated by Figure 3(b, bottom): it is therefore a marginal classifier, with score function

$$M_y(x) = \log(p(x|y)) = -\frac{(x-\rho y)^2}{2(1-\rho^2)} + \text{const.}$$



(a) (b)

Figure 3: **Toy example:** *Left:* The joint distribution of (X, Y) is bivariate normal with correlation $\rho = 0.7$. *Right:* A typical classification problem instance from the bivariate normal model with $k = 3$ classes. *(Top):* the conditional density of X given label Y, for $Y = \{y^{(1)}, y^{(2)}, y^{(3)}\}$. *(Bottom):* the Bayes classification regions for the three classes.

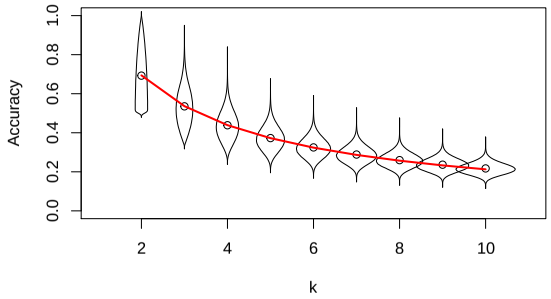


Figure 4: **Generalization accuracy for toy example:** The distribution of the generalization accuracy for $k = 2, 3, \ldots, 10$ for the bivariate normal model with $\rho = 0.7$. Circles indicate the average generalization accuracy AGA_k; the red curve is the theoretically computed average accuracy.

For this model, the generalization accuracy of the Bayes rule for any label set $\{y^{(1)}, \ldots, y^{(k)}\}$ is given by

$$\begin{aligned}\mathrm{GA}_k(h, \{y^{(1)}, \ldots, y^{(k)}\}) &= \frac{1}{k}\sum_{i=1}^{k} \Pr_{X\sim p(x|y^{(i)})}[p(X|y^{(i)}) = \max_{j=1}^{k} p(X|y^{(j)})] \\ &= \frac{1}{k}\sum_{i=1}^{k} \Phi\left(\frac{y^{[i+1]} - y^{[i]}}{2\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{y^{[i-1]} - y^{[i]}}{2\sqrt{1-\rho^2}}\right),\end{aligned}$$

where Φ is the standard normal cdf, $y^{[1]} < \cdots < y^{[k]}$ are the sorted labels, $y^{[0]} = -\infty$ and $y^{[k+1]} = \infty$, and h is the maximum-margin classifier $h(x) = \mathrm{argmax}_{y\in\{y^{(1)},\ldots,y^{(k)}\}} m_y(x)$. We numerically computed $\mathrm{GA}_k(h, \{Y^{(1)}, \ldots, Y^{(k)}\})$ for randomly drawn labels $Y^{(1)}, \ldots, Y^{(k)} \overset{iid}{\sim} N(0,1)$, and the distributions of GA_k for $k = 2, \ldots, 10$ are illustrated in Figure 4. The mean of the distribution of GA_k is the k-class average accuracy, AGA_k. The theory presented in the next section deals with how to analyze the average accuracy AGA_k as a function of k.

3. Extrapolation

The section is organized as follows. We begin by introducing an explicit formula for the average accuracy AGA_k. The formula reveals that AGA_k is determined by moments of a one-dimensional function $D(u)$. Using this formula, we can estimate $D(u)$ using subsampled accuracies. These estimates allow us to extrapolate the average generalization accuracy to an arbitrary number of labels.

The result of our analysis is to expose the average accuracy AGA_k as the weighted average of a function $D(u)$, where $D(u)$ is independent of k, and where k only changes the weighting.

One of the assumptions we rely on is the *tie-breaking condition*, which allows us to neglect specifying the case when margins are tied.

Definition 2 Tie-breaking condition*: for all $x \in \mathcal{X}$, $M_Y(x) \neq M_{Y'}(x)$ with probability one for Y, Y' independently drawn from π.*

In practice, one can simply break ties randomly, which is mathematically equivalent to adding a small amount of random noise ϵ to the function M.

The result is stated as follows.

Theorem 3 *Suppose π, $\{F_y\}_{y\in\mathcal{Y}}$, and score functions M_y satisfy the tie-breaking condition. Then, there exists a cumulative distribution function $D(u)$ defined on the interval $[0,1]$ such that*

$$AGA_k = 1 - (k-1)\int_0^1 D(u)u^{k-2}du. \tag{5}$$

3.1. Analysis of Average Accuracy

Recall that for marginal classifiers, the model M_Y should be independent of the other labels and independent of the test instances. We often consider a random label (Y) with its

associated score function (M_Y) and an example vector (X) drawn from label Y. Explicitly, this sampling can be written:

$$Y \sim \pi, \quad M_Y|Y \sim \nu_Y, \quad X|Y \sim F_Y.$$

Similarly we use $(Y', M_{Y'}, X')$ and (Y^*, M_{Y^*}, X^*) for two more triplets with independent and identical distributions. Specifically, X^* will typically note the test example, and therefore Y^* the true label and M_{Y^*} its score function.

The function D is related to a favorability function. Favorability measures the probability that the score for the example x is going to be maximized by a particular score function m_y, compared to a random competitor $M_{Y'}$. Formally, we write

$$U_x(m_y) = \Pr[m_y(x) > M_{Y'}(x)]. \tag{6}$$

Note that for fixed example x, favorability is monotonically increasing in $m_y(x)$. If $m_y(x) > m_{y^\dagger}(x)$, then $U_x(y) > U_x(y^\dagger)$, because the event $\{m_y(x) > M_{Y'}(x)\}$ contains the event $\{m_{y^\dagger}(x) > M_{Y'}(x)\}$.

Therefore, given labels $y^{(1)}, \ldots, y^{(k)}$ and test instance x, we can think of the classifier as choosing the label with the greatest favorability:

$$\hat{y} = \mathrm{argmax}_{y^{(i)}\in\mathcal{S}} m_{y^{(i)}}(x) = \mathrm{argmax}_{y^{(i)}\in\mathcal{S}} U_x(m_{y^{(i)}}).$$

Furthermore, via a conditioning argument, we see that this is still the case even when the test instance and labels are random, as long as the random example X^* is independent of Y. (Recall that in our notation, X and Y are dependent, but $X^* \perp Y$.)

$$\hat{Y} = \mathrm{argmax}_{Y^{(i)}\in\mathcal{S}} M_{Y^{(i)}}(X^*) = \mathrm{argmax}_{Y^{(i)}\in\mathcal{S}} U_{X^*}(M_{Y^{(i)}}).$$

The favorability takes values between 0 and 1, and when any of its arguments are random, it becomes a random variable with a distribution supported on $[0,1]$. In particular, we consider the following two random variables:

a. the *incorrect-label* favorability $U_{x^*}(M_Y)$ between a given fixed test instance x^*, and the score function of a random incorrect label M_Y, and

b. the *correct-label* favorability $U_{X^*}(M_{Y^*})$ between a random test instance X^*, and the score function of the correct label, M_{Y^*}.

3.1.1. Incorrect-Label Favorability

The incorrect-label favorability can be written explicitly as

$$U_{x^*}(M_Y) = \Pr[M_Y(x^*) > M_{Y'}(x^*)|M_Y(x^*)]. \tag{7}$$

Note that M_Y and $M_{Y'}$ are identically distributed, and both are unrelated to x^* that is fixed. This leads to the following result:

Lemma 4 *Under the tie-breaking condition, the incorrect-label favorability $U_{x^*}(M_Y)$ is uniformly distributed for any $x^* \in \mathcal{X}$, meaning*

$$\Pr[U_{x^*}(M_Y) \leq u] = u \tag{8}$$

for all $u \in [0,1]$.

Proof Write $U_{x^*}(M_Y) = \Pr[Z > Z'|Z]$, where $Z = M_Y(x)$ and $Z' = M_{Y'}(x)$ for $Y, Y' \overset{i.i.d.}{\sim} \pi$. The tie-breaking condition implies that $\Pr[Z = Z'] = 0$. Now observe that for independent random variables Z, Z' with $Z \overset{D}{=} Z'$ and $\Pr[Z = Z'] = 0$, the conditional probability $\Pr[Z > Z'|Z]$ is uniformly distributed. ■

3.1.2. Correct-Label Favorability

The correct-label favorability is

$$U^* = U_{X^*}(M_{Y^*}) = \Pr[M_{Y^*}(X^*) > M_{Y'}(X^*)|Y^*, M_{Y^*}(X^*), X^*]. \tag{9}$$

The distribution of U^* will depend on π, $\{F_y\}_{y\in\mathcal{S}}$ and $\{\nu_y\}_{y\in\mathcal{S}}$, and generally cannot be written in a closed form. However, this distribution is central to our analysis–indeed, we will see that the function D appearing in theorem 3 is defined as the cumulative distribution function of U^*.

The special case of $k = 2$ shows the relation between the distribution of U^* and the average generalization accuracy, AGA_2. In the two-class case, the average generalization accuracy is the probability that a random correct label score function gives a larger value than a random distractor:

$$\mathrm{AGA}_2 = \Pr[M_{Y^*}(X^*) > M_{Y'}(X^*)].$$

where Y^* is the correct label, and Y' is a random incorrect label. If we condition on Y^*, M_{Y^*} and X^*, we get

$$\mathrm{AGA}_2 = \mathbf{E}[\Pr[M_{Y^*}(X^*) > M_{Y'}(X^*)|Y^*, M_{Y^*}, X^*]].$$

Here, the conditional probability inside the expectation is the correct-label favorability. Therefore,

$$\mathrm{AGA}_2 = \mathbf{E}[U^*] = \int D(u)du,$$

where $D(u)$ is the cumulative distribution function of U^*, $D(u) = \Pr[U^* \le u]$. Theorem 3 extends this to general k; we now give the proof.

Proof Without loss of generality, suppose that the true label is Y^* and the incorrect labels are $Y^{(1)}, \ldots, Y^{(k-1)}$. We have

$$\mathrm{AGA}_k = \Pr[M_{Y^*}(X^*) > \max_{i=1}^{k-1} M_{Y^{(i)}}(X^*)] = \Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})],$$

recalling that $U^* = U_{X^*}(M_{Y^*})$. Now, if we condition on $X^* = x^*$, $Y^* = y^*$ and $M_{Y^*} = m_{y^*}$, then the random variable U^* becomes fixed, with value

$$u^* = U_{x^*}(m_{y^*}).$$

Therefore,

$$\begin{aligned}\mathrm{AGA}_k &= \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})|X^* = x^*, Y^* = y^*, M_{Y^*} = m_{y^*}]] \\ &= \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})|X^* = x^*, U^* = u^*]].\end{aligned}$$

Now define $U_{max,k-1} = \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})$. Since by Lemma 4, $U_{X^*}(M_{Y^{(i)}})$ are i.i.d. uniform conditional on $X^* = x^*$, we know that

$$U_{max,k-1}|X^* = x^* \sim \mathrm{Beta}(k-1, 1). \tag{10}$$

Furthermore, $U_{max,k-1}$ is independent of U^* conditional on X^*. Therefore, the conditional probability can be computed as

$$\Pr[U^* > U_{max,k-1}|X^* = x^*, U^* = u^*] = \int_0^{u^*} (k-1)u^{k-2}du.$$

Consequently,

$$\mathrm{AGA}_k = \mathbf{E}[\Pr[U^* > \max_{i=1}^{k-1} U_{X^*}(M_{Y^{(i)}})|X^* = x^*, U^* = u^*]] \tag{11}$$

$$= \mathbf{E}[\int_0^{U^*} (k-1)u^{k-2}du|U^* = u^*] \tag{12}$$

$$= \mathbf{E}[\int_0^1 I\{u \le U^*\}(k-1)u^{k-2}du] \tag{13}$$

$$= (k-1)\int_0^1 \Pr[u \le U^*]u^{k-2}du \tag{14}$$

$$= 1 - (k-1)\int_0^1 \Pr[u \ge U^*]u^{k-2}du. \tag{15}$$

By defining $D(u)$ as the cumulative distribution function of U^* on $[0, 1]$,

$$D(u) = \Pr[U_{X^*}(M_{Y^*}) \le u], \tag{16}$$

and substituting this definition into (15), we obtain the identity (5). ■

Theorem 3 expresses the average accuracy as a weighted integral of the function $D(u)$. Essentially, this theoretical result allows us to reduce the problem of estimating AGA_k to one of estimating $D(u)$. But how shall we estimate $D(u)$ from data? We propose using non-parametric regression for this purpose in Section 3.3.

3.2. Favorability and Average Accuracy for the Toy Example

Recall that for the toy example from Section 2.4, the score function M_y was a non-random function of y that measures the distance between x and ρy

$$M_y(x^*) = \log(p(x^*|y)) = -\frac{(x^* - \rho y)^2}{2(1-\rho^2)}.$$

For this model, the favorability function $U_{x^*}(m_y)$ compares the distance between x^* and ρy to the distance between x^* and $\rho Y'$ for a randomly chosen distractor $Y' \sim N(0,1)$:

$$\begin{aligned} U_{x^*}(m_y) &= \Pr[|\rho y - x^*| < |\rho Y' - x^*|] \\ &= \Phi\left(\frac{x^* + |\rho y - x^*|}{\rho}\right) - \Phi\left(\frac{x^* - |\rho y - x^*|}{\rho}\right), \end{aligned}$$

where Φ is the standard normal cumulative distribution function. Figure 5(a) illustrates the level sets of the function $U_{x^*}(m_y)$. The highest values of $U_{x^*}(m_y)$ are near the line $x^* = \rho y$ corresponding to the conditional mean of $X|Y$, and as one moves farther from the line, $U_{x^*}(m_y)$ decays. Note, however, that large values of x^* and y (with the same sign) result in larger values of $U_{x^*}(m_y)$ since it becomes unlikely for $Y' \sim N(0,1)$ to exceed $Y = y$.

Using the formula above, we can calculate the correct-label favorability $U^* = U_{X^*}(M_{Y^*})$ and its cumulative distribution function $D(u)$. The function D is illustrated in Figure 5(b) for the current example with $\rho = 0.7$. The red curve in Figure 4 was computed using the formula

$$\mathrm{AGA}_k = 1 - (k-1)\int D(u)u^{k-2}du.$$

It is illuminating to consider how the average accuracy curves and the $D(u)$ functions vary as we change the parameter ρ. Higher correlations ρ lead to higher accuracy, as seen in Figure 6(a), where the accuracy curves are shifted upward as ρ increases from 0.3 to 0.9. The favorability $U_{x^*}(m_y)$ tends to be higher on average as well, which leads to lower values of the cumulative distribution function–as we see in Figure 6(b), where the function $D(u)$ decreases as ρ increases, and therefore accuracy increases.

3.3. Estimation

Next, we discuss how to use data from smaller classification tasks to extrapolate average accuracy. We are seeking an unbiased estimator $\widehat{\mathrm{AGA}_k}$ such that

$$\mathbf{E}[\widehat{\mathrm{AGA}_k}] = \mathrm{AGA}_k.$$

Assume that we have data from a k_1-class random classification task, and would like to estimate the average accuracy AGA_{k_2} for $k_2 > k_1$ classes. Our estimation method will use the k-class average test accuracies, $\mathrm{ATA}_2, ..., \mathrm{ATA}_{k_1}$ (see Eq 4), for its inputs.

The key to understanding the behavior of the average accuracy AGA_k is the function D. We adopt a linear model

$$D(u) = \sum_{\ell=1}^{m} \beta_\ell h_\ell(u), \tag{17}$$

where $h_\ell(u)$ are known basis functions, and β_ℓ are the linear coefficients to be estimated. The linearity assumption (17) means that linear regression can be used to estimate the average accuracy curve. This the the idea behind our proposed method ClassExReg, meaning *Classification Extrapolation using Regression.*

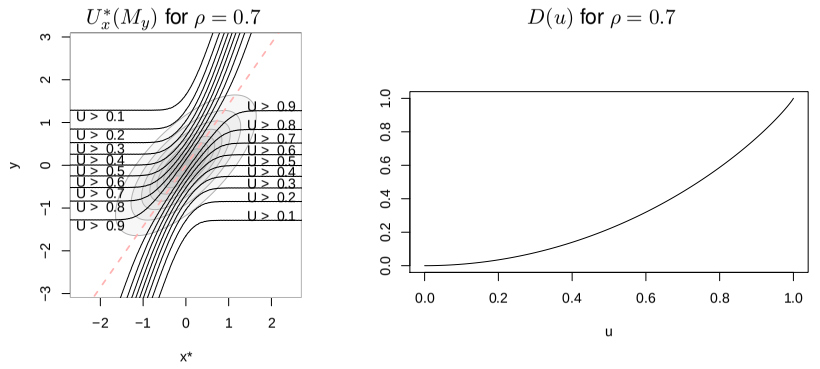


Figure 5: **Favorability for toy example:** *Left:* The level curves of the function $U_{x^*}(M_y)$ in the bivariate normal model with $\rho = 0.7$. *Right:* The function $D(u)$ gives the cumulative distribution function of the random variable $U_{X^*}(M_Y)$.

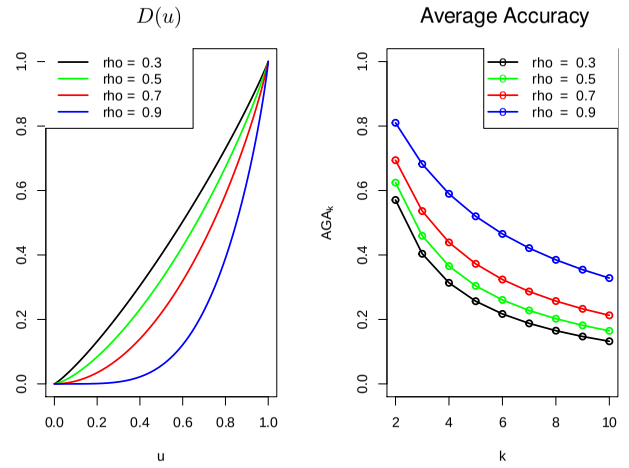


Figure 6: **Average accuracy with different ρ's:** *Left:* The average accuracy AGA_k. *Right:* $D(u)$ function for the bivariate normal model with $\rho \in \{0.3, 0.5, 0.7, 0.9\}$.

Conveniently, AGA_k can also be expressed in terms of the β_ℓ coefficients. If we plug in the assumed linear model (17) into the identity (5), then we get

$$1 - \mathrm{AGA}_k = (k-1)\int D(u)u^{k-2}du \tag{18}$$

$$= (k-1)\int_0^1 \sum_{\ell=1}^{m} \beta_\ell h_\ell(u)u^{k-2}du \tag{19}$$

$$= \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k}, \tag{20}$$

where

$$H_{\ell,k} = (k-1)\int_0^1 h_\ell(u)u^{k-2}du. \tag{21}$$

The constants $H_{\ell,k}$ are moments of the basis function h_ℓ. Note that $H_{\ell,k}$ can be precomputed numerically for any $k \geq 2$.

Now, since the test accuracies ATA_k are unbiased estimates of AGA_k, this implies that the regression estimate

$$\hat{\beta} = \mathrm{argmin}_\beta \sum_{k=2}^{k_1} \left((1 - \mathrm{ATA}_k) - \sum_{\ell=1}^{m} \beta_\ell H_{\ell,k} \right)^2,$$

is unbiased for β. The estimate of AGA_{k_2} is similarly obtained from (20), via

$$\widehat{\mathrm{AGA}}_{k_2} = 1 - \sum_{\ell=1}^{m} \hat{\beta}_\ell H_{\ell,k_2}. \tag{22}$$

3.4. Model Selection

Accurate extrapolation using ClassExReg depends on a good fit between the linear model (17) and the true discriminability function $D(u)$. However, since the function $D(u)$ depends on the unknown joint distribution of the data, it makes sense to let the data help us choose a good basis $\{h_u\}$ from a set of candidate bases.

Let $B_1, \ldots, B_s$ be a set of candidate bases, with $B_i = \{h_u^{(i)}\}_{u=1}^{m_i}$. Ideally, we would like our model selection procedure to choose the B_i that obtains the best root-mean-squared error (RMSE) on the extrapolation from k_1 to k_2 classes. As an approximation, we estimate the RMSE of extrapolation from $\frac{k_1}{2}$ source classes to k_1 target classes, by means of the "bootstrap principle." This amounts to a resampling-based model selection approach, where we perform extrapolations from $k_0 = \lfloor \frac{k_1}{2} \rfloor$ classes to k_1 classes, and evaluate methods based on how closely the predicted $\widehat{\mathrm{AGA}}_{k_1}$ matches the test accuracy ATA_{k_1}. To elaborate, our model selection procedure is as follows.

1. For $\ell = 1, \ldots, L$ resampling steps:
   (a) Subsample $\mathcal{S}_{k_0}^{(\ell)}$ from $\mathcal{S}_{k_1}$ uniformly with replacement.
   (b) Compute average test accuracies $\mathrm{ATA}_2^{(\ell)}, \ldots, \mathrm{ATA}_{k_0}^{(\ell)}$ from the subsample $\mathcal{S}_{k_0}^{(\ell)}$.

   (c) For each candidate basis B_i, with $i = 1, \ldots, s$:
      i. Compute $\hat{\beta}^{(i,\ell)}$ by solving the least-squares problem

$$\hat{\beta}^{(i,\ell)} = \mathrm{argmin}_\beta \sum_{k=2}^{k_0} \left((1 - \mathrm{ATA}_k^{(\ell)}) - \sum_{j=1}^{m_i} \beta_j H_{j,k}^{(i)} \right)^2.$$

      ii. Estimate $\widehat{\mathrm{AGA}}_{k_1}^{(i,\ell)}$ by

$$\widehat{\mathrm{AGA}}_{k_1}^{(i,\ell)} = \sum_{\ell=1}^{m_i} \hat{\beta}_j^{(i,\ell)} H_{j,k_1}^{(i)}.$$

2. Select the basis B_{i^*} by

$$i^* = \mathrm{argmin}_{i=1}^{s} \sum_{\ell=1}^{L} (\widehat{\mathrm{AGA}}_{k_1}^{(i,\ell)} - \mathrm{ATA}_{k_1})^2.$$

3. Use the basis B_{i^*} to extrapolate from k_1 classes (the full data) to k_2 classes.

3.5. The Kay KDE-based estimator

In their paper,[3] Kay et al. (2008) proposed a method for extrapolating classification accuracy to a larger number of classes. The method depends on repeated kernel-density estimation (KDE) steps. Because the method is only briefly motivated in the original text, we present it in our notation.

The idea of the method is to estimate separately for each example $x_\ell^{(i)}$ associated with class $y^{(i)}$ the probability of outscoring a random competitor:

$$\mathrm{Acc}_2(x_\ell^{(i)}; m) := \Pr[m_\ell^{(i,i)} > m_Y(x_\ell^{(i)})],$$

recalling that we defined $m_\ell^{(i,j)} = m_{y^{(j)}}(x_\ell^{(i)})$. For a given example, accurate classification against $k-1$ independent (random) distractor classes is equivalent to $k-1$ independent accurate classifications of a single random competitor. Therefore,

$$\mathrm{Acc}_k(x_\ell^{(i)}; m_{y^{(i)}}) := \Pr[m_\ell^{(i,i)} > \max_{j \neq i} m_\ell^{(i,j)}] = \mathrm{Acc}_2(x_\ell^{(i)}; m_{y^{(i)}})^{k-1}.$$

Given estimated accuracy values $a_\ell^{(i)} = \widehat{\mathrm{Acc}}_2(x_\ell^{(i)})$, $\ell = 1, \ldots, r$, $i = 1, ..., k_1$, we can average the $k-1$ powers:

$$\widehat{\mathrm{AGA}}_k = \frac{1}{rk_1} \sum_{i=1}^{k_1} \sum_{\ell=1}^{r} (1 - (1 - a_\ell^{(i)})^{k-1})$$

Note that if we have noisy but unbiased estimates of $\mathrm{Acc}_2(x_\ell^{(i)}; m_{y^{(i)}})$, then the estimates for $\mathrm{Acc}_k(x_\ell^{(i)}; m_{y^{(i)}})$ and $\widehat{\mathrm{AGA}}_K$ will be upward biased because $\mathbf{E}[X^K] \geq \mathbf{E}[X]^K$.

Accuracy for each example $\mathrm{Acc}_2(x_\ell^{(i)})$ is estimated in two steps:

3. The KDE extrapolation method is described in page 29 of supplement to Kay et al. (2008). While the method is only described for a one-nearest neighbor classifier and for the setting where there is at most one test observation per class, we have taken the liberty of extending it to a generic multi-class classification problem.

1. The density of the wrong-class scores is estimated by smoothing the observed scores with a kernel function $K(\cdot,\cdot)$ and bandwidth h

$$\hat{f}_\ell^{(i)}(m) = \frac{1}{k_1 - 1}\sum_{j\neq i} K_h(m_\ell^{(i,j)}, m).$$

2. The density $\hat{f}_\ell^{(i)}(m)$ is integrated below the observed true value $m_\ell^{(i,i)}$:

$$a_\ell^{(i)} = \widehat{\mathrm{Acc}}_2(x_\ell^{(i)}) = \int_{-\infty}^{m_\ell^{(i,i)}} \hat{f}_\ell^{(i)}(m)dm.$$

The smoothed density usually over-estimates the size of the right-tail of wrong class distribution compared to the observed proportion of errors, biasing downward the accuracy of each individual example.

Briefly, let us point out several key differences between our regression method and Kay's KDE method:

i. The KDE method balances two biases: an upward bias in exponentiating the estimated accuracy, and a downward bias in smoothing the wrong-class densities. The upward bias occurs because even if $\widehat{\mathrm{Acc}}_2(x_\ell^{(i)})$ is unbiased, $\widehat{\mathrm{Acc}}_2(x_\ell^{(i)})^k$ will *not* be unbiased for $\mathrm{Acc}_2(x_\ell^{(i)})^k$, and will typically be larger. The bias becomes more prominent for larger k, but decreases as the true accuracy approaches 1. The result of the KDE method therefore depends non-trivially on the choice of smoothing bandwidth used in the density estimation step. (Without smoothing, however, any example that was correctly estimated in the smaller set would have $\widehat{\mathrm{Acc}}_K = 1$). The method relies on smoothing of each class to generate the tail density that exceeds $m_\ell^{(i,i)}$, and therefore it is highly dependent on the choice of kernel bandwidth.

ii. The KDE method estimates the accuracy separately for every class. When the source-set size k is small, this might lead to less stable estimation for each class and therefore a larger bias after extrapolating. The regression estimator, on the other hand, estimates the average accuracy pooling together information across classes. This might lead to higher variance.

iii. The regression method does not look at the scores directly, only at the rankings. It is therefore blind to monotone transformations on the score functions. The KDE method, on the other hand, is sensitive to the distribution of observed scores.

4. Simulation Study

We ran simulations to check how the proposed extrapolation method, ClassExReg, performs in different settings. The results are displayed in Figure 7. We varied the number of classes k_1 in the source data set, the difficulty of classification, and the basis functions. We generated data according to a mixture of isotropic multivariate Gaussian distributions: labels Y were sampled from $Y \sim N(0, I_{10})$, and the examples for each label sampled from $X|Y \sim N(Y, \sigma^2 I_{10})$. The noise-level parameter σ determines the difficulty of classification.

Similarly to the real-data example, we consider a 1-nearest neighbor classifier, which is given a single training instance per class.

For the estimation, we use the model selection procedure described in Section 3.4 to select the parameter h of the "radial basis"

$$h_\ell(u) = \Phi\left(\frac{\Phi^{-1}(u) - t_\ell}{h}\right).$$

where t_ℓ are a set of regularly spaced knots which are determined by h and the problem parameters. Additionally, we add a constant element to the basis, equivalent to adding an intercept to the linear model (17).

The rationale behind the radial basis is to model the density of $\Phi^{-1}(U^*)$ as a mixture of Gaussian kernels with variance h^2. To control overfitting, the knots are separated by at least a distance of $h/2$, and the largest knots have absolute value $\Phi^{-1}(1 - \frac{1}{rk_1^2})$. The size of the maximum knot is set this way since rk_1^2 is the number of ranks that are calculated and used by our method. Therefore, we do not expect the training data to contain enough information to allow our method to distinguish between more than rk_1^2 possible accuracies, and hence we set the maximum knot to prevent the inclusion of a basis element that has on average a higher mean value than $u = 1 - \frac{1}{rk_1^2}$. However, in simulations we find that the performance of the basis depends only weakly on the exact positioning and maximum size of the knots, as long as sufficiently large knots are included. As is the case throughout non-parametric statistics, the bandwidth h is the most crucial parameter. In the simulation, we use a grid $h = \{0.1, 0.2, \ldots, 1\}$ for bandwidth selection.

Meanwhile, for the KDE method, we used a Gaussian kernel, with the bandwidth chosen via pseudolikelihood cross-validation (Cao et al., 1994), as recommended by Kay et al. (2008). Specifically, we used the two methods for cross-validated KDE estimation provided in the `stats` package in the `R` statistical computing environment: biased cross-validation and unbiased cross-validation (Scott, 1992).

4.1. Simulation Results

We see in Figure 7 that ClassExReg and the KDE methods with unbiased and biased cross-validation (KDE-UCV, KDE-BCV) perform comparably in the Gaussian simulations. We studied how the difficulty of extrapolation relates to both the absolute size of the number of classes and the extrapolation factor $\frac{k_2}{k_1}$. Our simulation has two settings for $k_1 = \{500, 5000\}$, and within each setting we have extrapolations to 2 times, 4 times, 10 times, and 20 times the number of classes.

Within each problem setting defined by the number of source and target classes (k_1, k_2), we use the maximum RMSE across all signal-to-noise settings to quantify the overall performance of the method, as displayed in Table 1.

The results also indicate that more accurate extrapolation appears to be possible for smaller extrapolation ratios $\frac{k_2}{k_1}$ and larger k_1. ClassExReg improves in worst-case RMSE when moving from $k_1 = 500$ to $k_1 = 5000$ while keeping the extrapolation factor fixed, most dramatically in the case $\frac{k_2}{k_1} = 2$ when it improves from a maximum RMSE of 0.032 ± 0.001 ($k_1 = 500$) to 0.009 ± 0.000 ($k_1 = 5000$), which is 3.5-fold reduction in worst-case RMSE,

k_1	k_2	ClassExReg	KDE-BCV	KDE-UCV
500	1000	**0.032** (0.001)	0.090 (0.001)	0.067 (0.001)
500	2000	**0.044** (0.002)	0.088 (0.001)	0.059 (0.001)
500	5000	0.073 (0.004)	0.079 (0.001)	**0.051** (0.001)
500	10000	0.098 (0.004)	0.076 (0.001)	**0.045** (0.001)
5000	10000	**0.009** (0.000)	0.038 (0.000)	0.028 (0.000)
5000	20000	**0.015** (0.001)	0.028 (0.000)	0.019 (0.000)
5000	50000	**0.032** (0.002)	0.035 (0.000)	0.053 (0.000)
5000	100000	**0.054** (0.003)	0.065 (0.000)	0.086 (0.000)

Table 1: Maximum RMSE (se) across all signal-to-noise-levels in predicting TA_{k_2} from k_1 classes in multivariate Gaussian simulation. Standard errors were computed by nesting the maximum operation within the bootstrap, to properly account for the variance of a maximum of estimated means.

but also benefiting from at least a 1.8-fold reduction in RMSE when going from the smaller problem to the larger problem in the other three cases.

The kernel-density method produces comparable results, but is seen to depend strongly on the choice of bandwidth selection: KDE-UCV and KDE-BCV show very different performance profiles, although they differ only in the method used to choose the bandwidth. This matches our analysis in Section 3.5 (item i), where we noted the sensitivity of the tail densities to the bandwidth. Also, the KDE methods show significant estimation bias, as can be seen from Figure 8. As we discussed in 3.5 (item i), this is due to the fact that the KDE method ignores the bias introduced by exponentiation. Meanwhile, ClassExReg avoids this source of bias by estimating the $(k-1)$st moment of $D(u)$ directly. As we see in Figure 8, correcting for the bias of exponentiation helps greatly to reduce the overall bias. Indeed, while ClassExReg shows comparable bias for the 500 to 10000 extrapolation, the bias is very well-controlled in all of the $k_1 = 5000$ extrapolations.

5. Experimental Evaluation

We demonstrate the extrapolation of average accuracy in two data examples: (i) predicting the accuracy of a face recognition on a large set of labels from the system's accuracy on a smaller subset, and (ii) extrapolating the performance of various classifiers on an optical character recognition (OCR) problem in the Telugu script, which has over 400 glyphs.

The face-recognition example takes data from the "Labeled Faces in the Wild" data set (Huang et al. (2007)), where we selected the 1672 individuals with at least 2 face photos. We form a data set consisting of photo-label pairs $(x_j^{(i)}, y^{(i)})$ for $i = 1, \ldots, 1672$ and $j = 1, 2$ by randomly selecting 2 face photos for each individual. We used the OpenFace (Amos et al. (2016)) embedding for feature extraction.[4] In order to identify a new photo x^*, we obtain the feature vector $g(x^*)$ from the OpenFace network, and guess the label $\hat{y}$ with the

4. For each photo x, a 128-dimensional feature vector $g(x)$ is obtained as follows. The computer vision library DLLib is used to detect landmarks in x, and to apply a nonlinear transformation to align x to a template. The aligned photograph is then downsampled to a 96×96 image. The downsampled image

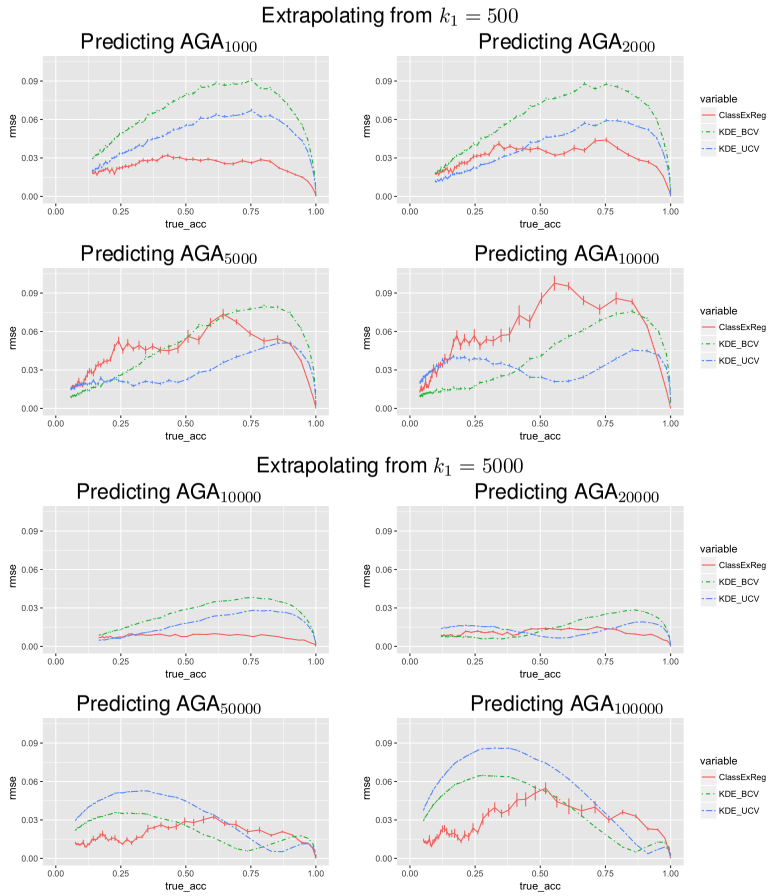


Figure 7: **Simulation results (RMSE):** Simulation study consisting of multivariate Gaussian Y with nearest neighbor classifier. Prediction RMSE vs true k_2-class accuracy for ClassExReg with radial basis (ClassExReg), KDE-based methods with biased cross-validation (KDE_BCV) and unbiased cross-validation (KDE_UCV).

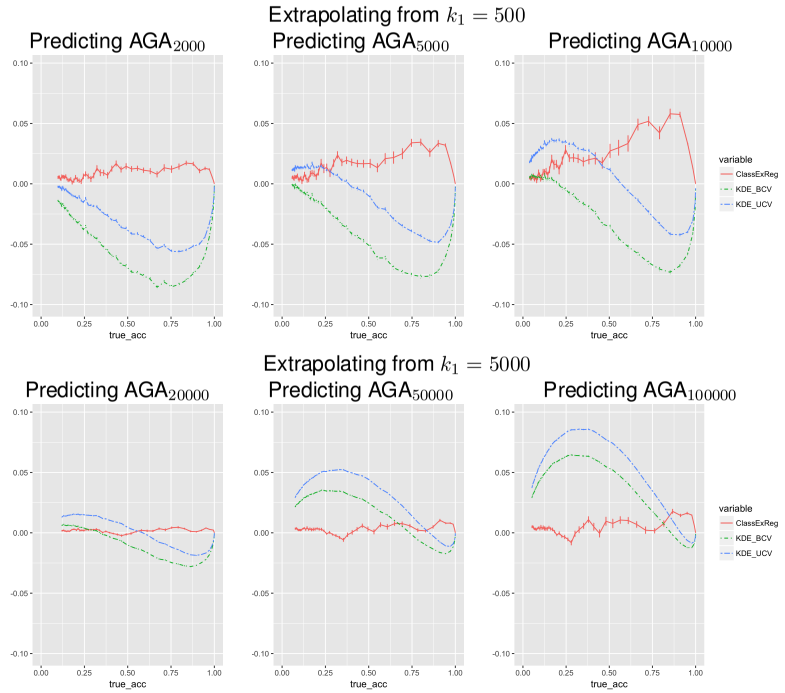


Figure 8: **Simulation results (biases):** Simulation study consisting of multivariate Gaussian Y with nearest neighbor classifier. Bias (mean predicted minus true accuracy) vs true k_2-class accuracy for ClassExReg with radial basis (ClassExReg), KDE-based methods with biased cross-validation (KDE_BCV) and unbiased cross-validation (KDE_UCV).

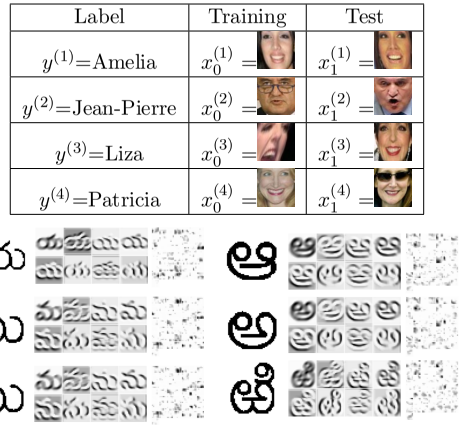


Figure 9: **Face recognition setup (top):** Examples of labels and features from the *Labeled Faces in the Wild* data set. **Telugu OCR (bottom)**: exemplars from six of the glyph classes, along with intermediate features and final transformations from the deep convolutional network.

minimal Euclidean distance between $g(y^{(i)})$ and $g(x^*)$, which implies a score function

$$M_{y^{(i)}}(x^*) = -||g(x_1^{(i)}) - g(x^*)||^2.$$

In this way, we can compute the test accuracy on all 1672 classes, TA$_{1672}$, but we also subsample $k_1 = \{100, 200, 400\}$ classes in order to extrapolate from k_1 to 1672 classes.

In the Telugu optical character recognition example (Achanta and Hastie (2015)), we consider the use of three different classifiers: logistic regression, linear support-vector machine (SVM), and a deep convolutional neural network.[5] The full data consists of 400 classes with 50 training and 50 test observations for each class. We create a nested hierarchy of subsampled data sets consisting of (i) a subset of 100 classes uniformly sampled without replacement from the 400 classes, and (ii) a subset consisting of 20 classes uniformly sampled without replacement from the size-100 subsample. We therefore study three different prediction extrapolation problems:

1. Predicting the accuracy on $k_2 = 100$ classes from $k_1 = 20$ classes, comparing the predicted accuracy to the test accuracy of the classifier on the 100-class subsample as ground truth.

is fed into a pre-trained deep convolutional neural network to obtain the 128-dimensional feature vector $g(x)$. More details are found in Amos et al. (2016).

5. The network architecture is as follows: `48x48-4C3-MP2-6C3-8C3-MP2-32C3-50C3-MP2-200C3-SM`.

k_1	ClassExReg	KDE-BCV	KDE-UCV
100	0.113 (0.002)	**0.053** (0.001)	0.082 (0.001)
200	0.058 (0.002)	**0.037** (0.001)	0.057 (0.001)
400	0.050 (0.001)	**0.024** (0.001)	0.035 (0.001)

Table 2: **Face-recognition extrapolation RMSEs**: RMSE (se) on predicting TA_{1672} from k_1 classes

2. Same as (1), but setting $k_2 = 400$ and $k_1 = 20$, and using the full data set for the ground truth.

3. Same as (2), but setting $k_2 = 400$ and $k_1 = 100$.

Note that unlike in the case of the face recognition example, here the assumption of marginal classification is satisfied for none of the classifiers. We compare the result of our model to the ground truth obtained by using the full data set.

5.1. Results

The extrapolation results for the face recognition problem can be seen in Figure 10, which plots the extrapolated accuracy curves for each method for 100 different subsamples of size k_1. As can be seen, for all three methods, the variances decrease rapidly as k_1 increases.

The root-mean-square errors at $k_2 = 1672$ can be seen in Table 2. KDE-BCV achieves the best extrapolation for all three cases $k_1 = \{100, 200, 400\}$ with KDE-UCV consistently achieving second place. These results differ from the ranking of the RMSEs for the analagous simulation when predicting $k_2 = 2000$ from $k_1 = 500$ for accuracies around 0.45: in the first row and second column of Figure 7, where the true accuracy is 0.43 (from setting $\sigma^2 = 0.2$), the lowest RMSE belongs to KDE-UCV (RMSE=0.0361 ± 0.001), followed closely by ClassExReg (RMSE=0.0372 ± 0.002), and KDE-BCV (RMSE=0.0635 ± 0.001) having the highest RMSE. These discrepancies could be explained by differences between the data distributions between the simulation and the face recognition example, and also by the fact that we only have access to the $k_2 = 1672$-class ground truth for the real data example.

The results for Telugu OCR classification are displayed in Table 3. If we rank the three extrapolation methods in terms of distance to the ground truth accuracy, we see a consistent pattern of rankings between the 20-to-100 extrapolation and the 100-to-400 extrapolation. As we remarked in the simulation, the difficulty of extrapolation appears to be primarily sensitive to the extrapolation ratio $\frac{k_2}{k_1}$, which are similar (5 versus 4) in the 20-to-100 and 100-to-400 problems. In both settings, ClassExReg comes closest to the ground truth for the Deep CNN and the SVM, but KDE-BCV comes closest to ground truth for the Logistic regression. However, even for logistic regression, ClassExReg does better or comparably to KDE-UCV.

In the 20-to-400 extrapolation, which has the highest extrapolation ratio ($\frac{k_2}{k_1} = 20$), none of the three extrapolation methods performs consistently well for all three classifiers. It could be the case that the variability is a dominating effect given the small training

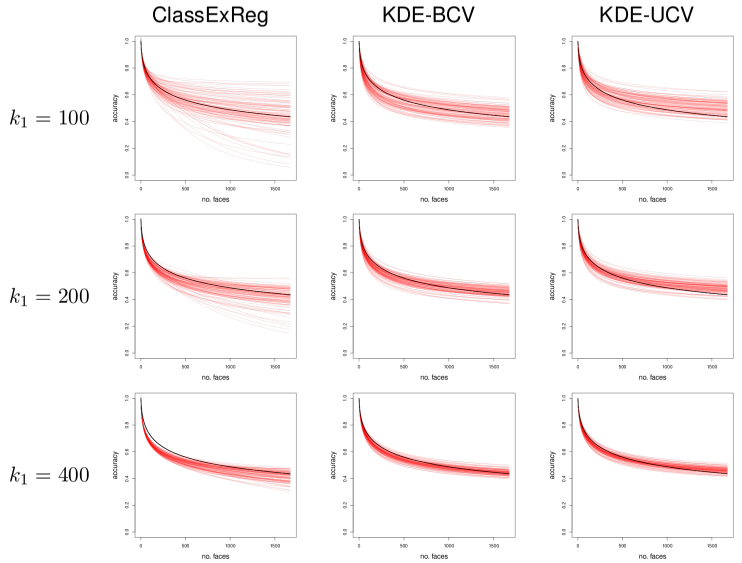


Figure 10: **Predicted accuracy curves for face-recognition example**: The plots show predicted accuracies. Each red curve represents the predicted accuracies using a single subsample of size k_1. The black curve shows the average test accuracy obtained from the full data set.

set, making it difficult to compare extrapolation methods using the 20-to-400 extrapolation task.

Unlike in the face recognition example, we did not resample training classes here, because that would require retraining all of the classifiers–which would be prohibitively time-consuming for the Deep CNN. Thus, we cannot comment on the robustness of the comparisons from this example, though it is likely that we would obtain different rankings under a new resampling of the training classes.

These empirical results indicate that in comparison to the KDE methods of Kay et al., ClassExReg has lower bias and better performance at small k (up to 500 classes), while the KDE methods start to achieve comparable results to ClassExReg when k is around 5,000 or more classes. This matches the theoretical intuition we developed in Section 3.5 (item ii): since the Kay et al. method works by estimating each the favorability of each example separately, that it requires more data (meaning more classes in the training set) to achieve robust estimation.

k_1	k_2	Classifier	True	ClassExReg	KDE-BCV	KDE-UCV
20	100	Deep CNN	0.9908	**0.9905**	0.7138	0.6507
		Logistic	0.8490	0.8980	**0.8414**	0.8161
		SVM	0.7582	**0.8192**	0.6544	0.5771
20	400	Deep CNN	0.9860	**0.9614**	0.4903	0.3863
		Logistic	0.7107	0.8824	0.7467	**0.7015**
		SVM	0.5452	0.6725	**0.5163**	0.4070
100	400	Deep CNN	0.9860	**0.9837**	0.8910	0.8625
		Logistic	0.7107	0.7214	**0.7089**	0.6776
		SVM	0.5452	**0.5969**	0.4369	0.3528

Table 3: **Telugu OCR extrapolated accuracies**: Extrapolating from k_1 to k_2 classes in Telugu OCR for three different classifiers: logistic regression, support vector machine, and deep convolutional network

6. Discussion

In this work, we suggest treating the class set in a classification task as random, in order to extrapolate classification performance on a small task to the expected performance on a larger unobserved task. We show that average generalized accuracy decreases with increased label set size like the $(k-1)$th moment of a distribution function. Furthermore, we introduce an algorithm for estimating this underlying distribution, that allows efficient computation of higher order moments. We additionally implement a kernel-density estimation based extrapolation, and discuss different regimes were the methods are useful. Code for the methods and the simulations can be found in `https://github.com/snarles/ClassEx`.

There are many choices and simplifying assumptions used in the description of the method. Here we discuss these decisions and map some alternative models or strategies for future work.

Two important practical aspects of real-world problems not currently handled by our analysis are (i) non-uniform prior distributions on the labels, and (ii) cost functions other than zero-one loss. In fact, a theory for arbitrary cost functions can double as a theory for non-uniform priors, because the risk incurred under non-uniform priors is equivalent to the risk incurred under a uniform prior but with a weighted cost function. Hence, we address both (i) and (ii) in forthcoming work that shows how performance extrapolation is possible under arbitrary cost functions.

Since our analysis is currently restricted to i.i.d. sampling of classes, one direction for future work is to generalize the sampling mechanism, such as to cluster sampling. More broadly, the assumption that the labels in $\mathcal{S}_k$ are a random sample from a homogeneous distribution π may be inappropriate. Many natural classification problems arise from hierarchically partitioning a space of instances into a set of labels. Therefore, rather than modeling $\mathcal{S}_k$ as a random sample, it may be more suitable to model it as a random hierarchical partition of $\mathcal{Y}$, such as one arising from an optional Pólya tree process (Wong and Ma, 2010). Finally, note that we assume no knowledge about the new class-set except for its size. Better accuracy might be achieved if some partial information is known.

A third direction of exploration is to impose additional modeling assumptions for specific problems. ClassExReg adopts a non-parametric model of the discriminability function $D(u)$, in the sense that $D(u)$ was defined via a spline expansion. However, an alternative approach is to assume a parametric family for $D(u)$ defined by a small number of parameters. In forthcoming work, we show that under certain limiting conditions, $D(u)$ is well-described by a two-parameter family. This substantially increases the efficiency of estimation in cases where the limiting conditions are well-approximated.

Acknowledgments

We thank Jonathan Taylor, Trevor Hastie, John Duchi, Steve Mussmann, Qingyun Sun, Robert Tibshirani, Patrick McClure, Francisco Pereira, and Gal Elidan for useful discussion. CZ is supported by an NSF graduate research fellowship, and would also like to thank the European Research Council under the ERC grant agreement n°[PSARPS-294519] for travel support. We would also like thank the anonymous reviewers for their comments, which improved the readability of the paper.

References

Felix Abramovich and Marianna Pensky. Feature selection and classification of high-dimensional normal vectors with possibly large number of classes. *arXiv preprint arXiv:1506.01567*, 2015.

Rakesh Achanta and Trevor Hastie. Telugu OCR framework using deep learning. *arXiv preprint arXiv:1509.05962*, 2015.

Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

Ricardo Cao, Antonio Cuevas, and Wensceslao González Manteiga. A comparative study of several smoothing methods in density estimation. *Computational Statistics & Data Analysis*, 17(2):153–176, 1994.

Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2(Dec):265–292, 2001.

Justin Davis, Marianna Pensky, and William Crampton. Bayesian feature selection for classification with possibly large number of classes. *Journal of Statistical Planning and Inference*, 141(9):3256–3266, 2011.

Jia Deng, Alexander C Berg, Kai Li, and Li Fei-Fei. What does classifying more than 10,000 image categories tell us? In *European conference on computer vision*, pages 71–84. Springer, 2010.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pages 647–655, 2014.

Pinar Duygulu, Kobus Barnard, Joao FG de Freitas, and David A Forsyth. Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In *European conference on computer vision*, pages 97–112. Springer, 2002.

Gregory Griffin, Alex Holub, and Pietro Perona. Caltech-256 object category dataset. 2007.

Maya R Gupta, Samy Bengio, and Jason Weston. Training highly multiclass classifiers. *Journal of Machine Learning Research*, 15(1):1461–1492, 2014.

Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(March):352–355, 2008.

Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1717–1724, 2014.

Rui Pan, Hansheng Wang, and Runze Li. Ultrahigh-dimensional multiclass linear discriminant analysis by pairwise sure independence screening. *Journal of the American Statistical Association*, 111(513):169–179, 2016.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2010.

Francisco Pereira, Bin Lou, Brianna Pritchett, Samuel Ritter, Samuel J Gershman, Nancy Kanwisher, Matthew Botvinick, and Evelina Fedorenko. Toward a universal decoder of linguistic meaning from brain activation. *Nature communications*, 9(1):963, 2018.

David W. Scott. *Multivariate Density Estimation: Theory, Practice, and Visualization*. Wiley, 1992.

Claude E. Shannon. A Mathematical Theory of Communication. *Bell System Technical Journal*, 27(3):379–423, 1948.

Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. CNN features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

Efstathios Stamatatos, Walter Daelemans, Ben Verhoeven, Patrick Juola, Aurelio López-López, Martin Potthast, and Benno Stein. Overview of the author identification task at PAN 2014. In *CLEF (Working Notes)*, pages 877–897, 2014.

Roberto Togneri and Daniel Pullella. An overview of speaker identification: Accuracy and robustness issues. *IEEE Circuits and Systems Magazine*, 11(2):23–61, 2011.

Jason Weston and Chris Watkins. Support vector machines for multi-class pattern recognition. In *ESANN*, volume 99, pages 219–224, 1999.

Wing H Wong and Li Ma. Optional Pólya tree and Bayesian inference. *The Annals of Statistics*, 38(3):1433–1459, 2010.

Journal of Machine Learning Research 19 (2018) 1-71 Submitted 12/17; Revised 7/18; Published 7/18

Inference via Low-Dimensional Couplings

Alessio Spantini SPANTINI@MIT.EDU
Daniele Bigoni DABI@MIT.EDU
Youssef Marzouk YMARZ@MIT.EDU
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Editor: Manfred Opper

Abstract

We investigate the low-dimensional structure of deterministic transformations between random variables, i.e., transport maps between probability measures. In the context of statistics and machine learning, these transformations can be used to couple a tractable "reference" measure (e.g., a standard Gaussian) with a target measure of interest. Direct simulation from the desired measure can then be achieved by pushing forward reference samples through the map. Yet characterizing such a map—e.g., representing and evaluating it—grows challenging in high dimensions. The central contribution of this paper is to establish a link between the Markov properties of the target measure and the existence of low-dimensional couplings, induced by transport maps that are *sparse* and/or *decomposable*. Our analysis not only facilitates the construction of transformations in high-dimensional settings, but also suggests new inference methodologies for continuous non-Gaussian graphical models. For instance, in the context of nonlinear state-space models, we describe new variational algorithms for filtering, smoothing, and sequential parameter inference. These algorithms can be understood as the natural generalization—to the non-Gaussian case—of the square-root Rauch–Tung–Striebel Gaussian smoother.

Keywords: transport map, variational inference, graphical models, sparsity, state-space models, joint parameter and state estimation

1. Introduction

This paper studies the low-dimensional structure of transformations between random variables. Such transformations, which can be understood as transport maps between probability measures, are ubiquitous in statistics and machine learning. They can be used for posterior sampling (Moselhy and Marzouk, 2012), possibly via deep neural networks (Rezende and Mohamed, 2015); for accelerating Markov chain Monte Carlo or importance sampling algorithms (Parno and Marzouk, 2018; Han and Liu, 2017); or as the building blocks of implicit generative models (Kingma and Welling, 2013; Goodfellow et al., 2014) and flexible methods for density estimation (Tabak and Turner, 2013; Dinh et al., 2016).

In the context of variational inference (Blei et al., 2016), a transport map can be used to define a deterministic coupling between a tractable reference measure $\boldsymbol{\nu}_\eta$ that we can easily simulate (e.g., a standard Gaussian) and an arbitrary target measure $\boldsymbol{\nu}_\pi$ that we wish to characterize (e.g., a posterior distribution). Given i.i.d. samples $(\boldsymbol{X}_i)$ from the reference measure, we can evaluate the transport map to obtain i.i.d. samples $(T(\boldsymbol{X}_i))$ from the target. In other words, the map allows any expectation $\int g \, \mathrm{d}\boldsymbol{\nu}_\pi$ over the target measure to be rewritten as an integral over the reference measure,

$$\int g(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{\nu}_\pi(\boldsymbol{x}) = \int g(T(\boldsymbol{x})) \, \mathrm{d}\boldsymbol{\nu}_\eta(\boldsymbol{x}) \,,$$

thus enabling the use of standard integration techniques for the tractable $\boldsymbol{\nu}_\eta$, including Monte Carlo sampling (Meng and Schilling, 2002) and deterministic quadratures.

We focus on absolutely continuous measures $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ on $\mathbb{R}^n$, for which the existence of a transport map $T : \mathbb{R}^n \to \mathbb{R}^n$ is guaranteed (Santambrogio, 2015). Such a map, however, is seldom unique. Identifying a particular map requires imposing additional structure on the problem. Optimal transport maps, for instance, define couplings that minimize a particular integrated *transport cost* expressing the effort required to rearrange samples (Villani, 2008). The analysis of such maps underpins a vast field that links geometry and partial differential equations, with applications in fluid dynamics, economics, statistics (Douglas, 1999; Kantorovich, 1965), and beyond. In recent years, several other couplings have been proposed for use in statistical problems, e.g., parametric approximations of the Knothe–Rosenblatt rearrangement (Moselhy and Marzouk, 2012), couplings induced by the flows of ODEs (Anderes and Coram, 2012; Heng et al., 2015), and couplings induced by the composition of many simple maps, including deep neural networks (Rezende and Mohamed, 2015; Liu and Wang, 2016). Yet the construction, representation, and evaluation of all these maps grows challenging in high dimensions. In the setting considered here, a transport map is a function from $\mathbb{R}^n$ onto itself; without specifying further structure, representing such a map or even realizing its action is often intractable as n increases.

The central contribution of this paper is to establish a link between the conditional independence structure of the reference-target pair—the so-called Markov properties (Lauritzen, 1996) of $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$—and the existence of low-dimensional couplings. These couplings are induced by transport maps that are *sparse* and/or *decomposable*. A sparse map consists of scalar-valued component functions that each depend only on a few input variables, whereas a decomposable map factorizes as the *exact* composition of finitely many functions of low effective dimension (i.e., $T = T_1 \circ \cdots \circ T_\ell$, where each T_i differs from the identity map only along a subset of its components). These properties, and their combinations, dramatically reduce the complexity of representing a transport map and can be deduced *before* the map is explicitly computed.

The utility of these results is twofold. First, they make the construction of couplings—and hence the characterization of complex probability distributions—tractable for a large class of inference problems. In particular, these results can be exploited in state-of-the-art approaches for the numerical computation of transport maps, including normalizing flows or Stein variational algorithms (Rezende and Mohamed, 2015; Detommaso et al., 2018). Second, these results suggest new algorithmic approaches for important classes of statistical models. For instance, our analysis of sparse triangular maps provides a general framework for describing continuous and non-Gaussian Markov random fields, and for exploiting the conditional independence structure of these fields in computation. Our analysis of decomposable transport maps yields new variational algorithms for sequential inference in nonlinear and non-Gaussian state space models. These algorithms characterize the full Bayesian solution to the smoothing and joint state–parameter inference problems by means

of a decomposable transport map, which is constructed (recursively) in a *single* forward pass using local operations. These algorithms can be understood as the natural generalization, to the non-Gaussian case, of the square-root Rauch-Tung-Striebel Gaussian smoother. Moreover, the results presented in this paper underpin recent efforts in structure learning for non-Gaussian graphical models (Morrison et al., 2017), and novel approaches to the filtering of high-dimensional spatiotemporal processes (Spantini, 2017, Ch. 6). Overall, we propose a range of techniques to address problems of inference in continuous non-Gaussian graphical models.

The paper is organized as follows. Section 2 introduces some notation used throughout the paper. Section 3 reviews the Knothe-Rosenblatt rearrangement, a key coupling for our analysis, while Section 4 briefly recalls some standard terminology for Markov random fields and graphical models. The main results are in Sections 5–7: Section 5 addresses the sparsity of triangular transports, while Section 6 introduces and develops the concept of decomposable transport maps for general Markov networks. These two sections can be read independently. Section 7 specializes the theory of Section 6 to state-space models, introducing new variational algorithms for filtering, smoothing, and parameter inference. Section 8 illustrates aspects of the theory with numerical examples. A final discussion is presented in Section 9. Appendix A collects some technical details on the Knothe-Rosenblatt rearrangement and its generalizations. Appendix B contains the proofs of the main results. Appendix C provides pseudocode for our variational algorithms applied to state-space models, and additional numerical experiments are described in Appendix D. Code and all numerical examples are available online.[1]

2. Notation

Here, we collect some useful notation used throughout the paper.

Notation for functions, sets, and graphs. For a pair of functions f and g, we denote their composition by $f \circ g$. We denote by $\partial_k f$ the partial derivative of f with respect to its kth input variable. By $\partial_k f = 0$, we mean that the function f does not depend on its kth input variable. Depending on the context, we can identify a matrix Q with its corresponding linear map, given by $\boldsymbol{x} \mapsto Q\boldsymbol{x}$.

For all $n > 0$, we let $\mathbb{N}_n = \{1, \dots, n\}$ denote the set of the first n integers. For any pair of sets, $\mathcal{A} \subset \mathcal{B}$ means that $\mathcal{A}$ is a subset of $\mathcal{B}$ (including the possibility of $\mathcal{A} = \mathcal{B}$). We denote by $|\mathcal{A}|$ the cardinality of $\mathcal{A}$.

Given a graph $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$ with vertices $\mathcal{V}$ and edges $\mathcal{E}$, we denote by $\mathrm{Nb}(k, \boldsymbol{\mathcal{G}})$ the neighborhood of a node k in $\boldsymbol{\mathcal{G}}$, while for any set $\mathcal{A} \subset \mathcal{V}$, we denote by $\boldsymbol{\mathcal{G}}_{\mathcal{A}} = (\mathcal{V}', \mathcal{E}')$ the subgraph given by $\mathcal{V}' = \mathcal{A}$ and $\mathcal{E}' = \mathcal{E} \cap (\mathcal{A} \times \mathcal{A})$.

Notation for measures and densities. In this paper, we mostly consider probability measures on $\mathbb{R}^n$ that are absolutely continuous with respect to the Lebesgue measure, $\boldsymbol{\lambda}$, and that are fully supported. We denote the set of such measures by $\mathscr{M}_+(\mathbb{R}^n)$. The *density* of a measure will always be intended with respect to $\boldsymbol{\lambda}$. For a pair of measures $\boldsymbol{\nu}_1, \boldsymbol{\nu}_2$, $\boldsymbol{\nu}_1 \ll \boldsymbol{\nu}_2$ means that $\boldsymbol{\nu}_1$ is absolutely continuous with respect to $\boldsymbol{\nu}_2$.

For any measure $\boldsymbol{\nu}$ and measurable map T, we denote by $T_\sharp \boldsymbol{\nu}$ the pushforward measure given by $\boldsymbol{\nu} \circ T^{-1}$, where for any set $\mathcal{B}$, $T^{-1}(\mathcal{B})$ is the set-valued preimage of $\mathcal{B}$ under T.

1. `http://transportmaps.mit.edu`

Similarly, we denote by $T^\sharp \boldsymbol{\nu}$ the pullback measure given by $\boldsymbol{\nu} \circ T$. Given a measure $\boldsymbol{\nu}$ with density π and a map T, we denote by $T_\sharp \pi$ the density of $T_\sharp \boldsymbol{\nu}$, provided it exists (depending on T). We call $T_\sharp \pi$ the *pushforward density* of π by T. Similarly, we define the pullback density $T^\sharp \pi$ as the density of $T^\sharp \boldsymbol{\nu}$, provided it exists. Whether the map T preserves the absolute continuity of the measure depends on the regularity of T. For instance, if $T : \mathbb{R}^n \to \mathbb{R}^n$ is a diffeomorphism—i.e., a differentiable bijection with differentiable inverse—then one has:

$$T_\sharp \pi(\boldsymbol{x}) = \pi(T^{-1}(\boldsymbol{x})) \, |\det \nabla T^{-1}(\boldsymbol{x})|, \qquad T^\sharp \pi(\boldsymbol{x}) = \pi(T(\boldsymbol{x})) \, |\det \nabla T(\boldsymbol{x})|, \tag{1}$$

where $\nabla T(\boldsymbol{x})$ denotes the Jacobian of T at $\boldsymbol{x}$. The regularity assumptions on T can be substantially weakened as long as one modifies (1) appropriately (Fremlin, 2000). We will give one such example shortly when dealing with triangular maps (see Section 3 or Appendix A). We denote by $\int f(\boldsymbol{x}) \, \boldsymbol{\nu}(\mathrm{d}\boldsymbol{x})$ the integration of a measurable function $f : \mathbb{R}^n \to \mathbb{R}$ with respect to a measure $\boldsymbol{\nu}$. For the Lebesgue measure, we simplify our notation as $\int f(\boldsymbol{x}) \, \boldsymbol{\lambda}(\mathrm{d}\boldsymbol{x}) = \int f(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$. Given a pair η, π of probability densities and a map $T : \mathbb{R}^n \to \mathbb{R}^n$, we say that T *pushes forward* η to π if and only if T couples the corresponding probability measures, i.e., $T_\sharp \boldsymbol{\nu}_\eta = \boldsymbol{\nu}_\pi$, with $\boldsymbol{\nu}_\eta(\mathcal{B}) = \int_{\mathcal{B}} \eta(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$ and $\boldsymbol{\nu}_\pi(\mathcal{B}) = \int_{\mathcal{B}} \pi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}$ for all measurable sets $\mathcal{B}$. (Notice that $T_\sharp \eta$ need not be given by (1) since we are not specifying any regularity on T.)

When it is clear from context, we will freely omit the qualifier a.e. to indicate a property that holds up to a set of measure zero.

Notation for random variables. We use boldface capital letters, e.g., $\boldsymbol{X}$, to denote random variables on $\mathbb{R}^n$ with $n > 1$, while we write scalar-valued random variables as X. The law of a random variable $\boldsymbol{X}$ defined on a probability space $(\Omega, \mathbb{P})$ is given by $\boldsymbol{X}_\sharp \mathbb{P}$. For a measure $\boldsymbol{\nu}$, $\boldsymbol{X} \sim \boldsymbol{\nu}$ means that $\boldsymbol{X}$ has law $\boldsymbol{\nu}$. If $\boldsymbol{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_p)$ is a collection of random variables and $\mathcal{A} \subset \mathbb{N}_p$, then $\boldsymbol{X}_{\mathcal{A}} = (\boldsymbol{X}_i, i \in \mathcal{A})$ denotes a subcollection of $\boldsymbol{X}$. In the same way, for $j < k$, $\boldsymbol{X}_{j:k} = (\boldsymbol{X}_j, \boldsymbol{X}_{j+1}, \dots, \boldsymbol{X}_k)$. If $\boldsymbol{X} = (\boldsymbol{X}_1, \dots, \boldsymbol{X}_p)$ has joint density π and $\mathcal{A} \subset \mathbb{N}_p$, we denote by $\pi_{\boldsymbol{X}_{\mathcal{A}}}$ the marginal of π along $\boldsymbol{X}_{\mathcal{A}}$, i.e., $\pi_{\boldsymbol{X}_{\mathcal{A}}}(\boldsymbol{x}_{\mathcal{A}}) = \int \pi(\boldsymbol{x}) \, \mathrm{d}\boldsymbol{x}_{\mathbb{N}_p \setminus \mathcal{A}}$. If π is the density of $\boldsymbol{Z} = (\boldsymbol{X}, \boldsymbol{Y})$, we denote by $\pi_{\boldsymbol{X}|\boldsymbol{Y}}$ the density of $\boldsymbol{X}$ given $\boldsymbol{Y}$, where

$$\pi_{\boldsymbol{X}|\boldsymbol{Y}}(\boldsymbol{x}|\boldsymbol{y}) = \begin{cases} \pi_{\boldsymbol{X},\boldsymbol{Y}}(\boldsymbol{x}, \boldsymbol{y}) / \pi_{\boldsymbol{Y}}(\boldsymbol{y}) & \text{if } \pi_{\boldsymbol{Y}}(\boldsymbol{y}) \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

We denote independence of a pair of random variables $\boldsymbol{X}, \boldsymbol{Y}$ by $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y}$. In the same way, $\boldsymbol{X} \perp\!\!\!\perp \boldsymbol{Y} | \boldsymbol{R}$ means that $\boldsymbol{X}$ and $\boldsymbol{Y}$ are independent given a third random variable $\boldsymbol{R}$.

3. Triangular Transport Maps: a Building Block

An important transport for our analysis is the Knothe-Rosenblatt (KR) rearrangement on $\mathbb{R}^n$ (Rosenblatt, 1952). For a pair of measures $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$, with densities η and π, respectively, the KR rearrangement is the unique monotone increasing lower triangular measurable map that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, i.e., $T_\sharp \boldsymbol{\nu}_\eta = \boldsymbol{\nu}_\pi$ (Carlier et al., 2010). Here, monotonicity is with respect to the lexicographic order on $\mathbb{R}^n$, while uniqueness is up to $\boldsymbol{\nu}_\eta$-null sets. A lower triangular map $T : \mathbb{R}^n \to \mathbb{R}^n$ is a multivariate function whose kth

component depends only on the first k input variables, i.e.,

$$T(\boldsymbol{x}) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_2) \\ \vdots \\ T^n(x_1, x_2, \dots x_n) \end{bmatrix}$$

for some collection of functions (T^k) and for all $\boldsymbol{x} = (x_1, \dots, x_n)$.

The distinction between lower, upper, or other more general forms of triangular map is a matter of convention. We will revisit this important point in Section 6. See Appendix A for a constructive definition of the KR rearrangement based on a sequence of one-dimensional transports. In our hypothesis, the KR rearrangement is always a bijection on $\mathbb{R}^n$, while each map

$$\xi \mapsto T^k(x_1, \dots, x_{k-1}, \xi) \tag{3}$$

is homeomorphic (continuous bijection with continuous inverse), strictly increasing, and differentiable a.e. (Santambrogio, 2015). Here, monotonicity with respect to the lexicographic order is equivalent to each function (3) being increasing. The resulting rearrangement T is far from being a diffeomorphism but is still regular enough to define a useful change of variables, as the following lemma proven in Bogachev et al. (2005) shows.

Lemma 1 *If T is a KR rearrangement pushing forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, then $\boldsymbol{\nu}_\eta$-a.e.,*

$$T^\sharp \pi(\boldsymbol{x}) = \pi(T(\boldsymbol{x})) \det \nabla T(\boldsymbol{x}) = \eta(\boldsymbol{x}), \tag{4}$$

where $\det \nabla T \coloneqq \prod_{i=1}^n \partial_k T^k$ exists a.e., and where $T^\sharp \pi$ is the density of $T^\sharp \boldsymbol{\nu}_\pi$.

In general, $\det \nabla T$ in (4) is not the determinant of the Jacobian of T since the map may not be differentiable, in which case it would not be possible to define ∇T in the classical sense; this is why $\det \nabla T$ is *redefined* in the lemma. Nevertheless, it is known that T inherits the same regularity as η and π, but not more (Santambrogio, 2015). See Appendix A for additional remarks on the regularity of the map.

An essential feature of the triangular transport map is its *anisotropic* dependence on the input variables. That is, even though each component of the transport map does not depend on all n inputs, the map is still capable of coupling arbitrary probability distributions. Informally, we can think of the KR rearrangement as imposing the *sparsest* possible structure that preserves generality of the coupling—in that the rearrangement is guaranteed to exist for any $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$. In Section 6, we will show that the anisotropy of the KR rearrangement is crucial to proving that certain "complex" (and generally non-triangular) transports can be factorized into compositions of a few *lower-dimensional* triangular maps. Thus we can think of the KR rearrangement as the fundamental building block of a more general class of non-triangular transports.

The KR rearrangement also enjoys many attractive computational features. As shown in Marzouk et al. (2016), it can be characterized as the unique minimizer of the Kullback–Leibler (KL) divergence $\mathcal{D}_{\mathrm{KL}}(\, T_\sharp \boldsymbol{\nu}_\eta \,||\, \boldsymbol{\nu}_\pi \,)$ over the cone $\mathcal{T}_\triangle$ of monotone increasing triangular

maps. From the perspective of function approximation, parameterizing a monotone triangular map is straightforward: it suffices to write each component of the map as[2]

$$T^k(\boldsymbol{x}) = a_k(x_1, \dots, x_{k-1}) + \int_0^{x_k} \exp\left(b_k(x_1, \dots, x_{k-1}, t)\right) \, \mathrm{d}t, \tag{5}$$

for some arbitrary functions $a_k : \mathbb{R}^{k-1} \to \mathbb{R}$ and $b_k : \mathbb{R}^k \to \mathbb{R}$ (Ramsay, 1998). For example, one could parameterize each a_k, b_k using a linear expansion

$$a_k(\boldsymbol{x}) = \sum_i a_{k,i} \, \psi_i(\boldsymbol{x}), \qquad b_k(\boldsymbol{x}) = \sum_j b_{k,j} \, \psi_j(\boldsymbol{x})$$

in terms of multivariate Hermite polynomials (ψ_i) and unknown coefficients $\boldsymbol{c} = (a_{k,i}, b_{k,j})$; alternatively, one could use a neural network representation of a_k and b_k. The resulting transport map $T[\boldsymbol{c}]$—parameterized by the coefficients $\boldsymbol{c}$—is monotone and invertible for all choices of $\boldsymbol{c}$. (In contrast, parameterizing general classes of monotone *non-triangular* maps is a difficult task.) The minimization of $\mathcal{D}_{\mathrm{KL}}(\, T_\sharp \boldsymbol{\nu}_\eta \,||\, \boldsymbol{\nu}_\pi \,)$ for a map in $\mathcal{T}_\triangle$ and for a pair of nonvanishing target (π) and reference (η) densities can be rewritten as

$$\begin{aligned} \min_{T} \quad & -\mathbb{E}\left[\log \pi(T(\boldsymbol{X})) + \sum_k \log \partial_k T^k(\boldsymbol{X}) - \log \eta(\boldsymbol{X})\right] \\ \text{s.t.} \quad & T \in \mathcal{T}_\triangle, \end{aligned} \tag{6}$$

where the expectation is with respect to the reference measure—which is the law of $\boldsymbol{X}$.

Two aspects of (6) are particularly important. First, for the purpose of optimization, the target density can be replaced with its unnormalized version $\bar{\pi}$. (This replacement is essential in Bayesian inference, where the posterior normalizing constant is usually unknown.) Second, (6) can be treated as a stochastic program and solved by means of sample-average approximation (SAA) or stochastic approximation (Shapiro, 2013; Kushner and Yin, 2003). Recall that the reference measure is a degree of freedom of the problem and is chosen precisely to make the integration in (6) feasible using, for instance, quadrature, Monte Carlo, or quasi-Monte Carlo methods (Dick et al., 2013).

Assuming some additional regularity for π (e.g., at least differentiability) and using the monotone parameterization of (5), then (6) becomes an unconstrained and differentiable optimization problem. In particular, we can use the gradient of $\log \pi$ to obtain an unbiased estimator for the gradient of (6) (Asmussen and Glynn, 2007). Alternatively, if $\nabla \log \pi$ is unavailable, we can use the *score method* (Glynn, 1990) to produce an estimator that is still unbiased, but with higher variance. For concreteness, consider the realization of an i.i.d. sample $(\boldsymbol{x}_i)_{i=1}^M$ from $\boldsymbol{\nu}_\eta$. Then a SAA of (6) reads as:

$$\begin{aligned} \min_{T} \quad & -\sum_{i=1}^M \left(\log \bar{\pi}(T(\boldsymbol{x}_i)) + \sum_k \log \partial_k T^k(\boldsymbol{x}_i) - \log \eta(\boldsymbol{x}_i) \right) \\ \text{s.t.} \quad & T \in \mathcal{T}_\triangle, \end{aligned} \tag{7}$$

2. For computational efficiency, one may substitue the exponential function with any other strictly positive expression, like a positively shifted square function.

which is now amenable to deterministic optimization techniques. The numerical solution of (7) by means of an iterative method (e.g., BFGS, Wright and Nocedal, 1999) produces a sequence of maps $\widetilde{T}_1, \widetilde{T}_2, \ldots$ that are increasingly better approximations of the KR rearrangement, in the sense defined by (7). In particular, we can interpret $(\widetilde{T}_k)_k$ as a discrete time flow that pushes forward the collection of reference samples, $(\boldsymbol{x}_i)_{i=1}^M$, to the target distribution. See Figure 1 for a simple illustration. As shown by Moselhy and Marzouk (2012), the KL divergence $\mathcal{D}_{\mathrm{KL}}(\,\widetilde{T}_\sharp \boldsymbol{\nu}_\eta \,||\, \boldsymbol{\nu}_\pi\,)$ for an approximate map $\widetilde{T}$ can be estimated as:

$$\mathcal{D}_{\mathrm{KL}}(\,\widetilde{T}_\sharp \boldsymbol{\nu}_\eta \,||\, \boldsymbol{\nu}_\pi\,) \approx \frac{1}{2}\mathbb{V}\mathrm{ar}\left[\log \bar{\pi}(\widetilde{T}(\boldsymbol{X})) + \sum_k \log \partial_k \widetilde{T}^k(\boldsymbol{X}) - \log \eta(\boldsymbol{X})\right], \tag{8}$$

up to second-order terms, in the limit of $\mathcal{D}_{\mathrm{KL}}(\,\widetilde{T}_\sharp \boldsymbol{\nu}_\eta \,||\, \boldsymbol{\nu}_\pi\,) \to 0$, even if the normalizing constant of π is unknown. This convergence criterion is rather useful for *any* variational inference method, and is usually not available for techniques like MCMC. In the same way, one can construct effective estimators for the normalizing constant $\beta := \bar{\pi}/\pi$ as

$$\hat{\beta} = \exp \mathbb{E}\left[\log \bar{\pi}(\widetilde{T}(\boldsymbol{X})) + \sum_k \log \partial_k \widetilde{T}^k(\boldsymbol{X}) - \log \eta(\boldsymbol{X})\right]. \tag{9}$$

We refer the reader to (Parno, 2015; Parno and Marzouk, 2018) for an alternative construction of the transport map that is useful when only *samples from the target measure* are available. An interesting application of the latter construction is the problem of density estimation or Bayesian inference with intractable likelihoods (Tabak and Turner, 2013; Csilléry et al., 2010). In this case, it turns out that the *inverse* transport $S = T^{-1}$ can be easily computed via convex optimization. (Notice that S is just an ordinary triangular transport map that pushes forward $\boldsymbol{\nu}_\pi$ to $\boldsymbol{\nu}_\eta$. The "inverse" descriptor will help distinguish S from the map T that pushes forward the reference to the target distribution. We refer to T as the *direct* transport.) We can then invert S at $\boldsymbol{x} \in \mathbb{R}^n$ to obtain the evaluation of the direct transport $T(\boldsymbol{x})$. Inverting a monotone triangular function is a computationally trivial task since it requires the solution of a sequence of one-dimensional root finding problems. In practice, one just needs to invert (3) for $k = 1, \ldots, n$. It is also possible to compute the inverse transport from the unnormalized target density, rather than from samples; here, it suffices to minimize $\mathcal{D}_{\mathrm{KL}}(\,\boldsymbol{\nu}_\eta \,||\, S_\sharp \boldsymbol{\nu}_\pi\,)$ for $S \in \mathcal{T}_\triangle$. The resulting variational problem is equivalent to (6) with the identity $S = T^{-1}$. By symmetry of our formulation, S has the same regularity as T. In particular, Lemma 1 holds for S as well, and gives a formula for the pushforward density $T_\sharp \eta$ as:

$$T_\sharp \eta(\boldsymbol{z}) = \eta(S(\boldsymbol{z}))\, \det \nabla S(\boldsymbol{z}) = \pi(\boldsymbol{z}),$$

where $\det \nabla S := \prod_{i=1}^n \partial_k S^k$ exists a.e., and where $T_\sharp \eta$ is the density of $T_\sharp \boldsymbol{\nu}_\eta$.

There is a growing body of literature on the efficient numerical approximation of transport maps (e.g., Rezende and Mohamed, 2015; Bigoni et al., 2019; Mendoza et al., 2018). Essentially all of these approaches employ numerical optimization to construct or realize the action of a map, and thus harness *optimization* to enhance *integration*. Yet all these approaches face a fundamental challenge: the transport map is a function from $\mathbb{R}^n$ onto

itself, and in high dimensions (i.e., for large n) the representation and approximation of such functions becomes increasingly intractable. In the ensuing sections, on the other hand, we will show that a large class of transport maps are in fact only superficially high-dimensional; that is, they possess some *hidden* low-dimensional structure that can facilitate their fast and reliable computation. This low-dimensional structure is linked to the Markov properties of the target measure, which we briefly review in the next section.

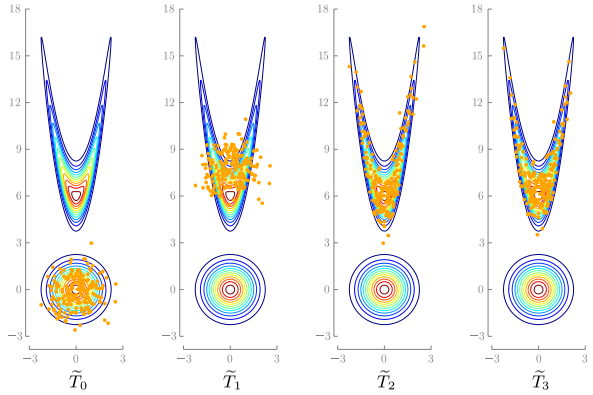


Figure 1: Computation of a simple transport map in two dimensions: The leftmost figure shows contours of the reference density η, which is a standard Gaussian, and of the target density π, which is a banana-shaped distribution in the tails of η. The target distribution has a nonlinear dependence structure. The orange dots in the leftmost figure correspond to 100 samples $(\boldsymbol{x}_i)$ from η and are used to make a sample-average approximation of (6). We adopt the triangular monotone parameterization of (5) for the candidate transport map, where the functions a_k, b_k are expanded in a multivariate Hermite polynomial basis of total degree two (Xiu, 2010). The resulting optimization problem is solved with a quasi-Newton method (BFGS). The kth figure from the left shows the pushforward of the original reference samples through the approximate transport map, $\widetilde{T}_k$, after k iterations of BFGS. The initial map $\widetilde{T}_0$ is chosen to be the identity. The reference samples flow *collectively* towards the target density and eventually settle on the support of π, capturing its structure after just a few iterations.

4. Markov Networks

Let $\boldsymbol{Z} = (Z_1, \ldots, Z_n)$ be a collection of random variables with law $\boldsymbol{\nu}_\pi$ and density π. We can represent a list of conditional independences satisfied by $\boldsymbol{Z}$—the so-called Markov properties—using a simple undirected graph $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$, where each node $k \in \mathcal{V}$ is associated

with a distinct random variable, Z_k, and where the edges in $\mathcal{E}$ encode a specific notion of probabilistic interaction among these random variables (Koller and Friedman, 2009). In particular, we say that $\boldsymbol{Z}$ is a Markov network—or a Markov random field (MRF)—with respect to $\boldsymbol{\mathcal{G}}$ if for any triplet $\mathcal{A}, \mathcal{S}, \mathcal{B}$ of disjoint subsets of $\mathcal{V}$, where $\mathcal{S}$ is a separator set for $\mathcal{A}$ and $\mathcal{B}$,[3] the subcollections $\boldsymbol{Z}_\mathcal{A}$ and $\boldsymbol{Z}_\mathcal{B}$ are conditionally independent given $\boldsymbol{Z}_\mathcal{S}$, i.e.,

$$\boldsymbol{Z}_\mathcal{A} \perp\!\!\!\perp \boldsymbol{Z}_\mathcal{B} \,|\, \boldsymbol{Z}_\mathcal{S}. \tag{10}$$

The measure $\boldsymbol{\nu}_\pi$ is said to satisfy the global Markov property, relative to $\boldsymbol{\mathcal{G}}$, if (10) holds. We can also say that $\boldsymbol{\nu}_\pi$ is globally Markov with respect to $\boldsymbol{\mathcal{G}}$. The corresponding graph is then called an independence map (I-map) for $\boldsymbol{\nu}_\pi$.

Intuitively, a sparse graph represents a family of distributions that enjoy many conditional independence properties. I-maps are in general not unique. Of particular interest are *minimal* I-maps, i.e., the sparsest graphs compatible with the conditional independence structure of $\boldsymbol{\nu}_\pi$.

Conditional independence is associated with factorization properties of π. For instance, $\boldsymbol{Z}_\mathcal{A} \perp\!\!\!\perp \boldsymbol{Z}_\mathcal{B} \,|\, \boldsymbol{Z}_\mathcal{S}$ if and only if $\pi_{\boldsymbol{Z}_\mathcal{A}, \boldsymbol{Z}_\mathcal{B} | \boldsymbol{Z}_\mathcal{S}} = \pi_{\boldsymbol{Z}_\mathcal{A} | \boldsymbol{Z}_\mathcal{S}} \, \pi_{\boldsymbol{Z}_\mathcal{B} | \boldsymbol{Z}_\mathcal{S}}$ a.e. (Lauritzen, 1996). We then say that $\boldsymbol{\nu}_\pi$ *factorizes* according to some graph $\boldsymbol{\mathcal{G}}$ if there exists a version of the density of $\boldsymbol{\nu}_\pi$ such that

$$\pi(\boldsymbol{z}) = \frac{1}{\mathfrak{c}} \prod_{\mathcal{C} \in \boldsymbol{\mathcal{C}}} \psi_\mathcal{C}(\boldsymbol{z}_\mathcal{C}), \tag{11}$$

for some nonnegative functions $(\psi_\mathcal{C})$ called *potentials*, where $\boldsymbol{\mathcal{C}}$ is the set of maximal cliques[4] of $\boldsymbol{\mathcal{G}}$ and $\mathfrak{c}$ is a normalizing constant. It is immediate to show that if $\boldsymbol{\nu}_\pi$ factorizes according to $\boldsymbol{\mathcal{G}}$, then $\boldsymbol{\nu}_\pi$ satisfies the global Markov property relative to $\boldsymbol{\mathcal{G}}$ (Lauritzen, 1996, Prop. 3.8). The converse is true only under additional assumptions: for instance, if $\boldsymbol{\nu}_\pi$ admits a continuous and strictly positive density (see the Hammersley-Clifford theorem; Hammersley and Clifford, 1971; Lauritzen, 1996).

A critical question then is how to characterize a suitable I-map for a given measure. There are several answers. First of all, in many applications that involve probabilistic modeling, the target distribution is defined in terms of its potentials, as in (11), because this is just a more convenient way to specify a high-dimensional distribution and to perform inference (or general probabilistic reasoning) with it. Finding a graph for which $\boldsymbol{\nu}_\pi$ factorizes is then a trivial task. See Figure 4 (*left*) for an example. Applications where this commonly holds range from spatial statistics and image analysis to speech recognition (Koller and Friedman, 2009; Rue and Held, 2005). In Section 7, for example, we focus exclusively on discrete-time Markov processes, where the Markov structure of the problem is self-evident. More specifically, Section 7 tackles the problem of recursive smoothing and static parameter estimation for a state-space model. In this context, the target measure $\boldsymbol{\nu}_\pi$ could represent the joint distribution of state and parameters, conditioned on all the available observations (see Figures 4 and 8). The reader might want to consider this sequential inference problem

3. $\mathcal{S}$ is a separator set for $\mathcal{A}$ and $\mathcal{B}$ if (1) $\mathcal{S}$ is disjoint from $\mathcal{A}$ and $\mathcal{B}$, and if (2) every path from $\alpha \in \mathcal{A}$ to $\beta \in \mathcal{B}$ intersects $\mathcal{S}$. If $\mathcal{A}$ and $\mathcal{B}$ are disconnected components of $\boldsymbol{\mathcal{G}}$, then $\mathcal{S} = \emptyset$ is a separator set for $\mathcal{A}$ and $\mathcal{B}$.
4. A clique is a fully connected subset of the vertices, whereas a maximal clique is a clique that is not a strict subset of another clique.

as a guiding application while reading the forthcoming Sections 5 and 6. We emphasize, however, that our theory is far more general and by no means restricted to any specific Markov structure.

In other settings, the graph is unknown and must be estimated. When only samples from $\boldsymbol{\nu}_\pi$ are available, this is a question of model learning (Koller and Friedman, 2009, Part III)—a problem with various applications (Hyvärinen, 2005; Meinshausen and Bühlmann, 2006; Lin et al., 2015). In case of a known and smooth target density, we can characterize pairwise conditional independence in terms of mixed second-order partial derivatives, as shown by the following lemma.

Lemma 2 (Pairwise conditional independence) *If $\boldsymbol{Z} \sim \boldsymbol{\nu}_\pi$ for a measure $\boldsymbol{\nu}_\pi$ with smooth and strictly positive density π, we have:*

$$Z_i \perp\!\!\!\perp Z_j \,|\, \boldsymbol{Z}_{\mathcal{V}\setminus(i,j)} \quad \iff \quad \partial^2_{i,j} \log \pi = 0 \text{ on } \mathbb{R}^n.$$

Thus, if we can evaluate π and its derivatives (up to a normalizing constant), we can use Lemma 2 to assess pairwise conditional independence and to define a minimal I-map for $\boldsymbol{\nu}_\pi$ as follows: add an edge between every pair of distinct nodes unless the corresponding random variables are conditionally independent (Koller and Friedman, 2009, Thm. 4.5).

Regardless of the many ways to obtain an I-map, there is a fundamental connection between Markov properties of a distribution and the existence of low-dimensional transport maps. The rest of the paper will elaborate precisely on this connection.

5. Sparsity of Triangular Transport Maps

We begin our investigation of low dimensional structure by considering the notion of sparse transport map. A sparse map is a multivariate function where each component does not depend on all of its input variables. According to this definition, a triangular transport is already sparse. In this section, however, we show that the KR rearrangement can be even *sparser*, depending on the Markov structure of the target distribution.

5.1. Sparsity Bounds

Given a lower triangular function T, we define its sparsity pattern, $\mathfrak{I}_T$, as the set of all integer pairs (j,k), with $j < k$, such that the kth component of the map does not depend on the jth input variable, i.e., $\mathfrak{I}_T = \{(j,k) : j < k, \, \partial_j T^k = 0\}$. (We do not include pairs $j > k$ in the definition of $\mathfrak{I}_T$ since, for a lower triangular function, $\partial_j T^k = 0$ for $j > k$ by construction.)

Knowing the sparsity pattern of the KR rearrangement *before* computing the actual transport has important computational implications. For instance, in the variational characterization of the transport described in (6), we can restrict the feasible domain to the set of triangular maps with sparsity pattern given by $\mathfrak{I}_T$, and still recover the desired KR rearrangement. That is, if $(j,k) \in \mathfrak{I}_T$, we can parameterize any candidate transport map by removing the dependence on the jth input variable from the kth component of the map. Thus, analyzing the Markov structure of the target distribution enables the representation and computation of maps in possibly higher-dimensional settings.

The following theorem, which is the main result of this section, characterizes *bounds* on the sparsity patterns of triangular transport maps given an I-map for the target measure. In the statement of the theorem, we denote the direct transport by T and the inverse transport by $S = T^{-1}$ (see Section 3). The theorem suggests that S and T can have quite different sparsity patterns.[5]

Theorem 3 (Sparsity of Knothe–Rosenblatt rearrangements) *Let $\boldsymbol{X} \sim \boldsymbol{\nu}_\eta$, $\boldsymbol{Z} \sim \boldsymbol{\nu}_\pi$ with $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$ and $\boldsymbol{\nu}_\eta$ a product measure on $\times_{i=1}^n \mathbb{R}$. Moreover, assume that $\boldsymbol{\nu}_\pi$ is globally Markov with respect to $\boldsymbol{\mathcal{G}}$, and define, recursively, the sequence of graphs $(\boldsymbol{\mathcal{G}}^k)_{k=1}^n$ as: (1) $\boldsymbol{\mathcal{G}}^n \coloneqq \boldsymbol{\mathcal{G}}$ and (2) for all $1 \leq k < n$, $\boldsymbol{\mathcal{G}}^{k-1}$ is obtained from $\boldsymbol{\mathcal{G}}^k$ by removing node k and by turning its neighborhood $\mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$ into a clique. Then the following hold:*

1. *If $\mathfrak{I}_S$ is the sparsity pattern of the inverse transport map S, then*

$$\widehat{\mathfrak{I}}_S \subset \mathfrak{I}_S, \tag{12}$$

   where $\widehat{\mathfrak{I}}_S$ is the set of integer pairs (j,k) such that $j \notin \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$.

2. *If $\mathfrak{I}_T$ is the sparsity pattern of the direct transport map T, then*

$$\widehat{\mathfrak{I}}_T \subset \mathfrak{I}_T, \tag{13}$$

   where $\widehat{\mathfrak{I}}_T$ is defined recursively as follows: for $k = 2, \dots, n$ the pair $(j,k) \in \widehat{\mathfrak{I}}_T$ if and only if $(j,i) \in \widehat{\mathfrak{I}}_T$ for all $i \in \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$.

3. *The predicted sparsity pattern of S is always greater than or equal to that of T, i.e.,*

$$\widehat{\mathfrak{I}}_T \subset \widehat{\mathfrak{I}}_S. \tag{14}$$

Several remarks are in order. First, we emphasize the fact that Theorem 3 characterizes sparsity patterns using *only* an I-map for $\boldsymbol{\nu}_\pi$, without requiring any actual computation of the transports. One only needs to perform simple graph operations on $\boldsymbol{\mathcal{G}}$ to build the sequence of graphs $(\boldsymbol{\mathcal{G}}^k)$. See Figure 2 for an illustration of this procedure, with the corresponding sparsity patterns in Figure 3. We refer to $(\boldsymbol{\mathcal{G}}^k)$ as the *marginal* graphs. In fact, the sequence $(\boldsymbol{\mathcal{G}}^k)$ is precisely the set of intermediate graphs produced by the variable elimination algorithm (Koller and Friedman, 2009, Ch. 9), when marginalizing with elimination ordering $(n, n-1, \dots, 1)$. This should not be surprising as the KR rearrangement is essentially a sequence of ordered marginalizations (Villani, 2008). The hypothesis that $\boldsymbol{\nu}_\eta$ is a product measure is important for the theorem to hold. If we pick a reference measure with an arbitrary Markov structure, there need not exist a sparse transport map coupling $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$, even if $\boldsymbol{\nu}_\pi$ has a sparse I-map. The role of a reference measure is somewhat peculiar to the world of couplings and is usually not addressed in classical treatments of graphical models. Nonetheless, this assumption on $\boldsymbol{\nu}_\eta$ is not restrictive in the present framework,

5. A note: as we already saw, the KR rearrangement is unique up to a set of measure zero. Theorem 3 characterizes the sparsity pattern of a particular *version* of the map, the one given by Definition 14 in Appendix A. We will implicitly make this assumption throughout the paper.

since the reference distribution is considered a degree of freedom of the problem. Theorem 3 gives sufficient but not necessary conditions on $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ for the existence of a sparse map. And it could not be otherwise: if $\boldsymbol{\nu}_\eta = \boldsymbol{\nu}_\pi$ then the identity map—the sparsest possible map—would be a valid coupling.

We also note that Theorem 3 does not provide the exact sparsity patterns of the triangular transport maps; instead, (12) and (13) provide *subsets* of $\mathfrak{I}_T$ and $\mathfrak{I}_S$. In other words, the actual transport maps might be sparser than predicted by the sets $\widehat{\mathfrak{I}}_S$ and $\widehat{\mathfrak{I}}_T$—but, crucially, they cannot be less sparse. Thus, we can think of Theorem 3 as providing *bounds* on the sparsity of triangular transports. An important fact is that, without additional information on $\boldsymbol{\nu}_\pi$, these bounds are sharp. That is, we can always find a pair of measures $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ satisfying the hypotheses of Theorem 3 and such that the predicted and actual sparsity patterns coincide, i.e., $\widehat{\mathfrak{I}}_T = \mathfrak{I}_T$ or $\widehat{\mathfrak{I}}_S = \mathfrak{I}_S$.

Part 3 of Theorem 3 shows that the predicted sparsity pattern of the inverse KR rearrangement is always larger than or equal to that of the direct transport, i.e., $\widehat{\mathfrak{I}}_T \subset \widehat{\mathfrak{I}}_S$. This does not mean that for every pair of measures $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$, the inverse triangular transport is always at least as sparse as the direct transport; in fact, it is possible to provide simple counterexamples. However, this result does imply that if we are only given an I-map for $\boldsymbol{\nu}_\pi$, then parameterizing candidate *inverse* triangular transports allows the imposition of more sparsity constraints than parameterizing candidate direct transports. In general, sparser transports are easier to represent. See Figure 4 (*right*) for a nontrivial example of sparsity patterns for a stochastic volatility model.

Indeed, (14) hints at a typical trend: inverse transport maps tend to be sparser (in many practical cases, *much* sparser) than their direct counterparts. Intuitively, the sparsity of a direct transport is associated with marginal independence in $\boldsymbol{Z}$, whereas the inverse transport inherits sparsity from the conditional independence structure of $\boldsymbol{Z}$. The latter is a weaker condition than mutual independence; for instance, the correlation length of a process modeled by a Markov random field may be much larger than the typical neighborhood size (Rue and Held, 2005). Thus, given a sparse I-map for the target measure, it can be computationally advantageous to characterize an inverse transport rather than a direct one, because the inverse transport can inherit a larger sparsity pattern. Given an inverse triangular transport S, we can then easily evaluate the direct transport $T = S^{-1}$ at any point $\boldsymbol{x} \in \mathbb{R}^n$ by inverting S pointwise, as described in Section 3. There is no need to have an explicit representation of the direct transport as long as it can be implicitly defined through its inverse.

5.2. Connection to Gaussian Markov Random Fields

The reader familiar with Gaussian Markov random fields (GMRFs), might see links between the preceding results and widespread approaches to the modeling of Gaussian fields. In this section, we clarify the extent of these connections.

Many applications (e.g., image analysis, spatial statistics, time series) involve modeling by means of high-dimensional Gaussian fields. Dealing with large and dense covariances, however, is often impractical; both storage and sampling of the Gaussian field are problematic. The usual workaround is to replace or approximate the Gaussian field with a *sparse* GMRF—i.e., a Gaussian Markov network that enforces locality in the probabilistic interac-

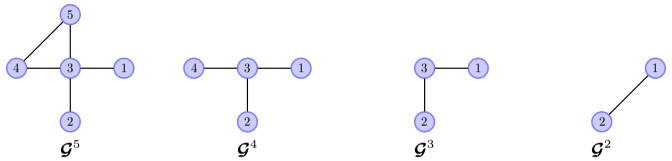


Figure 2: Sequence of graphs $(\mathcal{G}^k)$ described in Theorem 3 for a target measure in $\mathscr{M}_+(\mathbb{R}^5)$ with I-map illustrated by the leftmost graph, $\mathcal{G}^5$. Notice that to generate the graph $\mathcal{G}^2$, we remove node 3 from $\mathcal{G}^3$ and turn its neighborhood into a clique by adding the edge $(1, 2)$.

Figure 3: Sparsity patterns predicted by Theorem 3 for the target measure analyzed in Figure 2. We represent the sparsity patterns using a symbolic matrix notation: the (j, k)-th entry of the matrix is *not* colored if the kth component of the map (S or T) does not depend on the jth input variable, or, equivalently, if $(j, k) \in \widehat{\mathfrak{I}}_S$ (resp. $\widehat{\mathfrak{I}}_T$) (12). (Since we are considering lower triangular transports, all entries $j > k$ are uncolored. Note also that S^k and T^k are always functions of their kth input by strict monotonicity of the map.) The predicted sparsity pattern for the direct transport in this example is $\widehat{\mathfrak{I}}_T = \emptyset$.

tions among the underlying random variables. The minimal I-map for the GMRF is thus sparse, and so is the precision matrix Λ of the field (Rue and Held, 2005). The covariance matrix is still in general dense, but dealing with the sparse precision matrix is much easier. If $LL^\top$ is a (sparse) Cholesky decomposition of Λ, then $L^\top$ represents a linear triangular transport that pushes forward the joint distribution of the GMRF, $\boldsymbol{\nu}_\pi = \mathcal{N}(0, \Lambda^{-1})$, to a standard normal, $\boldsymbol{\nu}_\eta = \mathcal{N}(0, \mathbf{I})$. The key point is that for many Markov structures of interest, the Cholesky factor inherits sparsity from the underlying graph, so that sampling from $\boldsymbol{\nu}_\pi$ can be achieved at low cost as follows: if $\boldsymbol{X}$ is a sample from $\boldsymbol{\nu}_\eta$, then we can obtain a sample $\boldsymbol{Z}$ from $\boldsymbol{\nu}_\pi$ simply by solving the sparse triangular linear system $L^\top \boldsymbol{Z} = \boldsymbol{X}$. There is no need to explicitly represent or store the dense factor $L^{-\top}$, since we can implicitly represent its action by inverting a sparse triangular function.

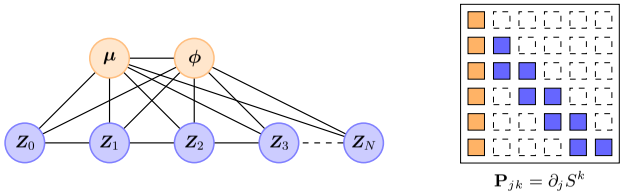


Figure 4: (*left*) Markov network for a stochastic volatility model (Kim et al., 1998). Blue nodes represent the discrete-time latent log-volatility process $(\boldsymbol{Z}_k)_{k=0}^N$, which obeys a simple autoregressive model with hyperparameters $\boldsymbol{\mu}, \boldsymbol{\phi}$. The graph above is a minimal I-map for the posterior density described in Section 8, $\pi_{\boldsymbol{\mu},\boldsymbol{\phi},\boldsymbol{Z}_{0:N}|\boldsymbol{y}_{0:N}}$, where $\boldsymbol{y}_{0:N}$ are some (fixed) observations. (*right*) The predicted sparsity pattern $\widehat{\mathfrak{I}}_S$ (only the top 6×6 block is shown) for the inverse transport corresponding to the model on the left: the first column/row of the matrix refer jointly to all of the hyperparameters. Each component S^k of the inverse transport can depend at most on four input variables, namely $\boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{Z}_{k-1}, \boldsymbol{Z}_k$, regardless of the overall dimension N of the problem. In order to apply the results of Theorem 3, we must select an ordering of the input variables; here, we used the ordering $(\boldsymbol{\mu}, \boldsymbol{\phi}, \boldsymbol{Z}_0, \ldots, \boldsymbol{Z}_N)$. Optimal orderings are further discussed in Section 5.3.

Now the connection with Section 5.1 is clear: $L^\top$ is an inverse triangular transport,[6] while $L^{-\top}$ is a direct one. Moreover, solving a triangular linear system is just a particular instance of inverting a nonlinear triangular function by performing a sequence of one-dimensional root-findings. Thus the developments of the previous section, which consider arbitrary *nonlinear* maps, are a natural generalization—to the *non-Gaussian* case—of modeling and sampling techniques for high-dimensional GMRFs (Rue and Held, 2005).

5.3. Ordering of Triangular Maps

The results of Theorem 3 suggest that the sparsity of a triangular transport map depends on the ordering of the input variables. See Figure 5 for a simple illustration. Indeed, the triangular transport itself depends anisotropically on the input variables and requires the definition of a proper ordering. A natural approach is then to seek the ordering that promotes the *sparsest* transport map possible.

Consider a pair of measures $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ that satisfies the hypotheses of Theorem 3. We associate an ordering of the input variables with a permutation σ on $\mathbb{N}_n = \{1, \ldots, n\}$, and define the *reordered* target measure $\boldsymbol{\nu}_\pi^\sigma$ as the pushforward of $\boldsymbol{\nu}_\pi$ by the matrix Q^σ that represents the permutation σ. In particular, $(Q^\sigma)_{ij} = (\boldsymbol{e}_{\sigma(i)})_j$, where $\boldsymbol{e}_i$ is the ith standard basis vector on $\mathbb{R}^n$. Moreover, if $\boldsymbol{\mathcal{G}}$ is an I-map for $\boldsymbol{\nu}_\pi$, then we denote an I-map for $\boldsymbol{\nu}_\pi^\sigma$ by

6. Actually, this transport is upper rather than lower triangular. This distinction plays no role in the following discussion, and the fact that a KR rearrangement is a lower triangular function is merely a matter of convention.

$\mathcal{G}^\sigma$. Notice that $\mathcal{G}^\sigma$ can be derived from $\mathcal{G}$ simply by relabeling its nodes according to the permutation σ. Then we can cast a variational problem for the *best* ordering σ^* as:

$$\begin{aligned} \sigma^* \in \arg\max_\sigma \quad & |\mathfrak{I}_S| \\ \text{s.t.} \quad & S_\sharp \boldsymbol{\nu}_\pi^\sigma = \boldsymbol{\nu}_\eta \\ & \sigma \in \mathfrak{P}(\mathbb{N}_n), \end{aligned} \tag{15}$$

where S is the KR rearrangement that pushes forward the reordered target $\boldsymbol{\nu}_\pi^\sigma$ to $\boldsymbol{\nu}_\eta$ and $\mathfrak{P}(\mathbb{N}_n)$ is the set of permutations of $\mathbb{N}_n$. The goal is to maximize the cardinality of the sparsity pattern of the inverse map, $|\mathfrak{I}_S|$. We restrict our attention to the sparsity of the inverse transport, since we know from Section 5.1 that the direct transport tends to be dense, even for the most trivial Markov structures.

Ideally, we would like to determine a good ordering for the map *before* computing the actual transport, and to use the resulting information about the sparsity pattern to simplify the optimization problem for S. However, evaluating the objective function of (15) requires computing a different inverse transport for each permutation σ. One possible way to relax (15) is to replace $\mathfrak{I}_S$ with the predicted sparsity pattern $\widehat{\mathfrak{I}}_S$ introduced in (12). The advantage of this approach is that the objective function of the relaxed problem can now be evaluated in closed form without computing any transport map, but rather by performing the simple sequence of graph operations on $\mathcal{G}^\sigma$ described by Theorem 3. The caveat is that, in general, $\widehat{\mathfrak{I}}_S \subset \mathfrak{I}_S$, and thus maximizing $|\widehat{\mathfrak{I}}_S|$ amounts to seeking the tightest lower bound on the sparsity pattern of the inverse transport. From the definition of $\widehat{\mathfrak{I}}_S$, it follows that the best ordering σ^* for the *relaxed* problem is one that introduces the fewest edges in the construction of the marginal graphs $\mathcal{G}^n, \dots, \mathcal{G}^1$, whenever $\mathcal{G}^n = \mathcal{G}^{\sigma^*}$. Thus, for a given I-map $\mathcal{G}$, we denote by $\mathfrak{F}(\sigma; \mathcal{G})$ the *fill-in* produced by the ordering σ. That is, $\mathfrak{F}(\sigma; \mathcal{G})$ is a set containing all the edges introduced in the construction of the marginal graphs $(\mathcal{G}^k)$ from $\mathcal{G}^\sigma$. A computationally feasible relaxation of (15) is then given by:

$$\begin{aligned} \sigma^* \in \arg\min_\sigma \quad & |\mathfrak{F}(\sigma; \mathcal{G})| \\ \text{s.t.} \quad & \sigma \in \mathfrak{P}(\mathbb{N}_n). \end{aligned} \tag{16}$$

(16) is a standard problem in graph theory; it arises in a variety of practical settings, including (most relatedly) finding the best elimination ordering for variable elimination in graphical models, or finding the permutation that minimizes the fill-in of the Cholesky factor of a positive definite matrix (George and Liu, 1989; Saad, 2003). From an algorithmic point of view, (16) is NP-complete (Yannakakis, 1981). This should not be surprising, as best–ordering problems are typically combinatorial in nature. Nevertheless, given its widespread applicability, a host of effective polynomial-time heuristics for (16) have been developed in past years (e.g., min-fill or weighted-min-fill, Koller and Friedman, 2009). Most importantly, (16) can be solved without ever touching the target measure (assuming, of course, that an I-map $\mathcal{G}$ for $\boldsymbol{\nu}_\pi$ is known). As a result, the cost of finding a good ordering is often negligible compared to the cost of characterizing a nonlinear transport map via optimization.

6. Decomposability of Transport Maps

Thus far, we have investigated the sparsity of triangular transport maps and found that inverse transports tend to inherit sparsity from the underlying Markov structure of the

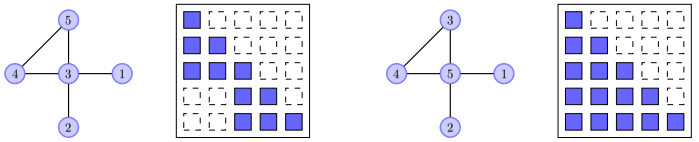


Figure 5: Illustration of how a simple re-ordering of the input variables can change the (predicted) sparsity pattern of the inverse map. On the left, $\mathcal{G}$ represents an I-map for the target measure considered in Figure 2, with ordering $(Z_1, Z_2, Z_3, Z_4, Z_5)$, together with its sparsity pattern $\widehat{\mathfrak{I}}_S$. (See Figure 3 for details on the "matrix" representation of sparsity patterns.) On the right, $\mathcal{G}'$ is an I-map for the same target measure but with the ordering $(Z_1, Z_2, Z_5, Z_4, Z_3)$. The corresponding sparsity pattern $\widehat{\mathfrak{I}}_{S'}$ is now the empty set.

target measure. Though direct triangular transports also inherit some sparsity according to Theorem 3, they tend to be more dense.

This section shows that direct transports enjoy a different form of low-dimensional structure: *decomposability*. A decomposable transport map is a function that can be written as the composition of a finite number of low-dimensional maps, e.g., $T = T_1 \circ \cdots \circ T_\ell$ for some integer $\ell \geq 2$. We use a very specific notion of low-dimensional map, as follows.

Definition 4 (Low-dimensional map with respect to a set) *A map $M : \mathbb{R}^n \to \mathbb{R}^n$ is low-dimensional with respect to a nonempty set $\mathcal{C} \subset \mathcal{V} \simeq \mathbb{N}_n$ if*

1. $M^k(\boldsymbol{x}) = x_k$ *for* $k \in \mathcal{C}$
2. $\partial_j M^k = 0$ *for* $j \in \mathcal{C}$ *and* $k \in \mathcal{V} \setminus \mathcal{C}$.

The effective dimension of M is the minimum cardinality $|\mathcal{V} \setminus \mathcal{C}|$ over all sets $\mathcal{C}$ with respect to which M is low-dimensional.

In particular, up to a permutation of its components, we can rewrite M as:

$$M(\boldsymbol{x}) = \begin{bmatrix} M^{\bar{\mathcal{C}}}(\boldsymbol{x}_{\bar{\mathcal{C}}}) \\ \boldsymbol{x}_{\mathcal{C}} \end{bmatrix},$$

where $\bar{\mathcal{C}} = \mathcal{V} \setminus \mathcal{C}$ denotes the complement of $\mathcal{C}$ in $\mathcal{V}$, and where for any map M and set $\mathcal{A} = \{a_1, \dots, a_k\}$, $M^{\mathcal{A}}$ denotes the multivariate function $\boldsymbol{x} \mapsto (M^{a_1}(\boldsymbol{x}), \dots, M^{a_k}(\boldsymbol{x}))$ obtained by stacking together the components of M with index in $\mathcal{A}$. Thus M is the trivial embedding of a $|\bar{\mathcal{C}}|$-dimensional function into the identity map and has *effective dimension* bounded by $|\bar{\mathcal{C}}| < n$. It is not surprising, then, that a decomposable transport $T = T_1 \circ \cdots \circ T_\ell$ should be easier to represent than an ordinary map. A perhaps less intuitive feature, however, is that the computation of a high-dimensional decomposable transport can be broken down into multiple simpler steps, each associated with the computation of a low-dimensional map T_j that accounts only for local features of the target measure.

The forthcoming analysis will consider *general*, and hence possibly non-triangular, transports. Thus its scope is much broader than that of Section 5, where we only focused on the sparsity of triangular transports. Yet, we will show that triangular maps are the building block of decomposable transports. The cornerstone of our analysis is Theorem 7, which characterizes the existence and structure of decomposable transports given only the Markov structure of the underlying target measure.

Our discussion will proceed in two stages: first, we show how to identify direct transports that decompose into two maps, i.e., $T = T_1 \circ T_2$, and then we explain how to apply this result recursively to obtain a general decomposition of the form $T = T_1 \circ \cdots \circ T_\ell$.

6.1. Preliminary Notions

Before addressing the decomposability of transport maps, we need to introduce two useful concepts: proper graph decompositions and generalized triangular functions. The decomposition of a graph is a standard notion (Lauritzen, 1996).

Definition 5 (Proper graph decomposition) *Given a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, a triple $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ of disjoint subsets of the vertex set $\mathcal{V}$ forms a proper decomposition of $\mathcal{G}$ if (1) $\mathcal{V} = \mathcal{A} \cup \mathcal{S} \cup \mathcal{B}$, (2) $\mathcal{A}$ and $\mathcal{B}$ are nonempty, (3) $\mathcal{S}$ separates $\mathcal{A}$ from $\mathcal{B}$, and (4) $\mathcal{S}$ is a clique.*

See Figure 6 (*top left*) for an example of a decomposition. Clearly, not every graph admits a proper decomposition; for instance, a fully connected graph does not have a separator set for nonempty $\mathcal{A}$ and $\mathcal{B}$. The idea we will pursue here is that graph decompositions lead to the existence of decomposable transports.

The notion of a generalized triangular function is perhaps less standard, but still relatively straightforward:

Definition 6 (Generalized triangular function) *A function $T : \mathbb{R}^n \to \mathbb{R}^n$ is said to be generalized triangular, or simply σ-triangular, if there exists a permutation σ of $\mathbb{N}_n$ such that the $\sigma(k)$th component of T depends only on the variables $x_{\sigma(1)}, \dots, x_{\sigma(k)}$, i.e., $T^{\sigma(k)}(\boldsymbol{x}) = T^{\sigma(k)}(x_{\sigma(1)}, \dots, x_{\sigma(k)})$ for all $\boldsymbol{x} = (x_1, \dots, x_n)$ and for all $k = 1, \dots, n$.*

We can think of a generalized triangular function as a map that is lower triangular up to a permutation. In particular, if σ is the identity on $\mathbb{N}_n$, then a σ-triangular function is simply a lower triangular map (see Section 3). To represent the permutation σ, we use the notation $\sigma(\{i_1, \dots, i_k\}) = \{\sigma(i_1), \dots, \sigma(i_k)\}$ to denote an ordered set that collects the action of the permutation on the elements (i_j). For example, if $T : \mathbb{R}^4 \to \mathbb{R}^4$ is a σ-triangular map with σ defined as $\sigma(\mathbb{N}_4) = \{1, 4, 2, 3\}$, then T will be of the form:

$$T(\boldsymbol{x}) = \begin{bmatrix} T^1(x_1) \\ T^2(x_1, x_4, x_2) \\ T^3(x_1, x_4, x_2, x_3) \\ T^4(x_1, x_4) \end{bmatrix}$$

for some collection (T^k). We regard each component $T^{\sigma(k)}$ as a map $\mathbb{R}^k \to \mathbb{R}$. We say that a σ-triangular function T is monotone increasing if each component T^k is a monotone increasing function of the input x_k. Moreover, for any $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$ and for any

permutation σ of $\mathbb{N}_n$, there exists a ($\boldsymbol{\nu}_\eta$-unique) monotone increasing σ-triangular map—which we call a σ-generalized KR rearrangement—that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$. We give a constructive definition for a generalized KR rearrangement in Appendix A.

A key property of a σ-generalized KR rearrangement is that it allows different sparsity patterns to be engineered, depending on σ, in a map that is otherwise fully general—in the sense of being able to couple arbitrary measures in $\mathscr{M}_+(\mathbb{R}^n)$. This feature will be essential to characterizing decomposable transport maps.

6.2. Decomposition and Graph Sparsification

We now characterize transports that decompose into a pair of low-dimensional maps, as described in the following theorem. We formulate the theorem for a generic target measure $\boldsymbol{\nu}_i$. Later we will apply the theorem recursively to a sequence $(\boldsymbol{\nu}_i)$ of different targets.

Theorem 7 (Decomposition of transport maps) *Let $\boldsymbol{X} \sim \boldsymbol{\nu}_\eta$, $\boldsymbol{Z}^i \sim \boldsymbol{\nu}_i$, with $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_i \in \mathscr{M}_+(\mathbb{R}^n)$ and $\boldsymbol{\nu}_\eta$ tensor product measure. Denote by η, π_i a pair of nonvanishing densities for $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_i$, respectively, and assume that $\boldsymbol{\nu}_i$ factorizes according to a graph $\boldsymbol{\mathcal{G}}^i$, which admits a proper decomposition $(\mathcal{A}, \mathcal{S}, \mathcal{B})$. Then the following hold:*

1. *There exists a factorization of π_i of the form*

$$\pi_i(\boldsymbol{z}) = \frac{1}{\mathfrak{c}} \, \psi_{\mathcal{A}\cup\mathcal{S}}(\boldsymbol{z}_{\mathcal{A}\cup\mathcal{S}}) \, \psi_{\mathcal{S}\cup\mathcal{B}}(\boldsymbol{z}_{\mathcal{S}\cup\mathcal{B}}), \tag{17}$$

   where $\psi_{\mathcal{A}\cup\mathcal{S}}$ is strictly positive and integrable, with $\mathfrak{c} = \int \psi_{\mathcal{A}\cup\mathcal{S}}$.

2. *For any factorization (17) and for any permutation σ of $\mathbb{N}_n$ with*

$$\sigma(k) \in \begin{cases} \mathcal{S} & \text{if } k = 1, \dots, |\mathcal{S}| \\ \mathcal{A} & \text{if } k = |\mathcal{S}| + 1, \dots, |\mathcal{A} \cup \mathcal{S}| \\ \mathcal{B} & \text{otherwise,} \end{cases} \tag{18}$$

   there exists a nonempty family, $\mathfrak{D}_i$, of decomposable transport maps $T = L_i \circ R$ parameterized by $R \in \mathfrak{R}_i$ such that each $T \in \mathfrak{D}_i$ pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_i$ and where:

   (a) *L_i is a σ-generalized KR rearrangement that pushes forward $\boldsymbol{\nu}_\eta$ to a measure with density $\psi_{\mathcal{A}\cup\mathcal{S}}(\boldsymbol{z}_{\mathcal{A}\cup\mathcal{S}}) \, \eta_{\boldsymbol{X}_{\mathcal{B}}}(\boldsymbol{z}_{\mathcal{B}})/\mathfrak{c}$ and is low-dimensional with respect to $\mathcal{B}$.*

   (b) *$\mathfrak{R}_i$ is the set of maps $\mathbb{R}^n \to \mathbb{R}^n$ that are low-dimensional with respect to $\mathcal{A}$ and that push forward $\boldsymbol{\nu}_\eta$ to the pullback $L_i^\sharp \, \boldsymbol{\nu}_i \in \mathscr{M}_+(\mathbb{R}^n)$.*

   (c) *If $\boldsymbol{Z}^{i+1} \sim L_i^\sharp \, \boldsymbol{\nu}_i$, then $\boldsymbol{Z}^{i+1}_{\mathcal{A}} \perp\!\!\!\perp \boldsymbol{Z}^{i+1}_{\mathcal{S}\cup\mathcal{B}}$ and $\boldsymbol{Z}^{i+1}_{\mathcal{A}} = \boldsymbol{X}_{\mathcal{A}}$ in distribution.*

   (d) *$L_i^\sharp \, \boldsymbol{\nu}_i$ factorizes according to a graph $\boldsymbol{\mathcal{G}}^{i+1}$ that can be derived from $\boldsymbol{\mathcal{G}}^i$ as follows:*

      – *Remove any edge from $\boldsymbol{\mathcal{G}}^i$ that is incident to any node in $\mathcal{A}$.*

      – *For any maximal clique $\mathcal{C} \subset \mathcal{S} \cup \mathcal{B}$ with nonempty intersection $\mathcal{C} \cap \mathcal{S}$, let $j_{\mathcal{C}}$ be the maximum integer j such that $\sigma(j) \in \mathcal{C} \cap \mathcal{S}$ and turn $\mathcal{C} \cup \{\sigma(1), \dots, \sigma(j_{\mathcal{C}})\}$ into a clique.*

We first look at the theorem for $i = 1$ and let $\boldsymbol{\nu}_1 \coloneqq \boldsymbol{\nu}_\pi$ and $\boldsymbol{\mathcal{G}}^1 \coloneqq \boldsymbol{\mathcal{G}}$, where $\boldsymbol{\nu}_\pi$ denotes our usual target measure with I-map $\boldsymbol{\mathcal{G}}$ and where $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ denotes a decomposition of $\boldsymbol{\mathcal{G}}$.

Among the infinitely many transport maps from $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, Theorem 7 identifies a family of decomposable ones. The existence of these maps relies exclusively on the Markov structure of $\boldsymbol{\nu}_\pi$: we just require $\boldsymbol{\mathcal{G}}$ to admit a (proper) decomposition.[7]

Each transport $T \in \mathfrak{D}_1$ pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ and is the composition of two low-dimensional maps, i.e., $T = L_1 \circ R$ for a fixed L_1 defined in Theorem 7[Part 2a] and for some $R \in \mathfrak{R}_1$. (We also write $\mathfrak{D}_1 \coloneqq L_1 \circ \mathfrak{R}_1$.[8]) The structure of these low-dimensional maps is quite interesting. Up to a reordering of their components, Theorem 7[Parts 2a and 2b] show that L_1 and R have an intuitive complementary form:

$$L_1(\boldsymbol{x}) = \begin{bmatrix} L_1^{\mathcal{A}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{A}}) \\ L_1^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}) \\ \boldsymbol{x}_{\mathcal{B}} \end{bmatrix}, \qquad R(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x}_{\mathcal{A}} \\ R^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{B}}) \\ R^{\mathcal{B}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{B}}) \end{bmatrix}. \tag{19}$$

(If $\mathcal{S} = \emptyset$, one can just remove $L_1^{\mathcal{S}}$ and $R^{\mathcal{S}}$ from (19), and drop the dependence of the remaining components on $\boldsymbol{x}_{\mathcal{S}}$.) In particular, L_1 and R have effective dimensions bounded by $|\mathcal{A} \cup \mathcal{S}|$ and $|\mathcal{S} \cup \mathcal{B}| = |\mathcal{V} \setminus \mathcal{A}|$, respectively (see Definition 4). Even though L_1 and R are low-dimensional maps, their composition is quite dense—in the sense of Section 5—and is in general nontriangular:

$$T(\boldsymbol{x}) = (L_1 \circ R)(\boldsymbol{x}) = \begin{bmatrix} L_1^{\mathcal{A}}(\, R^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{B}}),\, \boldsymbol{x}_{\mathcal{A}}) \\ L_1^{\mathcal{S}}(\, R^{\mathcal{S}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{B}})\,) \\ R^{\mathcal{B}}(\boldsymbol{x}_{\mathcal{S}}, \boldsymbol{x}_{\mathcal{B}}) \end{bmatrix},$$

and thus more difficult to represent and to work with. The key idea of decomposable transports is that they can be represented implicitly through the composition of their low-dimensional factors, similar to the way that direct transports can be represented implicitly through their sparse inverses (Section 5).

The sparsity patterns of L_1 and R in (19) are needed for the theorem to hold. In particular, L_1 must be a σ-triangular function with σ specified by (18). Notice that (18) does not prescribe an exact permutation, but just a few constraints on a feasible σ. Intuitively, these constraints say that L_1 should be a function whose components with indices in $\mathcal{S}$ depend only on the variables in $\mathcal{S}$ (whenever $\mathcal{S} \neq \emptyset$), and whose components with indices in $\mathcal{A}$ depend only on the variables in $\mathcal{A} \cup \mathcal{S}$. Thus, there is usually some freedom in the choice of σ. Different permutations lead to different families of decomposable transports, and can induce different sparsity patterns in an I-map, $\boldsymbol{\mathcal{G}}^2$, for $L_1^\sharp \boldsymbol{\nu}_\pi$ (Theorem 7[Part 2d]).

Part 2d of the theorem shows how to derive a possible I-map $\boldsymbol{\mathcal{G}}^2$—not necessarily minimal—by performing a sequence of graph operations on $\boldsymbol{\mathcal{G}}$. There are two steps: one that does not depend on σ and one that does. Let us focus first on the former: the idea is to remove from $\boldsymbol{\mathcal{G}}$ any edge that is incident to any node in $\mathcal{A}$, effectively disconnecting $\mathcal{A}$ from the rest of the graph. That is, if $\boldsymbol{Z}^2 \sim L_1^\sharp \boldsymbol{\nu}_\pi$, then, regardless of σ, L_1 makes $\boldsymbol{Z}^2_{\mathcal{A}}$ marginally

7. To obtain a proper decomposition of $\boldsymbol{\mathcal{G}}$, one is free to add edges to $\boldsymbol{\mathcal{G}}$ in order to turn the separator set $\mathcal{S}$ into a clique (see Definition 5); $\boldsymbol{\nu}_\pi$ still factorizes according to any less sparse version of $\boldsymbol{\mathcal{G}}$.
8. The notation here is intuitive: for a given $g : \mathbb{R}^n \to \mathbb{R}^n$ and for a given set of functions $\mathcal{F}$ from $\mathbb{R}^n$ to $\mathbb{R}^n$, $g \circ \mathcal{F}$ denotes the set of maps that can be written as $g \circ f$ for some $f \in \mathcal{F}$.

independent of $\boldsymbol{Z}^2_{\mathcal{S}\cup\mathcal{B}}$ by acting *locally* on $\boldsymbol{\mathcal{G}}$. And not only that: L_1 also ensures that the marginals of $\boldsymbol{\nu}_\eta$ and $L_1^\sharp \boldsymbol{\nu}_\pi$ agree along $\mathcal{A}$ (see Theorem 7[Part 2c]). Thus we should really interpret L_1 as the first step towards a progressive transport of $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$. L_1 is a local map: it can depend nontrivially only upon variables in $\boldsymbol{x}_{\mathcal{A}\cup\mathcal{S}}$. Indeed, in the most general case, $|\mathcal{A} \cup \mathcal{S}|$ is the minimum effective dimension of a low-dimensional map necessary to decouple $\mathcal{A}$ from the rest of the graph. The more edges incident to $\mathcal{A}$, the higher-dimensional a transport is needed. This type of *graph sparsification* requires a peculiar "block triangular" structure for L_1 as shown by (19): any σ-triangular function with σ given by (18) achieves this special structure. The second step of Part 2d shows that if $\mathcal{S} \neq \emptyset$, then it might be necessary to add edges to the subgraph $\boldsymbol{\mathcal{G}}_{\mathcal{S}\cup\mathcal{B}}$, depending on σ.[9] The relevant aspect of σ for this discussion is the definition of the permutation onto the first $|\mathcal{S}|$ integers. In general, there are $|\mathcal{S}|!$ different permutations that could induce different sparsity patterns in $\boldsymbol{\mathcal{G}}^2$. We shall see that permutations that add the fewest edges possible are of particular relevance.

6.3. Recursive Decompositions

The sparsity of $\boldsymbol{\mathcal{G}}^2$ is important because it affects the "complexity" of the maps in $\mathfrak{R}_1$: each $R \in \mathfrak{R}_1$ pushes forward $\boldsymbol{\nu}_\eta$ to $L_1^\sharp \boldsymbol{\nu}_\pi$. More specifically, by the previous discussion, we can see how the role of each $R \in \mathfrak{R}_1$ is really only that of matching the marginals of $\boldsymbol{\nu}_\eta$ and $L_1^\sharp \boldsymbol{\nu}_\pi$ along $\mathcal{V} \setminus \mathcal{A}$. A natural question then is whether we can break this matching step into simpler tasks, or, in the language of this section, whether $\mathfrak{R}_1$ contains transports that are further decomposable. Intuitively, we are seeking a finer-grained representation for some of the transports in $\mathfrak{R}_1$. The following lemma (for $i = 1$) provides a positive answer to this question as long as $\mathcal{V} \setminus \mathcal{A}$ is not fully connected in $\boldsymbol{\mathcal{G}}^2$. From now on, we denote $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ by $(\mathcal{A}_1, \mathcal{S}_1, \mathcal{B}_1)$, since we will be dealing with a sequence of different graph decompositions.

Lemma 8 (Recursive decompositions) *Let $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_i, \boldsymbol{\mathcal{G}}^i$ be defined as in the assumptions of Theorem 7 for a proper decomposition $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ of $\boldsymbol{\mathcal{G}}^i$, while let $\boldsymbol{\mathcal{G}}^{i+1}$ and $\mathfrak{D}_i = L_i \circ \mathfrak{R}_i$ be the resulting graph (Part 2d) and family of decomposable transports,[10] respectively. Then there are two possibilities:*

1. *$\mathcal{S}_i \cup \mathcal{B}_i$ is not a clique in $\boldsymbol{\mathcal{G}}^{i+1}$. In this case, it is possible to identify a proper decomposition $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ of $\boldsymbol{\mathcal{G}}^{i+1}$ for some $\mathcal{A}_{i+1}$ that is a strict superset of $\mathcal{A}_i$ by (possibly) adding edges to $\boldsymbol{\mathcal{G}}^{i+1}$ in order to turn $\mathcal{S}_{i+1}$ into a clique. Let $\mathfrak{D}_{i+1} = L_{i+1} \circ \mathfrak{R}_{i+1}$ be defined as in Theorem 7 for the pair of measures $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_{i+1} \coloneqq L_i^\sharp \boldsymbol{\nu}_i$ and $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$. Then the following hold:*
   (a) *$\mathfrak{R}_i \supset \mathfrak{D}_{i+1}$ and $L_i \circ \mathfrak{R}_i \supset L_i \circ L_{i+1} \circ \mathfrak{R}_{i+1}$.*
   (b) *L_{i+1} is low-dimensional with respect to $\mathcal{A}_i \cup \mathcal{B}_{i+1}$ and has effective dimension bounded by $|(\mathcal{A}_{i+1} \setminus \mathcal{A}_i) \cup \mathcal{S}_{i+1}|$.*
   (c) *Each $R \in \mathfrak{R}_{i+1}$ has effective dimension bounded by $|\mathcal{V} \setminus \mathcal{A}_{i+1}|$.*

9. This is not always the case. For instance, if $\mathcal{S}$ is a subset of every maximal clique of $\boldsymbol{\mathcal{G}}$ in $\mathcal{S} \cup \mathcal{B}$ that has nonempty intersection with $\mathcal{S}$, then, by Theorem 7[Part 2d], no edges need to be added.
10. Whenever we do not specify a permutation σ_i or a factorization (17) in the definition of L_i, it means that the claim holds true for any feasible choice of these parameters.

2. $\mathcal{S}_i \cup \mathcal{B}_i$ is a clique in $\mathcal{G}^{i+1}$. In this case, the decomposition of Part 1 does not exist.

Lemma 8[Part 1] shows that if $\mathcal{S}_1 \cup \mathcal{B}_1$ is not fully connected in $\mathcal{G}^2$, then there exists a proper decomposition $(\mathcal{A}_2, \mathcal{S}_2, \mathcal{B}_2)$ of $\mathcal{G}^2$ (obtained, possibly, by adding edges to $\mathcal{G}^2$ in $\mathcal{V} \setminus \mathcal{A}_1$) for which $\mathcal{A}_2$ is a strict superset of $\mathcal{A}_1$. One can then apply Theorem 7 for the pair $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_2 = L_1^\sharp \boldsymbol{\nu}_1$ and the decomposition $(\mathcal{A}_2, \mathcal{S}_2, \mathcal{B}_2)$. As a result, Part 1a of the lemma shows that $\mathfrak{R}_1$ contains a subset $\mathfrak{D}_2 = L_2 \circ \mathfrak{R}_2$ of decomposable transport maps where both L_2 and each $R \in \mathfrak{R}_2$ are local transports on $\mathcal{V} \setminus \mathcal{A}_1$, i.e., they are both low-dimensional with respect to $\mathcal{A}_1$. In particular, L_2 is responsible for decoupling $\mathcal{A}_2 \setminus \mathcal{A}_1$ from the rest of the graph and for matching the marginals of $\boldsymbol{\nu}_\eta$ and $L_2^\sharp L_1^\sharp \boldsymbol{\nu}_\pi = (L_1 \circ L_2)^\sharp \boldsymbol{\nu}_\pi$ along $\mathcal{A}_2 \setminus \mathcal{A}_1$. The effective dimension of L_2 is bounded above by the size of the separator set $\mathcal{S}_2$ plus the number of nodes in $\mathcal{A}_2 \setminus \mathcal{A}_1$ (Part 1b of the lemma). The effective dimension of each $R \in \mathfrak{R}_2$ is bounded by the cardinality of $\mathcal{V} \setminus \mathcal{A}_2$ and is, in the most general case, lower than that of the maps in $\mathfrak{R}_1$ (Part 1c). Moreover, by Part 1a, $\mathfrak{D}_1 = L_1 \circ \mathfrak{R}_1 \supset L_1 \circ L_2 \circ \mathfrak{R}_2$, which means that among the infinitely many decomposable transports that push forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, there exists at least one that factorizes as the composition of *three* low-dimensional maps as opposed to two, i.e., $T = L_1 \circ L_2 \circ R$ for some $R \in \mathfrak{R}_2$.

If, on the other hand, $\mathcal{S}_1 \cup \mathcal{B}_1$ is fully connected in $\mathcal{G}^2$, then by Lemma 8[Part 2] we know that the decomposition of Part 1 does not exist. As a result, we cannot use Theorem 7 to prove the existence of more finely decomposable transports in $\mathfrak{R}_1$. In other words, if we want to match the marginals of $\boldsymbol{\nu}_\eta$ and $L_1^\sharp \boldsymbol{\nu}_\pi$ along $\mathcal{V} \setminus \mathcal{A}_1 = \mathcal{S}_1 \cup \mathcal{B}_1$, then we must do so in one shot, using a *single* transport map.

The main idea, then, is to apply Lemma 8[Part 1], recursively, k times, where k is the first integer (possibly zero) for which $\mathcal{S}_{k+1} \cup \mathcal{B}_{k+1}$ is a clique in $\mathcal{G}^{k+2}$. After k iterations, the following inclusion must hold:

$$\mathfrak{D}_1 = L_1 \circ \mathfrak{R}_1 \supset L_1 \circ \cdots \circ L_{k+1} \circ \mathfrak{R}_{k+1}, \tag{20}$$

which shows that there exists a decomposable transport,

$$T = L_1 \circ \cdots \circ L_{k+1} \circ R, \tag{21}$$

for some $R \in \mathfrak{R}_{k+1}$, that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$. (Note that we can apply Lemma 8[Part 1] only finitely many times since $|\mathcal{V} \setminus \mathcal{A}_{i+1}|$ is an integer function strictly decreasing in i and bounded away from zero.) Each L_i in (20) is a σ_i-triangular map for some permutation σ_i that satisfies (21), and is low-dimensional with respect to $\mathcal{A}_{i-1} \cup \mathcal{B}_i$, i.e., for $i > 1$ and up to a permutation of its components,

$$L_i(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{x}_{\mathcal{A}_{i-1}} \\ L_i^{\mathcal{A}_i \setminus \mathcal{A}_{i-1}}(\boldsymbol{x}_{\mathcal{S}_i}, \boldsymbol{x}_{\mathcal{A}_i \setminus \mathcal{A}_{i-1}}) \\ L_i^{\mathcal{S}_i}(\boldsymbol{x}_{\mathcal{S}_i}) \\ \boldsymbol{x}_{\mathcal{B}_i} \end{bmatrix}.$$

The map R is low-dimensional with respect to $\mathcal{A}_{k+1}$ and can also be chosen as a generalized triangular function. Intuitively, we can think of L_i as decoupling nodes in $\mathcal{A}_i \setminus \mathcal{A}_{i-1}$ from the rest of the graph in an I-map for $(L_1 \circ \cdots \circ L_{i-1})^\sharp \boldsymbol{\nu}_\pi$. (Recall that by Lemma 8 all the

sets $(\mathcal{A}_i)$ are nested, i.e., $\mathcal{A}_1 \subset \cdots \subset \mathcal{A}_{k+1}$.) Figure 6 illustrates the mechanics underlying the recursive application of Lemma 8.

We emphasize that the existence and structure of (21) follow from simple graph operations on $\mathcal{G}$, and do not entail any actual computation with the target measure $\boldsymbol{\nu}_\pi$. Notice also that even if each map in the decomposition (20) is σ-triangular, the resulting transport map T need not be triangular at all. In other words, we obtain factorizations of general and possibly non-triangular transport maps in terms of low-dimensional generalized triangular functions. In this sense, we can regard triangular maps as a fundamental "building block" of a much larger class of transport maps.

Decomposable transports are clearly not unique. In particular, there are two factors that affect both the sparsity pattern and the number k of composed maps in the family $L_1 \circ \cdots \circ L_{k+1} \circ \mathfrak{R}_{k+1}$: the sequence of decompositions $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ and the sequence of permutations (σ_i). Usually, there is a certain freedom in the choice of these parameters, and each configuration might lead to a different family of decomposable transports. Of course some families might be more desirable than others: ideally, we would like the low-dimensional maps in the composition to have the smallest effective dimension possible. Recall that by Lemma 8 the effective dimension of each L_i can be bounded above by $|(\mathcal{A}_i \setminus \mathcal{A}_{i-1}) \cup \mathcal{S}_i|$ (with the convention $\mathcal{A}_0 = \emptyset$). Thus we should intuitively choose a decomposition $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ of $\mathcal{G}^i$ and a permutation σ_i for L_i that minimize the cardinality of $(\mathcal{A}_i \setminus \mathcal{A}_{i-1}) \cup \mathcal{S}_i$, and that, at the same time, minimize the number of edges added from $\mathcal{G}^i$ to $\mathcal{G}^{i+1}$. In principle, we should also account for the dimensions of all future maps in the recursion. In the most general case, this graph theoretic question could be addressed using dynamic programming (Bertsekas, 1995). In practice, however, we will often consider graphs for which a *good* sequence of decompositions and permutations is rather obvious (see Section 7). For instance, if the target distribution $\boldsymbol{\nu}_\pi$ factorizes according to a tree $\mathcal{G}$, then it is immediate to show the existence of a decomposable transport $T = T_1 \circ \cdots \circ T_{n-1}$ that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ and that factorizes as the composition of $n-1$ low-dimensional maps $(T_i)_{i=1}^{n-1}$, each associated to an edge of $\mathcal{G}$: it suffices to consider a sequence of decompositions $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ with $\mathcal{A}_1 \subset \mathcal{A}_2 \subset \cdots$, where, for a given rooted version of $\mathcal{G}$, $\mathcal{A}_i \setminus \mathcal{A}_{i-1}$ consists of a single node a_i with the largest depth in $\mathcal{G}_{\mathcal{V} \setminus \mathcal{A}_{i-1}}$, and where $\mathcal{S}_i$ contains the unique parent of that node. Remarkably, each map T_i has effective dimension less than or equal to two, independent of n—the size of the tree.

At this point, it might be natural to consider a junction tree decomposition of a triangulation of $\mathcal{G}$ (Koller and Friedman, 2009) as a convenient graphical tool to schedule the sequence of decompositions $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ needed to apply Lemma 8 recursively. Decomposable graphs are in fact ultimately chordal (Lauritzen, 1996). However, the situation might not be as straightforward. The problem is that the clique structure of $\mathcal{G}^i$, an I-map for $\boldsymbol{\nu}_i$, can be *very* different than that of $\mathcal{G}^{i+1}$, an I-map for $L_i^\sharp \boldsymbol{\nu}_i$; Theorem 7[Part 2d] shows that $\mathcal{G}^{i+1}$ might contain larger maximal cliques than those in $\mathcal{G}^i$, even if $\mathcal{G}^i$ is chordal (see Figure 6 for an example). Thus, working with a junction tree might require a bit of extra care.

6.4. Computation of Decomposable Transports

Given the existence and structure of a decomposable transport like (21), what to do with it? There are at least two possible ways of exploiting this type of information. First, one could solve a variational problem like (6) and enforce an explicit parameterization of the transport map as the composition $T = L_1 \circ \cdots \circ L_{k+1} \circ R$. In this scenario, one need only parameterize the low-dimensional maps (L_i, R) and optimize, jointly, over their composition. The advantage of this approach is that it bypasses the parameterization of a single high-dimensional function, T, altogether. See the literature on normalizing flows (Rezende and Mohamed, 2015) for possible computational ideas in this direction.

An alternative—and perhaps more intriguing—possibility is to compute the maps (L_i) sequentially by solving *separate* low-dimensional optimization problems—one for each map L_i. By Theorem 7[Part 2a] and Lemma 8, there exists a factorization (17) of π_i—a density of $L_{i-1}^\sharp \boldsymbol{\nu}_{i-1}$—for which L_i is a σ_i-generalized KR rearrangement that pushes forward $\boldsymbol{\nu}_\eta$ to a measure with density proportional to $\psi_{\mathcal{A}_i \cup \mathcal{S}_i}\, \eta_{\boldsymbol{X}_{\mathcal{B}_i}}$, where $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ is a decomposition of $\boldsymbol{\mathcal{G}}^i$ and $\boldsymbol{\mathcal{G}}^i$ is an I-map for $\boldsymbol{\nu}_i$. In general $\psi_{\mathcal{A}_i \cup \mathcal{S}_i}$ depends on L_{i-1}, and so the maps (L_i) must be computed sequentially.[11] In essence, decomposable transports break the inference task into smaller and possibly easier steps.

Note that we could define L_i with respect to any factorization (17) with $\psi_{\mathcal{A}_i \cup \mathcal{S}_i}$ integrable: these different factorizations would lead to a family of decomposable transports with the same low-dimensional structure and sparsity patterns (as predicted by Theorem 7). Thus, as long as we have access to a sequence of integrable factors $(\psi_{\mathcal{A}_i \cup \mathcal{S}_i})$, we can compute each map L_i individually by solving a low-dimensional optimization problem. (See Appendix A for computational remarks on generalized triangular functions.) Intuitively, since by Lemma 8[Part 1b] L_i is low-dimensional with respect to $\mathcal{A}_{i-1} \cup \mathcal{B}_i$, we really only need to optimize for a portion of the map, namely $L_i^{\mathcal{C}}$ for $\mathcal{C} = (\mathcal{A}_i \setminus \mathcal{A}_{i-1}) \cup \mathcal{S}_i$, which can be regarded effectively as a multivariate map on $\mathbb{R}^{|\mathcal{C}|}$. In the same way, the map R can be computed as any transport (possibly triangular) that pushes forward $\boldsymbol{\nu}_\eta$ to $L_{k+1}^\sharp \boldsymbol{\nu}_{k+1}$. Theorem 7[Part 2b] tells us that once again we only need to optimize for a low-dimensional portion of the map, namely, $R^{\mathcal{S}_{k+1} \cup \mathcal{B}_{k+1}}$.

While it might be difficult to access a sequence of factorizations (17) for a general problem, there are important applications, such as Bayesian filtering, smoothing, and joint parameter/state estimation, where the sequential computation of the transports (L_i, R) is always possible by construction. We discuss these applications in the next section.

7. Sequential Inference on State-Space Models: Variational Algorithms

In this section, we consider the problem of sequential Bayesian inference (or discrete-time data assimilation; Reich and Cotter, 2015) for continuous, nonlinear, and non-Gaussian state-space models.

Our goal is to specialize the theory developed in Section 6 to the solution of Bayesian filtering and smoothing problems. The key result of this section is a new variational algorithm

11. This is not always the case. For instance, given a rooted version of $\boldsymbol{\mathcal{G}}$ and a pair of consecutive *depths* (see the discussion at the end of Section 6.3), all the maps (L_i) associated with edges connecting nodes at these two depths can be computed in parallel.

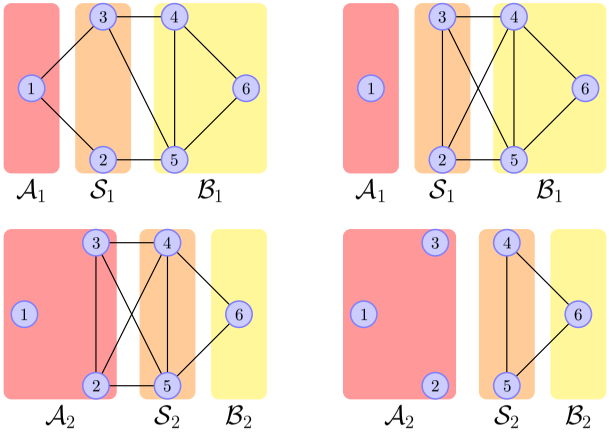


Figure 6: Sequence of graph decompositions associated with the recursive application of Lemma 8. On the *(top left)* there is an I-map, $\boldsymbol{\mathcal{G}}^1$, for $\boldsymbol{\nu}_\pi$, with $\boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^6)$. We first decompose this graph into $(\mathcal{A}_1, \mathcal{S}_1, \mathcal{B}_1)$ as indicated, and apply Theorem 7 to the pair $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi$. To do so, we first need to add edge $(2,3)$ to $\boldsymbol{\mathcal{G}}^1$ in order to turn $(\mathcal{A}_1, \mathcal{S}_1, \mathcal{B}_1)$ into a proper decomposition of $\boldsymbol{\mathcal{G}}^1$ with a fully connected $\mathcal{S}_1$. The resulting graph, $\boldsymbol{\mathcal{G}}^1_\star$, is now chordal (in fact, a triangulation of $\boldsymbol{\mathcal{G}}^1$, Lauritzen, 1996), but still an I-map for $\boldsymbol{\nu}_\pi$. The first map L_1 is σ_1-triangular with $\sigma_1(\mathbb{N}_6) = \{2,3,1,4,5,6\}$ and it is low-dimensional with respect to $\mathcal{B}_1$; The *(top right)* figure shows the I-map, $\boldsymbol{\mathcal{G}}^2$, for $L_1^\sharp \boldsymbol{\nu}_\pi$ as given by Theorem 7[Part 2d]: as expected, $\mathcal{A}_1$ is disconnected from $\mathcal{S}_1 \cup \mathcal{B}_1$; moreover, a new maximal clique $\{2,3,4,5\}$ appears in $\boldsymbol{\mathcal{G}}^2$. This new clique is larger than any of the maximal cliques in $\boldsymbol{\mathcal{G}}^1_\star$, even though $\boldsymbol{\mathcal{G}}^1_\star$ is chordal. (Notice that σ_1 is not the permutation that adds the fewest edges possible in $\boldsymbol{\mathcal{G}}^2$. An example of such "best" permutation would be $\sigma(\mathbb{N}_6) = \{3,2,1,4,5,6\}$.) Though Theorem 7 guarantees the existence of a low-dimensional map $R \in \mathfrak{R}_1$ that pushes forward $\boldsymbol{\nu}_\eta$ to $L_1^\sharp \boldsymbol{\nu}_\pi$, we instead proceed recursively by applying Lemma 8[Part 1] for a proper decomposition, $(\mathcal{A}_2, \mathcal{S}_2, \mathcal{B}_2)$, of $\boldsymbol{\mathcal{G}}^2$, where $\mathcal{A}_2$ is a strict superset of $\mathcal{A}_1$ *(bottom left)*. The lemma shows that $\mathfrak{R}_1 \supset L_2 \circ \mathfrak{R}_2$ for some σ_2-triangular map L_2, which is low-dimensional with respect to $\mathcal{A}_1 \cup \mathcal{B}_2$, and where each $R \in \mathfrak{R}_2$ pushes forward $\boldsymbol{\nu}_\eta$ to $(L_1 \circ L_2)^\sharp \boldsymbol{\nu}_\pi$. Can we apply Lemma 8 one more time to characterize decomposable transports in $\mathfrak{R}_2$? The answer is no, as the I-map for $(L_1 \circ L_2)^\sharp \boldsymbol{\nu}_\pi$ *(bottom right)* consists of a single clique in $\mathcal{S}_2 \cup \mathcal{B}_2$. Nevertheless, each $R \in \mathfrak{R}_2$ is still low-dimensional with respect to $\mathcal{A}_2$. Overall, we showed the existence of a transport map $T : \mathbb{R}^6 \to \mathbb{R}^6$ pushing forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ that decomposes as $T = L_1 \circ L_2 \circ R$, where L_1, L_2, R are effectively $\{3,4,3\}$-dimensional maps, respectively.

for characterizing the full posterior distribution of the sequential inference problem—e.g., not just a few filtering or smoothing marginals—via recursive lag–1 smoothing with transport maps. The proposed algorithm builds a decomposable high-dimensional transport map in a *single forward pass* by solving a sequence of local low-dimensional problems, without resorting to any backward pass on the state space model (see Theorem 9). These results extend naturally to the case of joint parameter and state estimation (see Section 7.3 and Theorem 12). Pseudocode for the algorithm is given in Appendix C.

A state-space model consists of a pair of discrete-time stochastic processes $(\boldsymbol{Z}_k, \boldsymbol{Y}_k)_{k\geq 0}$ indexed by the time k, where $(\boldsymbol{Z}_k)$ is a latent Markov process of interest and where $(\boldsymbol{Y}_k)$ is the observed process. We can think of each $\boldsymbol{Y}_k$ as a noisy and perhaps indirect measurement of $\boldsymbol{Z}_k$. The Markov structure corresponding to the joint process $(\boldsymbol{Z}_k, \boldsymbol{Y}_k)$ is shown in Figure 7. The generalization of the results of this section to the case of missing observations is straightforward and will not be addressed here.

We assume that we are given the transition densities $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k}$ for all $k \geq 0$, sometimes referred to as the "prior dynamic," together with the marginal density of the initial conditions $\pi_{\boldsymbol{Z}_0}$. (For instance, the prior dynamic could stem from the discretization of a continuous time stochastic differential equation; Oksendal, 2013.) We denote by $\pi_{\boldsymbol{Y}_k|\boldsymbol{Z}_k}$ the likelihood function, i.e., the density of $\boldsymbol{Y}_k$ given $\boldsymbol{Z}_k$, and assume that $\boldsymbol{Z}_k$ and $\boldsymbol{Y}_k$ are random variables taking values on $\mathbb{R}^n$ and $\mathbb{R}^d$, respectively. Moreover, we denote by $(\boldsymbol{y}_k)_{k\geq 0}$ a sequence of realizations of the observed process $(\boldsymbol{Y}_k)$ that will define the posterior distribution over the unobserved (hidden) states of the model, and make the following regularity assumption in our theorems: $\pi_{\boldsymbol{Z}_{0:k-1},\boldsymbol{Y}_{0:k-1}} > 0$ for all $k \geq 1$. (The existence of underlying fully supported measures will be left implicit throughout the section for notational convenience.)

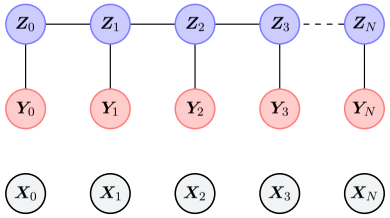


Figure 7: *(above)* I-map for the joint process $(\boldsymbol{Z}_k, \boldsymbol{Y}_k)_{k\geq 0}$ defining the state-space model. *(below)* I-map for the independent reference process $(\boldsymbol{X}_k)_{k\geq 0}$ used in Theorem 9.

7.1. Smoothing and Filtering: the Full Bayesian Solution

In typical applications of state-space modeling, the process $(\boldsymbol{Y}_k)$ is only observed sequentially, and thus the goal of inference is to characterize—sequentially in time and via a recursive algorithm—the joint distribution of the current and past states given currently available measurements, i.e.,

$$\pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}(\boldsymbol{z}_{0:k}) := \pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{Y}_{0:k}}(\boldsymbol{z}_{0:k}|\boldsymbol{y}_{0:k}) \tag{22}$$

for all $k \geq 0$. That is, we wish to characterize $\pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}$ based on our knowledge of the posterior distribution at the previous timestep, $\pi_{\boldsymbol{Z}_{0:k-1}|\boldsymbol{y}_{0:k-1}}$, and with an effort that is constant over time. We regard (22) as the full Bayesian solution to the sequential inference problem (Särkkä, 2013).

Usually, the task of updating $\pi_{\boldsymbol{Z}_{0:k-1}|\boldsymbol{y}_{0:k-1}}$ to yield $\pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}$ becomes increasingly challenging over time due to the widening inference horizon, making characterization of the full Bayesian solution impractical for large k. Thus, two simplifications of the sequential inference problem are frequently considered: filtering and smoothing (Särkkä, 2013). In filtering, we characterize $\pi_{\boldsymbol{Z}_k|\boldsymbol{y}_{0:k}}$ for all $k \geq 0$, while in smoothing we recursively update $\pi_{\boldsymbol{Z}_j|\boldsymbol{y}_{0:k}}$ for increasing $k > j$, where $\boldsymbol{Z}_j$ is some past state of the unobserved process. Both filtering and smoothing deliver particular low-dimensional *marginals* of the full Bayesian solution to the inference problem, and hence are often considered good candidates for numerical approximation (Doucet and Johansen, 2009).

The following theorem shows that characterizing the full Bayesian solution to the sequential inference problem via a decomposable transport map is essentially no harder than performing lag–1 smoothing, which, in turn, amounts to characterizing $\pi_{\boldsymbol{Z}_{k-1},\boldsymbol{Z}_k|\boldsymbol{y}_{0:k}}$ for all $k \geq 0$ (an operation only slightly harder than regular filtering). This result relies on the recursive application of the decomposition theorem for couplings (Theorem 7) to the *tree* Markov structure of $\pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}$. In what follows, let $(\boldsymbol{X}_k)_{k\geq 0}$ be an independent (reference) process with nonvanishing marginal densities $(\eta_{\boldsymbol{X}_k})$, with each $\boldsymbol{X}_k$ taking values on $\mathbb{R}^n$. See Figure 7 for the corresponding Markov network.

Theorem 9 (Decomposition theorem for state-space models) *Let $(\mathfrak{M}_i)_{i\geq 0}$ be a sequence of (σ_i)-generalized KR rearrangements on $\mathbb{R}^n \times \mathbb{R}^n$, which are of the form*

$$\mathfrak{M}_i(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) = \begin{bmatrix} \mathfrak{M}_i^0(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \\ \mathfrak{M}_i^1(\boldsymbol{x}_{i+1}) \end{bmatrix} \tag{23}$$

for some σ_i, $\mathfrak{M}_i^0 : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, $\mathfrak{M}_i^1 : \mathbb{R}^n \to \mathbb{R}^n$, and that are defined by the recursion:

- *$\mathfrak{M}_0$ pushes forward $\eta_{\boldsymbol{X}_0,\boldsymbol{X}_1}$ to $\pi^0 = \widetilde{\pi}^0/\mathfrak{c}_0$,*
- *$\mathfrak{M}_i$ pushes forward $\eta_{\boldsymbol{X}_i,\boldsymbol{X}_{i+1}}$ to $\pi^i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1}) = \eta_{\boldsymbol{X}_i}(\boldsymbol{z}_i)\, \widetilde{\pi}^i(\mathfrak{M}_{i-1}^1(\boldsymbol{z}_i), \boldsymbol{z}_{i+1})/\mathfrak{c}_i$,*

where $\mathfrak{c}_i$ is a normalizing constant and where $(\widetilde{\pi}^i)_{i\geq 0}$ are functions on $\mathbb{R}^n \times \mathbb{R}^n$ given by:

- *$\widetilde{\pi}^0(\boldsymbol{z}_0, \boldsymbol{z}_1) = \pi_{\boldsymbol{Z}_0,\boldsymbol{Z}_1}(\boldsymbol{z}_0, \boldsymbol{z}_1)\, \pi_{\boldsymbol{Y}_0|\boldsymbol{Z}_0}(\boldsymbol{y}_0|\boldsymbol{z}_0)\, \pi_{\boldsymbol{Y}_1|\boldsymbol{Z}_1}(\boldsymbol{y}_1|\boldsymbol{z}_1)$,*
- *$\widetilde{\pi}^i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1}) = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{Z}_i}(\boldsymbol{z}_{i+1}|\boldsymbol{z}_i)\, \pi_{\boldsymbol{Y}_{i+1}|\boldsymbol{Z}_{i+1}}(\boldsymbol{y}_{i+1}|\boldsymbol{z}_{i+1})$ for $i \geq 1$.*

Then, for all $k \geq 0$, the following hold:

1. *The map $\mathfrak{M}_k^1$ pushes forward $\eta_{\boldsymbol{X}_{k+1}}$ to $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$.* *[filtering]*
2. *The map $\overline{\mathfrak{M}}_k$, defined as ($\mathfrak{M}_{k-1}^1(\boldsymbol{x}) = \boldsymbol{x}$ for $k = 0$)*

$$\overline{\mathfrak{M}}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}) = \begin{bmatrix} \mathfrak{M}_{k-1}^1(\mathfrak{M}_k^0(\boldsymbol{x}_k, \boldsymbol{x}_{k+1})) \\ \mathfrak{M}_k^1(\boldsymbol{x}_{k+1}) \end{bmatrix}, \tag{24}$$

pushes forward $\eta_{\boldsymbol{X}_k,\boldsymbol{X}_{k+1}}$ to $\pi_{\boldsymbol{Z}_k,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$. *[lag–1 smoothing]*

3. The composition of transport maps $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$, where each T_i is defined as

$$T_i(\boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k+1}) = \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_{i-1} \\ \mathfrak{M}_i^0(\boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \\ \mathfrak{M}_i^1(\boldsymbol{x}_{i+1}) \\ \boldsymbol{x}_{i+2} \\ \vdots \\ \boldsymbol{x}_{k+1} \end{bmatrix}, \tag{25}$$

pushes forward $\eta_{\boldsymbol{X}_{0:k+1}}$ to $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$. *[full Bayesian solution]*

4. The model evidence (marginal likelihood) is given by

$$\pi_{\boldsymbol{Y}_{0:k+1}}(\boldsymbol{y}_{0:k+1}) = \prod_{i=0}^{k} \mathfrak{c}_i. \tag{26}$$

Theorem 9 suggests a variational algorithm for smoothing and filtering a continuous state-space model: compute the sequence of maps $(\mathfrak{M}_i)$, each of dimension $2n$; embed them into higher-dimensional identity maps to form (T_i) according to (25); then evaluate the composition $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ to sample directly from $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ (i.e., the full Bayesian solution) and obtain information about any smoothing or filtering distribution of interest.

Successive transports in the composition $(\mathfrak{T}_k)_{k\geq 0}$ are *nested* and thus ideal for sequential assimilation: given $\mathfrak{T}_{k-1}$, we can obtain $\mathfrak{T}_k$ simply by computing an additional map $\mathfrak{M}_k$ of dimension $2n$—with no need to recompute $(\mathfrak{M}_i)_{i<k}$. This step converts a transport map that samples $\pi_{\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}$ into one that samples $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$. This feature is important since $\mathfrak{M}_k$ is always a $2n$-dimensional map, while $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ is a density on $\mathbb{R}^{n(k+2)}$—a space whose dimension increases with time k. In fact, from the perspective of Section 6, Theorem 9 simply shows that each $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ can be represented via a *decomposable* transport $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$. The sparsity pattern of each map $\mathfrak{M}_i$, specified in (23), is necessary for Theorem 9 to hold: $\mathfrak{M}_i$ cannot be *any* transport map; it must be block upper triangular.

The proposed algorithm consists of a forward pass on the state-space model—wherein the sequence of transport maps $(\mathfrak{M}_i)$ are computed and stored—followed by a backward pass where the composition $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ is evaluated deterministically to sample $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$. This backward pass does not re-evaluate the potentials of the state-space model (e.g., transition kernels or likelihoods) at earlier times, nor does it perform any additional computation other than evaluating the maps $(\mathfrak{M}_i)$ in $\mathfrak{T}_k$.

Though each map T_j is usually trivial to evaluate—e.g., the map might be parameterized in terms of polynomials (Marzouk et al., 2016) and differ from the identity along only $2n$ components—it is true that the cost of evaluating $\mathfrak{T}_k$ grows linearly with k. This is hardly surprising since $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ is a density over spaces of increasing dimension. A direct approximation of $\mathfrak{T}_k$ is usually a bad idea since the map is high-dimensional and dense (in the sense defined by Section 6); it is better to store $\mathfrak{T}_k$ implicitly through the sequence of maps $(\mathfrak{M}_i)_{i\geq 0}^k$, and sample smoothed trajectories by evaluating $\mathfrak{T}_k$ only when it is needed. If

we are only interested in a particular smoothing marginal, e.g., $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k+1}}$ for all $k \geq 0$, then we can define a general forward recursion to sample $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k+1}}$ with a *single* transport map that is updated recursively over time, rather than with a growing composition of maps—and thus with a cost independent of k. This construction is given in Section 7.4.

Also, it is important to emphasize that in order to assimilate a new measurement, say $\boldsymbol{y}_{k+1}$, we do *not* need to evaluate the full composition $\mathfrak{T}_{k-1}$; we only need to compute a low-dimensional map $\mathfrak{M}_k$ whose target density π^k depends only on $\mathfrak{M}_{k-1}$. The previous maps $(\mathfrak{M}_i)_{i<k-1}$ are unnecessary at this stage. Thus the effort of assimilating a new piece of data is constant in time—modulo the complexity of each $\mathfrak{M}_k$.

The distribution $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ is not represented via a collection of particles as $k \geq 0$ increases, but rather via a growing composition of low-dimensional transport maps that yields *fully supported* approximations of $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$. These maps are computed via deterministic optimization: there are no importance sampling or resampling steps. Intuitively, the optimization step for $\mathfrak{M}_k$ *moves* the particles on which the map is evaluated, rather than reweighing them.

Part 1 of Theorem 9 shows that the lower subcomponent $\mathfrak{M}_k^1 : \mathbb{R}^n \to \mathbb{R}^n$ of the map $\mathfrak{M}_k$ characterizes the filtering distribution $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ for all $k \geq 0$, while Part 2 shows that each $\mathfrak{M}_k$ also characterizes the lag–1 smoothing distribution $\pi_{\boldsymbol{Z}_k,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ up to an invertible transformation of the marginal over $\boldsymbol{Z}_k$. Thus, Theorem 9 implies a deterministic algorithm for lag–1 smoothing that in fact fully characterizes the posterior distribution of the nonlinear state-space model—much in the same spirit as the Rauch-Tung-Striebel (RTS) smoothing algorithm for Gaussian models. We clarify this connection in Section 7.2.

A related perspective on the proposed smoothing algorithm is that the composition of maps $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ implements the following factorization of the full Bayesian solution,

$$\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}} = \pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}\, \pi_{\boldsymbol{Z}_k|\boldsymbol{Z}_{k+1},\boldsymbol{y}_{0:k}}\, \pi_{\boldsymbol{Z}_{k-1}|\boldsymbol{Z}_k,\boldsymbol{y}_{0:k-1}} \cdots \pi_{\boldsymbol{Z}_0|\boldsymbol{Z}_1,\boldsymbol{y}_0}, \tag{27}$$

wherein each map $\mathfrak{M}_i$, due to its block *upper* triangular structure, is associated with a specific factorization of the lag–1 smoothing density,

$$\pi_{\boldsymbol{Z}_{i+1},\boldsymbol{Z}_i|\boldsymbol{y}_{0:i+1}} = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}\, \pi_{\boldsymbol{Z}_i|\boldsymbol{Z}_{i+1},\boldsymbol{y}_{0:i}}.$$

Evaluating $\mathfrak{T}_k$ on samples drawn from the reference process $\eta_{\boldsymbol{X}_{0:k+1}}$ amounts to sampling first from the final filtering marginal $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ and then from the sequence of "backward" conditionals in (27). See also (Kitagawa, 1987; Doucet and Johansen, 2009; Godsill et al., 2004) for alternative approximations of the forward–filtering backward–smoothing formulas.

Note that the proposed approach does *not* reduce to the ensemble Kalman filter (EnKF) or to the ensemble Kalman smoother (EnKS) (Evensen, 2003; Evensen and Van Leeuwen, 2000), even if the maps $\{\mathfrak{M}_k\}$ are constrained to be linear. For one, the EnKF implements a two-step recursive approximation of each filtering marginal, which consists of (i) a particle approximation of the "forecast" distribution $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k}}$ obtained by *simulating* the transition kernel $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k}$, followed by (ii) a *linear* approximation of the forecast-to-analysis update (i.e., the update from $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k}}$ to $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$). In contrast, our approach constructs a recursive *variational* approximation of each lag–1 smoothing distribution, essentially using numerical optimization to minimize the KL divergence between $\pi_{\boldsymbol{Z}_k,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ and its transport map approximation. We do not make a particle approximation of the forecast

distribution by integrating the model dynamics, but instead require explicit evaluations of the transition density $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k}$. If, however, the dynamics of the state-space model are linear, with Gaussian transition/observational noise and Gaussian initial conditions, then the proposed algorithm is equivalent to filtering and smoothing via "exact" Kalman formulas; in this case, the EnKF and EnKS can be interpreted as Monte Carlo approximations of the recursions defined by the proposed algorithm (Raanes, 2016).

Numerical approximations. In general, the maps $(\mathfrak{M}_i)$ must be approximated numerically (see Section 3). As a result, Monte Carlo estimators associated with the evaluation of $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ are *biased*, although possibly with negligible variance, since it is trivial to evaluate the map a large number of times. This bias is *only* due to the numerical approximation of $(\mathfrak{M}_i)$, and not to the particular factorization properties of $\mathfrak{T}_k$. In practice, one might either accept this bias or try to reduce it. The bias can be reduced in at least two ways: (1) by enriching the parameterization of some $(\mathfrak{M}_i)$, and thus increasing the accuracy of the variational approximation, or (2) by using the map-induced proposal density $(\mathfrak{T}_k)_\sharp \eta_{\boldsymbol{X}_{0:k+1}}$—i.e., the pushforward of a marginal of the reference process through $\mathfrak{T}_k$—within importance sampling or MCMC (see Section 8). For instance, the weight function

$$w^{k+1}(\boldsymbol{x}) = \frac{\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}(\boldsymbol{x})}{(\mathfrak{T}_k)_\sharp \eta_{\boldsymbol{X}_{0:k+1}}(\boldsymbol{x})}$$

is readily available, and can be used to yield consistent estimators with respect to the smoothing distribution. However, the resulting weights cannot be computed recursively in time, because even though the small dimensional maps $\mathfrak{M}_k$ are computed sequentially, the map-induced proposal $(\mathfrak{T}_k)_\sharp \eta_{\boldsymbol{X}_{0:k+1}}$ changes entirely at every step.

In particle filters, the complexity of approximating the underlying distribution is given by the number of particles N. In the proposed variational approach, the complexity of the approximation depends on the parameterization of each map $\mathfrak{M}_i$. There is no single parameter like N to describe the complexity of the latter—though, broadly, it should depend on the number of degrees of freedom in the parameterization. In some cases, one might think of using the total order of a multivariate polynomial expansion of each component of the map as a tuning parameter. But this is far from general or practical in high dimensions. The virtue of a functional representation of the transport map is the ability to carefully select the degrees of freedom of the parameterization. For instance, we might model local interactions between different groups of input variables using different approximation orders or even different sets of basis functions. This freedom should not be frightening, but rather embraced as a rich opportunity to exploit the structure of the particular problem at hand. Spantini (2017, Ch. 6) gives an example of this practice in the context of filtering high-dimensional spatiotemporal processes with chaotic dynamics.

In general, richer parameterizations of the maps are more costly to characterize because they lead to higher-dimensional optimization problems (7). Yet, richer parameterizations can yield arbitrarily accurate results. There is clearly a tradeoff between computational cost and statistical accuracy. We investigate this tradeoff numerically in Section 8, where we report the cost of computing a transport map under different parameterizations and inference scenarios.

Another important note: the *sequential* approximation of the individual maps $(\mathfrak{M}_i)$ might present additional challenges due to the accumulation of error, since the target density for the k-th map $\mathfrak{M}_k$ depends on the numerical approximation of the previous map, $\mathfrak{M}_{k-1}$. This is not an issue with the factorization of $\mathfrak{T}_k$ per se, but rather with sequentially computing each element of the factorization. The analysis of sequential Monte Carlo methods (e.g., Crisan and Doucet, 2002; Del Moral, 2004; Smith et al., 2013) addresses a similar accumulation of error, but has not yet been extended to sequential variational inference techniques. In Section 8, we empirically investigate the stability of variational transport map approximations for a problem of very long time smoothing (see Figure 17), showing excellent results—at least for the reconstruction of low-order smoothing marginals.

As shown in (9), the computation of each $\mathfrak{M}_i$ is also associated with an approximation of the normalizing constant $\mathfrak{c}_i$ of its own target density, which then leads to a one-pass approximation of the marginal likelihood using (26).

One last remark: the proof of Theorem 9 shows that the triangular structure hypothesis for each $\mathfrak{M}_i$ can be relaxed provided that the underlying densities are regular enough. The following corollary clarifies this point.

Corollary 10 *The results of Theorem 9 still hold if we replace every KR rearrangement $\mathfrak{M}_i$ with a "block triangular" diffeomorphism of the form* (23) *that couples the same distributions, provided that such regular transport maps exist.*

Filtering and smoothing are of course very rich problems, and in this section we have by no means attempted to be exhaustive. Rather, our goal was to highlight some implications of decomposable transports on problems of sequential Bayesian inference, in a general non-Gaussian setting.

7.2. The Linear Gaussian Case: Connection with the RTS Smoother

In this section, we specialize the results of Theorem 9 to linear Gaussian state-space models, and make explicit the connection with the RTS Gaussian smoother (Rauch et al., 1965).

Consider a linear Gaussian state-space model defined by

$$\begin{aligned} \boldsymbol{Z}_{k+1} &= \boldsymbol{F}_k \boldsymbol{Z}_k + \boldsymbol{\varepsilon}_k \\ \boldsymbol{Y}_k &= \boldsymbol{H}_k \boldsymbol{Z}_k + \boldsymbol{\xi}_k \end{aligned}$$

for all $k \geq 0$, where $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, \boldsymbol{Q}_k)$, $\boldsymbol{\xi}_k \sim \mathcal{N}(0, \boldsymbol{R}_k)$, $\boldsymbol{F}_k \in \mathbb{R}^{n\times n}$, $\boldsymbol{H}_k \in \mathbb{R}^{d\times n}$, and $\boldsymbol{Z}_0 \sim \mathcal{N}(\mu_0, \boldsymbol{\Gamma}_0)$. Both $\boldsymbol{\varepsilon}_k$ and $\boldsymbol{\xi}_k$ are independent of $\boldsymbol{Z}_k$, while $\boldsymbol{Q}_k, \boldsymbol{R}_k$, and $\boldsymbol{\Gamma}_0$ are symmetric positive definite matrices for all $k \geq 0$.

If we choose an independent reference process $(\boldsymbol{X}_k)$ with standard normal marginals, i.e., $\eta_{\boldsymbol{X}_k} = \mathcal{N}(0, \mathbf{I})$, then the maps $(\mathfrak{M}_k)$ of Theorem 9 can be chosen to be linear:

$$\mathfrak{M}_k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1}) = \begin{bmatrix} \boldsymbol{A}_k & \boldsymbol{B}_k \\ \mathbf{0} & \boldsymbol{C}_k \end{bmatrix} \begin{Bmatrix} \boldsymbol{z}_k \\ \boldsymbol{z}_{k+1} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{a}_k \\ \boldsymbol{c}_k \end{Bmatrix}, \tag{28}$$

for some matrices $\boldsymbol{A}_k, \boldsymbol{B}_k, \boldsymbol{C}_k \in \mathbb{R}^{n\times n}$ and $\boldsymbol{a}_k, \boldsymbol{c}_k \in \mathbb{R}^n$. (Notice that in this case Corollary 10 applies and the matrices $\boldsymbol{A}_k, \boldsymbol{B}_k$ can be full and not necessarily triangular.) The following lemma gives a closed form expression for the maps $(\mathfrak{M}_k)$ with $k \geq 1$. ($\mathfrak{M}_0$ can be derived analogously with simple algebra.)

Lemma 11 (The linear Gaussian case) *For $k \geq 1$, the map $\mathfrak{M}_k$ in (28) can be defined as follows: if $(\boldsymbol{c}_k, \boldsymbol{C}_k)$ is the output of a square-root Kalman filter at time k (Bierman, 2006), i.e., if $\boldsymbol{c}_k$ and $\boldsymbol{C}_k$ are, respectively, the mean and square root of the covariance of the filtering distribution $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$, then one can set:*

$$\begin{aligned} \boldsymbol{A}_k &= \boldsymbol{J}_k^{-1/2} \\ \boldsymbol{B}_k &= -\boldsymbol{J}_k^{-1}\,\boldsymbol{P}_k\,\boldsymbol{C}_k \\ \boldsymbol{a}_k &= \boldsymbol{J}_k^{-1}\,\boldsymbol{P}_k\,(\boldsymbol{F}_k\,\boldsymbol{c}_{k-1} - \boldsymbol{c}_k), \end{aligned} \tag{29}$$

for $\boldsymbol{J}_k \coloneqq \mathbf{I} + \boldsymbol{C}_{k-1}^\top\,\boldsymbol{F}_k^\top\,\boldsymbol{Q}_k^{-1}\,\boldsymbol{F}_k\,\boldsymbol{C}_{k-1}$ and $\boldsymbol{P}_k = -\boldsymbol{C}_{k-1}^\top\,\boldsymbol{F}_k^\top\,\boldsymbol{Q}_k^{-1}$.

The formulas in Lemma 11 can be interpreted as one possible implementation of a square-root RTS smoother for Gaussian models: at each step k of a forward pass, the filtering estimates $(\boldsymbol{c}_k, \boldsymbol{C}_k)$ are augmented with a collection $(\boldsymbol{a}_k, \boldsymbol{A}_k, \boldsymbol{B}_k)$ of stored quantities, which can then be reused to sample the full Bayesian solution (or particular smoothing marginals) whenever needed, and without ever touching the state-space model again. In this sense, the algorithm proposed in Section 7.1 can be understood as the natural generalization—to the non-Gaussian case—of the square-root RTS smoother.

7.3. Sequential Joint Parameter and State Estimation

In defining a state-space model, it is common to parameterize the transition densities of the unobserved process or the likelihoods of the observables in terms of some hyperparameters $\boldsymbol{\Theta}$. The Markov structure of the resulting Bayesian hierarchical model, conditioned on the data, is shown in Figure 8. The state-space model is now fully specified in terms of the conditional densities $(\pi_{\boldsymbol{Y}_k|\boldsymbol{Z}_k,\boldsymbol{\Theta}})_{k\geq 0}$, $(\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k,\boldsymbol{\Theta}})_{k\geq 0}$, $\pi_{\boldsymbol{Z}_0|\boldsymbol{\Theta}}$, and the marginal $\pi_{\boldsymbol{\Theta}}$. We assume that the hyperparameters $\boldsymbol{\Theta}$ take values on $\mathbb{R}^p$, and that the following regularity conditions hold: $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k-1},\boldsymbol{Y}_{0:k-1}} > 0$ for all $k \geq 1$.

Given such a parameterization, one often wishes to *jointly* infer the hidden states and the hyperparameters of the model as observations of the process $(\boldsymbol{Y}_k)$ become available. That is, the goal of inference is to characterize, via a *recursive* algorithm, the sequence of posterior distributions given by

$$\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{0:k}) \coloneqq \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k}|\boldsymbol{Y}_{0:k}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{0:k}|\boldsymbol{y}_{0:k}) \tag{30}$$

for all $k \geq 0$ and for a sequence $(\boldsymbol{y}_k)_{k\geq 0}$ of observations. The following theorem shows that we can characterize (30) by computing a sequence of low-dimensional transport maps in the same spirit as Theorem 9. In what follows, let $(\boldsymbol{X}_k)$ be an independent process with marginals $(\eta_{\boldsymbol{X}_k})$ as defined in Theorem 9 and let $\boldsymbol{X}_{\boldsymbol{\Theta}}$ be a random variable on $\mathbb{R}^p$ that is independent of $(\boldsymbol{X}_k)$ and with nonvanishing density $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}}$.

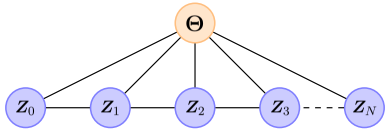


Figure 8: I-map for $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\ldots,\boldsymbol{Z}_N|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_N}$, for any $N > 0$.

Theorem 12 (Decomposition theorem for joint parameter and state estimation) *Let $(\mathfrak{M}_i)_{i\geq 0}$ be a sequence of (σ_i)-generalized KR rearrangements on $\mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^n$, which are of the form*

$$\mathfrak{M}_i(\boldsymbol{x}_\theta, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}) = \begin{bmatrix} \mathfrak{M}_i^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta) \\ \mathfrak{M}_i^0(\boldsymbol{x}_\theta, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \\ \mathfrak{M}_i^1(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) \end{bmatrix} \tag{31}$$

for some σ_i, $\mathfrak{M}_i^{\boldsymbol{\Theta}} : \mathbb{R}^p \to \mathbb{R}^p$, $\mathfrak{M}_i^0 : \mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n$, $\mathfrak{M}_i^1 : \mathbb{R}^p \times \mathbb{R}^n \to \mathbb{R}^n$, and that are defined by the recursion:

- *$\mathfrak{M}_0$ pushes forward $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}},\boldsymbol{X}_0,\boldsymbol{X}_1}$ to*

$$\pi^0 = \widetilde{\pi}^0/\mathfrak{c}_0, \tag{32}$$

- *$\mathfrak{M}_i$ pushes forward $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}},\boldsymbol{X}_i,\boldsymbol{X}_{i+1}}$ to*

$$\pi^i(\boldsymbol{z}_\theta, \boldsymbol{z}_i, \boldsymbol{z}_{i+1}) = \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}},\boldsymbol{X}_i}(\boldsymbol{z}_\theta, \boldsymbol{z}_i)\,\widetilde{\pi}^i(\mathfrak{T}_{i-1}^{\boldsymbol{\Theta}}(\boldsymbol{z}_\theta),\, \mathfrak{M}_{i-1}^1(\boldsymbol{z}_\theta, \boldsymbol{z}_i), \boldsymbol{z}_{i+1})/\mathfrak{c}_i, \tag{33}$$

where $\mathfrak{c}_i$ is a normalizing constant, the map $\mathfrak{T}_j^{\boldsymbol{\Theta}} \coloneqq \mathfrak{M}_0^{\boldsymbol{\Theta}} \circ \cdots \circ \mathfrak{M}_j^{\boldsymbol{\Theta}}$ for all $j \geq 0$, and where $(\widetilde{\pi}^i)_{i\geq 0}$ are functions on $\mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^n$ given by:

- *$\widetilde{\pi}^0(\boldsymbol{z}_\theta, \boldsymbol{z}_0, \boldsymbol{z}_1) = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\boldsymbol{Z}_1}(\boldsymbol{z}_\theta, \boldsymbol{z}_0, \boldsymbol{z}_1)\,\pi_{\boldsymbol{Y}_0|\boldsymbol{Z}_0,\boldsymbol{\Theta}}(\boldsymbol{y}_0|\boldsymbol{z}_0, \boldsymbol{z}_\theta)\,\pi_{\boldsymbol{Y}_1|\boldsymbol{Z}_1,\boldsymbol{\Theta}}(\boldsymbol{y}_1|\boldsymbol{z}_1, \boldsymbol{z}_\theta)$,*
- *$\widetilde{\pi}^i(\boldsymbol{z}_\theta, \boldsymbol{z}_i, \boldsymbol{z}_{i+1}) = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{Z}_i,\boldsymbol{\Theta}}(\boldsymbol{z}_{i+1}|\boldsymbol{z}_i, \boldsymbol{z}_\theta)\,\pi_{\boldsymbol{Y}_{i+1}|\boldsymbol{Z}_{i+1},\boldsymbol{\Theta}}(\boldsymbol{y}_{i+1}|\boldsymbol{z}_{i+1}, \boldsymbol{z}_\theta)$ for $i \geq 1$.*

Then, for all $k \geq 0$, the following hold:

1. *The map $\widetilde{\mathfrak{M}}_k$, defined as*

$$\widetilde{\mathfrak{M}}_k(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1}) = \begin{bmatrix} \mathfrak{T}_k^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta) \\ \mathfrak{M}_k^1(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1}) \end{bmatrix}, \tag{34}$$

   pushes forward $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}},\boldsymbol{X}_{k+1}}$ to $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{k+1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{k+1}}$. *[filtering]*

2. *The composition of transport maps $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$, where each T_i is defined as*

$$T_i(\boldsymbol{x}_\theta, \boldsymbol{x}_0, \ldots, \boldsymbol{x}_{k+1}) = \begin{bmatrix} \mathfrak{M}_i^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta) \\ \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_{i-1} \\ \mathfrak{M}_i^0(\boldsymbol{x}_\theta, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \\ \mathfrak{M}_i^1(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) \\ \boldsymbol{x}_{i+2} \\ \vdots \\ \boldsymbol{x}_{k+1} \end{bmatrix}, \tag{35}$$

   pushes forward $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}},\boldsymbol{X}_0,\ldots,\boldsymbol{X}_{k+1}}$ to $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\ldots,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{k+1}}$. *[full Bayesian solution]*

3. The model evidence (marginal likelihood) is given by (26).

Theorem 12 suggests a variational algorithm for the joint parameter and state estimation problem that is similar to the one proposed in Theorem 9: compute the sequence of maps $(\mathfrak{M}_i)$, each of dimension $2n+p$; embed them into higher-dimensional identity maps to form (T_i) according to (35); then evaluate the composition $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ to sample directly from $\pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ (i.e., the full Bayesian solution). See Appendix C for more details. Each map $\mathfrak{M}_i$ is now of dimension twice that of the model state plus the dimension of the hyperparameters. This dimension is slightly higher than that of the maps $(\mathfrak{M}_i)$ considered in Theorem 9, and should be regarded as the price to pay for introducing hyperparameters in the state-space model and having to deal with the Markov structure of Figure 8 as opposed to the tree structure of Figure 7. By Theorem 12[Part 1], the composition of maps $\mathfrak{T}_k^{\boldsymbol{\Theta}} = \mathfrak{M}_0^{\boldsymbol{\Theta}} \circ \cdots \circ \mathfrak{M}_k^{\boldsymbol{\Theta}}$ provides a recursive characterization of the posterior distribution over the static parameters, $\pi_{\boldsymbol{\Theta}|\boldsymbol{y}_{0:k+1}}$, for all $k \geq 0$. The latter is often the ultimate goal of inference (Andrieu et al., 2010). In order to have a sequential algorithm for parameter estimation, we also need to keep a running approximation of $\mathfrak{T}_k^{\boldsymbol{\Theta}}$ using the recursion $\mathfrak{T}_k^{\boldsymbol{\Theta}} = \mathfrak{T}_{k-1}^{\boldsymbol{\Theta}} \circ \mathfrak{M}_k^{\boldsymbol{\Theta}}$—e.g., via regression—so that the cost of evaluating $\mathfrak{T}_k^{\boldsymbol{\Theta}}$ does not grow with k.

Even in the joint parameter and state estimation case, only a single forward pass with local computations is necessary to gather all the information from the state-space model needed to sample the collection of posteriors $(\pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}})$. Notice that the accuracy of the variational procedure is only limited by the accuracy of each computed map, and that the proposed approach does not prescribe an artificial dynamic for the parameters (Kitagawa, 1998; Liu and West, 2001), or an *a priori* fixed-lag smoothing approximation (Polson et al., 2008). Yet there is no rigorous proof that the performance of the proposed sequential algorithm for parameter estimation does not deteriorate with time. Indeed, developing exact, sequential, and online algorithms for parameter estimation in general non-Gaussian state-space models is among the chief research challenges in SMC methods (Jacob, 2015). See (Chopin et al., 2013; Crisan and Miguez, 2013; Del Moral et al., 2017) for recent contributions in this direction and (Kantas et al., 2015) for a review of SMC approaches to Bayesian parameter inference. See also (Erol et al., 2017) for a hybrid approach that combines elements of variational inference with particle filters.

We refer the reader to Section 8 for a numerical illustration of parameter inference with transport maps involving a stochastic volatility model.

7.4. Fixed-Point Smoothing

Consider again the problem of sequential inference in a state-space model without static parameters (see Figure 7), and suppose that we are interested only in the smoothing marginal $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k}}$ for all $k \geq 0$; this is the fixed-point smoothing problem.

In Section 7.1 we showed that computing a sequence of maps $(\mathfrak{M}_i)$—each of dimension $2n$—is sufficient to sample the joint distribution $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ by evaluating the composition $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$, where each T_i is a trivial embedding of $\mathfrak{M}_i$ into an identity map. If we can sample $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$, then it is easy to obtain samples from the marginal $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k+1}}$: in fact, it suffices to evaluate only the first n components of $\mathfrak{T}_k$, which can be interpreted as a map from $\mathbb{R}^{n\times(k+2)}$ to $\mathbb{R}^n$. To do so, however, we need to evaluate k maps. A natural

question then is whether it is possible to characterize $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k+1}}$ via a *single* transport map that is updated recursively in time, as opposed to a growing composition of maps.

Here we propose a solution—certainly not the only possibility—based on the theory of Section 7.3. The idea is to treat $\boldsymbol{Z}_0$ as a static parameter, i.e., to set $\boldsymbol{\Theta} \coloneqq \boldsymbol{Z}_0$ and apply the results of Theorem 12 to the Markov structure of Figure 9. The resulting algorithm computes a sequence of maps $(\mathfrak{M}_i)$ of dimension $3n$, i.e., *three* times the state dimension, and keeps a running approximation of $\mathfrak{T}_k^{\boldsymbol{\Theta}}$ via the recursion $\mathfrak{T}_k^{\boldsymbol{\Theta}} = \mathfrak{T}_{k-1}^{\boldsymbol{\Theta}} \circ \mathfrak{M}_k^{\boldsymbol{\Theta}}$, where each $\mathfrak{M}_k^{\boldsymbol{\Theta}}$ is just a subcomponent of $\mathfrak{M}_k$. These maps $(\mathfrak{M}_i)$ are higher-dimensional than those considered in Section 7.1, but they do yield the desired result: each $\mathfrak{T}_k^{\boldsymbol{\Theta}} : \mathbb{R}^n \to \mathbb{R}^n$ characterizes the smoothing marginal $\pi_{\boldsymbol{Z}_0|\boldsymbol{y}_{0:k+1}}$, for all $k \geq 0$, via a single transport map that is updated recursively in time with just one forward pass (see Theorem 12[Part 1]).

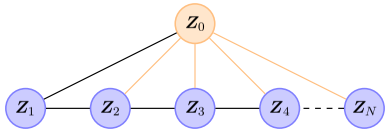


Figure 9: I-map (certainly not minimal) for $\pi_{\boldsymbol{Z}_0, \boldsymbol{Z}_{1:N}|\boldsymbol{y}_{0:N}}$, for any $N > 0$. Orange edges have been added compared to the tree structure of Figure 7.

8. Numerical Illustration

We illustrate some aspects of the preceding theory using a problem of sequential inference in a non-Gaussian state-space model. In particular, we show the application of decomposable transport maps (Sections 6 and 7) to joint state and parameter inference in a stochastic volatility model. This example is intended as a direct and simple illustration of the theory. The notion of decomposable transport maps is useful well beyond the sequential inference setting, and entails the general problem of inference in continuous non-Gaussian graphical models. We refer the reader to Morrison et al. (2017) for an application of the theory of sparse transports (Section 5) to the problem of learning the Markov structure of a non-Gaussian distribution, and we defer further numerical investigations to a dedicated paper (Bigoni et al., 2019).

Following (Kim et al., 1998; Rue et al., 2009), we model the scalar log-volatility $(\boldsymbol{Z}_k)$ of the return of a financial asset at time $k = 0, \dots, N$ using an autoregressive process of order one, which is fully specified by $\boldsymbol{Z}_{k+1} = \boldsymbol{\mu} + \boldsymbol{\phi}\,(\boldsymbol{Z}_k - \boldsymbol{\mu}) + \boldsymbol{\varepsilon}_k$, for all $k \geq 0$, where $\boldsymbol{\varepsilon}_k \sim \mathcal{N}(0, 1/16)$ is independent of $\boldsymbol{Z}_k$, $\boldsymbol{Z}_0|\boldsymbol{\mu}, \boldsymbol{\phi} \sim \mathcal{N}(\boldsymbol{\mu}, \frac{1}{1-\boldsymbol{\phi}^2})$, and where $\boldsymbol{\phi}$ and $\boldsymbol{\mu}$ represent scalar hyperparameters of the model. In particular, $\boldsymbol{\mu} \sim \mathcal{N}(0, 1)$ and $\boldsymbol{\phi} = 2\exp(\boldsymbol{\phi}^\star)/(1+\exp(\boldsymbol{\phi}^\star)) - 1$ with $\boldsymbol{\phi}^\star \sim \mathcal{N}(3, 1)$. We define $\boldsymbol{\Theta} \coloneqq (\boldsymbol{\mu}, \boldsymbol{\phi})$. The process $(\boldsymbol{Z}_k)$ and parameters $\boldsymbol{\Theta}$ are unobserved and must be estimated from an observed process $(\boldsymbol{Y}_k)$, which represents the mean return on holding the asset at time k, $\boldsymbol{Y}_k = \boldsymbol{\xi}_k \exp(\frac{1}{2}\boldsymbol{Z}_k)$, where $\boldsymbol{\xi}_k$ is a standard normal random variable independent of $\boldsymbol{Z}_k$. As a data set $(\boldsymbol{y}_k)_{k=0}^N$, we use the $N+1$ daily differences of the pound/dollar exchange rate starting on 1 October 1981, with $N = 944$ (Rue et al., 2009; Durbin and Koopman, 2000).

Our goal is to sequentially characterize $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k}|\boldsymbol{y}_{0:k}}$, for all $k = 0, \ldots, N$, as observations $(\boldsymbol{y}_k)$ become available. The Markov structure of $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:N}|\boldsymbol{y}_{0:N}}$ matches Figure 8. We solve the problem using the algorithm introduced in Section 7.3: we compute a sequence, $(\mathfrak{M}_j)_{j=0}^{N-1}$, of four-dimensional transport maps ($n = \dim(\boldsymbol{Z}_j) = 1$ and $p = \dim(\boldsymbol{\Theta}) = 2$) according to their definition in Theorem 12 and using the variational form (6). All reference densities are standard Gaussians. Then, by Theorem 12[part 1], for any $k < N$, we can easily sample the filtering marginal $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ by pushing forward a standard normal through the subcomponent $\mathfrak{M}_k^1$ of $\mathfrak{M}_k$, and we can also sample the posterior distribution over the static parameters $\pi_{\boldsymbol{\Theta}|\boldsymbol{y}_{0:k+1}}$ by pushing forward a standard normal through the map $\mathfrak{T}_k^{\boldsymbol{\Theta}}$. The map $\mathfrak{T}_k^{\boldsymbol{\Theta}} = \mathfrak{M}_0^{\boldsymbol{\Theta}} \circ \cdots \circ \mathfrak{M}_k^{\boldsymbol{\Theta}}$ is updated sequentially over time (via regression) using the recursion $\mathfrak{T}_k^{\boldsymbol{\Theta}} = \mathfrak{T}_{k-1}^{\boldsymbol{\Theta}} \circ \mathfrak{M}_k^{\boldsymbol{\Theta}}$, so that the cost of evaluating $\mathfrak{T}_k^{\boldsymbol{\Theta}}$ does not increase with k. The resulting algorithm for parameter estimation is thus sequential. Moreover, if we want to sample $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{Y}_{0:k+1}}$—the full Bayesian solution at time $k+1$—we simply need to embed each $\mathfrak{M}_j$ into an identity map to form the transport T_j, for $j = 0, \ldots, k$, and push forward reference samples through the composition $\mathfrak{T}_k = T_0 \circ \cdots \circ T_k$ (Theorem 12[part 2]). See Appendix C for pseudocode of the relevant algorithms.

Figures 10 and 11 show the resulting smoothing and filtering marginals of the states over time, respectively. Figures 12 and 13 collect the corresponding posterior marginals of the static parameters over time. Figure 14 illustrates marginals of the posterior predictive distribution of the data, together with the observed data $(\boldsymbol{y}_k)$, showing excellent coverage overall.

Our results rely on a numerical approximation of the desired transport maps. Each component of $\mathfrak{M}_k$ is parameterized via the monotone representation (5), with (a_k) and (b_k) chosen to be Hermite polynomials and functions, respectively, of total degree seven.The expectation in (6) is approximated using tensorized Gauss quadrature rules. The resulting minimization problems are solved sequentially using the Newton–CG method (Wright and Nocedal, 1999). This test case was run using the dedicated software package publicly available at `http://transportmaps.mit.edu`. The website contains details about additional possible parameterizations of the maps.

There are several ways to investigate the quality of these approximations. Figures 10, 12, and 13 compare the numerical approximation (via a decomposable transport map) of the smoothing marginals of the states and the posteriors of the static parameters to a "reference" solution obtained via MCMC. The MCMC chain is run until it yields 10^5 effectively independent samples. The two solutions agree remarkably well and are almost indistinguishable in most places. (Of course, MCMC in this context is not a data-sequential algorithm; it requires that all the data $(\boldsymbol{y}_k)_{k=0}^N$ be available simultaneously.) An important fact is that the MCMC chain is generated using an *independence* proposal (Robert and Casella, 2013) given by the pushforward of a standard Gaussian through the numerical approximation of $\mathfrak{T}_{N-1}$ (denoted as $\widetilde{\mathfrak{T}}_{N-1}$). The resulting MCMC chain has an acceptance rate slightly above 75%, confirming the overall quality of the variational approximation. We notice, however, a *slow* accumulation of error in the posterior marginal for the static parameter $\boldsymbol{\mu}$ (Figure 13). This is not surprising since we are performing *sequential* parameter inference (Jacob, 2015).

A second quality test can proceed as follows: since we use a standard Gaussian reference distribution $\boldsymbol{\nu}_\eta$, we expect the pullback of $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:N}|\boldsymbol{y}_{0:N}}$ through $\widetilde{\mathfrak{T}}_{N-1}$ to be close

to a standard Gaussian. Figure 15 supports this claim by showing a collection of random two-dimensional conditionals of the approximate pullback: these "slices" of the 947-dimensional (N+1 states plus two hyperparameters) pullback distribution are identical to a two-dimensional standard normal, as expected. The fact that we *can* evaluate the approximate pullback density is one of the key features of this variational approach to inference. Even more, we can use this approximate pullback density to estimate the KL divergence between our target $\boldsymbol{\nu}_\pi$ (the full Bayesian solution at time N) and the approximating measure $(\widetilde{\mathfrak{T}}_{N-1})_\sharp \boldsymbol{\nu}_\eta$, via the variance diagnostic in (8). A numerical realization of (8) yields $\mathcal{D}_{\mathrm{KL}}(\, (\widetilde{\mathfrak{T}}_{N-1})_\sharp \boldsymbol{\nu}_\eta \,\|\, \boldsymbol{\nu}_\pi \,) \approx 1.07 \times 10^{-1}$, which confirms the good numerical approximation of $\boldsymbol{\nu}_\pi$, a 947-dimensional target measure. For comparison, we note that the KL divergence from $\boldsymbol{\nu}_\pi$ to its Laplace approximation (a Gaussian approximation at the mode) is ≈ 5.68—considerably worse than what is achieved through optimization of a nonlinear transport map. Moreover, the Laplace approximation cannot be computed sequentially with a constant effort per time step.

While a slow accumulation of errors is expected for sequential parameter inference, we also wish to investigate the stability of our transport map approximation for recursive smoothing *without* static parameters. We try the following experiment: (1) compute the posterior medians of the static parameters after $N + 1 = 945$ days, i.e., $\boldsymbol{\theta}^* = \mathrm{med}[\Theta|\boldsymbol{y}_0, \ldots, \boldsymbol{y}_{944}]$; and then (2) use these parameters to characterize the smoothing distribution $\pi_{\boldsymbol{Z}_0,\ldots,\boldsymbol{Z}_{2500}|\boldsymbol{\theta}^*,\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{2500}}$ of the log-volatility over (roughly) the next ten years worth of exchanges, using the sequential algorithm proposed in Section 7.1, which in this case amounts to computing only a sequence of two-dimensional maps $(\mathfrak{M}_k)$. The resulting smoothing marginals are shown in Figure 16 and compared to those of a reference MCMC simulation with 10^5 effectively independent samples; we observe excellent agreement despite the long assimilation window. We then repeat the same experiment for an even longer assimilation window, i.e., 9009 steps or roughly 35 years. Figure 17 shows the remarkable stability of the resulting smoothing approximation, at least for low-order marginals. In fact, even the approximation of the *joint* distribution of the states is quite good, as reported in the last column of Table 1. Understanding how errors propagate in this variational framework—and what could be potential mechanisms for the "dissipation" of errors—is an exciting avenue for future work.

The results presented so far are very accurate, but also expensive. Table 1 collects the computational times for the joint state-parameter inference problem (approximately two days) and for the long-time (9009 step) smoothing problem (approximately 40 minutes), using a degree-seven map. While there remains a tremendous opportunity to develop more performance-oriented versions of our transport map code, specialized to the problem of sequential inference, the present framework also offers a practical and powerful tradeoff between computational cost and accuracy. In Appendix D, we re-run all our test cases using linear, rather than degree seven, parameterizations of the maps $\{\mathfrak{M}_k\}$. Table 1 shows that the computational times are dramatically reduced: from two days to approximately one minute for the joint state-parameter inference problem, and from 40 minutes to 7 minutes for the long-time smoothing problem. The reduction in computational time comes, of course, at the price of accuracy; see last column of Table 1. This reduction in accuracy may or may not be acceptable. For instance, in Figure 24, it is difficult to distinguish the linear map approximation from the reference MCMC solution. Quantitatively, we know

from Table 1 that a linear map is worse at approximating the full Bayesian solution than a degree-7 transformation. Yet, as far as quantiles of low-order marginals are concerned, the two solutions are indistinguishable (Figure 24); in an applied setting, this accuracy may be more than sufficient. In other cases, however, a linear map might be inadequate. For example, the parameter marginals in Figures 20 and 22, estimated using linear maps, are much worse than their degree-7 counterparts (Figures 12 and 13). In these cases, we *need* nonlinear transformations.

Clearly, there is a rich spectrum of possibilities between a linear and a high-order transport map. Some parameterizations can scale with dimension (e.g., separable but nonlinear representations), while others cannot (e.g., total-degree polynomial expansions). Depending on the problem, some parameterizations will lead to accurate results, while others will not. Yet, the cost-accuracy tradeoff in the transport framework can be *controlled*, e.g., by estimating the quality of a given approximation using (8).

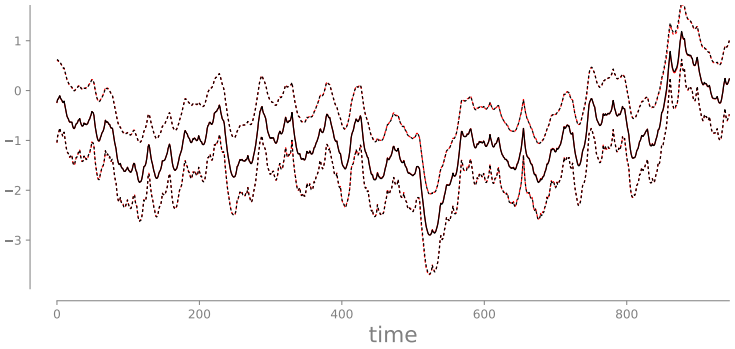


Figure 10: Comparison between the $\{5, 95\}$–percentiles (dashed lines) and the mean (solid line) of the numerical approximation of the smoothing marginals $\pi_{\boldsymbol{Z}_k|\boldsymbol{y}_{0:N}}$ via transport maps (red lines) versus a "reference" MCMC solution (black lines), for $k = 0, \ldots, N$. The two solutions are indistinguishable.

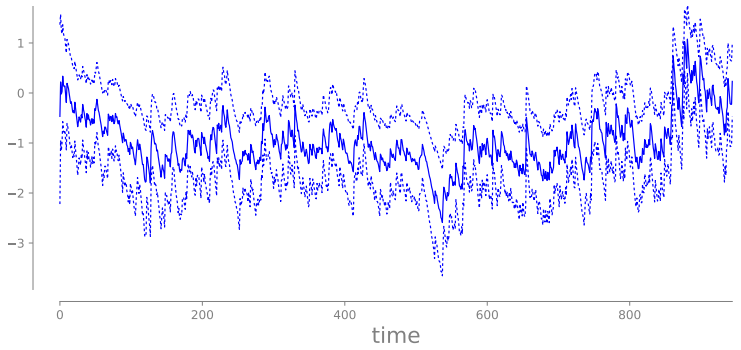


Figure 11: At each time k, we illustrate the $\{5, 95\}$–percentiles (dotted lines) and the mean (solid line) of the numerical approximation of the filtering distribution $\pi_{\boldsymbol{Z}_k|\boldsymbol{y}_{0:k}}$.

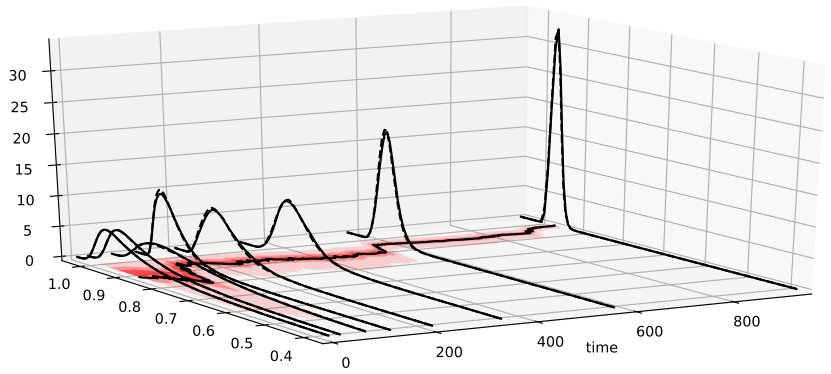


Figure 12: *(Horizontal plane)* At each time k, we illustrate the $\{5, 25, 40, 60, 75, 95\}$–percentiles (shaded regions) and the mean (solid line) of the numerical approximation of $\pi_{\boldsymbol{\phi}|\boldsymbol{y}_{0:k}}$, the posterior marginal of the static parameter $\boldsymbol{\phi}$. *(Vertical axis)* At several times k we also compare the transport map numerical approximation of $\pi_{\boldsymbol{\phi}|\boldsymbol{y}_{0:k}}$ (solid lines) with a reference MCMC solution (dashed lines). The two distributions agree remarkably well.

Type	**# steps**	**Order**	**Time [m : s]**	**# cores**	**Figures**	**Var. diag.** (8)
S/P	945	Laplace	00 : 04	1		5.68
		7	$\approx$ 2 days	64	10 – 15	1.07×10^{-1}
		linear	01 : 14	1	18 – 23	1.77
S	9009	Laplace	00 : 42	1		10.0
		7	42 : 50	1	17	1.19×10^{-1}
		linear	06 : 40	1	24	5.01

Table 1: Computational effort required to compute a decomposable transport map for different complexities of the transformations $\mathfrak{M}_k$—linear versus degree seven—and for different inference scenarios—smoothing and static parameter estimation *(top row)* or long-time smoothing without static parameters *(bottom row)*, for the stochastic volatility model of Section 8. The last column reports the variance diagnostic (8) for the corresponding *joint* posterior, not just a few marginals. It highlights a tradeoff between cost and accuracy, typical of the transport map approach to variational inference. For comparison, we also report the cost and accuracy of a simple Laplace approximation, which requires no formal optimization.

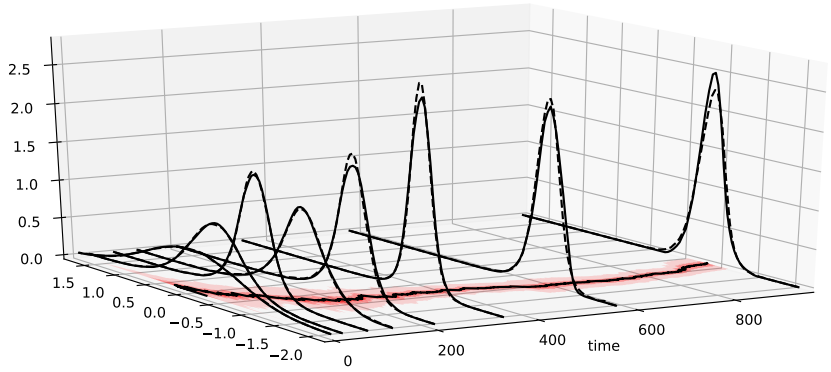


Figure 13: Same as Figure 12, but for the static parameter $\boldsymbol{\mu}$.

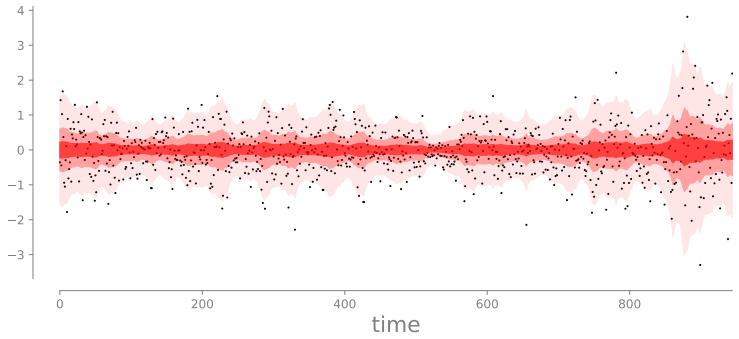


Figure 14: Shaded regions represent the $\{5, 25, 40, 60, 75, 95\}$–percentiles of the marginals of the posterior predictive distribution (conditioning on all the data), along with black dots that represent the observed data $(\boldsymbol{y}_k)_{k=0}^N$.

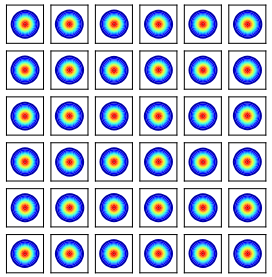


Figure 15: Randomly chosen two-dimensional conditionals of the pullback of $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:N}|\boldsymbol{y}_{0:N}}$ through the numerical approximation of $\mathfrak{T}_{N-1}$. Since we use a standard normal reference distribution, the numerical approximation of $\mathfrak{T}_{N-1}$ should be deemed satisfactory if the pullback density is close to a standard normal, as it is here.

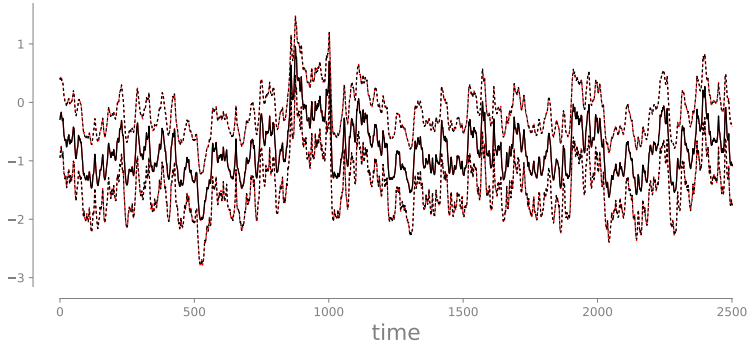


Figure 16: Comparison between the $\{5, 95\}$–percentiles (dashed lines) and the mean (solid line) of the transport map numerical approximation of the smoothing marginals $\pi_{\boldsymbol{Z}_k|\boldsymbol{\theta}^*,\boldsymbol{y}_{0:2500}}$, with $\boldsymbol{\theta}^* = \text{med}[\Theta|\boldsymbol{y}_0, \ldots, \boldsymbol{y}_N]$ (red lines), and a reference MCMC solution (black lines). The two solutions are indistinguishable.

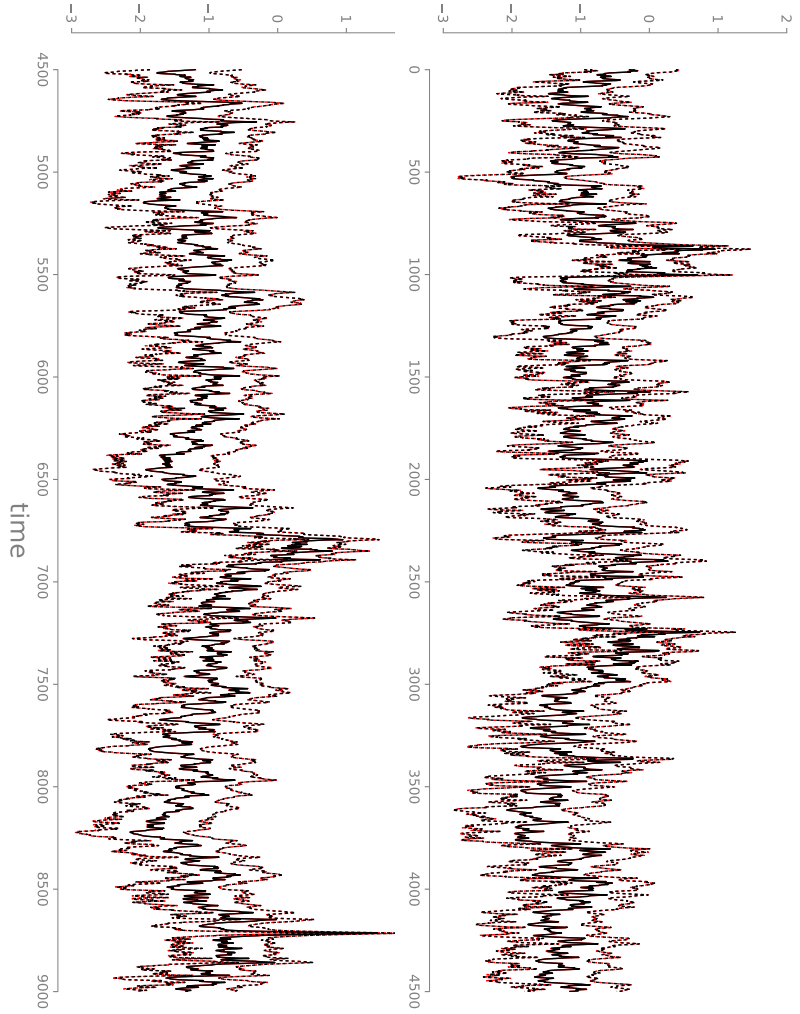


Figure 17: Same as Figure 16, but for a longer assimilation window, i.e., $\pi_{\boldsymbol{Z}_k|\boldsymbol{\theta}^*,\boldsymbol{y}_{0:9000}}$. The smoothing approximation remains excellent despite the widening inference horizon.

9. Discussion

This paper has focused on the problem of coupling a pair $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ of absolutely continuous measures on $\mathbb{R}^n$, for the purpose of sampling or integration. If $\boldsymbol{\nu}_\eta$ is a tractable measure (e.g., an isotropic Gaussian) and $\boldsymbol{\nu}_\pi$ is an intractable measure of interest (e.g., a posterior distribution), then a deterministic coupling enables principled approximations of integrals via the identity $\int g \, \mathrm{d}\boldsymbol{\nu}_\pi = \int g \circ T \, \mathrm{d}\boldsymbol{\nu}_\eta$. In other words, a deterministic coupling provides a simple way to simulate $\boldsymbol{\nu}_\pi$ by pushing forward samples from $\boldsymbol{\nu}_\eta$ through a transport map T. This idea, modulo some variations, has been exploited in a variety of statistical and machine learning applications—some old, some new—including random number generation (Marsaglia and Tsang, 2000), variational inference (Moselhy and Marzouk, 2012; Schillings and Schwab, 2016; Rezende and Mohamed, 2015), the computation of model evidence (Meng and Schilling, 2002), model learning and density estimation (Laparra et al., 2011; Anderes and Coram, 2012; Stavropoulou and Müller, 2015), non-Gaussian proposals for MCMC or importance sampling (Parno and Marzouk, 2018; Bardsley et al., 2014; Oliver, 2015), multiscale modeling (Parno et al., 2016), and filtering (Daum and Huang, 2008; Chorin and Tu, 2009; Reich, 2013), to name a few. Indeed there are infinitely many ways to transport one measure to another (Villani, 2008) and as many ways to compute one. Yet these maps are not equally easy to characterize.

This paper establishes an explicit link between the conditional independence structure of $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ and the existence of low-dimensional couplings induced by transport maps that are *sparse* and/or *decomposable*. These results can enhance a wide array of numerical approaches to the transportation of measures, including (Tabak and Turner, 2013; Rezende and Mohamed, 2015; Liu and Wang, 2016; Bigoni et al., 2019), and thus facilitate simulation with respect to complex distributions in high dimensions. We briefly discuss our main results below.

Sparse transports. A sparse transport is a map whose components do not depend on all input variables. Section 5 derives tight bounds on the sparsity pattern of the Knothe–Rosenblatt (KR) rearrangement (a triangular transport map) based solely on the Markov structure of $\boldsymbol{\nu}_\pi$, provided that $\boldsymbol{\nu}_\eta$ is a product measure (Theorem 3). This analysis shows that the inverse of the KR rearrangement is the natural generalization to the *non-Gaussian* case of the Cholesky factor of the precision matrix of a Gaussian MRF—in that both the inverse KR rearrangement (a potentially nonlinear map) and the Cholesky factor (a linear map) have the same sparsity pattern given target measures with the same Markov structure. Thus the KR rearrangement can be used to extend well-known modeling and sampling techniques for high-dimensional Gaussian MRFs (Rue and Held, 2005) to non-Gaussian fields (Section 5.2). These results are particularly useful when constructing a transport map from samples via convex optimization (Parno, 2015) and suggest novel approaches to model learning (Morrison et al., 2017) and high-dimensional filtering (Spantini, 2017, Ch. 6). Section 5 shows that sparsity is usually a feature of inverse transports, while direct transports tend to be dense, even for the most trivial Markov structures. In fact, the sparsity of direct transports stems from *marginal* (rather than conditional) independence—a property frequently exploited in localization schemes for high-dimensional covariance estimation (Gaspari and Cohn, 1999; Hamill et al., 2001).

Decomposable transports. A decomposable map is a function that can be written as the composition of *finitely* many low-dimensional maps that are triangular up to a permutation—i.e., $T = T_1 \circ \cdots \circ T_\ell$, where each T_i differs from the identity only along a small subset of its components and is a generalized triangular function as defined in Section 6. Theorem 7 shows that every target measure whose Markov network admits a graph decomposition can be coupled with a product (reference) measure via a decomposable map. Decomposable maps are important because they are much easier to represent than arbitrary multivariate functions on $\mathbb{R}^n$. In general, these maps are non-triangular, even though each map in the composition is generalized triangular.

The notion of a decomposable map is different from the composition-of-maps approaches advocated in the literature for the approximation of transport maps, e.g., consider normalizing flows (Rezende and Mohamed, 2015) or Stein variational algorithms (Anderes and Coram, 2012; Liu and Wang, 2016; Detommaso et al., 2018), but also (Tabak and Turner, 2013; Laparra et al., 2011). In these approaches, very simple maps $(M_i)_{i\geq 1}$ are composed in growing number to define a transport map of increasing complexity, $M = M_1 \circ \cdots \circ M_k$. The number of layers in M depends on the desired accuracy of the transport and can be arbitrarily large. On the other hand, a decomposable coupling is induced by a special transport map that can be written *exactly* as the composition of finitely many maps, $T = T_1 \circ \cdots \circ T_\ell$, where each T_i has a specific sparsity pattern that makes it low-dimensional. This definition does not specify a representation for T_i. In fact, each T_i could itself be approximated by the composition of simple maps using *any* of the aforementioned techniques. The advantage of targeting a decomposable transport is the fact that the (T_i) are *guaranteed* to be low-dimensional.

Approximate Markov properties. Sparsity and decomposability of certain transport maps are induced by the Markov properties of the target measure. A natural question is: what happens when $\boldsymbol{\nu}_\pi$ satisfies some Markov properties only *approximately*? In particular, let $\boldsymbol{\nu}_\pi$ be Markov with respect to $\boldsymbol{\mathcal{G}}$, and assume that there exists a measure $\hat{\boldsymbol{\nu}} \in \mathscr{M}_+(\mathbb{R}^n)$ which is Markov with respect to a graph $\hat{\boldsymbol{\mathcal{G}}}$ that is *sparser* than $\boldsymbol{\mathcal{G}}$ and such that $\mathcal{D}_{\mathrm{KL}}(\,\hat{\boldsymbol{\nu}}\,||\,\boldsymbol{\nu}_\pi\,) < \varepsilon$, for some $\varepsilon > 0$. For small ε, we would be tempted to use $\hat{\boldsymbol{\mathcal{G}}}$ to characterize couplings of $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ that are possibly sparser or more decomposable than those associated with $\boldsymbol{\mathcal{G}}$. Concretely, if we are interested in a triangular transport that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, we could minimize $\mathcal{D}_{\mathrm{KL}}(\,T_\sharp\,\boldsymbol{\nu}_\eta\,||\,\boldsymbol{\nu}_\pi\,)$ over the set of maps whose *inverse* has the same sparsity pattern as the KR rearrangement between $\hat{\boldsymbol{\nu}}$ and $\boldsymbol{\nu}_\eta$. Bounds on this sparsity pattern are given by Theorem 3 using only graph operations on $\hat{\boldsymbol{\mathcal{G}}}$; no explicit knowledge of $\hat{\boldsymbol{\nu}}$ is required. Alternatively, if we are interested in decomposable transports that push forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$, we could minimize $\mathcal{D}_{\mathrm{KL}}(\,T_\sharp\,\boldsymbol{\nu}_\eta\,||\,\boldsymbol{\nu}_\pi\,)$ over the set of maps that factorize as any of the decomposable transports between $\boldsymbol{\nu}_\eta$ and $\hat{\boldsymbol{\nu}}$. The shapes of these low-dimensional factorizations are given by Theorem 7 using, once again, only graph operations on $\hat{\boldsymbol{\mathcal{G}}}$.

Now let $\hat{\mathcal{T}}$ denote the set of maps whose structure is constrained by $\hat{\boldsymbol{\mathcal{G}}}$ in terms of sparsity or decomposability. It is easy to show that

$$\min_{T\in\hat{\mathcal{T}}} \mathcal{D}_{\mathrm{KL}}(\,T_\sharp\,\boldsymbol{\nu}_\eta\,||\,\boldsymbol{\nu}_\pi\,) < \varepsilon,$$

which means that the price of assuming that the coupling is either sparser or more decomposable than it ought to be is just a small error in the approximation of $\boldsymbol{\nu}_\pi$.

Of course, the pending question is whether $\boldsymbol{\nu}_\pi$ can be well approximated by a measure that satisfies additional Markov properties. There is some work on this topic, e.g., Johnson and Willsky, 2008; Jog and Loh, 2015; Cheng et al., 2015—especially in the case of Gaussian measures—but a more thorough investigation of the problem remains an open and important direction for future work. Interestingly, the transport map framework also allows one to *adaptively* discover information about low-dimensional couplings. For instance, one might start with a very sparse transport map and then incrementally decrease the sparsity level of the map until the resulting approximation of $\boldsymbol{\nu}_\pi$ becomes satisfactory. The same can be done for decomposable transports. See Bigoni et al. (2019) for some details on this idea.

Filtering and smoothing. Section 6.4 shows how not only the representation, but also the *computation*, of a decomposable map, $T = T_1 \circ \cdots \circ T_\ell$, can be broken into a sequence of ℓ simpler steps, each associated with a low-dimensional optimization problem whose solution yields T_i. We give a concrete example of this idea for filtering, smoothing, and joint state–parameter inference in nonlinear and non-Gaussian state-space models (Section 7). In this context, Theorems 9 and 12 introduce variational approaches for characterizing the full posterior distribution of the sequential inference problem, essentially by performing only recursive lag–1 smoothing with transport maps. The proposed approaches consist of a *single* forward pass on the state-space model, and generalize the square-root Rauch-Tung-Striebel smoother to non-Gaussian models (see Section 7.2). In practice, we should think of Theorems 9 and 12 as providing "meta-algorithms" within which all kinds of approximations can be introduced, e.g., linearizations of the forward model, restriction to linear maps, and approximate flows (Daum and Huang, 2008; Liu and Wang, 2016), to name a few. These approximations are the workhorse of modern approaches to large-scale filtering, e.g., data assimilation in geophysical applications (Särkkä, 2013; Evensen, 2007), and may play a key role in further instantiations of the "meta-algorithms" proposed in Section 7. Of course, it would be desirable to complement such variational approximations with a rigorous error analysis, analogous to the analysis available for SMC methods (Crisan and Doucet, 2002; Del Moral, 2004; Smith et al., 2013). It is also important to note that one can always use functionals like (8) to estimate the quality of a given approximate map, or use the map itself to build sophisticated proposals for sampling techniques like MCMC (Parno and Marzouk, 2018).

A recent approach that constructs an approximation of the KR rearrangement for sequential inference is the "Gibbs flow" of Heng et al. (2015); here, the authors define a proposal for SMC (or MCMC) methods using the solution map of a discretized ordinary differential equation (ODE) whose drift term depends only on the full conditionals of the target distribution. Evaluating the solution map only requires the evaluation of one-dimensional integrals, and the action of this map implicitly defines a transport, without any explicit parameterization of the transformation. Several other filtering approaches in the literature, e.g., (Daum and Huang, 2012; Yang et al., 2013), rely on the solution of ODEs that are different from Heng et al. (2015), but also inspired by ideas from mass transportation. Implicit sampling for particle filters (Chorin and Tu, 2009) also implicitly constructs a transport map, from a standard Gaussian to a particular approximation of the filtering distribution; the action of this transport is realized by solving an optimization/root-finding problem for each sample (Morzfeld et al., 2012).

One of the first contributions to use *optimal* transport in filtering is due to Reich (2013), who constructs an optimal transport plan between an empirical approximation of the forecast distribution (given by simulating the prior dynamic) and a corresponding empirical approximation of the filtering distribution, obtained by reweighing the forecast ensemble according to the likelihood. Thus, Reich (2013) solves a *discrete* Kantorovich optimal transport problem instead of a continuous problem for a transport map (cf. Section 7.1). A linear transformation of the forecast ensemble is then derived from the optimal plan. In this approach, the explicit construction of couplings is used only to update the forecast distribution, instead of the previous filtering marginal.

Further extensions. We envision many additional ways to extend the present work. For instance, it would be interesting to investigate the low-dimensional structure of deterministic couplings between pair of measures $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$ that are not absolutely continuous and that need not be defined on the same space $\mathbb{R}^n$. Such couplings are usually induced by "random" maps and can be particularly effective for approximating multi-modal distributions; see the warp bridge transformations in (Meng and Schilling, 2002; Wang and Meng, 2016) for some examples.

Finally, we emphasize that this paper characterizes some classes of low-dimensional maps, but certainly not all. In particular, low dimensionality need not stem from the Markov properties of the underlying measures. In ongoing work we are exploring the notion of low-rank couplings: these are induced by transport maps that are low-dimensional up to a rotation of the space, i.e., maps whose action is nontrivial only along a low-dimensional subspace. This type of structure appears quite naturally in certain high-dimensional Bayesian inference problems—e.g., inverse problems (Stuart, 2010) and spatial statistics—where the data may be informative only about a few linear combinations of the latent parameters (Spantini et al., 2015; Cui et al., 2014; Spantini et al., 2017). Low-rank structure can be detected via certain average derivative functionals (Samarov, 1993; Constantine et al., 2014) but cannot be deduced, in general, from the Markov structure of $(\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi)$.

Acknowledgments

We would like to thank Ricardo Baptista, Chi Feng, Jeremy Heng, Pierre Jacob, Qiang Liu, Rebecca Morrison, Zheng Wang, Alan Willsky, Olivier Zahm, and Benjamin Zhang for many insightful discussions and for pointing us to key references in the literature. This work was supported in part by the US Department of Energy, Office of Advanced Scientific Computing (ASCR), under grant numbers DE-SC0003908 and DE-SC0009297.

Appendix A. Generalized Knothe-Rosenblatt Rearrangement

In this section we first review the classical notion of KR rearrangement (Rosenblatt, 1952), and then give a formal definition for a *generalized* KR rearrangement, i.e., a transport map that is lower triangular up to a permutation. A disclaimer: these transports can also be defined under weaker conditions than those considered here, at the expense, however, of some useful regularity (Bogachev et al., 2005).

The following definition introduces the one-dimensional version of the KR-rearrangement, and it is key to extend the transport to higher dimensions.

Definition 13 (Increasing rearrangement on $\mathbb{R}$) *Let $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R})$, and let F, G be their respective cumulative distribution functions, i.e., $F(t) = \boldsymbol{\nu}_\eta((-\infty, t))$ and $G(t) = \boldsymbol{\nu}_\pi((-\infty, t))$. Then the increasing rearrangement on $\mathbb{R}$ is given by $T = G^{-1} \circ F$.*

Under the hypothesis of Definition 13, it is easy to see that both F and G are homeomorphisms, and that T is a strictly increasing map that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ (Santambrogio, 2015).

Definition 14 (Knothe-Rosenblatt rearrangement) *Given $\boldsymbol{X} \sim \boldsymbol{\nu}_\eta$, $\boldsymbol{Z} \sim \boldsymbol{\nu}_\pi$, with $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$, and a pair η, π of strictly positive densities for $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$, respectively, the corresponding KR rearrangement is a triangular map $T : \mathbb{R}^n \to \mathbb{R}^n$ defined, recursively, as follows. For all $\boldsymbol{x}_{1:k-1} \in \mathbb{R}^{k-1}$, the map $\xi \mapsto T^k(\boldsymbol{x}_{1:k-1}, \xi)$—the restriction of the kth component of T onto its first $k-1$ inputs—is defined as the increasing rearrangement on $\mathbb{R}$ that pushes forward $\xi \mapsto \eta_{X_k|\boldsymbol{X}_{1:k-1}}(\xi|\boldsymbol{x}_{1:k-1})$ to $\xi \mapsto \pi_{Z_k|\boldsymbol{Z}_{1:k-1}}(\xi|T^1(x_1), \dots, T^{k-1}(\boldsymbol{x}_{1:k-1}))$, where $\eta_{X_k|\boldsymbol{X}_{1:k-1}}$ and $\pi_{Z_k|\boldsymbol{Z}_{1:k-1}}$ are conditional densities defined as in (2).*

Notice that for any measure $\boldsymbol{\nu}$ in $\mathscr{M}_+(\mathbb{R}^n)$ there always exists a strictly positive *version* of its density. By considering such positive densities in Definition 14, we can define the KR rearrangement on the entire $\mathbb{R}^n$ (Bogachev et al., 2005). In fact, we should really think of Definition 14 as providing a possible *version* of the KR rearrangement (recall that the increasing triangular transport is unique up to sets of measure zero). Since in this case $\boldsymbol{\nu}_\pi$ is equivalent to the Lebesgue measure ($\boldsymbol{\nu}_\pi(\mathcal{A}) = \int_{\mathcal{A}} \pi(\boldsymbol{x})\, \boldsymbol{\lambda}(\mathrm{d}\boldsymbol{x}) = 0 \Rightarrow \boldsymbol{\lambda}(\mathcal{A}) = 0$ if $\pi > 0$ a.e.), the component (3) is also absolutely continuous on all compact intervals (Bogachev et al., 2005, Lemma 2.4). As a result, the rearrangement can be used to define general change of variables as well as pullbacks and pushforwards with respect to arbitrary densities, as shown by the following lemma adapted from Bogachev et al. (2005).

Lemma 15 *Let T be an increasing triangular bijection on $\mathbb{R}^n$ such that the functions*

$$\xi \mapsto T^k(x_1, \dots, x_{k-1}, \xi)$$

are absolutely continuous on all compact intervals for a.e. $(x_1, \dots, x_{k-1}) \in \mathbb{R}^{k-1}$. Then for any integrable function φ, it holds:

$$\int \varphi(\boldsymbol{y})\, \mathrm{d}\boldsymbol{y} = \int \varphi(T(\boldsymbol{x}))\, \det \nabla T(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x},$$

where $\det \nabla T \coloneqq \prod_{k=1}^n \partial_k T^k$. In particular, if $\boldsymbol{\nu}_\rho$ is a measure on $\mathbb{R}^n$ with density ρ, then we also have $T^\sharp \boldsymbol{\nu}_\rho \ll \boldsymbol{\lambda}$ with density (a.e.):

$$T^\sharp \rho(\boldsymbol{x}) = \rho(T(\boldsymbol{x}))\, \det \nabla T(\boldsymbol{x}). \tag{36}$$

The lemma can also be applied to the inverse KR rearrangement T^{-1} to show that $T_\sharp \boldsymbol{\nu}_\rho \ll \boldsymbol{\lambda}$, where the form of the corresponding pushforward density $T_\sharp \rho$ is given by replacing T with T^{-1} in (36). We will use these results extensively in the proofs of Appendix B. Notice,

however, that Lemma 15 does not hold for a generic triangular function: the map must be somewhat regular, in the sense specified by the lemma. Bogachev et al. (2005) give an in depth discussion on this topic.

We now give a constructive definition for a generalized KR rearrangement.

Definition 16 (Generalized Knothe-Rosenblatt rearrangement) *Given $\boldsymbol{X} \sim \boldsymbol{\nu}_\eta$, $\boldsymbol{Z} \sim \boldsymbol{\nu}_\pi$, with $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$, a pair η, π of strictly positive densities for $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$, respectively, and a permutation σ of $\mathbb{N}_n$, the corresponding σ-generalized KR rearrangement is a σ-triangular map[12] $T : \mathbb{R}^n \to \mathbb{R}^n$ defined at any $\boldsymbol{x} \in \mathbb{R}^n$ using the following recursion in k. The map $\xi \mapsto T^{\sigma(k)}(x_{\sigma(1)}, \dots, x_{\sigma(k-1)}, \xi)$ is defined as the increasing rearrangement on $\mathbb{R}$ that pushes forward $\xi \mapsto \eta_{X_{\sigma(k)}|\boldsymbol{X}_{\sigma(1:k-1)}}(\xi|\boldsymbol{x}_{\sigma(1:k-1)})$ to*

$$\xi \mapsto \pi_{Z_{\sigma(k)}|\boldsymbol{Z}_{\sigma(1:k-1)}}(\xi|T^{\sigma(1)}(x_{\sigma(1)}), \dots, T^{\sigma(k-1)}(\boldsymbol{x}_{\sigma(1:k-1)})),$$

where $\boldsymbol{x}_{\sigma(1:k-1)} = x_{\sigma(1)}, \dots, x_{\sigma(k-1)}$.

Existence of a generalized KR rearrangement follows trivially from its definition. Moreover, the transport map satisfies all the regularity properties discussed for the classic KR rearrangement, including Lemmas 1 and 15. Thus we will often cite these two results when dealing with generalized KR rearrangements in our proofs. The following lemma shows that the computation of a generalized KR rearrangement is also essentially no different than the computation of a lower triangular transport (and thus all the discussion of Section 3 readily applies).

Lemma 17 *Given $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$, let T be a σ-generalized KR rearrangement that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ for some permutation σ. Then $T = Q_\sigma^\top \circ T_\ell \circ Q_\sigma$ a.e., where $Q_\sigma \in \mathbb{R}^{n \times n}$ is a matrix representing the permutation, i.e., $(Q^\sigma)_{ij} = (\boldsymbol{e}_{\sigma(i)})_j$, and where T_ℓ is a (lower triangular) KR rearrangement that pushes forward $(Q_\sigma)_\sharp \boldsymbol{\nu}_\eta$ to $(Q_\sigma)_\sharp \boldsymbol{\nu}_\pi$.*

Proof If T_ℓ pushes forward $(Q_\sigma)_\sharp \boldsymbol{\nu}_\eta$ to $(Q_\sigma)_\sharp \boldsymbol{\nu}_\pi$, then $\boldsymbol{\nu}_\eta \circ Q_\sigma^\top \circ T_\ell^{-1} = \boldsymbol{\nu}_\pi \circ Q_\sigma^\top$, and so $T = Q_\sigma^\top \circ T_\ell \circ Q_\sigma$ must push forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$. Moreover, notice that $T^{\sigma(k)}(\boldsymbol{x}) = T_\ell^k(\boldsymbol{x}^\top \boldsymbol{e}_{\sigma(1)}, \dots, \boldsymbol{x}^\top \boldsymbol{e}_{\sigma(k)})$, which shows that T is a monotone increasing σ-generalized triangular function (see Definition 6). The lemma then follows by $\boldsymbol{\nu}_\eta$-uniqueness of a KR rearrangement. ■

Appendix B. Proofs of the Main Results

In this section we collect the proofs of the main results and claims of the paper, together with useful additional lemmas to support the technical derivations.

Proof of Lemma 2 The general solution of $\partial^2_{i,j} \log \pi = 0$ on $\mathbb{R}^n$ is given by $\log \pi(\boldsymbol{z}) = g(\boldsymbol{z}_{1:i-1}, \boldsymbol{z}_{i+1:n}) + h(\boldsymbol{z}_{1:j-1}, \boldsymbol{z}_{j+1:n})$ for some functions $g, h : \mathbb{R}^{n-1} \to \mathbb{R}$. Hence $Z_i \perp\!\!\!\perp Z_j \,|\, \boldsymbol{Z}_{\mathcal{V}\setminus(i,j)}$ (Lauritzen, 1996). Conversely, if $Z_i \perp\!\!\!\perp Z_j \,|\, \boldsymbol{Z}_{\mathcal{V}\setminus(i,j)}$, then π—which is the

12. See Definition 6.

density of $\boldsymbol{\nu}_\pi$ with respect to a tensor product Lebesgue measure (Lauritzen, 1996)—must factor as

$$\pi = \pi_{Z_i|\boldsymbol{Z}_{\mathcal{V}\setminus(i,j)}} \pi_{Z_j|\boldsymbol{Z}_{\mathcal{V}\setminus(i,j)}} \pi_{\boldsymbol{Z}_{\mathcal{V}\setminus(i,j)}},$$

so that $\partial^2_{i,j} \log \pi = 0$ on $\mathbb{R}^n$. ■

Proof of Theorem 3 We begin with Part 1 of the theorem. Let η, π be a pair of strictly positive densities for $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$, respectively (these positive densities exist since the measures are fully supported). Now consider a *version* of the KR rearrangement, S, that pushes forward $\boldsymbol{\nu}_\pi$ to $\boldsymbol{\nu}_\eta$ as given by Definition 14 for the pair η, π (Appendix A). By definition, and for all $\boldsymbol{z}_{1:k-1} \in \mathbb{R}^{k-1}$, the map $\xi \mapsto S^k(\boldsymbol{z}_{1:k-1}, \xi)$ is the monotone increasing rearrangement that pushes forward $\xi \mapsto \pi_{Z_k|\boldsymbol{Z}_{1:k-1}}(\xi|\boldsymbol{z}_{1:k-1})$ to the marginal η_{X_k} (recall that $\boldsymbol{\nu}_\eta$ is a tensor product measure). Moreover, it follows easily from (Lauritzen, 1996, Prop. 3.17), that each marginal $\pi_{\boldsymbol{Z}_{1:k}}$—or better yet, the corresponding measure—is globally Markov with respect to $\boldsymbol{\mathcal{G}}^k$, and that $\pi_{\boldsymbol{Z}_{1:k}}(\boldsymbol{z}_{1:k})\,\pi_{\mathcal{C}}(\boldsymbol{z}_{\mathcal{C}}) = \pi_{\boldsymbol{Z}_k,\boldsymbol{Z}_{\mathcal{C}}}(\boldsymbol{z}_k, \boldsymbol{z}_{\mathcal{C}})\,\pi_{\boldsymbol{Z}_{1:k-1}}(\boldsymbol{z}_{1:k-1})$, where $\mathcal{C} \coloneqq \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$, possibly empty. Thus, the conditional $\pi_{Z_k|\boldsymbol{Z}_{1:k-1}}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})$ is constant along any input $\boldsymbol{z}_j$ with $j \notin \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$. For any such j, S^k must be constant along its jth input, so that $(j,k) \in \widehat{\mathfrak{I}}_S$.

Part 2 of the theorem follows similarly. Consider the KR rearrangement, T, that pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_\pi$ as given by Definition 14. For all $\boldsymbol{x}_{1:k-1} \in \mathbb{R}^{k-1}$, the map $\xi \mapsto T^k(\boldsymbol{x}_{1:k-1}, \xi)$ is the monotone increasing rearrangement that pushes forward η_{X_k} to

$$\xi \mapsto \pi_{Z_k|\boldsymbol{Z}_{1:k-1}}(\xi|T^1(x_1), \ldots, T^{k-1}(\boldsymbol{x}_{1:k-1})).$$

We already know that $\pi_{Z_k|\boldsymbol{Z}_{1:k-1}}(\boldsymbol{z}_k|\boldsymbol{z}_{1:k-1})$ can only depend (nontrivially) on $\boldsymbol{z}_k$ and on $\boldsymbol{z}_j$ for $j \in \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$. Hence, if none of the components T^i, with $i \in \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$, depends on the jth input, then T^k is constant along its jth input as well, so that $(j,k) \in \widehat{\mathfrak{I}}_T$.

For Part 3, let $(j,k) \in \widehat{\mathfrak{I}}_T$. Then, by definition, $(j,i) \in \widehat{\mathfrak{I}}_T$ for all $i \in \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$, which also implies that $j \notin \mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k)$ since $j \neq i$ for all $(j,i) \in \widehat{\mathfrak{I}}_T$. Hence $(j,k) \in \widehat{\mathfrak{I}}_S$ and this shows the inclusion $\widehat{\mathfrak{I}}_T \subset \widehat{\mathfrak{I}}_S$.

These arguments show that there exists at least a *version* of the KR rearrangement that is exactly at least as sparse as predicted by the theorem. ■

The following lemma specializes the results of Theorem 3[Part 2] to the case of I-maps $\boldsymbol{\mathcal{G}}$ with a disconnected component, and will be useful in the proofs of Section 6.

Lemma 18 *Let $\boldsymbol{X} \sim \boldsymbol{\nu}_\eta$, $\boldsymbol{Z} \sim \boldsymbol{\nu}_\pi$ with $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_\pi \in \mathscr{M}_+(\mathbb{R}^n)$ and $\boldsymbol{\nu}_\eta$ tensor product measure, and let σ be any permutation of $\mathbb{N}_n$. Moreover, assume that $\boldsymbol{\nu}_\pi$ is globally Markov with respect to $\boldsymbol{\mathcal{G}} = (\mathcal{V}, \mathcal{E})$, and assume that there exists a nonempty set $\mathcal{A} \subset \mathcal{V} \simeq \mathbb{N}_n$ such that $\boldsymbol{Z}_\mathcal{A} \perp\!\!\!\perp \boldsymbol{Z}_{\mathcal{V}\setminus\mathcal{A}}$ and $\boldsymbol{Z}_\mathcal{A} = \boldsymbol{X}_\mathcal{A}$ in distribution. Then the σ-generalized KR rearrangement T given by Definition 16 (for a pair η, π of nonvanishing densities for $\boldsymbol{\nu}_\eta$ and $\boldsymbol{\nu}_\pi$, respectively) is low-dimensional with respect to $\mathcal{A}$, i.e.,*

1. *$T^k(\boldsymbol{x}) = x_k$ for $k \in \mathcal{A}$*
2. *$\partial_j T^k = 0$ for $j \in \mathcal{A}$ and $k \in \mathcal{V} \setminus \mathcal{A}$.*

Proof It suffices to prove the lemma for a lower triangular KR rearrangement; the result for an arbitrary σ then follows trivially. If $\mathcal{A} = \mathcal{V}$, then T is simply the identity map. Thus we assume that $\mathcal{V} \setminus \mathcal{A}$ is nonempty.

We begin with Part 1 of the lemma and use the results of Theorem 3[Part 2] to characterize the sparsity of the rearrangement. Let $k \in \mathcal{A}$ and notice that $\mathrm{Nb}(k, \boldsymbol{\mathcal{G}}^k) = \emptyset$, where $\boldsymbol{\mathcal{G}}^k$ is the marginal graph defined in Theorem 3. Thus $(j,k) \in \widehat{\mathfrak{I}}_T \subset \mathfrak{I}_T$ for all $j = 1, \ldots, k-1$, so that $T^k(\boldsymbol{x}) = x_k$ for all $k \in \mathcal{A}$.

Now let us focus on Part 2 and prove that $(j,k) \in \widehat{\mathfrak{I}}_T$ for all $j \in \mathcal{A}$ and $k \in \mathcal{V} \setminus \mathcal{A}$. We proceed by contradiction. Assume that there exists some pair $(j,k) \in \mathcal{A} \times (\mathcal{V} \setminus \mathcal{A})$ such that $(j,k) \notin \widehat{\mathfrak{I}}_T$. In particular, let $\boldsymbol{\mathcal{K}}$ be the set of $k \in \mathcal{V} \setminus \mathcal{A}$ for which there exists at least a $j \in \mathcal{A}$ such that $(j,k) \notin \widehat{\mathfrak{I}}_T$. Clearly $\boldsymbol{\mathcal{K}}$ is nonempty and finite. Let s be the minimum integer in $\boldsymbol{\mathcal{K}}$, and let $j \in \mathcal{A}$ be a corresponding index for which $(j,s) \notin \widehat{\mathfrak{I}}_T$. In this case, by Theorem 3[Part 2], there must exist an $i \in \mathrm{Nb}(s, \boldsymbol{\mathcal{G}}^s)$ such that $(j,i) \notin \widehat{\mathfrak{I}}_T$. Now there are two cases: either $i \in \mathcal{A}$ (for which we reach a contradiction by part 1 of the lemma) or $i \in \mathcal{V} \setminus \mathcal{A}$. In the latter case, we also reach a contradiction since $i < s$ and s was defined as the smallest index for which $(j,s) \notin \widehat{\mathfrak{I}}_T$ for some $j \in \mathcal{A}$. ■

Proof of Theorem 7 For notational convenience, we drop the subscript and superscript i from $\boldsymbol{\nu}_i$, π_i, $\boldsymbol{Z}^i$, and $\boldsymbol{\mathcal{G}}^i$. Consider a factorization of π of the form

$$\pi(\boldsymbol{z}) = \frac{1}{\mathfrak{c}}\, \psi_{\mathcal{A}\cup\mathcal{S}}(\boldsymbol{z}_{\mathcal{A}\cup\mathcal{S}})\, \psi_{\mathcal{S}\cup\mathcal{B}}(\boldsymbol{z}_{\mathcal{S}\cup\mathcal{B}}), \tag{37}$$

where $\psi_{\mathcal{A}\cup\mathcal{S}}$ is strictly positive and integrable, with $\mathfrak{c} = \int \psi_{\mathcal{A}\cup\mathcal{S}} < \infty$. A factorization like (37) always exist since $\boldsymbol{\nu}$ factorizes according to $\boldsymbol{\mathcal{G}}$—thus $\boldsymbol{\mathcal{G}}$ is an I-map for $\boldsymbol{\nu}$—and since $(\mathcal{A}, \mathcal{S}, \mathcal{B})$ is a proper decomposition of $\boldsymbol{\mathcal{G}}$. For instance, one can set $\psi_{\mathcal{A}\cup\mathcal{S}} = \pi_{\boldsymbol{Z}_{\mathcal{A}\cup\mathcal{S}}}$, $\mathfrak{c} = 1$, and $\psi_{\mathcal{S}\cup\mathcal{B}} = \pi_{\boldsymbol{Z}_\mathcal{B}|\boldsymbol{Z}_\mathcal{S}}$ since $\boldsymbol{Z}_\mathcal{A} \perp\!\!\!\perp \boldsymbol{Z}_\mathcal{B}|\boldsymbol{Z}_\mathcal{S}$ and since π is a nonvanishing density of $\boldsymbol{\nu}$. However, this is not the only possibility. See Section 7 for important examples where it is not convenient to assume that $\psi_{\mathcal{A}\cup\mathcal{S}}$ corresponds to a marginal of π. This proves Part 1 of the theorem.

By (Lauritzen, 1996, Prop. 3.16), we can rewrite $\psi_{\mathcal{S}\cup\mathcal{B}}$ as:

$$\psi_{\mathcal{S}\cup\mathcal{B}}(\boldsymbol{z}_{\mathcal{S}\cup\mathcal{B}}) = \prod_{\mathcal{C}\in\boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}} \psi_\mathcal{C}(\boldsymbol{z}_\mathcal{C}) \tag{38}$$

for some nonvanishing functions $(\psi_\mathcal{C})$, where $\boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}$ denotes the set of maximal cliques of the subgraph $\boldsymbol{\mathcal{G}}_{\mathcal{S}\cup\mathcal{B}}$. Since $\mathcal{S}$ is a fully connected separator set (possibly empty) for $\mathcal{A}$ and $\mathcal{B}$, the maximal cliques of $\boldsymbol{\mathcal{G}}_{\mathcal{S}\cup\mathcal{B}}$ are precisely the maximal cliques of $\boldsymbol{\mathcal{G}}$ that are a subset of $\mathcal{S}\cup\mathcal{B}$. We are going to use (38) shortly.

Define $\widetilde{\pi} : \mathbb{R}^n \to \mathbb{R}$ as $\widetilde{\pi}(\boldsymbol{z}) = \psi_{\mathcal{A}\cup\mathcal{S}}(\boldsymbol{z}_{\mathcal{A}\cup\mathcal{S}})\, \eta_{\boldsymbol{X}_\mathcal{B}}(\boldsymbol{z}_\mathcal{B})/\mathfrak{c}$, and notice that $\widetilde{\pi}$ is a nonvanishing probability density. Denote the corresponding measure by $\widetilde{\boldsymbol{\nu}} \in \mathscr{M}_+(\mathbb{R}^n)$. For an arbitrary permutation σ of $\mathbb{N}_n$ that satisfies (18), let L_i be the σ-generalized KR rearrangement that pushes forward $\boldsymbol{\nu}_\eta$ to $\widetilde{\boldsymbol{\nu}}$ as given by Definition 16 in Appendix A. By Lemma 18, L_i is low-dimensional with respect to $\mathcal{B}$ (Part 2a of the theorem). To see this, let $\widetilde{\boldsymbol{Z}} \sim \widetilde{\boldsymbol{\nu}}$, and notice that $\widetilde{\boldsymbol{Z}}_\mathcal{B} \perp\!\!\!\perp \widetilde{\boldsymbol{Z}}_{\mathcal{A}\cup\mathcal{S}}$ and $\widetilde{\boldsymbol{Z}}_\mathcal{B} = \boldsymbol{X}_\mathcal{B}$ in distribution. By Lemma 15, we can write a

density of the pullback measure $L_i^\sharp \boldsymbol{\nu}$ as:

$$\begin{aligned} L_i^\sharp \pi &= \pi \circ L_i \, |\det \nabla L_i| \\ &= \left(L_i^\sharp \widetilde{\pi}\right) \frac{\prod_{\mathcal{C}\in\boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}} \psi_{\mathcal{C}} \circ L_i^{\mathcal{C}}}{\eta_{\boldsymbol{X}_{\mathcal{B}}}} \\ &= \eta_{\boldsymbol{X}_{\mathcal{A}\cup\mathcal{S}}} \prod_{\mathcal{C}\in\boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}} \psi_{\mathcal{C}} \circ L_i^{\mathcal{C}}, \end{aligned} \tag{39}$$

where we used the identity $\pi = \widetilde{\pi}\, \psi_{\mathcal{S}\cup\mathcal{B}}/\eta_{\boldsymbol{X}_{\mathcal{B}}}$ together with (38) and the fact that $L_i^k(\boldsymbol{x}) = x_k$ for $k \in \mathcal{B}$ (Part 2a), and where, for any $\mathcal{C} = \{c_1, \dots, c_\ell\} \in \boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}$ with $\psi_{\mathcal{C}}(\boldsymbol{z}_{\mathcal{C}}) = \psi_{\mathcal{C}}(z_{c_1}, \dots, z_{c_\ell})$, $L_i^{\mathcal{C}}$ is a map $\mathbb{R}^n \to \mathbb{R}^\ell$ given by $\boldsymbol{x} \mapsto (L_i^{c_1}(\boldsymbol{x}), \dots, L_i^{c_\ell}(\boldsymbol{x}))$.

If $\boldsymbol{Z}' \sim L_i^\sharp \boldsymbol{\nu}$, then (39) shows that $\boldsymbol{Z}'_{\mathcal{A}} \perp\!\!\!\perp \boldsymbol{Z}'_{\mathcal{S}\cup\mathcal{B}}$ and that $\boldsymbol{Z}'_{\mathcal{A}} = \boldsymbol{X}_{\mathcal{A}}$ in distribution (Part 2c of the theorem). Moreover, from the factorization in (39), we can easily construct a graph for which $L_i^\sharp \boldsymbol{\nu}$ factorizes: it suffices to consider the scope of the factors $(\psi_{\mathcal{C}} \circ L_i^{\mathcal{C}})$, i.e., the indices of the input variables that each $\psi_{\mathcal{C}} \circ L_i^{\mathcal{C}}$ can depend on. Recall that for a σ-triangular map, the $\sigma(k)$th component can only depend on the variables $x_{\sigma(1)}, \dots, x_{\sigma(k)}$. For each $\mathcal{C} \in \boldsymbol{\mathcal{C}}_{\mathcal{S}\cup\mathcal{B}}$ there are two possibilites: Either $\mathcal{C} \cap \mathcal{S} = \emptyset$, in which case the scope of $\psi_{\mathcal{C}} \circ L_i^{\mathcal{C}}$ is simply $\mathcal{C}$ since $L_i^k(\boldsymbol{x}) = x_k$ for $k \in \mathcal{B}$. Or $\mathcal{C} \cap \mathcal{S}$ is nonempty, in which case let $j_{\mathcal{C}}$ be the maximum integer j such that $\sigma(j) \in \mathcal{C} \cap \mathcal{S}$, and notice that the scope of $\psi_{\mathcal{C}} \circ L_i^{\mathcal{C}}$ is simply $\mathcal{C} \cup \{\sigma(1), \dots, \sigma(j_{\mathcal{C}})\}$. Thus, we can modify $\boldsymbol{\mathcal{G}}$ to obtain an I-map for $L_i^\sharp \boldsymbol{\nu}$ as follows: (1) Remove any edge that is incident to any node in $\mathcal{A}$ because of Part 2c. (2) For every maximal clique $\mathcal{C}$ in $\boldsymbol{\mathcal{G}}$ that is a subset of $\mathcal{S} \cup \mathcal{B}$ and that has nonempty intersection with $\mathcal{S}$, turn $\mathcal{C} \cup \{\sigma(1), \dots, \sigma(j_{\mathcal{C}})\}$ into a clique. This proves Part 2d of the theorem.

Now let $\mathfrak{R}_i$ be the set of maps $\mathbb{R}^n \to \mathbb{R}^n$ that are low-dimensional with respect to $\mathcal{A}$ and that push forward $\boldsymbol{\nu}_\eta$ to $L_i^\sharp \boldsymbol{\nu}$. $\mathfrak{R}_i$ is nonempty. To see this, let R be the σ-generalized KR rearrangement that pushes forward $\boldsymbol{\nu}_\eta$ to $L_i^\sharp \boldsymbol{\nu}$, for an arbitrary permutation σ, as given by Definition 16 (for the pair of nonvanishing densities η and $L_i^\sharp \pi$). By Part 2c and Lemma 18, R is low-dimensional with respect to $\mathcal{A}$. Thus $R \in \mathfrak{R}_i$ (Part 2b of the theorem).

Let $\mathfrak{D}_i \coloneqq L_i \circ \mathfrak{R}_i$ be the set of maps that can be written as $L_i \circ R$ for some $R \in \mathfrak{R}_i$. By construction, each $T \in \mathfrak{D}_i$ pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}$ (part 2 of the theorem). ■

In the following corollary every symbol should be interpreted as in Theorem 7.

Corollary 19 *Given the hypothesis of Theorem 7, assume that there exists $\mathcal{A}^\perp \subset \mathcal{A}$ such that $\boldsymbol{Z}^i_{\mathcal{A}^\perp} \perp\!\!\!\perp \boldsymbol{Z}^i_{\mathcal{V}\setminus\mathcal{A}^\perp}$ and $\boldsymbol{Z}^i_{\mathcal{A}^\perp} = \boldsymbol{X}_{\mathcal{A}^\perp}$ in distribution. Then L_i is low-dimensional with respect to $\mathcal{A}^\perp \cup \mathcal{B}$, while each $T \in \mathfrak{D}_i$ is low-dimensional with respect to $\mathcal{A}^\perp$.*

Proof By Theorem 7[Part 2a], L_i is low-dimensional with respect to $\mathcal{B}$, while Lemma 18 shows that L_i is also low-dimensional with respect to $\mathcal{A}^\perp$. Moreover, notice that if $\mathcal{A}^\perp$ is nonempty, then for all $T = L_i \circ R$ in $\mathfrak{D}_i$, we have $T^k(\boldsymbol{x}) = x_k$ for $k \in \mathcal{A}^\perp$ since $L_i^k(\boldsymbol{x}) = x_k$ and $R^k(\boldsymbol{x}) = x_k$ for $k \in \mathcal{A}^\perp$ (Theorem 7[Parts 2b]). Additionally, $\partial_j T^k = 0$ for $j \in \mathcal{A}^\perp$ and $k \in \mathcal{V} \setminus \mathcal{A}^\perp$. To see this, notice that $T^k(\boldsymbol{x}) = L_i^k(R(\boldsymbol{x}))$ and that the following two facts hold: (1) The component L_i^k, for $k \in \mathcal{V} \setminus \mathcal{A}^\perp$, does not depend on input variables whose index is in $\mathcal{A}^\perp$ since L_i is low-dimensional with respect to $\mathcal{A}^\perp$; (2) The ℓth component of

R with $\ell \notin \mathcal{A}^\perp$ also does not depend on $\boldsymbol{x}_{\mathcal{A}^\perp}$ since R is low-dimensional with respect to $\mathcal{A}$ (Theorem 7[Parts 2b]). Hence, T must be a low-dimensional map with respect to $\mathcal{A}^\perp$. ■

Proof of Lemma 8 Let $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_i, \pi_i, \boldsymbol{\mathcal{G}}^i, \mathfrak{D}_i, L_i, \mathfrak{R}_i$, and $\boldsymbol{\mathcal{G}}^{i+1}$ be defined as in Theorem 7 for a proper decomposition $(\mathcal{A}_i, \mathcal{S}_i, \mathcal{B}_i)$ of $\boldsymbol{\mathcal{G}}^i$, a permutation σ_i that satisfies (18), and for any factorization (17) of π_i.

We first want to prove that $\mathcal{S}_i \cup \mathcal{B}_i$ is fully connected in $\boldsymbol{\mathcal{G}}^{i+1}$ if and only if the decomposition $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ of Part 1 does not exist. Let us start with one direction. Assume that a decomposition like the one in Part 1 does not exist, despite the possibility to add edges to $\boldsymbol{\mathcal{G}}^{i+1}$ in $\mathcal{V} \setminus \mathcal{A}_i$. We want to show that in this case $\mathcal{S}_i \cup \mathcal{B}_i$ must be a clique in $\boldsymbol{\mathcal{G}}^{i+1}$. Since $\mathcal{B}_i$ is nonempty, there are two possibilities: either $|\mathcal{S}_i \cup \mathcal{B}_i| = 1$ or $|\mathcal{S}_i \cup \mathcal{B}_i| > 1$. If $|\mathcal{S}_i \cup \mathcal{B}_i| = 1$, then $\mathcal{S}_i \cup \mathcal{B}_i$ consists of a single node and thus it is a trivial clique. If $|\mathcal{S}_i \cup \mathcal{B}_i| > 1$, then $\mathcal{S}_i \cup \mathcal{B}_i$ contains at least two nodes. In this case, let us proceed by contradiction and assume that $\mathcal{S}_i \cup \mathcal{B}_i$ is not fully connected in $\boldsymbol{\mathcal{G}}^{i+1} = (\mathcal{V}, \mathcal{E}^{i+1})$, i.e., there exist a pair of nodes $\alpha, \beta \in \mathcal{S}_i \cup \mathcal{B}_i$ such that $(\alpha, \beta) \notin \mathcal{E}^{i+1}$. Let $\mathcal{A}_{i+1} = \mathcal{A}_i \cup \{\alpha\}$, $\mathcal{B}_{i+1} = \{\beta\}$, and $\mathcal{S}_{i+1} = (\mathcal{V} \setminus \mathcal{A}_{i+1}) \setminus \mathcal{B}_{i+1}$. Notice that $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ forms a partition of $\mathcal{V}$, with nonempty $\mathcal{A}_{i+1}, \mathcal{B}_{i+1}$ and with $\mathcal{A}_{i+1}$ strict superset of $\mathcal{A}_i$. Moreover $\mathcal{S}_{i+1}$ must be a separator set for $\mathcal{A}_{i+1}$ and $\mathcal{B}_{i+1}$ since $(\alpha, \beta) \notin \mathcal{E}^{i+1}$ and $\mathcal{A}_i$ is disconnected from $\mathcal{S}_i \cup \mathcal{B}_i$ in $\boldsymbol{\mathcal{G}}^{i+1}$ (Theorem 7[Part 2d]). Now there are two cases: If $\mathcal{S}_{i+1} = \emptyset$, then $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ is a decomposition that satisfies Part 1 of the lemma (contradiction). If $\mathcal{S}_{i+1} \neq \emptyset$, then we can always add enough edges to $\boldsymbol{\mathcal{G}}^{i+1}$ in $\mathcal{S}_i \cup \mathcal{B}_i \supset \mathcal{S}_{i+1}$ in order to make $\mathcal{S}_{i+1}$ fully connected. Also in this case, the resulting decomposition $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ satisfies Part 1 of the lemma and thus leads to a contradiction.

Now the reverse direction. Assume that $\mathcal{S}_i \cup \mathcal{B}_i$ is a clique in $\boldsymbol{\mathcal{G}}^{i+1}$. If $|\mathcal{S}_i \cup \mathcal{B}_i| = 1$, then the decomposition of Part 1 cannot exist since both $\mathcal{A}_{i+1} \setminus \mathcal{A}_i$ and $\mathcal{B}_{i+1}$ should be nonempty. Hence, let $|\mathcal{S}_i \cup \mathcal{B}_i| > 1$ and proceed by contradiction. That is, let $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ be a proper decomposition that satisfies Part 1 of the lemma. Notice that this decomposition must have been achieved without adding any edge to $\boldsymbol{\mathcal{G}}^{i+1}$ in $\mathcal{S}_i \cup \mathcal{B}_i$ since this set is already fully connected. By hypothesis, there must exist α, β such that $\alpha \in \mathcal{A}_{i+1} \setminus \mathcal{A}_i$ and $\beta \in \mathcal{B}_{i+1}$. However, both α and β are also in $\mathcal{S}_i \cup \mathcal{B}_i$, and so they must be connected by an edge in $\boldsymbol{\mathcal{G}}^{i+1}$. Hence, $\mathcal{S}_{i+1}$ is not a separator set for $\mathcal{A}_{i+1}$ and $\mathcal{B}_{i+1}$ (contradiction).

The latter result proves directly Part 2 of the lemma. Moreover, it shows that if $\mathcal{S}_i \cup \mathcal{B}_i$ is not a clique in $\boldsymbol{\mathcal{G}}^{i+1}$, then there exists a proper decomposition $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ of $\boldsymbol{\mathcal{G}}^{i+1}$, where $\mathcal{A}_{i+1}$ is a strict superset of $\mathcal{A}_i$, obtained, possibly, by adding edges to $\boldsymbol{\mathcal{G}}^{i+1}$ in order to turn $\mathcal{S}_{i+1}$ into a clique. Note that even if we add edges to $\boldsymbol{\mathcal{G}}^{i+1}$, $L_i^\sharp \boldsymbol{\nu}_i$ still factorizes according to the resulting graph, which is then an I-map for $L_i^\sharp \boldsymbol{\nu}_i$. Moreover we can really only add edges in $\mathcal{V} \setminus \mathcal{A}_i$ since $\mathcal{A}_i$ must be a strict subset of $\mathcal{A}_{i+1}$, and thus $\mathcal{A}_i$ remains disconnected from $\mathcal{S}_i \cup \mathcal{B}_i$ in $\boldsymbol{\mathcal{G}}^{i+1}$. Let $\mathfrak{D}_{i+1}, L_{i+1}, \mathfrak{R}_{i+1}$ be defined as in Theorem 7 for the pair of measures $\boldsymbol{\nu}_\eta, \boldsymbol{\nu}_{i+1} = L_i^\sharp \boldsymbol{\nu}_i$, the decomposition $(\mathcal{A}_{i+1}, \mathcal{S}_{i+1}, \mathcal{B}_{i+1})$ of $\boldsymbol{\mathcal{G}}^{i+1}$, a permutation σ_{i+1} that satisfies (18), and for any factorization (17) (note that $L_i^\sharp \boldsymbol{\nu}_i \in \mathscr{M}_+(\mathbb{R}^n)$ by Theorem 7[Part 2b]). Fix $T \in \mathfrak{D}_{i+1}$. By Theorem 7[Part 2], T pushes forward $\boldsymbol{\nu}_\eta$ to $\boldsymbol{\nu}_{i+1} = L_i^\sharp \boldsymbol{\nu}_i$. Moreover, if $\boldsymbol{Z}^{i+1} \sim L_i^\sharp \boldsymbol{\nu}_i$, then by Theorem 7[Part 2c] we have $\boldsymbol{Z}^{i+1}_{\mathcal{A}_i} \perp\!\!\!\perp \boldsymbol{Z}^{i+1}_{\mathcal{S}_i\cup\mathcal{B}_i}$ and $\boldsymbol{Z}^{i+1}_{\mathcal{A}_i} = \boldsymbol{X}_{\mathcal{A}_i}$ in distribution. Then by Corollary 19 it must also be that T is low-dimensional with respect to $\mathcal{A}_i$. Thus $T \in \mathfrak{R}_i$, and this proves the inclusion $\mathfrak{R}_i \supset \mathfrak{D}_{i+1}$.

Now fix any $T \in L_i \circ L_{i+1} \circ \mathfrak{R}_{i+1} = L_i \circ \mathfrak{D}_{i+1}$. It must be that $T = L_i \circ g$ for some $g \in \mathfrak{D}_{i+1} \subset \mathfrak{R}_i$, so that $T \in L_i \circ \mathfrak{R}_i$, which shows the inclusion $L_i \circ \mathfrak{R}_i \supset L_i \circ L_{i+1} \circ \mathfrak{R}_{i+1}$ (Part 1a of the lemma). By Corollary 19, we have that L_{i+1} is low-dimensional with respect to $\mathcal{A}_i \cup \mathcal{B}_{i+1}$, and so its effective dimension is bounded above by $|\mathcal{V} \setminus (\mathcal{A}_i \cup \mathcal{B}_{i+1})| = |(\mathcal{A}_{i+1} \setminus \mathcal{A}_i) \cup \mathcal{S}_{i+1}|$ (Part 1b). Finally, by Theorem 7[Part 2b], each $R \in \mathfrak{R}_{i+1}$ is low-dimensional with respect to $\mathcal{A}_{i+1}$, and so its effective dimension is bounded by $|\mathcal{V} \setminus \mathcal{A}_{i+1}|$ (Part 1c). ■

Proof of Theorem 9 For the sake of clarity, we divide the proof in two parts: First, we show that the maps $(\mathfrak{M}_i)_{i\geq 0}$ are well-defined. Then, we prove the remaining claims of the theorem.

The maps $(\mathfrak{M}_i)_{i\geq 0}$ are well-defined as long as, for instance, we show that π^i is a probability density for all $i \geq 0$, and as long as there exist permutations (σ_i) that guarantee the block upper triangular structure of (23). As for the permutations, it suffices to consider $\sigma = \sigma_1 = \sigma_2 = \cdots$ with $\sigma(\mathbb{N}_{2n}) = \{2n, 2n-1, \ldots, 1\}$, i.e., upper triangular maps. (If $n > 1$, then there is some freedom in the choice of σ.) As for the targets (π^i), we now show that π^i is a nonvanishing density and that the marginal $\int \pi^i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1}) \,\mathrm{d}\boldsymbol{z}_i = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$, for all $i \geq 0$, using an induction argument over i. For the base case ($i = 0$), just notice that

$$\mathfrak{c}_0 = \int \widetilde{\pi}^0(\boldsymbol{z}_0, \boldsymbol{z}_1) \,\mathrm{d}\boldsymbol{z}_{0:1} = \pi_{\boldsymbol{Y}_0,\boldsymbol{Y}_1}(\boldsymbol{y}_0, \boldsymbol{y}_1) < \infty, \tag{40}$$

so that $\pi^0 = \widetilde{\pi}^0/\mathfrak{c}_0 > 0$ is a valid density. Moreover, we have the desired marginal, i.e.,

$$\int \pi^0(\boldsymbol{z}_0, \boldsymbol{z}_1) \,\mathrm{d}\boldsymbol{z}_0 = \int \pi_{\boldsymbol{Z}_0,\boldsymbol{Z}_1|\boldsymbol{Y}_0,\boldsymbol{Y}_1}(\boldsymbol{z}_0, \boldsymbol{z}_1|\boldsymbol{y}_0, \boldsymbol{y}_1) \,\mathrm{d}\boldsymbol{z}_0 = \pi_{\boldsymbol{Z}_1|\boldsymbol{Y}_0,\boldsymbol{Y}_1}(\boldsymbol{z}_1|\boldsymbol{y}_0, \boldsymbol{y}_1).$$

Now assume that π^i is a nonvanishing density and that the marginal $\int \pi^i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1}) \,\mathrm{d}\boldsymbol{z}_i = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ for some $i > 0$. The map $\mathfrak{M}_i$ is then well-defined. In particular, by definition of KR rearrangement, the submap $\mathfrak{M}_i^1$ pushes forward $\eta_{\boldsymbol{X}_{i+1}}$ to the marginal $\int \pi^i(\boldsymbol{z}_i, \boldsymbol{z}_{i+1}) \,\mathrm{d}\boldsymbol{z}_i$. Moreover, by Lemma 15, we have:

$$\begin{aligned}
\mathfrak{c}_{i+1} &= \int \eta_{\boldsymbol{X}_{i+1}}(\boldsymbol{z}_{i+1}) \, \widetilde{\pi}^{i+1}(\mathfrak{M}_i^1(\boldsymbol{z}_{i+1}), \boldsymbol{z}_{i+2}) \,\mathrm{d}\boldsymbol{z}_{i+1:i+2} \\
&= \int \pi_{\boldsymbol{Z}_{i+2},\boldsymbol{Y}_{i+2}|\boldsymbol{Y}_{0:i+1}}(\boldsymbol{z}_{i+2}, \boldsymbol{y}_{i+2}|\boldsymbol{y}_{0:i+1}) \,\mathrm{d}\boldsymbol{z}_{i+2} \\
&= \pi_{\boldsymbol{Y}_{i+2}|\boldsymbol{Y}_{0:i+1}}(\boldsymbol{y}_{i+2}|\boldsymbol{y}_{0:i+1}) < \infty,
\end{aligned} \tag{41}$$

where we used the change of variables $\boldsymbol{x}_{i+1} = \mathfrak{M}_i^1(\boldsymbol{z}_{i+1})$ and the fact that $(\mathfrak{M}_i^1)_\sharp \, \eta_{\boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ (induction hypothesis). Thus π^{i+1} is a nonvanishing density and by (41) we can easily verify that π^{i+1} has the desired marginal, i.e., $\int \pi^{i+1}(\boldsymbol{z}_{i+1}, \boldsymbol{z}_{i+2}) \,\mathrm{d}\boldsymbol{z}_{i+1} = \pi_{\boldsymbol{Z}_{i+2}|\boldsymbol{y}_{0:i+2}}$. This argument completes the induction step and shows that not only the maps $(\mathfrak{M}_i)_{i\geq 0}$ are well-defined—together with the maps $(T_i)_{i\geq 0}$ in (25)—but also that $(\mathfrak{M}_i^1)_\sharp \, \eta_{\boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ for all $i \geq 0$ (Part 1 of the theorem).

Now we move to Part 3 of the theorem and use another induction argument over $k \geq 0$. For the base case ($k = 0$), notice that $\mathfrak{T}_0 = T_0 = \mathfrak{M}_0$, and that, by definition, $\mathfrak{M}_0$ pushes forward $\eta_{\boldsymbol{X}_0,\boldsymbol{X}_1}$ to $\pi^0 = \pi_{\boldsymbol{Z}_0,\boldsymbol{Z}_1|\boldsymbol{y}_0,\boldsymbol{y}_1}$.

Assume that $\mathfrak{T}_k$ pushes forward $\eta_{\boldsymbol{X}_{0:k+1}}$ to $\pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ for some $k > 0$ ($\mathfrak{T}_k$ is well-defined for all k since the maps $(T_i)_{i\geq 0}$ in (25) are also well-defined), and notice that

$$\pi_{\boldsymbol{Z}_{0:k+2}|\boldsymbol{y}_{0:k+2}} = \pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}} \frac{\pi_{\boldsymbol{y}_{k+2}|\boldsymbol{Z}_{k+2}} \, \pi_{\boldsymbol{Z}_{k+2}|\boldsymbol{Z}_{k+1}}}{\pi_{\boldsymbol{y}_{k+2}|\boldsymbol{y}_{0:k+1}}} = \pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}} \frac{\widetilde{\pi}^{k+1}}{\mathfrak{c}_{k+1}},$$

where we used (41) and the definition of the collection $(\widetilde{\pi}^i)$. Let $\mathfrak{T}_{k+1} = T_0 \circ \cdots \circ T_{k+1}$ be defined as in Part 3 of the theorem, and observe that $\mathfrak{T}_{k+1} = A_{k+1} \circ T_{k+1}$ with

$$A_{k+1}(\boldsymbol{x}_{0:k+2}) = \begin{bmatrix} \mathfrak{T}_k(\boldsymbol{x}_{0:k+1}) \\ \boldsymbol{x}_{k+2} \end{bmatrix}, \quad T_{k+1}(\boldsymbol{x}_{0:k+2}) = \begin{bmatrix} \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_k \\ \mathfrak{M}_{k+1}^0(\boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}) \\ \mathfrak{M}_{k+1}^1(\boldsymbol{x}_{k+2}) \end{bmatrix}.$$

Thus the following hold:

$$\begin{aligned}
\mathfrak{T}_{k+1}^\sharp \, \pi_{\boldsymbol{Z}_{0:k+2}|\boldsymbol{Y}_{0:k+2}} &= T_{k+1}^\sharp \left(\left(\mathfrak{T}_k^\sharp \, \pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}} \right) \frac{\pi^{k+1}}{\eta_{\boldsymbol{X}_{k+1}}} \right) \\
&= T_{k+1}^\sharp \left(\eta_{\boldsymbol{X}_{0:k}} \, \pi^{k+1} \right) \\
&= \eta_{\boldsymbol{X}_{0:k}} \, \mathfrak{M}_{k+1}^\sharp \, \pi^{k+1} = \eta_{\boldsymbol{X}_{0:k+2}},
\end{aligned}$$

where we used the fact that by Lemma 15 (applied iteratively) it must be that $(A_{k+1} \circ T_{k+1})^\sharp \rho = T_{k+1}^\sharp A_{k+1}^\sharp \rho$ for all densities ρ. (Notice that A_{k+1} is the composition of functions which are trivial embeddings into the identity map of KR rearrangements that couple a pair of measures in $\mathscr{M}_+(\mathbb{R}^n \times \mathbb{R}^n)$, and thus each map in the composition satisfies the hypothesis of Lemma 15.) In particular, $(\mathfrak{T}_{k+1})_\sharp \, \eta_{\boldsymbol{X}_{0:k+2}} = \pi_{\boldsymbol{Z}_{0:k+2}|\boldsymbol{y}_{0:k+2}}$ (Part 3 of the theorem).

Now notice that each $\mathfrak{T}_k$ can also be written as

$$\mathfrak{T}_k(\boldsymbol{x}_{0:k+1}) = \begin{bmatrix} B_k(\boldsymbol{x}_{0:k+1}) \\ \overline{\mathfrak{M}}_k(\boldsymbol{x}_k, \boldsymbol{x}_{k+1}) \end{bmatrix}$$

for a multivariate function B_k—whose particular form is not relevant to this argument—and for a map, $\overline{\mathfrak{M}}_k$, defined in (24) as a function on $\mathbb{R}^n \times \mathbb{R}^n$. Since $(\mathfrak{T}_k)_\sharp \, \eta_{\boldsymbol{X}_{0:k+1}} = \pi_{\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$, the map $\overline{\mathfrak{M}}_k$ must also push forward $\eta_{\boldsymbol{X}_k,\boldsymbol{X}_{k+1}}$ to the lag-1 smoothing marginal $\pi_{\boldsymbol{Z}_k,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$. This proves Part 2 of the theorem.

For Part 4, just notice that

$$\pi_{\boldsymbol{Y}_{0:k+1}}(\boldsymbol{y}_{0:k+1}) = \pi_{\boldsymbol{Y}_0,\boldsymbol{Y}_1}(\boldsymbol{y}_0, \boldsymbol{y}_1) \prod_{i=1}^{k} \pi_{\boldsymbol{Y}_{i+1}|\boldsymbol{Y}_{0:i}}(\boldsymbol{y}_{i+1}|\boldsymbol{y}_{0:i}) = \prod_{i=0}^{k} \mathfrak{c}_i, \tag{42}$$

where we used both (40) and (41). ■

Proof of Lemma 11 First a remark about notation: we denote by $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ the density (as a function of $\boldsymbol{x}$) of a Gaussian with mean $\boldsymbol{\mu}$ and covariance $\boldsymbol{\Sigma}$.

Now let $k > 0$ and notice that $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k}(\boldsymbol{z}_{k+1}|\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{z}_{k+1}; \boldsymbol{F}_k\, \boldsymbol{z}_k, \boldsymbol{Q}_k)$, $\pi_{\boldsymbol{Y}_{k+1}|\boldsymbol{Z}_{k+1}}(\boldsymbol{y}_{k+1}|\boldsymbol{z}_{k+1}) = \mathcal{N}(\boldsymbol{y}_{k+1}; \boldsymbol{H}_{k+1}\, \boldsymbol{z}_{k+1}, \boldsymbol{R}_{k+1})$ and $\eta_{\boldsymbol{X}_k}(\boldsymbol{z}_k) = \mathcal{N}(\boldsymbol{z}_k; 0, \mathbf{I})$. By definition of the target π^k in Theorem 9, we have:

$$\begin{aligned}
\pi^k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1}) &= \eta_{\boldsymbol{X}_k}(\boldsymbol{z}_k)\, \pi_{\boldsymbol{Y}_{k+1}|\boldsymbol{Z}_{k+1}}(\boldsymbol{y}_{k+1}|\boldsymbol{z}_{k+1})\, \pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{Z}_k}(\boldsymbol{z}_{k+1}|\mathfrak{M}^1_{k-1}(\boldsymbol{z}_k)) \\
&= \mathcal{N}(\boldsymbol{z}_k; 0, \mathbf{I})\, \mathcal{N}(\boldsymbol{y}_{k+1}; \boldsymbol{H}_{k+1}\, \boldsymbol{z}_{k+1}, \boldsymbol{R}_{k+1}) \\
&\quad \mathcal{N}(\boldsymbol{z}_{k+1}; \boldsymbol{F}_k\, (\boldsymbol{C}_{k-1}\, \boldsymbol{z}_k + \boldsymbol{c}_{k-1}), \boldsymbol{Q}_k) \\
&\propto \exp(-\frac{1}{2} \boldsymbol{z}^\top \boldsymbol{J}\, \boldsymbol{z} + \boldsymbol{z}^\top \boldsymbol{h}),
\end{aligned}$$

where $\boldsymbol{z} = (\boldsymbol{z}_k, \boldsymbol{z}_{k+1}) \in \mathbb{R}^{2n}$, and where $\boldsymbol{J} \in \mathbb{R}^{2n \times 2n}, \boldsymbol{h} \in \mathbb{R}^{2n}$ are defined as

$$\boldsymbol{J} = \begin{bmatrix} \boldsymbol{J}_{11} & \boldsymbol{J}_{12} \\ \boldsymbol{J}_{12}^\top & \boldsymbol{J}_{22} \end{bmatrix}, \quad \boldsymbol{h} = \begin{bmatrix} \boldsymbol{h}_1 \\ \boldsymbol{h}_2 \end{bmatrix},$$

with:

$$\begin{cases}
\boldsymbol{J}_{11} = \mathbf{I} + \boldsymbol{C}_{k-1}^\top\, \boldsymbol{F}_k^\top\, \boldsymbol{Q}_k^{-1}\, \boldsymbol{F}_k\, \boldsymbol{C}_{k-1} \\
\boldsymbol{J}_{12} = -\boldsymbol{C}_{k-1}^\top\, \boldsymbol{F}_k^\top\, \boldsymbol{Q}_k^{-1} \\
\boldsymbol{J}_{22} = \boldsymbol{Q}_k^{-1} + \boldsymbol{H}_{k+1}^\top\, \boldsymbol{R}_{k+1}^{-1}\, \boldsymbol{H}_{k+1} \\
\boldsymbol{h}_1 = \boldsymbol{J}_{12}\, \boldsymbol{F}_k\, \boldsymbol{c}_{k-1} \\
\boldsymbol{h}_2 = \boldsymbol{Q}_k^{-1}\, \boldsymbol{F}_k\, \boldsymbol{c}_{k-1} + \boldsymbol{H}_{k+1}^\top\, \boldsymbol{R}_{k+1}^{-1}\, \boldsymbol{y}_{k+1}.
\end{cases}$$

In particular, we can rewrite π^k in *information form* (Koller and Friedman, 2009) as $\pi^k(\boldsymbol{z}) = \mathcal{N}^{-1}(\boldsymbol{z}; \boldsymbol{h}, \boldsymbol{J})$. Moreover we know by Theorem 9[Part 1], that the submap $\mathfrak{M}^1_k(\boldsymbol{z}_{k+1}) = \boldsymbol{C}_k\, \boldsymbol{z}_{k+1} + \boldsymbol{c}_k$ pushes forward $\eta_{\boldsymbol{X}_{k+1}}$ to the filtering marginal $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$. Hence $(\boldsymbol{c}_k, \boldsymbol{C}_k)$ should be, respectively, the mean and a square root of the covariance of $\pi_{\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$ — thus the output of any square-root Kalman filter at time $k+1$. Now we just need to determine the submap $\mathfrak{M}^0_k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1}) = \boldsymbol{A}_k\, \boldsymbol{z}_k + \boldsymbol{B}_k\, \boldsymbol{z}_{k+1} + \boldsymbol{a}_k$. Given that $\mathfrak{M}_k$ is a block upper triangular function, the map $\boldsymbol{z}_k \mapsto \mathfrak{M}^0_k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1})$ should push forward $\eta_{\boldsymbol{X}_k}$ to $\boldsymbol{z}_k \mapsto \pi^k_{\boldsymbol{Z}_k|\boldsymbol{Z}_{k+1}}(\boldsymbol{z}_k|\mathfrak{M}^1_k(\boldsymbol{z}_{k+1}))$. Notice that $\pi^k_{\boldsymbol{Z}_k|\boldsymbol{Z}_{k+1}}(\boldsymbol{z}_k|\boldsymbol{z}_{k+1}) = \mathcal{N}^{-1}(\boldsymbol{z}_k; \boldsymbol{h}_1 - \boldsymbol{J}_{12}\, \boldsymbol{z}_{k+1}, \boldsymbol{J}_{11}) = \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{J}_{11}^{-1}(\boldsymbol{h}_1 - \boldsymbol{J}_{12}\, \boldsymbol{z}_{k+1}), \boldsymbol{J}_{11}^{-1})$. Hence $\pi^k_{\boldsymbol{Z}_k|\boldsymbol{Z}_{k+1}}(\boldsymbol{z}_k|\mathfrak{M}^1_k(\boldsymbol{z}_{k+1})) = \mathcal{N}(\boldsymbol{z}_k; \boldsymbol{J}_{11}^{-1}\boldsymbol{J}_{12}(\boldsymbol{F}_k\, \boldsymbol{c}_{k-1} - \boldsymbol{C}_k\, \boldsymbol{z}_{k+1} - \boldsymbol{c}_k), \boldsymbol{J}_{11}^{-1})$, and so:

$$\mathfrak{M}^0_k(\boldsymbol{z}_k, \boldsymbol{z}_{k+1}) = \boldsymbol{J}_{11}^{-1}\boldsymbol{J}_{12}(\boldsymbol{F}_k\, \boldsymbol{c}_{k-1} - \boldsymbol{C}_k\, \boldsymbol{z}_{k+1} - \boldsymbol{c}_k) + \boldsymbol{J}_{11}^{-1/2}\boldsymbol{z}_k.$$

Simple algebra then leads to (29). ■

Proof of Theorem 12 We use a very similar argument to Theorem 9. We first show that the maps $(\mathfrak{M}_i)_{i \geq 0}$ are well-defined. These maps are well-defined as long as, for instance, we show that π^i is a probability density for all $i \geq 0$, and as long as there exist permutations (σ_i) that guarantee the generalized block triangular structure of (31). As for the permutations, it suffices to consider $\sigma = \sigma_1 = \sigma_2 = \cdots$ with $\sigma(\mathbb{N}_{p+2n}) = \{1, \ldots, p, p+2n, p+2n-1, \ldots, p+1\}$. As for the targets (π^i), we now use a (complete) induction argument over i to show that, for

all $i \geq 0$, π^i is a nonvanishing density and $\int \pi^i(\boldsymbol{z}_\theta, \boldsymbol{z}_i, \boldsymbol{z}_{i+1})\, \mathrm{d}\boldsymbol{z}_i = A_i^\sharp\, \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1})$ for a map A_i defined on $\mathbb{R}^p \times \mathbb{R}^n$ as

$$A_i(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) = \begin{bmatrix} \mathfrak{T}^{\boldsymbol{\Theta}}_{i-1}(\boldsymbol{x}_\theta) \\ \boldsymbol{x}_{i+1} \end{bmatrix},$$

with $\mathfrak{T}^{\boldsymbol{\Theta}}_{i-1}(\boldsymbol{x}_\theta) = \boldsymbol{x}_\theta$ if $i = 0$.

For the base case ($i = 0$), just notice that $\mathfrak{c}_0 = \pi_{\boldsymbol{Y}_0, \boldsymbol{Y}_1}(\boldsymbol{y}_0, \boldsymbol{y}_1) < \infty$, so that $\pi^0 = \bar{\pi}^0/\mathfrak{c}_0 > 0$ is a valid density. Moreover, we have the desired marginal, i.e.,

$$\int \pi^0(\boldsymbol{z}_\theta, \boldsymbol{z}_0, \boldsymbol{z}_1)\, \mathrm{d}\boldsymbol{z}_0 = \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_1|\boldsymbol{Y}_0, \boldsymbol{Y}_1}(\boldsymbol{z}_\theta, \boldsymbol{z}_1|\boldsymbol{y}_0, \boldsymbol{y}_1) = A_0^\sharp\, \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_1|\boldsymbol{y}_0, \boldsymbol{y}_1}(\boldsymbol{z}_\theta, \boldsymbol{z}_1),$$

since A_0 is the identity map on $\mathbb{R}^p \times \mathbb{R}^n$. Now assume that π^j is a nonvanishing density for all $j \leq i$ (complete induction) with $i > 0$, and that the marginal $\int \pi^i(\boldsymbol{z}_\theta, \boldsymbol{z}_i, \boldsymbol{z}_{i+1})\, \mathrm{d}\boldsymbol{z}_i = A_i^\sharp\, \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1})$. Under this hypothesis, the maps $(\mathfrak{M}_j)_{j \leq i}$ are well-defined, and so are A_i, A_{i+1} since $\mathfrak{T}^{\boldsymbol{\Theta}}_i = \mathfrak{M}^{\boldsymbol{\Theta}}_0 \circ \cdots \circ \mathfrak{M}^{\boldsymbol{\Theta}}_i$. Before checking the integrability of π^{i+1}, notice that by definition of $\mathfrak{M}_i$ (a KR rearrangement), the map B_i, given by

$$B_i(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) = \begin{bmatrix} \mathfrak{M}^{\boldsymbol{\Theta}}_i(\boldsymbol{x}_\theta) \\ \mathfrak{M}^1_i(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) \end{bmatrix},$$

pushes forward $\eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}}$ to the marginal $\int \pi^i(\boldsymbol{z}_\theta, \boldsymbol{z}_i, \boldsymbol{z}_{i+1})\, \mathrm{d}\boldsymbol{z}_i$, which equals $A_i^\sharp\, \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ (inductive hypothesis), i.e., $(B_i)_\sharp\, \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}} = A_i^\sharp\, \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$. In particular, it must also be that $(A_i \circ B_i)_\sharp\, \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$, where $A_i \circ B_i$ corresponds precisely to the map $\widetilde{\mathfrak{M}}_i$ defined in (34), so that $(\widetilde{\mathfrak{M}}_i)_\sharp\, \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$.

Now we can prove that $\mathfrak{c}_{i+1} < \infty$ using the following identities:

$$\begin{aligned}
\mathfrak{c}_{i+1} &= \int \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1}) \\
&\quad \widetilde{\pi}^{i+1}(\mathfrak{T}^{\boldsymbol{\Theta}}_i(\boldsymbol{z}_\theta), \mathfrak{M}^1_i(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1}), \boldsymbol{z}_{i+2})\, \mathrm{d}\boldsymbol{z}_\theta\, \mathrm{d}\boldsymbol{z}_{i+1:i+2} \\
&= \int (\widetilde{\mathfrak{M}}_i)_\sharp\, \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}}(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1})\, \pi_{\boldsymbol{Z}_{i+2}|\boldsymbol{Z}_{i+1}, \boldsymbol{\Theta}}(\boldsymbol{z}_{i+2}|\boldsymbol{x}_{i+1}, \boldsymbol{x}_\theta) \\
&\quad \pi_{\boldsymbol{Y}_{i+2}|\boldsymbol{Z}_{i+2}, \boldsymbol{\Theta}}(\boldsymbol{y}_{i+2}|\boldsymbol{z}_{i+2}, \boldsymbol{x}_\theta)\, \mathrm{d}\boldsymbol{x}_\theta\, \mathrm{d}\boldsymbol{x}_{i+1}\, \mathrm{d}\boldsymbol{z}_{i+2} \\
&= \int \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}(\boldsymbol{x}_\theta, \boldsymbol{x}_{i+1}) \\
&\quad \pi_{\boldsymbol{Z}_{i+2}, \boldsymbol{Y}_{i+2}|\boldsymbol{Z}_{i+1}, \boldsymbol{\Theta}}(\boldsymbol{z}_{i+2}, \boldsymbol{y}_{i+2}|\boldsymbol{x}_{i+1}, \boldsymbol{x}_\theta)\, \mathrm{d}\boldsymbol{x}_\theta\, \mathrm{d}\boldsymbol{x}_{i+1}\, \mathrm{d}\boldsymbol{z}_{i+2} \\
&= \pi_{\boldsymbol{Y}_{i+2}|\boldsymbol{Y}_{0:i+1}}(\boldsymbol{y}_{i+2}|\boldsymbol{y}_{0:i+1}) < \infty,
\end{aligned} \tag{43}$$

where we used the change of variables:

$$\begin{bmatrix} \boldsymbol{x}_\theta \\ \boldsymbol{x}_{i+1} \end{bmatrix} = \begin{bmatrix} \mathfrak{T}^{\boldsymbol{\Theta}}_i(\boldsymbol{z}_\theta) \\ \mathfrak{M}^1_i(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1}) \end{bmatrix} = \widetilde{\mathfrak{M}}_i(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1}), \tag{44}$$

and the fact that $(\widetilde{\mathfrak{M}}_i)_\sharp\, \eta_{\boldsymbol{X}_{\boldsymbol{\Theta}}, \boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{\Theta}, \boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ (induction hypothesis). (The change of variables in (44) is valid for the following reason: the map $\widetilde{\mathfrak{M}}_i$ can be factorized as the

composition of $i+1$ (generalized) triangular functions, all that fit the hypothesis of Lemma 15, so that (44) should really be interpreted as a sequence of $i+1$ change of variables—each associated with one map in the composition and justified by Lemma 15.) Therefore π^{i+1} is a nonvanishing density. Following the same derivations as in (43), it is not hard to show that π^{i+1} has also the desired marginal, i.e.,

$$\int \pi^{i+1}(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+1}, \boldsymbol{z}_{i+2})\,\mathrm{d}\boldsymbol{z}_{i+1} = A_{i+1}^\sharp\, \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{i+2}|\boldsymbol{y}_{0:i+2}}(\boldsymbol{z}_\theta, \boldsymbol{z}_{i+2}).$$

This argument completes the induction step and shows that not only the maps $(\mathfrak{M}_i)_{i\geq 0}$ are well-defined—together with the maps $(T_i)_{i\geq 0}$ in (35)—but also that $(\widetilde{\mathfrak{M}}_i^1)_\sharp\, \eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_{i+1}} = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{i+1}|\boldsymbol{y}_{0:i+1}}$ for all $i \geq 0$ (Part 1 of the theorem).

Now we prove Part 2 of the theorem using another induction argument on $k \geq 0$. For the base case ($k = 0$), notice that $\mathfrak{T}_0 = T_0 = \mathfrak{M}_0$, and that, by definition, $\mathfrak{M}_0$ pushes forward $\eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_0,\boldsymbol{X}_1}$ to $\pi^0 = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\boldsymbol{Z}_1|\boldsymbol{y}_0,\boldsymbol{y}_1}$.

Assume that $\mathfrak{T}_k$ pushes forward $\eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_{0:k+1}}$ to $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$ for some $k > 0$ ($\mathfrak{T}_k$ is well-defined for all k since the maps $(T_i)_{i\geq 0}$ in (35) are also well-defined), and notice that

$$\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+2}|\boldsymbol{y}_{0:k+2}} = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}\, \frac{\pi_{\boldsymbol{y}_{k+2}|\boldsymbol{Z}_{k+2},\boldsymbol{\Theta}}\, \pi_{\boldsymbol{Z}_{k+2}|\boldsymbol{Z}_{k+1},\boldsymbol{\Theta}}}{\pi_{\boldsymbol{y}_{k+2}|\boldsymbol{y}_{0:k+1}}} = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}\, \frac{\widetilde{\pi}^{k+1}}{\mathfrak{c}_{k+1}},$$

where we used (43) and the definition of the collection $(\widetilde{\pi}^i)$. Let $\mathfrak{T}_{k+1} = T_0 \circ \cdots \circ T_{k+1}$ be defined as in Part 2 of the theorem, and observe that $\mathfrak{T}_{k+1} = C_{k+1} \circ T_{k+1}$ with

$$C_{k+1}(\boldsymbol{x}_\theta, \boldsymbol{x}_{0:k+2}) = \begin{bmatrix} \mathfrak{T}_k(\boldsymbol{x}_\theta, \boldsymbol{x}_{0:k+1}) \\ \boldsymbol{x}_{k+2} \end{bmatrix}, \quad T_{k+1}(\boldsymbol{x}_\theta, \boldsymbol{x}_{0:k+2}) = \begin{bmatrix} \mathfrak{M}_{k+1}^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta) \\ \boldsymbol{x}_0 \\ \vdots \\ \boldsymbol{x}_k \\ \mathfrak{M}_{k+1}^0(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1}, \boldsymbol{x}_{k+2}) \\ \mathfrak{M}_{k+1}^1(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+2}) \end{bmatrix}.$$

Thus the following hold:

$$\begin{aligned} \mathfrak{T}_{k+1}^\sharp\, \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+2}|\boldsymbol{y}_{0:k+2}} &= T_{k+1}^\sharp \left(\left(\mathfrak{T}_k^\sharp\, \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}} \right) \frac{\pi^{k+1}}{\eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_{k+1}}} \right) \\ &= T_{k+1}^\sharp \left(\eta_{\boldsymbol{X}_{0:k}}\, \pi^{k+1} \right) \\ &= \eta_{\boldsymbol{X}_{0:k}}\, \mathfrak{M}_{k+1}^\sharp\, \pi^{k+1} = \eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_{0:k+2}}, \end{aligned}$$

where we used the fact that by Lemma 15 (applied iteratively) it must be that $(C_{k+1} \circ T_{k+1})^\sharp \rho = T_{k+1}^\sharp\, C_{k+1}^\sharp \rho$ for all densities ρ. (Notice that C_{k+1} is the composition of functions which are trivial embeddings into the identity map of KR rearrangements that couple a pair of measures in $\mathscr{M}_+(\mathbb{R}^p \times \mathbb{R}^n \times \mathbb{R}^n)$, and thus each map in the composition satisfies the hypothesis of Lemma 15.) Thus $(\mathfrak{T}_{k+1})_\sharp\, \eta_{\boldsymbol{X}_\Theta,\boldsymbol{X}_{0:k+2}} = \pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+2}|\boldsymbol{y}_{0:k+2}}$, and this concludes the induction argument and the proof of Part 2 of the theorem.

The proof of Part 3 follows from $\mathfrak{c}_0 = \pi_{\boldsymbol{Y}_0,\boldsymbol{Y}_1}(\boldsymbol{y}_0, \boldsymbol{y}_1)$, (43), and (42). ■

Appendix C. Algorithms for Inference on State-Space Models

Here we digest the smoothing and joint state-parameter inference methodologies discussed in Section 7 into a handful of algorithms, described with pseudocode. Algorithms 1 and 2 below are building blocks: they describe, respectively, how to approximate a transport map given an (unnormalized) target density, and how to project a given transport map onto a set of monotone transformations. Algorithm 3 shows how to build a recursive approximation of $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$—i.e., the full Bayesian solution to the problem of sequential inference in state-space models with static parameters—using a decomposable transport map. See details in Section 7.3. For simplicity, we always use a standard normal reference process $\eta_{\boldsymbol{X}}$, although more general choices are possible. Algorithm 4 shows how to sample from the resulting approximation of the joint distribution $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{0:k+1}|\boldsymbol{y}_{0:k+1}}$, whereas Algorithm 5 focuses on a particular "filtering" marginal, i.e., $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{k+1}|\boldsymbol{y}_{0:k+1}}$. The problem of sequential inference on state-space models *without* static parameters (see Section 7.1) can be tackled via a simplified version of Algorithm 3, wherein the formal dependence on $\boldsymbol{\Theta}$ is dropped. The actual implementation of these algorithms is available online at `http://transportmaps.mit.edu`.

Algorithm 1 (Computation of a monotone map)
Given an unnormalized target density $\bar{\pi}$ and a parametric triangular monotone map $T[\mathbf{c}]$ of the form (5), defined by an arbitrary set of coefficients $\mathbf{c} \in \mathbb{R}^N$, find the optimal coefficients $\mathbf{c}^\star$ according to (7).

procedure ComputeMap($\bar{\pi}$, $T[\mathbf{c}]$, m)
  Generate samples $(\boldsymbol{x}_i)_{i=1}^m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$
  Solve (e.g., via a quasi-Newton or Newton method),

$$\mathbf{c}^\star = \operatorname*{argmin}_{\mathbf{c}\in\mathbb{R}^N} -\frac{1}{m}\sum_{i=1}^m \left(\log \bar{\pi}(T[\mathbf{c}](\boldsymbol{x}_i)) + \sum_k \log \partial_k T[\mathbf{c}]^k(\boldsymbol{x}_i) \right)$$

  return $T[\mathbf{c}^\star]$
end procedure

Algorithm 2 (Regression of a monotone map)
Given a map M and a parametric triangular monotone map $T[\mathbf{c}]$ of the form (5) , defined by an arbitrary set of coefficients $\mathbf{c} \in \mathbb{R}^N$, find the coefficients $\mathbf{c}^\star$ minimizing the discrete L^2 norm between the two maps.

procedure RegressionMap(M, $T[\mathbf{c}]$, m)
  Generate samples $(\boldsymbol{x}_i)_{i=1}^m \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \mathbf{I})$
  Solve

$$\mathbf{c}^\star = \operatorname*{argmin}_{\mathbf{c}\in\mathbb{R}^N} \frac{1}{m}\sum_{i=1}^m \left(M(\boldsymbol{x}_i) - T[\mathbf{c}](\boldsymbol{x}_i) \right)^2$$

  return $T[\mathbf{c}^\star]$
end procedure

Algorithm 3 (Joint parameter and state inference)
Given observations $(\boldsymbol{y}_i)_{i=0}^{k+1}$, construct a transport map approximation of the smoothing distribution $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\ldots,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{k+1}}$ in terms of a list of maps $(\mathfrak{M}_j)_{j=0}^{k}$.

procedure AssimilATE($(\boldsymbol{y}_i)_{i=0}^{k+1}$, m)
  for $i \leftarrow 0$ to k **do** ▷ see Thm. 12
    if $i = 0$ **then**
      Define $\widetilde{\mathfrak{T}}_{i-1}^{\boldsymbol{\Theta}}$ to be the identity map
      Define π^i as in (32)
    else
      $\widetilde{\mathfrak{T}}_{i-1}^{\boldsymbol{\Theta}}[\mathbf{c}^\star] \leftarrow$ RegressionMap($\widetilde{\mathfrak{T}}_{i-2}^{\boldsymbol{\Theta}} \circ \mathfrak{M}_{i-1}^{\boldsymbol{\Theta}}$, $\widetilde{\mathfrak{T}}_{i-1}^{\boldsymbol{\Theta}}[\mathbf{c}]$, m)
      Define π^i as in (33)
    end if
    $\mathfrak{M}_i[\mathbf{c}^\star] \leftarrow$ ComputeMap(π^i, $\mathfrak{M}_i[\mathbf{c}]$, m)
    Append $\mathfrak{M}_i$ to the list $(\mathfrak{M}_j)_{j=0}^{i-1}$
  end for
  return $(\mathfrak{M}_j)_{j=0}^{k}, \widetilde{\mathfrak{T}}_{k-1}^{\boldsymbol{\Theta}}$
end procedure

Algorithm 4 (Sample the smoothing distribution)
Generate a sample from the smoothing distribution $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_0,\ldots,\boldsymbol{Z}_{k+1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{k+1}}$ using the maps computed in Algorithm 3.

procedure SampleSmoothing($(\mathfrak{M}_j)_{j=0}^{k}$)
  Generate $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I})$, with $\mathbf{I}$ the identity in $d_\theta + k \cdot d_{\boldsymbol{z}}$ dimensions
  for $j \leftarrow k$ to 0 **do** ▷ see Thm. 12 Part. 2
    $\boldsymbol{x}_\theta \leftarrow \mathfrak{M}_j^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta)$
    $\boldsymbol{x}_j \leftarrow \mathfrak{M}_j^{0}(\boldsymbol{x}_\theta, \boldsymbol{x}_j, \boldsymbol{x}_{j+1})$
    $\boldsymbol{x}_{j+1} \leftarrow \mathfrak{M}_j^{1}(\boldsymbol{x}_\theta, \boldsymbol{x}_{j+1})$
  end for
  return $\boldsymbol{x}$
end procedure

Algorithm 5 (Sample the filtering distribution)
Generate a sample from the marginal distribution $\pi_{\boldsymbol{\Theta},\boldsymbol{Z}_{k+1}|\boldsymbol{y}_0,\ldots,\boldsymbol{y}_{k+1}}$ using the maps computed in Algorithm 3.

procedure SampleFiltering($\mathfrak{M}_k$, $\widetilde{\mathfrak{T}}_{k-1}^{\boldsymbol{\Theta}}$)
  Generate $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I})$, with $\mathbf{I}$ the identity in $d_\theta + d_{\boldsymbol{z}}$ dimensions
  Define

$$\widetilde{\mathfrak{M}}_k(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1}) := \begin{bmatrix} \widetilde{\mathfrak{T}}_{k-1}^{\boldsymbol{\Theta}}(\mathfrak{M}_k^{\boldsymbol{\Theta}}(\boldsymbol{x}_\theta)) \\ \mathfrak{M}_k^{1}(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1}) \end{bmatrix}$$

  $\boldsymbol{y} \leftarrow \widetilde{\mathfrak{M}}_k(\boldsymbol{x}_\theta, \boldsymbol{x}_{k+1})$ ▷ see Thm. 12 Part. 1
  return $\boldsymbol{y}$
end procedure

Appendix D. Additional Results for the Stochastic Volatility Model

We revisit the numerical example of Section 8 and re-run both the joint state/parameter inference problem and the long-time smoothing problem with *linear* rather than nonlinear maps. The results are less accurate, but substantially faster; see Table 1 and the discussion of this comparison in Section 8.

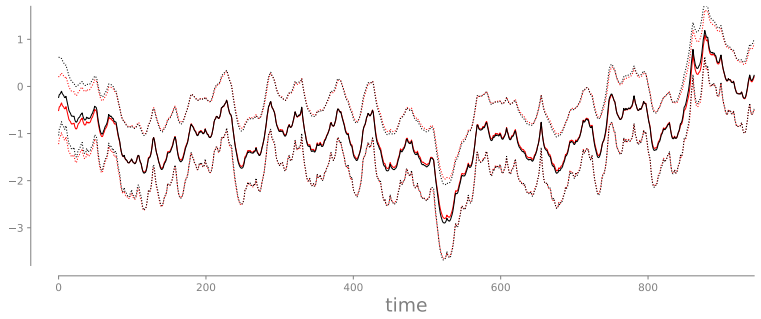


Figure 18: Same as Figure 10, but using linear maps. Compared to a high-order map, there seems to be only a minimal loss of accuracy, more prominent at earlier times.

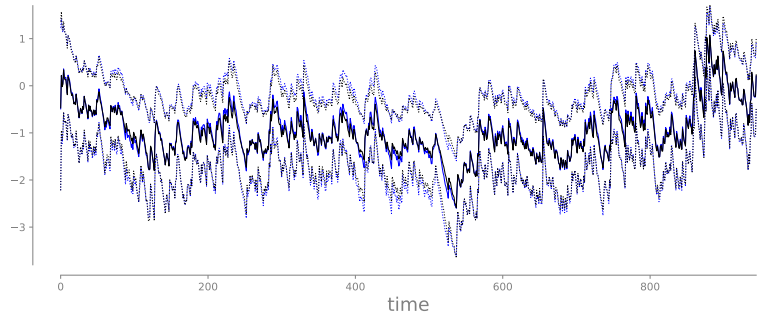


Figure 19: Comparison of the $\{5, 95\}$–percentiles (dashed lines) and the mean (solid line) of the numerical approximation of the filtering marginals using *linear* transport maps (blue lines) with those of a "reference" solution obtained via seventh-order maps (as shown in Figure 11). The two solutions look remarkably similar despite the enormous difference in computational cost (see Table 1).

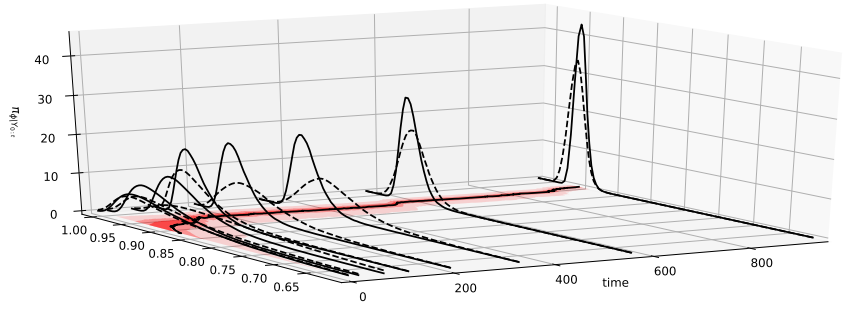


Figure 20: Same as Figure 12, but using linear maps. Here, the loss of accuracy is more dramatic than for the smoothing distribution of the state in Figure 18. Even though the approximate marginal captures the bulk of the true parameter marginals, for this specific problem of static parameter inference, a linear map is largely inadequate; hence the need for a higher-order nonlinear transformation.

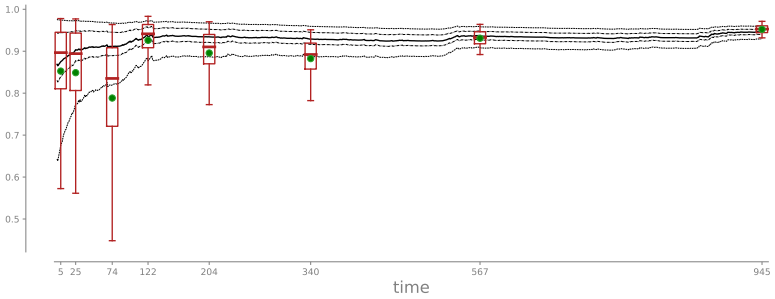


Figure 21: The horizontal plane of Figure 20 (*black lines*) overlaid with a selected number of box-and-whisker plots associated with the marginals of a "reference" MCMC solution. The ends of the whiskers represent the $\{5, 95\}$–percentiles, while the green dots correspond to the means of the reference distribution. Linear maps are insufficient to correctly characterize the parameter marginals, especially the transition at time 74 (cf. Figures 12 and 20)

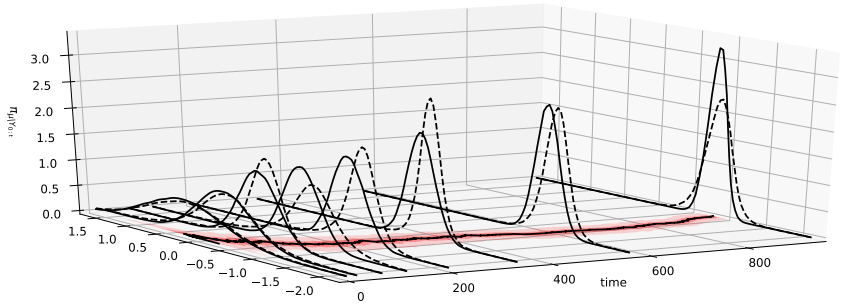


Figure 22: Same as Figure 13, but using linear maps. Once again, the linear map provides plausible, but somewhat inaccurate, results for sequential parameter inference. A nonlinear transformation is better suited for this problem.

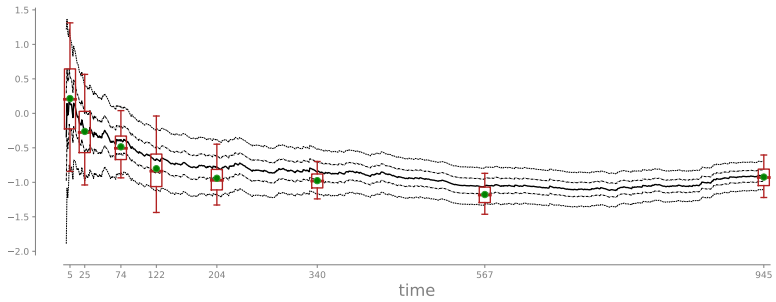


Figure 23: The horizontal plane of Figure 22 (*black lines*) overlaid with a selected number of box-and-whisker plots associated with the marginals of a "reference" MCMC solution. See Figure 21 caption for more details.

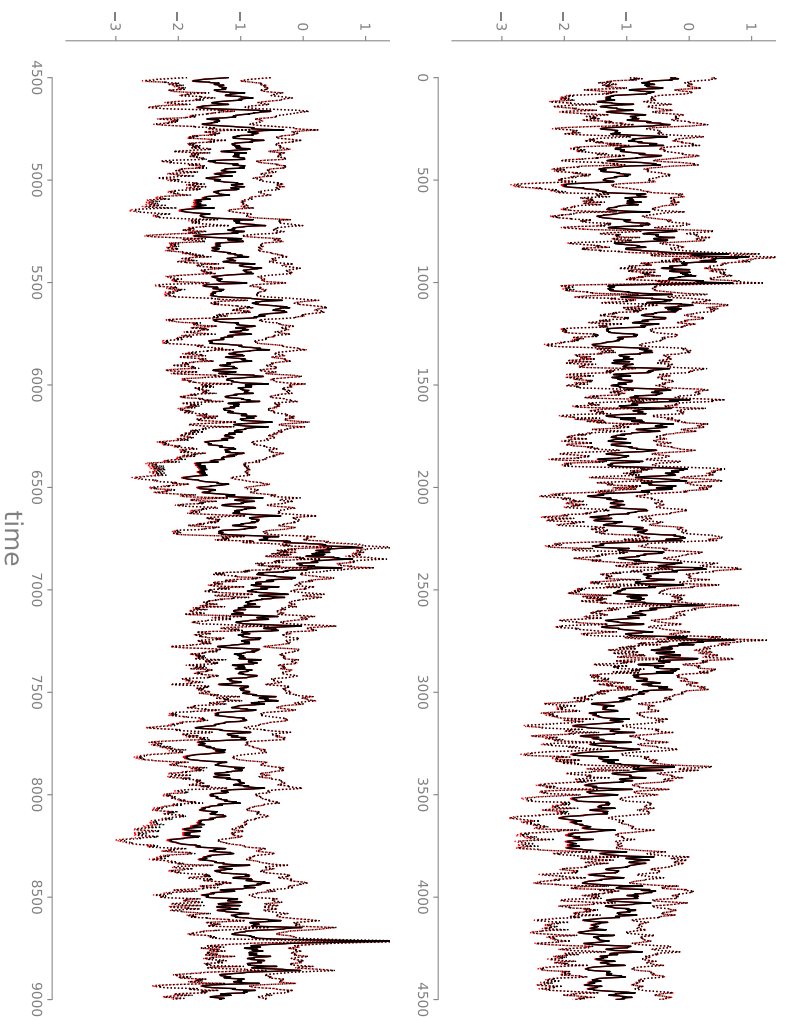


Figure 24: Same as Figure 17, but using linear maps. Long-time smoothing with no static parameters via linear maps yields accurate characterizations of the marginal distributions across all times, at a fraction of the cost of a high-order nonlinear transformation (see Table 1).

References

E. Anderes and M. Coram. A general spline representation for nonparametric and semi-parametric density estimates using diffeomorphisms. *arXiv:1205.5314*, 2012.

C. Andrieu, A. Doucet, and R. Holenstein. Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B*, 72(3):269–342, 2010.

S. Asmussen and P. W. Glynn. *Stochastic simulation: algorithms and analysis*, volume 57. Springer Science & Business Media, 2007.

J. M. Bardsley, A. Solonen, H. Haario, and M. Laine. Randomize-then-optimize: a method for sampling from posterior distributions in nonlinear inverse problems. *SIAM Journal on Scientific Computing*, 36(4):A1895–A1910, 2014. doi: 10.1137/140964023.

D. P. Bertsekas. *Dynamic programming and optimal control*, volume 1. Athena Scientific Belmont, MA, 1995.

G. J. Bierman. *Factorization methods for discrete sequential estimation*. Courier Corporation, 2006.

D. Bigoni, A. Spantini, and Y. Marzouk. On the computation of monotone transports. *In preparation*, 2019.

D. Blei, A. Kucukelbir, and J. McAuliffe. Variational inference: a review for statisticians. *arXiv:1601.00670*, 2016.

V. I. Bogachev, A. V. Kolesnikov, and K. V. Medvedev. Triangular transformations of measures. *Sbornik: Mathematics*, 196(3):309, 2005.

G. Carlier, A. Galichon, and F. Santambrogio. From Knothe's transport to Brenier's map and a continuation method for optimal transport. *SIAM Journal on Mathematical Analysis*, 41(6):2554–2576, 2010.

D. Cheng, Y. Cheng, Y. Liu, R. Peng, and S. Teng. Efficient sampling for Gaussian graphical models via spectral sparsification. In *Conference on Learning Theory*, pages 364–390, 2015.

N. Chopin, P. Jacob, and O. Papaspiliopoulos. SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426, 2013.

A. J. Chorin and X. Tu. Implicit sampling for particle filters. *Proceedings of the National Academy of Sciences*, 106(41):17249–17254, 2009.

P. G. Constantine, E. Dow, and Q. Wang. Active subspace methods in theory and practice: Applications to kriging surfaces. *SIAM Journal on Scientific Computing*, 36(4):A1500–A1524, 2014.

D. Crisan and A. Doucet. A survey of convergence results on particle filtering methods for practitioners. *IEEE Transactions on signal processing*, 50(3):736–746, 2002.

D. Crisan and J. Miguez. Nested particle filters for online parameter estimation in discrete-time state-space Markov models. *arXiv:1308.1883*, 2013.

K. Csilléry, M. G. B. Blum, O. E. Gaggiotti, and O. François. Approximate Bayesian Computation (ABC) in practice. *Trends in ecology & evolution*, 25(7):410–8, 2010.

T. Cui, J. Martin, Y. Marzouk, A. Solonen, and A. Spantini. Likelihood-informed dimension reduction for nonlinear inverse problems. *Inverse Problems*, 30(11):114015, 2014.

F. Daum and J. Huang. Particle flow for nonlinear filters with log-homotopy. In *SPIE Defense and Security Symposium*, pages 696918–696918. International Society for Optics and Photonics, 2008.

F. Daum and J. Huang. Particle flow and Monge-Kantorovich transport. In *Information Fusion (FUSION), 2012 15th International Conference on*, pages 135–142. IEEE, 2012.

P. Del Moral. Feynman-Kac formulae. In *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, pages 47–93. Springer, 2004.

P. Del Moral, A. Jasra, and Y. Zhou. Biased online parameter inference for state-space models. *Methodology and Computing in Applied Probability*, 19(3):727–749, 2017.

G. Detommaso, T. Cui, Y. Marzouk, R. Scheichl, and A. Spantini. A Stein variational Newton method. *Advances in Neural Information Processing Systems*, 2018. arXiv:1806.03085.

J. Dick, F. Y. Kuo, and I. H. Sloan. High-dimensional integration: the Quasi-Monte Carlo way. *Acta Numerica*, 22:133–288, 2013.

L. Dinh, J. Sohl-Dickstein, and S. Bengio. Density estimation using Real NVP. *arXiv:1605.08803*, 2016.

A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of nonlinear filtering*, 12(656-704):3, 2009.

R. J. Douglas. Applications of the Monge-Ampere equation and Monge transport problem to meteorology and oceanography. In *Monge Ampère Equation: Applications to Geometry and Optimization*, volume 226, page 33. American Mathematical Soc., 1999.

J. Durbin and S. J. Koopman. Time series analysis of non-Gaussian observations based on state-space models from both classical and Bayesian perspectives. *Journal of the Royal Statistical Society: Series B*, 62(1):3–56, 2000.

Y. B. Erol, Y. Wu, L. Li, and S. J. Russell. A nearly-black-box online algorithm for joint parameter and state estimation in temporal models. In *AAAI*, pages 1861–1869, 2017.

G. Evensen. The ensemble Kalman filter: Theoretical formulation and practical implementation. *Ocean dynamics*, 53(4):343–367, 2003.

G. Evensen. *Data Assimilation*. Springer, 2007.

G. Evensen and P. J. Van Leeuwen. An ensemble Kalman smoother for nonlinear dynamics. *Monthly Weather Review*, 128(6):1852–1867, 2000.

D. H. Fremlin. *Measure Theory*, volume 4. Torres Fremlin, 2000.

G. Gaspari and S. E. Cohn. Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society*, 125(554):723–757, 1999.

A. George and J. W. H. Liu. The evolution of the minimum degree ordering algorithm. *SIAM Review*, 31(1):1–19, 1989.

P. W. Glynn. Likelihood ratio gradient estimation for stochastic systems. *Communications of the ACM*, 33(10):75–84, 1990.

S. J. Godsill, A. Doucet, and M. West. Monte Carlo smoothing for nonlinear time series. *Journal of the American Statistical Association*, 99(465):156–168, 2004.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014.

T. M. Hamill, J. S. Whitaker, and C. Snyder. Distance-dependent filtering of background error covariance estimates in an ensemble Kalman filter. *Monthly Weather Review*, 129 (11):2776–2790, 2001.

J. M. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. *Unpublished manuscript*, 1971.

J. Han and Q. Liu. Stein variational adaptive importance sampling. *arXiv:1704.05201*, 2017.

J. Heng, A. Doucet, and Y. Pokern. Gibbs flow for approximate transport with applications to Bayesian computation. *arXiv:1509.08787*, 2015.

A. Hyvärinen. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709, 2005.

P. E. Jacob. Sequential Bayesian inference for implicit hidden Markov models and current limitations. *ESAIM: Proceedings and Surveys*, 51:24–48, 2015.

V. Jog and P. Loh. On model misspecification and KL separation for Gaussian graphical models. In *IEEE International Symposium on Information Theory*, pages 1174–1178, 2015.

J. K. Johnson and A. S. Willsky. A recursive model-reduction method for approximate inference in Gaussian Markov random fields. *IEEE Transactions on Image Processing*, 17(1):70–83, 2008.

N. Kantas, A. Doucet, S. S. Singh, J. Maciejowski, and N. Chopin. On particle methods for parameter estimation in state-space models. *Statistical Science*, 30(3):328–351, 2015.

L. V. Kantorovich. *The best use of economic resources.* Oxford & London: Pergamon Press., 1965.

S. Kim, N. Shephard, and S. Chib. Stochastic volatility: likelihood inference and comparison with ARCH models. *The Review of Economic Studies*, 65(3):361–393, 1998.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv:1312.6114*, 2013.

G. Kitagawa. Non-Gaussian state-space modeling of nonstationary time series. *Journal of the American Statistical Association*, 82(400):1032–1041, 1987.

G. Kitagawa. A self-organizing state-space model. *Journal of the American Statistical Association*, pages 1203–1215, 1998.

D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques.* MIT press, 2009.

H. Kushner and G. G. Yin. *Stochastic approximation and recursive algorithms and applications*, volume 35. Springer Science & Business Media, 2003.

V. Laparra, G. Camps-Valls, and J. Malo. Iterative Gaussianization: from ICA to random rotations. *IEEE transactions on neural networks*, 22(4):537–549, 2011.

S. L. Lauritzen. *Graphical Models.* Oxford University Press, 1996.

L. Lin, M. Drton, and A. Shojaie. High-dimensional inference of graphical models using regularized score matching. *arXiv:1507.00433*, 2015.

J. Liu and M. West. Combined parameter and state estimation in simulation-based filtering. In *Sequential Monte Carlo methods in practice*, pages 197–223. Springer, 2001.

Q. Liu and D. Wang. Stein variational gradient descent: a general purpose Bayesian inference algorithm. In *Advances in Neural Information Processing Systems*, pages 2370–2378, 2016.

G. Marsaglia and W. W. Tsang. The ziggurat method for generating random variables. *Journal of Statistical Software*, 5(8):1–7, 2000.

Y. Marzouk, T. Moselhy, M. Parno, and A Spantini. Sampling via measure transport: An introduction. In *Handbook of Uncertainty Quantification, R. Ghanem, D. Higdon, and H. Owhadi, editors.* Springer, 2016.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the Lasso. *The Annals of Statistics*, pages 1436–1462, 2006.

M. Mendoza, A. Allegra, and T. P. Coleman. Bayesian Lasso posterior sampling via parallelized measure transport. *arXiv:1801.02106*, 2018.

X. Meng and S. Schilling. Warp bridge sampling. *Journal of Computational and Graphical Statistics*, 11(3):552–586, 2002.

R. Morrison, R. Baptista, and Y. Marzouk. Beyond normality: Learning sparse probabilistic graphical models in the non-Gaussian setting. *Advances in Neural Information Processing Systems*, 2017. arXiv:1711.00950.

M. Morzfeld, X. Tu, E. Atkins, and A. J. Chorin. A random map implementation of implicit filters. *Journal of Computational Physics*, 231(4):2049–2066, 2012.

T. Moselhy and Y. Marzouk. Bayesian inference with optimal maps. *Journal of Computational Physics*, 231(23):7815–7850, 2012.

B. Oksendal. *Stochastic differential equations: an introduction with applications*. Springer Science & Business Media, 2013.

D. S. Oliver. Metropolized randomized maximum likelihood for sampling from multimodal distributions. *arXiv:1507.08563*, 2015.

M. Parno. *Transport maps for accelerated Bayesian computation*. PhD thesis, Massachusetts Institute of Technology, 2015.

M. Parno and Y. Marzouk. Transport map accelerated Markov chain Monte Carlo. *SIAM/ASA Journal on Uncertainty Quantification*, 6(2):645–682, 2018.

M. Parno, T. Moselhy, and Y. Marzouk. A multiscale strategy for Bayesian inference using transport maps. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):1160–1190, 2016.

N. G. Polson, J. R. Stroud, and P. Müller. Practical filtering with sequential parameter learning. *Journal of the Royal Statistical Society: Series B*, 70(2):413–428, 2008.

P. N. Raanes. On the ensemble Rauch-Tung-Striebel smoother and its equivalence to the ensemble Kalman smoother. *Quarterly Journal of the Royal Meteorological Society*, 142 (696):1259–1264, 2016.

J. O. Ramsay. Estimating smooth monotone functions. *Journal of the Royal Statistical Society: Series B*, pages 365–375, 1998.

H. E. Rauch, C. T. Striebel, and F. Tung. Maximum likelihood estimates of linear dynamic systems. *AIAA Journal*, 3(8):1445–1450, 1965.

S. Reich. A nonparametric ensemble transform method for Bayesian inference. *SIAM Journal on Scientific Computing*, 35(4):A2013–A2024, 2013.

S. Reich and C. Cotter. *Probabilistic Forecasting and Bayesian Data Assimilation*. Cambridge University Press, 2015.

D. J. Rezende and S. Mohamed. Variational inference with normalizing flows. *arXiv:1505.05770*, 2015.

C. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Science & Business Media, 2013.

M. Rosenblatt. Remarks on a multivariate transformation. *The Annals of Mathematical Statistics*, pages 470–472, 1952.

H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. CRC Press, 2005.

H. Rue, S. Martino, and N. Chopin. Approximate Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations. *Journal of the Royal Statistical Society: Series B*, 71(2):319–392, 2009.

Y. Saad. *Iterative Methods for Sparse Linear Systems*. SIAM, 2003.

A. M. Samarov. Exploring regression structure using nonparametric functional estimation. *Journal of the American Statistical Association*, 88(423):836–847, 1993.

F. Santambrogio. *Optimal Transport for Applied Mathematicians*, volume 87. Springer, 2015.

S. Särkkä. *Bayesian Filtering and Smoothing*, volume 3. Cambridge University Press, 2013.

C. Schillings and C. Schwab. Scaling limits in computational Bayesian inversion. *ESAIM: Mathematical Modelling and Numerical Analysis*, 50(6):1825–1856, 2016.

A. Shapiro. *Sample Average Approximation*, pages 1350–1355. Springer US, Boston, 2013.

A. Smith, A. Doucet, N. de Freitas, and N. Gordon. *Sequential Monte Carlo methods in practice*. Springer Science & Business Media, 2013.

A. Spantini. *On the low-dimensional structure of Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2017.

A. Spantini, A. Solonen, T. Cui, J. Martin, L. Tenorio, and Y. Marzouk. Optimal low-rank approximations of Bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 37(6):A2451–A2487, 2015.

A. Spantini, T. Cui, K. Willcox, L. Tenorio, and Y. M. Marzouk. Goal-oriented optimal approximations of Bayesian linear inverse problems. *SIAM Journal on Scientific Computing*, 39(5):S167–S196, 2017.

F. Stavropoulou and J. Müller. Parametrization of random vectors in polynomial chaos expansions via optimal transportation. *SIAM Journal on Scientific Computing*, 37(6): A2535–A2557, 2015.

A. M. Stuart. Inverse problems: a Bayesian perspective. *Acta Numerica*, 19:451–559, 2010.

E. G. Tabak and C. V. Turner. A family of nonparametric density estimation algorithms. *Communications on Pure and Applied Mathematics*, 66(2):145–164, 2013.

C. Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

L. Wang and X. Meng. Warp bridge sampling: the next generation. *arXiv:1609.07690*, 2016.

S. J. Wright and J. Nocedal. *Numerical Optimization*, volume 2. Springer New York, 1999.

D. Xiu. *Numerical methods for stochastic computations: a spectral method approach*. Princeton University Press, 2010.

T. Yang, P. G. Mehta, and S. P. Meyn. Feedback particle filter. *IEEE transactions on Automatic control*, 58(10):2465–2480, 2013.

M. Yannakakis. Computing the minimum fill-in is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 2(1):77–79, 1981.

Journal of Machine Learning Research 19 (2018) 1-34 Submitted 12/17; Revised 10/18; Published 11/18

Efficient Bayesian Inference of Sigmoidal Gaussian Cox Processes

Christian Donner christian.donner@bccn-berlin.de
Manfred Opper manfred.opper@tu-berlin.de
Artificial Intelligence Group
Technische Universität Berlin
Berlin, Germany

Editor: Ryan Adams

Abstract

We present an approximate Bayesian inference approach for estimating the intensity of a inhomogeneous Poisson process, where the intensity function is modelled using a Gaussian process (GP) prior via a sigmoid link function. Augmenting the model using a latent marked Poisson process and Pólya–Gamma random variables we obtain a representation of the likelihood which is conjugate to the GP prior. We estimate the posterior using a variational free–form mean field optimisation together with the framework of sparse GPs. Furthermore, as alternative approximation we suggest a sparse Laplace's method for the posterior, for which an efficient expectation–maximisation algorithm is derived to find the posterior's mode. Both algorithms compare well against exact inference obtained by a Markov Chain Monte Carlo sampler and standard variational Gauss approach solving the same model, while being one order of magnitude faster. Furthermore, the performance and speed of our method is competitive with that of another recently proposed Poisson process model based on a quadratic link function, while not being limited to GPs with squared exponential kernels and rectangular domains.

Keywords: Poisson process; Cox process; Gaussian process; data augmentation; variational inference

1. Introduction

Estimating the intensity rate of discrete events over a continuous space is a common problem for real world applications such as modeling seismic activity (Ogata, 1998), neural data (Brillinger, 1988), forestry (Stoyan and Penttinen, 2000) and so forth. A particularly common approach is a Bayesian model based on a so–called Cox process (Cox, 1955). The observed events are assumed to be generated from a Poisson process, whose intensity function is modeled as another random process with a given prior probability measure. The problem of inference for such type of models has also attracted interest in the Bayesian machine learning community in recent years. Møller et al. (1998); Brix and Diggle (2001); Cunningham et al. (2008) assumed that the intensity function is sampled from a Gaussian Process (GP) prior (Rasmussen and Williams, 2006). However, to restrict the intensity function of the Poisson process to nonnegative values, a common strategy is to choose a nonlinear link function which takes the GP as its argument and returns a valid intensity. Based on the success of variational approximations to deal with complex Gaussian process models, the inference problem for such Poisson models has attracted considerable interest in the machine learning community.

While powerful black–box variational Gaussian inference algorithms are available which can be applied to arbitrary link–functions, the choice of link–functions is not only crucial for defining the prior over intensities but can also be important for the efficiency of variational inference. The 'standard' choice of Cox processes with an exponential link function was treated in (Hensman et al., 2015). However, variational Gaussian inference for this link function has the disadvantage that the posterior variance becomes decoupled from the observations (Lloyd et al., 2015).[1] An interesting choice is the quadratic link function of (Lloyd et al., 2015) for which integrations over the data domain, which are necessary for sparse GP inference, can be (for specific kernel) computed analytically.[2] For both models, the minimisation of the variational free energies is performed by gradient descent techniques.

In this paper we will deal with approximate inference for a model with a sigmoid link–function. This model was introduced by (Adams et al., 2009) together with a MCMC sampling algorithm which was further improved by (Gunter et al., 2014) and (Teh and Rao, 2011). Kirichenko and van Zanten (2015) have shown that the model has favourable (frequentist) theoretical properties provided priors and hyperparameters are chosen appropriately. In contrast to a direct variational Gaussian approximation for the posterior distribution of the latent function, we will introduce an alternative type of variational approximation which is specially designed for the *sigmoidal Gaussian Cox process*. We build on recent work on Bayesian logistic regression by data augmentation with Pólya–Gamma random variables (Polson et al., 2013). This approach was already used in combination with GPs (Linderman et al., 2015; Wenzel et al., 2017), for stochastic processes in discrete time (Linderman et al., 2017), and for jump processes (Donner and Opper, 2017). We extend this method to an augmentation by a latent, marked Poisson process, where the marks are distributed according to a Pólya–Gamma distribution.[3] In this way, the augmented likelihood becomes conjugate to a GP distribution. Using a combination of a mean–field variational approximation together with sparse GP approximations (Csató and Opper, 2002; Csató, 2002; Titsias, 2009) we obtain explicit analytical variational updates leading to fast inference. In addition, we show that the same augmentation can be used for the computation of the maximum a posteriori (MAP) estimate by an expectation–maximisation (EM) algorithm. With this we obtain a Laplace approximation to the non–augmented posterior.

The paper is organised as follows: In section 2, we introduce the sigmoidal Gaussian Cox process model and its transformation by the variable augmentation. In section 3, we derive a variational mean field method and an EM–algorithm to obtain the MAP estimate, followed by the Laplace approximation of the posterior. Both methods are based on a sparse GP approximation to make the infinite dimensional problem tractable. In section 4, we demonstrate the performance of our method on synthetic data sets and compare with the results of a Monte Carlo sampling method for the model and the variational approximation of Hensman et al. (2015), which we modify to solve the Cox–process model with the scaled sigmoid link function. Then we compare our method to the state-of-the-art inference

1. Samo and Roberts (2015) propose an efficient approximate sampling scheme.
2. For a frequentist nonparametric approach to this model, see (Flaxman et al., 2017). For a Bayesian extension see (Walder and Bishop, 2017).
3. For a different application of marked Poisson processes, see (Lloyd et al., 2016).

algorithm (Lloyd et al., 2015) on artificial and real data sets with up to 10^4 observations. Section 5 presents a discussion and an outlook.

2. The Inference problem

We assume that N events $\mathcal{D} = \{\boldsymbol{x}_n\}_{n=1}^N$ are generated by a Poisson process. Each point $\boldsymbol{x}_n$ is a d–dimensional vector in the compact domain $\mathcal{X} \subset \mathbb{R}^d$. The goal is to infer the varying *intensity function* $\Lambda(\boldsymbol{x})$ (the mean measure of the process) for all $\boldsymbol{x} \in \mathcal{X}$ based on the likelihood

$$L(\mathcal{D}|\Lambda) = \exp\left(-\int_{\mathcal{X}} \Lambda(\boldsymbol{x}) d\boldsymbol{x}\right) \prod_{n=1}^{N} \Lambda(\boldsymbol{x}_n),$$

which is equal (up to a constant) to the density of a Poisson process having intensity Λ (see Appendix C and (Konstantopoulos et al., 2011)) with respect to a Poisson process with unit intensity. In a Bayesian framework, a prior over the intensity makes Λ a random process. Such a doubly stochastic point process is called *Cox process* (Cox, 1955). Since one needs $\Lambda(\boldsymbol{x}) \geq 0$, Adams et al. (2009) suggested a reparametrization of the intensity function by $\Lambda(\boldsymbol{x}) = \lambda\sigma(g(\boldsymbol{x}))$, where $\sigma(x) = (1 + e^{-x})^{-1}$ is the sigmoid function and λ is the maximum intensity rate. Hence, the intensity $\Lambda(\boldsymbol{x})$ is positive everywhere, for any arbitrary function $g(\boldsymbol{x}) : \mathcal{X} \to \mathbb{R}$ and the inference problem is to determine this function. Throughout this work we assume that $g(\cdot)$ will be modelled as a GP (Rasmussen and Williams, 2006) and the resulting process is called *sigmoidal Gaussian Cox process*. The likelihood for g becomes

$$L(\mathcal{D}|g, \lambda) = \exp\left(-\int_{\mathcal{X}} \lambda\sigma(g(\boldsymbol{x})) d\boldsymbol{x}\right) \prod_{n=1}^{N} \lambda\sigma(g_n), \tag{1}$$

where $g_n \doteq g(\boldsymbol{x}_n)$. For Bayesian inference we define a GP prior measure P_{GP} with zero mean and covariance kernel $k(\boldsymbol{x}, \boldsymbol{x}') : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^+$. λ has as prior density (with respect to the ordinary Lebesgue measure) $p(\lambda)$ which we take to be a Gamma density with shape-, and rate parameter α_0 and β_0, respectively. Hence, for the prior we get the product measure $dP_{\mathrm{prior}} = dP_{\mathrm{GP}} \times p(\lambda)d\lambda$. The posterior density $\boldsymbol{p}$ (with respect to the prior measure) is given by

$$\boldsymbol{p}(g, \lambda|\mathcal{D}) \doteq \frac{dP_{\mathrm{posterior}}}{dP_{\mathrm{prior}}}(g, \lambda|\mathcal{D}) = \frac{L(\mathcal{D}|g, \lambda)}{\mathbb{E}_{P_{\mathrm{prior}}}[L(\mathcal{D}|g, \lambda)]}. \tag{2}$$

The normalising expectation in the denominator on the right hand side is with respect to the probability measure P_{prior}. To deal with the infinite dimensionality of GPs and Poisson processes we require a minimum of extra notation. We introduce densities or *Radon–Nikodým derivatives* such as defined in Equation (2) (see Appendix C or de G. Matthews et al. (2016)) with respect to infinite dimensional measures by boldface symbols $\boldsymbol{p}(\boldsymbol{z})$. On the other hand, non–bold densities $p(\boldsymbol{z})$ denote densities in the 'classical' sense, which means they are with respect to Lebesgue measure $d\boldsymbol{z}$.

Bayesian inference for this model is known to be doubly intractable (Murray et al., 2006). The likelihood in Equation (1) contains the integral of g over the space $\mathcal{X}$ in the exponent and the normalisation of the posterior in Equation (2) requires calculating expectation of Equation (1). In addition inference is hampered by the fact, that likelihood (1) depends

non–linearly on g (through sigmoid and exponent of sigmoid). In the following we tackle this by an augmentation scheme for the likelihood, such that it becomes conjugate to a GP prior and we subsequently can derive an analytic form of a variational posterior given one simple mean field assumption (Section 3).

2.1. Data augmentation I: Latent Poisson process

We will briefly introduce a data augmentation scheme by a latent Poisson process which forms the basis of the sampling algorithm of Adams et al. (2009). We will then extend this method further to an augmentation by a *marked* Poisson process. We focus on the exponential term in Equation (1). Utilizing the well known property of the sigmoid that $\sigma(x) = 1 - \sigma(-x)$ we can write

$$\exp\left(-\int_{\mathcal{X}} \lambda\sigma(g(\boldsymbol{x})) d\boldsymbol{x}\right) = \exp\left(-\int_{\mathcal{X}} (1 - \sigma(-g(\boldsymbol{x}))) \lambda d\boldsymbol{x}\right). \tag{3}$$

The left hand side has the form of a characteristic functional of a Poisson process. Generally, for a random set of points $\Pi_{\mathcal{Z}} = \{\boldsymbol{z}_m; \boldsymbol{z}_m \in \mathcal{Z}\}$ on a space $\mathcal{Z}$ and with a function $h(\boldsymbol{z})$, this is defined as

$$\mathbb{E}_{P_\Lambda}\left[\prod_{\boldsymbol{z}_m \in \Pi_{\mathcal{Z}}} e^{h(\boldsymbol{z}_m)}\right] = \exp\left(-\int_{\mathcal{Z}} \left(1 - e^{h(\boldsymbol{z})}\right) \Lambda(\boldsymbol{z}) d\boldsymbol{z}\right), \tag{4}$$

where P_Λ is the probability measure of a Poisson process with intensity $\Lambda(\boldsymbol{z})$. Equation (4) can be derived by Campbell's theorem (see Appendix A and (Kingman, 1993, chap. 3)) and identifies a Poisson process uniquely.

Setting $h(\boldsymbol{z}) = \ln\sigma(-g(\boldsymbol{z}))$, and $\mathcal{Z} = \mathcal{X}$, and combining Equation (3) and (4) we obtain the likelihood used by Adams et al. (2009, Eq. 4). However, in this work we make use of another augmentation, before invoking Campbell's theorem. This will result in a likelihood which is conjugate to the model priors and further simplifies inference.

2.2. Data augmentation II: Pólya–Gamma variables and marked Poisson process

Following Polson et al. (2013) we represent the inverse of the hyperbolic cosine as a scaled Gaussian mixture model

$$\cosh^{-b}(z/2) = \int_0^\infty e^{-\frac{z^2}{2}\omega} p_{\mathrm{PG}}(\omega|b, 0) d\omega, \tag{5}$$

where p_{PG} is a *Pólya–Gamma* density (Appendix B). We further define the *tilted* Pólya–Gamma density by

$$p_{\mathrm{PG}}(\omega|b, c) \propto e^{-\frac{c^2}{2}\omega} p_{\mathrm{PG}}(\omega|b, 0), \tag{6}$$

where $b > 0$ and c are parameters. We will not need an explicit form of this density, since the subsequently derived inference algorithms will only require the first moments. Those can be obtained directly from the moment generating function, which can be calculated straightforwardly from Equation (5) and (6) (see Appendix B). Equation (5) allows us to

rewrite the sigmoid function as

$$\sigma(z) = \frac{e^{\frac{z}{2}}}{2\cosh(\frac{z}{2})} = \int_0^\infty e^{f(\omega,z)} p_{\mathrm{PG}}(\omega|1,0)d\omega, \tag{7}$$

where we define

$$f(\omega, z) \doteq \frac{z}{2} - \frac{z^2}{2}\omega - \ln 2.$$

Setting $z = -g(\boldsymbol{x})$ in Equation (3) and substituting Equation (7) we get

$$\exp\left(-\int_{\mathcal{X}} \lambda\left(1-\sigma(-g(\boldsymbol{x}))\right) d\boldsymbol{x}\right) = \exp\left(-\int_{\mathcal{X}\times\mathbb{R}^+} \left(1 - e^{f(\omega,-g(\boldsymbol{x}))}\right)\ p_{\mathrm{PG}}(\omega|1,0)\ \lambda d\omega d\boldsymbol{x}\right). \tag{8}$$

Finally, we apply Campbell's theorem (Equation (4)) to Equation (8). The space is a product space $\mathcal{Z} = \hat{\mathcal{X}} \doteq \mathcal{X} \times \mathbb{R}^+$ and the intensity $\Lambda(\boldsymbol{x}, \omega) = \lambda p_{\mathrm{PG}}(\omega|1,0)$. This results in the final representation of the exponential in Equation (8)

$$\exp\left(-\int_{\hat{\mathcal{X}}} \left(1 - e^{f(\omega,-g(\boldsymbol{x}))}\right) \Lambda(\boldsymbol{x},\omega)\ d\omega d\boldsymbol{x}\right) = \mathbb{E}_{P_\Lambda}\left[\prod_{(\boldsymbol{x},\omega)_m \in \Pi_{\hat{\mathcal{X}}}} e^{f(\omega_m,-g_m)}\right].$$

Interestingly, the new Poisson process $\Pi_{\hat{\mathcal{X}}}$ with measure P_Λ has the form of a *marked* Poisson process (Kingman, 1993, chap. 5), where the latent Pólya-Gamma variables ω_m denote the 'marks' being independent random variables at each location $\boldsymbol{x}_m$. It is straightforward to sample such processes by first sampling the inhomogeneous Poisson process on domain $\mathcal{X}$ (for example by 'thinning' a process with constant rate (Lewis and Shedler, 1979; Adams et al., 2009)) and then drawing a mark ω on each event independently from the density $p_{\mathrm{PG}}(\omega|1,0)$.

Finally, using the Pólya–Gamma augmentation also for the discrete likelihood factors corresponding to the observed events in Equation (1) we obtain the following joint likelihood of the model

$$\begin{aligned} L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda) &\doteq \frac{dP_{\mathrm{joint}}}{dP_{\mathrm{aug}}}(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda) \\ &= \prod_{(\boldsymbol{x},\omega)_m \in \Pi_{\hat{\mathcal{X}}}} e^{f(\omega_m,-g_m)} \prod_{n=1}^N \lambda e^{f(\omega_n,g_n)}, \end{aligned} \tag{9}$$

where we define the prior measure of augmented variables as $P_{\mathrm{aug}} = P_\Lambda \times P_{\boldsymbol{\omega}_N}$ and where $\boldsymbol{\omega}_N = \{\omega_n\}_{n=1}^N$ are the Pólya–Gamma variables for the observations $\mathcal{D}$ with the prior measure $dP_{\boldsymbol{\omega}_N} = \prod_{n=1}^N p(\omega_n|1,0)d\omega_n$. This augmented representation of the likelihood contains the function $g(\cdot)$ only linearly and quadratically in the exponents and is thus conjugate to the GP prior of $g(\cdot)$. Note that the original likelihood in Equation (1) can be recovered by $\mathbb{E}_{P_{\mathrm{aug}}}\left[L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)\right] = L(\mathcal{D}|g,\lambda)$.

3. Inference in the augmented space

Based on the augmentation we define a posterior density for the joint model with respect to the product measure $P_{\mathrm{prior}} \times P_{\mathrm{aug}}$

$$\begin{aligned} \boldsymbol{p}(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}, g, \lambda|\mathcal{D}) &\doteq \frac{dP_{\mathrm{posterior}}}{d(P_{\mathrm{prior}} \times P_{\mathrm{aug}})}(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}, g, \lambda|\mathcal{D}) \\ &= \frac{L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)}{L(\mathcal{D})}, \end{aligned} \tag{10}$$

where the denominator is the marginal likelihood $L(\mathcal{D}) = \mathbb{E}_{P_{\mathrm{prior}} \times P_{\mathrm{aug}}}\left[L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)\right]$. The posterior density of Equation (10) could be sampled using Gibbs sampling with explicit, tractable conditional densities. Similar to the variational approximation in the next section, one can show that the conditional measure of the point sets $\Pi_{\hat{\mathcal{X}}}$ and the variables $\boldsymbol{\omega}_N$, given the function $g(\cdot)$ and maximal intensity λ is a product of a specific marked Poisson process and independent (tilted) Pólya–Gamma densities. On the other hand, the distribution over function $g(\cdot)$ conditioned on $\Pi_{\hat{\mathcal{X}}}$ and $\boldsymbol{\omega}_N$ is a Gaussian process. Note, however, one needs to sample this GP only at the finite points $\boldsymbol{x}_m$ in the random set $\Pi_{\hat{\mathcal{X}}}$ and the fixed set $\mathcal{D}$.

3.1. Variational mean–field approximation

For variational inference one assumes that the desired posterior probability measure belongs to a family of measures for which the inference problem is tractable. Here we make a simple structured mean field assumption in order to fully utilise its conjugate structure: We approximate the posterior measure by

$$P_{\mathrm{posterior}}(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}, g, \lambda|\mathcal{D}) \approx Q_1(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}) \times Q_2(g,\lambda), \tag{11}$$

meaning that the dependencies between the Pólya–Gamma variables $\boldsymbol{\omega}_N$ and the marked Poisson process $\Pi_{\hat{\mathcal{X}}}$ on the one hand, and the function g and the maximal intensity λ on the other hand, are neglected. As we will see in the following, this simple mean–field assumption allows us to derive the posterior approximation analytically.

The variational approximation is optimised by minimising the Kullback–Leibler divergence between exact and approximated posteriors. This is equivalent to maximising the lower bound on the marginal likelihood of the observations

$$\mathcal{L}(\boldsymbol{q}) = \mathbb{E}_Q\left[\log\left\{\frac{L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)}{\boldsymbol{q}_1(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}})\boldsymbol{q}_2(g,\lambda)}\right\}\right] \leq \log L(\mathcal{D}), \tag{12}$$

where Q is the probability measure of the variational posterior in Equation (11) and we introduced approximate likelihoods

$$\boldsymbol{q}_1(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}) \doteq \frac{dQ_1}{dP_{\mathrm{aug}}}(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}), \qquad \boldsymbol{q}_2(g,\lambda) \doteq \frac{dQ_2}{dP_{\mathrm{prior}}}(g,\lambda).$$

Using standard arguments for mean field variational inference (Bishop, 2006, chap. 10) and Equation (11), one can then show that the optimal factors satisfy

$$\ln \boldsymbol{q}_1\left(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}\right) = \mathbb{E}_{Q_2}\left[\log L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)\right] + \mathrm{const.} \tag{13}$$

and

$$\ln \boldsymbol{q}_2(g,\lambda) = \mathbb{E}_{Q_1}\left[\log L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)\right] + \text{const.}\,, \tag{14}$$

respectively. These results lead to an iterative scheme for optimising $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ in order to increase the lower bound in Equation (12) in every step. From the structure of the likelihood one derives two further factorisations:

$$\boldsymbol{q}_1(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}) = \boldsymbol{q}_1(\boldsymbol{\omega}_N)\boldsymbol{q}_1(\Pi_{\hat{\mathcal{X}}}), \tag{15}$$
$$\boldsymbol{q}_2(g,\lambda) = \boldsymbol{q}_2(g)\boldsymbol{q}_2(\lambda), \tag{16}$$

where the densities are defined with respect to the measures $dP(\boldsymbol{\omega}_N)$, dP_Λ, dP_{GP}, and $p(\lambda)d\lambda$, respectively. The subsequent section describes these updates explicitly.

Optimal Pólya–Gamma density Following Equation (13) and (15) we obtain

$$\boldsymbol{q}_1(\boldsymbol{\omega}_N) = \prod_{n=1}^{N} \frac{\exp\left(-\frac{c_1^{(n)}}{2}\omega_n\right)}{\cosh^{-1}\left(c_1^{(n)}/2\right)} = \prod_{n=1}^{N} \frac{p_{\text{PG}}\left(\omega_n|1,c_1^{(n)}\right)}{p_{\text{PG}}\left(\omega_n|1,0\right)},$$

where the factors are *tilts* of the prior Pólya-Gamma densities (see Equation (6) and Appendix B) with $c_1^{(n)} = \sqrt{\mathbb{E}_{Q_2}\left[g_n^2\right]}$. By simple density transformation we obtain the density with respect to the Lebesgue measure as

$$q_1(\boldsymbol{\omega}_N) = \boldsymbol{q}_1(\boldsymbol{\omega}_N)\left|\frac{dP_{\boldsymbol{\omega}_N}}{d\boldsymbol{\omega}_N}\right| = \prod_{n=1}^{N} p_{\text{PG}}\left(\omega_n|1,c_1^{(n)}\right), \tag{17}$$

being a product of *tilted* Pólya–Gamma densities.

Optimal Poisson process Using Equation (13) and (15) we obtain

$$\boldsymbol{q}_1(\Pi_{\hat{\mathcal{X}}}) = \frac{\prod_{(\boldsymbol{x},\omega)_m \in \Pi_{\hat{\mathcal{X}}}} e^{\mathbb{E}_{Q_2}[f(\omega_m,-g_m)]}\lambda_1}{\exp\left(\int_{\hat{\mathcal{X}}}\left(e^{\mathbb{E}_{Q_2}[f(\omega,-g(\boldsymbol{x}))]}-1\right)\lambda_1 p_{\text{PG}}(\omega|1,0)d\boldsymbol{x}d\omega\right)}, \tag{18}$$

with $\lambda_1 \doteq e^{\mathbb{E}_{Q_2}[\log\lambda^*]}$. Note, that $\mathbb{E}_{Q_2}\left[f(\omega_m,-g_m)\right]$ involves the expectations $\mathbb{E}_{Q_2}\left[g_m\right]$ and $\mathbb{E}_{Q_2}\left[(g_m)^2\right]$. One can show, that Equation (18) is again a marked Poisson process with intensity

$$\begin{aligned}\Lambda_1(\boldsymbol{x},\omega) =& \lambda_1 \frac{\exp\left(-\frac{\mathbb{E}_{Q_2}[g(\boldsymbol{x})]}{2}\right)}{2\cosh\left(\frac{c_1(\boldsymbol{x})}{2}\right)} p_{\text{PG}}\left(\omega|1,c_1(\boldsymbol{x})\right) \\ =& \lambda_1\sigma(-c_1(\boldsymbol{x}))\exp\left(\frac{c_1(\boldsymbol{x})-\mathbb{E}_{Q_2}\left[g(\boldsymbol{x})\right]}{2}\right) p_{\text{PG}}\left(\omega|1,c_1(\boldsymbol{x})\right)\end{aligned} \tag{19}$$

where $c_1(\boldsymbol{x}) = \sqrt{\mathbb{E}_{Q_2}\left[g(\boldsymbol{x})^2\right]}$ (for a proof see Appendix D).

Optimal Gaussian process From Equation (14) and (16) we obtain the optimal approximation of the posterior likelihood (note that this is defined relative to GP prior)

$$\boldsymbol{q}_2(g) \propto e^{U(g)},$$

where the effective log–likelihood is given by

$$U(g) = \mathbb{E}_{Q_1}\left[\sum_{(\boldsymbol{x},\omega)_m \in \Pi_{\hat{\mathcal{X}}}} f(\omega_m, -g_m)\right] + \sum_{n=1}^{N}\mathbb{E}_{Q_1}\left[f(\omega_n, g(\boldsymbol{x}_n))\right].$$

The first expectation is over the variational Poisson process $\Pi_{\hat{\mathcal{X}}}$ and the second one over the Pólya–Gamma variables $\boldsymbol{\omega}_N$. These can be easily evaluated (see Appendix A) and one finds

$$U(g) = -\frac{1}{2}\int_{\mathcal{X}} A(\boldsymbol{x})g(\boldsymbol{x})^2 d\boldsymbol{x} + \int_{\mathcal{X}} B(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}, \tag{20}$$

with

$$\begin{aligned}A(\boldsymbol{x}) =& \sum_{n=1}^{N}\mathbb{E}_{Q_1}\left[\omega_n\right]\delta(\boldsymbol{x}-\boldsymbol{x}_n) + \int_0^\infty \omega\Lambda_1(\boldsymbol{x},\omega)d\omega, \\ B(\boldsymbol{x}) =& \frac{1}{2}\sum_{n=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_n) - \frac{1}{2}\int_0^\infty \Lambda_1(\boldsymbol{x},\omega)d\omega,\end{aligned}$$

where $\delta(\cdot)$ is the Dirac delta function. The expectations and integrals over ω are

$$\begin{aligned}&\mathbb{E}_{Q_1}\left[\omega_n\right] = \frac{1}{2c_1^{(n)}}\tanh\left(\frac{c_1^{(n)}}{2}\right), \\ &\int_0^\infty \Lambda_1(\boldsymbol{x},\omega)d\omega = \lambda_1\sigma(-c_1(\boldsymbol{x}))\exp\left(\frac{c_1(\boldsymbol{x})-\mathbb{E}_{Q_2}\left[g(\boldsymbol{x})\right]}{2}\right) \doteq \Lambda_1(\boldsymbol{x}), \\ &\int_0^\infty \omega\Lambda_1(\boldsymbol{x},\omega)d\omega = \frac{1}{2c_1(\boldsymbol{x})}\tanh\left(\frac{c_1(\boldsymbol{x})}{2}\right)\Lambda_1(\boldsymbol{x}).\end{aligned}$$

The resulting variational distribution defines a Gaussian process. Because of the mean–field assumption the integrals in Equation (20) do not require integration over random variables, but only solving two deterministic integrals over space $\mathcal{X}$. However, those integrals depend on function g over the entire space and it is not possible for a general kernel to compute the marginal posterior density at an input $\boldsymbol{x}$ in closed form. For specific GP kernel operators, which are the inverses of differential operators, a solution in terms of linear partial differential equations would be possible. This could be of special interest for one–dimensional problems where Matern kernels with integer parameters (Rasmussen and Williams, 2006) fulfill this condition. Here, the problem becomes equivalent to inference for a (continuous time) Gaussian hidden Markov model and could be solved by performing a forward–backward algorithm (Solin, 2016). This would reduce the computations to the solution of ordinary differential equations. We will discuss details of such an approach elsewhere. To deal with general kernels we will resort instead to a the well known variational sparse GP approximation with inducing points.

Optimal sparse Gaussian process The sparse variational Gaussian approximation follows the standard approach (Csató and Opper, 2002; Csató, 2002; Titsias, 2009) and its generalisation to a continuum likelihood (Batz et al., 2018; de G. Matthews et al., 2016). For completeness, we repeat the derivation here and more detailed in Appendix E. We approximate $\boldsymbol{q}_2(g)$ by a sparse likelihood GP $\boldsymbol{q}_2^s(g)$ with respect to the GP prior

$$\frac{dQ_2^s}{dP}(g) = \boldsymbol{q}_2^s(\boldsymbol{g}_s), \tag{21}$$

which depends only on a finite dimensional vector of function values $\boldsymbol{g}_s = (g(\boldsymbol{x}_1), \ldots, g(\boldsymbol{x}_L))^\top$ at a set of *inducing points* $\{\boldsymbol{x}_l\}_{l=1}^L$. With this approach it is again possible to marginalise out exactly all the infinitely many function values outside of the set of inducing points. The sparse likelihood $\boldsymbol{q}_2^s$ is optimised by minimising the Kullback–Leibler divergence

$$\mathrm{D}_{\mathrm{KL}}(Q_2^s \| Q_2) = \mathbb{E}_{Q_2^s}\left[\log \frac{\boldsymbol{q}_2^s(g)}{\boldsymbol{q}_2(g)}\right].$$

A short computation (Appendix E) shows that

$$q_2^s(\boldsymbol{g}_s) \propto e^{U^s(\boldsymbol{g}_s)} \qquad \text{with } U^s(\boldsymbol{g}_s) = \mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[U(g)\right],$$

where the conditional expectation is with respect to the GP prior measure given the function $\boldsymbol{g}_s$ at the inducing points. The explicit calculation requires the conditional expectations of $g(\boldsymbol{x})$ and of $(g(\boldsymbol{x}))^2$. We get

$$\mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[g(\boldsymbol{x})\right] = \boldsymbol{k}_s(\boldsymbol{x})^\top K_s^{-1} \boldsymbol{g}_s, \tag{22}$$

where $\boldsymbol{k}_s(\boldsymbol{x}) = (k(\boldsymbol{x}, \boldsymbol{x}_1), \ldots, k(\boldsymbol{x}, \boldsymbol{x}_L))^\top$ and K_s is the kernel matrix between inducing points. For the second expectation, we get

$$\mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[g^2(\boldsymbol{x})\right] = \left(\mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[g(\boldsymbol{x})\right]\right)^2 + \text{const.} \tag{23}$$

The constant equals the conditional variance of $g(\boldsymbol{x})$ which does not depend on the sparse set $\boldsymbol{g}_s$, but only on the locations of the sparse points. Because we are dealing now with a finite problem we can define the 'ordinary' posterior density of the GP at the inducing points with respect to the Lebesgue measure $d\boldsymbol{g}_s$. From Equation (20), (22), and (23), we conclude that the sparse posterior at the inducing variables is a multivariate Gaussian density

$$q_2^s(\boldsymbol{g}_s) = \mathcal{N}(\boldsymbol{\mu}_2^s, \Sigma_2^s), \tag{24}$$

with the covariance matrix given by

$$\Sigma_2^s = \left[K_s^{-1} \int_{\mathcal{X}} A(\boldsymbol{x}) \boldsymbol{k}_s(\boldsymbol{x}) \boldsymbol{k}_s(\boldsymbol{x})^\top d\boldsymbol{x}\, K_s^{-1} + K_s^{-1}\right]^{-1}, \tag{25}$$

and the mean

$$\boldsymbol{\mu}_2^s = \Sigma_2^s \left(K_s^{-1} \int_{\mathcal{X}} B(\boldsymbol{x}) \boldsymbol{k}_s(\boldsymbol{x}) d\boldsymbol{x}\right). \tag{26}$$

In contrast to other variational approximations (see for example (Lloyd et al., 2015; Hensman et al., 2015)) we obtain a closed analytic form of the variational posterior mean and

covariance which holds for arbitrary GP kernels. However, these results depend on finite dimensional integrals over the space $\mathcal{X}$ which cannot be computed analytically. This is different to the sparse approximation for the Poisson model with square link function (Lloyd et al., 2015), where similar integrals in the case of the squared exponential kernel can be obtained analytically. Hence, we resort to a simple Monte–Carlo integration, where *integration points* are sampled uniformly on $\mathcal{X}$ as

$$I_F = \int_{\mathcal{X}} F(\boldsymbol{x}) d\boldsymbol{x} \approx \frac{|\mathcal{X}|}{R} \sum_{r=1}^{R} F(\boldsymbol{x}_r).$$

The set of integration points $\{\boldsymbol{x}_r\}_{r=1}^R$ is drawn uniformly from the space $\mathcal{X}$.

Finally, from Equation (21) and (24) we obtain the mean function and the variance of the sparse approximation for every point $\boldsymbol{x} \in \mathcal{X}$, which is

$$\mu_2(\boldsymbol{x}) = \mathbb{E}_{Q_2}\left[g(\boldsymbol{x})\right] = \boldsymbol{k}_s(\boldsymbol{x})^\top K_s^{-1} \boldsymbol{\mu}_2^s, \tag{27}$$

and variance

$$(s_2(\boldsymbol{x}))^2 = k(\boldsymbol{x}, \boldsymbol{x}) - \boldsymbol{k}_s(\boldsymbol{x})^\top K_s^{-1} \left(\mathbf{I} - \Sigma_2^s K_s^{-1}\right) \boldsymbol{k}_s(\boldsymbol{x}), \tag{28}$$

where $\mathbf{I}$ is the identity matrix.

Optimal density for maximal intensity λ From Equation (14) we identify the optimal density as a Gamma density

$$q_2(\lambda) = \mathrm{Gamma}(\lambda | \alpha_2, \beta_2) = \frac{\beta_2^{\alpha_2}(\lambda)^{\alpha_2 - 1} \mathrm{e}^{-\beta_2 \lambda}}{\Gamma(\alpha_2)}, \tag{29}$$

where $\alpha_2 = N + \mathbb{E}_{Q_1}\left[\mathbf{1}_\Pi(\boldsymbol{x})\right] + \alpha_0$, $\beta_2 = \beta_0 + \int_{\mathcal{X}} d\boldsymbol{x}$ and $\Gamma(\cdot)$ is the gamma function. $\mathbf{1}_\Pi(\boldsymbol{x})$ denotes the indicator function being 1 if $\boldsymbol{x} \in \Pi$ and 0 otherwise and the integral is again solved by Monte Carlo integration. This defines the required expectations for updating q_1 by $\mathbb{E}_{Q_2}\left[\lambda\right] = \frac{\alpha_2}{\beta_2}$ and $\mathbb{E}_{Q_2}\left[\log \lambda\right] = \psi(\alpha_2) - \log \beta_2$, where $\psi(\cdot)$ is the digamma function.

Hyperparameters Hyperparameters of the model are (i) the covariance parameters $\boldsymbol{\theta}$ of the GP, (ii) the locations of the inducing points $\{\boldsymbol{x}_l\}_{l=1}^L$, and (iii) the prior parameters α_0, β_0 for the maximal intensity λ. The covariance parameters (i) $\boldsymbol{\theta}$ are optimised by gradient ascent following the gradient of the lower bound in Equation (12) with respect to $\boldsymbol{\theta}$ (Appendix F). As gradient ascent algorithm we employ the ADAM algorithm (Kingma and Ba, 2014). We perform always one step after the variational posterior q is updated as described before. (ii) The locations of the sparse GP $\{\boldsymbol{x}_l\}_{l=1}^L$ could in principle be optimised as well, but we keep them fixed and position them on a regular grid over the space $\mathcal{X}$. From this choice it follows that K_s is a Toeplitz matrix, when the kernel is translationally invariant. This could be inverted in $\mathcal{O}(L(\log L)^2)$ instead of $\mathcal{O}(L^3)$ operations (Press et al., 2007) but we do not employ this fact. Finally, (iii) the value for prior parameters α_0 and β_0 are chosen such that $p(\lambda)$ has a mean twice and standard deviation once the intensity one would expect for a homogeneous Poisson Process observing $\mathcal{D}$. The complete variational procedure is outlined in Algorithm 1.

Algorithm 1: Variational Bayes algorithm for sigmoidal Gaussian Cox process.

Init: $\mathbb{E}_Q\left[g(\boldsymbol{x})\right], \mathbb{E}_Q\left[(g(\boldsymbol{x}))^2\right]$ at $\mathcal{D}$ and integration points, and $\mathbb{E}_Q\left[\lambda\right], \mathbb{E}_Q\left[\log\lambda\right]$
while $\mathcal{L}$ *not converged* **do**
  Update q_1
    PG distributions at observations: $q_1(\boldsymbol{\omega}_N)$ with Eq. (17)
    Rate of latent process: $\Lambda_1(\boldsymbol{x}, \omega)$ at integration points with Eq. (19)
  Update q_2
    Sparse GP distribution: $\Sigma_2^s, \boldsymbol{\mu}_2^s$ with Eq. (25), (26)
    GP at $\mathcal{D}$ and integration points: $\mathbb{E}_{Q_2}\left[g(\boldsymbol{x})\right], \mathbb{E}_{Q_2}\left[(g(\boldsymbol{x}))^2\right]$ with Eq. (27), (28)
    Gamma-distribution of λ: α_2, β_2 with Eq. (29)
  Update kernel parameters with gradient ascent
end

3.2. Laplace approximation

In this section we will show that our variable augmentation method is well suited for computing a Laplace approximation (Bishop, 2006, chap. 4) to the joint posterior of the GP function $g(\cdot)$ and the maximal intensity λ as an alternative to the previous variational scheme. To do so we need the maximum a posteriori (MAP) estimate (equal to the mode of the posterior distribution) and a second order Taylor expansion around this mode. The augmentation method will be used to compute the MAP estimator iteratively using an EM algorithm.

Obtaining the MAP estimate In general, a proper definition of the posterior mode would be necessary, because the GP posterior is over a space of functions, which is an infinite dimensional object and does not have a density with respect to Lebesgue measure. A possibility to avoid this problem would be to discretise the spatial integral in the likelihood and to approximate the posterior by a multivariate Gaussian density for which the mode can then be computed by setting the gradient equal to zero. In this paper, we will use a different approach which defines the mode directly in function space and allows us to utilise the sparse GP approximation developed previously for the computations. A mathematically proper way would be to derive the MAP estimator by maximising a properly penalised log–likelihood. As discussed e.g. in Rasmussen and Williams (2006, chap. 6) for GP models with likelihoods which depend on finitely many inputs only, this penalty is given by the squared reproducing kernel Hilbert space (RKHS) norm that corresponds to the GP kernel. Hence, we would have

$$(g^*, \lambda^*) = \operatorname{argmin}_{g\in\mathcal{H}_k,\lambda}\left\{-\ln L(\mathcal{D}|g,\lambda) - \ln p(\lambda) + \frac{1}{2}\|g\|^2_{\mathcal{H}_k}\right\},$$

where $\|g\|^2_{\mathcal{H}_k}$ is the RKHS norm for the kernel k. This penalty term can be understood as a proper generalisation of a Gaussian log–prior density to function space. We will not give a formal definition here but work on a more heuristic level in the following. Rather than attempting a direct optimisation, we will use an EM algorithm instead, applying the

variable augmentation with the Poisson process and Pólya–Gamma variables introduced in the previous sections. In this case, the likelihood part of the resulting '$\mathcal{Q}$–function'

$$\mathcal{Q}((g,\lambda)|(g,\lambda)^{\text{old}}) \doteq \mathbb{E}_{P(\boldsymbol{\omega}_N,\Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}})}\left[\ln L(\mathcal{D},\boldsymbol{\omega}_N,\Pi_{\hat{\mathcal{X}}}|g,\lambda)\right] + \ln p(\lambda) - \frac{1}{2}\|g\|^2_{\mathcal{H}_k}, \quad (30)$$

that needs to be maximised in the M–step becomes (as in the variational approach before) the likelihood of *a Gaussian model* in the GP function g. Hence, we can argue that the function g which maximises $\mathcal{Q}$ is equal to the *posterior mean* of the resulting Gaussian model and can be computed without discussing the explicit form of the RKHS norm.

The conditional probability measure $P(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}})$ is easily obtained similar to the optimal measure Q_1 by not averaging over g and λ. This gives us straightforwardly the density

$$\boldsymbol{p}(\boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}}) = \boldsymbol{p}(\boldsymbol{\omega}_N|(g,\lambda)^{\text{old}})\boldsymbol{p}(\Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}}).$$

The first factor is

$$p(\boldsymbol{\omega}_N|(g,\lambda)^{\text{old}}) = \boldsymbol{p}(\boldsymbol{\omega}_N|(g,\lambda)^{\text{old}})\left|\frac{dP_{\boldsymbol{\omega}_N}}{d\boldsymbol{\omega}_N}\right| = \prod_{n=1}^{N} p_{\text{PG}}\left(\omega_n|1,\tilde{c}_n\right),$$

with $\tilde{c}_n = |g_n^{\text{old}}|$. The latent point process $\Pi_{\hat{\mathcal{X}}}$ is again a Poisson process density

$$\boldsymbol{p}(\Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}}) = \frac{dP_{\tilde{\Lambda}}}{dP_{\Lambda}}(\Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}}),$$

where the intensity is

$$\tilde{\Lambda}(\boldsymbol{x},\omega) = \lambda^{\text{old}}\sigma(-g^{\text{old}}(\boldsymbol{x}))p_{\text{PG}}\left(\omega|1,\tilde{c}(\boldsymbol{x})\right),$$

with $\tilde{c}(\boldsymbol{x}) = |g^{\text{old}}(\boldsymbol{x})|$. The first term in the $\mathcal{Q}$–function is

$$\begin{aligned} U(g,\lambda) &\doteq \mathbb{E}_{P(\boldsymbol{\omega}_N,\Pi_{\hat{\mathcal{X}}}|(g,\lambda)^{\text{old}})}\left[\ln L(\mathcal{D},\boldsymbol{\omega}_N,\Pi_{\hat{\mathcal{X}}}|g,\lambda)\right] \\ &= -\frac{1}{2}\int_{\mathcal{X}}\tilde{A}(\boldsymbol{x})g(\boldsymbol{x})^2 d\boldsymbol{x} + \int_{\mathcal{X}}\tilde{B}(\boldsymbol{x})g(\boldsymbol{x})d\boldsymbol{x}, \end{aligned}$$

with

$$\begin{aligned} \tilde{A}(\boldsymbol{x}) &= \sum_{n=1}^{N}\mathbb{E}_{P(\omega_n|(g,\lambda)^{\text{old}})}\left[\omega_n\right]\delta(\boldsymbol{x}-\boldsymbol{x}_n) + \int_0^\infty \mathbb{E}_{P(\omega|(g,\lambda)^{\text{old}})}\left[\omega\right]\tilde{\Lambda}(\boldsymbol{x},\omega)d\omega, \\ \tilde{B}(\boldsymbol{x}) &= \frac{1}{2}\sum_{n=1}^{N}\delta(\boldsymbol{x}-\boldsymbol{x}_n) - \frac{1}{2}\int_0^\infty \tilde{\Lambda}(\boldsymbol{x},\omega)d\omega. \end{aligned}$$

We have already tackled almost identical log–likelihood expressions in Section 3.1 (see Equation (20)). While for specific priors (with precision kernels given by differential operators) an exact treatment in terms of solutions of ODEs or PDEs is possible, we will again resort to the sparse GP approximation instead. The sparse version $U^s(\boldsymbol{g}_s,\lambda)$ is obtained by replacing $g(\boldsymbol{x}) \to \mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[g(\boldsymbol{x})\right]$ in $U(g,\lambda)$. From this we obtain the sparse $\mathcal{Q}$–function as

$$\mathcal{Q}^s((\boldsymbol{g}_s,\lambda)|(\boldsymbol{g}_s,\lambda)^{\text{old}}) \doteq U^s(\boldsymbol{g}_s,\lambda) + \ln p(\lambda) - \frac{1}{2}\boldsymbol{g}_s^\top K_s^{-1}\boldsymbol{g}_s. \quad (31)$$

The function values $\boldsymbol{g}_s$ and the maximal intensity λ that maximise Equation (31) can be found analytically by solving

$$\frac{\partial \mathcal{Q}^s}{\partial \boldsymbol{g}_s} = \mathbf{0} \text{ and } \frac{\partial \mathcal{Q}^s}{\partial \lambda} = 0.$$

The final MAP estimate is obtained after convergence of the EM algorithm and the desired sparse MAP solution for $g(x)$ is given by (see Equation (27))

$$g_{MAP}(\boldsymbol{x}) = \boldsymbol{k}_s(\boldsymbol{x})^\top K_s^{-1} \boldsymbol{g}^s$$

As for the variational scheme, integrals over the space $\mathcal{X}$ are approximated by Monte–Carlo integration. An alternative derivation of the sparse MAP solution can be based on restricting the minimisation of (30) to functions which are linear combinations of kernels centred at the inducing points and using the definition of the RKHS norm (see (Rasmussen and Williams, 2006, chap. 6)).

Sparse Laplace posterior To complete the computation of the Laplace approximation, we need to evaluate the quadratic fluctuations around the MAP solution. We will also do this with the previously obtained sparse approximation. The idea is that from the converged MAP solution, we define a sparse likelihood of the Poisson model via the replacement

$$L^s(\boldsymbol{g}_s, \lambda) \doteq L(\mathcal{D}|\mathbb{E}_{P(g|\boldsymbol{g}_s)}[g], \lambda)$$

For this sparse likelihood it is easy to compute the Laplace posterior using second derivatives. Here, the change of variables $\rho = \ln \lambda$ will be made to ensure that $\lambda > 0$. This results in an effective log–normal density over the maximal intensity rate λ. While we do not address hyperparameter selection for the Laplace posterior in this work, a straightforward approach, as suggested by Flaxman et al. (2017), could be to use cross validation to optimise the kernel parameters while finding the MAP estimate or to use the Laplace approximation to approximate the evidence. As in the variational case the inducing point locations $\{\boldsymbol{x}_l\}_{l=1}^L$ will be on a regular grid over space $\mathcal{X}$.

Note that for the Laplace approximation, the augmentation scheme is only used to compute the MAP estimate in an efficient way. There are no further mean–field approximations involved. This also implies, that dependencies between $\boldsymbol{g}_s$ and λ are retained.

3.3. Predictive density

Both variational and Laplace approximation yield a posterior distribution q over $\boldsymbol{g}_s$ and λ. The GP approximation at any given points in $\mathcal{X}$ is given by

$$q(g(\boldsymbol{x})) = \int\int p(g(\boldsymbol{x})|\boldsymbol{g}_s) q(\boldsymbol{g}_s, \lambda)\, d\boldsymbol{g}_s d\lambda,$$

which for both methods results in a normal density. To find the posterior mean of the intensity function at a point $\boldsymbol{x} \in \mathcal{X}$ one needs to compute

$$\mathbb{E}_Q[\Lambda(\boldsymbol{x})] = \mathbb{E}_Q\left[\lambda \int_{-\infty}^{\infty} \sigma(g(\boldsymbol{x}))\right].$$

For variational and Laplace posterior the expectation over λ can be computed analytically, leaving the expectation over $g(\boldsymbol{x})$, which is computed numerically via quadrature methods. To evaluate the performance of inference results we are interested in computing the likelihood on test data $\mathcal{D}_{\text{test}}$, generated from the ground truth. We will consider two methods:

Sampling GPs g from the posterior we calculate the (log) mean of the test likelihood

$$\begin{aligned}\ell(\mathcal{D}_{\text{test}}) &= \ln \mathbb{E}_P\left[L(\mathcal{D}_{\text{test}}|\Lambda)|\mathcal{D}\right] \approx \ln \mathbb{E}_Q\left[L(\mathcal{D}_{\text{test}}|\Lambda)\right] \\ &= \ln \mathbb{E}_Q\left[\exp\left(-\int_{\mathcal{X}} \lambda\sigma(g(\boldsymbol{x}))d\boldsymbol{x}\right) \prod_{\boldsymbol{x}_n \in \mathcal{D}_{\text{test}}} \lambda\sigma(g(\boldsymbol{x}_n))\right]\end{aligned} \tag{32}$$

where the integral in the exponent is approximated by Monte–Carlo integration. The expectation is approximated by averaging over 2×10^3 samples from the inferred posterior Q of λ and g at the observations of $\mathcal{D}_{\text{test}}$ and the integration points.
Instead of sampling one can also obtain an analytic approximation for the log test likelihood in Equation (32) by a second order Taylor expansion around the mean of the obtained posterior. Applying this idea to the variational mean field posterior we get

$$\begin{aligned}\ell(\mathcal{D}_{\text{test}}) \approx &\ln L(\mathcal{D}_{\text{test}}|\Lambda_Q) + \frac{1}{2}\mathbb{E}_Q\left[(\boldsymbol{g}_s - \boldsymbol{\mu}_2^s)^\top \mathbf{H}_{\boldsymbol{g}_s}\big|_{\Lambda_Q} (\boldsymbol{g}_s - \boldsymbol{\mu}_2^s)\right] \\ &+ \frac{1}{2} H_\lambda|_{\Lambda_Q} \operatorname{Var}_Q(\lambda),\end{aligned} \tag{33}$$

where $\Lambda_Q(\boldsymbol{x}) = \mathbb{E}_Q[\lambda]\, \sigma(\mathbb{E}_Q[g(\boldsymbol{x})])$ and $\mathbf{H}_{\boldsymbol{g}_s}\big|_{\Lambda_Q}$, $H_\lambda|_{\Lambda_Q}$ are the second order derivative of the likelihood in Equation (1) with respect to $\boldsymbol{g}_s$ and λ at Λ_Q. While an approximation only involving the first term would neglect the uncertainties in the posterior (as done by John and Hensman (2018)), the second and third term take these into account.

4. Results

Generating data from the model To evaluate the two newly developed algorithms we generate data according to the sigmoidal Gaussian Cox process model

$$\begin{aligned} g &\sim \boldsymbol{p}_{\text{GP}}(\cdot|0, k), \\ \mathcal{D} &\sim \boldsymbol{p}_\Lambda(\cdot), \end{aligned}$$

where $\boldsymbol{p}_\Lambda(\cdot)$ is the Poisson process density over sets of point with $\Lambda(\boldsymbol{x}) = \lambda\sigma(g(\boldsymbol{x}))$ and $\boldsymbol{p}_{\text{GP}}(\cdot|0, k)$ is a GP density with mean 0 and covariance function k. As kernel we choose a squared exponential function

$$k(\boldsymbol{x}, \boldsymbol{x}') = \theta \prod_{i=1}^{d} \exp\left(-\frac{(x_i - x_i')^2}{2\nu_i^2}\right),$$

where the hyperparameters are scalar θ and length scales $\boldsymbol{\nu} = (\nu_1, \ldots, \nu_d)^\top$. Sampling of the inhomogeneous Poisson process is done via *thinning* (Lewis and Shedler, 1979; Adams et al., 2009). We assume that hyperparameters are known for subsequent experiments with data sampled from the generative model.

Benchmarks for sigmoidal Gaussian Cox process inference We compare the proposed algorithms to two alternative inference methods for the sigmoidal Gaussian Cox process model. As an exact inference method we use the sampling approach of Adams et al. (2009)[4]. In terms of speed, a competitor is a different variational approach given by Hensman et al. (2015) who proposed to discretise space $\mathcal{X}$ in several regular bins with size Δ. Then the likelihood in Equation (1) is approximated by

$$L(\mathcal{D}|\lambda\sigma(g(\boldsymbol{x}))) \approx \prod_i p_{\text{po}}(n_i|\lambda\sigma(g(\boldsymbol{x}_i))\Delta),$$

where p_{po} is the Poisson distribution conditioned on the mean parameter, $\boldsymbol{x}_i$ is the centre of bin i, and n_i the number of observations within this bin. Using a (sparse) Gaussian variational approximation the corresponding Kullback–Leibler divergence is minimised by gradient ascent to find the optimal posterior over the GP g and a point estimate for λ. This method was originally proposed for the log Cox-process ($\Lambda(\boldsymbol{x}) = e^{g(\boldsymbol{x})}$), but with the elegant GPflow package (Matthews et al., 2017) implementation of the scaled sigmoid link function is straightforward. It should be noted, that this method requires numerical integration over the sigmoid link function to evaluate the variational lower bound at every spatial bin and every gradient step, since it does not make use of our augmentation scheme (see Section 5 for discussion, how the proposed augmentation can be used for this model). We refer to this inference algorithm as 'variational Gauss'. To have fair comparison between the different methods, the inducing points for all algorithms (except for the sampler) are equal and the number of bins used to discretise the domain $\mathcal{X}$ for the variational Gauss algorithm is set equal to the number of integration points used for the MC integration in the variational mean field and the Laplace method.

Experiments on data from generative model As an illustrative example we sample a one dimensional Poisson process with the generative model and perform inference with the sampler (2×10^3 samples after 10^3 burn-in iterations), the mean field algorithm, the Laplace approximation and the variational Gauss. In Figure 1 **(a)**–**(d)** the different posterior mean intensity functions with their standard deviations are shown. For **(b)**–**(d)** 50 regularly spaced inducing points are used. For **(b)**–**(c)** 2×10^3 random integration points are drawn uniformly over the space $\mathcal{X}$, while for **(d)** $\mathcal{X}$ is discretised into the same number of bins. All algorithms recover the true intensity well. The mean field and the Laplace algorithm show smaller posterior variance compared to the sampler. The fastest inference result is obtained by the Laplace algorithm in 0.02 s, followed by the mean field (0.09), variational Gauss (80) and the sampler (1.8×10^3). The fast convergence of the Laplace and the variational mean field algorithm is illustrated in Figure 1 **(e)**, where objective functions of our two algorithms (minus the maximum they converged to) is shown as a function of run time. Both algorithms reach a plateau in only a few (~ 6) iterations. To compare performance in terms of log expected test likelihood ℓ_{test} (test sets $\mathcal{D}_{\text{test}}$ sampled from the ground truth), we averaged results over ten independent data sets. The posterior of the sampler yields the highest value with 875.5, while variational ($\ell_{\text{test}} = 686.2$, approximation by Equation (33) yields 686.5), variational Gauss (686.7) and Laplace (686.1) yield all similar results (see also Figure 4 **(a)**). The posterior density of the maximal intensity λ is shown in Figure 1 **(f)**.

4. To increase efficiency, the GP values g are sampled by elliptical slice sampling (Murray et al., 2010).

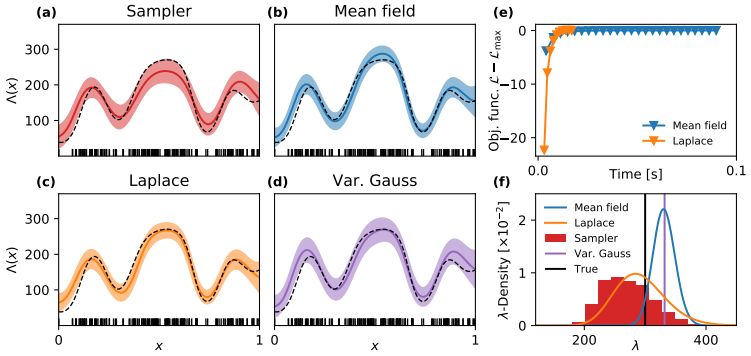


Figure 1: **Inference on 1D dataset.** **(a)**–**(d)** Inference result for sampler, mean field algorithm, Laplace approximation, and variational Gauss. Solid coloured lines denote the mean intensity function, shaded areas mean ± standard deviation, and dashed black lines the true rate functions. Vertical bars are observations $\mathcal{D}$. **(e)** Convergence of mean field and EM algorithm. Objective functions (Lower bound for mean–field and log likelihood for EM algorithm, shifted such that convergence is at 0) as function of run time (triangle marks one finished iteration of the respective algorithm). **(f)** Inferred posterior densities over the maximal intensity λ. Variational Gauss provides only a point estimate. Black vertical bar denotes the true λ.

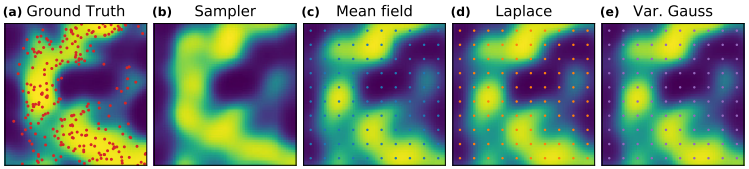


Figure 2: **Inference on 2D dataset.** **(a)** Ground truth intensity function $\Lambda(\boldsymbol{x})$ with observed dataset $\mathcal{D}$ (red dots).**(b)**–**(e)** Mean posterior intensity of the sampler, mean field algorithm, Laplace, and variational Gauss are shown. 100 inducing points on a regular grid (shown as coloured points) and 2500 integration points/bins are used.

In Figure 2 we show inference results for a two dimensional Cox process example. 10×10 inducing points and 2500 integration points/bins are used for mean field, Laplace and variational Gauss algorithm. The posterior mean of sampler **(b)**, of the mean field **(c)**, of the Laplace **(d)** and of the variational Gauss algorithm **(e)** recover the true intensity rate $\Lambda(\boldsymbol{x})$ **(a)** well.

To evaluate the role of the number of inducing points and number of integration points we generate 10 test sets $\mathcal{D}_{\text{test}}$ from a process with the same intensity as in Figure 2**(a)**. We evaluate the log expected likelihood (Equation (32)) on these test sets and compute the average. The result is shown for different numbers of inducing points (Figure 3**(a)** with 2500 integration points) and different numbers of integration points (Figure 3**(b)** with 10×10 inducing points). To account for randomness of integration points the fitting is repeated five times and the shaded area is between the minimum and maximum obtained by these fits. For all approximate algorithms the log predictive test likelihood saturates already for few inducing points (≈ 49 (7×7)) of the sparse GP. However, as expected, the inference approximations are slightly inferior to the sampler. The log expected test likelihood is hardly affected by the number of integration points as shown in Figure 3 **(b)**. Also the approximated test likelihood for the mean field algorithm in Equation (33) yields good estimates of the sampled value (dashed line in **(a)** and **(b)**). In terms of runtime (Figure 4 **(c)**–**(d)**) the mean field algorithm and the Laplace approximation are superior by more than one order of magnitude to the variational Gauss algorithm for this particular example. Difference increases with increasing number of inducing points.

In Figure 4 the four algorithms are compared on five different data sets sampled from the generative model. As we observed for the previous examples the three different approximating algorithms yield qualitatively similar performance in terms of log test likelihood ℓ_{test}, but the sampler is superior. Again the approximated test likelihood in Equation (33) (blue star) provides good estimate of the sampled value. In addition we provide the approximated root mean squared error (RMSE, evaluated on a fine grid and normalised by maximal intensity λ) between inferred mean and ground truth. In terms of run time the mean field and Laplace algorithm are by at least on order of magnitude faster than the vari-

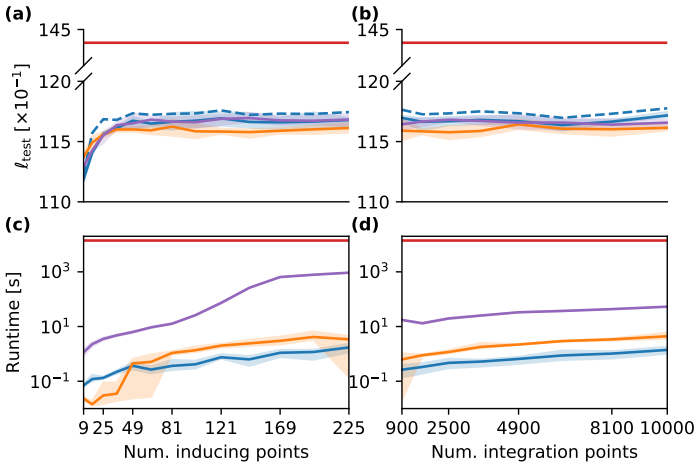


Figure 3: **Evaluation of inference.** **(a)** The log expected predictive likelihood averaged over ten test sets as a function of the number of inducing points. Number of integration points/bins is fixed to 2500. Results for sampler in (red), mean field (blue), Laplace (orange), and variational Gauss (purple) algorithm. Solid line denotes mean over five fits (same data), and shaded area denotes min. and max. result. Dashed blue line shows the approximated log expected predictive likelihood for the mean field algorithm. **(b)** Same as (a), but as function of number of integration points. Number of inducing points is fixed to 10×10. Below: Run time of the different algorithms as function of number of inducing points **(c)** and number of integration points **(d)**. Data are the same as in Figure 2.

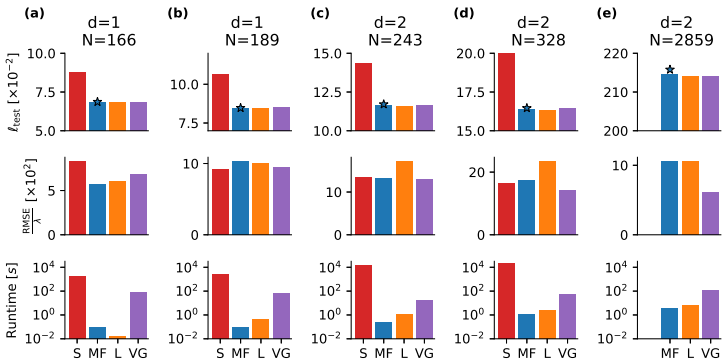


Figure 4: **Performance on different artificial datasets.** The sampler (S), the mean field algorithm (MF), the Laplace (L), and variational Gauss (VG) are compared on five different datasets with d–dimensions and N observations (one column corresponds to one dataset). Top row: Log expected test likelihood of the different inference results. The star denotes the approximated test likelihood of the variational algorithm. Center row: The approximated root mean squared error (normalised by true maximal intensity rate λ). Bottom row: Run time in seconds. The dataset **(e)** is intractable for the sampler due to the many observations. Data in Figure 1 and 2 correspond to **(a)** and **(c)**.

ational Gauss algorithm. In general, the mean–field algorithm seems to be slightly faster than the Laplace.

General data sets and comparison to the approach of Lloyd et al. Next, we test our variational mean field algorithm on data sets not coming from the generative model. On such data sets we do not know, whether our model provides a good prior. As discussed previously an alternative model was proposed by Lloyd et al. (2015) making use of the link function $\Lambda(\boldsymbol{x}) = g^2(\boldsymbol{x})$. While the sigmoidal Gaussian Cox process with the proposed augmentation scheme has analytic updates for the variational posterior, in case of the squared Gaussian Cox process the likelihood integral can be solved analytically and does not need to be sampled (if the kernel is a squared exponential and the domain is rectangular). Both algorithms rely on the sparse GP approximation. To compare the two methods empirically first we consider one dimensional data generated using a known intensity function. We choose $\Lambda(x) = 2\exp(-x/15) + \exp(-(x-25)^2/100)$ on an interval $[0, 50]$ already proposed by Adams et al. (2009). We generate three training and test sets, where we scale this rate function by factors of 1, 10, and 100 and fit the sigmoidal and squared Gaussian Cox

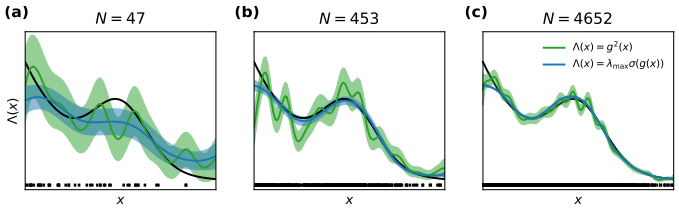


Figure 5: **1D example.** Observations (black bars) are sampled from the same function (black line) scaled by **(a)** 1, **(b)** 10, and **(c)** 100. Blue and green line show the mean posterior of the sigmoidal and squared Gaussian Cox process, respectively. Shaded area denotes mean ± standard deviation.

	$\Lambda(x) = \lambda_{\max}\sigma(g(x))$			$\Lambda(x) = g^2(x)$		
N	Runtime [s]	RMSE	ℓ_{test}	Runtime [s]	RMSE	ℓ_{test}
47	0.27 ± 0.30	0.24 ± 0.02	-43.43 ± 0.42	0.41 ± 0.05	0.24	-44.26 ± 0.09
453	0.50 ± 0.04	0.97 ± 0.13	720.81 ± 0.28	0.23 ± 0.05	2.11	710.43 ± 1.38
4652	0.41 ± 0.01	7.68 ± 0.75	17497.31 ± 2.13	0.79 ± 0.09	8.16	17496.75 ± 1.65

Table 1: **Benchmarks for Figure 5** The mean and standard deviation of runtime, RMSE, and log expected test likelihood for Figure 5**(a)**–**(c)** obtained from 5 fits. Note that the RMSE for $\Lambda(\boldsymbol{x}) = g^2(\boldsymbol{x})$ has no standard deviation, because the inference algorithm is deterministic.

process with their corresponding variational algorithm to each training set[5]. The number of inducing points is 40 in this example. For our variational mean field algorithm we used 5000 integration points. The posterior intensity $\Lambda(\boldsymbol{x})$ for the three data sets can be seen in Figure 5. The model with the sigmoidal link function infers smoother posterior functions with smaller variance compared to the posterior with the squared link function. For data sets shown in Figure 5 we run the fits five times and report mean and standard deviation of runtime, RMSE and log expected test likelihood ℓ_{test} in Table 1. Run times of the two algorithms are comparable, where for the intermediate data set the algorithm with the squared link function is faster while for the largest data set the one with the sigmoidal link function converges first. RMSE and ℓ_{test} are also comparable except for the intermediate data set, where the sigmoidal model is the superior one.

Next we deal with two real world two dimensional data sets for comparison. The first one is neuronal data, where spiking activity was recorded from a mouse, that was freely moving in an arena (For The Biology Of Memory and Sargolini, 2014; Sargolini et al., 2006). Here we consider as data $\mathcal{D}$ the position of the mouse when the recorded cell fired and the observations are randomly assigned to either training or test set. In Figure 6 **(a)**

5. We thank Chris Lloyd and Tom Gunter for providing the code for inferring the variational posterior of the squared Gaussian Cox process.

the observations in the training set ($N = 583$) are shown. In Figure 6 **(b)** and **(c)** the variational posterior's mean intensity $\Lambda(\boldsymbol{x})$ is shown obtained for the sigmoidal and the squared link function, respectively, inferred with a regular grid of 20×20 inducing points. As in Figure 5 we see that the sigmoidal posterior is the smoother one. The major difference between the two algorithms (apart from the link function) is the fact that for the sigmoidal model we are required to sample an interval over the space. We investigate the effect of the number of integration points in terms of runtime[6] and log expected test likelihood in Figure 6 **(d)**. First, we observe regardless of the number of integration points that the variational posterior of the squared link function yields the superior expected test likelihood. For the sigmoidal model the test likelihood does not improve significantly with more integration points. Runtimes of both algorithms are comparable, when 5000 integration points are chosen. A speed up for our mean field algorithm is achieved by first fitting the model with 1000 integration points and once converged, redrawing the desired number of integration points and rerun the algorithm (dotted line in Figure 6**(d)**). This method allows for a significant speed up without loss in terms of test likelihood ℓ_{test}. The variational mean-field algorithm with the sigmoid link function is faster with up to 5000 integration points and equally fast with 10000 integration points.

As second data set we consider the Porto taxi data set (Moreira-Matias et al., 2013). This data contains trajectories of taxi travels from the years 2013/14 in the city of Porto. As John and Hensman (2018) we consider the pick-ups as observations of a Poisson process[7]. We consider 20000 taxi rides randomly split into training and test set ($N = 10017$ and $N = 9983$, respectively). The training set is shown in Figure 6**(e)**. Inducing points are positioned on a regular grid of 20×20. The variational posterior mean of the respective intensity is shown in Figure 6 **(f)** and **(g)**. With as many data points as in these data the differences between the two models are more subtle as compared to **(b)** and **(c)**. In terms of test likelihood ℓ_{test} the variational posterior of the sigmoidal model (with ≥ 2000 integration points) outperforms the model with squared link function (Figure 6 **(h)**). For similar test likelihoods ℓ_{test} our variational algorithm is $\sim 2\times$ faster than the variational posterior with squared link function. The results show that the choice of number of integration points reduces to the question of speed vs accuracy trade–off. As for the previous data set, the strategy of first fitting the posterior with 1000 integration points and then with the desired number of integration points (dotted line) proves that we can get a significant speed up without loosing predictive power.

5. Discussion and Outlook

Using a combination of two known variable augmentation methods, we derive a conjugate representation for the posterior measure of a sigmoidal Gaussian Cox process. The approximation of the augmented posterior by a simple mean field factorisation yields an efficient variational algorithm. The rationale behind this method is that the variational updates in the conjugate model are explicit and analytical and do not require (black–box) gradient

6. Note, that - in contrast to Figures 3 and 4 - the runtime is displayed on linear scale, meaning both algorithms are of same order of magnitude.
7. As John and Hensman (2018) report some regions to be highly peaked we consider only pickups happening within the coordinates $(41.147, -8.58)$ and $(41.18, -8.65)$ in order to exclude those regions.

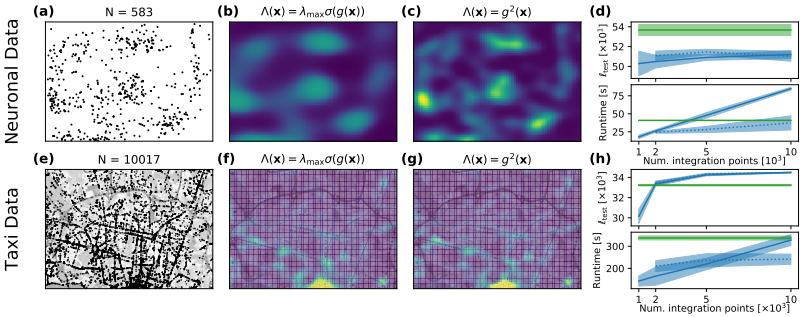


Figure 6: **Fits to real world data sets.** **(a)** Position of the mouse while the recorded neuron spiked. **(b)** Posterior mean obtained by the variational mean–field algorithm for the sigmoidal Gaussian Cox process. **(c)** Same as in (b) for the variational approximation of the squared Gaussian Cox process. **(d)** Log expected test–likelihood ℓ_{test} and runtime as function of number of integration points for both algorithms. The dotted line is obtained by first fitting the sigmoidal model with 1000 integration points and then with the number that is indicated on the x-axis. Shaded area is mean $\pm$ standard deviation obtained in 5 repeated fits. **(e)**–**(h)** Same as (a)–(d), but for a data set, where the observations are positions of taxi pick–ups in the city of Porto.

descent methods. In fact, a comparison with a different variational algorithm for the same model - not based on augmentation, but on direct approximation of the posterior with a Gaussian - shows that the qualities of inference for both approaches are similar, while the mean field algorithm is at least one order of magnitude faster. We use the same variable augmentation method for computation of the MAP estimate for the (unaugmented) posterior by a fast EM algorithm. This is finally applied to the calculation of Laplace's approximation. Both methods yield an explicit result for the approximate GP posterior. Since the corresponding effective likelihood contains a continuum of the GP latent variables, the exact computations of means and marginal variances would require the inversion of a linear operator instead of a simpler matrix inverse. While for specific priors, this problem could be solved by PDE or ODE methods, we resort to a well known sparse GP approach with inducing points in this paper. We can apply this to arbitrary kernels but need to solve spatial integrals over the domain. These can be (at least for moderate dimensionality) well approximated by simple Monte Carlo integration. Advantage of this approach is, that one is not limited to rectangular domains. The only requirement is that the volume $|\mathcal{X}|$ is known. An alternative Poisson model for which similar spatial integrals can be performed analytically (Lloyd et al., 2015) within the sparse GP approximation (limited to squared exponential kernels and rectangular domains) is based on a quadratic link function (Lloyd et al., 2015; Flaxman et al., 2017; John and Hensman, 2018). We compare our variational algorithm with the variational algorithm of Lloyd et al. (2015) on different data sets and observe that both algorithms act on the same order of magnitude in terms of runtime (with slight advantages for our variational mean field algorithm). As expected, we show that whether one or the other model is better in predictive power is highly data dependent.

As an alternative to the Monte Carlo integration in our approach we could avoid the infinite dimensionality of the latent GP from the beginning by working with a binning scheme for the Poisson observations as in Hensman et al. (2015). It would be straightforward to adopt our augmentation method to this case. The resulting Poisson likelihoods would then be augmented by pairs of Poisson and Pólya–Gamma variables (see Donner and Opper (2017)) for each bin. This approach could be favourable when the number of observed data points becomes very large, because the discretisation method does not scale with the number data points but with the resolution of discretisation. However, we do expect, that any approach based on either spatial discretisation or on the sparse, inducing point method would become problematic for large or high dimensional domains $\mathcal{X}$. Alternative methods based on spectral representations of kernels (Knollmüller et al., 2017; John and Hensman, 2018) are promising for tackling those problems.

It will be interesting to apply the variable augmentation method to other Bayesian models with the sigmoid link function. For example, the inherent boundedness of the resulting intensity can be crucial for point processes such as the nonlinear *Hawkes process* (Hawkes, 1971) which is widely used for modelling stock market data (Embrechts et al., 2011) or seismic activity (Ogata, 1998). For other point process models the sigmoid function appears naturally. We mention the kinetic Ising model, a Markov jump process (Donner and Opper, 2017) which was originally introduced to model the dynamics of classical spin systems in physics. More recently it was used to model the joint activity of neurons (Dunn et al., 2015). Finally, a Gaussian process density model introduced by (Murray et al., 2009) can be treated by the augmentations developed in this work (Donner and Opper, 2018).

Acknowledgments

CD was supported by the Deutsche Forschungsgemeinschaft (GRK1589/2) and partially funded by Deutsche Forschungsgemeinschaft (DFG) through grant CRC 1294 "Data Assimilation", Project (A06) "Approximative Bayesian inference and model selection for stochastic differential equations (SDEs)".

Appendix A. Poisson processes

In this paragraph we briefly summarise those properties of a Poisson process, which are relevant for this work. For a thorough and more complete description we recommend the concise book by Kingman (1993), particularly chapter 3 and 5.

We consider a general space $\mathcal{Z}$ and a countable subset $\Pi_{\mathcal{Z}} = \{\boldsymbol{z}; \boldsymbol{z} \in \mathcal{Z}\}$.

Definition of a Poisson process A random countable subset $\Pi_{\mathcal{Z}} \subset \mathcal{Z}$ is a Poisson process on $\mathcal{Z}$, if

i) for any sequence of disjoint subsets $\{\mathcal{Z}_k \subset \mathcal{Z}\}_{k=1}^K$ the cardinality of the union $N(\mathcal{Z}_k) \doteq |\{\Pi_{\mathcal{Z}} \cap \mathcal{Z}_k\}|$ is independent of $N(\mathcal{Z}_l)$ for all $l \neq k$.

ii) $N(\mathcal{Z}_k)$ is Poisson distributed with mean $\int_{\mathcal{Z}_k} \Lambda(\boldsymbol{z}) d\boldsymbol{z}$, and mean measure $\Lambda(\boldsymbol{z}) : \mathcal{X} \to \mathbb{R}^+$.

If the mean measure is constant ($\Lambda(\boldsymbol{z}) = \Lambda$) the Poisson process is *homogeneous*, and *inhomogeneous* otherwise.

Campbell's Theorem Let $\Pi_{\mathcal{Z}}$ be a Poisson process on $\mathcal{Z}$ with mean measure $\Lambda(\boldsymbol{z})$. Furthermore, we define a function $h(\boldsymbol{z}) : \mathcal{Z} \to \mathbb{R}$ and the sum

$$H(\Pi_{\mathcal{Z}}) = \sum_{\boldsymbol{z} \in \Pi_{\mathcal{Z}}} h(\boldsymbol{z}).$$

If $\Lambda(\boldsymbol{z}) < \infty$ for $\boldsymbol{z} \in \mathcal{Z}$, then

$$\mathbb{E}_{P_\Lambda}\left[e^{\xi H(\Pi_{\mathcal{Z}})}\right] = \exp\left\{\int_{\mathcal{Z}} \left(e^{\xi h(\boldsymbol{z})} - 1\right) \Lambda(\boldsymbol{z}) d\boldsymbol{z}\right\}, \tag{36}$$

for any $\xi \in \mathbb{C}$, such that the integral converges. P_Λ is the probability measure of a Poisson process with intensity $\Lambda(\boldsymbol{z})$. Mean and variance are obtained as

$$\mathbb{E}_{P_\Lambda}\left[H(\Pi_{\mathcal{Z}})\right] = \int_{\mathcal{Z}} h(\boldsymbol{z}) \Lambda(\boldsymbol{z}) d\boldsymbol{z},$$
$$\mathrm{Var}_{P_\Lambda}\left[H(\Pi_{\mathcal{Z}})\right] = \int_{\mathcal{Z}} [h(\boldsymbol{z})]^2 \Lambda(\boldsymbol{z}) d\boldsymbol{z}.$$

Note, that Equation (36) defines the *characteristic functional* of a Poisson process.

Marked Poisson process Let $\Pi_{\mathcal{Z}} = \{\boldsymbol{z}_n\}_{n=1}^N$ a Poisson process on $\mathcal{Z}$ with intensity $\Lambda(\boldsymbol{z})$. Then $\Pi_{\hat{\mathcal{Z}}} = \{(\boldsymbol{z}_n, \boldsymbol{m}_n)\}_{n=1}^N$ is again a Poisson process on the product space $\hat{\mathcal{Z}} = \mathcal{Z} \times \mathcal{M}$, if $\boldsymbol{m}_n \sim p(\boldsymbol{m}_n|\boldsymbol{z}_n)$ is drawn independently at each $\boldsymbol{z}_n$. The $\boldsymbol{m}_n \in \mathcal{M}$ are the so–called 'marks', and the resulting Process is a *marked Poisson process* with intensity

$$\Lambda(\boldsymbol{z}, \boldsymbol{m}) = \Lambda(\boldsymbol{z}) p(\boldsymbol{m}|\boldsymbol{z}).$$

It is straightforward to extend Campbell's theorem and to show that the characteristic functional of such a process is

$$\mathbb{E}_{P_\Lambda}\left[e^{\xi H(\Pi_{\hat{\mathcal{Z}}})}\right] = \exp\left\{\int_{\hat{\mathcal{Z}}} \left(e^{\xi h(\boldsymbol{z},\boldsymbol{m})} - 1\right) \Lambda(\boldsymbol{z}, \boldsymbol{m})\, d\boldsymbol{m} d\boldsymbol{z}\right\}, \tag{37}$$

with $h(\boldsymbol{z}, \boldsymbol{m}) : \hat{\mathcal{Z}} \to \mathbb{R}$ and $H(\Pi_{\hat{\mathcal{Z}}}) = \sum_{(\boldsymbol{z},\boldsymbol{m}) \in \Pi_{\hat{\mathcal{Z}}}} h(\boldsymbol{z}, \boldsymbol{m})$.

Appendix B. The Pólya-Gamma density

The Pólya-Gamma density (Polson et al., 2013) has the useful property, that it allows to represent the inverse hyperbolic cosine by an infinite Gaussian mixture as

$$\cosh^{-b}(c/2) = \int_0^\infty \exp\left(-\frac{c^2}{2}\omega\right) p_{\mathrm{PG}}(\omega|b, 0) d\omega,$$

with parameter $b > 0$. Furthermore, one can define a *tilted Pólya-Gamma density* as

$$p_{\mathrm{PG}}(\omega|b, c) = \frac{\exp\left(-\frac{c^2}{2}\omega\right)}{\cosh^{-b}(c/2)} p_{\mathrm{PG}}(\omega|b, 0).$$

From those two equations the moment generating function can be obtained from the basic definition, being

$$\int_0^\infty e^{\xi\omega} p_{\mathrm{PG}}(\omega|b, c) d\omega = \frac{\cosh^b(c/2)}{\cosh^b\left(\sqrt{\frac{c^2/2 - \xi}{2}}\right)},$$

and differentiating with respect to ξ at $\xi = 0$ yields the first moment

$$\mathbb{E}_{p_{\mathrm{PG}}}[\omega] = \frac{b}{2c} \tanh(c/2).$$

Appendix C. Variational inference for stochastic processes

Densities for random processes A stochastic process X with probability measure $P(X)$ often has no density with respect to Lebesgue measure, since X can be an infinite dimensional object such as a function for the case of a Gaussian process. However, one can define densities with respect to another (reference) measure $R(X)$ written as

$$\boldsymbol{p}(X) = \frac{dP}{dR}(X), \tag{38}$$

if $R(X)$ is absolutely continuous with respect to $P(X)$ (if $R(X) = 0$ then $P(X) = 0$). Using such a density, expectations are

$$\mathbb{E}_P[f(X)] = \int f(X) dP(X) = \int f(x) \boldsymbol{p}(x) dR(X) = \mathbb{E}_R[f(x)\boldsymbol{p}(x)].$$

The density in Equation (38) is known as the *Radon–Nikodým derivative* of R with respect to P (Konstantopoulos et al., 2011).

Poisson process density As specific example consider the prior density of the Poisson process in Equation (9), which is defined with respect to a reference measure

$$\boldsymbol{p}_\Lambda(\Pi_{\mathcal{Z}}) = \frac{dP_\Lambda}{dP_{\Lambda_0}}(\Pi_{\mathcal{Z}}) = \exp\left(-\int_{\mathcal{Z}} (\Lambda(\boldsymbol{z}) - \Lambda_0(\boldsymbol{z})) d\boldsymbol{z}\right) \prod_{\boldsymbol{z}_n \in \Pi_{\mathcal{Z}}} \frac{\Lambda(\boldsymbol{z}_n)}{\Lambda_0(\boldsymbol{z}_n)},$$

where P_{Λ_0} is the probability measure with intensity Λ_0 and the expectation is defined as

$$\mathbb{E}_{P_\Lambda}\left[\sum_{\boldsymbol{z}_n\in\Pi_\mathcal{Z}} u(\boldsymbol{z}_n)\right] = \mathbb{E}_{P_{\Lambda_0}}\left[\boldsymbol{p}_\Lambda(\Pi_\mathcal{Z})\sum_{\boldsymbol{z}_n\in\Pi_\mathcal{Z}} u(\boldsymbol{z}_n)\right]. \tag{39}$$

Calculating the expectation of $e^{\xi H(\Pi_\mathcal{Z})}$ with Equation (39) we identify the characteristic function of a Poisson process (see Equation (37)) with intensity $\Lambda(\boldsymbol{z})$.

Kullback-Leibler divergence Using these densities we can express the Kullback-Leibler divergence between two probability measures.

The KL–divergence between $\boldsymbol{q}(X)$ and $\boldsymbol{p}(X)$ is defined as

$$\mathrm{D}_{\mathrm{KL}}(Q\|P) = \mathbb{E}_Q\left[\log\frac{dQ}{dP}(X)\right] = \int \log\frac{\boldsymbol{q}(X)}{\boldsymbol{p}(X)} dQ(X),$$

where

$$\boldsymbol{q}(X) = \frac{dQ}{dR}(X),$$

and where $R(X)$ also is absolutely continuous to $Q(X)$. The KL–divergence does not depend on the reference measure $R(X)$.

Appendix D. The posterior point process is a marked Poisson process

Here we prove that the optimal variational posterior point process in Equation (18) again is a Poisson process using Campbell's theorem. As posterior process in Equation (18) one gets

$$\boldsymbol{q}(\Pi_\mathcal{Z}) = \frac{dQ}{dP_\lambda}(\Pi_\mathcal{Z}) = \frac{\prod_{\boldsymbol{z}_m\in\Pi_\mathcal{Z}} e^{f(\boldsymbol{z}_m)}}{\mathbb{E}_{P_\lambda}\left[\prod_{\boldsymbol{z}_m\in\Pi_\mathcal{Z}} e^{f(\boldsymbol{z}_m)}\right]} = \frac{\prod_{\boldsymbol{z}_m\in\Pi_\mathcal{Z}} e^{f(\boldsymbol{z}_m)}}{\exp\left(\int_\mathcal{Z}(e^{f(\boldsymbol{z})}-1)\lambda(\boldsymbol{z})d\boldsymbol{z}\right)},$$

where $\Pi_\mathcal{Z}$ is some random set of points on space $\mathcal{Z}$ and P_λ is a random Poisson measure with intensity $\lambda(\boldsymbol{z})$. To proof, that the resulting point process density $\boldsymbol{q}(\Pi_\mathcal{Z})$ is again a Poisson process we calculate the characteristic functional for some arbitrary function $h:\mathcal{Z}\to\mathbb{R}$

$$\begin{aligned}
\mathbb{E}_Q\left[\prod_{\boldsymbol{z}_m\in\Pi_\mathcal{Z}} e^{h(\boldsymbol{z}_m)}\right] &= \frac{\mathbb{E}_{P_\lambda}\left[\prod_{\boldsymbol{z}_m\in\Pi_\mathcal{Z}} e^{h(\boldsymbol{z}_m)+f(\boldsymbol{z}_m)}\right]}{\exp\left(\int_\mathcal{Z}(e^{f(\boldsymbol{z})}-1)\lambda(\boldsymbol{z})d\boldsymbol{z}\right)} \\
&= \frac{\exp\left(\int_\mathcal{Z}(e^{h(\boldsymbol{z})+f(\boldsymbol{z})}-1)\lambda(\boldsymbol{z})d\boldsymbol{z}\right)}{\exp\left(\int_\mathcal{Z}(e^{f(\boldsymbol{z})}-1)\lambda(\boldsymbol{z})d\boldsymbol{z}\right)} \\
&= \exp\left(\int_\mathcal{Z}(e^{h(\boldsymbol{z})}-1)e^{f(\boldsymbol{z})}\lambda(\boldsymbol{z})d\boldsymbol{z}\right) \\
&= \exp\left(\int_\mathcal{Z}(e^{h(\boldsymbol{z})}-1)\Lambda_Q(\boldsymbol{z})d\boldsymbol{z}\right).
\end{aligned}$$

We identify the last row as the generating functional of a Poisson process (37) with $\xi=1$. The intensity of the process is $\Lambda_Q(\boldsymbol{z}) = e^{f(\boldsymbol{z})}\lambda(\boldsymbol{z})$. With the fact that a Poisson process is uniquely characterised by its generating function (Kingman, 1993, chap. 3), the proof is complete.

Appendix E. Sparse Gaussian process approximation

To solve the inference problem for the function g, we define a sparse GP, using the same prior P, but by an effective likelihood which depends on a finite set of function values $\boldsymbol{g}_s = (g_1,\ldots,g_L)^\top$ only. Hence, we get

$$\frac{dQ_2^s}{dP}(g) = \boldsymbol{q}_2^s(\boldsymbol{g}_s) \tag{40}$$

and the sparse posterior measure is

$$dQ_2^s(g) = \boldsymbol{q}_2^s(\boldsymbol{g}_s)dP(g) = dP(g|\boldsymbol{g}_s)\times\boldsymbol{q}_2^s(\boldsymbol{g}_s)dP(\boldsymbol{g}_s),$$

where the last equality holds true, since Equation (40) only depends on $\boldsymbol{g}_s$. The KL–divergence between the full posterior density

$$\boldsymbol{q}_2(g) = \frac{dQ_2}{dP}(g) = \frac{e^{U(g)}}{\mathbb{E}_P\left[e^{U(g)}\right]}$$

and the sparse one $\boldsymbol{q}_2^s(\boldsymbol{g}_s)$ is given by

$$\begin{aligned}
\mathrm{D}_{\mathrm{KL}}(Q_2^s\|Q_2) &= \mathbb{E}_{Q_2^s}\left[\log\frac{\boldsymbol{q}_2^s(\boldsymbol{g}_s)}{\boldsymbol{q}_2(g)}\right] = \mathbb{E}_{P(\boldsymbol{g}_s)}\left[\boldsymbol{q}_2^s(\boldsymbol{g}_s)\mathbb{E}_{P(g|\boldsymbol{g}_s)}\left[\log\frac{\boldsymbol{q}_2^s(\boldsymbol{g}_s)}{e^{U(g)}}\right]\right] + \text{const.} \\
&= \mathbb{E}_{P(\boldsymbol{g}_s)}\left[\boldsymbol{q}_2^s(\boldsymbol{g}_s)\log\frac{\boldsymbol{q}_2^s(\boldsymbol{g}_s)}{e^{\mathbb{E}_{P(g|\boldsymbol{g}_s)}[U(g)]}}\right] + \text{const.}
\end{aligned}$$

From this we derive directly the posterior density for the sparse GP

$$\boldsymbol{q}_2^s(g) \propto e^{U^s(\boldsymbol{g}_s)},$$

with the sparse log–likelihood

$$U^s(\boldsymbol{g}_s) = \mathbb{E}_{P(g|\boldsymbol{g}_s)}[U(g)] = \int U(g)dP(g|\boldsymbol{g}_s).$$

Appendix F. Lower bound & hyperparameter optimization

The lower bound in Equation (12) is given by

$$\begin{aligned}
\mathcal{L}(\boldsymbol{q}) =& \mathbb{E}_Q\left[\log \frac{L(\mathcal{D}, \boldsymbol{\omega}_N, \Pi_{\hat{\mathcal{X}}}|g,\lambda)}{\boldsymbol{q}_1(\boldsymbol{\omega}_N)\boldsymbol{q}_1(\Pi_{\hat{\mathcal{X}}})\boldsymbol{q}_2^s(g)\boldsymbol{q}_2(\lambda)}\right] \\
=& \int_{\hat{\mathcal{X}}} \left(\mathbb{E}_Q\left[f(\omega,-g(\boldsymbol{x}))\right] - \mathbb{E}_Q\left[\log\Lambda_1\right] + \mathbb{E}_Q\left[\log\lambda\right] + 1\right)\Lambda_1(\boldsymbol{x},\omega)d\boldsymbol{x}d\omega \\
& - \int_{\hat{\mathcal{X}}} \Lambda_1(\boldsymbol{x},\omega)d\boldsymbol{x}d\omega \\
& + \sum_{n=1}^{N}\left(\mathbb{E}_Q\left[f(\omega_n,g_n)\right] + \mathbb{E}_Q\left[\log\lambda\right] - \cosh\left(\frac{c_1^{(n)}}{2}\right) + \frac{\left(c_1^{(n)}\right)^2}{2}\mathbb{E}_Q\left[\omega_n\right]\right) \\
& - \frac{1}{2}trace(K_s^{-1}(\Sigma_2^s + \boldsymbol{\mu}_2^s(\boldsymbol{\mu}_2^s)^\top)) - \frac{1}{2}\log\det(2\pi K_s) + \frac{1}{2}\log\det(2\pi e\Sigma_2^s) \\
& + \alpha_0\log\beta_0 - \log(\Gamma(\alpha_0)) + (\alpha_0-1)\mathbb{E}_Q\left[\log\lambda\right] - \beta_0\mathbb{E}_Q\left[\lambda\right] \\
& + \alpha_2 - \log\beta_2 + \log\Gamma(\alpha_2) + (1-\alpha_2)\psi(\alpha_2).
\end{aligned}$$

To optimise the covariance kernel parameters $\boldsymbol{\theta}$ we differentiate the lower bound with respect to these parameters and perform then gradient ascent. The gradient for one specific parameter θ is given by

$$\begin{aligned}
\frac{\partial\mathcal{L}(\boldsymbol{q})}{\partial\theta} =& \int_{\hat{\mathcal{X}}} \frac{\partial\mathbb{E}_Q\left[f(\omega,-g(\boldsymbol{x}))\right]}{\partial\theta}\Lambda_1(\boldsymbol{x},\omega)d\boldsymbol{x}d\omega + \sum_{n=1}^{N}\frac{\partial\mathbb{E}_Q\left[f(\omega_n,g(\boldsymbol{x}_n))\right]}{\partial\theta} \\
& - \frac{1}{2}\frac{trace(K_s^{-1}(\Sigma_2^s + \boldsymbol{\mu}_2^s(\boldsymbol{\mu}_2^s)^\top))}{\partial\theta} - \frac{1}{2}\frac{\partial\log\det(2\pi K_s)}{\partial\theta} \\
=& \int_{\hat{\mathcal{X}}} \frac{\partial\mathbb{E}_Q\left[f(\omega,-g(\boldsymbol{x}))\right]}{\partial\theta}\Lambda_1(\boldsymbol{x},\omega)d\boldsymbol{x}d\omega + \sum_{n=1}^{N}\frac{\partial\mathbb{E}_Q\left[f(\omega_n,g(\boldsymbol{x}_n))\right]}{\partial\theta} \\
& + \frac{1}{2}trace\left(K_s^{-1}\frac{\partial K_s}{\partial\theta}K_s^{-1}(\Sigma_2^s + \boldsymbol{\mu}_2^s(\boldsymbol{\mu}_2^s)^\top)\right) \\
& - \frac{1}{2}trace\left(K_s^{-1}\frac{\partial K_s}{\partial\theta}\right).
\end{aligned}$$

The derivatives of function $\mathbb{E}_Q\left[f(\omega,g(\boldsymbol{x}))\right]$ are

$$\frac{\partial\mathbb{E}_Q\left[f(\omega,g(\boldsymbol{x}))\right]}{\partial\theta} = \frac{1}{2}\left(\frac{\partial\mathbb{E}_Q\left[g(\boldsymbol{x})\right]}{\partial\theta} - \frac{\partial\mathbb{E}_Q\left[g(\boldsymbol{x})^2\right]}{\partial\theta}\mathbb{E}_Q\left[\omega\right]\right),$$

with

$$\begin{aligned}
\frac{\partial\mathbb{E}_Q\left[g(\boldsymbol{x})\right]}{\partial\theta} =& \frac{\partial\boldsymbol{\kappa}(\boldsymbol{x})}{\partial\theta}\boldsymbol{\mu}_2^s, \\
\frac{\partial\mathbb{E}_Q\left[g(\boldsymbol{x})^2\right]}{\partial\theta} =& \frac{\partial\tilde{k}(\boldsymbol{x},\boldsymbol{x})}{\partial\theta} + \frac{\partial\boldsymbol{\kappa}(\boldsymbol{x})}{\partial\theta}^\top\left(\Sigma_2^s + \boldsymbol{\mu}_2^s(\boldsymbol{\mu}_2^s)^\top\right)\boldsymbol{\kappa}(\boldsymbol{x}) + \boldsymbol{\kappa}(\boldsymbol{x})^\top\left(\Sigma_2^s + \boldsymbol{\mu}_2^s(\boldsymbol{\mu}_2^s)^\top\right)\frac{\partial\boldsymbol{\kappa}(\boldsymbol{x})}{\partial\theta},
\end{aligned}$$

where $\boldsymbol{\kappa}(\boldsymbol{x}) = \boldsymbol{k}_s(\boldsymbol{x})^\top K_s^{-1}$ and $\tilde{k}(\boldsymbol{x},\boldsymbol{x}) = k(\boldsymbol{x},\boldsymbol{x}) - \boldsymbol{k}_s(\boldsymbol{x})K_s^{-1}\boldsymbol{k}_s(\boldsymbol{x})^\top$. The remaining two terms are:

$$\begin{aligned}
\frac{\partial\tilde{k}(\boldsymbol{x},\boldsymbol{x})}{\partial\theta} =& \frac{\partial k(\boldsymbol{x},\boldsymbol{x})}{\partial\theta} - \frac{\partial\boldsymbol{\kappa}(\boldsymbol{x})}{\partial\theta}\boldsymbol{k}_s(\boldsymbol{x}) - \boldsymbol{\kappa}(\boldsymbol{x})\frac{\partial\boldsymbol{k}_s(\boldsymbol{x})}{\partial\theta}, \\
\frac{\partial\boldsymbol{\kappa}(\boldsymbol{x})}{\partial\theta} =& \frac{\partial\boldsymbol{k}_s(\boldsymbol{x})}{\partial\theta}^\top K_s^{-1} - \boldsymbol{k}_s(\boldsymbol{x})K_s^{-1}\frac{\partial K_s}{\partial\theta}K_s^{-1}.
\end{aligned}$$

After each variational step the hyperparameters are updated by

$$\boldsymbol{\theta}_{\text{new}} = \boldsymbol{\theta}_{\text{old}} + \varepsilon\frac{\partial\mathcal{L}(q)}{\partial\boldsymbol{\theta}},$$

where ε is the step size.

References

Ryan P. Adams, Iain Murray, and David J. C. MacKay. Tractable nonparametric bayesian inference in poisson processes with gaussian process intensities. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 9–16, 2009. doi: 10.1145/1553374.1553376.

Philipp Batz, Andreas Ruttor, and Manfred Opper. Approximate Bayes learning of stochastic differential equations. *Phys. Rev.*, E98(2):022109, 2018. doi: 10.1103/PhysRevE.98.022109.

Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg, 2006. ISBN 0387310738.

David R. Brillinger. Maximum likelihood analysis of spike trains of interacting nerve cells. *Biological Cybernetics*, 59(3):189–200, 1988. doi: 10.1007/BF00318010.

Anders Brix and Peter J. Diggle. Spatiotemporal prediction for log-gaussian cox processes. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63(4):823–841, 2001. doi: 10.1111/1467-9868.00315.

D. R. Cox. Some statistical methods connected with series of events. *Journal of the Royal Statistical Society. Series B (Methodological)*, 17(2):129–164, 1955. ISSN 00359246.

Lehel Csató. Gaussian processes-iterative sparse approximations. 2002. URL `http://publications.aston.ac.uk/1327/`.

Lehel Csató and Manfred Opper. Sparse on-line gaussian processes. *Neural Computation*, 14(3):641–668, 2002. doi: 10.1162/089976602317250933.

John P Cunningham, Byron M Yu, Krishna V Shenoy, and Maneesh Sahani. Inferring neural firing rates from spike trains using gaussian processes. In *Advances in Neural Information Processing Systems 20*, pages 329–336. 2008. URL `http://papers.nips.cc/paper/3229-inferring-neural-firing-rates-from-spike-trains-using-gaussian-processes.pdf`.

Alexander G. de G. Matthews, James Hensman, Richard Turner, and Zoubin Ghahramani. On sparse variational methods and the kullback-leibler divergence between stochastic processes. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 231–239, 2016. URL http://proceedings.mlr.press/v51/matthews16.html.

Christian Donner and Manfred Opper. Inverse ising problem in continuous time: A latent variable approach. *Phys. Rev. E*, 96:062104, 2017. doi: 10.1103/PhysRevE.96.062104.

Christian Donner and Manfred Opper. Efficient bayesian inference for a gaussian process density model. In *Proceedings of the 34th Conference on Uncertainty in Artificial Intelligence*, 2018. URL http://auai.org/uai2018/proceedings/papers/34.pdf.

Benjamin Dunn, Maria Mrreaunet, and Yasser Roudi. Correlations and functional connections in a population of grid cells. *PLOS Computational Biology*, 11(2):1–21, 2015. doi: 10.1371/journal.pcbi.1004052.

Paul Embrechts, Thomas Liniger, and Lu Lin. Multivariate hawkes processes: an application to financial data. *Journal of Applied Probability*, 48(A):367378, 2011. doi: 10.1239/jap/1318940477.

Seth Flaxman, Yee Whye Teh, and Dino Sejdinovic. Poisson intensity estimation with reproducing kernels. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 270–279. PMLR, 2017. URL http://proceedings.mlr.press/v54/flaxman17a.html.

Centre For The Biology Of Memory and Fransesca Sargolini. Grid cell data of sargolini et al 2006. 2014. doi: 10.11582/2014.00003.

Tom Gunter, Chris Lloyd, Michael A. Osborne, and Stephen J. Roberts. Efficient bayesian nonparametric modelling of structured point processes. In *Uncertainty in Artificial Intelligence (UAI)*, 2014. URL https://arxiv.org/abs/1407.6949.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58(1):83–90, 1971. ISSN 00063444.

James Hensman, Alexander G Matthews, Maurizio Filippone, and Zoubin Ghahramani. Mcmc for variationally sparse gaussian processes. In *Advances in Neural Information Processing Systems 28*, pages 1648–1656. 2015. URL http://papers.nips.cc/paper/5875-mcmc-for-variationally-sparse-gaussian-processes.pdf.

ST John and James Hensman. Large-scale Cox process inference using variational Fourier features. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pages 2362–2370, 2018. URL http://proceedings.mlr.press/v80/john18a.html.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *preprint arXiv*, abs/1412.6980, 2014. URL http://arxiv.org/abs/1412.6980.

John Frank Charles Kingman. *Poisson processes*. Oxford University Press, 1993. ISBN 9780198536932.

Alisa Kirichenko and Harry van Zanten. Optimality of poisson processes intensity learning with gaussian processes. *Journal of Machine Learning Research*, 16:2909–2919, 2015. URL http://jmlr.org/papers/v16/kirichenko15a.html.

J. Knollmüller, T. Steininger, and T. A. Enßlin. Inference of signals with unknown correlation structure from nonlinear measurements. *ArXiv e-prints*, 2017. URL https://arxiv.org/abs/1711.02955.

Takis Konstantopoulos, Zurab Zerakidze, and Grigol Sokhadze. *Radon–Nikodým Theorem*, pages 1161–1164. 2011. ISBN 978-3-642-04898-2.

P. A. W Lewis and G. S. Shedler. Simulation of nonhomogeneous poisson processes by thinning. *Naval Research Logistics Quarterly*, 26(3):403–413, 1979. doi: 10.1002/nav.3800260304.

Scott Linderman, Matthew Johnson, and Ryan P Adams. Dependent multinomial models made easy: Stick-breaking with the polya-gamma augmentation. In *Advances in Neural Information Processing Systems 28*, pages 3456–3464. 2015. URL http://papers.nips.cc/paper/5660-dependent-multinomial-models-made-easy-stick-breaking-with-the-polya-gamma-augmentation.

Scott Linderman, Matthew Johnson, Andrew Miller, Ryan Adams, David Blei, and Liam Paninski. Bayesian Learning and Inference in Recurrent Switching Linear Dynamical Systems. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54, pages 914–922, 2017. URL http://proceedings.mlr.press/v54/linderman17a.html.

Chris Lloyd, Tom Gunter, Michael Osborne, and Stephen Roberts. Variational inference for gaussian process modulated poisson processes. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1814–1822, 2015. URL http://proceedings.mlr.press/v37/lloyd15.html.

Chris Lloyd, Tom Gunter, Michael Osborne, Stephen Roberts, and Tom Nickson. Latent point process allocation. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51, pages 389–397, 2016. URL http://proceedings.mlr.press/v51/lloyd16.html.

Alexander G. de G. Matthews, Mark van der Wilk, Tom Nickson, Keisuke. Fujii, Alexis Boukouvalas, Pablo Leon-Villagra, Zoubin Ghahramani, and James Hensman. Gpflow: A gaussian process library using tensorflow. *Journal of Machine Learning Research*, 18 (40):1–6, 2017. URL http://jmlr.org/papers/v18/16-537.html.

Jesper Møller, Anne Randi Syversveen, and Rasmus Plenge Waagepetersen. Log gaussian cox processes. *Scandinavian Journal of Statistics*, 25(3):451–482, 1998. doi: 10.1111/1467-9469.00115.

Luis Moreira-Matias, Joao Gama, Michel Ferreira, Joao Mendes-Moreira, and Luis Damas. Predicting taxi–passenger demand using streaming data. *IEEE Transactions on Intelligent Transportation Systems*, 14(3):1393–1402, 2013.

Iain Murray, Zoubin Ghahramani, and David J. C. MacKay. Mcmc for doubly-intractable distributions. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 359–366, 2006. ISBN 0-9749039-2-2.

Iain Murray, David MacKay, and Ryan P Adams. The gaussian process density sampler. In *Advances in Neural Information Processing Systems 21*, pages 9–16. 2009. URL `http://papers.nips.cc/paper/3410-the-gaussian-process-density-sampler.pdf`.

Iain Murray, Ryan Adams, and David MacKay. Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9, pages 541–548, 2010. URL `http://proceedings.mlr.press/v9/murray10a.html`.

Yosihiko Ogata. Space-time point-process models for earthquake occurrences. *Annals of the Institute of Statistical Mathematics*, 50(2):379–402, 1998. doi: 10.1023/A:1003403601725.

Nicholas G. Polson, James G. Scott, and Jesse Windle. Bayesian inference for logistic models using plyagamma latent variables. *Journal of the American Statistical Association*, 108 (504):1339–1349, 2013. doi: 10.1080/01621459.2013.829001.

William H Press, Brian P Flannery, Saul A Teukolsky, William T Vetterling, et al. *Numerical recipes*, volume 3. Cambridge University Press, 2007. ISBN 978-0-521-88068-8.

Carl E. Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, Cambridge, MA, USA, 2006. ISBN 0-262-18253-X.

Yves-Laurent Kom Samo and Stephen Roberts. Scalable nonparametric bayesian inference on point processes with gaussian processes. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 2227–2236, 2015. URL `http://proceedings.mlr.press/v37/samo15.html`.

Francesca Sargolini, Marianne Fyhn, Torkel Hafting, Bruce L. McNaughton, Menno P. Witter, May-Britt Moser, and Edvard I. Moser. Conjunctive representation of position, direction, and velocity in entorhinal cortex. *Science*, 312(5774):758–762, 2006. doi: 10.1126/science.1125572.

Arno Solin. *Stochastic Differential Equation Methods for Spatio-Temporal Gaussian Process Regression*. Aalto University, 2016. ISBN 978-952-60-6711-7.

Dietrich Stoyan and Antti Penttinen. Recent applications of point process methods in forestry statistics. *Statistical Science*, 15(1):61–78, 2000. ISSN 08834237.

Yee W. Teh and Vinayak Rao. Gaussian process modulated renewal processes. In *Advances in Neural Information Processing Systems 24*, pages 2474–2482, 2011. URL `http://papers.nips.cc/paper/4358-gaussian-process-modulated-renewal-processes.pdf`.

Michalis Titsias. Variational learning of inducing variables in sparse gaussian processes. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5, pages 567–574, 2009. URL `http://proceedings.mlr.press/v5/titsias09a.html`.

Christian J. Walder and Adrian N. Bishop. Fast Bayesian intensity estimation for the permanental process. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3579–3588, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL `http://proceedings.mlr.press/v70/walder17a.html`.

Florian Wenzel, Théo Galy-Fajou, Christian Donner, Marius Kloft, and Manfred Opper. Scalable logit gaussian process classification. In *Advances in Approximate Bayesian Inference, NIPS Workshop*, 2017. URL `http://approximateinference.org/2017/accepted/WenzelEtAl2017.pdf`.

Journal of Machine Learning Research 19 (2018) 1-33 Submitted 1/18; Revised 9/18; Published 10/18

Multivariate Bayesian Structural Time Series Model

Jinwen Qiu JQIU@PSTAT.UCSB.EDU
S. Rao Jammalamadaka RAO@PSTAT.UCSB.EDU
Ning Ning NING@PSTAT.UCSB.EDU
Department of Statistics and Applied Probability
University of California
Santa Barbara, CA 93106, USA

Editor: Robert McCulloch

Abstract

This paper deals with inference and prediction for multiple correlated time series, where one also has the choice of using a candidate pool of contemporaneous predictors for each target series. Starting with a structural model for time series, we use Bayesian tools for model fitting, prediction and feature selection, thus extending some recent works along these lines for the univariate case. The Bayesian paradigm in this multivariate setting helps the model avoid overfitting, as well as captures correlations among multiple target time series with various state components. The model provides needed flexibility in selecting a different set of components and available predictors for each target series. The cyclical component in the model can handle large variations in the short term, which may be caused by external shocks. Extensive simulations were run to investigate properties such as estimation accuracy and performance in forecasting. This was followed by an empirical study with one-step-ahead prediction on the max log return of a portfolio of stocks that involve four leading financial institutions. Both the simulation studies and the extensive empirical study confirm that this multivariate model outperforms three other benchmark models, viz. a model that treats each target series as independent, the autoregressive integrated moving average model with regression (ARIMAX), and the multivariate ARIMAX (MARIMAX) model.

Keywords: Multivariate Time Series, Feature Selection, Bayesian Model Averaging, Cyclical Component, Estimation and Prediction

1. Introduction

The analysis of "Big Data" through the application of a new breed of analytical tools for manipulating and analyzing vast caches of data, is one of the cutting edge new areas. As a byproduct of the extensive use of the internet in collecting data on economic transactions, such data are growing exponentially every day. According to (Varian, 2014) and the references therein, Google has 30 trillion URLs and crawls over 20 billion of those each day. Conventional statistical and econometric techniques become increasingly inadequate to deal with such big data problems. For a good introduction to the new trends in data science, see (Blei and Smyth, 2017). Machine Learning as a field of computer science has strong ties to mathematical optimization and delivers methods, theory and applications, giving computers the ability to learn without being explicitly programmed (see a classical book, Mohri et al., 2012). Machine Learning indeed helps in developing high-performance computer tools, which often provide useful predictions in the presence of challenging computational needs. However, the result is one that we might call "pure prediction" and is not necessarily based on substantive knowledge. Also, typical assumptions such as the data being independent and identically (or at least independently) distributed, are not satisfactory when dealing with time stamped data, which is driven by multiple "predictors" or "features". We need to employ time series analysis for such series of data that are dependent, such as macroeconomic indicators of the national economy, enterprise operational management, market forecasting, weather and hydrology prediction.

Our focus here is on new techniques that work well for feature selection problems in time series applications. Scott and Varian (2014, 2015) introduced and further explored the Bayesian Structural Time Series (BSTS) model, a technique that can be used for feature selection, time series forecasting, nowcasting, inferring causal relationships (see Brodersen et al., 2015 and Peters et al., 2017), among others. One main ingredient of the BSTS model is that the time series aspect is handled through the Kalman filter (see Harvey, 1990; Durbin and Koopman, 2002; Petris et al., 2009) while taking into account the trend, seasonality, regression, and other common time series factors. The second aspect is the "spike and slab" variable selection, which was developed by George and McCulloch (1997) and Madigan and Raftery (1994), by which the most important regression predictors are selected at each step. The third aspect is the Bayesian model averaging (see Hoeting et al., 1999), which combines the feature selection results and prediction calculation. All these three parts have natural Bayesian interpretations and tend to play well together so that the resulting BSTS model discovers not only correlations but also causations in the underlying data. Some excellent related literature includes, but is not limited to the following: Dy and Brodley (2004); Cortes and Vapnik (1995); Guyon and Elisseeff (2003); Koo et al. (2007); Bach et al. (2013); Keerthi and Lin (2003); Nowozin and Lampert (2011); Krishnapuram et al. (2005); Caron et al. (2006); Csató and Opper (2002).

In this paper, we extend the BSTS model to the multivariate target time series with various components, and label it the Multivariate Bayesian Structural Time Series (MBSTS) model. For instance, the MBSTS model can be used to explicitly model the correlations between different stock returns in a portfolio through the covariance structure specified by Σ_t, see Equation (1). In this model, we allow a cyclical component with a shock damping parameter to specially model the influence of a shock to the time series, in addition to a standard local linear trend component, a seasonal component, and a regression component. One motivation for this is provided by the 2007–2008 financial crisis to the stock market. In examples with simulated data, the properties of our model such as estimation and prediction accuracy is investigated. As an illustration, through an empirical case study, we predict the max log returns over 5 consecutive business days of a stock portfolio with 4 stocks: Bank of America (BOA), Capital One Financial Corporation (COF), J.P. Morgan (JPM) and Wells Fargo (WFC), using domestic Google trends and 8 stock technical indicators as predictors.

Extensive analysis on both simulated data and real stock market data verifies that the MBSTS model gives much better prediction accuracy compared to the univariate BSTS model, the autoregressive integrated moving average with regression (ARIMAX) model, and the multivariate ARIMAX (MARIMAX) model. Some of the reasons for this can be seen in the following: the MBSTS model is strong in forecasting since it incorporates information of different components in the target time series, rather than merely historical values of the

same component; the Bayesian paradigm and the MCMC algorithm can perform variable selection at the same time during model training and thus prevent overfitting, even if some spurious predictors are added into the candidate pool; the MBSTS model benefits from taking correlations among multiple target time series into account, which helps boost the forecasting power and is a significant improvement over the univariate BSTS model.

The rest of the paper is organized as follows. In Section 2, we build the basic model framework. Extensive simulations are carried out in Section 3 to examine how the model performs under various conditions. In Section 4, an empirical study on the stock portfolio is done to show how well our model performs with real-world data. Section 5 concludes with some final remarks.

2. The MBSTS Model

In this section, we introduce the MBSTS model including model structure, state components, prior elicitation and posterior inference. Then we describe the algorithm for training the model and performing forecasts. In the sequel, the symbol " $\sim$ " and the superscript "(i)" will denote a column vector and the i-th component of a vector respectively, such as a $m \times 1$ vector $\tilde{y}_t = [y_t^{(1)}, \cdots, y_t^{(m)}]^T$.

2.1 Structural Time Series

Structural time series models belong to state space models for time series data given by the following set of equations:

$$\tilde{y}_t = Z_t^T \alpha_t + \tilde{\epsilon}_t, \qquad \tilde{\epsilon}_t \sim N_m(0, \Sigma_t), \tag{1}$$

$$\alpha_{t+1} = T_t \alpha_t + R_t \eta_t, \qquad \eta_t \sim N_q(0, Q_t), \tag{2}$$

$$\alpha_0 \sim N_d(\mu_0, \Sigma_0). \tag{3}$$

Equation (1) is called the observation equation, as it links the $m \times 1$ vector $\tilde{y}_t$ of observations at time t with a $d \times 1$ vector α_t denoting the unobserved latent states, where d is the total number of latent states for all entries in $\tilde{y}_t$. Equation (2) is called the transition equation because it defines how the latent states evolve over time. The model matrices Z_t, T_t, and R_t typically contain unknown parameters and known values which are often set as 0 and 1. Z_t is a $d \times m$ output matrix, T_t is a $d \times d$ transition matrix, and R_t is a $d \times q$ control matrix. The $m \times 1$ vector $\tilde{\epsilon}_t$ denotes observation errors with a $m \times m$ variance-covariance matrix Σ_t, and η_t is a q-dimensional system error with a $q \times q$ state diffusion matrix Q_t, where $q \leq d$. Note that any linear dependencies in the state vector can be moved from Q_t to R_t, hence Q_t can be set as a full rank variance matrix.

Structural time series models constructed in terms of components have a direct interpretation. For example, one may consider the classical decomposition in which a series can be seen as the sum of trend, season, cycle and regression components. In general, the model in state space form can be written as:

$$\tilde{y}_t = \tilde{\mu}_t + \tilde{\tau}_t + \tilde{\omega}_t + \tilde{\xi}_t + \tilde{\epsilon}_t, \qquad \tilde{\epsilon}_t \overset{iid}{\sim} N_m(0, \Sigma_\epsilon), \quad t = 1, 2 \ldots, n, \tag{4}$$

where $\tilde{y}_t$, $\tilde{\mu}_t$, $\tilde{\tau}_t$, $\tilde{\omega}_t$, $\tilde{\xi}_t$ and $\tilde{\epsilon}_t$ are m-dimension vectors, representing target time series, linear trend component, seasonal component, cyclical component, regression component and observation error terms respectively. Based on the state space form, α_t is the collection of these components, namely $\alpha_t = [\tilde{\mu}_t^T, \tilde{\tau}_t^T, \tilde{\omega}_t^T, \tilde{\xi}_t^T]^T$. Here Σ_ϵ is a $m \times m$ matrix, positive definite and is assumed to be constant over time for simplicity. Structural time series models allow us to examine the time series and flexibly select suitable components for trend, seasonality, and either static or dynamic regression. In the current model, all state components are assembled independently, with each component yielding an additive contribution to $\tilde{y}_t$. The flexibility of the model allows us to include different model components for each target series.

2.2 Components of State

The first component is a local linear trend. The specification of a time series model for the trend component varies according to the features displayed by the series under investigation and any prior knowledge. The most elementary structural model deals with a series whose underlying level changes over time. Moreover, it also sometimes displays a steady upward or downward movement, suggesting to incorporate a slope or a drift into the model for the trend. The resulting model, a generalization of the local linear trend model where the slope exhibits stationarity instead of obeying a random walk, is expressed in the form as:

$$\tilde{\mu}_{t+1} = \tilde{\mu}_t + \tilde{\delta}_t + \tilde{u}_t, \qquad \tilde{u}_t \overset{iid}{\sim} N_m(0, \Sigma_\mu), \tag{5}$$

$$\tilde{\delta}_{t+1} = \tilde{D} + \tilde{\rho}(\tilde{\delta}_t - \tilde{D}) + \tilde{v}_t, \qquad \tilde{v}_t \overset{iid}{\sim} N_m(0, \Sigma_\delta), \tag{6}$$

where $\tilde{\delta}_t$ and $\tilde{D}$ are m-dimension vectors. $\tilde{\delta}_t$ is the expected increase in $\tilde{\mu}_t$ between times t and $t+1$, so it can be thought as the slope at time t and $\tilde{D}$ is the long-term slope. The parameter $\tilde{\rho}$ is a $m \times m$ diagonal matrix, whose diagonal entries $0 \leq \rho_{ii} \leq 1$ for $i = 1, 2, \cdots, m$, represent the learning rates at which the local trend is updated for $\{y_t^{(i)}\}_{i=1,2,\cdots,m}$. Thus, the model balances short-term information with long-term information. When $\rho_{ii} = 1$, the corresponding slope becomes a random walk.

The second component is the one that captures seasonality. One frequently used model in the time domain is:

$$\tau_{t+1}^{(i)} = -\sum_{k=0}^{S_i - 2} \tau_{t-k}^{(i)} + w_t^{(i)}, \qquad \tilde{w}_t = [w_t^{(1)}, \cdots, w_t^{(m)}]^T \overset{iid}{\sim} N_m(0, \Sigma_\tau), \tag{7}$$

where S_i represents the number of seasons for $y^{(i)}$ and a m-dimension vector $\tilde{\tau}_t$ denotes their joint contribution to the observed target time series $\tilde{y}_t$. When we add a seasonal component, S_i seasonal effects are set in the state space form for $y^{(i)}$. However, only one seasonal effect has error term based on equation (7) and other effects are represented by itself in a deterministic equation. More specifically, the part of the transition matrix T_t representing the seasonal effects is an $(S_i - 1) \times (S_i - 1)$ matrix with -1 along the top row, 1 along the subdiagonal and 0 elsewhere. In addition, the expectation of the summation of S_i seasonal effects for $y^{(i)}$ is zero with variance equal to the i-th diagonal element of Σ_τ.

For each target series $y^{(i)}$, the model allows for various seasonal components with different periods as shown in equation (7). For instance, we might include a seasonal component with $S_i = 7$ to capture day-of-the-week effect for target series $y^{(i)}$, and $S_j = 30$ indicating day-of-the-month effect for another target series $y^{(j)}$ when modeling daily data. The corresponding seasonal transition matrix in state space setting is a 6×6 matrix and a 29×29 matrix with nonzero error variance for $y^{(i)}$ and $y^{(j)}$ respectively.

The third component is the one accounting for cyclical effects in the series. In economics, the term "business cycle" broadly refers to recurrent, not exactly periodic, deviations around the long-term path of the series. A model with a cyclical component is capable of reproducing commonly acknowledged essential features, such as the presence of strong autocorrelation, recurrence and alternation of phases, dampening of fluctuations, and zero long run persistence. A stochastic trend model of a seasonally adjusted economic time series does not capture the short-term movement of the series by itself. Including a serially correlated stationary component, the short-term movement could be captured, and this is the model incorporating cyclical effect (see Harvey et al., 2007). The cycle component is postulated as:

$$\begin{aligned} \tilde{\omega}_{t+1} &= \tilde{\varrho}\widehat{\cos(\lambda)}\tilde{\omega}_t + \tilde{\varrho}\widehat{\sin(\lambda)}\tilde{\omega}_t^\star + \tilde{\kappa}_t, \qquad \tilde{\kappa}_t \overset{iid}{\sim} N_m(0, \Sigma_\omega), \\ \tilde{\omega}_{t+1}^\star &= -\tilde{\varrho}\widehat{\sin(\lambda)}\tilde{\omega}_t + \tilde{\varrho}\widehat{\cos(\lambda)}\tilde{\omega}_t^\star + \tilde{\kappa}_t^\star, \qquad \tilde{\kappa}_t^\star \overset{iid}{\sim} N_m(0, \Sigma_\omega), \end{aligned} \tag{8}$$

where $\tilde{\varrho}$, $\widehat{\sin(\lambda)}$, $\widehat{\cos(\lambda)}$ are $m \times m$ diagonal matrices with diagonal entries equal to ϱ_{ii} (a damping factor for target series $y^{(i)}$ such that $0 < \varrho_{ii} < 1$), $\sin(\lambda_{ii})$ where $\lambda_{ii} = 2\pi/q_i$ is the frequency with q_i being a period such that $0 < \lambda_{ii} < \pi$, and $\cos(\lambda_{ii})$ respectively. When λ_{ii} is 0 or π, the model degenerates to the AR(1) process. The damping factor should be strictly less than one for stationary purpose. When the damping factor is bigger than one, there will be no restriction for the cyclical movement, resulting in extending the amplitude of the cycle.

These three time series components are illustrated in Figure 1. The big difference between the cyclical component and the seasonal component is the damping factor. The amplitude of the cyclical component will decay as time goes by, which can be applied to target time series affected by external shocks. Here Σ_μ, Σ_δ, Σ_τ and Σ_ω are $m \times m$ variance-covariance matrices for error terms of different time series components, and for simplicity we assume they are diagonal.

The fourth component is the regression component with static coefficients written as follows:

$$\xi_t^{(i)} = \beta_i^T x_t^{(i)}. \tag{9}$$

Here $\tilde{\xi}_t = [\xi_t^{(1)}, \cdots, \xi_t^{(m)}]^T$ is the collection of all elements in the regression component. For target series $y^{(i)}$, $x_t^{(i)} = [x_{t1}^{(i)}, \ldots, x_{tk_i}^{(i)}]^T$ is the pool of all available predictors at time t, and $\beta_i = [\beta_{i1}, \ldots, \beta_{ij}, \ldots, \beta_{ik_i}]^T$ represents corresponding static regression coefficients. All predictors are supposed to be contemporaneous with a known lag, which can be easily incorporated by shifting the corresponding predictors in time.

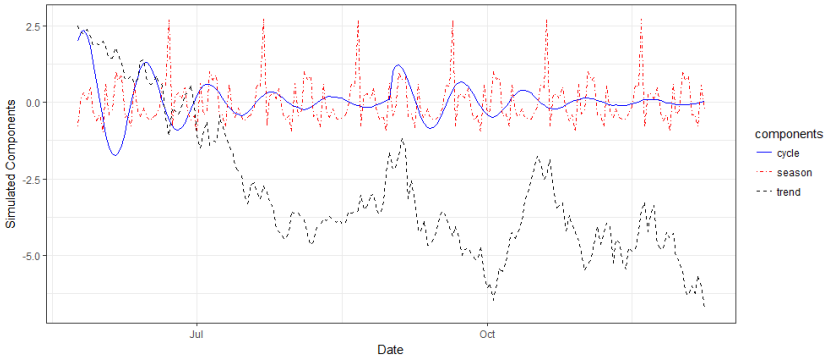


Figure 1: Simulated time series components include generalized linear trend, seasonality and cycle, generated by equations (5), (6), (7) and (8) with $\tilde{\rho} = [0.6]$, $\tilde{D} = [0]$, $\Sigma_\mu = [0.5^2]$, $\Sigma_\delta = [0.08^2]$, $S = 30$, $\Sigma_\tau = [0.01^2]$, $\lambda = \pi/10$, $\tilde{\varrho} = [0.97]$ and $\Sigma_\omega = [0.01^2]$, to show different contributions in explaining variations in target time series.

2.3 Spike and Slab Regression

In feature selection, a high degree of sparsity is expected, in the sense that coefficients of the vast majority of predictors are expected to be zero. A natural way to represent sparsity in the Bayesian paradigm is through the spike and slab coefficients. One advantage of working in a fully Bayesian setting is that we do not need to commit to a fixed set of predictors.

2.3.1 Matrix Representation

In order to assign appropriate prior distributions to parameters, we first combine $\tilde{y}_t$, $\tilde{\mu}_t$, $\tilde{\tau}_t$, $\tilde{\omega}_t$, $\tilde{\epsilon}_t$ into a $n \times m$ matrix as follows: $Y = [\tilde{y}_1, \ldots, \tilde{y}_t, \ldots, \tilde{y}_n]^T$, $M = [\tilde{\mu}_1, \ldots, \tilde{\mu}_t, \ldots, \tilde{\mu}_n]^T$, $T = [\tilde{\tau}_1, \ldots, \tilde{\tau}_t, \ldots, \tilde{\tau}_n]^T$, $W = [\tilde{\omega}_1, \ldots, \tilde{\omega}_t, \ldots, \tilde{\omega}_n]^T$ and $E = [\tilde{\epsilon}_1, \ldots, \tilde{\epsilon}_t, \ldots, \tilde{\epsilon}_n]^T$. Then the model can be written in a long matrix form as follows:

$$\tilde{Y} = \tilde{M} + \tilde{T} + \tilde{W} + X\beta + \tilde{E}, \tag{10}$$

where $\tilde{Y} = vec(Y)$, $\tilde{M} = vec(M)$, $\tilde{T} = vec(T)$, $\tilde{W} = vec(W)$, $\tilde{E} = vec(E)$, and X, β are written as:

$$X = \begin{bmatrix} X_1 & 0 & 0 & \ldots & 0 \\ 0 & X_2 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & X_m \end{bmatrix}, \qquad \beta = \begin{bmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix}, \tag{11}$$

where X_i being a $n \times k_i$ matrix, representing all observations of k_i candidate predictors for $y^{(i)}$, which is all observations of the i-th target series. The regression matrix X is of dimension $(nm \times K)$ with $K = \sum_{i=1}^{m} k_i$. Moreover, X_i and X_j can be the same or only contain a portion of common predictors. The regression coefficients for $y^{(i)}$ denoted as $\beta_i = [\beta_{i1}, \ldots, \beta_{ij}, \ldots, \beta_{ik_i}]^T$ is a k_i-dimension vector. Reformulating the model in this way facilitates the mathematical derivation in selecting a different set of available predictors at each iteration for $y^{(i)}$.

2.3.2 Prior distribution and elicitation

We define $\gamma_{ij} = 1$ if $\beta_{ij} \neq 0$, and $\gamma_{ij} = 0$ if $\beta_{ij} = 0$. Then $\gamma = [\gamma_1, \ldots, \gamma_m]$ where $\gamma_i = [\gamma_{i1}, \ldots, \gamma_{ik_i}]$. Denote β_γ as the subset of elements of β where $\beta_{ij} \neq 0$, and let X_γ be the subset of columns of X where $\gamma_{ij} = 1$. The spike prior is written as:

$$\gamma \sim \prod_{i=1}^{m} \prod_{j=1}^{k_i} \pi_{ij}^{\gamma_{ij}} (1 - \pi_{ij})^{1-\gamma_{ij}}, \qquad i = 1, \cdots, m, \tag{12}$$

where π_{ij} is the prior inclusion probability of the j-th predictor for the i-th target time series. Equation (12) is often further simplified by setting all the π_{ij} for $j = 1, 2, \cdots, k_i$ as the same value π_i for $y^{(i)}$ if prior information about effects of specific predictors on each target series are not available. With sufficient prior information available, assigning different subjectively determined values to π_{ij} might provide more robust results without a great amount of computational burden. An easy way to elicit π_i is to ask researchers for an "expected model size", so that if one expects q_i nonzero predictors for $y^{(i)}$, then $\pi_i = q_i/k_i$, where k_i is the total number of candidate predictors for the i-th target series. Under some circumstances, π_{ij} could be set as 0 or 1, for some specific predictors of $y^{(i)}$, forcing certain variables to be excluded or included. The spike prior can be specified by researchers in different distributional forms.

The natural conjugate prior for the multivariate model with the same set of predictors has the conjugate prior on β depending on Σ_ϵ. However, the multivariate extension with different set of predictors in each equation will destroy the conjugacy (Rossi et al. (2012)). Conjugate priors such as the normal distribution and the inverse Wishart distribution can still be used in a nonconjugate context, since models can be conjugate conditional on some other parameters. In order to obtain this conditional conjugate, we stack up the regression equations into one shown in equation (11). A simple slab prior specification is to make β and Σ_ϵ prior independent (see Griffiths, 2003):

$$\begin{aligned} p(\beta, \Sigma_\epsilon, \gamma) &= p(\beta|\gamma)p(\Sigma_\epsilon|\gamma)p(\gamma), \\ \beta|\gamma &\sim N_K(b_\gamma, A_\gamma^{-1}), \\ \Sigma_\epsilon|\gamma &\sim IW(v_0, V_0), \end{aligned} \tag{13}$$

where b_γ is the vector of prior means and $A_\gamma = \kappa X_\gamma^T X_\gamma / n$ is the full-model prior information matrix, with κ the number of observations worth of weight on the prior mean vector b_γ. If $X_\gamma^T X_\gamma$ is not positive definite due to perfect collinearity among predictors, $A_\gamma = \kappa(\omega X_\gamma^T X_\gamma + (1-\omega) diag(X_\gamma^T X_\gamma))/n$ can be used instead to guarantee propriety. Given analysts' specification, A_γ can be set in other forms. Here, $IW(v_0, V_0)$ is the inverse

Wishart distribution with v_0 the number of degrees of freedom and V_0 a $m \times m$ scale matrix. Although these priors are not conjugate, they are conditionally conjugate.

Equation (13) is the so-called "slab" because one can choose the prior parameters to make it only very weakly informative (close to flat), conditional on γ. The vector b_γ encodes our prior expectation about the value of each element of β_γ. In practice, one usually sets $b = 0$. The values of v_0 and V_0 can be set by asking analysts for an expected R^2 form the regression, and a number of observations worth of weight v_0, which must be greater than the dimension of $\tilde{y}_t$ plus one. Then $V_0 = (v_0 - m - 1) * (1 - R^2) * \Sigma_y$, where Σ_y is the variance-covariance matrix for multiple target time series Y.

Prior distributions of other variance-covariance matrices can be expressed as:

$$\Sigma_u \sim IW(w_u, W_u), \quad \text{for } u \in \{\mu, \delta, \tau, \omega\}. \tag{14}$$

By the assumption that all components are independent of each other, the prior distributions in multivariate form can reduced to their univariate counterparts since the matrices are diagonal. In other words, each diagonal entry of these matrices follows inverse gamma distributions as introduced in BSTS.

2.3.3 Posterior Inference

By the law of total probability, the full likelihood function is given by

$$p(\tilde{Y}^\star, \beta, \Sigma_\epsilon, \gamma) = p(\tilde{Y}^\star|\beta, \Sigma_\epsilon, \gamma) \times p(\beta|\gamma) \times p(\Sigma_\epsilon|\gamma) \times p(\gamma), \tag{15}$$

$$p(\tilde{Y}^\star|\beta, \Sigma_\epsilon, \gamma) \propto |\Sigma_\epsilon|^{-n/2} \exp\left(-\frac{1}{2}(\tilde{Y}^\star - X_\gamma\beta_\gamma)^T(\Sigma_\epsilon^{-1} \otimes I_n)(\tilde{Y}^\star - X_\gamma\beta_\gamma)\right), \tag{16}$$

$$p(\beta|\gamma) \propto |A_\gamma|^{1/2} \exp\left(-\frac{1}{2}(\beta_\gamma - b_\gamma)^T A_\gamma (\beta_\gamma - b_\gamma)\right), \tag{17}$$

$$p(\Sigma_\epsilon|\gamma) \propto |\Sigma_\epsilon|^{-(v_0+m+1)/2} \exp\left(tr(-\frac{1}{2}V_0\Sigma_\epsilon^{-1})\right), \tag{18}$$

where $\tilde{Y}^\star = \tilde{Y} - \tilde{M} - \tilde{T} - \tilde{W}$ is the multiple target time series $\tilde{Y}$ with time series components (trend, seasonality and cycle) subtracted out. Conditional on Σ_ϵ, one can introduce a normal prior, standardize the observations to remove correlation, and produce a posterior. However, we cannot find a convenient prior to integrate out Σ_ϵ from this conditional posterior. We tackle this issue by transforming equation $\tilde{Y}^\star = X\beta + \tilde{E}$ into a system with uncorrelated errors, using the square root of the variance-covariance matrix, $\Sigma_\epsilon = U^T U$. That is, if we multiply $((U^{-1})^T \otimes I_n)$ both sides of the equation, by the fact that $(U^{-1})^T \Sigma_\epsilon U^{-1} = I$, the transformed system has uncorrelated errors:

$$\begin{gathered} \hat{Y}^\star = \hat{X}\beta + \hat{E}, \quad \hat{Y}^\star = ((U^{-1})^T \otimes I_n)\tilde{Y}^\star, \quad \hat{X} = ((U^{-1})^T \otimes I_n)X, \\ \mathrm{Var}(\hat{E}) = \mathbb{E}((U^{-1})^T \otimes I_n)\tilde{E}\tilde{E}^T((U^{-1})^T \otimes I_n)] = I_m \otimes I_n. \end{gathered} \tag{19}$$

Then the full conditional distribution of $\beta|\hat{Y}^\star, \Sigma_\epsilon, \gamma$ can be expressed as:

$$p(\beta|\hat{Y}^\star, \Sigma_\epsilon, \gamma) \propto \exp\left(-\frac{1}{2}((\hat{Y}^\star - \hat{X}_\gamma\beta_\gamma)^T(\hat{Y}^\star - \hat{X}_\gamma\beta_\gamma) + (\beta_\gamma - b_\gamma)^T A_\gamma(\beta_\gamma - b_\gamma))\right). \tag{20}$$

Let us combine the two terms in exponential:

$$\begin{aligned}&(\hat{Y}^\star - \hat{X}_\gamma\beta_\gamma)^T(\hat{Y}^\star - \hat{X}_\gamma\beta_\gamma) + (\beta_\gamma - b_\gamma)^T A_\gamma(\beta_\gamma - b_\gamma)\\ =&\beta_\gamma^T(\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)\beta_\gamma - \beta_\gamma^T(\hat{X}_\gamma^T\hat{Y}^\star + A_\gamma b_\gamma) - (\hat{X}_\gamma^T\hat{Y}^\star + A_\gamma b_\gamma)^T\beta_\gamma + Const \qquad (21)\\ =&(\beta_\gamma - \tilde{\beta}_\gamma)^T(\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)(\beta_\gamma - \tilde{\beta}_\gamma) + Const,\end{aligned}$$

where $\tilde{\beta}_\gamma = (\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)^{-1}(\hat{X}_\gamma^T\hat{Y}^\star + A_\gamma b_\gamma)$. Then, a normal prior for β_γ is conjugate with the conditional likelihood for the transformed system:

$$\beta|\hat{Y}^\star, \Sigma_\epsilon, \gamma \sim N_K(\tilde{\beta}_\gamma, (\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)^{-1}). \qquad (22)$$

As A_γ gets smaller, the prior becomes flatter. The mean $\tilde{\beta}_\gamma$ can be recognized as the generalized least squares estimator.

The posterior of $\Sigma_\epsilon|\hat{Y}^\star, \beta, \gamma$ is in the inverted Wishart form. To see this, firstly recall that given β_γ we can observe or compute the errors $\tilde{E}$. Then the problem becomes a standard inference problem of a variance-covariance matrix using a multivariate normal sample. From equations (15), (16), (17) and (18), we know that

$$p(\Sigma_\epsilon|\tilde{Y}^\star, \beta, \gamma) \propto |\Sigma_\epsilon|^{-(n+v_0+m+1)/2}\exp\left(-\frac{1}{2}\{\tilde{E}_\gamma^T(\Sigma_\epsilon^{-1}\otimes I_n)\tilde{E}_\gamma + tr(V_0\Sigma_\epsilon^{-1})\}\right), \qquad (23)$$

where $\tilde{E}_\gamma = \tilde{Y}^\star - X_\gamma\beta_\gamma$. The terms in the exponential part can be expressed in a trace form:

$$\tilde{E}_\gamma^T(\Sigma_\epsilon^{-1}\otimes I_n)\tilde{E}_\gamma = vec(E_\gamma)^T(\Sigma_\epsilon^{-1}\otimes I_n)vec(E_\gamma) = tr(E_\gamma^T E_\gamma\Sigma_\epsilon^{-1}), \qquad (24)$$

where $E_\gamma = Y^\star - X_\gamma^\star B_\gamma$, $Y^\star = Y - M - T - W$, $X_\gamma^\star = [X_1, X_2, \ldots, X_M]_\gamma$ is a $(n \times K)$ matrix, and B_γ is a $(K \times m)$ matrix expressed as follows:

$$B_\gamma = \begin{bmatrix}\beta_1 & 0 & 0 & \ldots & 0\\ 0 & \beta_2 & 0 & \ldots & 0\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & 0 & \ldots & \beta_m\end{bmatrix}_\gamma. \qquad (25)$$

Then the full conditional distribution of Σ_ϵ is inverted Wishart as follows:

$$p(\Sigma_\epsilon|\tilde{Y}^\star, \beta, \gamma) \propto |\Sigma_\epsilon|^{-(n+v_0+m+1)/2}\exp\left(-\frac{1}{2}\{tr[(E_\gamma^T E_\gamma + V_0)\Sigma_\epsilon^{-1}]\}\right), \qquad (26)$$

$$\Sigma_\epsilon|\tilde{Y}^\star, \beta, \gamma \sim IW(v_0 + n, E_\gamma^T E_\gamma + V_0). \qquad (27)$$

Note that, if we let the prior precision goes to zero, the posterior on Σ_ϵ would center over the sum of squared residuals matrices.

Since there is no conjugacy in this prior setting, we can not get an analytic solution of the marginal distribution of γ. However, the conditional distribution of $\gamma|\Sigma_\epsilon, \tilde{Y}^\star$ can be

derived by the properties of conditional conjugacy. The joint probability density function $p(\Sigma_\epsilon, \tilde{Y}^\star, \gamma)$ can be obtained as follows:

$$\begin{aligned}p(\Sigma_\epsilon, \tilde{Y}^\star, \gamma) =& \int_{-\infty}^{+\infty} p(\beta, \Sigma_\epsilon, \tilde{Y}^\star, \gamma)d\beta\\ \propto&|\Sigma_\epsilon|^{-(v_0+m++n+1)/2}\exp\left(-\frac{1}{2}\{tr(V_0\Sigma_\epsilon^{-1}) + (\hat{Y}^\star)^T\hat{Y}^\star\}\right) \qquad (28)\\ &\times\frac{|A_\gamma|^{1/2}p(\gamma)}{|\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma|^{1/2}}\exp\left(-\frac{1}{2}\{b_\gamma^T A_\gamma b_\gamma - Z_\gamma^T(\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)^{-1}Z_\gamma\}\right),\end{aligned}$$

where $Z_\gamma = (\hat{X}_\gamma^T\hat{Y}^\star + A_\gamma b_\gamma)$. Then the conditional distribution of $\gamma|\Sigma_\epsilon, \tilde{Y}^\star$ can be expressed as:

$$p(\gamma|\Sigma_\epsilon, \tilde{Y}^\star) = C(\Sigma_\epsilon, \tilde{Y}^\star)\frac{|A_\gamma|^{1/2}p(\gamma)}{|\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma|^{1/2}}\exp\left(-\frac{1}{2}\{b_\gamma^T A_\gamma b_\gamma - Z_\gamma^T(\hat{X}_\gamma^T\hat{X}_\gamma + A_\gamma)^{-1}Z_\gamma\}\right), \qquad (29)$$

where $C(\Sigma_\epsilon, \tilde{Y}^\star)$ is a normalizing constant that only depends on Σ_ϵ and $\tilde{Y}^\star$. Note that, matrices needed to be computed here are of low dimension, in the sense that equation (29) places positive probabilities on coefficients being zero, leading to the sparsity of these matrices. In general, as a feature of the full posterior distribution, sparsity in this model enables equation (29) to be evaluated in an inexpensive way.

Next we need to derive conditional posterior of Σ_u for $u \in \{\mu, \delta, \tau, \omega\}$. Given the draws of states, parameters drawn are straightforward for all state components except the static regression coefficients. All time series components that solely depend on their variance parameters would translate their draws back to the error terms and accumulate sums of squares. For the reason that inverse Wishart distribution is the conjugate prior of a multivariate normal distribution with known mean and variance-covariance, the posterior distribution is still inverse Wishart distributed

$$\Sigma_u|u \sim IW(w_u + n, W_u + AA^T), \quad \text{for } u \in \{\mu, \delta, \tau, \omega\}, \qquad (30)$$

where $A = [\tilde{A}_1, \ldots, \tilde{A}_n]$ is a $m \times n$ matrix, representing a collection of residues of each time series component.

2.4 Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) methods are a class of algorithms to sample from a probability distribution based on constructing a Markov chain that has the desired distribution as its equilibrium distribution. The state of the chain after a number of steps is then used as a sample from the desired distribution. The quality of the sample improves as an increasing function of the number of steps.

2.4.1 Model Training

Let $\theta = (\Sigma_\mu, \Sigma_\delta, \Sigma_\tau, \Sigma_\omega)$ denotes the set of state component parameters. The posterior distribution of the model can be simulated by a Markov chain Monte Carlo algorithm given in Algorithm 1. Looping through the five steps yields a sequence of draws $\tilde{\psi} = (\alpha, \theta, \gamma, \Sigma_\epsilon, \beta)$

from a Markov chain with stationary distribution $p(\tilde{\psi}|Y)$, the posterior distribution of $\tilde{\psi}$ given Y.

Algorithm 1 MBSTS Model Training

1: Draw the latent state $\alpha = (\tilde{\mu}, \tilde{\delta}, \tilde{\tau}, \tilde{\omega})$ from given model parameters and $\tilde{Y}$, namely $p(\alpha|\tilde{Y}, \theta, \gamma, \Sigma_\epsilon, \beta)$, using the posterior simulation algorithm from Durbin and Koopman (2002).
2: Draw time series state component parameters θ given α, namely simulating $\theta \sim p(\theta|\tilde{Y}, \alpha)$ based on equation (30).
3: Loop over i in an random order, draw each $\gamma_i|\gamma_{-i}, \tilde{Y}, \alpha, \Sigma_\epsilon$, namely simulating $\gamma \sim p(\gamma|\tilde{Y}^\star, \Sigma_\epsilon)$ one by one based on equation (29), using the stochastic search variable selection (SSVS) algorithm from George and McCulloch (1997).
4: Draw β given Σ_ϵ, γ, α and $\tilde{Y}$, namely simulating $\beta \sim p(\beta|\Sigma_\epsilon, \gamma, \tilde{Y}^\star)$ based on equation (22).
5: Draw Σ_ϵ given γ, α, β and $\tilde{Y}$, namely simulating $\Sigma_\epsilon \sim p(\Sigma_\epsilon|\gamma, \tilde{Y}^\star, \beta)$ based on equation (27).

2.4.2 Target Series Forecasting

As typically in Bayesian data analysis, forecasts using our model are based on the posterior predictive distribution. Given draws of model parameters and latent states from their posterior distribution, we can draw samples from the posterior predictive distribution. Let $\hat{Y}$ represents the set of values to be forecast. The posterior predictive distribution of $\hat{Y}$ can be expressed as follows:

$$p(\hat{Y}|Y) = \int p(\hat{Y}|\tilde{\psi}) p(\tilde{\psi}|Y) d\tilde{\psi} \tag{31}$$

where $\tilde{\psi}$ is the set of all the model parameters and latent states randomly drawn from $p(\tilde{\psi}|Y)$. We can draw samples of $\hat{Y}$ from $p(\hat{Y}|\tilde{\psi})$ by simply iterating equations (5), (6), (7), (8) and (9) to move forward from initial values of states α with initial values of parameters θ, β and Σ_ϵ. In the one-step-ahead forecast, we draw samples from the multivariate normal distribution with mean equal to $\tilde{\mu}_n + \tilde{\delta}_n + \sum_{k=0}^{S-2} \tilde{\tau}_{n-k} + \tilde{\varrho}\widehat{cos(\lambda)}\tilde{\omega}_n + \tilde{\varrho}\widehat{sin(\lambda)}\tilde{\omega}_n^\star + \beta^{(k)} x_{n+1}$ and variance equal to $\Sigma_\epsilon + \Sigma_\mu + \Sigma_\tau + \Sigma_\omega$. Therefore, the samples drawn in this way have the same distribution as those simulated directly from the posterior predictive distribution.

Note that, the predictive probability density is not conditioned on parameter estimates, and inclusion or exclusion of predictors with static regression coefficients, all of which have been integrated out. Thus, through Bayesian model averaging, we commit neither to any particular set of covariates which helps avoid arbitrary selection, nor to point estimates of their coefficients which prevents overfitting. By the multivariate nature in our MBSTS model, the correlations among multiple target series are naturally taken into account, when sampling for prediction values of several target series. The posterior predictive density in equation (31), is defined as a joint distribution over all predicted target series, rather than as a collection of univariate distributions, which enables us to properly forecast multiple target series as a whole instead of predicting them individually. This is crucial, especially when

generating summary statistics, such as mean and variance-covariance from joint empirical distribution of forecast values.

3. Application to Simulated Data

In order to investigate the properties of our model, in this section, we analyze computer-generated data through a series of independent simulations. We generated multiple data sets with different time spans, local trends, number of regressors, dimensions of target series and correlations among several target series to analyze three aspects of generated data: accuracy in parameter estimation, ability to select the correct variables, and forecast performance of the model.

3.1 Generated Data

To check whether the estimation error and estimation standard deviation decrease as sample size increases, we built four different models in equation (32), each of which generates two target time series data with different numbers of observations (50, 100, 200, 400, 800, 1600, 3200). These data sets are simulated using latent states and a static regression component with four explanatory variables, one of which has no effect on each target series with zero coefficient. Specifically, each target series was generated with a different set of state components and explanatory variables, while the insignificant variable for each target series is not the same.

The latent states were generated using a local linear trend component with and without a global slope, a seasonality component with period equal to four, and/or a cyclical component with $\lambda = \pi/10$ for both target series. All initial values are drawn from normal distribution with a mean of zero. The detailed model description is presented as follows:

$$\begin{aligned}
\tilde{y}_t &= \tilde{\alpha}_t + B^T \tilde{x}_t + \tilde{\epsilon}_t \\
Model\ 1: \ \tilde{y}_t &= \tilde{\mu}_t + B^T \tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\alpha}_t = \tilde{\mu}_t \\
Model\ 2: \ \tilde{y}_t &= \tilde{\mu}'_t + B^T \tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\alpha}_t = \tilde{\mu}'_t \\
Model\ 3: \ \tilde{y}_t &= \tilde{\mu}'_t + \tilde{\tau}_t + B^T \tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\alpha}_t = \tilde{\mu}'_t + \tilde{\tau}_t \\
Model\ 4: \ \tilde{y}_t &= \tilde{\mu}'_t + \tilde{\tau}_t + \tilde{\omega}_t + B^T \tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\alpha}_t = \tilde{\mu}'_t + \tilde{\tau}_t + \tilde{\omega}_t
\end{aligned} \tag{32}$$

$$\begin{gathered}
\tilde{\epsilon}_t \overset{iid}{\sim} N_2(0, \Sigma_\epsilon) \qquad \Sigma_\epsilon = \begin{bmatrix} 1.1 & 0.7 \\ 0.7 & 0.9 \end{bmatrix} \\
B = \begin{bmatrix} 2 & -1 & -0.5 & 0 \\ -1.5 & 4 & 0 & 2.5 \end{bmatrix}^T \qquad \tilde{x}_t = [x_{t1}, x_{t2}, x_{t3}, x_{t4}]^T \\
x_{t1} \overset{iid}{\sim} N(5, 5^2) \quad x_{t2} \overset{iid}{\sim} Pois(10) \quad x_{t3} \overset{iid}{\sim} B(1, 0.5) \quad x_{t4} \overset{iid}{\sim} N(-2, 5^2)
\end{gathered} \tag{33}$$

$$\begin{gathered}
\tilde{\mu}_{t+1} = \begin{bmatrix} \mu_{1,t+1} \\ \mu_{2,t+1} \end{bmatrix} = \begin{bmatrix} \mu_{1,t} \\ \mu_{2,t} \end{bmatrix} + \begin{bmatrix} \delta_{1,t} \\ 0 \end{bmatrix} + \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} \\
\delta_{1,t} \overset{iid}{\sim} N(\delta_{1,t-1}, 0.08^2) \qquad \begin{bmatrix} u_{1,t} \\ u_{2,t} \end{bmatrix} \overset{iid}{\sim} N_2\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0.5^2 & 0 \\ 0 & 1 \end{bmatrix} \right)
\end{gathered} \tag{34}$$

$$\begin{aligned}\tilde{\mu}'_{t+1} &= \begin{bmatrix}\mu'_{1,t+1}\\ \mu'_{2,t+1}\end{bmatrix} = \begin{bmatrix}\mu'_{1,t}\\ \mu'_{2,t}\end{bmatrix} + \begin{bmatrix}\delta'_{1,t}\\ \delta'_{2,t}\end{bmatrix} + \begin{bmatrix}u_{1,t}\\ u_{2,t}\end{bmatrix}\\ \begin{bmatrix}\delta'_{1,t}\\ \delta'_{2,t}\end{bmatrix} &\overset{iid}{\sim} N_2\left(\begin{bmatrix}0.6\delta'_{1,t-1}+0.4*0.02\\ \delta'_{2,t-1}\end{bmatrix}, \begin{bmatrix}0.08^2 & 0\\ 0 & 0.16^2\end{bmatrix}\right)\end{aligned} \tag{35}$$

$$\tilde{\tau}_{t+1} = \begin{bmatrix}\tau_{1,t+1}\\ \tau_{2,t+1}\end{bmatrix} = \begin{bmatrix}-\sum_{k=0}^{2}\tau_{1,t-k}\\ 0\end{bmatrix} + \begin{bmatrix}w_{1,t}\\ 0\end{bmatrix} \qquad w_{1,t} \overset{iid}{\sim} N(0, 0.01^2) \tag{36}$$

$$\begin{aligned}\tilde{\omega}_{t+1} &= \begin{bmatrix}\omega_{1,t+1}\\ \omega_{2,t+1}\end{bmatrix} = \begin{bmatrix}0\\ 0.5*\cos(\lambda_{22})\omega_{2,t}\end{bmatrix} + \begin{bmatrix}0\\ 0.5*\sin(\lambda_{22})\omega^{\star}_{2,t}\end{bmatrix} + \begin{bmatrix}0\\ \kappa_{2,t}\end{bmatrix}\\ \tilde{\omega}^{\star}_{t+1} &= \begin{bmatrix}\omega^{\star}_{1,t+1}\\ \omega^{\star}_{2,t+1}\end{bmatrix} = \begin{bmatrix}0\\ -0.5*\sin(\lambda_{22})\omega_{2,t}\end{bmatrix} + \begin{bmatrix}0\\ 0.5*\cos(\lambda_{22})\omega^{\star}_{2,t}\end{bmatrix} + \begin{bmatrix}0\\ \kappa^{\star}_{2,t}\end{bmatrix}\\ \kappa_{2,t} &\overset{iid}{\sim} N(0, 0.01^2) \qquad \kappa^{\star}_{2,t} \overset{iid}{\sim} N(0, 0.01^2).\end{aligned} \tag{37}$$

To check the model performance with more than two series, two more data sets were generated by Model 5 and Model 6 according to equations (38) and (39), respectively, where for simplicity we consider latent states only include a generalized local linear trend with and without a global slope. The specific settings are given below:

$$\begin{aligned}&Model\ 5: \ \tilde{y}_t = \tilde{\mu}''_t + B^T\tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\epsilon}_t \overset{iid}{\sim} N_3(0, \Sigma_\epsilon)\\ &B = \begin{bmatrix}2 & -1 & -0.5 & 0\\ -1.5 & 4 & 0 & 2.5\\ 3 & 0 & 3.5 & -2\end{bmatrix}^T \qquad \Sigma_\epsilon = \begin{bmatrix}1.1 & 0.7 & 0.7\\ 0.7 & 0.9 & 0.7\\ 0.7 & 0.7 & 1.0\end{bmatrix}\\ &\tilde{\mu}''_{t+1} = \begin{bmatrix}\mu''_{1,t+1}\\ \mu''_{2,t+1}\\ \mu''_{3,t+1}\end{bmatrix} = \begin{bmatrix}\mu''_{1,t}\\ \mu''_{2,t}\\ \mu''_{3,t}\end{bmatrix} + \begin{bmatrix}\delta''_{1,t}\\ \delta''_{2,t}\\ \delta''_{3,t}\end{bmatrix} + \begin{bmatrix}u_{1,t}\\ u_{2,t}\\ u_{3,t}\end{bmatrix}\\ &\begin{bmatrix}\delta''_{1,t}\\ \delta''_{2,t}\\ \delta''_{3,t}\end{bmatrix} \overset{iid}{\sim} N_3\left(\begin{bmatrix}0.6\delta''_{1,t-1}+0.4*0.02\\ \delta''_{2,t-1}\\ 0.3\delta''_{3,t-1}+0.7*0.01\end{bmatrix}, \begin{bmatrix}0.08^2 & 0 & 0\\ 0 & 0.16^2 & 0\\ 0 & 0 & 0.12^2\end{bmatrix}\right)\\ &\begin{bmatrix}u_{1,t}\\ u_{2,t}\\ u_{3,t}\end{bmatrix} \overset{iid}{\sim} N_3\left(\begin{bmatrix}0\\ 0\\ 0\end{bmatrix}, \begin{bmatrix}0.5^2 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0.7^2\end{bmatrix}\right).\end{aligned} \tag{38}$$

$$\begin{aligned}&Model\ 6: \ \tilde{y}_t = \tilde{\mu}'''_t + B^T\tilde{x}_t + \tilde{\epsilon}_t \qquad \tilde{\epsilon}_t \overset{iid}{\sim} N_4(0, \Sigma_\epsilon)\\ &B = \begin{bmatrix}2 & -1 & -0.5 & 0\\ -1.5 & 4 & 0 & 2.5\\ 3 & 0 & 3.5 & -2\\ 0 & 1 & 1.5 & -0.5\end{bmatrix}^T \qquad \Sigma_\epsilon = \begin{bmatrix}1.1 & 0.7 & 0.7 & 0.7\\ 0.7 & 0.9 & 0.7 & 0.7\\ 0.7 & 0.7 & 1.0 & 0.7\\ 0.7 & 0.7 & 0.7 & 1.2\end{bmatrix}\\ &\tilde{\mu}'''_{t+1} = \begin{bmatrix}\mu'''_{1,t+1}\\ \mu'''_{2,t+1}\\ \mu'''_{3,t+1}\\ \mu'''_{4,t+1}\end{bmatrix} = \begin{bmatrix}\mu'''_{1,t}\\ \mu'''_{2,t}\\ \mu'''_{3,t}\\ \mu'''_{4,t}\end{bmatrix} + \begin{bmatrix}\delta'''_{1,t}\\ \delta'''_{2,t}\\ \delta'''_{3,t}\\ \delta'''_{4,t}\end{bmatrix} + \begin{bmatrix}u_{1,t}\\ u_{2,t}\\ u_{3,t}\\ u_{4,t}\end{bmatrix}\\ &\begin{bmatrix}\delta'''_{1,t}\\ \delta'''_{2,t}\\ \delta'''_{3,t}\\ \delta'''_{4,t}\end{bmatrix} \overset{iid}{\sim} N_4\left(\begin{bmatrix}0.6\delta'''_{1,t-1}+0.4*0.02\\ \delta'''_{2,t-1}\\ 0.3\delta'''_{3,t-1}+0.7*0.01\\ 0.5\delta'''_{4,t-1}\end{bmatrix}, \begin{bmatrix}0.08^2 & 0 & 0 & 0\\ 0 & 0.16^2 & 0 & 0\\ 0 & 0 & 0.12^2 & 0\\ 0 & 0 & 0 & 0.10^2\end{bmatrix}\right)\\ &\begin{bmatrix}u_{1,t}\\ u_{2,t}\\ u_{3,t}\\ u_{4,t}\end{bmatrix} \overset{iid}{\sim} N_4\left(\begin{bmatrix}0\\ 0\\ 0\\ 0\end{bmatrix}, \begin{bmatrix}0.5^2 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0.7^2 & 0\\ 0 & 0 & 0 & 0.6^2\end{bmatrix}\right).\end{aligned} \tag{39}$$

Model 7 was used to generate data to examine the accuracy in Bayesian point and interval estimations and covariates inclusion probabilities. The model is described as follows:

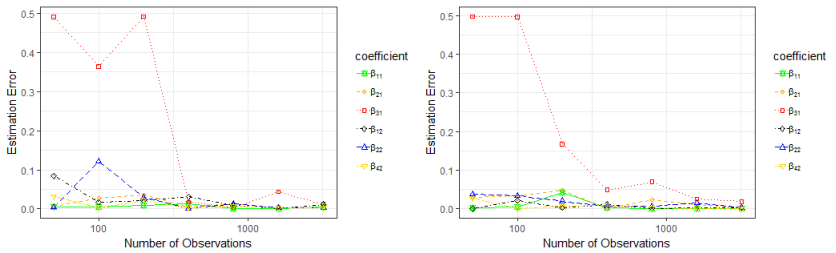
$$\begin{aligned}&Model\ 7: \ \tilde{y}_t = \tilde{\mu}'_t + \tilde{\tau}_t + \tilde{\omega}_t + diag(B^T\tilde{x}_t) + \tilde{\epsilon}_t \qquad \tilde{\epsilon}_t \overset{iid}{\sim} N_2(0, \Sigma_\epsilon)\\ &B = \begin{bmatrix}2 & -1 & -0.5 & 0 & 1.5 & -2 & 0 & 3.5\\ -1.5 & 4 & 0 & 2.5 & -1 & 0 & -3 & 0.5\end{bmatrix}^T\\ &\tilde{x}_t = \begin{bmatrix}x_{t1} & x_{t2} & x_{t3} & x_{t4} & x_{t5} & x_{t6} & x_{t7} & x^{\star}_{t8}\\ x_{t1} & x^{\star}_{t2} & x_{t3} & x_{t4} & x_{t5} & x_{t6} & x_{t7} & x_{t8}\end{bmatrix}^T\\ &x_{t5} \overset{iid}{\sim} N(-5, 5^2) \quad x_{t6} \overset{iid}{\sim} Pois(15) \quad x_{t7} \overset{iid}{\sim} Pois(20) \quad x_{t8} \overset{iid}{\sim} N(0, 10^2),\end{aligned} \tag{40}$$

where $x^{\star}_2$, $x^{\star}_5$ and $x^{\star}_8$ are variables whose values were obtained by rearranging a partial portion of data values for x_2, x_5 and x_8, and the $diag()$ operator extracts diagonal entries in the matrix to form a column vector. In Model 7, the first target series was generated by (x_1, x_2, x_3, x_4, x_5, x_6, x_7, $x^{\star}_8$) and the second target series was generated by (x_1, $x^{\star}_2$, x_3, x_4, x_5, x_6, x_7, x_8). Therefore, when explanatory variables (x_1, $x^{\star}_2$, x_3, x_4, $x^{\star}_5$, x_6, x_7, $x^{\star}_8$) are used for model training, regression coefficients of $x^{\star}_2$ (resp. $x^{\star}_5$) for the first target series generation are expected not to reflect the true linear relationship between $y^{(1)}$ and x_2 (resp. x_5). Similarly, regression coefficients of $x^{\star}_5$ (resp. $x^{\star}_8$) for the second target series generation are expected not to reflect the true linear relationship between $y^{(2)}$ and x_5 (resp. x_8). In sum, each distinct target series has a unique pattern generated by a particular set of explanatory variables and state components (the first target series affected by seasonality, not cyclical effect; the second target series affected by cyclical effect, not seasonality). Then we apply the MBSTS model on generated data sets to study its different properties.

3.2 Estimation and Model Selection Accuracy

From three perspectives, we explored properties of our model. More specifically, they include how the number of observations affects Bayesian estimation accuracy, how likely the 90% credible interval contains the true values of coefficients, and how possible the model selects the most important explanatory variables and ignores variables that do not contribute as desired, with results given in Figures 2, 3 and 4, respectively.

With the advent of the "big data" era, a huge amount of time series data are available to be analyzed from various sources. In the first analysis, we want to check whether a larger sample size improves the model performance in terms of Bayesian point estimation accuracy. After model training, we drew 2000 samples for each coefficient to be estimated during MCMC iterations. To reduce the influence of initial values on posterior inferences, we discarded an initial portion of the Markov chain samples. Specifically based on trial and error, the first 200 drawn samples were removed and the rest of them were used to build a sample posterior distribution for each parameter.

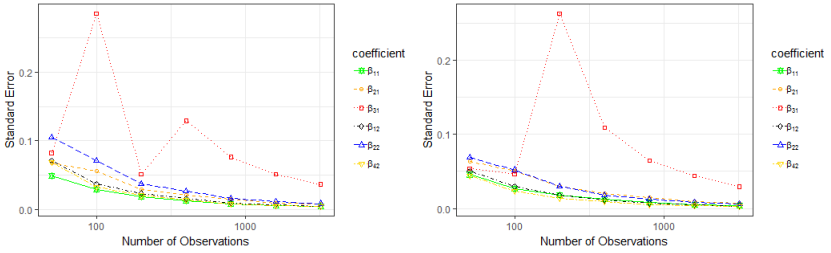


(a) Model 1 (b) Model 2

(c) Model 3 (d) Model 4

Figure 2: Estimation error for regression coefficients with different sample size. (a), (b), (c) and (d) display results using generated data sets by four different models in equation (32).

Based on the theory of Bayesian estimation, the sample mean from posterior distribution is considered to be the best point estimator for unknown parameters in terms of the mean squared error. We firstly consider the estimation error defined as the absolute value of difference between the true value and its Bayesian point estimate. The plots in Figure 2 illustrate how estimation errors of coefficients change as the sample size increases. The first target series was generated not using covariate x_4, while the second target series was generated not using covariate x_3, as shown in equation (33). Those zero coefficients are not displayed in these line plots. Figure 2 shows that only the estimation error for coefficient β_{31} goes down dramatically when sample size expands in these four cases. The remaining estimation errors stay almost the same regardless of different sample sizes, which implies that the number of observations significantly affect only the point estimation accuracy of coefficients for binary variables, not for numerical or ordinal variables. Even if only a small amount of data is available, our approach still performs well when binary or factor variables are not involved in the analysis.

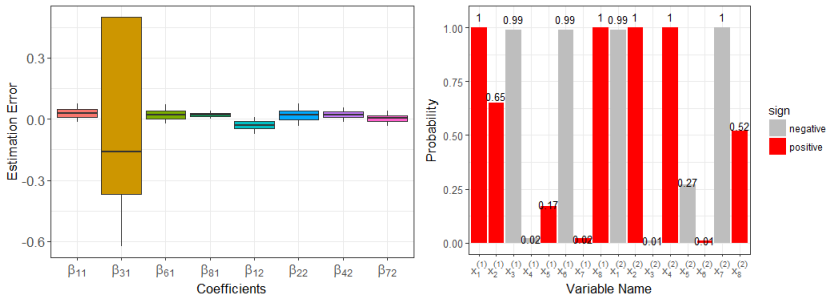


(a) Model 1 (b) Model 2

(c) Model 3 (d) Model 4

Figure 3: Standard error for regression coefficients with different sample size. Here, standard error is the empirical standard deviation of draws from equation (22). (a), (b), (c) and (d) display results using generated data sets by four different models in equation (32).

The sample standard error, defined as the posterior standard deviation of the regression coefficient, is used to illustrate the spread of the Bayesian estimator. To further explore other properties of the posterior distribution of draws, the standard errors were checked for each coefficient with different sample sizes. Figure 3 shows that all standard errors for covariates' coefficients except β_{31} gradually decline with a larger amount of data. The standard error for coefficients β_{31} peaks when the number of observations is 100 for Model 1 or 200 for models 2, 3 and 4, and then begin to drop very sharply. In general, a larger sample size helps shrink the standard errors of all coefficients, especially for binary or factor covariates' coefficients, as one would expect. In other words, collecting more data allows us to shrink the dispersion of the posterior empirical distribution from Monte Carlo draws, and hence build a narrower credible interval.



(a) Boxplot with 90% Credible Interval (b) Inclusion Probability

Figure 4: Empirical posterior distribution of estimated coefficients and indicators. (a) Box plots of the difference between draws from equation (22) and true values of regression coefficients. The top and bottom correspond to the 95% upper bound and 5% low bound, respectively. (b) Bar plot of empirical inclusion probability illustrates the proportion of Monte Carlo draws with $\gamma_{ij} = 1$. The red color shows positive estimated values of regression coefficients, while gray color displays negative values.

In the second analysis, we assess the coverage properties of the posterior credible intervals based on the empirical posterior distribution of each covariate's coefficients. In other words, the 90% credible interval contains the ground truth in 90% of the simulations. In Model 7 equation (40), $x_8^\star$ instead of x_8 was used to generate the first target series $y^{(1)}$, and $x_2^\star$ instead of x_2 was used to generate the second target series $y^{(2)}$. Therefore, when the explanatory variables (x_1, $x_2^\star$, x_3, x_4, $x_5^\star$, x_6, x_7, $x_8^\star$) are used for model training, the resulting coefficients β_{21} and β_{51} for $y^{(1)}$ as well as β_{52} and β_{82} for $y^{(2)}$, cannot reveal a true linear relationship. The box plot in Figure 4 displays the empirical posterior distribution of estimated coefficients for significant explanatory variables whose values were not randomly shuffled, and indicates that the true values of all coefficients are within 90% credible intervals. In addition, we can see that the 90% credible interval of binary covariates' coefficients is much wider than others, due to their larger standard errors.

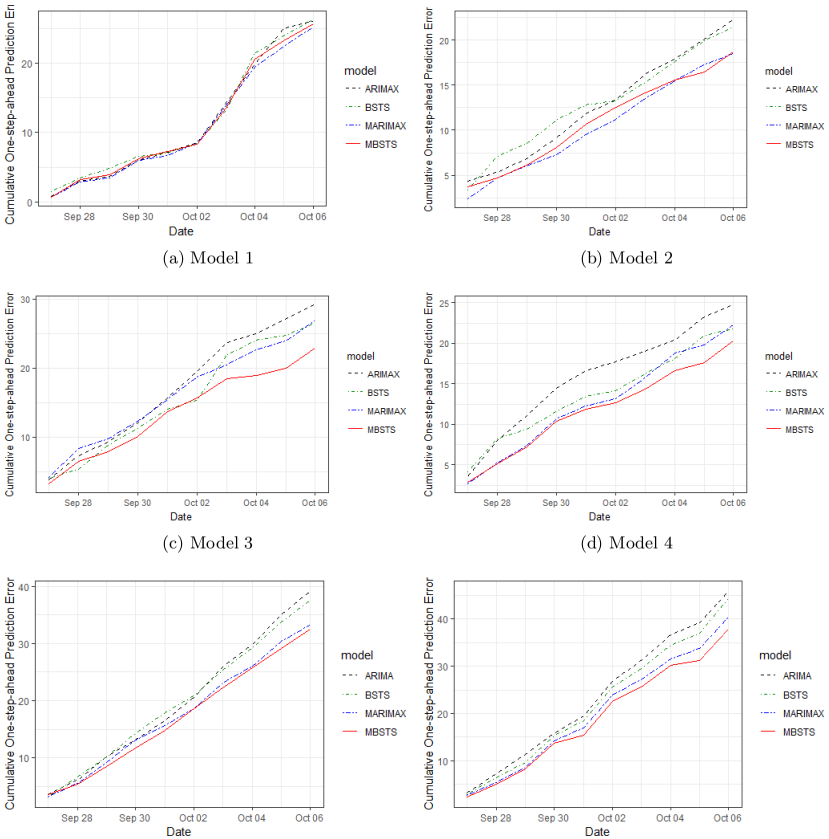
In the third analysis, one important property of our model is to reduce data dimension by variable selection in model training. In Figure 4, the bar plot of empirical inclusion probabilities based on the proportion of MCMC draws shows a clear picture of which variables are used to generate data and which are the ones with shuffled values. For the first target series $y^{(1)}$, the empirical inclusion probabilities of covariates x_1, x_3, x_6 and x_8 as one or close to one indicate that they were all, or almost all, selected during MCMC iterations, which is exactly how the data set was generated; the covariates x_4 and x_7 with zero coefficients indicate that they are rarely selected during MCMC iterations. Some covariates with partially shuffled values, such as x_2 and x_5, are more likely be selected than those with no effect on this target series, but they are not so important as x_1, x_3, x_6 and x_8. Similar striking results were achieved for the second target series. Moreover, we can see that as expected, the inclusion probability of $x_5^\star$ is just 0.17 (resp. 0.27) for the first (resp. second) target series. In a word, our MBSTS model is good at variables selection, even if the variation of each target series is explained by a different set of explanatory variables.

It is worth emphasizing that our model performs very well in terms of estimation accuracy and variables selection ability, even if each target series has a different set of latent states and explanatory variables from others. However, all preceding results depend on the assumption that the model structure remains intact throughout the modeling period. In other words, even though the model is built on the idea of multiple non-stationary components such as a time-varying local trend, seasonal effect, and potentially dynamic regression coefficients, the structure itself remains unchanged. If the model structure does change over time (e.g. local trend disappears or the static regression coefficients become dynamic), the estimation accuracy may suffer. Therefore, a preliminary data exploration and acquiring a background knowledge about the data set before applying our model is suggested, although it has the strength in allowing users to adjust the model components flexibly for each target series.

3.3 Model Performance Comparison

The generated data sets were split into a certain period of training data and a subsequent period of testing set. The standard approach would use the training data to develop the model that would then be applied to obtain predictions for the testing period. We use a growing window approach, which simply adds one new observation in the test set to the existing training set, obtaining a new model with fresher data and then constantly forecasting a new value in the test set.

To evaluate the performance of the MBSTS model, we use three other models: autoregressive integrated moving average model with regression (ARIMAX), multivariate ARIMAX (MARIMAX), and the BSTS model, as benchmark models. We replace ARIMAX and MARIMAX with seasonal ARIMAX (SARIMAX) and multivariate seasonal ARIMAX (MSARIMAX), when seasonality exists. In this study, applying the growing window approach, all models were trained by the training set and then were used to make a one-step-ahead prediction. More specifically, the univariate BSTS and ARIMAX were trained for each target time series individually, but MBSTS and MARIMAX were applied on the mul-



(a) Model 1 (b) Model 2

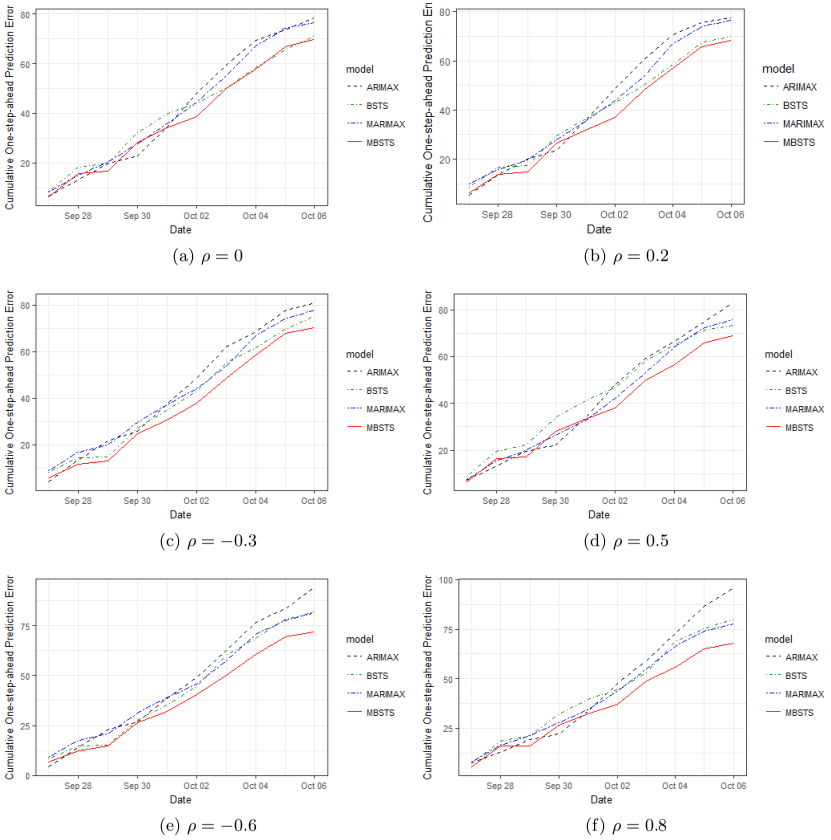
(c) Model 3 (d) Model 4

(e) Model 5 (f) Model 6

Figure 5: Cumulative absolute one-step-ahead prediction error for generated multiple target series containing different components. (a)-(f) display results using generated data sets by six different models in equations (32), (38) and (39). Three other benchmark models (BSTS, ARIMAX and MARIMAX) are also trained and used to make a prediction.

tidimensional series data set as a whole. Then we compared the performances of the other three models with that of MBSTS in terms of cumulative one-step ahead prediction errors. The prediction error at each step PE_t is defined by summing up the absolute values of the

differences between the true values and their own predicted values over all target time series, i.e. $\sum_{i=1}^{m} |y_t^{(i)} - \hat{y}_t^{(i)}|$. Figure 5 and Figure 6 are generated to demonstrate our model's comparison performance under the influence of complexity in different kinds and numbers of multiple target time series, and under various correlations ($\rho = 0, 0.2, -0.3, 0.5, -0.6, 0.8$), respectively.



(a) $\rho = 0$ (b) $\rho = 0.2$

(c) $\rho = -0.3$ (d) $\rho = 0.5$

(e) $\rho = -0.6$ (f) $\rho = 0.8$

Figure 6: Cumulative absolute one-step-ahead prediction error for generated multiple target series with different correlations. (a)-(f) display results using generated data sets by equation (40) with various correlation coefficients in Σ_t. Other three benchmark models (BSTS, ARIMAX and MARIMAX) are also trained and used to make a prediction.

Figure 5 shows cumulative one-step-ahead prediction errors of six time series models, which were trained using a set of data sets with each containing one thousand observations generated by equations (32), (38) and (39). We can see that the MBSTS model does not show an obvious advantage in the first two plots, since the generated target time series have only a local trend or a linear trend. However, the MBSTS model beats other benchmark models in plot 3 and plot 4, where the target series contain seasonality or cycle components. Clearly, the BSTS or MBSTS model has a strong ability to capture seasonality and cycle embodied in the series. The performance evaluations in plot 5 (three target time series) and plot 6 (four target time series) demonstrate the forecast advantage of our MBSTS model over other benchmark models, even with an increased number of target series. In general, the multivariate models outperform their corresponding univariate ones due to the influence of correlations among multiple target time series. Moreover, BSTS is better than ARIMAX, and MBSTS outperforms all other models, thanks to the Bayesian model averaging and time series structure of target series.

Figure 6 provides a clear picture of an impressive fact: the higher correlation among multiple target time series, the better performance of the MBSTS model over other models. Generally, the MBSTS model outperforms the traditional ARIMAX or MARIMAX model for the reason that averaging algorithm helps hedge against selecting the "wrong" set of predictors in prediction steps. The gaps of cumulative prediction errors between models in a multivariate version and their univariate counterparts increase as multiple target time series have stronger correlations. Therefore, it is better to model multiple target time series as a whole by MBSTS rather than model them individually by BSTS, especially when strong correlations appear in the multiple target time series, as illustrated in Figure 6

4. Application to Empirical Data

Predicting stock prices (for example, of a group of leading companies) is extremely important to Wall Street practitioners for investment and/or risk management purposes. In the following, we forecast the future values of stock portfolio return using the proposed MBSTS model and compare its performance with three other benchmark models: BSTS, ARIMAX and MARIMAX. In this section, we analyze the data of Bank of America (BOA), Capital One Financial Corporation (COF), J.P. Morgan (JPM) and Wells Fargo (WFC). The daily data sample is from 11/27/2006 to 11/03/2017 and obtained from Google Finance.

4.1 Target Time Series

We perceive the stock as worthwhile in terms of trading when its future price is predicted to vary more than $p\%$ of its current price. In this context, we forecast the trend of stock movements in the next $k(=5)$ transaction days, which is especially helpful when liquidation risk is in consideration given a sign of sale, and useful to avoid a large amount purchase driving up the stock prices given a sign of buying. In this study, we provide daily predictions sequentially of the overall price dynamics in the next k transaction days.

Following Torgo (2011), we approximate the daily average price as: $\bar{P}_t = (C_t+H_t+L_t)/3$, where C_t, H_t and L_t are the close, high and low quotes for day t respectively. However, instead of using the arithmetic returns, we are interested in the log return V_t defined as $V_t = \{\log(\bar{P}_{t+j}/C_t)\}_{j=1}^{k}$. We consider the indicator variable $y_t = \max\{v \in V_t\}$, the maximum

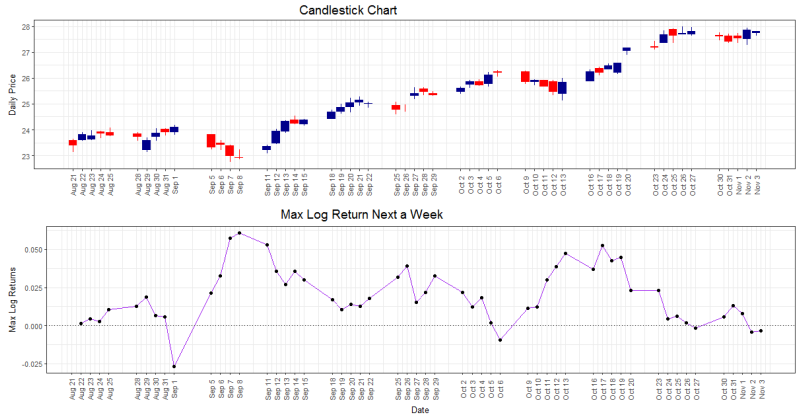


Figure 7: The candlestick chart and max log return. The top panel displays a candlestick chart of BOA from Aug 21st to Nov 3rd, containing information such as open and closing quotes. The bottom panel shows corresponding max log returns over the next five transaction days, which is the target time series.

value of log returns over the next k transaction days. A high positive value of y_t means that there is at least one future daily price that is much higher than today's close quote, indicating potential opportunities to issue a buy order, as we predict the prices will rise. A trivial value of y_t around zero can be seen as the sign of no action that should be taken at this moment. In this study, we calculated y_t for four leading companies in the financial industry (BOA, COF, JPM and WFC), whose stock prices are affected by economic activities. Visualization of a part of the daily prices time series and their corresponding y_t indicators for BOA can be seen in Figure 7.

4.2 Predictors

To better capture market information and different properties of the stock price time series and to facilitate the forecasting task, we use the following fundamental and technical predictors.

Fundamental Part Fundamental analysis claims that markets may incorrectly price a security in the short run but will eventually correct it. Profits can be achieved by purchasing the undervalued security and then waiting for the market to recognize its "mistake" and bounce back to the fundamental value. Since macroeconomy has a significant effect on the financial market, economical analysis plays an important role in fundamental analysis in giving a precise stock return prediction.

For economic analysis, we know that it is difficult to collect important economic indicators on a daily basis. However, starting from the year 2004, Google has been collecting the daily volume of searches related to various aspects of macroeconomics. This database is publicly available as "Google Domestic Trends". In a recent study, Preis et al. (2013) showed correlations between Google domestic trends and the equity market. In this study, we use the Google domestic trends data as a representation of the public interest in various macroeconomic factors, and include 27 domestic trends which are listed in Table 1 with their abbreviations.

Trend	Abbr.	Trend	Abbr.
Advertising & marketing	advert	Air travel	airtvl
Auto buyers	auto	Auto financing	autoby
Automotive	autofi	Business & industrial	bizind
Bankruptcy	bnkrpt	Commercial Lending	comlnd
Computers & electronics	comput	Construction	constr
Credit cards	crcard	Durable goods	durble
Education	educat	Finance & investing	invest
Financial planning	finpln	Furniture	furntr
Insurance	insur	Jobs	jobs
Luxury goods	luxury	Mobile & wireless	mobile
Mortgage	mtge	Real estate	rlest
Rental	rental	Shopping	shop
Small business	smallbiz	Travel	travel
Unemployment	unempl		

Table 1: Google domestic trends

Technical Part Technical analysis claims that useful information is already reflected in stock prices. We selected a representative set of technical indicators to capture the volatility, close location value, potential reversal, momentum and trend of each stock. Eight variables are calculated for each company as listed in Table 2:

Variable	Abbr.
Chaikin volatility	ChaVol
Yang and Zhang Volatility historical estimator	Vol
Arms' Ease of Movement Value	EMV
Moving Average Convergence/Divergence	MACD
Money Flow Index	MFI
Aroon Indicator	AROON
Parabolic Stop-and-Reverse	SAR
Close Location Value	CLV

Table 2: Stock Technical Predictors

- The ChaVol indicator depicts volatility by calculating the difference between the high and low for each period or trading bar, and measures the difference between two moving averages of a volume weighted accumulation distribution line.

- The Vol indicator has the minimum estimation error, and is independent of drift and opening gaps, which can be interpreted as a weighted average of the Rogers and Satchell estimator, the close-open volatility, and the open-close volatility.

- The EMV indicator is a momentum indicator developed by Richard W. Arms, Jr., which takes into account both volume and price changes to quantify the ease (or difficulty) of price movements.

- The MACD indicator is a trading indicator used in stock prices' technical analysis, created by Gerald Appel in the late 1970s, supposed to reveal changes in the strength, direction, momentum and duration of a trend in a stock's price.

- The MFI indicator is a ratio of positive and negative money flow over time and starts with the typical price for each period. It is an oscillator that uses both price and volume to measure buying and selling pressure, created by Gene Quong and Avrum Soudack.

- The AROON indicator is a technical indicator used to identify trends in an underlying security and the likelihood that the trends will reverse, including "Aroon up" (resp. "Aroon down") for measurement of the strength of the uptrend (resp. downtrend), and reports the time it takes for the price to reach the highest and lowest points over a given time period.

- The SAR indicator is a method proposed by J. Welles Wilder, Jr., to find potential reversals in the market price direction of traded goods such as securities.

- The CLV indicator is used to measure the closes quote relative to the day's high and low, which varies in range between -1 and $+1$.

4.3 Training Result

It is worth noting that all predictors do not show obvious trends and most of them are stationary in the sense that their unit-root null hypotheses have p-values less than 0.05 in the augmented Dickey-Fuller test (see Said and Dickey, 1984). However, some of them indicate seasonal patterns. We can remove seasonal patterns of these predictors by subtracting the estimated seasonal component computed by the STL procedure (see Cleveland et al., 1990). Then we test our MBSTS model with and without deseasonalizing the predictors.

These eight technical predictors are calculated for each financial institution and then exclusive to others. Domestic Google trends serve as common predictors available to all companies. Based on the forecast output, the model trained without deseasonal predictors performs better than the corresponding one with deseasonal predictors. Therefore, the training results shown in Figure 8 are from a model with original predictors.

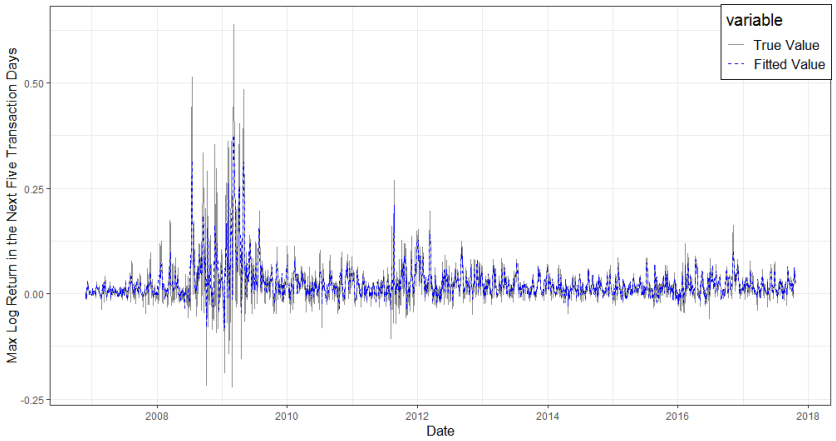
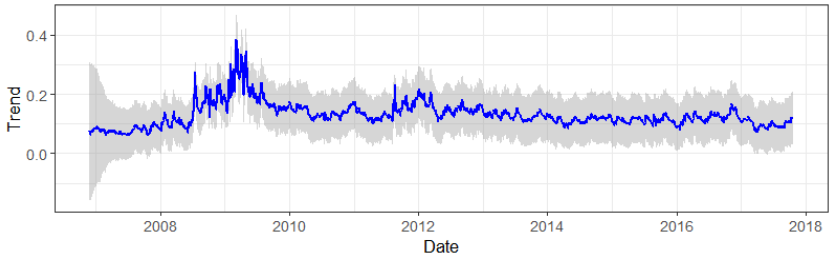


Figure 8: True and fitted values of max log returns from 11/27/2006 to 10/20/2017 (BOA)

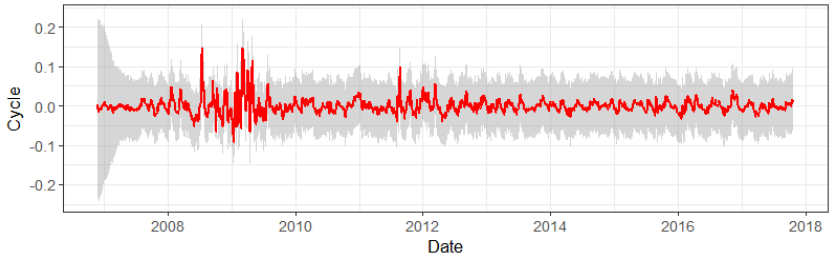
4.3.1 Decomposition of State Components

The business cycle describes the fluctuations in economic activities that an economy experiences over a period of time. It typically involves shifts over time between periods of expansions and recessions, which has a great impact on institutions in financial industry, especially investment and commercial banks. We use BOA as an example to illustrate target series and its corresponding state components. Visually checking the time series of max log returns over the next five transaction days in Figure 8, we see strong fluctuations during 2008-2009, which is right after the outbreak of the subprime mortgage crisis. There is also an obvious subsequent strong variation during 2012. Therefore, in order to capture recurrent economic shocks, it is necessary to incorporate the cyclical component in our model. In fact, applying the trend-cycle model can capture both short-term and long-term movements of the series.

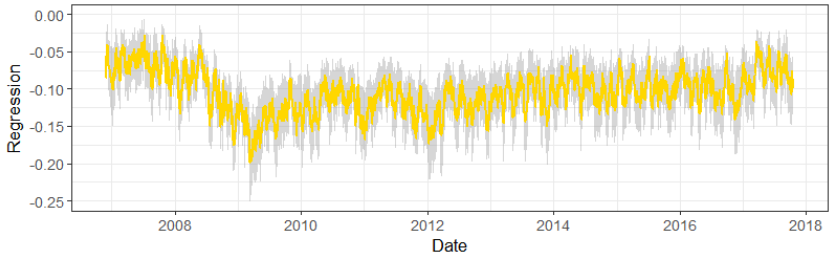
By spectral analysis, we find the corresponding period equals 274, which is almost one year of transaction days. Through cross validation, we find the optimal damping factor equals 0.95 in terms of cumulative one-step prediction errors. Figure 9 shows how much variation in the max log return time series is explained by the trend, cyclical and regression components. The trend component shows the highest peak is around 2009, and provides a general picture of how the series would evolve in the long run. The comparatively stronger variation between 2009 and 2012 is reflected in the cyclical component, which captures the economic shocks that occurred. The fluctuations gradually become stable as the effects of shocks diminish. Both trend and cyclical components handle the series with unequal variances over time. On the contrary, the regression component varies more frequently but with no obvious peaks. It accounts for local movements without the impact of external



(a) Trend Component



(b) Cyclical Component



(c) Regression Component

Figure 9: Contributions of state components to max log return (BOA). The fitted target series is decomposed into three state components: (a) trend (local level in this example) component, (b) cycle component and (c) regression component, with shaded areas indicating the 90% confidence bands based on MCMC draws.

shocks. In sum, decomposing the target time series into three components provides us enough information on how each component contributes in explaining variations.

4.3.2 Feature Selection

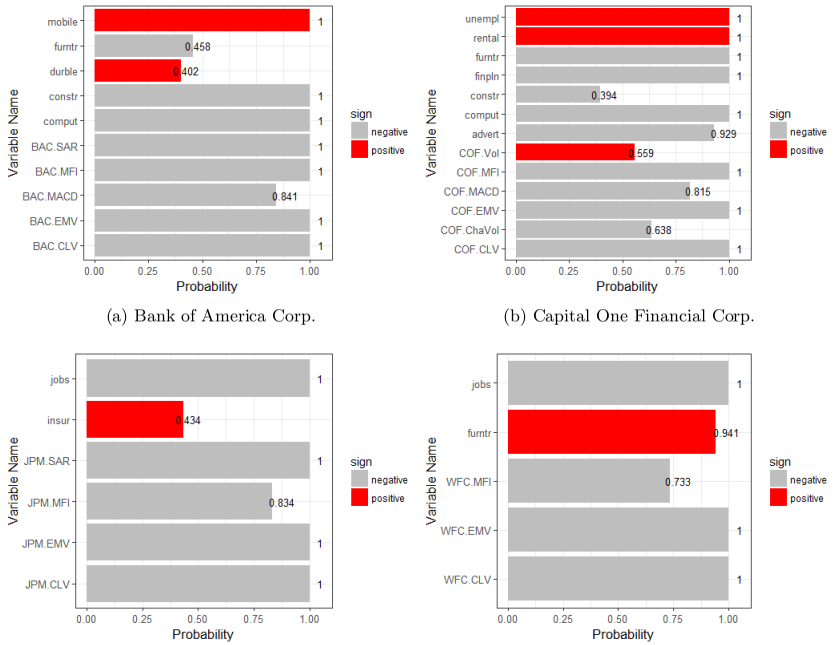
Thanks to the spike and slab regression, one advantage of the MBSTS model is that feature selection and model training can be done simultaneously, which prevents overfitting and avoids redundant or spurious predictors. That is, the MBSTS model is flexible in that it selects a different set of predictors for each target time series during the MCMC iterations. Moreover, we can set a different model size for each target time series by assigning appropriate values to the prior inclusion probabilities $\{\pi_{ij}\}$. The empirical posterior inclusion probability, as a useful indicator of the importance of one specific predictor, is the proportion of number of times that the predictor is selected to the total count of MCMC iterations. A higher inclusion probability indicates more variation of target time series can be explained by that predictor, whose chance of being selected depends on equation (29).

Figure 10 displays the predictors whose empirical posterior inclusion probabilities are greater than 0.2 for four companies. For the predictors with empirical inclusion probabilities equal to one, we can see that Bank of America has seven, Capital One Financial Corporation has eight, J.P. Morgan has four, and Wells Fargo has three. That is, the sets of predictors are different among these four companies; hence, the expected model size for each company also differs from each other. In general, sparsity was produced by our algorithm, and the size of the resulting model for each company is much less that of the total number of candidate predictors.

No such domestic Google trends contribute significantly to the variations of max log returns for all companies. Different sets of domestic Google trends capture the variations of max log returns of these four companies; more specifically, "mobile", "constr" and "comput" are the most important economic indicators for Bank of America, "unempl", "rental", "furntr", "finpln" and "comput" for Capital One Financial Corporation, "jobs" for J.P. Morgan and Wells Fargo. Among all the technical predictors, MFI, EMV and CLV were favored by the sampling algorithm for all companies, indicating the importance of these predictors in explaining the variations of max log returns.

4.4 Target Series Forecast

Time series forecasting is challenging, especially when it comes to multivariate target time series. One strength of our model is that it can make predictions for multiple target time series (i.e. max log returns of a stock portfolio) with a great number of contemporaneous predictors. Moreover, the Bayesian paradigm together with the spike-slab regression and MCMC algorithm can further improve prediction accuracy through model averaging technique. Similar to the performance analysis on simulated data, we compared the MBSTS model's performance using real financial market data with three other benchmark models: BSTS, ARIMAX and MARIMAX, measured by cumulative one-step-ahead prediction errors.



(a) Bank of America Corp.

(b) Capital One Financial Corp.

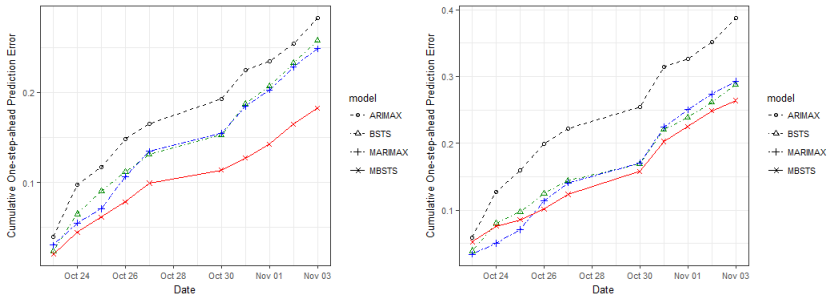
(c) JPMorgan Chase & Co.

(d) Wells Fargo & Co.

Figure 10: Empirical posterior inclusion probabilities for the most likely predictors of max log return. (a), (b), (c) and (d) display important predictors for BOA, COF, JPM and WFC respectively. Each bars is colored (red or gray) according to the sign (positive or negative) of the estimated value of the corresponding regression coefficient.

4.4.1 Model Comparison

Figure 11 shows the cumulative one-step-ahead prediction errors of these four models without and with deseasonalized predictors, respectively. We can see that the MBSTS model outperforms other benchmark models with smaller cumulative prediction errors at almost every step in these two cases. We can also see that models with original predictors outperform those using deseasonalized predictors. There are two obvious reasons to explain why the MBSTS model is the best. Firstly, benefiting from the multivariate setting, it captures the inherent correlations of multiple target time series after subtracting the effects of trend, seasonality and cycle components; these enables MBSTS to outperform the univariate BSTS model that is trained by each target time series individually. Secondly, Bayesian



(a) All Predictors Without Deaseasonal (b) Partial Predictors With Deaseasonal

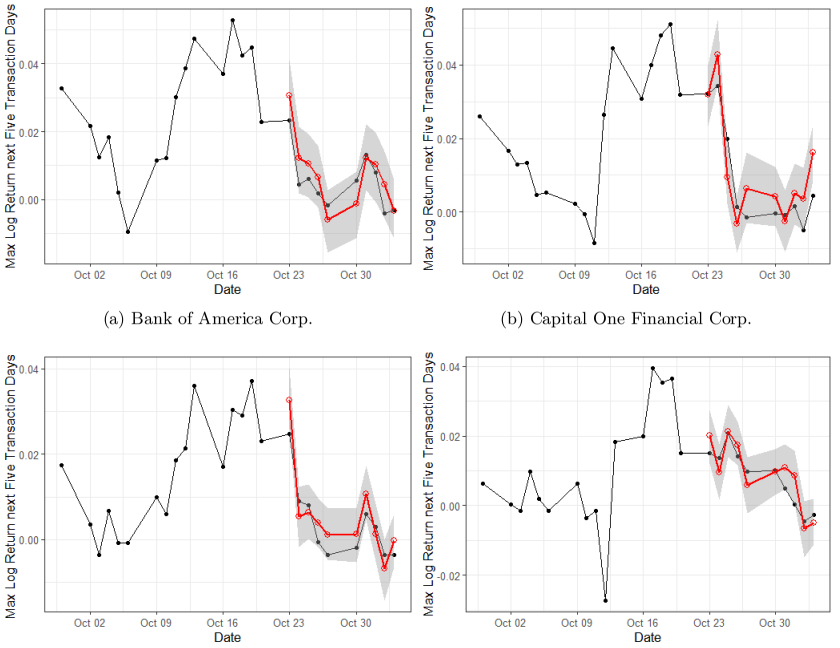
Figure 11: Performance analysis measured by cumulative one-step-ahead prediction errors: (a) displays the result when all predictors are original; (b) shows the result with some predictors with detected seasonality being deseasonalized. Other three benchmark models (BSTS, ARIMAX and MARIMAX) were also trained to make predictions.

model averaging helps avoid arbitrary selection and sticking to a fixed set of predictors, and the cyclical component can capture dramatic shocks to variations in target time series with diminishing impact, both of which enable our MBSTS model to outperform the MARIMAX model.

4.4.2 Trading Strategy

In finance, a trading strategy is a set of objective rules defining the conditions that must be met for a trade entry or exit action. Thanks to the strong prediction power, our model can provide supplemental guidelines to trading, given the current information of domestic Google trends and technical indexes. In other words, security strategists can decide when and how to trade based on the predictions from the MBSTS model.

Figure 12 shows one-step-ahead predictions by the MBSTS model for these four companies over two weeks. The shaded areas are the 40% prediction intervals generated by draws from the posterior distribution of $\hat{y}$. All true values are covered by the prediction intervals. The predicted value of max log return can be used as an indicator of whether to trade a stock or not. For example, if the lower bound of the predicted max log return is a large positive number, it is a strong signal that future prices will go substantially above the closing price of that day, thus buying this stock that day should be seriously considered. When the predicted value is positive but not large enough to cover transaction cost, it is a weak buying signal and a second thought should be given before making a decision. Selling or shorting the stock is suggested if the predicted max log return in the next five transaction days, is negative.



(a) Bank of America Corp. (b) Capital One Financial Corp.

(c) JPMorgan Chase & Co. (d) Wells Fargo & Co.

Figure 12: One-step-ahead predictions of max log returns: (a), (b), (c) and (d) display predicted and true max log return values for BOA, COF, JPM and WFC, respectively. Black lines with dots represent the true values, while red line with dots indicate predicted values. The gray shaded areas are 40% prediction bands.

5. Conclusion

In this paper, we have proposed a Multivariate Bayesian Structural Time Series (MBSTS) model for dealing with multiple target time series (e.g. max log returns of a stock portfolio), which helps in feature selection and forecasting in the presence of related external information. We evaluated the forecast performance of our model using both simulated and empirical data, and found that the MBSTS model outperforms three other benchmark models: BSTS, ARIMAX and MARIMAX. This superior performance can be attributed mainly to the following three reasons. Firstly, the MBSTS model derives its strength in forecasting from the fact that it incorporates information about other variables, rather than merely historical values of its own. Secondly, the Bayesian paradigm and the MCMC algorithm can perform variable selection at the same time as model training and thus prevent

overfitting even if some spurious predictors are added into the candidate pool. Thirdly, the MBSTS model benefits from taking correlations among multiple target time series into account, which helps boost the forecasting power. Therefore, this model, as expected, is able to provide more accurate forecasts than the univariate BSTS model and the traditional time series models such as ARIMA or MARIMA, when multiple target time series need to be modeled.

The excellent performance of the MBSTS model comes with high computation requirements in the MCMC iterations. Clearly, one would also not expect this model to show significant advantages over the univariate BSTS model, when multiple target series are independent of each other. But some preliminary exploratory analysis as well as professional insight would help to tell whether correlations in multiple target time series are strong enough in specific cases. Two open questions that are currently under investigation include: whether and how prior information such as model size and estimated coefficients can improve estimation accuracy and forecasting performance; the other is how to adjust this model to satisfy the need of analysis of non-Gaussian observations. Overall, it is fair to conclude that the MBSTS model offers practitioners a very good option to model or forecast multiple correlated target time series with a pool of available predictors.

Acknowledgements

We would like to thank the journal editor and the anonymous reviewers who provided us with many constructive and helpful comments.

References

Stephen Bach, Bert Huang, Ben London, and Lise Getoor. Hinge-loss Markov random fields: Convex inference for structured prediction. *arXiv:1309.6813*, 2013.

D. Blei and P. Smyth. Science and data science. *Proceedings of the National Academy of Sciences*, 114(33):8689–8692, 2017.

Kay H Brodersen, Fabian Gallusser, Jim Koehler, Nicolas Remy, and Steven L Scott. Inferring causal impact using Bayesian structural time-series models. *The Annals of Applied Statistics*, 9(1):247–274, 2015.

Francois Caron, Emmanuel Duflos, Denis Pomorski, and Philippe Vanheeghe. GPS/IMU data fusion using multisensor Kalman filtering: introduction of contextual aspects. *Information fusion*, 7(2):221–230, 2006.

Robert B Cleveland, William S Cleveland, and Irma Terpenning. STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, 6(1):3, 1990.

Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3): 273–297, 1995.

Lehel Csató and Manfred Opper. Sparse on-line Gaussian processes. *Neural computation*, 14(3):641–668, 2002.

James Durbin and Siem Jan Koopman. A simple and efficient simulation smoother for state space time series analysis. *Biometrika*, 89(3):603–616, 2002.

Jennifer G Dy and Carla E Brodley. Feature selection for unsupervised learning. *Journal of machine learning research*, 5(Aug):845–889, 2004.

Edward I George and Robert E McCulloch. Approaches for Bayesian variable selection. *Statistica sinica*, pages 339–373, 1997.

William E Griffiths. Bayesian inference in the seemingly unrelated regressions model. In *Computer-aided econometrics*, pages 263–290. CRC Press, 2003.

Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

Andrew C Harvey. *Forecasting, structural time series models and the Kalman filter*. Cambridge university press, 1990.

Andrew C Harvey, Thomas M Trimbur, and Herman K Van Dijk. Trends and cycles in economic time series: A Bayesian approach. *Journal of Econometrics*, 140(2):618–649, 2007.

Jennifer A Hoeting, David Madigan, Adrian E Raftery, and Chris T Volinsky. Bayesian model averaging: a tutorial. *Statistical science*, pages 382–401, 1999.

S Sathiya Keerthi and Chih-Jen Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural computation*, 15(7):1667–1689, 2003.

Terry Koo, Amir Globerson, Xavier Carreras Pérez, and Michael Collins. Structured prediction models via the matrix-tree theorem. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 141–150, 2007.

Balaji Krishnapuram, Lawrence Carin, Mario AT Figueiredo, and Alexander J Hartemink. Sparse multinomial logistic regression: Fast algorithms and generalization bounds. *IEEE transactions on pattern analysis and machine intelligence*, 27(6):957–968, 2005.

David Madigan and Adrian E Raftery. Model selection and accounting for model uncertainty in graphical models using Occam's window. *Journal of the American Statistical Association*, 89(428):1535–1546, 1994.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

Sebastian Nowozin and Christoph H Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3–4):185–365, 2011.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. Elements of causal inference: foundations and learning algorithms. *MIT Press*, 12:978–0, 2017.

Giovanni Petris, Sonia Petrone, and Patrizia Campagnoli. Dynamic linear models. *Dynamic Linear Models with R*, pages 31–84, 2009.

Tobias Preis, Helen Susannah Moat, and H Eugene Stanley. Quantifying trading behavior in financial markets using Google Trends. *Scientific reports*, 3:srep01684, 2013.

Peter E Rossi, Greg M Allenby, and Rob McCulloch. *Bayesian statistics and marketing*. John Wiley & Sons, 2012.

Said E Said and David A Dickey. Testing for unit roots in autoregressive-moving average models of unknown order. *Biometrika*, 71(3):599–607, 1984.

Steven L Scott and Hal R Varian. Predicting the present with Bayesian structural time series. *International Journal of Mathematical Modelling and Numerical Optimisation*, 5 (1-2):4–23, 2014.

Steven L Scott and Hal R Varian. Bayesian variable selection for nowcasting economic time series. In *Economic analysis of the digital economy*, pages 119–135. University of Chicago Press, 2015.

Luis Torgo. Data mining with R. *Learning with case studies. CRC, Boca Raton*, 2011.

Hal R Varian. Big data: New tricks for econometrics. *The Journal of Economic Perspectives*, 28(2):3–27, 2014.

Journal of Machine Learning Research 19 (2018) 1-45 Submitted 3/18; Revised 10/18; Published 10/18

Inverse Reinforcement Learning via Nonparametric Spatio-Temporal Subgoal Modeling

Adrian Šošić ADRIAN.SOSIC@SPG.TU-DARMSTADT.DE
Abdelhak M. Zoubir ZOUBIR@SPG.TU-DARMSTADT.DE
Signal Processing Group
Technische Universität Darmstadt
64283 Darmstadt, Germany

Elmar Rueckert RUECKERT@ROB.UNI-LUEBECK.DE
Institute for Robotics and Cognitive Systems
University of Lübeck
23538 Lübeck, Germany

Jan Peters MAIL@JAN-PETERS.NET
Autonomous Systems Labs
Technische Universität Darmstadt
64289 Darmstadt, Germany

Heinz Koeppl HEINZ.KOEPPL@BCS.TU-DARMSTADT.DE
Bioinspired Communication Systems
Technische Universität Darmstadt
64283 Darmstadt, Germany

Editor: George Konidaris

Abstract

Advances in the field of inverse reinforcement learning (IRL) have led to sophisticated inference frameworks that relax the original modeling assumption of observing an agent behavior that reflects only a single intention. Instead of learning a global behavioral model, recent IRL methods divide the demonstration data into parts, to account for the fact that different trajectories may correspond to different intentions, e.g., because they were generated by different domain experts. In this work, we go one step further: using the intuitive concept of *subgoals*, we build upon the premise that even a single trajectory can be explained more efficiently *locally* within a certain context than globally, enabling a more compact representation of the observed behavior. Based on this assumption, we build an implicit intentional model of the agent's goals to forecast its behavior in unobserved situations. The result is an integrated Bayesian prediction framework that significantly outperforms existing IRL solutions and provides smooth policy estimates consistent with the expert's plan. Most notably, our framework naturally handles situations where the intentions of the agent change over time and classical IRL algorithms fail. In addition, due to its probabilistic nature, the model can be straightforwardly applied in active learning scenarios to guide the demonstration process of the expert.

Keywords: Learning from Demonstration, Inverse Reinforcement Learning, Bayesian Nonparametric Modeling, Subgoal Inference, Graphical Models, Gibbs Sampling

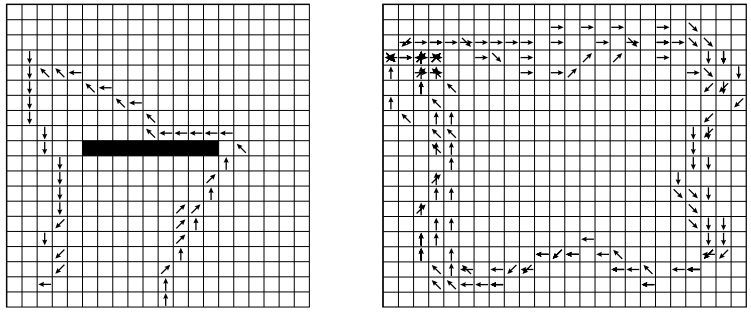
1. Introduction

Inverse reinforcement learning (IRL) refers to the problem of inferring the intention of an agent, called *the expert*, from observed behavior. Under the Markov decision process (MDP) formalism (Sutton and Barto, 1998), that intention is encoded in the form of a reward function, which provides the agent with instantaneous feedback for each situation encountered during the decision-making process. Classical IRL methods (Ng and Russell, 2000; Abbeel and Ng, 2004; Ziebart et al., 2008; Ramachandran and Amir, 2007; Levine et al., 2011) assume there exists a single *global* reward model that explains the entire set of demonstrations provided by the expert. In order to relax this rather restrictive modeling assumption, recent IRL methods allow that the agent's intention can change over time (Nguyen et al., 2015), or they presume that the demonstration data set is inherently composed of several parts (Dimitrakakis and Rothkopf, 2011), where different trajectories reflect the intentions of different domain experts.

In this work, we go a step further and start from the premise that — even in the case of a single expert or trajectory — the demonstrated behavior can be explained more efficiently *locally* (i.e., within a certain context) than by a global reward model. As an illustrative example, we may consider the task shown in Figure 1a, where the expert approaches a set of intermediate target positions before finally heading toward a global goal state. Similarly, in Figure 1b, the agent eventually returns to its initial position, from where the cyclic process repeats. Despite the simplicity of these tasks, the encoding of such behaviors in a global intention model requires a reward structure that comprises a comparably large number of redundant state-action-based rewards. Alternative modeling strategies rely on task-dependent expansions of the agent's state representation, e.g., to memorize the last visited goal (Krishnan et al., 2016), or they resort to more general decision-making frameworks like semi-MDPs/options (Bradtke and Duff, 1994; Sutton et al., 1999) in order to achieve the necessary level of task abstraction.

In this paper, we present a substantially simpler modeling framework that requires only minimal adaptations to the standard MDP formalism but comes with a hypothesis space of behavioral models that is sufficiently large to cover a broad class of expert policies. The key insight that motivates our approach is that many tasks, like those in Figure 1, can be decomposed into smaller subtasks that require considerably less modeling effort. The resulting low-level task descriptions can then be used as building blocks to synthesize arbitrarily complex behavioral strategies through a suitable sequencing of subtasks. This offers the possibility to learn comparably simple task representations using the intuitive concept of *subgoals*, which is achieved by efficiently encoding the expert behavior using task-adapted partitionings of the system state space/the expert data.

The proposed framework builds upon the method of Bayesian nonparametric inverse reinforcement learning (BNIRL, Michini and How, 2012), which can be used to build a subgoal representation of a task based on demonstration data — however, without learning the underlying subgoal relationships or providing a policy model that can generalize the strategy of the demonstrator. In order to address this limitation, we generalize the BNIRL model using insights from our previous works on nonparametric subgoal modeling (Šošić et al., 2018a) and policy recognition (Šošić et al., 2018b), building a compact intentional model of the expert's behavior that explicitly describes the local dependencies between the



(a) sequenced target positions (Michini and How, 2012)

(b) cyclic behavior

Figure 1: Two simple behavior examples that motivate the subgoal principle. The setting is based on the grid world dynamics described in Section 5.1. In both cases, a task description based on a global reward function is inefficient as it requires many state-action-based rewards to explain the observed trajectory structures. However, the data can be described efficiently through subgoal-based encodings. Both scenarios are analyzed in detail in Section 5.

demonstrations and the underlying subgoal structure. The result is an integrated Bayesian prediction framework that exploits the spatio-temporal context of the demonstrations and is capable of producing smooth policy estimates that are consistent with the expert's plan. Furthermore, capturing the full posterior information of the data set enables us to apply the proposed approach in an active learning setting, where the data acquisition process is controlled by the posterior predictive distribution of our model.

In our experimental study, we compare the proposed approach with common baseline methods on a variety of benchmark tasks and real-world scenarios. The results reveal that our approach performs significantly better than the original BNIRL model and alternative IRL solutions on all considered tasks. Interestingly enough, our algorithm outperforms the baselines even when the expert's true reward structure is dense and the underlying subgoal assumption is violated.

1.1 Related Work

The idea of decomposing complex behavior into smaller parts has been around for long and researchers have approached the problem in many different ways. While the overall field of methods is too large to be covered here, most existing approaches can be clearly categorized according to certain criteria. Often, two approaches differ in their exact problem formulation, i.e., we can distinguish between *active* methods, where the learning algorithm can interact freely with the environment (e.g., hierarchical reinforcement learning, Botvinick, 2012; Al-Emran, 2015), and *passive* methods, where the behavioral model is trained solely through observation (learning from demonstration, Argall et al., 2009). Furthermore, we

can discriminate between methods that build an explicit intentional model of the underlying task (IRL and option-based models, Choi and Kim, 2012; Sutton et al., 1999), and such that work directly on the control/trajectory level (skill learning, movement primitives, Konidaris et al., 2012; Schaal et al., 2005). The latter distinction is sometimes also referred to as *intentional/subintentional* approaches (Albrecht and Stone, 2017; Panella and Gmytrasiewicz, 2017). In order to give a concise summary of the work that is most relevant to ours, we restrict ourselves to passive approaches, with a focus on intentional methods, the field of which is considerably smaller. For an overview of active approaches, we refer to existing literature, e.g., the work by Daniel et al. (2016a).

First, there is the class of methods that pursue a decomposition of the observed behavior on the global level, using trajectory-based IRL approaches. For example, Dimitrakakis and Rothkopf (2011) proposed a hierarchical prior over reward functions to account for the fact that different trajectories in a data set could reflect different behavioral intentions, e.g., because they were generated by different domain experts. Similarly, Babeş-Vroman et al. (2011) follow an expectation-maximization-based clustering approach to group individual trajectories according to their underlying reward functions. Choi and Kim (2012) generalized this idea by proposing a nonparametric Bayesian model in which the number of intentions is a priori unbounded.

While the above methods consider the expert data at a global scale, our work is concerned with the problem of *subgoal modeling*, which is often conducted in the form of option-based reasoning (Sutton et al., 1999). For instance, Tamassia et al. (2015) proposed a clustering approach based on state distances to find a minimal set of options that can explain the expert behavior. While the method provides a simple alternative to handcrafting options, it does not allow any probabilistic treatment of the data and involves many ad-hoc design choices. Going in the same direction, Daniel et al. (2016a) presented a more principled, probabilistic option framework based on expectation-maximization. Not only is the framework capable of inferring sub-policies automatically, it can be also used in a reinforcement learning context for intra-option learning. However, the resulting behavioral model is based on point estimates of the policy parameters, and the number of sub-policies needs to be specified manually. The latter problem was solved by Krishnan et al. (2016), who proposed a hierarchical nonparametric IRL framework to learn a sequential representation of the demonstrated task, based on a set of transition regions that are defined through local changes in linearity of the observed behavior. However, in contrast to the work by Daniel et al. (2016a), inference is not performed jointly but in several isolated stages where, again, each stage only propagates a point estimate of the associated model parameters. Moreover, the temporal relationship of the demonstration data, used to identify the local linearity changes, is considered only in an ad-hoc fashion with the help of a windowing function.

Another general class of models, which explicitly addresses this issue, employs a hidden Markov model (HMM) structure to establish a temporal relationship between the demonstrations. For instance, the work presented by Nguyen et al. (2015) can be regarded as a generalization of the model by Babeş-Vroman et al. (2011), which extends the expectation-maximization framework by imposing a Markov structure on the reward model. Similarly, Niekum et al. (2012) use an extended HMM to segment the demonstrations into vector autoregressive models, in order to learn a suitable set of movement primitives. However, the learning of those primitives is done in a post-processing step, meaning that the quality of

the final representation crucially depends on the success of the initial segmentation stage. In contrast, the method by Rueckert et al. (2013) automatically learns the position and timing of subgoals in the form of via-points, but the number of via-points is assumed to be known and the system objective gets finally encoded in form of a global cost function. Recently, Lioutikov et al. (2017) presented a related approach based on probabilistic movement primitives that jointly solves the segmentation and learning step for an unknown number of primitives, using an expectation-maximization framework. Yet, the model operates purely on the trajectory level and cannot reveal the latent intentions of the demonstrator. Another variant of the approach by Niekum et al. (2012) that explicitly addresses this problem was proposed by Surana and Srivastava (2014). In their paper, the authors propose to replace the HMM emission model with an MDP model, in order to infer a policy model from the segmented trajectories instead of recognizing changes in the dynamics. The model was later extended by Ranchod et al. (2015), who augmented the HMM representation with a beta process model to facilitate skill sharing across trajectories. While the resulting model formulation is highly flexible, its major drawback is that inference becomes computationally expensive as it involves multiple IRL iterations per Gibbs step.

In contrast to the HMM-based solutions, which by their sequential nature focus on the *temporal* relationship of subtasks, the approach presented in this paper establishes a more general correlation structure between demonstrations by employing *non-exchangeable prior distributions* over subgoal assignments, i.e., without committing to purely temporal factorizations of subgoals. This results in a compact model representation (e.g., it avoids the need of estimating latent subgoal transition probabilities required in an HMM structure) and adds the flexibility to capture both, the temporal and the spatial dependencies between subtasks.

1.2 Paper Outline

The organization of the paper is as follows: in Section 2, we briefly revisit the BNIRL model and discuss its limitations, which forms the basis for our work. Section 3 then introduces a new intentional subgoal framework, which addresses the shortcomings of BNIRL discussed in Section 2. In Section 4, we derive a sampling-based inference scheme for our model and explain how the new framework can be used for subgoal extraction and action prediction. Experimental results on both synthetic and real-world data are presented in Section 5 before we finally conclude our work in Section 6.

2. Bayesian Nonparametric Inverse Reinforcement Learning

The purpose of this section is to recapitulate the principle of Bayesian nonparametric inverse reinforcement learning. After briefly discussing all building blocks of the model, we focus on the limitations of the framework, which motivates the need for an extended model formulation and finally leads to a new inference approach, presented afterwards in Section 3.

2.1 Revisiting the BNIRL Framework

Following the common IRL paradigm (Ng and Russell, 2000; Zhifei and Joo, 2012), the goal of BNIRL is to infer the intentions of an agent based on demonstration data. Starting from a standard MDP model, the problem is formalized on a finite state space $\mathcal{S}$, assuming a

time-invariant state transition model $T : \mathcal{S} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, where $\mathcal{A}$ is a finite set of actions available to the agent at each state. For notational convenience, we represent the states in $\mathcal{S}$ by the integer values $\{1, \dots, |\mathcal{S}|\}$, where $|\mathcal{S}|$ denotes the cardinality of the state space.

In BNIRL, it is assumed that we can observe a number of expert demonstrations provided in the form of state-action pairs, $\mathcal{D} \coloneqq \{(s_d, a_d)\}_{d=1}^{D}$, where each pair $(s_d, a_d) \in \mathcal{S} \times \mathcal{A}$ consists of a state s_d visited by the agent and the corresponding action a_d taken. Herein, D denotes the size of the demonstration set. Throughout the rest of this paper, we will use the shorthand notations $\mathbf{s} \coloneqq \{s_d\}_{d=1}^{D}$ and $\mathbf{a} \coloneqq \{a_d\}_{d=1}^{D}$ to access the collections of expert states and actions individually. Note that the BNIRL model makes no assumptions about the temporal ordering of the demonstrations, i.e., each state-action pair is considered to have arisen from a specific but arbitrary time instant of the agent's decision-making process. We will come back to this point later in Sections 2.2 and 3.3.

In contrast to the classical MDP formalism and most other IRL frameworks, BNIRL does *not* presuppose that the observed expert behavior necessarily originates from a single underlying reward function. Instead, it introduces the concept of *subgoals* (and corresponding *subgoal assignments*) with the underlying assumption that, at each decision instant, the expert selects a particular subgoal to plan the next action. Each subgoal is herein represented by a certain reward function defined on the system state space; in the simplest case, it corresponds to a single reward mass placed at a particular goal state in $\mathcal{S}$, which we identify with a reward function $R_g : \mathcal{S} \to \{0, C\}$ of the form

$$R_g(s) \coloneqq \begin{cases} C & \text{if } g = s, \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where $g \in \{1, \dots, |\mathcal{S}|\}$ indicates the subgoal location and $C \in (0, \infty)$ is some positive constant (compare Şimşek et al., 2005; Stolle and Precup, 2002; Tamassia et al., 2015).

Although in principle it is legitimate to associate each subgoal with an arbitrary reward structure to encode more complex forms of goal-oriented behavior (see, for example, Ranchod et al., 2015), the restriction to the reward function class in Equation (1) is sufficient in the sense that the same behavioral complexity can be synthesized through a combination of subgoals. This is made possible by the nonparametric nature of BNIRL, i.e., because the number of possible subgoals is assumed to be unbounded. The use of the reward model in Equation (1) has the advantage, however, that posterior inference about the expert's subgoals becomes computationally tractable, as will be explained in Section 4.5. In the following, we therefore focus on the above reward model and summarize the infinite collection of subgoals in the multiset $\mathcal{G} \coloneqq \{g_k\}_{k=1}^{\infty} \in \bigtimes_{k=1}^{\infty} \mathcal{S}$, where we adopt the assumption that $p(\mathcal{G} \,|\, \mathbf{s}) = \prod_{k=1}^{\infty} p_g(g_k \,|\, \mathbf{s})$.[1]

The subgoal assignment in BNIRL is achieved using a set of indicator variables $\tilde{\mathbf{z}} \coloneqq \{\tilde{z}_d \in \mathbb{N}\}_{d=1}^{D}$, which annotate each demonstration pair (s_d, a_d) with its unique subgoal index. The prior distribution $p(\tilde{\mathbf{z}})$ is modeled by a Chinese restaurant process (CRP, Aldous, 1985),

1. Notice that the subgoal prior distribution in the original BNIRL formulation does not take the state variable $\mathbf{s}$ as an argument. Nonetheless, the authors of BNIRL suggest to restrict the support of the distribution to the set of visited states, which indeed implies a conditioning on $\mathbf{s}$.

which assigns the event that indicator $\tilde{z}_d$ points to the jth subgoal the prior probability

$$p(\tilde{z}_d = j \,|\, \tilde{\mathbf{z}}_{\setminus d}) \propto \begin{cases} n_j & \text{if } j \in \{1, \dots, K\}, \\ \alpha & \text{if } j = K + 1, \end{cases}$$

where $\tilde{\mathbf{z}}_{\setminus d} \coloneqq \{\tilde{z}_d\} \setminus \tilde{z}_d$ is a shorthand notation for the collection of all indicator variables except $\tilde{z}_d$. Further, n_j denotes the number of assignments to the jth subgoal in $\tilde{\mathbf{z}}_{\setminus d}$, K represents the number of distinct entries in $\tilde{\mathbf{z}}_{\setminus d}$, and $\alpha \in [0, \infty)$ is a parameter controlling the diversity of assignments.

Having targeted a particular subgoal $g_{\tilde{z}_d}$ while being at some state s_d, the expert is assumed to choose the next action a_d according to a softmax decision rule, $\pi : \mathcal{A} \times \mathcal{S} \times \mathcal{S} \to [0, 1]$, which weighs the expected returns of all actions against one another,

$$\pi(a_d \,|\, s_d, g_{\tilde{z}_d}) \coloneqq \frac{\exp\left\{\beta Q^*(s_d, a_d \,|\, g_{\tilde{z}_d})\right\}}{\sum_{a \in \mathcal{A}} \exp\left\{\beta Q^*(s_d, a \,|\, g_{\tilde{z}_d})\right\}}. \tag{2}$$

Herein, $Q^*(s, a \,|\, g)$ denotes the state-action value (or *Q-value*, Sutton and Barto, 1998) of action a at state s under an optimal policy for the subgoal reward function R_g,

$$Q^*(s, a \,|\, g) \coloneqq \max_{\bar{\pi}} \mathbb{E}\left[\sum_{n=0}^{\infty} \gamma^n R_g(s_{t=n}) \;\middle|\; s_{t=0} = s, a_{t=0} = a, \bar{\pi}\right], \tag{3}$$

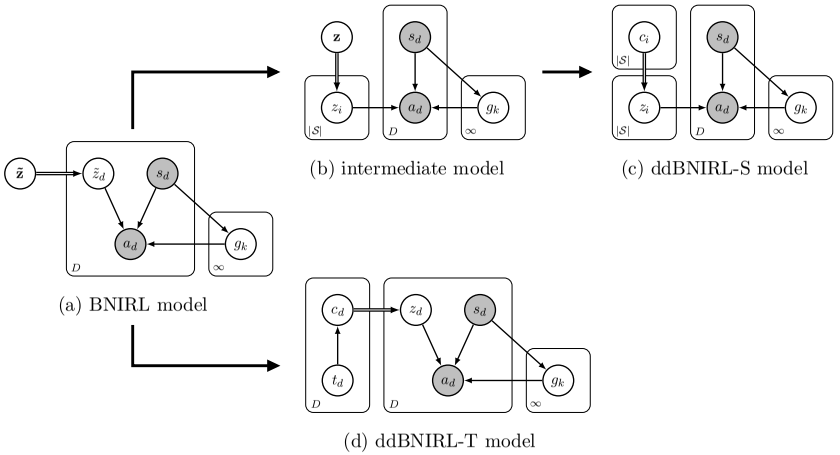
where the expectation is with respect to the stochastic state-action sequence induced by the fixed policy $\bar{\pi} : \mathcal{S} \to \mathcal{A}$, with initial action a executed at the starting state s. The explicit notation $s_{t=n}$ and $a_{t=n}$ is used to disambiguate the temporal index of the decision-making process from the demonstration index of the state-action pairs $\{(s_d, a_d)\}$.

The softmax policy π models the expert's (in-)ability to maximize the future expected return in view of the targeted subgoal, while the coefficient $\beta \in [0, \infty)$ is used to express the expert's level of confidence in the optimal action. Combined with the subgoal prior distribution p_g and the partitioning model $p(\tilde{\mathbf{z}})$, we obtain the joint distribution of all demonstrated actions $\mathbf{a}$, subgoals $\mathcal{G}$, and subgoal assignments $\tilde{\mathbf{z}}$ as

$$p(\mathbf{a}, \tilde{\mathbf{z}}, \mathcal{G} \,|\, \mathbf{s}) = p(\tilde{\mathbf{z}}) \prod_{k=1}^{\infty} p_g(g_k \,|\, \mathbf{s}) \prod_{d=1}^{D} \pi(a_d \,|\, s_d, g_{\tilde{z}_d}). \tag{4}$$

The structure of this distribution is visualized in form of a Bayesian network in Figure 2a. It is worth emphasizing that π — although referred to as the likelihood model for the state-action *pairs* in the original BNIRL paper — is really just a model for the actions *conditional on the states*. In contrast to what is stated in the original paper, the distribution in Equation (4) therefore takes the form of a *conditional* distribution (i.e., conditional on $\mathbf{s}$), which does not provide any generative model for the state variables.

Posterior inference in BNIRL refers to the (approximate) computation of the conditional distribution $p(\tilde{\mathbf{z}}, \mathcal{G} \,|\, \mathcal{D})$, which allows to identify potential subgoal locations and the corresponding subgoal assignments based on the available demonstration data. For further details, the reader is referred to the original paper (Michini and How, 2012).



(a) BNIRL model

(b) intermediate model

(c) ddBNIRL-S model

(d) ddBNIRL-T model

Figure 2: Relationships between all discussed subgoal models, illustrated in the form of Bayesian networks. Shaded nodes represent observed variables; deterministic dependencies are highlighted using double strokes.

2.2 Limitations of BNIRL

Subgoal-based inference is a well-motivated approach to IRL and the BNIRL framework has shown promising results in a variety of real-world scenarios. Yet, the model formulation by Michini and How (2012) comes with a number of significant conceptual limitations, which we explain in detail in the following paragraphs.

Limitation 1: Subgoal Exchangeability and Posterior Predictive Policy

The central limitation of BNIRL is that the framework is restricted to pure subgoal extraction and does *not* inherently provide a reasonable mechanism to generalize the expert behavior based on the inferred subgoals. The reason lies in the particular design of the framework, which, at its heart, treats the subgoal assignments $\tilde{\mathbf{z}}$ as *exchangeable random variables* (Aldous, 1985). By implication, the induced partitioning model $p(\tilde{\mathbf{z}})$ is agnostic about the covariate information contained in the data set and the resulting behavioral model is unable to propagate the expert knowledge to new situations.

To illustrate the problem, let us investigate the predictive action distribution that arises from the original BNIRL formulation. For simplicity and without loss of generality, we may assume that we have perfectly inferred all subgoals $\mathcal{G}$ and corresponding subgoal assignments $\tilde{\mathbf{z}}$ from the demonstration set $\mathcal{D}$. Denoting by $a^* \in \mathcal{A}$ the predicted action at some

new state $s^* \in \mathcal{S}$, the BNIRL model yields

$$\begin{aligned} p(a^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) &= \sum_{\tilde{z}^* \in \mathbb{N}} p(a^*, \tilde{z}^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) \\ &= \sum_{\tilde{z}^* \in \mathbb{N}} p(a^* \mid \tilde{z}^*, s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) p(\tilde{z}^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G}) \\ &\overset{(\star)}{=} \sum_{\tilde{z}^* \in \mathbb{N}} p(a^* \mid \tilde{z}^*, s^*, g_{\tilde{z}^*}) p(\tilde{z}^* \mid \tilde{\mathbf{z}}), \end{aligned} \tag{5}$$

where $\tilde{z}^* \in \mathbb{N}$ is the latent subgoal index belonging to s^*. Note that $p(a^* \mid \tilde{z}^*, s^*, g_{\tilde{z}^*})$ can either represent the softmax decision rule $\pi(a^* \mid s^*, g_{\tilde{z}^*})$ from Equation (2) or an optimal (deterministic) policy for subgoal $g_{\tilde{z}^*}$, depending on whether we aspire to describe the *noisy expert behavior* at s^* or want to determine an *optimal action* according to the inferred reward model. The last equality in Equation (5), indicated by $(\star)$, follows from the conditional independence properties implied by Equation (4), which can be easily verified using d-separation (Koller and Friedman, 2009) on the graphical model in Figure 2a.

As Equation (5) reveals, the predictive model is characterized by the posterior distribution $p(\tilde{z}^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G})$ of the latent subgoal assignment $\tilde{z}^*$ of state s^* — the intuition being that, in order to generalize the expert's plan to a new situation, we need to take into account the gathered information about what would be a likely subgoal targeted by the expert at s^*. However, in BNIRL, the distribution $p(\tilde{z}^* \mid s^*, \mathcal{D}, \tilde{\mathbf{z}}, \mathcal{G})$ is modeled *without consideration of the query state s^*, or any other observed variable*. By conditional independence (Equation 4), the distribution effectively reduces $(\star)$ to the CRP prior $p(\tilde{z}^* \mid \tilde{\mathbf{z}})$, which, due to its intrinsic exchangeability property, only considers the subgoal frequencies of the readily inferred assignments $\tilde{\mathbf{z}}$. Clearly, a subgoal assignment mechanism based solely on frequency information is of little use when it comes to predicting the expert behavior as it will inevitably ignore the structural information contained in the demonstration set and always return the same subgoal probabilities at all query states, regardless of the agent's actual situation. By contrast, a reasonable assignment mechanism should inherently take into account the context of the agent's current state s^* when deciding about the next action.

While the authors of BNIRL discuss the action selection problem in their paper and propose an assignment strategy for new states based on action marginalization, their approach does not provide a satisfactory solution to the problem because the alleged conditioning on the query state (see Equation 19 in the original paper, Michini and How, 2012) has no effect on the involved subgoal indicator variable, as shown by Equation (5) above. The only way to remedy the problem without modifying the model is to use an external post-processing scheme like the waypoint method, discussed in the next section.

Limitation 2: Spatial and Temporal Context

The waypoint method, described at full length in a follow-up paper by Michini et al. (2015), is a post-processing routine to convert the subgoals identified through BNIRL into a valid option model (Sutton et al., 1999). The obtained model reconstructs the high-level plan of the demonstrator by sequencing the inferred subgoals in a way that complies with the spatio-temporal relationships of the expert's decisions as observed during the demonstration phase. To this end, the required initiation and termination sets of the option-policies are

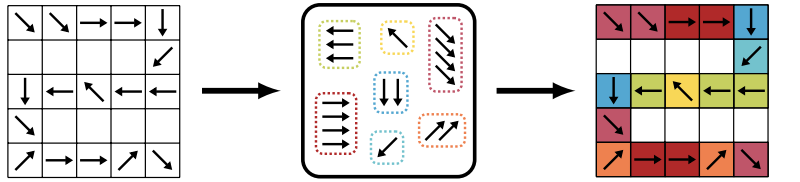


Figure 3: A diagram to illustrate the implications of the exchangeability assumption in BNIRL. Similar to a bag-of-words model (Blei et al., 2003; Yang et al., 2007), the BNIRL partitioning mechanism ignores the spatio-temporal context of the data, which makes it difficult to discriminate demonstration noise from a real change of the agent's intentions. Note that the diagram illustrates the partitioning process in a simplified way as it only shows the effect of the prior $p(\tilde{\mathbf{z}})$ but neglects the impact of the likelihood model π. While the latter *does* indeed consider the state context of the actions, it cannot account for spatial or temporal patterns in the data as it processes all state-action pairs separately.

constructed by considering the state distances to the identified subgoals as well as their temporal ordering prescribed by the expert.

When combined with BNIRL, this method allows to synthesize a behavioral model that mimics the observed expert behavior. However, the strategy comes with a number of significant drawbacks:

(i) Using the waypoint method, the spatio-temporal relationships between the individual demonstrations are explored only in a post-hoc fashion and are largely ignored during the actual inference procedure (the state information enters via the likelihood model π but is not considered by the partitioning model $p(\tilde{\mathbf{z}})$, as explained in Limitation 1). This lack of context-awareness makes the inference mechanism overly prone to demonstration noise (see Figure 3 and results in Section 5).

(ii) Measuring proximities to subgoals in order to determine the right visitation order requires some form of distance metric defined on the state space. If the system states correspond to physical locations, constructing such a metric is usually straightforward. However, in the general case where states encode arbitrary abstract information (see example in Section 5.2), it can become difficult to design that metric by hand. Unfortunately, the BNIRL framework does not provide any solution to this problem.

(iii) The waypoint method cannot be applied to multiple unaligned trajectories (e.g., obtained from different experts) or in cases where the data set does not carry any temporal information. This situation occurs, for instance, when the expert data is provided as separate state-action pairs with unknown timestamps and not given in form of coherent trajectories (see again example in Section 5.2).

(iv) Assigning a particular visitation order to the inferred subgoals is meaningful only if the expert eventually reaches those subgoals during the demonstration phase (or if, at least, the subgoals lie "close" to the visited states in terms of the aforementioned distance metric). Finding subgoals with such properties can be guaranteed by constraining the support of the subgoal prior distribution p_g to states that are near to the expert data (see footnote on page 6) but this reduces the flexibility of the model and potentially disables compact encodings of the task (Figure 4).

Limitation 3: Inconsistency under Time-Invariance

Reasoning about the intentions of an agent, there are two basic types of behavior one may encounter:

- either the agent follows a static strategy to optimize a fixed objective (as assumed in the standard MDP formalism, Sutton and Barto, 1998), or
- the intentions of the agent change over time.

The latter is clearly the more general case but also poses a more difficult inference problem in that it requires us both, to identify the intentions of the agent and to understand their temporal relationship. The static scenario, in contrast, implies that there exists an optimal policy for the task in form of a simple state-to-action mapping $\pi : \mathcal{S} \to \mathcal{A}$ (Puterman, 1994), which from the very beginning imprints a specific structure on the inference problem.

The BNIRL model generally falls into the second category since it freely allocates its subgoals per *decision instant* and not per state, allowing a flexible change of the agent's objective. Yet, it is important to understand that the model does not actually distinguish between the two described scenarios. As explained in Limitation 2, the temporal aspect of the data is not explicitly modeled by the BNIRL framework, even though the waypoint method subsequently tries to capture the overall chronological order of events. As a consequence, the model is not tailored to either of the two scenarios: on the one hand, it ignores the valuable temporal context that is needed in the time-varying case to reliably discriminate demonstration noise from a real change of the agent's intention. On the other hand, the model is agnostic about the predefined time-invariant nature of the optimal policy in the static scenario. This lack of structure not only makes the inference problem harder than necessary in both cases; it also allows the model to learn inconsistent data representations in the static case since the same state can be potentially assigned to more than one subgoal, violating the above-mentioned state-to-action rule (Figure 5).

Limitation 4: Subgoal Likelihood Model

Apart from the discussed limitations of the BNIRL partitioning model, it turns out there are two problematic issues concerning the softmax likelihood model in Equation (2). On the following pages, we demonstrate that the specific form of the model encodes a number of properties that are indeed contradictory to our intuitive understanding of subgoals. While these properties are less critical for the final prediction of the expert behavior, it turns out they drastically affect the *localization* of subgoals. Since the cause of these effects is somewhat hidden in the model equation, we defer the detailed explanation to Section 3.1.

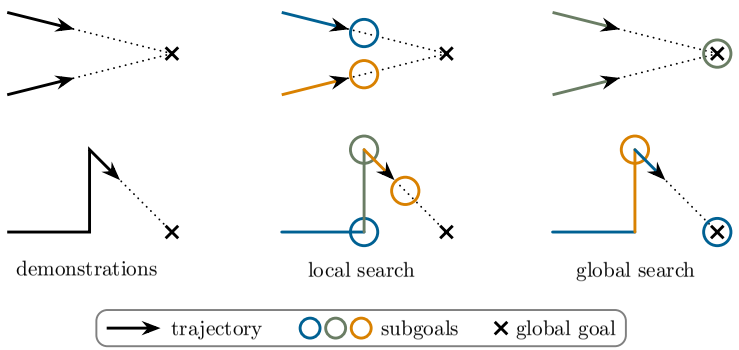


Figure 4: Difference between local (constrained) and global (unconstrained) subgoal search. The top and the bottom row depict two different sets of demonstration data (solid lines), together with potential goal/subgoal locations (crosses/circles) that explain the observed behavior. Color indicates the corresponding subgoal assignment of each trajectory segment. **Top:** two trajectories approaching the same goal. **Bottom:** the agent is heading toward a global goal, gets temporarily distracted, and then follows up on its original plan. **Left:** observed trajectories. **Center:** example partitioning under the assumption that the expert reached all subgoals during the demonstration. **Right:** example partitioning without restriction on the subgoal locations, yielding a more compact encoding of the task.

(a) time-varying intentions (b) time-invariant intentions

Figure 5: Schematic comparison of the two basic behavior types, illustrated using two different agent trajectories. Color indicates the temporal progress. (a) Time-varying intentions may cause the agent to perform a different action when revisiting a state (dotted circle). (b) By contrast, time-invariant intentions imply a simple state-to-action policy: the agent has no incentive to perform a different action at an already visited state since — by definition — the underlying objective has remained unchained. Diverging actions, as observed at the crossing point in the left subfigure, can therefore only be explained as a result of suboptimal behavior.

Limitation 5: State-Action Demonstrations

Lastly, a minor problem of the original BNIRL framework is that the inference algorithm expects the demonstration data to be provided in the form of state-action pairs, which requires full access to the expert's action record. This assumption is restrictive from a practical point of view as it confines the application of the model to settings with laboratory-like conditions that allow a complete monitoring of the expert. For this reason, it is important to note that an estimate of the expert's action sequence can be recovered through BNIRL with the help of an additional sampling stage (omitted in the original paper), provided that we know the successor state reached by the expert after each decision. For the marginalized inference scheme described in this paper, we present the corresponding sampling stage in Section 4.4.

3. Nonparametric Spatio-Temporal Subgoal Modeling

In this section, we introduce a redesigned inference framework, which, in analogy to BNIRL, we refer to as *distance-dependent Bayesian nonparametric IRL* (ddBNIRL). We derive the model by making a series of modifications to the original BNIRL framework that address the previously described shortcomings on the conceptual level. Rethinking each part of the original framework, we begin with a discussion of the commonly used softmax action selection strategy (Equation 2) in the context of subgoal inference, which finally leads to a redesign of the subgoal likelihood model (Limitation 4). Next, we focus on the subgoal allocation mechanism itself and introduce two closely related model formulations, each targeting one of the basic behavior types described in Figure 5, thereby addressing Limitations 1, 2 and 3. For the time-invariant case, we begin with an intermediate model that introduces a subtle yet important structural modification to the BNIRL framework. In a second step, we generalize that new model to account for the spatial structure of the control problem, which finally allows us to extrapolate the expert behavior to unseen situations. As part of this generalization, we present a new state space metric that arises naturally in the context of subgoal inference (see Limitation 2, second point). Lastly, we tackle the time-varying case and present a variant of the model that explicitly considers the temporal aspect of the subgoal problem. A solution to Limitation 5 is discussed later in Section 4.

In contrast to BNIRL, both presented models can be used likewise for *subgoal extraction* and *action prediction*. Moreover, sticking with the Bayesian methodology, the presented approach provides complete posterior information at all levels.

3.1 The Subgoal Likelihood Model

Like many other approaches found in the (I)RL literature, BNIRL exploits a softmax weighting (Equation 2) to transform the Q-values of an optimal policy into a valid subgoal likelihood model. The softmax action rule has its origin in RL where it is known as the Boltzmann exploration strategy (Cesa-Bianchi et al., 2017; Sutton and Barto, 1998), which is commonly applied to cope with the *exploration-exploitation dilemma* (Ghavamzadeh et al., 2015). In recent years, however, it has also become the de facto standard for describing the (imperfect) decision-making strategy of an observed demonstrator (see, for example, Dimitrakakis and Rothkopf, 2011; Ramachandran and Amir, 2007; Rothkopf and Dimitrakakis, 2011; Choi and Kim, 2012; Neu and Szepesvári, 2007; Babeş-Vroman et al., 2011).

In the following paragraphs, we focus on the implications of this model on the subgoal extraction problem and show that it contradicts our intuitive understanding of what characteristics a reasonable subgoal model should have. In particular, we argue that the subgoal posterior distribution arising from the BNIRL softmax model is of limited use for inferring the latent intention of the agent, due to subgoal artifacts caused by the system dynamics that cannot be reconciled with the evidence provided by the demonstrations. Based on these insights, we propose an alternative transformation scheme that is more consistent with the subgoal principle.

3.1.1 Scale of the Reward Function

The first implication of the softmax likelihood model concerns the choice of the uncertainty coefficient β. To explain the problem, we consider the thought experiment of an agent located at some state s targeting a particular subgoal g. The likelihood $\pi(a \,|\, s, g)$ in Equation (2) quantifies the probability that the agent decides for a specific action a, based on the corresponding state-action values $Q(\cdot, s \,|\, g)$. Since those values are linear in the underlying reward function R_g (Equation 3), the softmax likelihood model implies that the expert's ability to maximize the long-term reward, reflected by the spread of the probability mass in $\pi(\cdot \,|\, s, g)$, rises with the magnitude C of the assumed subgoal reward (more concentrated probability mass signifies a higher confidence in the action choice). In other words, assuming a higher goal reward virtually increases our level of confidence in the expert, even though the difficulty of the underlying task and the optimal policy remain unchanged. Nonetheless, the BNIRL model requires us to readjust the uncertainty coefficient β in order to keep both models consistent. However, as the model provides no reference level for the expert's uncertainty across different scenarios, the choice of β becomes nontrivial. Yet, the parameter has a significant impact on the granularity of the learned subgoal model as it trades off purposeful goal-oriented behavior against random decisions.

Note that the described effect is not specific to the subgoal reward model in Equation (1) but is really a consequence of the softmax transformation in Equation (2). In fact, the same problem occurs when the model is applied in a regular MDP environment with arbitrary reward function, for example, when the agent is provided an additional constant reward at all states. Clearly, such a constant reward provides no further information about the underlying task and should hence not affect the agent's belief about the optimal choice of actions (compare discussion on constant reward functions and transformations of rewards, Ng and Russell, 2000; Ng et al., 1999). Based on these two observations, our intuition tells us that we seek for a rationality model that is *invariant to affine transformations of the reward signal*, meaning that any two reward functions $R : \mathcal{S} \to \mathbb{R}$ and $\bar{R} \coloneqq xR + y$ with $x \in (0, \infty), y \in \mathbb{R}$, should give rise to the same intentional representation. As we shall see in Section 3.1.3, this can be achieved by modeling the behavior of an agent based on the *relative advantages* of actions rather than on their absolute expected returns.

3.1.2 Impact of the Transition Dynamics

The second implication of the softmax likelihood model is less immediate and inherently tied to the dynamics of the system. To explain the problem, we consider a scenario where we have a precise idea about the potential goals of the expert. For our example, we adopt

the grid world dynamics described in Section 5.1 and consider a simple upward-directed trajectory of state-action pairs, which we aspire to explain using a single (sub-)goal. The complete setting is depicted in Figure 6.

Intuitively, the shown demonstration set should lead to goals that are located in the upper region of the state space and concentrated around the vertical center line. Moreover, as we move away from that center line, we expect to observe a smooth decrease in the subgoal likelihood, while the rate of the decay should reflect our assumed level of confidence in the expert. As it turns out, the induced BNIRL subgoal posterior distribution, shown in the top row for different values of β, contradicts this intuition. In particular, we observe that the model yields unreasonably high posterior values at the upper border states and corners of the state space, which, according to our intuitive understanding of the problem, cannot be justified by the given demonstration set.

To pin down the cause of this effect, we recall from Equation (2) that the likelihood of an action grows with the corresponding Q-value. Hence, we need to ask what causes the Q-values of the demonstrated actions to be large when the subgoal is assumed to be located at one of the upper corner/border states of the space. Using Bellman's principle, we can express the optimal Q-function for any subgoal g as

$$\begin{aligned} Q^*(s,a\,|\,g) &= R_g(s) + \gamma\,\mathbb{E}_T\big[V^*(s'\,|\,g)\,|\,s,a\big] \\ &= R_g(s) + \gamma\,\mathbb{E}_T\big[\mathbb{E}_{\rho^{\pi_g}}[R_g(s'')\,|\,s']\,|\,s,a\big] \\ &= R_g(s) + \gamma\,\mathbb{E}_T\big[C\rho^{\pi_g}(g\,|\,s')\,|\,s,a\big], \end{aligned} \tag{6}$$

where $V^*(s\,|\,g) \coloneqq \max_{a\in\mathcal{A}} Q^*(s,a\,|\,g)$, $\pi_g(s) \coloneqq \arg\max_{a\in\mathcal{A}} Q^*(s,a\,|\,g)$ is the optimal policy for subgoal g, and C is the subgoal reward from Equation (1). Lastly, $\rho^{\pi_g}(s'\,|\,s) \coloneqq \sum_{t=0}^{\infty}\gamma^t p_t(s'\,|\,s,\pi_g)$ denotes the (improper) discounted state distribution generated by executing policy π_g from the considered initial state s, where $p_t(s'\,|\,s,\pi_g)$ refers to the probability of reaching state s' from state s under policy π_g after exactly t steps, which is defined implicitly via the transition model T.

The outer expectation in Equation (6) accounts for the stochastic transition to the successor state s', while the inner expectation evaluates the expected cumulative reward over all states s'' that are reachable from s'. It is important to note that — by the construction of the Q-function — only the first move of the agent to state s' depends on the choice of action a whereas all remaining moves (i.e., the argument of the expectation in the last line) are purely determined by the system dynamics and the subgoal policy π_g. Focusing on that inner part, we conclude that, *regardless of the chosen action* a, the Q-values will be large whenever the assumed subgoal induces a high state visitation frequency ρ^{π_g} at its own location g. The latter is fulfilled if

(i) the chance of reaching the goal in a small number of steps is high so that the effect of discounting is small and/or

(ii) the controlled transition dynamics $T(s'\,|\,s,\pi_g(s))$ that are induced by the subgoal lead to a high chance of hitting the goal frequently.

Note that the first condition implies that the model generally prefers subgoals that are close to the demonstration set — a property that cannot be justified in all cases. For example, the recording of the demonstrations could have simply ended before the expert was able to

reach the goal (Figure 5). Yet, if desired, this proximity property should be more naturally attributed to the subgoal prior model $p_g(g\,|\,\mathbf{s})$.

Moreover, we observe that the second condition depends primarily on the system dynamics T, which can be more or less strongly influenced by the actions of the agent, depending on the scenario. In fact, in a pathological example, T could be even independent of the agent's decisions, meaning that the agent has no control over its state. An example illustrating this extreme case would be a scenario where the agent gets always driven to the same terminal state, regardless of the executed policy. Although it is somewhat pointless speak of "subgoals" in this context, that terminal state would exhibit a high subgoal likelihood according to the softmax model because the corresponding visitation frequency would be inevitably large. A softened variant of this condition can occur at corner/border states (i.e., states in which the agent experiences fewer degrees of freedom and which are hence more difficult to leave than others) and transition states (i.e., states that must be passed in order to get from certain regions of the space to others), which naturally exhibit an increased visitation frequency due to the characteristics of the environment.

In our example in Figure 6, we can observe the symptoms of both described conditions clearly. In particular, for an upward-directed policy as it is implied by the shown demonstration set, the induced state visitation distribution exhibits increased values at exactly the aforementioned border and corner states (due to the reflections occurring to the agent when hitting the state space boundary) as well as close to the trajectory ending (caused by the proximity condition).

3.1.3 The Normalized Likelihood Model

To address these problems, we modify the likelihood model using a rescaling of the involved Q-values. Let $Q^{\wedge}(s\,|\,g)$ and $Q^{\vee}(s\,|\,g)$ denote the maximum and minimum Q-values at state s for subgoal g, i.e., $Q^{\wedge}(s\,|\,g) \coloneqq \max_{a\in\mathcal{A}} Q^*(s,a\,|\,g)$ and $Q^{\vee}(s\,|\,g) \coloneqq \min_{a\in\mathcal{A}} Q^*(s,a\,|\,g)$. We then define the normalized state-action value function $Q^{\bullet}: \mathcal{S}\times\mathcal{A}\times\mathcal{S}\to[0,1]$ as

$$Q^{\bullet}(s,a\,|\,g) \coloneqq \begin{cases} \frac{Q^*(s,a\,|\,g)-Q^{\vee}(s\,|\,g)}{Q^{\wedge}(s\,|\,g)-Q^{\vee}(s\,|\,g)} & \text{if } Q^{\wedge}(s\,|\,g) \neq Q^{\vee}(s\,|\,g), \\ \epsilon & \text{otherwise,} \end{cases} \tag{7}$$

where $\epsilon\in(0,1]$ is an arbitrary constant that is canceled out in Equation (8). In contrast to the Bellman state-action value function Q^*, which quantifies the expected return of an action, the normalized function $Q^{\bullet}$ assesses the return of that action in relation to the returns of all other actions. This concept is similar to that of the advantage function (Baird, 1993) with the important difference that the values returned by $Q^{\bullet}$ are normalized to the range $[0,1]$ and thus serve as an indicator for the *relative* quality of actions. Accordingly, the values can be interpreted as *relative advantages* (i.e., relative to the maximum possible advantage among all actions). The normalized subgoal likelihood model is then constructed analogously to the BNIRL likelihood model,

$$\pi^{\bullet}(a_d\,|\,s_d,g_{\tilde{z}_d}) \propto \exp\big\{\beta Q^{\bullet}(s_d,a_d\,|\,g_{\tilde{z}_d})\big\}. \tag{8}$$

The key property of this model is that it is invariant to affine transformations of the reward function, as summarized by the following proposition.

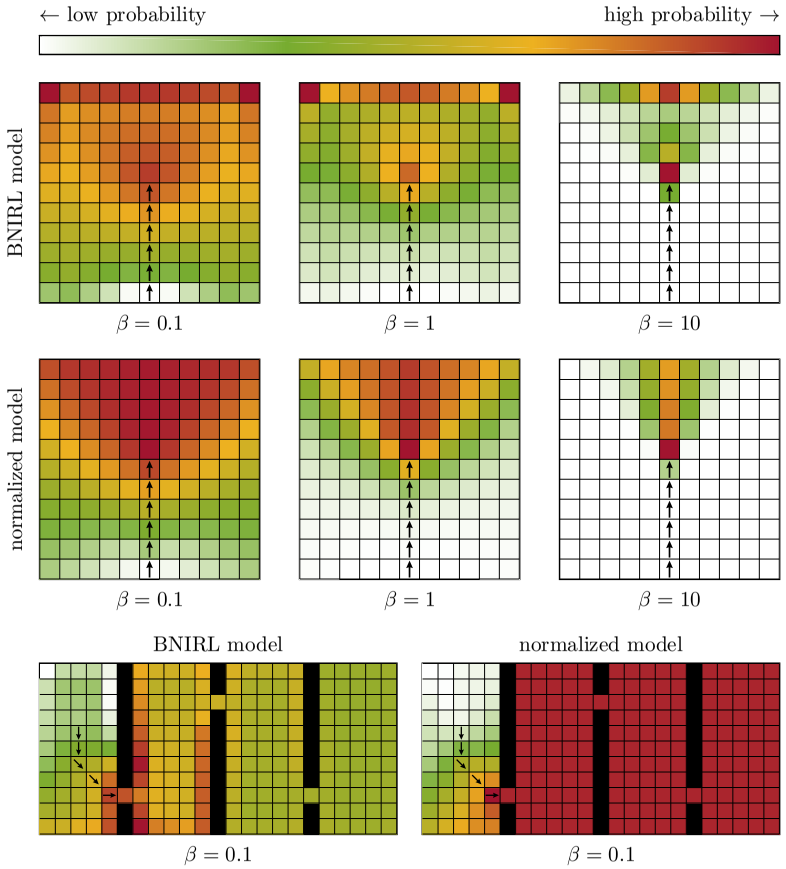


Figure 6: Comparison of the subgoal posterior distributions induced by the original BNIRL likelihood model and by the proposed normalized model, based on the grid world dynamics described in Section 5.1 and a uniform subgoal prior distribution p_g. The range of the shown color scheme is to be understood per subfigure. Black squares indicate wall states. The BNIRL likelihood model yields unreasonably high subgoal posterior mass at the border states and corners of the state space (due to locally increased state visitation probabilities arising from wall reflections) as well as at trajectory endings (caused by the implicit proximity property of the model)—see Section 3.1.2 for details. Both effects are mitigated by the proposed normalized likelihood model, which describes the action-selection process of the agent using relative advantages of actions instead of absolute returns.

Proposition 1 (Affine Invariance) *Consider an MDP with reward function $R : \mathcal{S} \to \mathbb{R}$ and let $Q^*(s, a \,|\, R)$ denote the corresponding optimal state-action value function. For the corresponding normalized function $Q^\bullet$ it holds that $Q^\bullet(s, a \,|\, R) = Q^\bullet(s, a \,|\, xR + y)$ $\forall x \in (0, \infty), y \in \mathbb{R}, s \in \mathcal{S}, a \in \mathcal{A}$. Hence, the subgoal likelihood model in Equation (8) is invariant to affine transformations of R.*

Proof Due to the linear dependence of Q^* on the reward function R (Equation 3) it holds that $Q^*(s, a \,|\, xR + y) = xQ^*(s, a \,|\, R) + \frac{y}{1-\gamma}$. Using this relationship in Equation (7), it follows immediately that $Q^\bullet(s, a \,|\, R) = Q^\bullet(s, a \,|\, xR + y)$. ■

Using the proposed likelihood model offers several advantages. First of all, it enables a more generic choice of the uncertainty coefficient β (Section 3.1.1). This is because the returned Q$^\bullet$-values lie in the fixed range [0,1], where 0 always indicates the lowest and 1 indicates the highest confidence. For example, setting $\beta = \log(\beta')$ for some $\beta' \in (0, \infty)$ always corresponds to the assumption that the expert chooses the optimal action with a probability that is β' times higher than the probability of choosing the least favorable action, irrespective of the underlying system model.

Moreover, as the results in Figure 6 reveal, the induced subgoal posterior distribution is notably closer to our expectation. The reason for this is twofold: first, a likelihood computation based on relative advantages mitigates the influence of the transition dynamics discussed in Section 3.1.2. This is because the described cumulation effect of the state visitation distribution ρ^{π_g} (Equation 6) is present in the returns of all actions and is thus reduced through the proposed normalization. For instance, if the agent in our grid world follows a policy that is all upward directed (as shown in the example), the induced state visitation distribution exhibits increased values at the upper border states of the world, even if we manipulated the *first* action of the agent (as considered in the Bellman Q-function). Accordingly, the original model would indicate an increased subgoal likelihood at those states. The normalized model, by contrast, is less affected as it constructs the likelihood by considering the increased visitation frequencies *relative* to each other.

Second, since the normalization diminishes the effect of the discounting, the subgoal posterior distribution is less concentrated around the trajectory ending and shows significant mass along the extrapolated path of the agent. This property allows us to identify far located states as potential goal locations, which adds more flexibility to the inferred subgoal constellation (compare Figure 4). As an illustrating example, consider the scenario shown in the bottom part of Figure 6. We observe that the normalized model assigns high posterior mass to all states in the right three corridors since any subgoal located in those corridors explains the demonstration set equally well. Here, the difference between the two models is even more pronounced because the transition dynamics have a strong impact on the agent behavior due to the added wall states. For further details, we refer to Section 5.1, where we provide additional insights into the subgoal inference mechanism.

3.2 Modeling Time-Invariant Intentions

With our redesigned likelihood model, we now focus on the partitioning structure of the model. Herein, we first consider the case where the intentions of the agent are constant with

respect to time. As explained in Limitation 3, this setting is consistent with the standard MDP formalism in the sense that the optimal policy for the considered task can be described in the form of a state-to-action mapping.

As a first step, to account for this relation, we establish a link between the model partitioning structure and the underlying system state space by replacing the demonstration-based indicators $\tilde{\mathbf{z}} = \{\tilde{z}_d \in \mathbb{N}\}_{d=1}^{D}$ with a new set of variables $\mathbf{z} \coloneqq \{z_i \in \mathbb{N}\}_{i=1}^{|\mathcal{S}|}$. Unlike $\tilde{\mathbf{z}}$, these new indicators do not operate directly on the data but are instead tied to the elements in $\mathcal{S}$. Although they formally represent a new type of variable, we can still imagine that their distribution follows a CRP. This yields an intermediate model of the form

$$p(\mathbf{a}, \mathbf{z}, \mathcal{G} \,|\, \mathbf{s}) = p(\mathbf{z}) \prod_{k=1}^{\infty} p_g(g_k \,|\, \mathbf{s}) \prod_{d=1}^{D} \pi^{\bullet}(a_d \,|\, s_d, g_{\mathbf{z}_{s_d}}),$$

whose structure is illustrated in Figure 2b. To see the difference to Equation (4), notice the way the subgoals are indexed in this model.

The intermediate model makes it possible to reason about the policy (or, more suggestively, the underlying state-to-action rule approximated by the expert) at visited parts of the state space. Yet, the model is unable to extrapolate the gathered information to unvisited states, for the reasons explained in Section 2.2. This problem can be solved by replacing the exchangeable prior distribution over subgoal assignments induced by the CRP with a non-exchangeable one, in order to account explicitly for the covariate state information contained in the demonstration set. Based on our insights from Bayesian policy recognition (Šošić et al., 2018b), we use the distance-dependent Chinese restaurant process (ddCRP, Blei and Frazier, 2011) for this purpose, which allows a very intuitive handling of the state context, as explained below. For alternatives, we point to the survey paper by Foti and Williamson (2015).

In contrast to the CRP, which assigns states to partitions, the ddCRP assigns states to other states, based on their pairwise distances. These "to-state" assignments are described by a set of indicators $\mathbf{c} \coloneqq \{c_i \in \mathcal{S}\}_{i=1}^{|\mathcal{S}|}$ with prior distribution $p(\mathbf{c}) = \prod_{i=1}^{|\mathcal{S}|} p(c_i)$,

$$p(c_i = j) = \begin{cases} \nu & \text{if } i = j, \\ f(\Delta_{i,j}) & \text{otherwise,} \end{cases} \tag{9}$$

for $i, j \in \mathcal{S}$. Herein, $\nu \in [0, \infty)$ is called the self-link parameter of the process, $\Delta_{i,j}$ denotes the distance from state i to state j, and $f : [0, \infty) \to [0, \infty)$ is a monotone decreasing score function. Note that the distances $\{\Delta_{i,j}\}$ can be obtained via a suitable metric defined on the state space, which may be furthermore used for calibrating the score function f (see subsequent section). The state partitioning structure itself is then determined by the connected components of the induced ddCRP graph (Figure 7). Our joint distribution, visualized in Figure 2c, thus reads as

$$p(\mathbf{a}, \mathbf{c}, \mathcal{G} \,|\, \mathbf{s}) = p(\mathbf{c}) \prod_{k=1}^{\infty} p_g(g_k \,|\, \mathbf{s}) \prod_{d=1}^{D} \pi^{\bullet}(a_d \,|\, s_d, g_{\mathbf{z}(\mathbf{c})|_{s_d}}), \tag{10}$$

where $\mathbf{z}(\mathbf{c})|_s$ denotes the subgoal label of state s arising from the considered indicator set $\mathbf{c}$. In order to highlight the state dependence of the underlying subgoal mechanism, we refer to this model as ddBNIRL-S.

3.2.1 The Canonical State Metric for Spatial Subgoal Modeling

The use of the ddCRP as a prior model for the state partitioning in Equation (10) inevitably requires some notion of distance between any two states of the system, in order to compute the involved function scores $\{f(\Delta_{i,j})\}$. When no such distances are provided by the problem setting (see Limitation 2, second point), a suitable (quasi-)metric can be derived from the transition dynamics of the system, which turns out to be the canonical choice for the ddBNIRL-S model. Consider the Markov chain governing the state process $\{s_{t=n}\}_{n=1}^{\infty}$ of an agent for some specific policy π. For any ordered pair of states (i, j), the chain naturally induces a value $\mathrm{T}_{i \to j}^{\pi}$, called a *hitting time* (Taylor and Karlin, 1984; Tewari and Bartlett, 2008), which represents the expected number of steps required until the state process, initialized at i, eventually reaches state j for the first time,

$$\mathrm{T}_{i \to j}^{\pi} \coloneqq \mathbb{E}\big[\min\{n \in \mathbb{N} : s_{t=n} = j\} \,|\, s_0 = i, \pi\big].$$

In the context of our subgoal problem, the natural quasi-metric to measure the directed distance between two states i and j is thus given by the time it takes to reach the goal state j from the starting state i under the corresponding optimal subgoal policy $\pi_j(s) = \arg\max_{a \in \mathcal{A}} Q^*(s, a \,|\, j)$, i.e., $\Delta_{i,j} \coloneqq \mathrm{T}_{i \to j}^{\pi_j}$. For ddBNIRL-S (as well as for the waypoint method in BNIRL), this choice is particularly appealing since the subgoal policies $\{\pi_j\}$ are already available within the inference procedure after the state-action values have been computed for the likelihood model (more on this in Section 4.5). The corresponding distances $\{\Delta_{i,j}\}$ can be obtained efficiently in a single policy evaluation step since $\Delta_{i,j}$ corresponds to the optimal (negative) expected return at the starting state i for the special setting where the respective target state j is made absorbing with zero reward while all other states are assigned a reward of -1.

3.2.2 Choice of the Score Function

From Equation (9) it is evident that the ddCRP model favors partitioning structures that result from the connection of nearby states. In the context of the subgoal problem, this property translates to the prior assumption that, most likely, each subgoal is approached by the expert from only one specific localized region in the system state space. While this assumption may be reasonable for some tasks, other tasks require that certain target states be approached more than one time, from different regions in the system state space. In such cases, it is beneficial if the model can reuse the same subgoal in various contexts, in order to obtain a more efficient task encoding (Figure 4).

From a mathematical point of view, the prerequisite for learning such encodings is that the score function f does not shrink to zero at large distance values, so that there remains a non-zero probability of connecting states that are far apart from each other. This can be achieved, for example, by representing f as a convex combination of a monotone decreasing zero-approaching function $\bar{f} : [0, \infty) \to [0, \infty)$ and some constant offset $\kappa \in (0, 1]$,

$$f(\Delta) = (1 - \kappa)\bar{f}(\Delta) + \kappa,$$

where $\bar{f}$ is chosen, e.g., as a radial basis function (Šošić et al., 2018b). Note that, in order to implement a desired degree of locality in the model, the scale of the decay function f (or $\bar{f}$, respectively) can be further calibrated based on the quantiles of the distribution of the given distances $\{\Delta_{i,j}\}$.

3.3 Modeling Time-Varying Intentions

For the case of changing expert intentions, we need to keep the flexibility of BNIRL to select a new subgoal at each decision instant, instead of restricting our policy to target a unique subgoal per state (Figure 5). Hence, we retain the basic BNIRL structure in this case and define the subgoal allocation mechanism using a set of data-related indicator variables. However, in contrast to BNIRL, which makes no assumptions about the temporal relationship of the subgoals and thus allows arbitrary changes of the expert's intentions (Section 2.1), we design our joint distribution in a way that favors smooth action plans in which the expert persistently follows a subgoal over an extended period of time. Again, we can make use of the ddCRP properties to encode the underlying smoothness assumption, but this time using a score function defined on the *temporal distance* between demonstration pairs. For this purpose, we require an additional piece of information, namely the unique timestamp of each demonstration example. Accordingly, we need to assume that our data set is of the form $\widetilde{\mathcal{D}} \coloneqq \{(s_d, a_d, t_d)\}_{d=1}^{D}$, where t_d denotes the recording time of the dth demonstration pair (s_d, a_d).[2]

The prior distribution over data partitionings can then be written as $p(\tilde{\mathbf{c}}) = \prod_{d=1}^{D} p(\tilde{c}_d)$,

$$p(\tilde{c}_d = d') \propto \begin{cases} \nu & \text{if } d = d', \\ f(\widetilde{\Delta}_{d,d'}) & \text{otherwise,} \end{cases}$$

where the indices $d, d' \in \{1, \dots, D\}$ range over the size of the demonstration set. Herein, $\widetilde{\Delta}_{d,d'} \coloneqq |t_d - t_{d'}|$ denotes the temporal distance between the data points d and d'. As before, we use the "$\sim$"-notation to distinguish the data-related partitioning variables $\tilde{\mathbf{c}}$, $\tilde{\mathbf{z}}$ and distances $\{\widetilde{\Delta}_{d,d'}\}$ from their state-space-related counterparts $\mathbf{c}$, $\mathbf{z}$ and $\{\Delta_{i,j}\}$ used in ddBNIRL-S. Note, however, that the score function f is independent of the underlying model type and may be chosen as described in Section 3.2.1, with a scale calibrated to the duration of the demonstrated task.

With that, we obtain our temporal subgoal model as

$$p(\mathbf{a}, \tilde{\mathbf{c}}, \mathcal{G} \mid \mathbf{s}) = p(\tilde{\mathbf{c}}) \prod_{k=1}^{\infty} p_g(g_k \mid \mathbf{s}) \prod_{d=1}^{D} \pi^{\bullet}(a_d \mid s_d, g_{\tilde{\mathbf{z}}(\tilde{\mathbf{c}})|_d}), \tag{11}$$

where $\tilde{\mathbf{z}}(\tilde{\mathbf{c}})|_d$ refers to the subgoal label of the dth demonstration pair induced by the given assignment $\tilde{\mathbf{c}}$. Analogous to our spatial subgoal model, we refer to this model as ddBNIRL-T. The structural differences between all models can be seen from Figure 2.

3.3.1 Relationship to BNIRL

Since the distance-dependent CRP contains the classical CRP as a special case for a specific choice of distance metric and score function (Blei and Frazier, 2011), the ddBNIRL-T model can be considered a strict generalization of the original BNIRL framework (neglecting the likelihood normalization in Section 3.1). In the same way, ddBNIRL-S generalizes

2. Note that the timestamps $\{t_d\}$ are naturally available if the demonstrations are recorded in trajectory form, where we observe several consecutive state-action pairs. In fact, the temporal information of the data is also required for the waypoint method to work (Limitation 2), even though the authors of BNIRL formally assume to have access to the reduced data set of state-action pairs only.

the intermediate model presented in Section 3.2 (Figure 2). However, although the BNIRL model can be recovered from ddBNIRL, it is important to note that the sampling mechanisms of both frameworks are fundamentally different. Whereas in BNIRL the subgoal assignments are sampled directly, the clustering structure in ddBNIRL is defined *implicitly* via the assignment variables $\mathbf{c}$ and $\tilde{\mathbf{c}}$, respectively. As explained by Blei and Frazier (2011), this has the effect that the Markov chain governing the Gibbs sampler mixes significantly faster because several cluster assignments can be altered in a single step, which effectively realizes a blocked Gibbs sampler (Roberts and Sahu, 1997).

3.4 Static versus Dynamic Subgoal Allocation

With the model structures described in Sections 3.2 and 3.3, we have presented two alternative views on the subgoal problem. Naturally, the question arises which of the two approaches is better suited for a particular application scenario. As explained in the previous paragraphs, the main difference between the two models lies in their structure, i.e., in the way subgoals are allocated. While ddBNIRL-S relies on a static assignment mechanism that consistently links the individual states of a system to their corresponding subgoals, ddBNIRL-T allocates its subgoals per demonstration pair. The latter means that different state-action pairs observed at the *same* state can be explained using *different* intentional settings (Figure 5). To answer the above question, we hence need to ask under which conditions an observed decision-making process can be described via a static assignment rule that *uniquely* characterizes each system state, and in which situations we require a more flexible model that allows to take into account additional side information.

From decision-making theory, we know that the optimal solutions for time-invariant MDPs can be formulated as a deterministic *time-invariant* Markov policies (Puterman, 1994), the class of which is fully covered by the static ddBNIRL-S framework.[3] Therefore, assuming that the transition dynamics of our system are constant with respect to time and that the agent acts rationally while having complete knowledge of the environment, there exist only two plausible reasons why we would potentially observe the agent execute a time-variant policy:

- either, the reward model of the agent changes over time,
- or, the observed decision-making process is not Markovian with respect to the assumed state space model (i.e., the agent's decisions depend on additional context information that is not explicitly captured in our state representation).

Accordingly, if we assume that the Markov property holds (meaning that the chosen state representation is sufficiently rich to capture the decision-making strategy of the agent), the only theoretical justification to prefer a dynamic subgoal model like ddBNIRL-T over a static one such as ddBNIRL-T would be if we assume that the intentions of the agent are truly time-dependent.

Practically speaking, however, there can be several reasons why a given state representation might not fulfill the Markov requirement. One obvious explanation would be that the actual state space of the demonstrator is not perfectly known. This situation occurs, for

3. While we omit a rigorous proof here, this can be seen intuitively by noticing that any state-to-action rule that is optimal for a given MDP reward function can be synthesized via ddBNIRL-S by assuming an individual subgoal for each state in the extreme case.

example, if not all state context available to the agent is observable by the modeler. Another potential situation is when the strategy of the agent depends on information that is independent of the system dynamics and hence deliberately excluded from the state variable (i.e., parameters that are unaffected by the actions of the agent, such as the preselection of a specific high-level strategy). A generic framework for such settings is described by Daniel et al. (2016b), where the agent learns multiple sub-policies that are triggered depending on context information that is treated separately from the state.

To an external observer who is unaware of that context information, the resulting policy of the agent would potentially appear time-dependent, in which case the only chance to disentangle the individual sub-policies would be to resort to a dynamic subgoal encoding, such as provided by ddBNIRL-T. However, if the context is known (like the temporal information in Section 3.3 as a particular example), both approaches can be used equivalently and will only differ in the resulting state representation. More specifically, we can either fall back on the static ddBNIRL-S model by augmenting the state variable with the context information accordingly, or we can resort to the dynamic subgoal allocation scheme of ddBNIRL-T, using a distance metric that accounts for the context. Conversely, when considered in a purely time-invariant setting (where the context is described by some other known quantity), ddBNIRL-S and ddBNIRL-T can be regarded as two sides of the same coin, i.e., both can be used to describe the time-invariant policy of an observed demonstrator but they differ in the way the side information is represented.

4. Prediction and Inference

Having introduced the ddBNIRL framework, we now explain how it can be used to generalize a given expert behavior. To this end, we first focus on the task of action prediction at a given query state, and then explain in a second step how to extract the necessary information from the demonstration data. Along the way, we also give insights into the implicit intentional model learned through the framework.

Note: In order to keep the level of redundancy at a minimum, the following considerations are based on the ddBNIRL-S model. The results for ddBNIRL-T follow straightforwardly; the only change in the equations is the way the subgoals are referenced. To obtain the corresponding expressions, we simply replace the assignment variables $\mathbf{c}$ with $\tilde{\mathbf{c}}$ and change the cluster definition in Equation (16) to $\mathcal{C}_k \coloneqq \{d \in \{1,\dots,D\} : \tilde{\mathbf{z}}(\tilde{\mathbf{c}})|_d = k\}$. Accordingly, all occurrences of $\mathbf{z}(\mathbf{c})|_{s^*}$ change to $\tilde{\mathbf{z}}(\tilde{\mathbf{c}})|_{d^*}$, $\mathbf{z}(\mathbf{c})|_{s_d}$ becomes $\tilde{\mathbf{z}}(\tilde{\mathbf{c}})|_d$, and $s_d \in \mathcal{C}_k$ is replaced with $d \in \mathcal{C}_k$.

4.1 Action Prediction

Similar to the work by Abbeel and Ng (2004), we consider the task of predicting an action $a^* \in \mathcal{A}$ at some query state $s^* \in \mathcal{S}$ that is optimal with respect to the expert's *unknown* reward model. However, in contrast to most existing IRL methods, our approach is not based on point estimates of the expert's reward function but takes into account the entire hypothesis space of reward models. This allows us to obtain the full posterior predictive policy from the expert data. Mathematically, the task is formulated as computing the predictive action distribution $p(a^* \mid s^*, \mathcal{D})$, which captures the full information about the expert

behavior contained in the demonstration set $\mathcal{D}$. We start by expanding that distribution with the help of the latent state assignments $\mathbf{c}$,

$$p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c} \in \mathcal{S}^{|\mathcal{S}|}} p(a^* \mid s^*, \mathcal{D}, \mathbf{c}) p(\mathbf{c} \mid \mathcal{D}).$$

The conditional distribution $p(a^* \mid s^*, \mathcal{D}, \mathbf{c})$ can be expressed in terms of the posterior distribution of the subgoal targeted at the query state s^*,

$$p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c} \in \mathcal{S}^{|\mathcal{S}|}} p(\mathbf{c} \mid \mathcal{D}) \sum_{i \in \mathcal{S}} p(a^* \mid s^*, \mathbf{c}, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i) p(g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i \mid \mathcal{D}, \mathbf{c}),$$

where we used the fact that the prediction a^* is conditionally independent of the demonstration set $\mathcal{D}$ given the state partitioning structure and the corresponding subgoal assigned to s^* (that is, given $\mathbf{c}$ and $g_{\mathbf{z}(\mathbf{c})|_{s^*}}$). From the joint distribution in Equation (10), it follows that

$$p(g_k \mid \mathcal{D}, \mathbf{c}) = \frac{1}{Z_k(\mathcal{D}, \mathbf{c})} p_g(g_k \mid \mathbf{s}) \prod_{d : \mathbf{z}(\mathbf{c})|_{s_d} = k} \pi(a_d \mid s_d, g_k), \tag{12}$$

where $Z_k(\mathcal{D}, \mathbf{c})$ is the corresponding normalizing constant,

$$Z_k(\mathcal{D}, \mathbf{c}) \coloneqq \sum_{i \in \mathrm{supp}(p_g)} p_g(g_k = i \mid \mathbf{s}) \prod_{d : \mathbf{z}(\mathbf{c})|_{s_d} = k} \pi(a_d \mid s_d, g_k = i). \tag{13}$$

Using this relationship, we get

$$\begin{aligned} p(a^* \mid s^*, \mathcal{D}) = \sum_{\mathbf{c} \in \mathcal{S}^{|\mathcal{S}|}} \frac{1}{Z_k(\mathcal{D}, \mathbf{c})} p(\mathbf{c} \mid \mathcal{D}) \sum_{i \in \mathrm{supp}(p_g)} p_g(g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i \mid \mathbf{s}) \ \dots \\ \dots \times \prod_{d : \mathbf{z}(\mathbf{c})|_{s_d} = \mathbf{z}(\mathbf{c})|_{s^*}} \pi(a_d \mid s_d, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i) p(a^* \mid s^*, \mathbf{c}, g_{\mathbf{z}(\mathbf{c})|_{s^*}} = i). \end{aligned}$$

In contrast to the summation over subgoal locations i, whose computational complexity is determined by the support of the subgoal prior distribution p_g and which grows at most linearly with the size of $\mathcal{S}$, the marginalization with respect to the indicator variables $\mathbf{c}$ involves the summation of $|\mathcal{S}|^{|\mathcal{S}|}$ terms and becomes quickly intractable even for small state spaces. Therefore, we approximate this operation via Monte Carlo integration, which yields

$$p(a^* \mid s^*, \mathcal{D}) \approx \frac{1}{N} \sum_{n=1}^{N} \sum_{i \in \mathrm{supp}(p_g)} p(g_{\mathbf{z}(\mathbf{c}^{\{n\}})|_{s^*}} = i \mid \mathcal{D}, \mathbf{c}^{\{n\}}) p(a^* \mid s^*, \mathbf{c}^{\{n\}}, g_{\mathbf{z}(\mathbf{c}^{\{n\}})|_{s^*}} = i),$$

where $\mathbf{c}^{\{n\}} \sim p(\mathbf{c} \mid \mathcal{D})$. The final prediction step can then be performed, for example, via the maximum a posteriori (MAP) policy estimate,

$$\hat{\pi}(s^*) \coloneqq \arg\max_{a^* \in \mathcal{A}} p(a^* \mid s^*, \mathcal{D}). \tag{14}$$

The inference task, hence, reduces to the computation of the posterior samples $\{\mathbf{c}^{\{n\}}\}$, which is described in the next section.

4.2 Partition Inference

Based on the joint model in Equation (10), we obtain the posterior distribution $p(\mathbf{c} \mid \mathcal{D})$ in factorized form as

$$p(\mathbf{c} \mid \mathcal{D}) = p(\mathbf{c}) \prod_{k=1}^{\infty} \sum_{g_k \in \mathrm{supp}(p_g)} p_g(g_k \mid \mathbf{s}) \prod_{d=1}^{D} \pi(a_d \mid s_d, g_{\mathbf{z}(\mathbf{c})|_{s_d}})$$
$$= p(\mathbf{c}) \prod_{k=1}^{|\mathbf{z}(\mathbf{c})|} \sum_{g_k \in \mathrm{supp}(p_g)} p_g(g_k \mid \mathbf{s}) \prod_{d: s_d \in \mathcal{C}_k} \pi(a_d \mid s_d, g_k), \tag{15}$$

where $\mathcal{C}_k$ denotes the kth state cluster induced by the assignment $\mathbf{c}$,

$$\mathcal{C}_k \coloneqq \{s \in \mathcal{S} : \mathbf{z}(\mathbf{c})|_s = k\}, \tag{16}$$

and $|\mathbf{z}(\mathbf{c})|$ is the total number of clusters defined by $\mathbf{c}$. As explained by Blei and Frazier (2011), the indicator samples $\{\mathbf{c}^{\{n\}}\}$ can be efficiently generated using a fast-mixing Gibbs chain. Starting from a given ddCRP graph defined by the subset of indicators $\mathbf{c}_{\setminus i} \coloneqq \{c_j\} \setminus c_i$, the insertion of an additional edge c_i will result in one of three possible outcomes, as illustrated in Figure 7: in the case of adding a self-loop ($c_i = i$), the underlying partitioning structure stays unaffected. Setting $c_i \neq i$ either leaves the structure unchanged (if the target state is already in the same cluster as state i) or creates a new link between two clusters. In the latter case, the involved clusters are merged, which corresponds to a merging of the associated sums in Equation (15). According to these three cases, the conditional distribution for the Gibbs procedure is obtained as

$$p(c_i = j \mid \mathbf{c}_{\setminus i}, \mathcal{D}) \;\propto\; \begin{cases} \nu & \text{if } i = j, \\ f(d_{i,j}) & \text{if no clusters are merged,} \\ f(d_{i,j}) \frac{\mathcal{L}(\mathcal{C}_{z_i} \cup \mathcal{C}_{z_j})}{\mathcal{L}(\mathcal{C}_{z_i}) \cdot \mathcal{L}(\mathcal{C}_{z_j})} & \text{if clusters } \mathcal{C}_{z_i} \text{ and } \mathcal{C}_{z_j} \text{ are merged.} \end{cases} \tag{17}$$

Herein, $\mathcal{L}(\mathcal{C})$ denotes the marginal action likelihood of all demonstrations accumulated in cluster $\mathcal{C}$,

$$\mathcal{L}(\mathcal{C}) = \sum_{g \in \mathrm{supp}(p_g)} p_g(g \mid \mathbf{s}) \prod_{d: s_d \in \mathcal{C}} \pi(a_d \mid s_d, g), \tag{18}$$

which further represents the normalizing constant for the posterior distribution of the cluster subgoal (Equation 13). Accordingly, the fraction in Equation (17) can be interpreted as the likelihood ratio of the partitioning defined by $\mathbf{c}_{\setminus i}$ and the merged structure after inserting the new edge c_i.

4.3 Subgoal Inference

It is important to note that the inference method described in Sections 4.1 and 4.2 is based on a collapsed sampling scheme where all subgoals of our model are marginalized out. In fact, the ddBNIRL framework differs from BNIRL and other IRL methods in that the reward model of the expert is never made explicit for predicting new actions. Nonetheless, if desired (e.g., for the purpose of analyzing the expert's intentions), an estimate of the subgoal locations can be obtained in a post-hoc fashion from the subgoal posterior distribution in Equation (12) for any given assignment $\mathbf{c}$. Examples are provided in Figure 8.

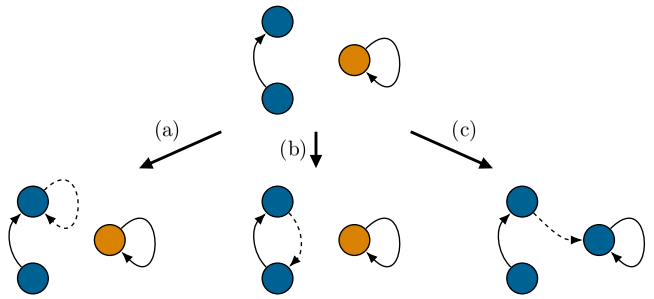


Figure 7: Insertion of an edge (dashed arrow) to the ddCRP graph. Colors indicate the cluster memberships of the nodes, which are defined implicitly via the connected components of the graph. (a) Adding a self-loop or (b) inserting an edge between two already connected nodes does not alter the clustering structure. (c) Adding an edge between two unconnected components merges the associated clusters.

4.4 Action Inference

As mentioned in Section 2.2, the original BNIRL algorithm requires complete knowledge of the expert's action record $\mathbf{a}$, which limits the range of potential application scenarios. For this reason, we generalize our inference scheme to the case where we have access to state information only, provided in the form of an alternative data set $\overline{\mathcal{D}} \coloneqq \{(s_d, \bar{s}_d)\}_{d=1}^{D}$, where $\bar{s}_d$ refers to the state visited by the expert immediately after s_d. In this setting, inference can be performed by extending the Gibbs procedure with an additional collapsed sampling stage,

$$p(a_d \mid \mathbf{a}_{\setminus d}, \overline{\mathcal{D}}, \mathbf{c}) \propto T(\bar{s}_d \mid s_d, a_d) \sum_{i \in \mathrm{supp}(p_g)} p_g(g_{\mathbf{z}(\mathbf{c})|_{s_d}} = i) \prod_{d': \mathbf{z}(\mathbf{c})|_{s_{d'}} = \mathbf{z}(\mathbf{c})|_{s_d}} \pi(a_{d'} \mid s_{d'}, g_{\mathbf{z}(\mathbf{c})|_{s_d}} = i), \tag{19}$$

which, for a fixed assignment $\mathbf{c}$, recovers an estimate of the latent action set $\mathbf{a}$ from the observed state transitions. Note that knowledge of the transition model T is required for this step as it provides the necessary link between the expert's actions and the observed successor states. The same extension is possible for the ddBNIRL-T model, provided that the transition timestamps $\{t_d\}$ are known (Section 3.3).

4.5 Computational Complexity

As a last point in this section, we would like to discuss the computational complexity of our approach. For this purpose, here a quick reminder on the used notation: we write $|\mathcal{S}|$ and $|\mathcal{A}|$ for the cardinalities of the state and action space, respectively, and use the letter D for the size of the demonstration set. Further, we write $\mathcal{C}_k$ to refer to the kth state cluster (ddBNIRL-S) or data cluster (ddBNIRL-T). In the subsequent paragraphs, we additionally use the notation $N_D(\mathcal{C}_k)$ to access the number of demonstration data points associated with

cluster $\mathcal{C}_k$, K to indicate the number of clusters in the current iteration, $N_g \coloneqq |\text{supp}(p_g)|$ as a shorthand for the size of the support of the subgoal prior distribution, and N_c for the number of indicator variables, i.e., $N_c \coloneqq |S|$ for ddBNIRL-S and $N_c \coloneqq D$ for ddBNIRL-T.

Initialization Phase: Common to all discussed models (including BNIRL) is that they depend on a preceding planning phase, where we compute, potentially in parallel, the state-action value functions (Equation 3) for all N_g considered subgoals, which allows us to construct the subgoal likelihood model (Equation 2 or 8). The overall computational complexity of this procedure is of order $\mathcal{O}(N_g \mathbb{C}_{\text{MDP}}(|\mathcal{S}|, |\mathcal{A}|))$, where $\mathbb{C}_{\text{MDP}}(x, y)$ denotes the complexity of the used planning routine to (approximately) solve an MDP of size x with a total number of y actions. Using a value iteration algorithm, for instance, this can be achieved in $\mathcal{O}(\mathbb{C}_{\text{MDP}}(|\mathcal{S}|, |\mathcal{A}|)) = \mathcal{O}(|\mathcal{S}|^2|\mathcal{A}|)$ steps (Littman et al., 1995). If we assume that the expert reaches all subgoals during the demonstration phase (Michini and How, 2012), we can restrict the support of the subgoal prior to the visited states, so that N_g is upper-bounded by $\min(|\mathcal{S}|, D)$. Note that there exist approximation techniques that make the computation tractable in large/continuous state spaces (see discussion in Section 6).

Before we start the sampling procedure, we compute all single-cluster likelihoods $\{\mathcal{L}(\mathcal{C}_k)\}$ and pairwise likelihoods $\{\mathcal{L}(\mathcal{C}_k \cup \mathcal{C}_{k'})\}$ according to Equation (18), based on some (random) initial cluster structure. The likelihood computation for the kth cluster $\mathcal{C}_k$ involves a product over $N_D(\mathcal{C}_k)$ data points, which needs to be calculated for each of the N_g subgoals before taking their weighted average. This step has to be executed (potentially in parallel) for all clusters. However, because each demonstration is associated with exactly one cluster (either directly as in ddBNIRL-T or via the corresponding state variable as in ddBNIRL-S) and hence $\sum_k N_D(\mathcal{C}_k) = D$, the total complexity for computing all single-cluster likelihoods is of order $\mathcal{O}(N_g D)$, irrespective of the actual cluster structure. A similar line of reasoning applies to the computation of the pairwise likelihoods, yielding the same complexity order. Yet, for the latter we need to consider all possible cluster combinations. Assuming an initial number of K clusters, there are in total $K(K-1)/2$ pairwise likelihoods to be computed. Hence, the overall complexity of the initialization phase can be summarized as $\mathcal{O}(N_g D K^2)$.

Partition Inference: For the partition inference, the bulk of the computation lies in the repeated construction of the likelihood term in Equation (17), which needs to be updated whenever the cluster structure changes. To analyze the complexity, we consider the sampling step of an individual assignment variable c_i (or likewise $\tilde{c}_i$). In the worst case, removing the edge that belongs to c_i from the ddCRP graph divides the associated cluster into two parts (Figure 7), so that two new single-cluster likelihoods need to be computed. With the upper bound D on the number of data points associated with the cluster before the division, this operation is of worst-case complexity $\mathcal{O}(N_g D)$ (see initialization phase). Irrespective of whether a division occurs, we then need to compute all pairwise cluster likelihoods with the (new) cluster connected via c_i. For a total of $K-1$ possible choices, this is done in $\mathcal{O}(N_g D K)$ operations (see initialization phase). After assigning the indicator, we move on to the next variable where the process repeats. If we assume, for simplicity, that the number of clusters stays constant during a full Gibbs cycle, the total complexity of updating all cluster assignments is hence of order $\mathcal{O}(N_g D K N_c)$. A (pessimistic) upper bound for the general case can be obtained by assuming that each data point defines its own cluster, in

which case the complexity increases to $\mathcal{O}(N_g D^2 N_c)$. Note that, in order to identify the new cluster structure after changing an assignment, we additionally need to track the connected components of the underlying ddCRP graph. As explained by Kapron et al. (2013), this can be done in polylogarithmic worst-case time.

Action Sampling: In order compute the conditional probability distribution of a particular action a_d, we need to evaluate a product involving all actions that belong to the same cluster as action a_d (Equation 19). First, we can compute the product over all actions except a_d itself, where the number of involved terms is again upper-bounded by D. Appending the term that belongs to a_d for all possible action choices requires another $|\mathcal{A}|$ operations. These two steps need to be repeated for all possible subgoals, yielding an upper bound on the complexity of order $\mathcal{O}(N_g(D + |\mathcal{A}|))$. For a full Gibbs cycle, which involves sampling all D action variables, the overall (worst-case) complexity is hence of order $\mathcal{O}(N_g(D + |\mathcal{A}|)D)$.

5. Experimental Results

In this section, we present experimental results for our framework. The evaluation is separated into four parts:

(i) a proof of concept and conceptual comparison to BNIRL (Section 5.1),
(ii) a performance comparison with related algorithms (Section 5.2),
(iii) a real data experiment conducted on a KUKA robot (Section 5.3) and
(iv) an active learning task (Section 5.4).

5.1 Proof of Concept

To illustrate the conceptual differences to BNIRL and provide additional insights into the latent intentional model learned through our framework, we begin with the motivating data set from Figure 1a, which had been originally presented by Michini and How (2012). The considered system environment, defined by $|\mathcal{S}| = 20 \times 20 = 400$ grid positions, is again shown in the top left corner of Figure 8. Nine of those positions correspond to inaccessible wall states, marked by the horizontal black bar. At the valid states, the expert can choose from an action set comprising a total of eight actions, each initiating a noisy state transition toward one of the (inter-)cardinal directions. The observed state-action pairs are depicted in the form of arrows, whose colors indicate the MAP partitioning learned through BNIRL. The remaining subfigures show the results of the ddBNIRL framework, which were obtained from a posterior sample returned by the respective algorithm (ddBNIRL-S/T) at a low temperature in a simulated annealing schedule (Kirkpatrick et al., 1983).

Comparing the obtained results, we observe the following main differences to the original approach:

(i) Unlike BNIRL, the proposed framework allows to choose between a spatial and a temporal encoding of the observed task, providing the possibility to account explicitly for the type of demonstrated behavior (static/dynamic). As explained in Section 3.3.1, the context-unaware (yet in principle dynamic) vanilla BNIRL inference scheme is still included as a special case.

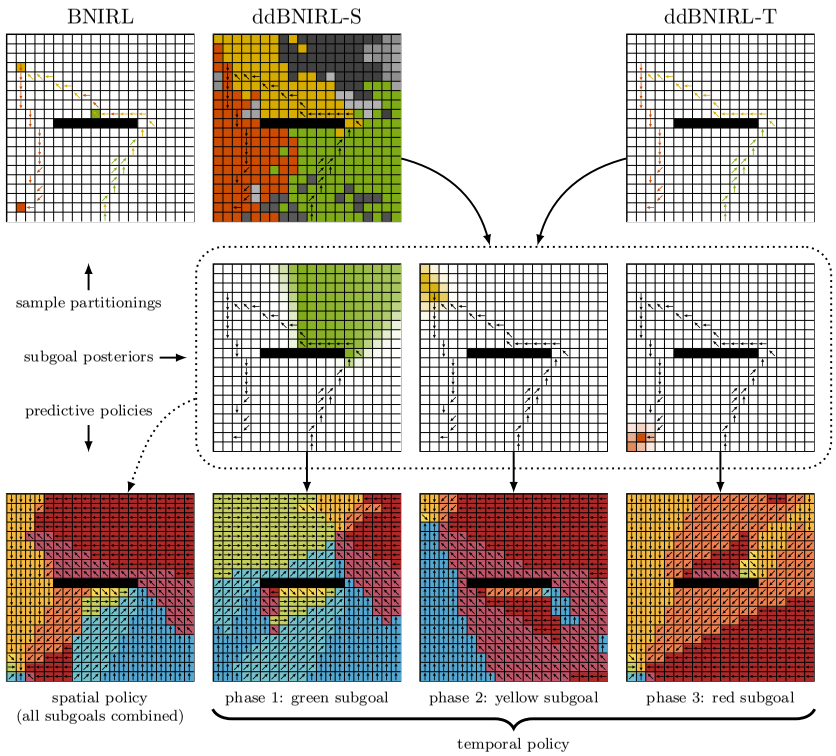


Figure 8: Results on the BNIRL data set (Michini and How, 2012). **Top row:** demonstration data and sample partitionings generated by the different inference algorithms. **Center row:** subgoal posterior distributions associated with the partitions found by ddBNIRL-S and ddBNIRL-T. For a clearer overview, the corresponding BNIRL distributions are omitted (see Figure 6 for a comparison). **Bottom row:** time-invariant ddBNIRL-S policy model synthesized from all three detected subgoals (left) and temporal phases identified by ddBNIRL-T (right). The background colors have no particular meaning and were added only to highlight the structures of the policies. Because of its missing generalization mechanism, BNIRL does not itself provide a reasonable predictive policy model (Limitation 1).

(ii) Exploiting the spatial/temporal context of the data, the ddBNIRL solution is inherently robust to demonstration noise, giving rise to notably smoother partitioning structures (top row). This effect is particularly pronounced in the case of real data, as we shall see later in Section 5.3.2.

(iii) For each state partition or trajectory segment, we obtain an implicit representation of the associated subgoal in the form of a posterior distribution, without the need of assigning point estimates (center row). It is striking that the posterior distribution corresponding to the green state partition has a comparably large spread on the upper side of the wall. This can be explained intuitively by the fact that any subgoal located in this high posterior region could have potentially caused the green state sequence, which circumvents the wall from the right. At the same time, the green area of high posterior values exhibits a sharp boundary on the left side since a subgoal located in the upper left region of the state space would have more likely resulted in a trajectory approaching from the left.

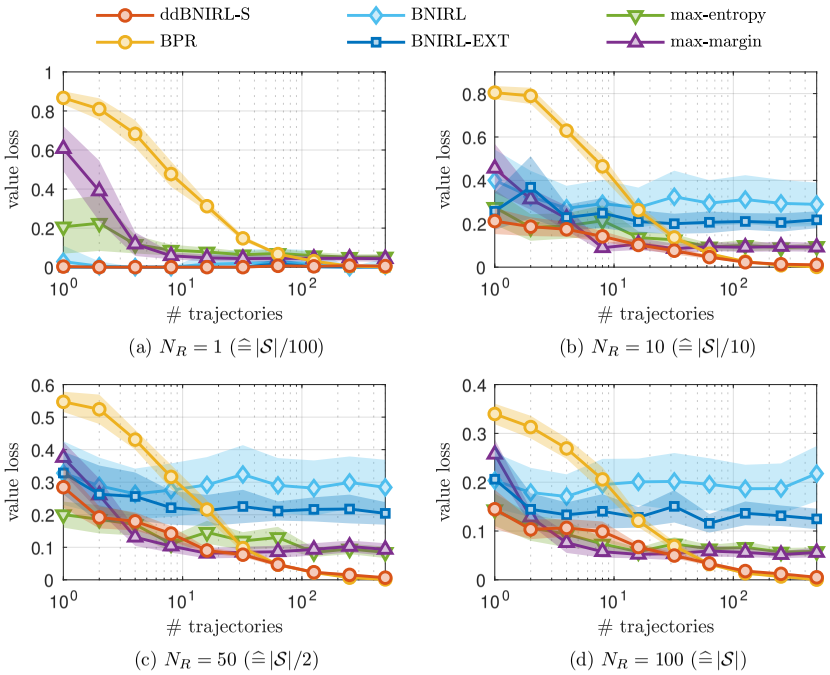
(iv) In contrast to BNIRL, which has no built-in generalization mechanism (Limitation 1), our method returns a predictive policy model comprising the full posterior action information at all states. Note that we only show the resulting MAP policy estimates here (bottom row), computed according to Equation (14). Additional results concerning the posterior uncertainty are provided in Sections 5.3 and 5.4.

The example illustrates how the synthesis of the predictive policy differs between ddBNIRL-S (bottom left) and ddBNIRL-T (bottom row, rightmost three subfigures). While ddBNIRL-T uses a set of (conditionally) independent policy models to describe the different identified behavioral phases, ddBNIRL-S maps the entire subgoal schedule onto a single time-invariant policy representation. Looking closer at the learned models, we recognize that the ddBNIRL-S solution in fact realizes a spatial combination of the three temporal ddBNIRL-T components, where each component is activated in the corresponding cluster region of the state space. This gives us two alternative interpretations of the same behavior.

5.2 Random MDP Scenario

Our next experiment is designed to provide insights into the generalization abilities of the framework. For this purpose, we consider a class of randomly generated MDPs similar to the Garnet problems (Bhatnagar et al., 2009). The transition dynamics $\{T(\cdot \mid s, a)\}$ are sampled independently from a symmetric Dirichlet distribution with a concentration parameter of 0.01, where we choose $|\mathcal{S}| = 100$ and $|\mathcal{A}| = 10$. For each repetition of the experiment, N_R states are selected uniformly at random and assigned rewards that are, in turn, sampled uniformly from the interval $[0, 1]$. All other states contain zero reward. Next, we compute an optimal deterministic MDP policy π^* with respect to a discount factor of $\gamma = 0.9$ and generate a number of expert trajectories of length 10. Herein, we let the expert select the optimal action with probability 0.9 and a random, suboptimal action with probability 0.1. The obtained state sequences are passed to the algorithms and we compute the normalized value loss of the reconstructed policies according to

$$\mathrm{L}(\pi^*, \hat{\pi}) := \frac{\|\mathbf{V}^* - \mathbf{V}^{\hat{\pi}}\|_2}{\|\mathbf{V}^*\|_2}, \tag{20}$$



(a) $N_R = 1$ ($\widehat{=}\,|\mathcal{S}|/100$) (b) $N_R = 10$ ($\widehat{=}\,|\mathcal{S}|/10$)

(c) $N_R = 50$ ($\widehat{=}\,|\mathcal{S}|/2$) (d) $N_R = 100$ ($\widehat{=}\,|\mathcal{S}|$)

Figure 9: Comparison of all inference methods in the random MDP scenario for different reward densities. Shown are the empirical mean values and standard deviations of the resulting value losses, obtained from 100 Monte Carlo runs. The graphs show a clear difference between BNIRL, BNIRL-EXT and ddBNIRL-S, which illustrates the importance of considering the spatial context for subgoal extraction.

where $\mathbf{V}^*$ and $\mathbf{V}^{\hat{\pi}}$ represent, respectively, the vectorized value functions of the optimal policy π^* and the reconstruction $\hat{\pi}$.

Since the considered system belongs to the class of time-invariant MDPs, ddBNIRL-S lends itself as the natural choice to model the expert behavior. As baseline methods, we adopt our subintentional Bayesian policy recognition framework (BPR, Šošić et al., 2018b), as well as maximum-margin IRL (Abbeel and Ng, 2004), maximum-entropy IRL (Ziebart et al., 2008), and vanilla BNIRL. Due to the missing generalization abilities of BNIRL (Limitation 1) and because the waypoint method (Section 2.2) does not straightforwardly apply to the considered scenario of multiple unaligned trajectories, we further compare our algorithm to an extension of BNIRL, which we refer to as BNIRL-EXT. Mimicking the ddBNIRL-S principle, the method accounts for the spatial context of the demonstrations

by assigning each state to the BNIRL subgoal that is targeted by the closest (see metric in Section 3.2.1) state-action pair — however, these assignments are made *after* the actual subgoal inference. When compared to ddBNIRL-S, this provides a reference of how much can be gained by considering the spatial relationship of the data *during* the inference. For the experiment, both ddBNIRL-S and BNIRL(-EXT) are augmented with their corresponding action sampling stages (Section 4.4) since the action sequences of the expert are discarded from the data set, in order to enable a fair comparison to the remaining algorithms.

Figure 9 shows the value loss over the size of the demonstration set for different reward settings. For small N_R, both BNIRL(-EXT) and ddBNIRL-S significantly outperform the reference methods. This is because the sparse reward structure allows for an efficient subgoal-based encoding of the expert behavior, which enables the algorithms to reconstruct the policy even from minimal amounts of demonstration data. However, the BNIRL(-EXT) solutions drastically deteriorate for denser reward structures. In particular, we observe a clear difference in performance between the cases where

(i) we do not account for the spatial information in the partitioning model (BNIRL),

(ii) include it in a post-processing step (BNIRL-EXT), and

(iii) exploit it during the inference itself (ddBNIRL-S),

which demonstrates the importance of processing the context information. Most tellingly, ddBNIRL-S outperforms the baseline methods even in the dense reward regimes, although the subgoal-based encoding loses its efficiency here. In fact, the results reveal that the proposed approach combines the merits of both model types, i.e., the sample efficiency of the intentional models (max-margin/max-entropy) required for small data set sizes, as well as the asymptotic accuracy and fully probabilistic nature of the subintentional Bayesian framework (BPR).[4]

5.3 Robot Experiment

In the next experiment, we test the ddBNIRL framework on various real data sets, which we recorded on a KUKA lightweight robotic arm (Figure 10) via kinesthetic teaching. Videos of all demonstrated tasks can be found at `http://www.spg.tu-darmstadt.de/jmlr2018`.

The system has seven degrees of freedom, corresponding to the seven joints of the arm. Each joint is equipped with a torque sensor and an angle encoder, providing recordings of joint angles, velocities and accelerations. For our experiments, we only consider the xy-Cartesian coordinates spanning the transverse plane, which we computed from the raw measurements using a forward kinematic model. The data was recorded at a sampling rate of 50 Hz and further downsampled by a factor of 10, yielding an effective sample rate of 5 Hz, which provided a sufficient temporal resolution for the considered scenario.

The goal of the experiment is to learn a set of high-level intentional models for the recorded behavior types by partitioning the data sets into meaningful parts that can be used to predict the desired motion direction of the expert. For simplicity and to demonstrate the

4. The comparably large loss of BPR for small data set sizes can be explained by the fact that the framework is based on a more general policy model in which the expert behavior is assumed to be *inherently* stochastic, in contrast to the here considered setting where stochasticity arises merely a consequence of suboptimal decision-making.

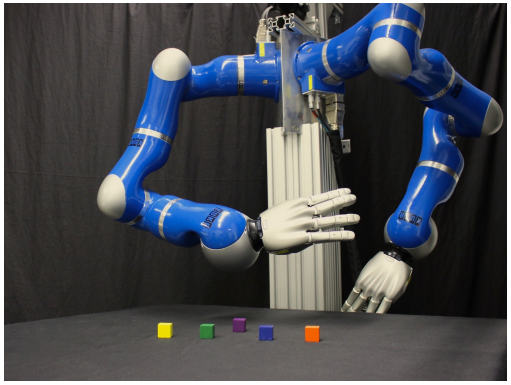


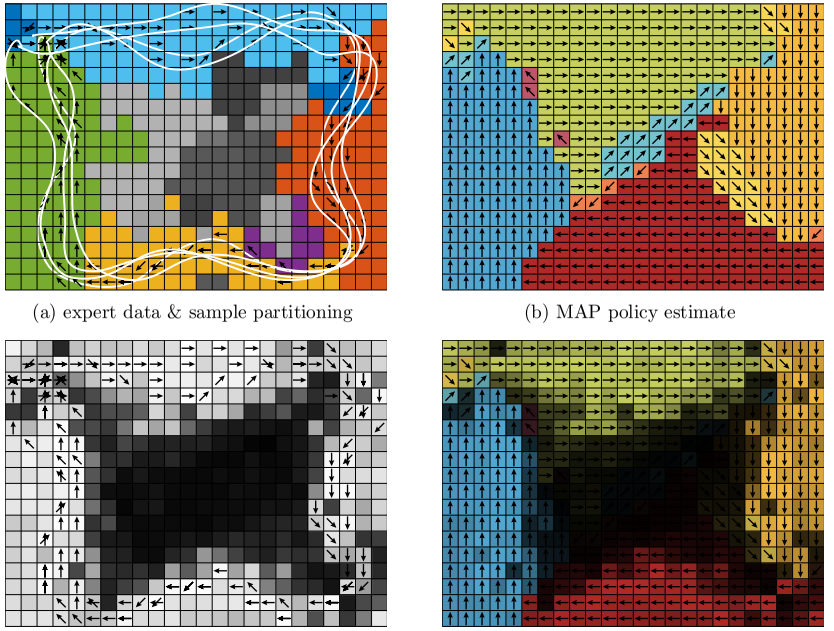
Figure 10: KUKA lightweight robotic arm.

algorithm's robustness to modeling errors, we adopt the simplistic transition model from Section 5.1 with the same action set containing the eight (inter-)cardinal motion directions. The high measurement accuracy of the end-effector position allows us to extract these high-level actions directly from the raw data, i.e., by selecting the directions with the smallest angular deviations from the ground truth (see example in Figure 11a). The underlying state space is obtained by discretizing the part of the coordinate range that is covered by the measurements into blocks of predefined size (see next sections for details). Apart from this discretization step and the aforementioned data downsampling, no preprocessing is applied.

5.3.1 Spatial Partitioning

First, we consider a case where the expert behavior can be described using a time-invariant policy model, which we aspire to capture via ddBNIRL-S. For our example, we consider the "Cycle" task shown in the video and in Figure 14. The same setting is analyzed using the time-variant ddBNIRL-T model in Section 5.3.2, which allows a direct comparison of the two approaches. The task consists in approaching a number of target positions, indicated by a set of markers (see video), before eventually returning to the initial state. The setting can be regarded as a real-world version of the "Loop" problem described by Michini and How (2012). As explained in their paper, classical IRL algorithms that rely on a global state-based reward model (such as max-margin IRL and max-entropy IRL) completely fail on this problem, due to the periodic nature of the task.

Figure 11a shows the downsampled and discretized data set (black arrows) obtained from four expert trajectories (white lines). For visualization purposes, the discretization block size is chosen as $2\,\mathrm{cm} \times 2\,\mathrm{cm}$, giving rise to a total of $18 \times 24 = 432$ states. As in the top row of Figure 8 (ddBNIRL-S), the coloring of the background indicates the learned partitioning structure, computed from a low-temperature posterior sample. We observe that the found state clusters clearly reveal the modular structure of the task, providing an intuitive and interpretable explanation of the data. However, although the induced policy



(a) expert data & sample partitioning

(b) MAP policy estimate

(c) uncertainty estimate

(d) final predictive model

Figure 11: Results of ddBNIRL-S on the "Cycle" task. (a) Raw measurements (white lines) and discretized demonstration data (black arrows). The coloring of the background indicates a partitioning structure obtained from a low-temperature posterior sample. (b) Maximum a posteriori policy estimate. (c) Visualization of the model's prediction uncertainty at all system states, represented by the entropies of the corresponding posterior predictive action distributions. Dark background indicates high uncertainty. (d) Illustration of the final predictive model, comprising both the action information and the prediction uncertainty.

model (Figure 11b) smoothly captures the cyclic nature of the task, we cannot expect to obtain trustworthy predictions in the center region of the state space, due to the lack of additional demonstration data that would be required to unveil the expert's true intention in that region. Clearly, a point estimate such as the shown MAP policy cannot reflect this prediction uncertainty since it does not carry any confidence information. Yet, following a Bayesian approach, we can naturally quantify the prediction uncertainty at any query state s^* based on the shape of the corresponding posterior predictive action distribution $p(a^* \mid s^*, \mathcal{D})$. A straightforward option is, for example, to consider the prediction entropy,

defined as

$$H(s^*) \coloneqq \sum_{a^* \in \mathcal{A}} p(a^* \mid s^*, \mathcal{D}) \log p(a^* \mid s^*, \mathcal{D}).$$

In order to obtain an unbiased approximation of the true *non-tempered* predictive distribution $p(a^* \mid s^*, \mathcal{D})$, we run a second Gibbs chain with unaltered temperature in parallel to the tempered chain. The resulting entropy estimates are summarized in an uncertainty map (Figure 11c), which we overlaid on the original prediction result to produce the final figure shown at the bottom right. Note that the obtained posterior uncertainty information of the model can be further used in an active learning setting, as demonstrated in Section 5.4.

5.3.2 Temporal Partitioning

Next, we turn our attention to the ddBNIRL-T model, which we test against the vanilla BNIRL approach. For this purpose, we consider the full collection of tasks shown in the supplementary video, which comprises different time-dependent expert behaviors of varying complexity. In order to obtain a quantitative performance measure for our evaluation, we conducted a manual segmentation of all recorded trajectories, thereby creating a set of ground truth subgoal labels for all observed decision times. The result of this segmentation step is depicted in the appendix (Figure 14, center column). Note that the ground truth subgoals are assumed immediately at the ends of the corresponding segments.

The left and right column of Figure 14 show, respectively, the partitioning structures found by BNIRL and ddBNIRL-T, based on a uniform subgoal prior distribution with support at the visited states. The underlying state discretization block size is chosen as $1\,\mathrm{cm} \times 1\,\mathrm{cm}$, as indicated by the regular grid in the background. A simple visual comparison of the learned structures reveals the clear superiority of ddBNIRL-T over vanilla BNIRL on this problem set.

For our quantitative comparison, we consider the instantaneous subgoal localization errors of the two models over the entire course of a demonstration (Figure 12). Herein, the instantaneous localization error for a given state-action pair is measured in terms of the Euclidean distance between the grid location of the ground truth subgoal associated with the pair and the corresponding subgoal location predicted by the model. Note that the predictions of both models are based on the *entire* trajectory data of an experiment, considering the full posterior information after completing the demonstration. For ddBNIRL-T, which does not directly return a subgoal location estimate but instead provides access to the full subgoal posterior distribution, the error is computed with respect to the MAP subgoal locations $\{\hat{g}_k\}$,

$$\hat{g}_k \coloneqq \operatorname*{arg\,max}_{g_k \in \mathrm{supp}(p_g)} p(g_k \mid \mathcal{D}, \tilde{\mathbf{c}}),$$

using the ddBNIRL-T version of Equation (12) — see note at the beginning of Section 4.

The black dots in the figure indicate the time instants where the ground truth annotations change. At those time instants, we observe significantly increased localization errors for both models, which can be explained by the fact that the ground truth annotation is somewhat subjective around the switching points (see labeling in Figure 14). Also, we notice a comparably high error at the beginning and the end of some trajectories, which stems from the imperfect synchronization between the recording interval and the execution

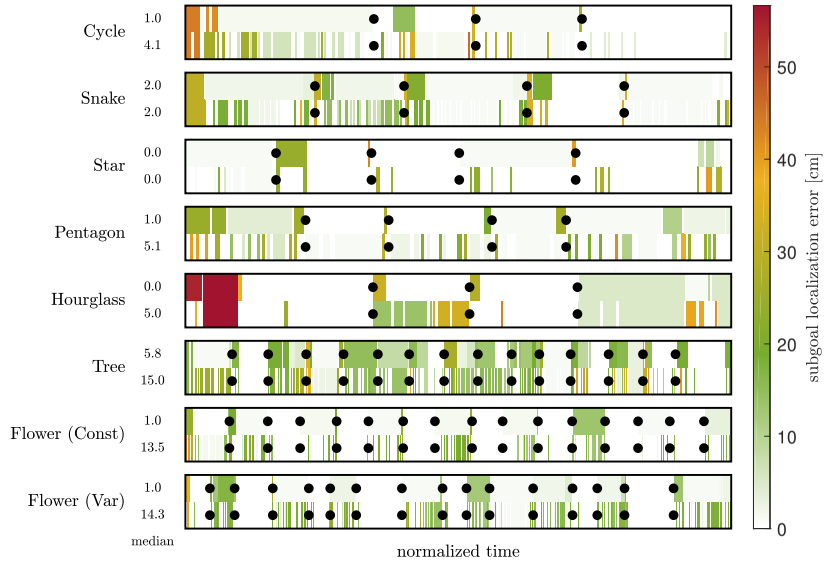


Figure 12: Instantaneous subgoal localization errors of ddBNIRL-T (upper rows) and BNIRL (lower rows) for the eight recorded data sets. The black dots indicate the subgoal switching times in the corresponding ground truth subgoal annotation, depicted in the center column of Figure 14. On average, the localization error of ddBNIRL-T is significantly lower compared to the BNIRL approach, as indicated by the median values shown on the left. For a qualitative comparison of the underlying partitioning structures, see Appendix A.

of the task (recall that we skipped the corresponding data preprocessing step). Hence, to capture the accuracy in a single figure of performance, we consider the median localization error of each time series, as it masks out these outliers and provides a more realistic error quantification than the sample mean. The obtained values are shown next to the error plots in Figure 12, indicating that the ddBNIRL-T localization error is in the range of the discretization interval in most cases. *Compared to BNIRL, the proposed method yields an error reduction of more than 70% on average.*

5.4 Active Learning

In Section 5.3, we saw that the posterior predictive action distribution $p(a^* \mid s^*, \mathcal{D})$ provides a natural way to quantify the prediction uncertainty of our model at any given query state s^*. This offers the opportunity to apply the framework in an active learning setting, since the

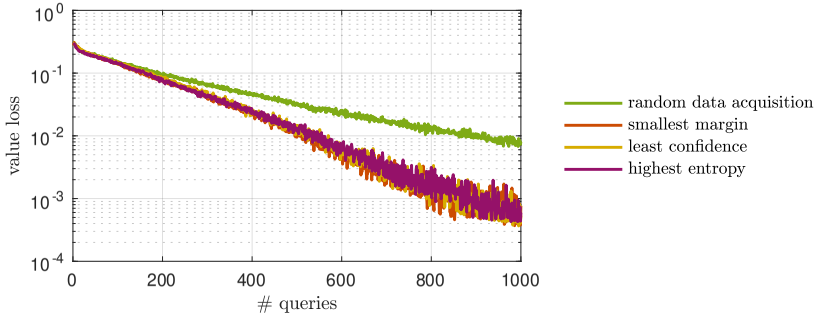


Figure 13: Comparison between random data acquisition and active learning in the random MDP scenario. Shown are the empirical mean value losses of the obtained policy models over the number of data queries, obtained from 200 Monte Carlo runs.

induced uncertainty map (see example in Figure 11c) indicates in which parts of the state space the trained model can process further instructions from the expert most effectively.

To demonstrate the basic procedure, we reconsider the random MDP problem from Section 5.2 in an active learning context, where we compare different active strategies with the previously used random data acquisition scheme. As an initialization for the learning procedure, we request a single state-action pair (s_1, a_1) from the demonstrator, which we store in the initial data set $\mathcal{D}_1 \coloneqq \{(s_1, a_1)\}$. Herein, the state s_1 is drawn uniformly at random from $\mathcal{S}$ and the action $a_1 \sim \pi_E(a \,|\, s_1)$ is generated according to the noisy expert policy $\pi_E : \mathcal{A} \times \mathcal{S} \to [0, 1]$ described in Section 5.2. Continuing from this point, each of the considered active learning algorithms requests a series of subsequent demonstrations $((s_2, a_2), (s_3, a_3), \ldots)$, inducing a sequence of data sets $(\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3, \ldots)$, where the next query state s_{d+1} is chosen according to the specific data acquisition criterion f_{acq} of the algorithm evaluated on the current predictive model,

$$\begin{aligned} \mathcal{D}_{d+1} &= \mathcal{D}_d \cup \{(s_{d+1}, a_{d+1})\} \\ s_{d+1} &= \operatorname*{arg\,max}_{s^* \in \mathcal{S}} f_{\text{acq}}\big[p(a^* \,|\, s^*, \mathcal{D}_d)\big] \\ a_{d+1} &\sim \pi_E(a \,|\, s_{d+1}). \end{aligned}$$

The purpose of the acquisition criterion is to assess the uncertainty of the model at all possible query states, so that the next demonstration can be requested in the high uncertainty region of the state space (see *uncertainty sampling*, Settles, 2010). For our experiment, we consider the following three common choices,

- highest entropy: $f_{\text{acq}}(p) \coloneqq -\sum_{a \in \mathcal{A}} p(a) \log p(a)$,
- least confidence: $f_{\text{acq}}(p) \coloneqq 1 - \max_{a \in \mathcal{A}} p(a)$,
- smallest margin: $f_{\text{acq}}(p) \coloneqq p(\hat{a}_2) - p(\hat{a}_1)$,

where $\hat{a}_1$ and $\hat{a}_2$ denote, respectively, the most likely and second most likely action according to the considered distribution p, i.e., $\hat{a}_1 \coloneqq \arg\max_{a \in \mathcal{A}} p(a)$ and $\hat{a}_2 \coloneqq \arg\max_{a \in \mathcal{A} \setminus \hat{a}_1} p(a)$. At each iteration, we compute the value losses (Equation 20) of the induced policy models and compare them with the corresponding loss obtained from random data acquisition. The resulting curves are delineated in Figure 13. As expected, the learning speed of the model is significantly improved under all active acquisition schemes, which reduces the number of expert demonstrations required to successfully learn the observed task.

6. Conclusion

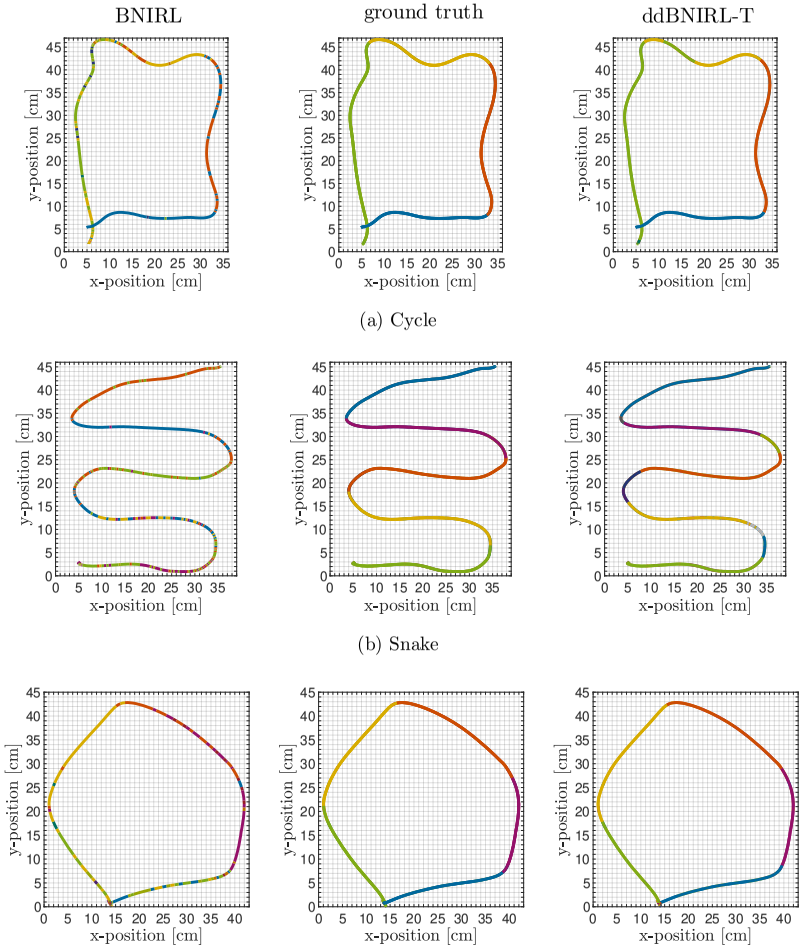
Building upon the principle of Bayesian nonparametric inverse reinforcement learning, we proposed a new framework for data-efficient IRL that leverages the context information of the demonstration set to learn a predictive model of the expert behavior from small amounts of training data. Central to our framework are two model architectures, one designed for learning spatial subgoal plans, the other to capture time-varying intentions. In contrast to the original BNIRL model, both architectures explicitly consider the covariate information contained in the demonstration set, giving rise to predictive models that are inherently robust to demonstration noise. While the original BNIRL model can be recovered as a special case of our framework, the conducted experiments show a drastic improvement over the vanilla BNIRL approach in terms of the achieved subgoal localization accuracy, which stems from both an improved likelihood model and a context-aware clustering of the data. Most notably, our framework outperforms all tested reference methods in the analyzed benchmark scenarios while it additionally captures the full posterior information about the learned subgoal representation. The resulting prediction uncertainty about the expert behavior, reflected by the posterior predictive action distribution, provides a natural basis to apply our method in an active learning setting where the learning system can request additional demonstration data from the expert.

The current limitation of our approach is that both presented architectures require an MDP model with discrete state and action space. While the subgoal principle carries over straightforwardly to continuous metric spaces, the construction of the likelihood model becomes difficult in these environments as it requires knowledge of the optimal state-action value functions for all potential subgoal locations. However, for BNIRL, there exist several ways to approximate the likelihood in these cases (Michini et al., 2013) and the same concepts apply equally to ddBNIRL. Thus, an interesting future study would be to compare the efficacy of both model types on larger problems involving continuous spaces, where it appears even more natural to follow a distance-based approach.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No #713010 (GOALRobots) and No #640554 (SKILLS4ROBOTS).

Appendix A. Robot experiment

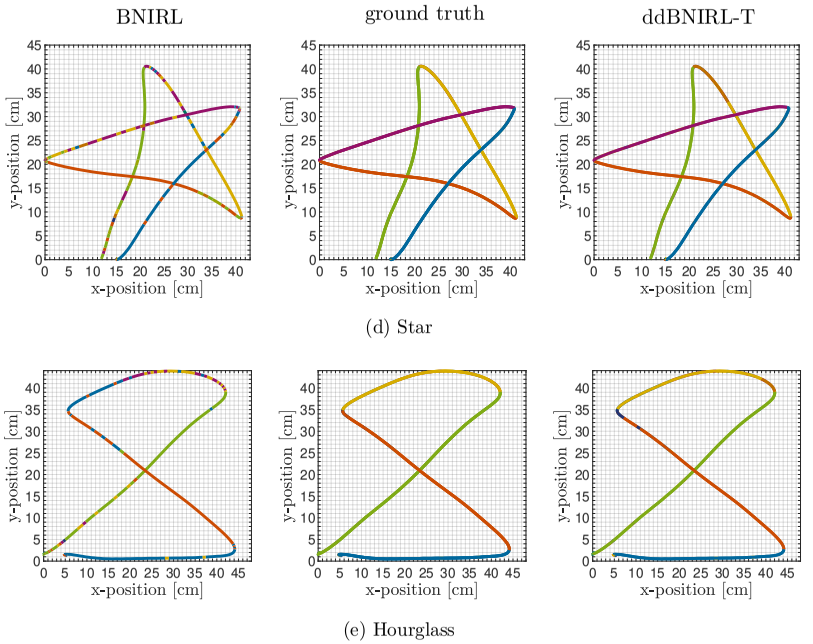


(a) Cycle

(b) Snake

(c) Pentagon

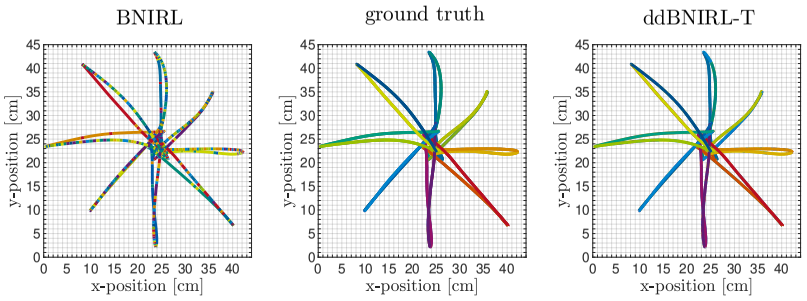
Figure 14: Motion sequences without trajectory crossings, which can be represented using a spatial subgoal pattern.



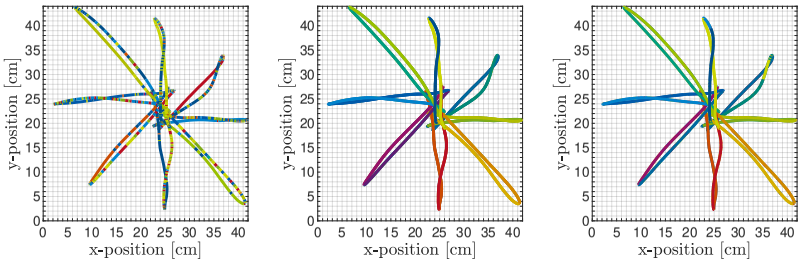
(d) Star

(e) Hourglass

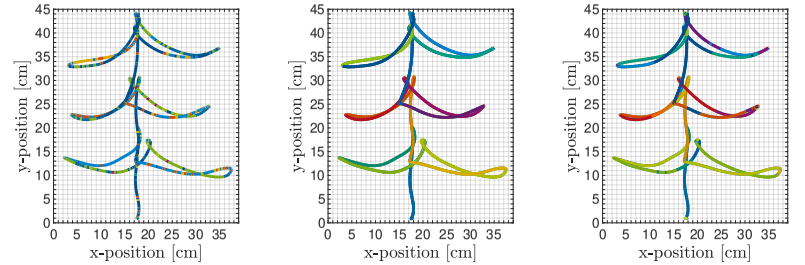
Figure 14 (continued): Motion sequences with few trajectory crossings, requiring a time-varying subgoal representation.



(f) Flower (Const)



(g) Flower (Var)



(h) Tree

Figure 14 (continued): Long motion sequences comprising a large number of sub-patterns with overlapping parts that can be only separated by considering the temporal context. Flower (Const): all strokes are performed with the same absolute velocity. Flower (Var): the individual strokes are performed with alternating velocity.

References

P. Abbeel and A. Y. Ng. Apprenticeship learning via inverse reinforcement learning. In *International Conference on Machine Learning*, page 1, 2004.

M. Al-Emran. Hierarchical reinforcement learning: a survey. *International Journal of Computing and Digital Systems*, 4(2), 2015.

S. V. Albrecht and P. Stone. Autonomous agents modelling other agents: a comprehensive survey and open problems. `arXiv:1709.08071 [cs.AI]`, 2017.

D. J. Aldous. *Exchangeability and Related Topics*. Springer, 1985.

B. D. Argall, S. Chernova, M. Veloso, and B. Browning. A survey of robot learning from demonstration. *Robotics and Autonomous Systems*, 57(5):469–483, 2009.

M. Babeş-Vroman, V. Marivate, K. Subramanian, and M. Littman. Apprenticeship learning about multiple intentions. In *International Conference on Machine Learning*, pages 897–904, 2011.

L. C. Baird. Advantage updating. Technical report, Wright Lab, 1993.

S. Bhatnagar, R. Sutton, M. Ghavamzadeh, and M. Lee. Natural actor-critic algorithms. *Automatica*, 45(11), 2009.

D. M. Blei and P. I. Frazier. Distance dependent Chinese restaurant processes. *Journal of Machine Learning Research*, 12(Nov):2461–2488, 2011.

D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3(Jan):993–1022, 2003.

M. M. Botvinick. Hierarchical reinforcement learning and decision making. *Current Opinion in Neurobiology*, 22(6):956–962, 2012.

S. J. Bradtke and M. O. Duff. Reinforcement learning methods for continuous-time Markov decision problems. In *Advances in Neural Information Processing Systems*, pages 393–400, 1994.

N. Cesa-Bianchi, C. Gentile, G. Neu, and G. Lugosi. Boltzmann exploration done right. In *Advances in Neural Information Processing Systems*, pages 6275–6284, 2017.

J. Choi and K.-E. Kim. Nonparametric Bayesian inverse reinforcement learning for multiple reward functions. In *Advances in Neural Information Processing Systems*, pages 305–313, 2012.

C. Daniel, H. Van Hoof, J. Peters, and G. Neumann. Probabilistic inference for determining options in reinforcement learning. *Machine Learning*, 104(2-3):337–357, 2016a.

Christian Daniel, Gerhard Neumann, Oliver Kroemer, and Jan Peters. Hierarchical relative entropy policy search. *Journal of Machine Learning Research*, 17(1):3190–3239, 2016b.

C. Dimitrakakis and C. A. Rothkopf. Bayesian multitask inverse reinforcement learning. In *European Workshop on Reinforcement Learning*, pages 273–284, 2011.

N. J. Foti and S. A. Williamson. A survey of non-exchangeable priors for Bayesian nonparametric models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(2):359–371, 2015.

M. Ghavamzadeh, S. Mannor, J. Pineau, and A. Tamar. Bayesian reinforcement learning: a survey. *Foundations and Trends in Machine Learning*, 8(5-6):359–483, 2015.

B. M. Kapron, V. King, and B. Mountjoy. Dynamic graph connectivity in polylogarithmic worst case time. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1131–1142, 2013.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

G. Konidaris, S. Kuindersma, R. Grupen, and A. Barto. Robot learning from demonstration by constructing skill trees. *International Journal of Robotics Research*, 31(3):360–375, 2012.

S. Krishnan, A. Garg, R. Liaw, L. Miller, F. T. Pokorny, and K. Goldberg. HIRL: hierarchical inverse reinforcement learning for long-horizon tasks with delayed rewards. `arXiv:1604.06508 [cs.RO]`, 2016.

S. Levine, Z. Popovic, and V. Koltun. Nonlinear inverse reinforcement learning with Gaussian processes. In *Advances in Neural Information Processing Systems*, pages 19–27, 2011.

R. Lioutikov, G. Neumann, G. Maeda, and J. Peters. Learning movement primitive libraries through probabilistic segmentation. *International Journal of Robotics Research*, 36(8): 879–894, 2017.

M. L. Littman, T. L. Dean, and L. P. Kaelbling. On the complexity of solving Markov decision problems. In *Conference on Uncertainty in Artificial Intelligence*, pages 394–402, 1995.

B. Michini and J. P. How. Bayesian nonparametric inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163, 2012.

B. Michini, M. Cutler, and J. P. How. Scalable reward learning from demonstration. In *IEEE International Conference on Robotics and Automation*, pages 303–308, 2013.

B. Michini, T. J. Walsh, A.-A. Agha-Mohammadi, and J. P. How. Bayesian nonparametric reward learning from demonstration. *IEEE Transactions on Robotics*, 31(2):369–386, 2015.

G. Neu and C. Szepesvári. Apprenticeship learning using inverse reinforcement learning and gradient methods. In *Conference on Uncertainty in Artificial Intelligence*, pages 295–302, 2007.

A. Y. Ng and S. J. Russell. Algorithms for inverse reinforcement learning. In *International Conference on Machine Learning*, pages 663–670, 2000.

A. Y. Ng, D. Harada, and S. Russell. Policy invariance under reward transformations: Theory and application to reward shaping. In *International Conference on Machine Learning*, pages 278–287, 1999.

Q. P. Nguyen, B. K. H. Low, and P. Jaillet. Inverse reinforcement learning with locally consistent reward functions. In *Advances in Neural Information Processing Systems*, pages 1747–1755, 2015.

S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto. Learning and generalization of complex tasks from unstructured demonstrations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5239–5246, 2012.

A. Panella and P Gmytrasiewicz. Interactive POMDPs with finite-state models of other agents. *Autonomous Agents and Multi-Agent Systems*, pages 861–904, 2017.

M. L. Puterman. *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley & Sons, Inc., 1994.

D. Ramachandran and E. Amir. Bayesian inverse reinforcement learning. *International Joint Conference on Artificial Intelligence*, pages 2586–2591, 2007.

P. Ranchod, B. Rosman, and G. Konidaris. Nonparametric Bayesian reward segmentation for skill discovery using inverse reinforcement learning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 471–477, 2015.

G. O. Roberts and S. K. Sahu. Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 59(2):291–317, 1997.

C. A. Rothkopf and C. Dimitrakakis. Preference elicitation and inverse reinforcement learning. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 34–48, 2011.

E. Rueckert, G. Neumann, M. Toussaint, and W. Maass. Learned graphical models for probabilistic planning provide a new class of movement primitives. *Frontiers in Computational Neuroscience*, 6:97, 2013.

S. Schaal, J. Peters, J. Nakanishi, and A. Ijspeert. Learning movement primitives. *Robotics Research*, pages 561–572, 2005.

B. Settles. Active learning literature survey. Technical report, University of Wisconsin-Madison, 2010.

Ö Şimşek, A. P. Wolfe, and A. G. Barto. Identifying useful subgoals in reinforcement learning by local graph partitioning. In *International Conference on Machine Learning*, pages 816–823, 2005.

A. Šošić, A. M. Zoubir, and H. Koeppl. Inverse reinforcement learning via nonparametric subgoal modeling. In *AAAI Spring Symposium on Data-Efficient Reinforcement Learning*, 2018a.

A. Šošić, A. M. Zoubir, and H. Koeppl. A Bayesian approach to policy recognition and state representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1295–1308, 2018b.

M. Stolle and D. Precup. Learning options in reinforcement learning. In *International Symposium on Abstraction, Reformulation, and Approximation*, pages 212–223, 2002.

A. Surana and K. Srivastava. Bayesian nonparametric inverse reinforcement learning for switched markov decision processes. In *IEEE International Conference on Learning and Applications*, pages 47–54, 2014.

R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, 1998.

R. S. Sutton, D. Precup, and S. Singh. Between MDPs and semi-MDPs: a framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1):181–211, 1999.

M. Tamassia, F. Zambetta, W. Raffe, and X. Li. Learning options for an MDP from demonstrations. In *Australasian Conference on Artificial Life and Computational Intelligence*, pages 226–242, 2015.

H. M. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, 1984.

A. Tewari and P. L. Bartlett. Optimistic linear programming gives logarithmic regret for irreducible MDPs. In *Advances in Neural Information Processing Systems*, pages 1505–1512, 2008.

J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo. Evaluating bag-of-visual-words representations in scene classification. In *International Workshop on Multimedia Information Retrieval*, pages 197–206, 2007.

S. Zhifei and E. M. Joo. A survey of inverse reinforcement learning techniques. *International Journal of Intelligent Computing and Cybernetics*, 5(3):293–311, 2012.

B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey. Maximum entropy inverse reinforcement learning. In *AAAI Conference on Artificial Intelligence*, pages 1433–1438, 2008.

Journal of Machine Learning Research 19 (2018) 1-57 Submitted 4/18; Published 11/18

The Implicit Bias of Gradient Descent on Separable Data

Daniel Soudry DANIEL.SOUDRY@GMAIL.COM
Elad Hoffer ELAD.HOFFER@GMAIL.COM
Mor Shpigel Nacson MOR.SHPIGEL@GMAIL.COM
Department of Electrical Engineering,Technion
Haifa, 320003, Israel

Suriya Gunasekar SURIYA@TTIC.EDU
Nathan Srebro NATI@TTIC.EDU
Toyota Technological Institute at Chicago
Chicago, Illinois 60637, USA

Editor: Leon Bottou

Abstract

We examine gradient descent on unregularized logistic regression problems, with homogeneous linear predictors on linearly separable datasets. We show the predictor converges to the direction of the max-margin (hard margin SVM) solution. The result also generalizes to other monotone decreasing loss functions with an infimum at infinity, to multi-class problems, and to training a weight layer in a deep network in a certain restricted setting. Furthermore, we show this convergence is very slow, and only logarithmic in the convergence of the loss itself. This can help explain the benefit of continuing to optimize the logistic or cross-entropy loss even after the training error is zero and the training loss is extremely small, and, as we show, even if the validation loss increases. Our methodology can also aid in understanding implicit regularization in more complex models and with other optimization methods.

Keywords: gradient descent, implicit regularization, generalization, margin, logistic regression

1. Introduction

It is becoming increasingly clear that implicit biases introduced by the optimization algorithm play a crucial role in deep learning and in the generalization ability of the learned models (Neyshabur et al., 2014, 2015; Zhang et al., 2017; Keskar et al., 2017; Neyshabur et al., 2017; Wilson et al., 2017). In particular, minimizing the training error, without explicit regularization, over models with more parameters and capacity than the number of training examples, often yields good generalization. This is despite the fact that the empirical optimization problem being highly underdetermined. That is, there are many global minima of the training objective, most of which will not generalize well, but the optimization algorithm (*e.g.* gradient descent) biases us toward a particular minimum that *does* generalize well. Unfortunately, we still do not have a good understanding of the biases introduced by different optimization algorithms in different situations.

We do have an understanding of the implicit regularization introduced by early stopping of stochastic methods or, at an extreme, of one-pass (no repetition) stochastic gradient descent (Hardt et al., 2016). However, as discussed above, in deep learning we often benefit from implicit bias even when optimizing the training error to convergence (without early stopping) using stochastic or batch methods. For loss functions with attainable, finite minimizers, such as the squared loss, we have some understanding of this: in particular, when minimizing an underdetermined least squares problem using gradient descent starting from the origin, it can be shown that we will converge to the minimum Euclidean norm solution. However, the logistic loss, and its generalization the cross-entropy loss which is often used in deep learning, do not admit finite minimizers on separable problems. Instead, to drive the loss toward zero and thus minimize it, the norm of the predictor must diverge toward infinity.

Do we still benefit from implicit regularization when minimizing the logistic loss on separable data? Clearly the norm of the predictor itself is not minimized, since it grows to infinity. However, for prediction, only the direction of the predictor, *i.e.* the normalized $\mathbf{w}(t)/\left\|\mathbf{w}(t)\right\|$, is important. How does $\mathbf{w}(t)/\left\|\mathbf{w}(t)\right\|$ behave as $t\to\infty$ when we minimize the logistic (or similar) loss using gradient descent on separable data, *i.e.*, when it is possible to get zero misclassification error and thus drive the loss to zero?

In this paper, we show that even without any explicit regularization, for all linearly separable datasets, when minimizing logistic regression problems using gradient descent, we have that $\mathbf{w}(t)/\left\|\mathbf{w}(t)\right\|$ converges to the L_2 maximum margin separator, *i.e.* to the solution of the hard margin SVM for homogeneous linear predictors. This happens even though neither the norm $\|\mathbf{w}\|$, nor the margin constraint, are part of the objective or explicitly introduced into optimization. More generally, we show the same behavior for generalized linear problems with any smooth, monotone strictly decreasing, lower bounded loss with an exponential tail. Furthermore, we characterize the rate of this convergence, and show that it is rather slow, wherein for almost all datasets, the distance to the max-margin predictor decreasing only as $O(1/\log(t))$, and in some degenerate datasets, the rate further slows down to $O(\log\log(t)/\log(t))$. This explains why the predictor continues to improve even when the training loss is already extremely small. We emphasize that this bias is specific to gradient descent, and changing the optimization algorithm, *e.g.* using adaptive learning rate methods such as ADAM (Kingma and Ba, 2015), changes this implicit bias.

2. Main Results

Consider a dataset $\{\mathbf{x}_n, y_n\}_{n=1}^N$, with $\mathbf{x}_n\in\mathbb{R}^d$ and binary labels $y_n\in\{-1,1\}$. We analyze learning by minimizing an empirical loss of the form

$$\mathcal{L}\left(\mathbf{w}\right)=\sum_{n=1}^{N}\ell\left(y_n\mathbf{w}^\top\mathbf{x}_n\right). \tag{1}$$

where $\mathbf{w}\in\mathbb{R}^d$ is the weight vector. To simplify notation, we assume that all the labels are positive: $\forall n:\ y_n=1$ — this is true without loss of generality, since we can always re-define $y_n\mathbf{x}_n$ as $\mathbf{x}_n$.

We are particularly interested in problems that are linearly separable, and the loss is smooth strictly decreasing and non-negative:

Assumption 1 *The dataset is linearly separable:* $\exists\mathbf{w}_*$ *such that* $\forall n:\ \mathbf{w}_*^\top\mathbf{x}_n>0$.

Assumption 2 $\ell\left(u\right)$ *is a positive, differentiable, monotonically decreasing to zero*[1]*, (so* $\forall u:\ \ell\left(u\right)>0, \ell'\left(u\right)<0$, $\lim_{u\to\infty}\ell\left(u\right)=\lim_{u\to\infty}\ell'\left(u\right)=0$*), a* β*-smooth function,* i.e. *its derivative is* β*-Lipshitz and* $\lim_{u\to-\infty}\ell'\left(u\right)\neq 0$.

1. The requirement of non-negativity and that the loss asymptotes to zero is purely for convenience. It is enough to require the loss is monotone decreasing and bounded from below. Any such loss asymptotes to some constant, and is thus equivalent to one that satisfies this assumption, up to a shift by that constant.

Assumption 2 includes many common loss functions, including the logistic, exp-loss[2] and probit losses. Assumption 2 implies that $\mathcal{L}(\mathbf{w})$ is a $\beta\sigma_{\max}^2(\mathbf{X})$-smooth function, where $\sigma_{\max}(\mathbf{X})$ is the maximal singular value of the data matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$.

Under these conditions, the infimum of the optimization problem is zero, but it is not attained at any finite $\mathbf{w}$. Furthermore, no finite critical point $\mathbf{w}$ exists. We consider minimizing eq. 1 using Gradient Descent (GD) with a fixed learning rate η, *i.e.,* with steps of the form:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta\nabla\mathcal{L}(\mathbf{w}(t)) = \mathbf{w}(t) - \eta\sum_{n=1}^{N}\ell'\left(\mathbf{w}(t)^\top\mathbf{x}_n\right)\mathbf{x}_n. \tag{2}$$

We do not require convexity. Under Assumptions 1 and 2, gradient descent converges to the global minimum (*i.e.* to zero loss) even without it:

Lemma 1 *Let $\mathbf{w}(t)$ be the iterates of gradient descent (eq. 2) with $\eta < 2\beta^{-1}\sigma_{\max}^{-2}(\mathbf{X})$ and any starting point $\mathbf{w}(0)$. Under Assumptions 1 and 2, we have: (1) $\lim_{t\to\infty}\mathcal{L}(\mathbf{w}(t)) = 0$, (2) $\lim_{t\to\infty}\|\mathbf{w}(t)\| = \infty$, and (3) $\forall n : \lim_{t\to\infty}\mathbf{w}(t)^\top\mathbf{x}_n = \infty$.*

Proof Since the data is linearly separable, $\exists\mathbf{w}_*$ which linearly separates the data, and therefore

$$\mathbf{w}_*^\top\nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}\ell'\left(\mathbf{w}^\top\mathbf{x}_n\right)\mathbf{w}_*^\top\mathbf{x}_n.$$

For any finite $\mathbf{w}$, this sum cannot be equal to zero, as a sum of negative terms, since $\forall n : \mathbf{w}_*^\top\mathbf{x}_n > 0$ and $\forall u : \ell'(u) < 0$. Therefore, there are no finite critical points $\mathbf{w}$, for which $\nabla\mathcal{L}(\mathbf{w}) = \mathbf{0}$. But gradient descent on a smooth loss with an appropriate stepsize is always guaranteed to converge to a critical point: $\nabla\mathcal{L}(\mathbf{w}(t)) \to \mathbf{0}$ (see, *e.g.* Lemma 10 in Appendix A.4, slightly adapted from Ganti (2015), Theorem 2). This necessarily implies that $\|\mathbf{w}(t)\| \to \infty$ while $\forall n : \mathbf{w}(t)^\top\mathbf{x}_n > 0$ for large enough t—since only then $\ell'\left(\mathbf{w}(t)^\top\mathbf{x}_n\right) \to 0$. Therefore, $\mathcal{L}(\mathbf{w}) \to 0$, so GD converges to the global minimum. ■

The main question we ask is: can we characterize the direction in which $\mathbf{w}(t)$ diverges? That is, does the limit $\lim_{t\to\infty}\mathbf{w}(t)/\|\mathbf{w}(t)\|$ always exist, and if so, what is it?

In order to analyze this limit, we will need to make a further assumption on the tail of the loss function:

Definition 2 *A function $f(u)$ has a "tight exponential tail", if there exist positive constants c, a, μ_+, μ_-, u_+ and u_- such that*

$$\forall u > u_+ : f(u) \le c\left(1+\exp\left(-\mu_+ u\right)\right)e^{-au}$$
$$\forall u > u_- : f(u) \ge c\left(1-\exp\left(-\mu_- u\right)\right)e^{-au}.$$

Assumption 3 *The negative loss derivative $-\ell'(u)$ has a tight exponential tail (Definition 2).*

For example, the exponential loss $\ell(u) = e^{-u}$ and the commonly used logistic loss $\ell(u) = \log(1+e^{-u})$ both follow this assumption with $a = c = 1$. We will assume $a = c = 1$ — without loss of generality, since these constants can be always absorbed by re-scaling $\mathbf{x}_n$ and η.

We are now ready to state our main result:

2. The exp-loss does not have a global β smoothness parameter. However, if we initialize with $\eta < 1/\mathcal{L}(\mathbf{w}(0))$ then it is straightforward to show the gradient descent iterates maintain bounded local smoothness.

Theorem 3 *For any dataset which is linearly separable (Assumption 1), any β-smooth decreasing loss function (Assumption 2) with an exponential tail (Assumption 3), any stepsize $\eta < 2\beta^{-1}\sigma_{\max}^{-2}(\mathbf{X})$ and any starting point $\mathbf{w}(0)$, the gradient descent iterates (as in eq. 2) will behave as:*

$$\mathbf{w}(t) = \hat{\mathbf{w}}\log t + \boldsymbol{\rho}(t)\ , \tag{3}$$

where $\hat{\mathbf{w}}$ is the L_2 max margin vector (the solution to the hard margin SVM):

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}}\|\mathbf{w}\|^2 \ \text{ s.t. } \mathbf{w}^\top\mathbf{x}_n \ge 1, \tag{4}$$

and the residual grows at most as $\|\boldsymbol{\rho}(t)\| = O(\log\log(t))$, and so

$$\lim_{t\to\infty}\frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|}.$$

Furthermore, for almost all data sets (all except measure zero), the residual $\rho(t)$ is bounded.

Proof Sketch We first understand intuitively why an exponential tail of the loss entail asymptotic convergence to the max margin vector: Assume for simplicity that $\ell(u) = e^{-u}$ exactly, and examine the asymptotic regime of gradient descent in which $\forall n : \mathbf{w}(t)^\top\mathbf{x}_n \to \infty$, as is guaranteed by Lemma 1. If $\mathbf{w}(t)/\|\mathbf{w}(t)\|$ converges to some limit $\mathbf{w}_\infty$, then we can write $\mathbf{w}(t) = g(t)\mathbf{w}_\infty + \boldsymbol{\rho}(t)$ such that $g(t) \to \infty$, $\forall n : \mathbf{x}_n^\top\mathbf{w}_\infty > 0$, and $\lim_{t\to\infty}\boldsymbol{\rho}(t)/g(t) = 0$. The gradient can then be written as:

$$-\nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_n\right)\mathbf{x}_n = \sum_{n=1}^{N}\exp\left(-g(t)\mathbf{w}_\infty^\top\mathbf{x}_n\right)\exp\left(-\boldsymbol{\rho}(t)^\top\mathbf{x}_n\right)\mathbf{x}_n\ . \tag{5}$$

As $g(t) \to \infty$ and the exponents become more negative, only those samples with the largest (*i.e.*, least negative) exponents will contribute to the gradient. These are precisely the samples with the smallest margin $\operatorname{argmin}_n\mathbf{w}_\infty^\top\mathbf{x}_n$, aka the "support vectors". The negative gradient (eq. 5) would then asymptotically become a non-negative linear combination of support vectors. The limit $\mathbf{w}_\infty$ will then be dominated by these gradients, since any initial conditions become negligible as $\|\mathbf{w}(t)\| \to \infty$ (from Lemma 1). Therefore, $\mathbf{w}_\infty$ will also be a non-negative linear combination of support vectors, and so will its scaling $\hat{\mathbf{w}} = \mathbf{w}_\infty/\left(\min_n\mathbf{w}_\infty^\top\mathbf{x}_n\right)$. We therefore have:

$$\hat{\mathbf{w}} = \sum_{n=1}^{N}\alpha_n\mathbf{x}_n \qquad \forall n\ \left(\alpha_n \ge 0 \text{ and } \hat{\mathbf{w}}^\top\mathbf{x}_n = 1\right)\ \text{OR}\ \left(\alpha_n = 0 \text{ and } \hat{\mathbf{w}}^\top\mathbf{x}_n > 1\right) \tag{6}$$

These are precisely the KKT conditions for the SVM problem (eq. 4) and we can conclude that $\hat{\mathbf{w}}$ is indeed its solution and $\mathbf{w}_\infty$ is thus proportional to it.

To prove Theorem 3 rigorously, we need to show that $\mathbf{w}(t)/\|\mathbf{w}(t)\|$ has a limit, that $g(t) = \log(t)$ and to bound the effect of various residual errors, such as gradients of non-support vectors and the fact that the loss is only approximately exponential. To do so, we substitute eq. 3 into the gradient descent dynamics (eq. 2), with $\mathbf{w}_\infty = \hat{\mathbf{w}}$ being the max margin vector and $g(t) = \log t$. We then show that, except when certain degeneracies occur, the increment in the norm of $\boldsymbol{\rho}(t)$ is bounded by $C_1 t^{-\nu}$ for some $C_1 > 0$ and $\nu > 1$, which is a converging series. This happens because the increment in the max margin term, $\hat{\mathbf{w}}\left[\log(t+1) - \log(t)\right] \approx \hat{\mathbf{w}}t^{-1}$, cancels out the dominant t^{-1} term in the gradient $-\nabla\mathcal{L}(\mathbf{w}(t))$ (eq. 5 with $g(t) = \log(t)$ and $\mathbf{w}_\infty^\top\mathbf{x}_n = 1$).

Degenerate and Non-Degenerate Data Sets An earlier conference version of this paper (Soudry et al., 2018) included a partial version of Theorem 3, which only applies to almost all data sets, in which case we can ensure the residual $\rho(t)$ is bounded. This partial statement (for almost all data sets) is restated and proved as Theorem 9 in Appendix A. It applies, *e.g.* with probability one for data sampled from any absolutely continuous distribution. It does not apply in "degenerate" cases where some of the support vectors $\mathbf{x}_n$ (for which $\hat{\mathbf{w}}^\top \mathbf{x}_n = 1$) are associated with dual variables that are zero ($\alpha_n = 0$) in the dual optimum of 4. As we show in Appendix B, this only happens on measure zero data sets. Here, we prove the more general result which applies for all data sets, including degenerate data sets. To do so, in Theorem 13 in Appendix C we provide a more complete characterization of the iterates $\mathbf{w}(t)$ that explicitly specifies all unbounded components even in the degenerate case. We then prove the Theorem by plugging in this more complete characterization and showing that the residual is bounded, thus also establishing Theorem 3.

Parallel Work on the Degenerate Case Following the publication of our initial version, and while preparing this revised version for publication, we learned of parallel work by Ziwei Ji and Matus Telgarsky that also closes this gap. Ji and Telgarsky (2018) provide an analysis of the degenerate case, establishing converges to the max margin predictor by showing that $\left\| \frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} - \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|} \right\| = O\left(\sqrt{\frac{\log\log t}{\log t}}\right)$. Our analysis provides a more precise characterization of the iterates, and also shows the convergence is actually quadratically faster (see Section 3). However, Ji and Telgarsky go even further and provide a characterization also when the data is non-separable but $\mathbf{w}(t)$ still goes to infinity.

More Refined Analysis of the Residual In some non-degenerate cases, we can further characterize the asymptotic behavior of $\boldsymbol{\rho}(t)$. To do so, we need to refer to the KKT conditions (eq. 6) of the SVM problem (eq. 4) and the associated support vectors $\mathcal{S} = \operatorname{argmin}_n \hat{\mathbf{w}}^\top \mathbf{x}_n$. We then have the following Theorem, proved in Appendix A:

Theorem 4 *Under the conditions and notation of Theorem 3, for almost all datasets, if in addition the support vectors span the data (*i.e. $\operatorname{rank}(\mathbf{X}_\mathcal{S}) = \operatorname{rank}(\mathbf{X})$*, where $\mathbf{X}_\mathcal{S}$ is a matrix whose columns are only those data points $\mathbf{x}_n$ s.t. $\hat{\mathbf{w}}^\top \mathbf{x}_n = 1$), then $\lim_{t\to\infty} \boldsymbol{\rho}(t) = \hat{\mathbf{w}}$, where $\hat{\mathbf{w}}$ is a solution to*

$$\forall n \in \mathcal{S}: \ \eta \exp\left(-\mathbf{x}_n^\top \hat{\mathbf{w}}\right) = \alpha_n \tag{7}$$

Analogies with Boosting Perhaps most similar to our study is the line of work on understanding AdaBoost in terms its implicit bias toward large L_1-margin solutions, starting with the seminal work of Schapire et al. (1998). Since AdaBoost can be viewed as coordinate descent on the exponential loss of a linear model, these results can be interpreted as analyzing the bias of coordinate descent, rather then gradient descent, on a monotone decreasing loss with an exact exponential tail. Indeed, with small enough step sizes, such a coordinate descent procedure does converge precisely to the maximum L_1-margin solution (Zhang et al., 2005; Telgarsky, 2013). In fact, Telgarsky (2013) also generalizes these results to other losses with tight exponential tails, similar to the class of losses we consider here.

Also related is the work of Rosset et al. (2004). They considered the regularization path $\mathbf{w}_\lambda = \arg\min \mathcal{L}(\mathbf{w}) + \lambda \|\mathbf{w}\|_p^p$ for similar loss functions as we do, and showed that $\lim_{\lambda\to 0} \mathbf{w}_\lambda / \|\mathbf{w}_\lambda\|_p$ is proportional to the maximum L_p margin solution. That is, they showed how adding infinitesimal L_p (*e.g.* L_1 and L_2) regularization to logistic-type losses gives rise to the corresponding max-margin predictor.[3] However, Rosset et al. do not consider the effect of the optimization algorithm, and instead add explicit regularization. Here we are specifically interested in the bias implied by the algorithm *not* by adding (even infinitesimal) explicit regularization. We see that coordinate descent gives rise to the max L_1 margin predictor, while gradient descent gives rise to the max L_2 norm predictor. In Section 4.3 and in follow-up work (Gunasekar et al., 2018) we discuss also other optimization algorithms, and their implied biases.

Non-homogeneous linear predictors In this paper we focused on homogeneous linear predictors of the form $\mathbf{w}^\top \mathbf{x}$, similarly to previous works (*e.g.*, Rosset et al. (2004); Telgarsky (2013)). Specifically, we did not have the common intercept term: $\mathbf{w}^\top \mathbf{x} + b$. One may be tempted to introduce the intercept in the usual way, *i.e.*, by extending all the input vectors $\mathbf{x}_n$ with an additional $'1'$ component. In this extended input space, naturally, all our results hold. Therefore, we converge in direction to the L_2 max margin solution (eq. 4) in the extended space. However, if we translate this solution to the original $\mathbf{x}$ space we obtain

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \|\mathbf{w}\|^2 + b^2 \text{ s.t. } \mathbf{w}^\top \mathbf{x}_n + b \geq 1,$$

which is not the L_2 max margin (SVM) solution

$$\operatorname*{argmin}_{\mathbf{w}\in\mathbb{R}^d, b\in\mathbb{R}} \|\mathbf{w}\|^2 \text{ s.t. } \mathbf{w}^\top \mathbf{x}_n + b \geq 1,$$

where we do not have a b^2 penalty in the objective.

3. Implications: Rates of convergence

The solution in eq. 3 implies that $\mathbf{w}(t) / \|\mathbf{w}(t)\|$ converges to the normalized max margin vector $\hat{\mathbf{w}} / \|\hat{\mathbf{w}}\|$. Moreover, this convergence is very slow— logarithmic in the number of iterations. Specifically, our results imply the following tight rates of convergence:

Theorem 5 *Under the conditions and notation of Theorem 3, for any linearly separable data set, the normalized weight vector converges to the normalized max margin vector in L_2 norm*

$$\left\| \frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} - \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|} \right\| = O\left(\frac{\log\log t}{\log t}\right), \tag{8}$$

with this rate improving to $O(1/\log(t))$ for almost every dataset; and in angle

$$1 - \frac{\mathbf{w}(t)^\top \hat{\mathbf{w}}}{\|\mathbf{w}(t)\| \|\hat{\mathbf{w}}\|} = O\left(\left(\frac{\log\log t}{\log t}\right)^2\right), \tag{9}$$

with this rate improving to $O(1/\log^2(t))$ for almost every dataset; and the margin converges as

$$\frac{1}{\|\hat{\mathbf{w}}\|} - \frac{\min_n \mathbf{x}_n^\top \mathbf{w}(t)}{\|\mathbf{w}(t)\|} = O\left(\frac{1}{\log t}\right). \tag{10}$$

On the other hand, the loss itself decreases as

$$\mathcal{L}(\mathbf{w}(t)) = O\left(\frac{1}{t}\right). \tag{11}$$

3. In contrast, with non-vanishing regularization (*i.e.*, $\lambda > 0$), $\arg\min_\mathbf{w} \mathcal{L}(\mathbf{w}) + \lambda \|\mathbf{w}\|_p^p$ is generally *not* a max margin solution.

All the rates in the above Theorem are a direct consequence of Theorem 3, except for avoiding the $\log\log t$ factor for the degenerate cases in eq. 10 and eq. 11 (*i.e.*, establishing that the rates $1/\log t$ and $1/t$ always hold)—this additional improvement is a consequence of the more complete characterization of Theorem 13. Full details are provided in Appendix D. In this appendix, we also provide a simple construction showing all the rates in Theorem 5 are tight (except possibly for the $\log\log t$ factors).

The sharp contrast between the tight logarithmic and $1/t$ rates in Theorem 5 implies that the convergence of $\mathbf{w}(t)$ to the max-margin $\hat{\mathbf{w}}$ can be logarithmic in the loss itself, and we might need to wait until the loss is exponentially small in order to be close to the max-margin solution. This can help explain why continuing to optimize the training loss, even after the training error is zero and the training loss is extremely small, still improves generalization performance—our results suggests that the margin could still be improving significantly in this regime.

A numerical illustration of the convergence is depicted in Figure 1. As predicted by the theory, the norm $\|\mathbf{w}(t)\|$ grows logarithmically (note the semi-log scaling), and $\mathbf{w}(t)$ converges to the max-margin separator, but only logarithmically, while the loss itself decreases very rapidly (note the log-log scaling).

An important practical consequence of our theory, is that although the margin of $\mathbf{w}(t)$ keeps improving, and so we can expect the population (or test) misclassification error of $\mathbf{w}(t)$ to improve for many datasets, the same cannot be said about the expected population loss (or test loss)! At the limit, the direction of $\mathbf{w}(t)$ will converge toward the max margin predictor $\hat{\mathbf{w}}$. Although $\hat{\mathbf{w}}$ has zero training error, it will not generally have zero misclassification error on the population, or on a test or a validation set. Since the norm of $\mathbf{w}(t)$ will increase, if we use the logistic loss or any other convex loss, the loss incurred on those misclassified points will also increase. More formally, consider the logistic loss $\ell(u) = \log(1+e^{-u})$ and define also the hinge-at-zero loss $h(u) = \max(0,-u)$. Since $\hat{\mathbf{w}}$ classifies all training points correctly, we have that on the training set $\sum_{n=1}^{N} h(\hat{\mathbf{w}}^\top \mathbf{x}_n) = 0$. However, on the population we would expect some errors and so $\mathbb{E}[h(\hat{\mathbf{w}}^\top \mathbf{x})] > 0$. Since $\mathbf{w}(t) \approx \hat{\mathbf{w}} \log t$ and $\ell(\alpha u) \to \alpha h(u)$ as $\alpha \to \infty$, we have:

$$\mathbb{E}[\ell(\mathbf{w}(t)^\top \mathbf{x})] \approx \mathbb{E}[\ell((\log t)\hat{\mathbf{w}}^\top \mathbf{x})] \approx (\log t)\mathbb{E}[h(\hat{\mathbf{w}}^\top \mathbf{x})] = \Omega(\log t). \tag{12}$$

That is, the population loss increases logarithmically while the margin and the population misclassification error improve. Roughly speaking, the improvement in misclassification does not out-weight the increase in the loss of those points still misclassified.

The increase in the test loss is practically important because the loss on a validation set is frequently used to monitor progress and decide on stopping. Similar to the population loss, the validation loss will increase logarithmically with t, if there is at least one sample in the validation set which is classified incorrectly by the max margin vector (since we would not expect zero validation error). More precisely, as a direct consequence of Theorem 3 (as shown on Appendix D):

Corollary 6 *Let ℓ be the logistic loss, and $\mathcal{V}$ be an independent validation set, for which $\exists \mathbf{x} \in \mathcal{V}$ such that $\mathbf{x}^\top \hat{\mathbf{w}} < 0$. Then the validation loss increases as*

$$\mathcal{L}_{\text{val}}\left(\mathbf{w}\left(t\right)\right) = \sum_{\mathbf{x}\in\mathcal{V}} \ell\left(\mathbf{w}\left(t\right)^\top \mathbf{x}\right) = \Omega(\log(t)).$$

This behavior might cause us to think we are over-fitting or otherwise encourage us to stop the optimization. However, this increase does not actually represent the model getting worse, merely

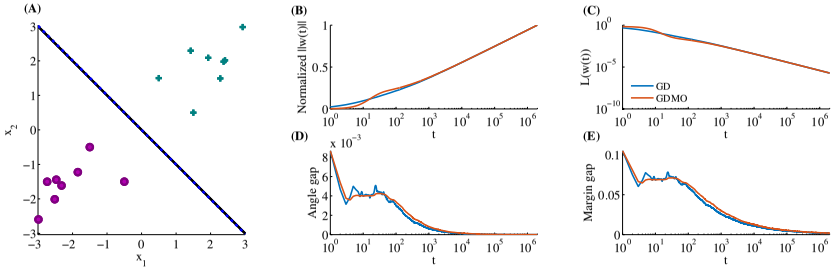


Figure 1: Visualization of or main results on a synthetic dataset in which the L_2 max margin vector $\hat{\mathbf{w}}$ is precisely known. **(A)** The dataset (positive and negatives samples ($y = \pm 1$) are respectively denoted by $'+'$ and $'\circ'$), max margin separating hyperplane (black line), and the asymptotic solution of GD (dashed blue). For both GD and GD with momentum (GDMO), we show: **(B)** The norm of $\mathbf{w}\left(t\right)$, normalized so it would equal to 1 at the last iteration, to facilitate comparison. As expected (eq. 3), the norm increases logarithmically; **(C)** the training loss. As expected, it decreases as t^{-1} (eq. 11); and **(D&E)** the angle and margin gap of $\mathbf{w}\left(t\right)$ from $\hat{\mathbf{w}}$ (eqs. 9 and 10). As expected, these are logarithmically decreasing to zero. **Implementation details:** The dataset includes four support vectors: $\mathbf{x}_1 = (0.5, 1.5)\,, \mathbf{x}_2 = (1.5, 0.5)$ with $y_1 = y_2 = 1$, and $\mathbf{x}_3 = -\mathbf{x}_1$, $\mathbf{x}_4 = -\mathbf{x}_2$ with $y_3 = y_4 = -1$ (the L_2 normalized max margin vector is then $\hat{\mathbf{w}} = (1,1)/\sqrt{2}$ with margin equal to $\sqrt{2}$), and 12 other random datapoints (6 from each class), that are not on the margin. We used a learning rate $\eta = 1/\sigma^2_{\max}\left(\mathbf{X}\right)$, where $\sigma^2_{\max}\left(\mathbf{X}\right)$ is the maximal singular value of $\mathbf{X}$, momentum $\gamma = 0.9$ for GDMO, and initialized at the origin.

$\|\mathbf{w}(t)\|$ getting larger, and in fact the model might be getting better (increasing the margin and possibly decreasing the error rate).

4. Extensions

4.1. Multi-Class Classification with Cross-Entropy Loss

So far, we have discussed the problem of binary classification, but in many practical situations we have more then two classes. For multi-class problems, the labels are the class indices $y_n \in [K] \triangleq \{1, \ldots, K\}$ and we learn a predictor $\mathbf{w}_k$ for each class $k \in [K]$. A common loss function in multi-class classification is the following cross-entropy loss with a softmax output, which is a generalization of the logistic loss:

$$\mathcal{L}\left(\{\mathbf{w}_k\}_{k\in[K]}\right) = -\sum_{n=1}^{N} \log\left(\frac{\exp\left(\mathbf{w}_{y_n}^\top \mathbf{x}_n\right)}{\sum_{k=1}^{K}\exp\left(\mathbf{w}_k^\top \mathbf{x}_n\right)}\right) \tag{13}$$

What do the linear predictors $\mathbf{w}_k(t)$ converge to if we minimize the cross-entropy loss by gradient descent on the predictors? In Appendix E we analyze this problem for separable data, and show that

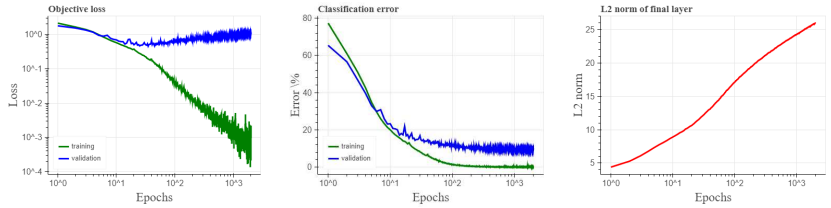


Figure 2: Training of a convolutional neural network on CIFAR10 using stochastic gradient descent with constant learning rate and momentum, softmax output and a cross entropy loss, where we achieve 8.3% final validation error. We observe that, approximately: (1) The training loss decays as a t^{-1}, (2) the L_2 norm of last weight layer increases logarithmically, (3) after a while, the validation loss starts to increase, and (4) in contrast, the validation (classification) error slowly improves.

again, the predictors diverge to infinity and the loss converges to zero. Furthermore, we prove the following Theorem:

Theorem 7 *For almost all multiclass datasets (*i.e.*, except for a measure zero) which are linearly separable (i.e. the constraints in eq. 15 below are feasible), any starting point* $\mathbf{w}(0)$ *and any small enough stepsize, the iterates of gradient descent on 13 will behave as:*

$$\mathbf{w}_k(t) = \hat{\mathbf{w}}_k \log(t) + \boldsymbol{\rho}_k(t), \tag{14}$$

where the residual $\boldsymbol{\rho}_k(t)$ *is bounded and* $\hat{\mathbf{w}}_k$ *is the solution of the K-class SVM:*

$$\operatorname{argmin}_{\mathbf{w}_1,\ldots,\mathbf{w}_k} \sum_{k=1}^{K} ||\mathbf{w}_k||^2 \; s.t. \; \forall n, \forall k \neq y_n : \mathbf{w}_{y_n}^\top \mathbf{x}_n \geq \mathbf{w}_k^\top \mathbf{x}_n + 1. \tag{15}$$

4.2. Deep networks

So far we have only considered linear prediction. Naturally, it is desirable to generalize our results also to non-linear models and especially multi-layer neural networks.

Even without a formal extension and description of the precise bias, our results already shed light on how minimizing the cross-entropy loss with gradient descent can have a margin maximizing effect, how the margin might improve only logarithmically slow, and why it might continue to improve even as the validation loss increases. These effects are demonstrated in Figure 2 and Table 1 which portray typical training of a convolutional neural network using unregularized gradient descent[4]. As can be seen, the norm of the weight increases, but the validation error continues decreasing, albeit very slowly (as predicted by the theory), even after the training error is zero and the training loss is extremely small. We can now understand how even though the loss is already extremely small, some sort of margin might be gradually improving as we continue optimizing. We can also observe how the validation loss increases despite the validation error decreasing, as discussed in Section 3.

4. Code available here: `https://github.com/paper-submissions/MaxMargin`

Epoch	50	100	200	400	2000	4000
L_2 norm	13.6	16.5	19.6	20.3	25.9	27.54
Train loss	0.1	0.03	0.02	0.002	10^{-4}	$3 \cdot 10^{-5}$
Train error	4%	1.2%	0.6%	0.07%	0%	0%
Validation loss	0.52	0.55	0.77	0.77	1.01	1.18
Validation error	12.4%	10.4%	11.1%	9.1%	8.92%	8.9%

Table 1: Sample values from various epochs in the experiment depicted in Fig. 2.

As an initial advance toward tackling deep network, we can point out that for several special cases, our results may be directly applied to multi-layered networks. First, somewhat trivially, our results may be applied directly to the last weight layer of a neural network if the last hidden layer becomes fixed and linearly separable after a certain number of iterations. This can become true, either approximately, if the input to the last hidden layer is normalized (*e.g.*, using batch norm), or exactly, if the last hidden layer is quantized (Hubara et al., 2018).

Second, as we show next, our results may be applied exactly on deep networks if only a single weight layer is being optimized, and, furthermore, after a sufficient number of iterations, the activation units stop switching and the training error goes to zero.

Corollary 8 *We examine a multilayer neural network with component-wise ReLU functions* $f(z) = \max[z, 0]$*, and weights* $\{\mathbf{W}_l\}_{l=1}^{L}$*. Given input* $\mathbf{x}_n$ *and target* $y_n \in \{-1, 1\}$*, the DNN produces a scalar output*

$$u_n = \mathbf{W}_L f\left(\mathbf{W}_{L-1} f\left(\cdots \mathbf{W}_2 f\left(\mathbf{W}_1 \mathbf{x}_n\right)\right)\right)$$

and has loss $\ell(y_n u_n)$*, where* ℓ *obeys assumptions 2 and 3.*

If we optimize a single weight layer $\mathbf{w}_l = \operatorname{vec}\left(\mathbf{W}_l^\top\right)$ *using gradient descent, so that* $\mathcal{L}(\mathbf{w}_l) = \sum_{n=1}^{N} \ell\left(y_n u_n(\mathbf{w}_l)\right)$ *converges to zero, and* $\exists t_0$ *such that* $\forall t > t_0$ *the ReLU inputs do not switch signs, then* $\mathbf{w}_l(t) / \left\|\mathbf{w}_l(t)\right\|$ *converges to*

$$\hat{\mathbf{w}}_l = \operatorname*{argmin}_{\mathbf{w}_l} \|\mathbf{w}_l\|^2 \;\text{ s.t. }\; y_n u_n(\mathbf{w}_l) \geq 1.$$

Proof We examine the output of the network given a single input $\mathbf{x}_n$, for $t > t_0$**.** Since the ReLU inputs do not switch signs, we can write $\mathbf{v}_l$, the output of layer l, as

$$\mathbf{v}_{l,n} = \prod_{m=1}^{l} \mathbf{A}_{m,n} \mathbf{W}_m \mathbf{x}_n \,,$$

where we defined $\mathbf{A}_{l,n}$ for $l < L$ as a diagonal 0-1 matrix, which diagonal is the ReLU slopes at layer l, sample n, and $\mathbf{A}_{L,n} = 1$. Additionally, we define

$$\boldsymbol{\delta}_{l,n} = \mathbf{A}_{l,n} \prod_{m=L}^{l+1} \mathbf{W}_m^\top \mathbf{A}_{m,n} \,;\; \tilde{\mathbf{x}}_{l,n} = \boldsymbol{\delta}_{l,n} \otimes \mathbf{u}_{l-1,n} \,.$$

Using this notation we can write

$$u_n(\mathbf{w}_l) = v_{L,n} = \prod_{m=1}^{L} \mathbf{A}_{m,n} \mathbf{W}_m \mathbf{x}_n = \boldsymbol{\delta}_{l,n}^\top \mathbf{W}_l \mathbf{u}_{l-1,n} = \tilde{\mathbf{x}}_{l,n}^\top \mathbf{w}_l \,. \tag{16}$$

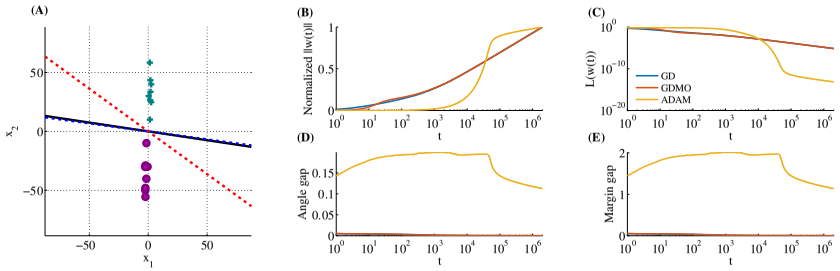


Figure 3: Same as Fig. 1, except we multiplied all x_2 values in the dastaset by 20, and also train using ADAM. The final weight vector produced after $2 \cdot 10^6$ epochs of optimization using ADAM (red dashed line) does not converge to L2 max margin solution (black line), in contrast to GD (blue dashed line), or GDMO.

This implies that

$$\mathcal{L}(\mathbf{w}_l) = \sum_{n=1}^{N} \ell\left(y_n u_n(\mathbf{w}_l)\right) = \sum_{n=1}^{N} \ell\left(y_n \tilde{\mathbf{x}}_{l,n}^\top \mathbf{w}_l\right),$$

which is the same as the original linear problem. Since the loss converges to zero, the dataset $\{\tilde{\mathbf{x}}_{l,n}, y_n\}_{n=1}^N$ must be linearly separable. Applying Theorem 3, and recalling that $u(\mathbf{w}_l) = \tilde{\mathbf{x}}_l^\top \mathbf{w}_l$ from eq. 16, we prove this corollary. ■

Importantly, this case is non-convex, unless we are optimizing the last layer. Note we assumed ReLU functions for simplicity, but this proof can be easily generalized for any other piecewise linear constant activation functions (*e.g.*, leaky ReLU, max-pooling).

Lastly, in a follow-up work (Gunasekar et al., 2018b), given a few additional assumptions, extended our results to linear predictors which can be written as a homogeneous polynomial in the parameters. These results seem to indicate that, in many cases, GD operating on exp-tailed loss with positively homogeneous predictors aims to a specific direction. This is the direction of the max margin predictor minimizing the L_2 norm in the parameter space. It is not yet clear how to generally translate such an implicit bias in the parameter space to the implicit bias in the predictor space — except in special cases, such as deep linear neural nets, as we have shown in (Gunasekar et al., 2018b). Moreover, in non-linear neural nets, there are many equivalent max-margin solutions which minimize the L_2 norm of the parameters. Therefore, it is natural to expect that GD would have additional implicit biases, which select a specific subset of these solutions.

4.3. Other optimization methods

In this paper we examined the implicit bias of gradient descent. Different optimization algorithms exhibit different biases, and understanding these biases and how they differ is crucial to understanding

and constructing learning methods attuned to the inductive biases we expect. Can we characterize the implicit bias and convergence rate in other optimization methods?

In Figure 1 we see that adding momentum does not qualitatively affect the bias induced by gradient descent. In Figure 4 in Appendix F we also repeat the experiment using stochastic gradient descent, and observe a similar asymptotic bias (this was later proved in Nacson et al. (2018)). This is consistent with the fact that momentum, acceleration and stochasticity do not change the bias when using gradient descent to optimize an under determined least squares problem. It would be beneficial, though, to rigorously understand how much we can generalize our result to gradient descent variants, and how the convergence rates might change in these cases.

On the other hand, as an example of how changing the optimization algorithm does change the bias, consider adaptive methods, such as AdaGrad (Duchi et al., 2011) and ADAM (Kingma and Ba, 2015). In Figure 3 we show the predictors obtained by ADAM and by gradient descent on a simple data set. Both methods converge to zero training error solutions. But although gradient descent converges to the L_2 max margin predictor, as predicted by our theory, ADAM does not. The implicit bias of adaptive methods has in fact been a recent topic of interest, with Hoffer et al. (2017) and Wilson et al. (2017) suggesting they lead to worse generalization, and Wilson et al. (2017) providing examples of the differences in the bias for linear regression problems with the squared loss. Can we characterize the bias of adaptive methods for logistic regression problems? Can we characterize the bias of other optimization methods, providing a general understanding linking optimization algorithms with their biases?

In a follow-up paper (Gunasekar et al., 2018) provided initial answers to these questions. Gunasekar et al. (2018) derived a precise characterization of the limit direction of steepest descent for general norms when optimizing the exp-loss, and show that for adaptive methods such as Adagrad the limit direction can depend on the initial point and step size and is thus not as predictable and robust as with non-adaptive methods.

4.4. Other loss functions

In this work we focused on loss functions with exponential tail and observed a very slow, logarithmic convergence of the normalized weight vector to the L_2 max margin direction. A natural question that follows is how does this behavior change with types of loss function tails. Specifically, does the normalized weight vector always converge to the L_2 max margin solution? How is the convergence rate affected? Can we improve the convergence rate beyond the logarithmic rate found in this work?

In a follow-up work Nacson et al. (2018) provided partial answers to these questions. They proved that the exponential tail has the optimal convergence rate, for tails for which $\ell'(u)$ is of the form $\exp(-u^\nu)$ with $\nu > 0.25$. They then conjectured, based on heuristic analysis, that the exponential tail is optimal among all possible tails. Furthermore, they demonstrated that polynomial or heavier tails do not converge to the max margin solution. Lastly, for the exponential loss they proposed a normalized gradient scheme which can significantly improve convergence rate, achieving $O(\log(t)/\sqrt{t})$.

4.5. Matrix Factorization

With multi-layered neural networks in mind, Gunasekar et al. (2017) recently embarked on a study of the implicit bias of under-determined matrix factorization problems, where the *squared loss* of the linear observation of a matrix is minimized by gradient descent on its factorization. Since a

matrix factorization can be viewed as a two-layer network with linear activations, this is perhaps the simplest deep model one can study in full, and can thus provide insight and direction to studying more complex neural networks. Gunasekar et al. conjectured, and provided theoretical and empirical evidence, that gradient descent on the factorization for an under-determined problem converges to the minimum nuclear norm solution, but only if the initialization is infinitesimally close to zero and the step-sizes are infinitesimally small. With finite step-sizes or finite initialization, Gunasekar et al. could not characterize the bias.

The follow-up paper (Gunasekar et al., 2018) studied this same problem with exponential loss instead of squared loss. Under additional assumptions on the asymptotic convergence of update directions and gradient directions, they were able to relate the direction of gradient descent iterates on the factorized parameterization asymptotically to the maximum margin solution with unit nuclear norm. Unlike the case of squared loss, the result for exponential loss are independent of initialization and with only mild conditions on the step size. Here again, we see the asymptotic nature of exponential loss on separable data nullifying the initialization effects thereby making the analysis simpler compared to squared loss.

5. Summary

We characterized the implicit bias induced by gradient descent on homogeneous linear predictors when minimizing smooth monotone loss functions with an exponential tail. This is the type of loss commonly being minimized in deep learning. We can now rigorously understand:

1. How gradient descent, without early stopping, induces implicit L_2 regularization and converges to the maximum L_2 margin solution, when minimizing for binary classification with logistic loss, exp-loss, or other exponential tailed monotone decreasing loss, as well as for multi-class classification with cross-entropy loss. Notably, even though the logistic loss and the exp-loss behave very different on non-separable problems, they exhibit the same behaviour for separable problems. This implies that the non-tail part does not affect the bias. The bias is also independent of the step-size used (as long as it is small enough to ensure convergence) and is also independent on the initialization (unlike for least square problems).

2. The convergence of the direction of gradient descent updates to the maximum L_2 margin solution, however is very slow compared to the convergence of training loss, which explains why it is worthwhile continuing to optimize long after we have zero training error, and even when the loss itself is already extremely small.

3. We should not rely on plateauing of the training loss or on the loss (logistic or exp or cross-entropy) evaluated on a validation data, as measures to decide when to stop. Instead, we should look at the 0–1 error on the validation dataset. We might improve the validation and test errors even when when the decrease in the training loss is tiny and even when the validation loss itself increases.

Perhaps that gradient descent leads to a max L_2 margin solution is not a big surprise to those for whom the connection between L_2 regularization and gradient descent is natural. Nevertheless, we are not familiar with any prior study or mention of this fact, let alone a rigorous analysis and study of how this bias is exact and independent of the initial point and the step-size. Furthermore, we also analyze the rate at which this happens, leading to the novel observations discussed above. Even more

importantly, we hope that our analysis can open the door to further analysis of different optimization methods or in different models, including deep networks, where implicit regularization is not well understood even for least square problems, or where we do not have such a natural guess as for gradient descent on linear problems. Analyzing gradient descent on logistic/cross-entropy loss is not only arguably more relevant than the least square loss, but might also be technically easier.

Acknowledgments

The authors are grateful to J. Lee, and C. Zeno for helpful comments on the manuscript. The research of DS was supported by the Israel Science Foundation (grant No. 31/1031), by the Taub foundation and of NS by the National Science Foundation.

Appendix

Appendix A. Proof of Theorems 3 and 4 for almost every dataset

In the following sub-sections we first prove Theorem 9 below, which is a version of Theorem 3, specialized for almost every dataset. We then prove Theorem 4 (which is already stated for almost every dataset).

Theorem 9 *For almost every dataset which is linearly separable (Assumption 1), any β-smooth decreasing loss function (Assumption 2) with an exponential tail (Assumption 3), any stepsize $\eta < 2\beta^{-1}\sigma_{\max}^{-2}(\mathbf{X})$ and any starting point $\mathbf{w}(0)$, the gradient descent iterates (as in eq. 2) will behave as:*

$$\mathbf{w}(t) = \hat{\mathbf{w}}\log t + \boldsymbol{\rho}(t)\ , \tag{17}$$

where $\hat{\mathbf{w}}$ is the L_2 max margin vector

$$\hat{\mathbf{w}} = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{w}\|^2 \ \text{ s.t. } \forall n:\ \mathbf{w}^\top\mathbf{x}_n \geq 1,$$

the residual $\rho(t)$ is bounded, and so

$$\lim_{t\to\infty}\frac{\mathbf{w}(t)}{\|\mathbf{w}(t)\|} = \frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|}.$$

In the following proofs, for any solution $\mathbf{w}(t)$, we define

$$\mathbf{r}(t) = \mathbf{w}(t) - \hat{\mathbf{w}}\log t - \tilde{\mathbf{w}},$$

where $\hat{\mathbf{w}}$ and $\tilde{\mathbf{w}}$ follow the conditions of Theorems 3 and 4, *i.e.* $\hat{\mathbf{w}}$ is the L_2 is the max margin vector defined above, and $\tilde{\mathbf{w}}$ is a vector which satisfies eq. 7:

$$\forall n\in\mathcal{S}:\ \eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right) = \alpha_n\,, \tag{18}$$

where we recall that we denoted $\mathbf{X}_\mathcal{S}\in\mathbb{R}^{d\times|\mathcal{S}|}$ as the matrix whose columns are the support vectors, a subset $\mathcal{S}\subset\{1,\ldots,N\}$ of the columns of $\mathbf{X} = [\mathbf{x}_1,\ldots,\mathbf{x}_N]\in\mathbb{R}^{d\times N}$.

In Lemma 12 (Appendix B) we prove that for almost every dataset $\boldsymbol{\alpha}$ is uniquely defined, there are no more then d support vectors and $\alpha_n\neq 0$, $\forall n\in\mathcal{S}$. Therefore, eq. 18 is well-defined in those cases. If the support vectors do not span the data, then the solution $\tilde{\mathbf{w}}$ to eq. 18 might not be unique. In this case, we can use any such solution in the proof.

We furthermore denote the minimum margin to a non-support vector as:

$$\theta = \min_{n\notin\mathcal{S}}\mathbf{x}_n^\top\hat{\mathbf{w}} > 1\,, \tag{19}$$

and by C_i,ϵ_i,t_i ($i\in\mathbb{N}$) various positive constants which are independent of t. Lastly, we define $\mathbf{P}_1\in\mathbb{R}^{d\times d}$ as the orthogonal projection matrix[5] to the subspace spanned by the support vectors (the columns of $\mathbf{X}_\mathcal{S}$), and $\bar{\mathbf{P}}_1 = \mathbf{I} - \mathbf{P}_1$ as the complementary projection (to the left nullspace of $\mathbf{X}_\mathcal{S}$).

5. This matrix can be written as $\mathbf{P}_1 = \mathbf{X}_\mathcal{S}\mathbf{X}_\mathcal{S}^+$, where $\mathbf{M}^\dagger$ is the Moore-Penrose pseudoinverse of $\mathbf{M}$.

A.1. Simple proof of Theorem 9

In this section we first examine the special case that $\ell(u) = e^{-u}$ and take the continuous time limit of gradient descent: $\eta\to 0$, so

$$\dot{\mathbf{w}}(t) = -\nabla\mathcal{L}(\mathbf{w}(t))\ .$$

The proof in this case is rather short and self-contained (*i.e.*, does not rely on any previous results), and so it helps to clarify the main ideas of the general (more complicated) proof which we will give in the next sections.

Recall we defined

$$\mathbf{r}(t) = \mathbf{w}(t) - \log(t)\,\hat{\mathbf{w}} - \tilde{\mathbf{w}}\,. \tag{20}$$

Our goal is to show that $\|\mathbf{r}(t)\|$ is bounded, and therefore $\boldsymbol{\rho}(t) = \mathbf{r}(t) + \tilde{\mathbf{w}}$ is bounded. Eq. 20 implies that

$$\dot{\mathbf{r}}(t) = \dot{\mathbf{w}}(t) - \frac{1}{t}\hat{\mathbf{w}} = -\nabla\mathcal{L}(\mathbf{w}(t)) - \frac{1}{t}\hat{\mathbf{w}} \tag{21}$$

and therefore

$$\begin{aligned}
&\frac{1}{2}\frac{d}{dt}\|\mathbf{r}(t)\|^2 = \dot{\mathbf{r}}^\top(t)\,\mathbf{r}(t)\\
&= \sum_{n=1}^{N}\exp\left(-\mathbf{x}_n^\top\mathbf{w}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t) - \frac{1}{t}\hat{\mathbf{w}}^\top\mathbf{r}(t)\\
&= \left[\sum_{n\in\mathcal{S}}\exp\left(-\log(t)\,\hat{\mathbf{w}}^\top\mathbf{x}_n - \tilde{\mathbf{w}}^\top\mathbf{x}_n - \mathbf{x}_n^\top\mathbf{r}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t) - \frac{1}{t}\hat{\mathbf{w}}^\top\mathbf{r}(t)\right]\\
&+ \left[\sum_{n\notin\mathcal{S}}\exp\left(-\log(t)\,\hat{\mathbf{w}}^\top\mathbf{x}_n - \tilde{\mathbf{w}}^\top\mathbf{x}_n - \mathbf{x}_n^\top\mathbf{r}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t)\right],
\end{aligned} \tag{22}$$

where in the last equality we used eq. 20 and decomposed the sum over support vectors $\mathcal{S}$ and non-support vectors. We examine both bracketed terms. Recall that $\hat{\mathbf{w}}^\top\mathbf{x}_n = 1$ for $n\in\mathcal{S}$, and that we defined (in eq. 18) $\tilde{\mathbf{w}}$ so that $\sum_{n\in\mathcal{S}}\exp\left(-\tilde{\mathbf{w}}^\top\mathbf{x}_n\right)\mathbf{x}_n = \hat{\mathbf{w}}$. Thus, the first bracketed term in eq. 22 can be written as

$$\begin{aligned}
&\frac{1}{t}\sum_{n\in\mathcal{S}}\exp\left(-\tilde{\mathbf{w}}^\top\mathbf{x}_n - \mathbf{x}_n^\top\mathbf{r}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t) - \frac{1}{t}\sum_{n\in\mathcal{S}}\exp\left(-\tilde{\mathbf{w}}^\top\mathbf{x}_n\right)\mathbf{x}_n^\top\mathbf{r}(t)\\
=&\frac{1}{t}\sum_{n\in\mathcal{S}}\exp\left(-\tilde{\mathbf{w}}^\top\mathbf{x}_n\right)\left(\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right) - 1\right)\mathbf{x}_n^\top\mathbf{r}(t) \leq 0,
\end{aligned} \tag{23}$$

since $\forall z,\ z(e^{-z}-1)\leq 0$. Furthermore, since $\forall z\ e^{-z}z\leq 1$ and $\theta = \operatorname{argmin}_{n\notin\mathcal{S}}\mathbf{x}_n^\top\hat{\mathbf{w}} > 1$ (eq. 19), the second bracketed term in eq. 22 can be upper bounded by

$$\sum_{n\notin\mathcal{S}}\exp\left(-\log(t)\,\hat{\mathbf{w}}^\top\mathbf{x}_n - \tilde{\mathbf{w}}^\top\mathbf{x}_n\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t) \leq \frac{1}{t^\theta}\sum_{n\notin\mathcal{S}}\exp\left(-\tilde{\mathbf{w}}^\top\mathbf{x}_n\right). \tag{24}$$

Substituting eq. 23 and 24 into eq. 22 and integrating, we obtain, that $\exists C, C'$ such that

$$\forall t_1, \forall t > t_1:\ \|\mathbf{r}(t)\|^2 - ||\mathbf{r}(t_1)||^2 \leq C\int_{t_1}^{t}\frac{dt}{t^\theta} \leq C' < \infty\,,$$

since $\theta > 1$ (eq. 19). Thus, we showed that $\mathbf{r}(t)$ is bounded, which completes the proof for the special case. ■

A.2. Complete proof of Theorem 9

Next, we give the proof for the general case (non-infinitesimal step size, and exponentially-tailed functions). Though it is based on a similar analysis as in the special case we examined in the previous section, it is somewhat more involved since we have to bound additional terms.

First, we state two auxiliary lemmata, that are proven below in appendix sections A.4 and A.5:

Lemma 10 *Let $\mathcal{L}(\mathbf{w})$ be a β-smooth non-negative objective. If $\eta < 2\beta^{-1}$, then, for any $\mathbf{w}(0)$, with the GD sequence*

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla \mathcal{L}(\mathbf{w}(t)) \tag{25}$$

we have that $\sum_{u=0}^{\infty} \|\nabla \mathcal{L}(\mathbf{w}(u))\|^2 < \infty$ and therefore $\lim_{t\to\infty} \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 = 0$.

Lemma 11 *We have*

$$\exists C_1, t_1 : \forall t > t_1 : \ (\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \le C_1 t^{-\min(\theta, 1+1.5\mu_+, 1+0.5\mu_-)} . \tag{26}$$

Additionally, $\forall \epsilon_1 > 0$, $\exists C_2, t_2$, such that $\forall t > t_2$, if

$$\|\mathbf{P}_1 \mathbf{r}(t)\| \ge \epsilon_1, \tag{27}$$

then the following improved bound holds

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \le -C_2 t^{-1} < 0 . \tag{28}$$

Our goal is to show that $\|\mathbf{r}(t)\|$ is bounded, and therefore $\boldsymbol{\rho}(t) = \mathbf{r}(t) + \tilde{\mathbf{w}}$ is bounded. To show this, we will upper bound the following equation

$$\|\mathbf{r}(t+1)\|^2 = \|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 + 2(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) + \|\mathbf{r}(t)\|^2 \tag{29}$$

First, we note that first term in this equation can be upper-bounded by

$$\begin{aligned}
&\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 \\
&\overset{(1)}{=} \|\mathbf{w}(t+1) - \hat{\mathbf{w}}\log(t+1) - \tilde{\mathbf{w}} - \mathbf{w}(t) + \hat{\mathbf{w}}\log(t) + \tilde{\mathbf{w}}\|^2 \\
&\overset{(2)}{=} \|-\eta \nabla \mathcal{L}(\mathbf{w}(t)) - \hat{\mathbf{w}}[\log(t+1) - \log(t)]\|^2 \\
&= \eta^2 \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 \log^2\left(1 + t^{-1}\right) + 2\eta \hat{\mathbf{w}}^\top \nabla \mathcal{L}(\mathbf{w}(t)) \log\left(1 + t^{-1}\right) \\
&\overset{(3)}{\le} \eta^2 \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 t^{-2}
\end{aligned} \tag{30}$$

where in (1) we used eq. 20, in (2) we used eq. 2, and in (3) we used $\forall x > 0 : \ x \ge \log(1+x) > 0$, and also that

$$\hat{\mathbf{w}}^\top \nabla \mathcal{L}(\mathbf{w}(t)) = \sum_{n=1}^{N} \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \hat{\mathbf{w}}^\top \mathbf{x}_n \le 0 , \tag{31}$$

since $\hat{\mathbf{w}}^\top \mathbf{x}_n \ge 1$ (from the definition of $\hat{\mathbf{w}}$) and $\ell'(u) \le 0$.

Also, from Lemma 10 we know that

$$\|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 = o(1) \text{ and } \sum_{t=0}^{\infty} \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 < \infty . \tag{32}$$

Substituting eq. 32 into eq. 30, and recalling that a $t^{-\nu}$ power series converges for any $\nu > 1$, we can find C_0 such that

$$\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 = o(1) \text{ and } \sum_{t=0}^{\infty} \|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 = C_0 < \infty . \tag{33}$$

Note that this equation also implies that $\forall \epsilon_0$

$$\exists t_0 : \forall t > t_0 : |\|\mathbf{r}(t+1)\| - \|\mathbf{r}(t)\|| < \epsilon_0 . \tag{34}$$

Next, we would like to bound the second term in eq. 29. From eq. 26 in Lemma 11, we can find t_1, C_1 such that $\forall t > t_1$:

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \le C_1 t^{-\min(\theta, 1+1.5\mu_+, 1+0.5\mu_-)} . \tag{35}$$

Thus, by combining eqs. 35 and 33 into eq. 29, we find

$$\begin{aligned}
&\|\mathbf{r}(t)\|^2 - \|\mathbf{r}(t_1)\|^2 \\
&= \sum_{u=t_1}^{t-1} \left[\|\mathbf{r}(u+1)\|^2 - \|\mathbf{r}(u)\|^2\right] \\
&\le C_0 + 2\sum_{u=t_1}^{t-1} C_1 u^{-\min(\theta, 1+1.5\mu_+, 1+0.5\mu_-)}
\end{aligned}$$

which is a bounded, since $\theta > 1$ (eq. 19) and $\mu_-, \mu_+ > 0$ (Definition 2). Therefore, $\|\mathbf{r}(t)\|$ is bounded. ■

A.3. Proof of Theorem 4

All that remains now is to show that $\|\mathbf{r}(t)\| \to 0$ if $\operatorname{rank}(\mathbf{X}_{\mathcal{S}}) = \operatorname{rank}(\mathbf{X})$, and that $\tilde{\mathbf{w}}$ is unique given $\mathbf{w}(0)$. To do so, this proof will continue where the proof of Theorem 3 stopped, using notations and equations from that proof.

Since $\mathbf{r}(t)$ has a bounded norm, its two orthogonal components $\mathbf{r}(t) = \mathbf{P}_1 \mathbf{r}(t) + \bar{\mathbf{P}}_1 \mathbf{r}(t)$ also have bounded norms (recall that $\mathbf{P}_1, \bar{\mathbf{P}}_1$ were defined in the beginning of appendix section A). From eq. 2, $\nabla \mathcal{L}(\mathbf{w})$ is spanned by the columns of $\mathbf{X}$. If $\operatorname{rank}(\mathbf{X}_{\mathcal{S}}) = \operatorname{rank}(\mathbf{X})$, then it is also spanned by the columns of $\mathbf{X}_{\mathcal{S}}$, and so $\bar{\mathbf{P}}_1 \nabla \mathcal{L}(\mathbf{w}) = 0$. Therefore, $\bar{\mathbf{P}}_1 \mathbf{r}(t)$ is not updated during GD, and remains constant. Since $\tilde{\mathbf{w}}$ in eq. 20 is also bounded, we can absorb this constant $\bar{\mathbf{P}}_1 \mathbf{r}(t)$ into $\tilde{\mathbf{w}}$ without affecting eq. 7 (since $\forall n \in \mathcal{S} : \ \mathbf{x}_n^\top \bar{\mathbf{P}}_1 \mathbf{r}(t) = 0$). Thus, without loss of generality, we can assume that $\mathbf{r}(t) = \mathbf{P}_1 \mathbf{r}(t)$.

We define the set

$$\mathcal{T} = \{t > \max[t_2, t_0] : \|\mathbf{r}(t)\| < \epsilon_1\} .$$

By contradiction, we assume that the complementary set is not finite,

$$\bar{\mathcal{T}} = \{t > \max[t_2, t_0] : \|\mathbf{r}(t)\| \geq \epsilon_1\} .$$

Additionally, the set $\mathcal{T}$ is not finite: if it were finite, it would have had a finite maximal point $t_{\max} \in \mathcal{T}$, and then, combining eqs. 28, 29, and 33, we would find that $\forall t > t_{\max}$

$$\|\mathbf{r}(t)\|^2 - \|\mathbf{r}(t_{\max})\|^2 = \sum_{u=t_{\max}}^{t-1} \left[\|\mathbf{r}(u+1)\|^2 - \|\mathbf{r}(u)\|^2\right] \leq C_0 - 2C_2 \sum_{u=t_{\max}}^{t-1} u^{-1} \to -\infty ,$$

which is impossible since $\|\mathbf{r}(t)\|^2 \geq 0$. Furthermore, eq. 33 implies that

$$\sum_{u=0}^{t} \|\mathbf{r}(u+1) - \mathbf{r}(t)\|^2 = C_0 - h(t)$$

where $h(t)$ is a positive monotone function decreasing to zero. Let t_3, t be any two points such that $t_3 < t$, $\{t_3, t_3+1, \dots t\} \subset \bar{\mathcal{T}}$, and $(t_3 - 1) \in \mathcal{T}$. For all such t_3 and t, we have

$$\begin{aligned}
\|\mathbf{r}(t)\|^2 &\leq \|\mathbf{r}(t_3)\|^2 + \sum_{u=t_3}^{t-1} \left[\|\mathbf{r}(u+1)\|^2 - \|\mathbf{r}(u)\|^2\right] \\
&= \|\mathbf{r}(t_3)\|^2 + \sum_{u=t_3}^{t-1} \left[\|\mathbf{r}(u+1) - \mathbf{r}(u)\|^2 + 2(\mathbf{r}(u+1) - \mathbf{r}(u))^\top \mathbf{r}(u)\right] \\
&\leq \|\mathbf{r}(t_3)\|^2 + h(t_3) - h(t-1) - 2C_2 \sum_{u=t_3}^{t-1} u^{-1} \\
&\leq \|\mathbf{r}(t_3)\|^2 + h(t_3) . \qquad (36)
\end{aligned}$$

Also, recall that $t_3 > t_0$, so from eq. 34, we have that $|\|\mathbf{r}(t_3)\| - \|\mathbf{r}(t_3 - 1)\|| < \epsilon_0$. Since $\|\mathbf{r}(t_3 - 1)\| < \epsilon_1$ (from $\mathcal{T}$ definition), we conclude that $\|\mathbf{r}(t_3)\| \leq \epsilon_1 + \epsilon_0$. Moreover, since $\bar{\mathcal{T}}$ is an infinite set, we can choose t_3 as large as we want. This implies that $\forall \epsilon_2 > 0$ we can find t_3 such that $\epsilon_2 > h(t_3)$, since $h(t)$ is a monotonically decreasing function. Therefore, from eq. 36, $\forall \epsilon_1, \epsilon_0, \epsilon_2$, $\exists t_3 \in \bar{\mathcal{T}}$ such that

$$\forall t > t_3 : \ \|\mathbf{r}(t)\|^2 \leq \epsilon_1 + \epsilon_0 + \epsilon_2 .$$

This implies that $\|\mathbf{r}(t)\| \to 0$.

Lastly, we note that since $\bar{\mathbf{P}}_1 \mathbf{r}(t)$ is not updated during GD, we have that $\bar{\mathbf{P}}_1(\tilde{\mathbf{w}} - \mathbf{w}(0)) = 0$. This sets $\tilde{\mathbf{w}}$ uniquely, together with eq. 7. ■

A.4. Proof of Lemma 10

Lemma 10 *Let $\mathcal{L}(\mathbf{w})$ be a β-smooth non-negative objective. If $\eta < 2\beta^{-1}$, then, for any $\mathbf{w}(0)$, with the GD sequence*

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla \mathcal{L}(\mathbf{w}(t)) \qquad (25)$$

we have that $\sum_{u=0}^{\infty} \|\nabla \mathcal{L}(\mathbf{w}(u))\|^2 < \infty$ and therefore $\lim_{t\to\infty} \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 = 0$.

This proof is a slightly modified version of the proof of Theorem 2 in (Ganti, 2015). Recall a well-known property of β-smooth functions:

$$\left|f(\mathbf{x}) - f(\mathbf{y}) - \nabla f(\mathbf{y})^\top (\mathbf{x} - \mathbf{y})\right| \leq \frac{\beta}{2} \|\mathbf{x} - \mathbf{y}\|^2 . \qquad (37)$$

From the β-smoothness of $\mathcal{L}(\mathbf{w})$

$$\begin{aligned}
\mathcal{L}(\mathbf{w}(t+1)) &\leq \mathcal{L}(\mathbf{w}(t)) + \nabla \mathcal{L}(\mathbf{w}(t))^\top (\mathbf{w}(t+1) - \mathbf{w}(t)) + \frac{\beta}{2} \|\mathbf{w}(t+1) - \mathbf{w}(t)\|^2 \\
&= \mathcal{L}(\mathbf{w}(t)) - \eta \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 + \frac{\beta \eta^2}{2} \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 \\
&= \mathcal{L}(\mathbf{w}(t)) - \eta \left(1 - \frac{\beta \eta}{2}\right) \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2
\end{aligned}$$

Thus, we have

$$\frac{\mathcal{L}(\mathbf{w}(t)) - \mathcal{L}(\mathbf{w}(t+1))}{\eta\left(1 - \frac{\beta\eta}{2}\right)} \geq \|\nabla \mathcal{L}(\mathbf{w}(t))\|^2$$

which implies

$$\sum_{u=0}^{t} \|\nabla \mathcal{L}(\mathbf{w}(u))\|^2 \leq \sum_{u=0}^{t} \frac{\mathcal{L}(\mathbf{w}(u)) - \mathcal{L}(\mathbf{w}(u+1))}{\eta\left(1 - \frac{\beta\eta}{2}\right)} = \frac{\mathcal{L}(\mathbf{w}(0)) - \mathcal{L}(\mathbf{w}(t+1))}{\eta\left(1 - \frac{\beta\eta}{2}\right)} .$$

The right hand side is upper bounded by a finite constant, since $L(\mathbf{w}(0)) < \infty$ and $0 \leq \mathcal{L}(\mathbf{w}(t+1))$. This implies

$$\sum_{u=0}^{\infty} \|\nabla \mathcal{L}(\mathbf{w}(u))\|^2 < \infty ,$$

and therefore $\|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 \to 0$. ■

A.5. Proof of Lemma 11

Recall that we defined $\mathbf{r}(t) = \mathbf{w}(t) - \hat{\mathbf{w}} \log t - \tilde{\mathbf{w}}$, with $\hat{\mathbf{w}}$ and $\tilde{\mathbf{w}}$ follow the conditions of the Theorems 3 and 4, *i.e*, $\hat{\mathbf{w}}$ is the L_2 max margin vector and (eq. 4), and eq. 7 holds

$$\forall n \in \mathcal{S} : \ \eta \exp\left(-\mathbf{x}_n^\top \tilde{\mathbf{w}}\right) = \alpha_n .$$

Lemma 11 *We have*

$$\exists C_1, t_1 : \forall t > t_1 : \ (\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq C_1 t^{-\min(\theta, 1+1.5\mu_+, 1+0.5\mu_-)} . \qquad (26)$$

Additionally, $\forall \epsilon_1 > 0$, $\exists C_2, t_2$, such that $\forall t > t_2$, if

$$\|\mathbf{P}_1 \mathbf{r}(t)\| \geq \epsilon_1, \qquad (27)$$

then the following improved bound holds

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq -C_2 t^{-1} < 0 . \qquad (28)$$

From Lemma 1, $\forall n: \lim_{t\to\infty} \mathbf{w}(t)^\top \mathbf{x}_n = \infty$. In addition, from assumption 3 the negative loss derivative $-\ell'(u)$ has an exponential tail e^{-u} (recall we assume $a = c = 1$ without loss of generality). Combining both facts, we have positive constants μ_-, μ_+, t_- and t_+ such that $\forall n$

$$\forall t > t_+ : -\ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \le \left(1 + \exp\left(-\mu_+ \mathbf{w}(t)^\top \mathbf{x}_n\right)\right) \exp\left(-\mathbf{w}(t)^\top \mathbf{x}_n\right) \tag{38}$$

$$\forall t > t_- : -\ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \ge \left(1 - \exp\left(-\mu_- \mathbf{w}(t)^\top \mathbf{x}_n\right)\right) \exp\left(-\mathbf{w}(t)^\top \mathbf{x}_n\right) \tag{39}$$

Next, we examine the expression we wish to bound, recalling that $\mathbf{r}(t) = \mathbf{w}(t) - \hat{\mathbf{w}} \log t - \tilde{\mathbf{w}}$:

$$\begin{aligned}
&\left(\mathbf{r}(t+1) - \mathbf{r}(t)\right)^\top \mathbf{r}(t) \\
&= \left(-\eta \nabla \mathcal{L}(\mathbf{w}(t)) - \hat{\mathbf{w}} \left[\log(t+1) - \log(t)\right]\right)^\top \mathbf{r}(t) \\
&= -\eta \sum_{n=1}^{N} \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n^\top \mathbf{r}(t) - \hat{\mathbf{w}}^\top \mathbf{r}(t) \log\left(1 + t^{-1}\right) \\
&= \hat{\mathbf{w}}^\top \mathbf{r}(t) \left[t^{-1} - \log\left(1 + t^{-1}\right)\right] - \eta \sum_{n \notin \mathcal{S}} \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n^\top \mathbf{r}(t) \\
&- \eta \sum_{n \in \mathcal{S}} \left[t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) + \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right)\right] \mathbf{x}_n^\top \mathbf{r}(t)
\end{aligned} \tag{40}$$

where in last line we used eqs. 6 and 7 to obtain

$$\hat{\mathbf{w}} = \sum_{n \in \mathcal{S}} \alpha_n \mathbf{x}_n = \eta \sum_{n \in \mathcal{S}} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{x}_n .$$

We examine the three terms in eq. 40. The first term can be upper bounded by

$$\begin{aligned}
&\hat{\mathbf{w}}^\top \mathbf{r}(t) \left[t^{-1} - \log\left(1 + t^{-1}\right)\right] \\
&\le \max\left[\hat{\mathbf{w}}^\top \mathbf{r}(t), 0\right] \left[t^{-1} - \log\left(1 + t^{-1}\right)\right] \\
&\overset{(1)}{\le} \max\left[\hat{\mathbf{w}}^\top \mathbf{P}_1 \mathbf{r}(t), 0\right] t^{-2} \\
&\overset{(2)}{\le} \begin{cases} \|\hat{\mathbf{w}}\| \epsilon_1 t^{-2} & , \text{ if } \|\mathbf{P}_1 \mathbf{r}(t)\| \le \epsilon_1 \\ o\left(t^{-1}\right) & , \text{ if } \|\mathbf{P}_1 \mathbf{r}(t)\| > \epsilon_1 \end{cases}
\end{aligned} \tag{41}$$

where in (1) we used that $\bar{\mathbf{P}}_1 \hat{\mathbf{w}} = \bar{\mathbf{P}}_1 \mathbf{X}_{\mathcal{S}} \boldsymbol{\alpha} = 0$ from eq. 6, and in (2) we used that $\hat{\mathbf{w}}^\top \mathbf{r}(t) = o(t)$, since

$$\begin{aligned}
\hat{\mathbf{w}}^\top \mathbf{r}(t) &= \hat{\mathbf{w}}^\top \left(\mathbf{w}(0) - \eta \sum_{u=0}^{t} \nabla \mathcal{L}(\mathbf{w}(u)) - \hat{\mathbf{w}} \log(t) - \tilde{\mathbf{w}}\right) \\
&\le \hat{\mathbf{w}}^\top \left(\mathbf{w}(0) - \tilde{\mathbf{w}} - \hat{\mathbf{w}} \log(t)\right) - \eta t \min_{0 \le u \le t} \hat{\mathbf{w}}^\top \nabla \mathcal{L}(\mathbf{w}(u)) = o(t)
\end{aligned}$$

where in the last line we used that $\nabla \mathcal{L}(\mathbf{w}(t)) = o(1)$, from Lemma 10.

Next, we upper bound the second term in eq. 40. From eq. 38 $\exists t'_+$, such that $\forall > t_0 > t'_+$,

$$\ell'(\mathbf{w}(t)^\top \mathbf{x}_n) \le 2 \exp(-\mathbf{w}(t)^\top \mathbf{x}_n). \tag{42}$$

Therefore, $\forall t > t'_+$:

$$\begin{aligned}
&-\eta \sum_{n \notin \mathcal{S}} \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n^\top \mathbf{r}(t) \\
&\le -\eta \sum_{n \notin \mathcal{S}: \mathbf{x}_n^\top \mathbf{r}(t) \ge 0} \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n^\top \mathbf{r}(t) \\
&\overset{(1)}{\le} \eta \sum_{n \notin \mathcal{S}: \mathbf{x}_n^\top \mathbf{r}(t) \ge 0} 2 \exp\left(-\mathbf{w}(t)^\top \mathbf{x}_n\right) \mathbf{x}_n^\top \mathbf{r}(t) \\
&\overset{(2)}{\le} \eta \sum_{n \notin \mathcal{S}: \mathbf{x}_n^\top \mathbf{r}(t) \ge 0} 2 t^{-\mathbf{x}_n^\top \hat{\mathbf{w}}} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n - \mathbf{x}_n^\top \mathbf{r}(t)\right) \mathbf{x}_n^\top \mathbf{r}(t) \\
&\overset{(3)}{\le} \eta \sum_{n \notin \mathcal{S}: \mathbf{x}_n^\top \mathbf{r}(t) \ge 0} 2 t^{-\mathbf{x}_n^\top \hat{\mathbf{w}}} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \\
&\overset{(4)}{\le} \eta N \exp\left(-\min_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right) t^{-\theta}
\end{aligned} \tag{43}$$

where in (1) we used eq. 42, in (2) we used $\mathbf{w}(t) = \hat{\mathbf{w}} \log t + \tilde{\mathbf{w}} + \mathbf{r}(t)$, in (3) we used $xe^{-x} \le 1$ and $\mathbf{x}_n^\top \mathbf{r}(t) \ge 0$,and in (4) we used $\theta > 1$, from eq. 19.

Lastly, we will bound the sum in the third term in eq. 40

$$-\eta \sum_{n \in \mathcal{S}} \left[t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) + \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right)\right] \mathbf{x}_n^\top \mathbf{r}(t) . \tag{44}$$

We examine each term n in this sum, and divide into two cases, depending on the sign of $\mathbf{x}_n^\top \mathbf{r}(t)$.

First, if $\mathbf{x}_n^\top \mathbf{r}(t) \ge 0$, then term n in eq. 44 can be upper bounded $\forall t > t_+$, using eq. 38, by

$$\eta t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \left[\left(1 + t^{-\mu_+} \exp\left(-\mu_+ \tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\right) \exp\left(-\mathbf{x}_n^\top \mathbf{r}(t)\right) - 1\right] \mathbf{x}_n^\top \mathbf{r}(t) \tag{45}$$

We further divide into cases:

1. If $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| \le C_0 t^{-0.5\mu_+}$, then we can upper bound eq. 45 with

$$\eta \exp\left(-(1 + \mu_+) \min_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right) C_0 t^{-1-1.5\mu_+} . \tag{46}$$

2. If $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| > C_0 t^{-0.5\mu_+}$, then we can find $t''_+ > t'_+$ to upper bound eq. 45 $\forall t > t''_+$:

$$\begin{aligned}
&\eta t^{-1} e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n} \left[\left(1 + t^{-\mu_+} e^{-\mu_+ \tilde{\mathbf{w}}^\top \mathbf{x}_n}\right) \exp\left(-C_0 t^{-0.5\mu_+}\right) - 1\right] \mathbf{x}_n^\top \mathbf{r}(t) \\
&\overset{(1)}{\le} \eta t^{-1} e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n} \left[\left(1 + t^{-\mu_+} e^{-\mu_+ \tilde{\mathbf{w}}^\top \mathbf{x}_n}\right) \left(1 - C_0 t^{-0.5\mu_+} + C_0^2 t^{-\mu_+}\right) - 1\right] \mathbf{x}_n^\top \mathbf{r}(t) \\
&\le \eta t^{-1} e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n} \left[\left(1 - C_0 t^{-0.5\mu_+} + C_0^2 t^{-\mu_+}\right) e^{-\mu_+ \min_n \tilde{\mathbf{w}}^\top \mathbf{x}_n} t^{-\mu_+} - C_0 t^{-0.5\mu_+} + C_0^2 t^{-\mu_+}\right] \mathbf{x}_n^\top \mathbf{r}(t) \\
&\overset{(2)}{\le} 0, \ \forall t > t''_+
\end{aligned} \tag{47}$$

where in (1) we used the fact that $e^{-x} \le 1 - x + x^2$ for $x \ge 0$ and in (2) we defined t''_+ so that the previous expression is negative — since $t^{-0.5\mu_+}$ decreases slower than $t^{-\mu_+}$.

3. If $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| \geq \epsilon_2$, then we define $t'''_+ > t''_+$ such that $t'''_+ > \exp\left(\min_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\left[e^{0.5\epsilon_2}-1\right]^{-1/\mu_+}$, and therefore $\forall t > t'''_+$, we have $\left(1+t^{-\mu_+}\exp\left(-\mu_+ \tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\right)e^{-\epsilon_2} < e^{-0.5\epsilon_2}$.

   This implies that $\forall t > t'''_+$ we can upper bound eq. 45 by

$$-\eta \exp\left(-\max_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\left(1-e^{-0.5\epsilon_2}\right)\epsilon_2 t^{-1}. \tag{48}$$

Second, if $\mathbf{x}_n^\top \mathbf{r}(t) < 0$, we again further divide into cases:

1. If $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| \leq C_0 t^{-0.5\mu_-}$, then, since $-\ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right) > 0$, we can upper bound term n in eq. 44 with

$$\eta t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| \leq \eta \exp\left(-\min_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right) C_0 t^{-1-0.5\mu_-} \tag{49}$$

2. If $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| > C_0 t^{-0.5\mu_-}$, then, using eq. 39 we upper bound term n in eq. 44 with

$$\begin{aligned}
&\eta\left[-t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n} - \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_n\right)\right]\mathbf{x}_n^\top \mathbf{r}(t)\\
\leq&\eta\left[-t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n} + \left(1-\exp\left(-\mu_- \mathbf{w}(t)^\top \mathbf{x}_n\right)\right)\exp\left(-\mathbf{w}(t)^\top \mathbf{x}_n\right)\right]\mathbf{x}_n^\top \mathbf{r}(t)\\
=&\eta t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}\left[1-\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\left(1-\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\right]^{\mu_-}\right)\right]\left|\mathbf{x}_n^\top \mathbf{r}(t)\right|
\end{aligned} \tag{50}$$

   Next, we will show that $\exists t'_- > t_-$ such that the last expression is strictly negative $\forall t > t'_-$. Let $M > 1$ be some arbitrary constant. Then, since $\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\right]^{\mu_-} = \exp\left(-\mu_- \mathbf{w}(t)^\top \mathbf{x}_n\right) \to 0$ from Lemma 1, $\exists t_M > \max(t_-, Me^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n})$ such that $\forall t > t_M$, if $\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right) \geq M > 1$ then

$$\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\left(1-\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\right]^{\mu_-}\right) \geq M' > 1. \tag{51}$$

   Furthermore, if $\exists t > t_M$ such that $\exp\left(\mathbf{r}(t)^\top \mathbf{x}_n\right) < M$, then

$$\begin{aligned}
&\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\left(1-\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\right]^{\mu_-}\right)\\
>&\exp\left(-\mathbf{r}(t)^\top \mathbf{x}_n\right)\left(1-\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}M\right]^{\mu_-}\right).
\end{aligned} \tag{52}$$

   which is lower bounded by

$$\begin{aligned}
&\left(1+C_0 t^{-0.5\mu_-}\right)\left(1-t^{-\mu_-}\left[e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}M\right]^{\mu_-}\right)\\
\geq&1+C_0 t^{-0.5\mu_-} - t^{-\mu_-}\left[e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}M\right]^{\mu_-} - t^{-1.5\mu_-}\left[e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_n}M\right]^{\mu_-} C_0
\end{aligned}$$

since $\left|\mathbf{x}_n^\top \mathbf{r}(t)\right| > C_0 t^{-0.5\mu_-}$, $\mathbf{x}_n^\top \mathbf{r}(t) < 0$ and $e^x \geq 1+x$. In this case last line is strictly larger than 1 for sufficiently large t. Therefore, after we substitute eqs. 51 and 52 into 50, we find that $\exists t'_- > t_M > t_-$ such that $\forall t > t'_-$, term k in eq. 44 is strictly negative

$$\eta\left[-t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_k} - \ell'\left(\mathbf{w}(t)^\top \mathbf{x}_k\right)\right]\mathbf{x}_k^\top \mathbf{r}(t) < 0 \tag{53}$$

3. If $\left|\mathbf{x}_k^\top \mathbf{r}(t)\right| \geq \epsilon_2$, which is a special case of the previous case ($\left|\mathbf{x}_k^\top \mathbf{r}(t)\right| > C_0 t^{-0.5\mu_-}$) then $\forall t > t'_-$, either eq. 51 or 52 holds. Furthermore, in this case, $\exists t''_- > t'_-$ and $M'' > 1$ such that $\forall t > t''_-$ eq. 52 can be lower bounded by

$$\exp(\epsilon_2)\left(1-\left[t^{-1}e^{-\tilde{\mathbf{w}}^\top \mathbf{x}_k}M\right]^{\mu_-}\right) > M'' > 1.$$

   Substituting this, together with eq. 51, into eq. 50, we can find $C'_0 > 0$ such we can upper bound term k in eq. 44 with

$$-C'_0 t^{-1}, \forall t > t''_-. \tag{54}$$

To conclude, we choose $t_0 = \max\left[t'''_+, t''_-\right]$:

1. If $\|\mathbf{P}_1 \mathbf{r}(t)\| \geq \epsilon_1$ (as in Eq. 27), we have that

$$\max_{n\in\mathcal{S}}\left|\mathbf{x}_n^\top \mathbf{r}(t)\right|^2 \overset{(1)}{\geq} \frac{1}{|\mathcal{S}|}\sum_{n\in\mathcal{S}}\left|\mathbf{x}_n^\top \mathbf{P}_1 \mathbf{r}(t)\right|^2 = \frac{1}{|\mathcal{S}|}\left\|\mathbf{X}_\mathcal{S}^\top \mathbf{P}_1 \mathbf{r}(t)\right\|^2 \overset{(2)}{\geq} \frac{1}{|\mathcal{S}|}\sigma_{\min}^2(\mathbf{X}_\mathcal{S})\epsilon_1^2 \tag{55}$$

   where in (1) we used $\mathbf{P}_1^\top \mathbf{x}_n = \mathbf{x}_n\ \forall n \in \mathcal{S}$, in (2) we denoted by $\sigma_{\min}(\mathbf{X}_\mathcal{S})$, the minimal non-zero singular value of $\mathbf{X}_\mathcal{S}$ and used eq. 27. Therefore, for some k, $\left|\mathbf{x}_k^\top \mathbf{r}\right| \geq \epsilon_2 \triangleq \sqrt{|\mathcal{S}|^{-1}\sigma_{\min}^2(\mathbf{X}_\mathcal{S})\epsilon_1^2}$. In this case, we denote C''_0 as the minimum between C'_0 (eq. 54) and $\eta\exp\left(-\max_n \tilde{\mathbf{w}}^\top \mathbf{x}_n\right)\left(1-e^{-0.5\epsilon_2}\right)\epsilon_2$ (eq. 48). Then we find that eq. 44 can be upper bounded by $-C''_0 t^{-1} + o\left(t^{-1}\right), \forall t > t_0$, given eq. 27. Substituting this result, together with eqs. 41 and 43 into eq. 40, we obtain $\forall t > t_0$

$$\left(\mathbf{r}(t+1)-\mathbf{r}(t)\right)^\top \mathbf{r}(t) \leq -C''_0 t^{-1} + o\left(t^{-1}\right).$$

   This implies that $\exists C_2 < C''_0$ and $\exists t_2 > t_0$ such that eq. 28 holds. This implies also that eq. 26 holds for $\|\mathbf{P}_1 \mathbf{r}(t)\| \geq \epsilon_1$.

2. Otherwise, if $\|\mathbf{P}_1 \mathbf{r}(t)\| < \epsilon_1$, we find that $\forall t > t_0$, each term in eq. 44 can be upper bounded by either zero (eqs. 47 and 53), or terms proportional to $t^{-1-1.5\mu_+}$ (eq. 46) or $t^{-1-0.5\mu_-}$, (eq. 49). Combining this together with eqs. 41, 43 into eq. 40 we obtain (for some positive constants C_3, C_4, C_5, and C_6)

$$\left(\mathbf{r}(t+1)-\mathbf{r}(t)\right)^\top \mathbf{r}(t) \leq C_3 t^{-1-1.5\mu_+} + C_4 t^{-1-0.5\mu_-} + C_5 t^{-2} + C_6 t^{-\theta}.$$

   Therefore, $\exists t_1 > t_0$ and C_1 such that eq. 26 holds. ■

Appendix B. Generic solutions of the KKT conditions in eq. 6

Lemma 12 *For almost all datasets there is a unique $\boldsymbol{\alpha}$ which satisfies the KKT conditions (eq. 6):*

$$\hat{\mathbf{w}} = \sum_{n=1}^N \alpha_n \mathbf{x}_n \qquad \forall n\ \left(\alpha_n \geq 0 \text{ and } \hat{\mathbf{w}}^\top \mathbf{x}_n = 1\right) \text{ OR } \left(\alpha_n = 0 \text{ and } \hat{\mathbf{w}}^\top \mathbf{x}_n > 1\right)$$

Furthermore, in this solution $\alpha_n \neq 0$ if $\hat{\mathbf{w}}^\top \mathbf{x}_n = 1$, i.e., *$\mathbf{x}_n$ is a support vector ($n \in \mathcal{S}$), and there are at most d such support vectors.*

For almost every set $\mathbf{X}$, no more than d points $\mathbf{x}_n$ can be on the same hyperplane. Therefore, since all support vectors must lie on the same hyperplane, there can be at most d support vectors, for almost every $\mathbf{X}$.

Given the set of support vectors, $\mathcal{S}$, the KKT conditions of eq. 6 entail that $\alpha_n = 0$ if $n \notin \mathcal{S}$ and

$$\mathbf{1} = \mathbf{X}_{\mathcal{S}}^{\top}\hat{\mathbf{w}} = \mathbf{X}_{\mathcal{S}}^{\top}\mathbf{X}_{\mathcal{S}}\boldsymbol{\alpha}_{\mathcal{S}}\,, \tag{56}$$

where we denoted $\boldsymbol{\alpha}_{\mathcal{S}}$ as $\boldsymbol{\alpha}$ restricted to the support vector components. For almost every set $\mathbf{X}$, since $d \geq |\mathcal{S}|$, $\mathbf{X}_{\mathcal{S}}^{\top}\mathbf{X}_{\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times|\mathcal{S}|}$ is invertible. Therefore, $\boldsymbol{\alpha}_{\mathcal{S}}$ has the unique solution

$$\left(\mathbf{X}_{\mathcal{S}}^{\top}\mathbf{X}_{\mathcal{S}}\right)^{-1}\mathbf{1} = \boldsymbol{\alpha}_{\mathcal{S}}\,. \tag{57}$$

This implies that $\forall n \in \mathcal{S}$, α_n is equal to a rational function in the components of $\mathbf{X}_S$, *i.e.*, $\alpha_n = p_n\left(\mathbf{X}_{\mathcal{S}}\right)/q_n\left(\mathbf{X}_{\mathcal{S}}\right)$, where p_n and q_n are polynomials in the components of $\mathbf{X}_S$. Therefore, if $\alpha_n = 0$, then $p_n\left(\mathbf{X}_{\mathcal{S}}\right) = 0$, so the components of $\mathbf{X}_{\mathcal{S}}$ must be at a root of the polynomial p_n. The roots of the polynomial p_n have measure zero, unless $\forall \mathbf{X}_{\mathcal{S}}:\ p_n\left(\mathbf{X}_{\mathcal{S}}\right) = 0$. However, p_n cannot be identically equal to zero, since, for example, if $\mathbf{X}_{\mathcal{S}}^{\top} = \left[\mathbf{I}_{|\mathcal{S}|\times|\mathcal{S}|}, \mathbf{0}_{|\mathcal{S}|\times(d-|\mathcal{S}|)}\right]$, then $\mathbf{X}_{\mathcal{S}}^{\top}\mathbf{X}_{\mathcal{S}} = \mathbf{I}_{|\mathcal{S}|\times|\mathcal{S}|}$, and so in this case $\forall n \in \mathcal{S}$, $\alpha_n = 1 \neq 0$, from eq. 57.

Therefore, for a given $\mathcal{S}$, the event that "eq. 56 has a solution with a zero component" has a zero measure. Moreover, the union of these events, for all possible $\mathcal{S}$, also has zero measure, as a finite union of zero measures sets (there are only finitely many possible sets $\mathcal{S} \subset \{1,\ldots,N\}$). This implies that, for almost all datasets $\mathbf{X}$, $\alpha_n = 0$ only if $n \notin \mathcal{S}$. Furthermore, for almost all datasets the solution $\boldsymbol{\alpha}$ is unique: for each dataset, $\mathcal{S}$ is uniquely determined, and given $\mathcal{S}$, the solution eq. 56 is uniquely given by eq. 57. ■

Appendix C. Completing the proof of Theorem 3 for zero measure cases

In the preceding Appendices, we established Theorem 4, which only applied when all support vectors are associated with non-zero coefficients. This characterizes almost all data sets, *i.e.* all except for measure zero. We now turn to presenting and proving a more complete characterization of the limit behaviour of gradient descent, which covers all data sets, including those degenerate data sets not covered by Theorem 4, thus establishing Theorem 3.

In order to do so, we first have to introduce additional notation and a recursive treatment of the data set. We will define a sequence of data sets $\bar{\mathbf{P}}_m\mathbf{X}_{\bar{\mathcal{S}}_m}$ obtained by considering only a subset $\bar{\mathcal{S}}_m$ of the points, and projecting them using the projection matrix $\bar{\mathbf{P}}_m$. We start, for $m = 0$, with the full original data set, *i.e.* $\bar{\mathcal{S}}_0 = \{1,\ldots,N\}$ and $\bar{\mathbf{P}}_0 = \mathbf{I}_{d\times d}$. We then define $\hat{\mathbf{w}}_m$ as the max margin predictor for $\bar{\mathbf{P}}_{m-1}\mathbf{X}_{\bar{\mathcal{S}}_{m-1}}$, *i.e.*:

$$\hat{\mathbf{w}}_m = \underset{\mathbf{w}\in\mathbb{R}^d}{\operatorname{argmin}} \|\mathbf{w}\|^2 \ \text{ s.t. } \mathbf{w}^{\top}\bar{\mathbf{P}}_{m-1}\mathbf{x}_n \geq 1\ \forall n \in \bar{\mathcal{S}}_{m-1}\,. \tag{58}$$

In particular, $\hat{\mathbf{w}}_1$ is the max margin predictor for the original data set. We then denote $\mathcal{S}_m^+$ the indices of non-support vectors for 58, $\mathcal{S}_m$ the indices of support vector of 58 with non-zero coefficients for the dual variables corresponding to the margin constraints (for some dual solution), and $\bar{\mathcal{S}}_m$ the set

of support vector with zero coefficients. That is:

$$\begin{aligned}
\mathcal{S}_m^+ &= \left\{n \in \bar{\mathcal{S}}_{m-1} | \hat{\mathbf{w}}_m^{\top}\bar{\mathbf{P}}_{m-1}\mathbf{x}_n > 1\right\}\\
\mathcal{S}_m^= &= \left\{n \in \bar{\mathcal{S}}_{m-1} | \hat{\mathbf{w}}_m^{\top}\bar{\mathbf{P}}_{m-1}\mathbf{x}_n = 1\right\} = \bar{\mathcal{S}}_m \setminus \mathcal{S}_m^+\\
\mathcal{S}_m &= \left\{n \in \mathcal{S}_m^= | \exists \boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^N : \hat{\mathbf{w}}_m = \sum_{k=1}^{N}\alpha_k\bar{\mathbf{P}}_{m-1}\mathbf{x}_k, \alpha_n > 0, \forall i \notin \mathcal{S}_m^=:\ \alpha_i = 0\right\}\\
\bar{\mathcal{S}}_m &= \mathcal{S}_m^= \setminus \mathcal{S}_m\,.
\end{aligned} \tag{59}$$

The problematic degenerate case, not covered by the analysis of Theorem 4, is when there are support vectors with zero coefficients, *i.e.*, when $\bar{\mathcal{S}}_m \neq \emptyset$. In this case we recurse on these zero-coefficient support vectors (*i.e.*, on $\bar{\mathcal{S}}_m$), but only consider their components orthogonal to the non-zero-coefficient support vectors (*i.e.*, not spanned by points in $\mathcal{S}_m$). That is, we project using:

$$\bar{\mathbf{P}}_m = \bar{\mathbf{P}}_{m-1}\left(\mathbf{I}_d - \mathbf{X}_{\mathcal{S}_m}\mathbf{X}_{\mathcal{S}_m}^{\dagger}\right) \tag{60}$$

where we denoted $\mathbf{A}^{\dagger}$ as the Moore-Penrose pseudo-inverse of $\mathbf{A}$. We also denote $\mathbf{P}_m = \mathbf{I}_d - \bar{\mathbf{P}}_m$.

This recursive treatment continues as long as $\bar{\mathcal{S}}_m \neq \emptyset$, defining a sequence $\hat{\mathbf{w}}_m$ of max margin predictors, for smaller and lower dimensional data sets $\bar{\mathbf{P}}_{m-1}\mathbf{X}_{\bar{\mathcal{S}}_{m-1}}$. We stop when $\bar{\mathcal{S}}_m = \emptyset$ and denote the stopping stage M—that is, M is the minimal m such that $\bar{\mathcal{S}}_m = \emptyset$. Our characterization will be in terms of the sequence $\hat{\mathbf{w}}_1,\ldots,\hat{\mathbf{w}}_M$. As established in Lemma 12 of Appendix B, for almost all data sets we will not have support vectors with non-zero coefficients, and so we will have $M = 1$, and so the characterization only depends on the max margin predictor $\hat{\mathbf{w}}_1$ of the original data set. But, even for the measure zero of data sets in which $M > 1$, we provide the following more complete characterization:

Theorem 13 *For all datasets which are linearly separable (Assumption 1) and given a β-smooth loss function (Assumption 2) with an exponential tail (Assumption 3), gradient descent (as in eq. 2) with step size $\eta < 2\beta^{-1}\sigma_{\max}^{-2}\left(\mathbf{X}\right)$ and any starting point $\mathbf{w}(0)$, the iterates of gradient descent can be written as:*

$$\mathbf{w}\left(t\right) = \sum_{m=1}^{M}\hat{\mathbf{w}}_m\log^{\circ m}\left(t\right) + \boldsymbol{\rho}\left(t\right)\,, \tag{61}$$

where $\log^{\circ m}\left(t\right) = \overbrace{\log\log\cdots\log}^{m\,\text{times}}\left(t\right)$, $\hat{\mathbf{w}}_m$ is the L_2 max margin vector defined in eq. 58, and the residual $\boldsymbol{\rho}\left(t\right)$ is bounded.

C.1. Auxiliary notation

We say that a function $f:\mathbb{N}\to\mathbb{R}$ is absolutely summable if $\sum_{t=1}^{\infty}|f\left(t\right)| < \infty$, and then we denote $f\left(t\right) \in L_1$. Furthermore, we define

$$\mathbf{r}\left(t\right) = \mathbf{w}\left(t\right) - \sum_{m=1}^{M}\left[\hat{\mathbf{w}}_m\log^{\circ m}\left(t\right) + \tilde{\mathbf{w}}_m + \sum_{k=1}^{m-1}\frac{\breve{\mathbf{w}}_{k,m}}{\prod_{r=k}^{m-1}\log^{\circ r}\left(t\right)}\right]$$

where $\tilde{\mathbf{w}}_m$ and $\breve{\mathbf{w}}_{k,m}$ are defined next, and additionally, we denote

$$\tilde{\mathbf{w}} = \sum_{m=1}^{M} \tilde{\mathbf{w}}_m \,.$$

We define, $\forall m \geq 1$, $\tilde{\mathbf{w}}_m$ as the solution of

$$\forall m \geq 1 : \forall n \in \mathcal{S}_m : \eta \sum_{n \in \mathcal{S}_m} \exp\left(-\sum_{k=1}^{m} \tilde{\mathbf{w}}_k^\top \mathbf{x}_n\right) \bar{\mathbf{P}}_{m-1} \mathbf{x}_n = \hat{\mathbf{w}}_m \,, \tag{62}$$

such that

$$\mathbf{P}_{m-1} \tilde{\mathbf{w}}_m = 0 \,\text{and}\, \bar{\mathbf{P}}_m \tilde{\mathbf{w}}_m = 0. \tag{63}$$

The existence and uniqueness of the solution, $\tilde{\mathbf{w}}_m$ are proved in appendix section C.4.

Lastly, we define, $\forall m > k \geq 1$, $\check{\mathbf{w}}_{k,m}$ as the solution of

$$\sum_{n \in \mathcal{S}_m} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{P}_{m-1} \mathbf{x}_n = \sum_{k=1}^{m-1} \left[\sum_{n \in \mathcal{S}_k} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{x}_n \mathbf{x}_n^\top\right] \check{\mathbf{w}}_{k,m} \tag{64}$$

such that

$$\mathbf{P}_{k-1} \check{\mathbf{w}}_{k,m} = 0 \,\text{and}\, \bar{\mathbf{P}}_k \check{\mathbf{w}}_{k,m} = 0 \,. \tag{65}$$

The existence and uniqueness of the solution $\check{\mathbf{w}}_{k,m}$ are proved in appendix section C.5.

Together, eqs. 62-65 entail the existence of a unique decomposition, $\forall m \geq 1$:

$$\hat{\mathbf{w}}_m = \eta \sum_{n \in \mathcal{S}_m} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{x}_n - \eta \sum_{k=1}^{m-1} \left[\sum_{n \in \mathcal{S}_k} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{x}_n \mathbf{x}_n^\top\right] \check{\mathbf{w}}_{k,m} \tag{66}$$

given the constraints in eqs. 63 and 65 hold.

C.2. Proof of Theorem 13

In the following proofs, for any solution $\mathbf{w}(t)$, we define

$$\boldsymbol{\tau}(t) = \sum_{m=2}^{M} \hat{\mathbf{w}}_m \log^{\circ m}(t) + \sum_{m=1}^{M} \sum_{k=1}^{m-1} \frac{\check{\mathbf{w}}_{k,m}}{\prod_{r=k}^{m-1} \log^{\circ r}(t)}$$

noting that

$$\|\boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t)\| \leq \frac{C_\tau}{t \log(t)}$$

and

$$\mathbf{r}(t) = \mathbf{w}(t) - \hat{\mathbf{w}}_1 \log(t) - \tilde{\mathbf{w}} - \boldsymbol{\tau}(t) \tag{67}$$

where $\tilde{\mathbf{w}}$ follow the conditions of Theorem 13. Our goal is to show that $\|\mathbf{r}(t)\|$ is bounded. To show this, we will upper bound the following equation

$$\|\mathbf{r}(t+1)\|^2 = \|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 + 2\left(\mathbf{r}(t+1) - \mathbf{r}(t)\right)^\top \mathbf{r}(t) + \|\mathbf{r}(t)\|^2 \tag{68}$$

First, we note that $\exists t_0$ such that $\forall t > t_0$ the first term in this equation can be upper bounded by

$$\begin{aligned}
&||\mathbf{r}(t+1) - \mathbf{r}(t)||^2 \\
&\overset{(1)}{=} ||\mathbf{w}(t+1) - \hat{\mathbf{w}}_1 \log(t+1) - \boldsymbol{\tau}(t+1) - \mathbf{w}(t) + \hat{\mathbf{w}}_1 \log(t) + \boldsymbol{\tau}(t)||^2 \\
&\overset{(2)}{=} ||-\eta \nabla L(\mathbf{w}(t)) - \hat{\mathbf{w}}_1 (\log(t+1) - \log(t)) - (\boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t))||^2 \\
&= \eta^2 ||\nabla L(\mathbf{w}(t))||^2 + \|\hat{\mathbf{w}}_1\|^2 \log^2\left(1 + t^{-1}\right) + \|\boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t)\|^2 \\
&+ 2\eta \nabla L(\mathbf{w}(t))^\top \left(\hat{\mathbf{w}}_1 \log\left(1 + t^{-1}\right) + \boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t)\right) \\
&+ 2\hat{\mathbf{w}}_1^\top (\boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t)) \log\left(1 + t^{-1}\right) \\
&\overset{(3)}{\leq} \eta^2 ||\nabla L(\mathbf{w}(t))||^2 + \|\hat{\mathbf{w}}_1\|^2 t^{-2} + C_\tau^2 t^{-2} \log^{-2}(t) + 2C_\tau \|\hat{\mathbf{w}}_1\| t^{-2} \log^{-1}(t) \;\;, \forall t > t_0
\end{aligned} \tag{69}$$

where in (1) we used eq. 67, in (2) we used eq. 2 and in (3) we used $\forall x > 0 : x \geq \log(1+x) > 0$, and also using $\ell'(\mathbf{w}(t)^\top \mathbf{x}_n) < 0$ for large enough t, we have that

$$\left(\hat{\mathbf{w}}_1 \log\left(1 + t^{-1}\right) + \boldsymbol{\tau}(t+1) - \boldsymbol{\tau}(t)\right)^\top \nabla \mathcal{L}(\mathbf{w}(t)) \leq \sum_{n=1}^{N} \ell'(\mathbf{w}(t)^\top \mathbf{x}_n) \left(\hat{\mathbf{w}}_1^\top \mathbf{x}_n \log\left(1 + t^{-1}\right) - \frac{\|\mathbf{x}_n\| C'_\tau}{t \log(t)}\right) \tag{70}$$

which is negative for sufficiently large t_0 (since $\log\left(1 + t^{-1}\right)$ decreases as t^{-1}, which is slower then $1/(t \log(t))$), $\forall n : \hat{\mathbf{w}}_1^\top \mathbf{x}_n \geq 1$ and $\ell'(u) \leq 0$.

Also, from Lemma 10 we know that:

$$\|\nabla \mathcal{L}(\mathbf{w}(t))\|^2 = o(1) \text{ and } \sum_{u=0}^{\infty} \|\nabla \mathcal{L}(\mathbf{w}(u))\|^2 < \infty \tag{71}$$

Substituting eq. 71 into eq. 69, and recalling that $t^{-\nu_1} \log^{-\nu_2}(t)$ converges for any $\nu_1 > 1$ and any ν_2, and so

$$\kappa_0(t) \triangleq \|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 \in L_1 \,. \tag{72}$$

Also, in the next subsection we will prove that

Lemma 14 *Let $\kappa_1(t)$ and $\kappa_2(t)$ be functions in L_1, then*

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq \kappa_1(t) \|\mathbf{r}(t)\| + \kappa_2(t) \tag{73}$$

Thus, by combining eqs. 73 and 72 into eq. 68, we find

$$\|\mathbf{r}(t+1)\|^2 \leq \kappa_0(t) + 2\kappa_1(t) \|\mathbf{r}(t)\| + 2\kappa_2(t) + \|\mathbf{r}(t)\|^2$$

On this result we apply the following lemma (with $\phi(t) = \|\mathbf{r}(t)\|$, $h(t) = 2\kappa_1(t)$, and $z(t) = \kappa_0(t) + 2\kappa_2(t)$), which we prove in appendix C.6:

Lemma 15 *Let $\phi(t), h(t), z(t)$ be three functions from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$, and C_1, C_2, C_3 be three positive constants. Then, if $\sum_{t=1}^{\infty} h(t) \leq C_1 < \infty$, and*

$$\phi^2(t+1) \leq z(t) + h(t) \phi(t) + \phi^2(t) \tag{74}$$

we have

$$\phi^2(t+1) \leq C_2 + C_3 \sum_{u=1}^{t} z(u) \tag{75}$$

and obtain that

$$\|\mathbf{r}(t+1)\|^2 \leq C_2 + C_3 \sum_{u=1}^{t} \left(\kappa_0(u) + 2\kappa_2(u)\right) \leq C_4 < \infty\,,$$

since we assumed that $\forall i = 0,1,2:\ \kappa_i(t) \in L_1$. This completes our proof. ■

C.3. Proof of Lemma 14

Before we prove Lemma 14, we prove the following auxilary Lemma:

Lemma 16 *Consider the function* $f(t) = t^{-\nu_1}(\log(t))^{-\nu_2}(\log\log(t))^{-\nu_3}\ldots(\log^{\circ M}(t))^{-\nu_{M+1}}$. *If* $\exists m_0 \leq M+1$ *such that* $\nu_{m_0} > 1$ *and for all* $m' < m_0, \nu_{m'} = 1$, *then* $f(t) \in L_1$.

Proof To prove Lemma 16, we will show that the improper integeral $\int_{t_1}^{\infty} f(t)dt$ for any $t_1 > 0$ is bounded, *i.e.*, $\forall t_1 > 0, \int_{t_1}^{\infty} f(t)dt < C$. Using the integeral test for convergence (or Maclaurin–Cauchy test) this in turn implies that $\forall t_1 > 0, \sum_{t_1}^{\infty} f(t) < C$, and thus $f(t) \in L_1$.

First, if $m_0 > 1$, then $\nu_1 = \nu_2 \ldots = \nu_{m_0-1} = 1$ and $\nu_{m_0} = 1+\epsilon$ for some $\epsilon > 0$. Using change of variables $y = \log^{\circ(m_0-1)}(t)$, we have

$$\mathrm{d}y = \left(t\prod_{r=1}^{m_0-2}\log^{\circ r}(t)\right)^{-1}\mathrm{d}t = t^{-\nu_1}\prod_{r=1}^{m_0-2}\left(\log^{\circ r}(t)\right)^{-\nu_{r+1}}\mathrm{d}t$$

and for all $m > m_0$, $\left(\log^{\circ(m-1)}(t)\right)^{-\nu_m} = \left(\log^{\circ(m-m_0)}(y)\right)^{-\nu_m} \leq (\log(y))^{|\nu_m|}$. Thus, denoting $\tilde{\nu} = \sum_{m=m_0+1}^{M+1}|\nu_m|$ and $\log^{\circ(m_0-1)}(t_1) = y_1$, we have

$$\int_{t_1}^{\infty} f(t)\mathrm{d}t = \int_{y_1}^{\infty} y^{-\nu_{m_0}} \prod_{m=m_0+1}^{M+1}\left(\log^{\circ m-m_0}(y)\right)^{-\nu_m}\mathrm{d}(y) \leq \int_{y_1}^{\infty}\frac{(\log(y))^{\tilde{\nu}}}{y^{1+\epsilon}}\mathrm{d}y. \tag{76}$$

For $m_0 = 1$, we have $\nu_1 = 1+\epsilon$ for some $\epsilon > 0$, and for $m > 1$, $\left(\log^{\circ(m-1)}(t)\right)^{-\nu_m} \leq (\log(t))^{|\nu_m|}$. Thus, denoting, $\tilde{\nu} = \sum_{m=2}^{M+1}|\nu_m|$, we have $\int_{t_1}^{\infty} f(t)\mathrm{d}t \leq \int_{t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}}}{t^{1+\epsilon}}\mathrm{d}t$.

Thus, for any m_0, we only need to show that for all $t_1 > 0, \epsilon > 0$ and $\tilde{\nu} > 0$, $\int_{t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}}}{t^{1+\epsilon}}\mathrm{d}t < \infty$.

Let us now look at $\int_{t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}}}{t^{1+\epsilon}}\mathrm{d}t$. using $u = (\log(t))^{\tilde{\nu}}$ and $\mathrm{d}v = \frac{1}{t^{1+\epsilon}}$, we have $\mathrm{d}u = \tilde{\nu}t^{-1}(\log(t))^{\tilde{\nu}-1}$ and $v = -\frac{1}{\epsilon t^{\epsilon}}$. Using integration by parts, $\int u\mathrm{d}v = uv - \int v\mathrm{d}u$, we have

$$\int\frac{(\log(t))^{\tilde{\nu}}}{t^{1+\epsilon}}\mathrm{d}t = -\frac{(\log(t))^{\tilde{\nu}}}{\epsilon t^{\epsilon}} + \frac{\tilde{\nu}}{\epsilon}\int\frac{(\log(t))^{\tilde{\nu}-1}}{t^{1+\epsilon}}\mathrm{d}t$$

Recursing the above equation K times such that $\tilde{\nu} - K < 0$, we have positive constants $c_0, c_1, \ldots c_K > 0$ independent of t, such that

$$\begin{aligned}\int_{t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}}}{t^{1+\epsilon}}\mathrm{d}t &= \left[-\sum_{k=0}^{K-1}\frac{c_k(\log(t))^{\tilde{\nu}-k}}{\epsilon t^{\epsilon}}\right]_{t=t_1}^{\infty} + c_K\int_{t=t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}-K}}{t^{1+\epsilon}}\mathrm{d}t \\ &\overset{(1)}{=} \sum_{k=0}^{K-1}\frac{c_k(\log(t_1))^{\tilde{\nu}-k}}{\epsilon t_1^{\epsilon}} + c_K\int_{t=t_1}^{\infty}\frac{(\log(t))^{\tilde{\nu}-K}}{t^{1+\epsilon}}\mathrm{d}t \\ &\overset{(2)}{\leq} \sum_{k=0}^{K-1}\frac{c_k(\log(t_1))^{\tilde{\nu}-k}}{\epsilon t_1^{\epsilon}} + c_K\int_{t=t_1}^{\infty}\frac{1}{t^{1+\epsilon}}\overset{(3)}{} \\ &= \sum_{k=0}^{K-1}\frac{c_k(\log(t_1))^{\tilde{\nu}-k}}{\epsilon t_1^{\epsilon}}y + \frac{c_K}{\epsilon t_1^{\epsilon}} < \infty\end{aligned} \tag{77}$$

where (1) follows as $\sum_{k=0}^{K-1}\frac{c_k(\log(t))^{\tilde{\nu}-k}}{\epsilon t^{\epsilon}} \overset{t\to\infty}{\to} 0$, (2) follows as K is chosen such that $\tilde{\nu} - K < 0$ and hence for all $t > 0$, $(\log(t))^{\tilde{\nu}-K} < 1$. This completes the proof of the lemma. ■

Lemma 14 *Let* $\kappa_1(t)$ *and* $\kappa_2(t)$ *be functions in* L_1, *then*

$$\left(\mathbf{r}(t+1) - \mathbf{r}(t)\right)^{\top}\mathbf{r}(t) \leq \kappa_1(t)\|\mathbf{r}(t)\| + \kappa_2(t) \tag{73}$$

Proof Recall that we defined

$$\mathbf{r}(t) = \mathbf{w}(t) - \mathbf{q}(t) \tag{78}$$

where

$$\mathbf{q}(t) = \sum_{m=1}^{M}\left[\hat{\mathbf{w}}_m\log^{\circ m}(t) + \mathbf{h}_m(t)\right]. \tag{79}$$

$$\mathbf{h}_m(t) = \tilde{\mathbf{w}}_m + \sum_{k=1}^{m-1}\frac{\check{\mathbf{w}}_{k,m}}{\prod_{r=k}^{m-1}\log^{\circ r}(t)} \tag{80}$$

with $\hat{\mathbf{w}}_m$, $\tilde{\mathbf{w}}_m$ and $\check{\mathbf{w}}_{k,m}$ defined in eqs. 58, 62 and 64, respectively. We note that

$$\|\mathbf{q}(t+1) - \mathbf{q}(t) - \dot{\mathbf{q}}(t)\| \leq C_q t^{-2} \in L_1 \tag{81}$$

where

$$\dot{\mathbf{q}}(t) = \sum_{m=1}^{M}\hat{\mathbf{w}}_m\frac{1}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)} + \dot{\mathbf{h}}_m(t)\,. \tag{82}$$

Additionally, we define C_h, C_h' so that

$$\|\mathbf{h}_m(t)\| \leq \|\tilde{\mathbf{w}}_m\| + \sum_{k=1}^{m}\|\check{\mathbf{w}}_{k,m}\| \leq C_h \tag{83}$$

and

$$\left\|\dot{\mathbf{h}}_m(t)\right\| \leq \frac{C_h'}{t\left(\prod_{r=1}^{m-2}\log^{\circ r}(t)\right)\left(\log^{\circ(m-1)}(t)\right)^2} \in L_1\,. \tag{84}$$

We wish to calculate

$$\begin{aligned}
&(\mathbf{r}(t+1)-\mathbf{r}(t))^\top \mathbf{r}(t)\\
&\overset{(1)}{=}[\mathbf{w}(t+1)-\mathbf{w}(t)-[\mathbf{q}(t+1)-\mathbf{q}(t)]]^\top \mathbf{r}(t)\\
&\overset{(2)}{=}[-\eta\nabla\mathcal{L}(\mathbf{w}(t))-\dot{\mathbf{q}}(t)]^\top \mathbf{r}(t)-[\mathbf{q}(t+1)-\mathbf{q}(t)-\dot{\mathbf{q}}(t)]^\top \mathbf{r}(t)
\end{aligned} \tag{85}$$

where in (1) we used eq. 78 and in (2) we used the definition of GD in eq. 2. We can bound the second term using Cauchy-Shwartz inequality and eq. 81:

$$[\mathbf{q}(t+1)-\mathbf{q}(t)-\dot{\mathbf{q}}(t)]^\top \mathbf{r}(t) \leq \|\mathbf{q}(t+1)-\mathbf{q}(t)-\dot{\mathbf{q}}(t)\|\,\|\mathbf{r}(t)\| \leq C_q t^{-2}\|\mathbf{r}(t)\|\,.$$

Next, we examine the second term in eq. 85

$$\begin{aligned}
&[-\eta\nabla\mathcal{L}(\mathbf{w}(t))-\dot{\mathbf{q}}(t)]^\top \mathbf{r}(t)\\
&=\left[-\eta\sum_{n=1}^{N}\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n-\dot{\mathbf{q}}(t)\right]^\top \mathbf{r}(t)\\
&\overset{(1)}{=}-\sum_{m=1}^{M}\dot{\mathbf{h}}_m(t)^\top \mathbf{r}(t)-\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m^+}\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}(t)\\
&+\left[\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n-\sum_{m=1}^{M}\hat{\mathbf{w}}_m\frac{1}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]^\top \mathbf{r}(t)\,,
\end{aligned} \tag{86}$$

where in (1) recall from eq. 59 that $\mathcal{S}_m, \mathcal{S}_m^+$ are mutually exclusive and $\cup_{m=1}^M \mathcal{S}_m \cup \mathcal{S}_m^+ = [N]$.

Next we upper bound the three terms in eq. 86.

To bound the first term in eq. 86 we use Cauchy-Shartz, and eq. 84.

$$\sum_{m=1}^{M}\dot{\mathbf{h}}_m(t)^\top \mathbf{r}(t) \leq \sum_{m=1}^{M}\left\|\dot{\mathbf{h}}_m(t)\right\|\|\mathbf{r}(t)\| \leq \frac{MC_h'}{t\left(\prod_{r=1}^{m-2}\log^{\circ r}(t)\right)\left(\log^{\circ(m-1)}(t)\right)^2}\|\mathbf{r}(t)\|$$

In bounding the second term in eq. 86, note that for tight exponential tail loss, since $\mathbf{w}(t)^\top\mathbf{x}_n \to \infty$, for large enough t_0, we have $-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n) \leq (1+\exp(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n))\exp(-\mathbf{w}(t)^\top\mathbf{x}_n) \leq 2\exp(-\mathbf{w}(t)^\top\mathbf{x}_n)$ for all $t > t_0$. The first term in eq. 86 can be bounded by the following set of

inequalities, for $t > t_0$,

$$\begin{aligned}
&\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m^+}-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}(t) \leq \eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m^+:\mathbf{x}_n^\top\mathbf{r}(t)\geq 0}-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}(t)\\
&\overset{(1)}{\leq}2\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m^+:\mathbf{x}_n^\top\mathbf{r}(t)\geq 0}\exp\left(-\sum_{l=1}^{M}\left[\hat{\mathbf{w}}_l^\top\mathbf{x}_n\log^{\circ l}(t)+\mathbf{x}_n^\top\mathbf{h}_l(t)\right]-\mathbf{x}_n^\top\mathbf{r}(t)\right)\mathbf{x}_n^\top\mathbf{r}(t)\\
&\overset{(2)}{\leq}2\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m^+:\mathbf{x}_n^\top\mathbf{r}(t)\geq 0}\exp\left(-\sum_{l=1}^{M}\left[\hat{\mathbf{w}}_l^\top\mathbf{x}_n\log^{\circ l}(t)+\mathbf{x}_n^\top\mathbf{h}_l(t)\right]\right)\\
&\overset{(3)}{\leq}2\eta\sum_{m=1}^{M}\left|\mathcal{S}_m^+\max_{n\in\mathcal{S}_m^+}\right|\exp\left(M\|\mathbf{x}_n\|C_h\right)\exp\left(-\sum_{l=1}^{M}\hat{\mathbf{w}}_l^\top\mathbf{x}_n\log^{\circ l}(t)\right)\\
&\overset{(4)}{\leq}\begin{cases}\sum_{m=1}^{M}\frac{2\eta\left|\mathcal{S}_m^+\right|\exp\left(M\max_{n\in\mathcal{S}_m^+}\|\mathbf{x}_n\|C_h\right)}{t\left(\prod_{k=1}^{m-1}\log^{\circ k}(t)\right)\left(\log^{\circ m-1}(t)\right)^{\theta_m}\left(\prod_{k=m}^{M-1}(\log^{\circ m}(t))^{\hat{\mathbf{w}}_k^\top\mathbf{x}_n}\right)} & \text{if } M>1\\ \frac{2\eta\left|\mathcal{S}_1^+\right|\exp(\max_n\|\mathbf{x}_n\|C_h)}{t^{\theta_1}} & \text{if } M=1\end{cases} \in L_1.
\end{aligned} \tag{87}$$

where in (1) we used eqs. 78 and 79, in (2) we used that $\forall x: xe^{-x} \leq 1$ and $\mathbf{x}_n^\top\mathbf{r}(t) \geq 0$, (3) we used eq. 83 and in (4) we denoted $\theta_m = \min_{n\in\mathcal{S}_m^+}\hat{\mathbf{w}}_m^\top\mathbf{x}_n > 1$ and the last line is integrable based on Lemma 16.

Next, we bound the last term in eq. 86. For exponential tailed losses (Assumption 3), since $\mathbf{w}(t)^\top\mathbf{x}_n \to \infty$, we have positive constants $\mu_-, \mu_+ > 0$, t_- and t_+ such that $\forall n$

$$\begin{aligned}
&\forall t > t_+ : -\ell'\left(\mathbf{w}(t)^\top\mathbf{x}_n\right) \leq \left(1+\exp\left(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n\right)\right)\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_n\right)\\
&\forall t > t_- : -\ell'\left(\mathbf{w}(t)^\top\mathbf{x}_n\right) \geq \left(1-\exp\left(-\mu_-\mathbf{w}(t)^\top\mathbf{x}_n\right)\right)\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_n\right)
\end{aligned}$$

We define $\gamma_n(t)$ as

$$\gamma_n(t) = \begin{cases}(1+\exp(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n) & \text{if } \mathbf{r}(t)^\top\mathbf{x}_n \geq 0\\ (1-\exp(-\mu_-\mathbf{w}(t)^\top\mathbf{x}_n) & \text{if } \mathbf{r}(t)^\top\mathbf{x}_n < 0\end{cases}. \tag{88}$$

This implies $t > \max(t_+, t_-)$, $-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}(t) \leq \gamma_n(t)\exp\left(-\mathbf{w}^\top(t)\,\mathbf{x}_n\right)\mathbf{x}_n$.

From this result, we have the following set of inequalities:

$$\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}\,(t)\le\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\gamma_n(t)\exp\left(-\mathbf{w}\,(t)^\top\mathbf{x}_n\right)\mathbf{x}_n^\top\mathbf{r}\,(t)$$

$$\overset{(1)}{=}\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\gamma_n(t)\exp\left(-\sum_{l=1}^{M}\left[\hat{\mathbf{w}}_l^\top\mathbf{x}_n\log^{ol}(t)+\mathbf{x}_n^\top\breve{\mathbf{w}}_l+\sum_{k=1}^{l-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{k,l}}{\prod_{r=k}^{l-1}\log^{or}(t)}\right]-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)$$

$$\overset{(2)}{=}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\exp\left(-\sum_{k=1}^{m}\sum_{l=k+1}^{M}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{k,l}}{\prod_{r=k}^{l-1}\log^{or}(t)}\right)$$

$$\overset{(3)}{=}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\exp\left(-\sum_{l=m+1}^{M}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l}}{\prod_{r=m}^{l-1}\log^{or}(t)}\right)\psi_m\,(t)$$

$$\le\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\psi_m\,(t)\left[\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right.$$
$$\left.+\exp\left(-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)-\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right]\tag{89}$$

where in (1) we used eqs. 78 and 79, and in (2) we used $\mathbf{P}_{k-1}\breve{\mathbf{w}}_{k,m}=0$ from eq. 65 (so $\mathbf{x}_n^\top\breve{\mathbf{w}}_{k,l}=0$ if $m<k$) and in (3) defined

$$\psi_m\,(t)=\exp\left(-\sum_{k=1}^{m-1}\sum_{l=k+1}^{M}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{k,l}}{\prod_{r=k}^{l-1}\log^{or}(t)}\right).\tag{90}$$

Note $\exists t_\psi$ such that $\forall t>t_\psi$, we can bound $\psi_m\,(t)$ by

$$\exp\left(\frac{-M\max_n\|\mathbf{x}_n\|\,C_h}{\log^{\circ(m-1)}(t)}\right)\le\psi_m\,(t)\le 1\,.\tag{91}$$

Thus, the third term in 86 is given by

$$\eta\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}-\ell'(\mathbf{w}(t)^\top\mathbf{x}_n)\,\mathbf{x}_n^\top\mathbf{r}\,(t)-\sum_{m=1}^{M}\frac{\tilde{\mathbf{w}}_m^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}$$
$$\overset{(1)}{\le}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\psi_m\,(t)\left[\exp\left(-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right.$$
$$\left.-\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right]$$
$$+\sum_{m=1}^{M}\left[\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\psi_m\,(t)\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right.$$
$$\left.-\frac{\mathbf{r}(t)^\top\hat{\mathbf{w}}_m}{t\prod_{r=1}^{m-1}\log^{or}(t)}\right],\tag{92}$$

where (1) follows from the bound in eq. 89.

We examine the first term in eq. 92

$$\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\psi_m\,(t)$$
$$\cdot\left[\exp\left(-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)-\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right]$$

$\forall t>t_1>t_\psi$, where we will determine t_1 later. We have the following for all $m\in[M]$

$$\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\,(t)\right)\mathbf{x}_n^\top\mathbf{r}\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\psi_m\,(t)$$
$$\cdot\left[\exp\left(-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)-\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right]$$
$$\overset{(1)}{\le}\sum_{\substack{n\in\mathcal{S}_m:\\ \mathbf{x}_n^\top\mathbf{r}(t)\ge 0}}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\psi_m\,(t)}{t\prod_{r=1}^{m-1}\log^{or}(t)}\left[\exp\left(-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)-\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)\right]$$
$$\overset{(2)}{\le}\sum_{\substack{n\in\mathcal{S}_m:\\ \mathbf{x}_n^\top\mathbf{r}(t)\ge 0}}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)(t)}{t\prod_{r=1}^{m-1}\log^{or}\psi_m\,(t)}\left(\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}\right)^2\in L_1\,,\tag{93}$$

where we set $t_1>0$ such that $\forall t>t_1$ the term in the square bracket is positive and

$$\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\breve{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}(t)}>-1\,,$$

in (1) we used that since $e^{-x}\ge 1-x$, and also from using $e^{-x}x\le 1$ and in (2) we use that $\forall x\ge -1$ we have that $e^{-x}\le 1-x+x^2$ and $\psi_m\,(t)\le 1$ from eq. 91.

We examine the second term in eq. 92 using the decomposition of $\hat{\mathbf{w}}_m$ from eq. 66

$$\sum_{m=1}^{M}\left[\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\psi_m\left(t\right)\left(1-\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\tilde{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}\left(t\right)}\right)-\frac{\mathbf{x}_n^\top\hat{\mathbf{w}}_m}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\right]$$

$$\overset{(1)}{=}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\left(\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\psi_m\left(t\right)-1\right)$$

$$-\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\psi_m(t)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\sum_{l=m}^{M-1}\frac{\mathbf{x}_n^\top\tilde{\mathbf{w}}_{m,l+1}}{\prod_{r=m}^{l}\log^{or}\left(t\right)}$$

$$+\sum_{m=1}^{M}\sum_{k=1}^{m-1}\sum_{n\in\mathcal{S}_k}\frac{\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\mathbf{x}_n^\top\tilde{\mathbf{w}}_{k,m}}{\prod_{r=1}^{m-1}t\log^{or}\left(t\right)}$$

$$\overset{(2)}{=}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\frac{\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\left(\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\psi_m\left(t\right)-1\right)$$

$$-\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\sum_{l=m}^{M-1}\frac{\eta\gamma_n(t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\psi_m(t)\mathbf{x}_n^\top\tilde{\mathbf{w}}_{m,l+1}}{t\prod_{r=1}^{l}\log^{or}\left(t\right)}$$

$$+\sum_{k=1}^{M}\sum_{n\in\mathcal{S}_k}\sum_{m=k}^{M-1}\frac{\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\mathbf{x}_n^\top\tilde{\mathbf{w}}_{k,m+1}}{\prod_{r=1}^{m}t\log^{or}\left(t\right)}$$

$$\overset{(3)}{=}\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\left[\frac{1}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}-\sum_{k=m}^{M-1}\frac{\mathbf{x}_n^\top\tilde{\mathbf{w}}_{m,k+1}}{t\prod_{r=1}^{k}\log^{or}\left(t\right)}\right]\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\left(\gamma_n(t)\psi_m\left(t\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)$$

$$:=\sum_{m=1}^{M}\sum_{n\in\mathcal{S}_m}\Gamma_{m,n}(t), \tag{94}$$

where in (1) we used eq. 66, in (2) we re-arranged the order of summation in the last term, and in (3) we just use a change of variables.

Next, we examine $\Gamma_{m,n}(t)$ for each m and $n\in\mathcal{S}_m$ in eq. 94. Note that, $\exists t_2>t_\psi$ such that $\forall t>t_2$ we have

$$\left|\sum_{k=m}^{M-1}\frac{\mathbf{x}_n^\top\tilde{\mathbf{w}}_{m,k+1}}{t\prod_{r=1}^{k}\log^{or}\left(t\right)}\right|\leq\frac{0.5}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}.$$

In this case, $\forall t>t_2$

$$\Gamma_{m,n}(t)\overset{(1)}{\leq}\eta\left[\frac{\kappa(n,t)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\right]\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right), \tag{95}$$

where in (1) follows from the definition of t_2, wherein

$$\kappa_n(t)=\begin{cases}1.5 & \text{if }\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)>0\\ 0.5 & \text{if }\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)<0\end{cases}.$$

1. First, if $\mathbf{x}_n^\top\mathbf{r}\left(t\right)>0$, then $\gamma_n(t)=(1+\exp(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n))>0$.

We further divide into two cases. In the following C_0, C_1 are some constants independent of t.

(a) If $\left|\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right|>C_0t^{-0.5\mu_+}$, then we have the following

$$\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)$$

$$\overset{(1)}{\leq}\left(1+\exp\left(-\mu_+\sum_{l=1}^{M}\left[\hat{\mathbf{w}}_l^\top\mathbf{x}_n\log^{ol}(t)+\mathbf{h}_l^\top\mathbf{x}_n\right]\right)\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)$$

$$\overset{(2)}{\leq}\left(1+\frac{\exp(\mu_+C_h\left\|\mathbf{x}_n\right\|)}{\left(t\prod_{r=1}^{m-1}\log^{or}\left(t\right)\right)^{\mu_+}}\right)\exp(-C_0t^{-0.5\mu_+})$$

$$\overset{(3)}{\leq}\left(1+C_1t^{-\mu_+}\right)\left(1-C_0t^{-0.5\mu_+}+0.5C_0^2t^{-\mu_+}\right),\forall t>t'_+$$

$$\leq 1-C_0t^{-0.5\mu_+}\left(1+C_1t^{-\mu_+}\right)+0.5C_0^2t^{-\mu_+}\left(1+C_1t^{-\mu_+}\right)\overset{(4)}{\leq}1,\forall t>t''_+, \tag{96}$$

where in (1), we use $\psi_m(t)\leq1$ from eq. 91 and using eq. 78, in (2) we used bound on $\mathbf{h}_m$ from eq. 83, in (3) for some large enough $t'_+>t_+$, we have $\frac{\exp(\mu_+C_h\|\mathbf{x}_n\|)}{\left(\prod_{r=1}^{m-1}\log^{or}(t)\right)^{\mu_+}}\leq C_1$, and for the second term we used the inequality $e^{-x}\leq1-x+0.5x^2$ for $x>0$, and (4) holds asymptotically for $t>t''_+$ for large enough $t''_+>t'_+$ as $C_0t^{-0.5\mu_+}$ converges slower than $0.5C_0^2t^{-\mu_+}$ to 0.

Thus, using eq. 96 in eq. 95, $\forall t>\max\left(t_2,t''_+\right)$, we have

$$\Gamma_{m,n}(t)\leq\left[\frac{\eta\kappa(n,t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\right]\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\leq0$$

(b) If $0<\mathbf{x}_n^\top\mathbf{r}\left(t\right)<C_0t^{-0.5\mu_+}$, then we have the following: $\psi_m(t)\leq1$ from eq. 91, $\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\leq1$ as $\mathbf{x}_n^\top\mathbf{r}\left(t\right)>0$, and since $\mathbf{w}(t)^\top\mathbf{x}_n\to\infty$, for large enough $t>t'''_+$, $\gamma_n(t)=\left(1+\exp\left(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n\right)\right)\leq2$

This gives us, $\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}\left(t\right)\leq\mathbf{x}_n^\top\mathbf{r}\left(t\right)\leq C_0t^{-0.5\mu_+}$, and using this in eq. 95, $\forall t>\max\left(t_2,t'_+\right)$

$$\Gamma_{m,n}(t)\leq\left[\frac{\eta\kappa(n,t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\right]C_0t^{-0.5\mu_+}\in L_1.$$

2. Second, if $\mathbf{x}_n^\top\mathbf{r}\left(t\right)\leq0$, then $\gamma_n(t)=(1-\exp(-\mu_-\mathbf{w}(t)^\top\mathbf{x}_n))\in(0,1)$. We again divide into following special cases.

(a) If $\left|\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right|\leq C_0\left(\log^{o(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}$, where $\tilde{\mu}_-=\min\left(\mu_-,1\right)$, then we have

$$\Gamma_{m,n}(t)\leq\left[\frac{1.5\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{or}\left(t\right)}\right]\left(1-\gamma_n(t)\psi_m\left(t\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\right)\left|\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right|$$

$$\overset{(1)}{\leq}\left[\frac{1.5\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-2}\log^{or}\left(t\right)}\right]C_0\left(\log^{o(m-1)}(t)\right)^{-1-0.5\tilde{\mu}_-}\in L_1.$$

where in (1) we used that $\left(1-\gamma_n(t)\psi_m\left(t\right)\exp\left(-\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right)\right)<1$ and $\left|\mathbf{x}_n^\top\mathbf{r}\left(t\right)\right|\leq C_0\left(\log^{o(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}$.

(b) If $\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)<1$, then, from eq. 91

$$\frac{-M\max_n\|\mathbf{x}_n\|C_h}{\log^{\circ(m-1)}(t)}\le\log\psi_m(t)<\mathbf{x}_n^\top\mathbf{r}(t)\,. \tag{97}$$

In this case, since $\gamma_n(t)=1-\exp(-\mathbf{w}(t)^\top\mathbf{x}_n)<1$, we also have $\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)<1$, and hence $\left(\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)-1\right)\mathbf{x}_n^\top\mathbf{r}(t)>0$. Thus, $\forall t>t_2$, in 95, $\kappa_n(t)=1.5$, and we have

$$\Gamma_{m,n}(t)\le\left[\frac{1.5\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]\left(1-\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)\right)\left|\mathbf{x}_n^\top\mathbf{r}(t)\right|$$
$$\overset{(1)}{\le}\left[\frac{1.5\eta\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]\frac{M\max_n\|\mathbf{x}_n\|C_h}{\log^{\circ(m-1)}(t)}\le\frac{C_2}{t\prod_{r=1}^{m-2}\log^{\circ r}(t)\left(\log^{\circ(m-1)}(t)\right)^2}\in L_1,$$

where (1) follows from $\left(1-\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)\right)<1$ and the bound on $|\mathbf{x}_n^\top\mathbf{r}(t)|=-\mathbf{x}_n^\top\mathbf{r}(t)$ from eq. 97.

(c) If $\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)>1$, and $\left|\mathbf{x}_n^\top\mathbf{r}(t)\right|>C_0\left(\log^{\circ(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}$, where $\tilde{\mu}_=\min(1,\mu_-)$.
Since, $\mathbf{x}_n^\top\mathbf{w}(t)\to\infty$ and $\psi_m(t)\to1$ from eq. 90, for large enough $t'_->t_-$, we have $\forall t>t'_-$, $\psi_m(t)>0.5$ and $\gamma_n(t)=(1-\exp(-\mu_-\mathbf{x}_n^\top\mathbf{w}(t)))>0.5$. Let $\tau>\max(4,t'_-)$ be an arbitrarily large constant. For all $t>\tau$, if $\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)>\tau\ge4$, then $\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)>0.25\tau\ge1$.
On the other hand, if there exists $t>\tau\ge4$, such that $\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)<\tau$, then for some constants C_1,C_2 we have the following

(i) $\exp(-\mathbf{x}_n^\top\mathbf{r}(t))=\exp(|\mathbf{x}_n^\top\mathbf{r}(t)|)\ge\left(1+C_0\left(\log^{\circ(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}\right)$, since $e^x>1+x$ for all x,

(ii) $\psi_m(t)\ge\exp\left(-C_1\left(\log^{\circ(m-1)}(t)\right)^{-1}\right)\ge\left(1-C_1\left(\log^{\circ(m-1)}(t)\right)^{-1}\right)$ from eq. 91 and again using $e^x>1+x$ for all x,

(iii)

$$\gamma_n(t)=\left(1-\left[\frac{\exp(-\mathbf{h}_l(t)^\top\mathbf{x}_n)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]^{\mu_-}\right)$$
$$\ge\left(1-\left[\frac{\exp(-C_h\|x_n\|)\tau}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]^{\mu_-}\right)\ge\left(1-\left(C_2\log^{\circ(m-1)}(t)\right)^{-\mu_-}\right),\forall t>t''_-$$

where the last inequality follows as for large enough $t''_->t'_-$, we have $\frac{\exp(-C_h\|x_n\|)\tau}{t\prod_{r=1}^{m-2}\log^{\circ r}(t)}\le C_2$.

Using the above inequalities, we have

$$\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)$$
$$\ge\left(1+C_0\left(\log^{\circ(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}\right)\left(1-C_1\left(\log^{\circ(m-1)}(t)\right)^{-1}\right)\left(1-C_2\left(\log^{\circ(m-1)}(t)\right)^{-\mu_-}\right)$$
$$\overset{(1)}{\ge}1+C_0\left(\log^{\circ(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}-C_1\left(\log^{\circ(m-1)}(t)\right)^{-1}-C_2\left(\log^{\circ(m-1)}(t)\right)^{-\mu_-}$$
$$-C_0C_2\left(\log^{\circ(m-1)}(t)\right)^{-\mu_-0.5\tilde{\mu}_-}-C_0C_1\left(\log^{\circ(m-1)}(t)\right)^{-1-0.5\tilde{\mu}_-}\overset{(2)}{\ge}1,\forall t>t'''_-, \tag{98}$$

where in (1) we dropped the other positive terms, and (2) follows for large enough $t'''_->t''_-$ as the $C_0\log\left(\log^{\circ(m-1)}(t)\right)^{-0.5\tilde{\mu}_-}$ converges to 0 more slowly than the other negative terms. Finally, using eq. 98 in eq. 95, we have for all $t>\max(t_2,\tau,t_\psi,t'''_-)$

$$\Gamma_{m,n}(t)\le\left[\frac{\eta\kappa(n,t)\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}\right)}{t\prod_{r=1}^{m-1}\log^{\circ r}(t)}\right]\left(1-\gamma_n(t)\psi_m(t)\exp\left(-\mathbf{x}_n^\top\mathbf{r}(t)\right)\right)\left|\mathbf{x}_n^\top\mathbf{r}(t)\right|\le0 \tag{99}$$

Collecting all the terms from the above special cases, and substituting back into eq. 85, we note that all terms are either negative, in L_1, or of the form $f(t)\|\mathbf{r}(t)\|$, where $f(t)\in L_1$, thus proving the lemma. ■

C.4. Proof of the existence and uniqueness of the solution to eqs. 62-63

We wish to prove that $\forall m\ge1$:

$$\sum_{n\in\mathcal{S}_m}\exp\left(-\sum_{k=1}^m\tilde{\mathbf{w}}_k^\top\mathbf{x}_n\right)\bar{\mathbf{P}}_{m-1}\mathbf{x}_n=\hat{\mathbf{w}}_m\,, \tag{100}$$

such that

$$\mathbf{P}_{m-1}\tilde{\mathbf{w}}_m=0\,\text{and}\,\bar{\mathbf{P}}_m\tilde{\mathbf{w}}_m=0, \tag{101}$$

we have a unique solution. From eq. 101, we can modify eq. 100 to

$$\sum_{n\in\mathcal{S}_m}\exp\left(-\sum_{k=1}^m\tilde{\mathbf{w}}_k^\top\bar{\mathbf{P}}_{k-1}\mathbf{x}_n\right)\bar{\mathbf{P}}_{m-1}\mathbf{x}_n=\hat{\mathbf{w}}_m\,,.$$

To prove this, without loss of generality, and with a slight abuse of notation, we will denote $\mathcal{S}_m$ as $\mathcal{S}_1$, $\bar{\mathbf{P}}_{m-1}\mathbf{x}_n$ as $\mathbf{x}_n$ and $\beta_n=\exp\left(-\sum_{k=1}^{m-1}\tilde{\mathbf{w}}_k^\top\bar{\mathbf{P}}_{k-1}\mathbf{x}_n\right)$, so we can write the above equation as

$$\sum_{n\in\mathcal{S}_1}\mathbf{x}_n\beta_n\exp\left(-\mathbf{x}_n^\top\tilde{\mathbf{w}}_1\right)=\hat{\mathbf{w}}_1$$

In the following Lemma 17 we prove this equation $\forall\boldsymbol{\beta}\in\mathbb{R}_{>0}^{|\mathcal{S}_1|}$.

Lemma 17 *$\forall \boldsymbol{\beta} \in \mathbb{R}_{>0}^{|\mathcal{S}_1|}$ we can find a unique $\tilde{\mathbf{w}}$ such that*

$$\sum_{n\in\mathcal{S}_1} \mathbf{x}_n \beta_n \exp\left(-\mathbf{x}_n^\top \tilde{\mathbf{w}}_1\right) = \hat{\mathbf{w}}_1 \tag{102}$$

and for $\forall \mathbf{z} \in \mathbb{R}^d$ such that $\mathbf{z}^\top \mathbf{X}_{\mathcal{S}_1} = 0$ we would have $\tilde{\mathbf{w}}_1^\top \mathbf{z} = 0$.

Proof Let $K = \operatorname{rank}(\mathbf{X}_{\mathcal{S}_1})$. Let and $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_d] \in \mathbb{R}^{d\times d}$ be a set of orthonormal vectors (*i.e.*, $\mathbf{U}\mathbf{U}^\top = \mathbf{U}^\top\mathbf{U} = \mathbf{I}$) such that $\mathbf{u}_1 = \hat{\mathbf{w}}_1 / \|\hat{\mathbf{w}}_1\|$, and

$$\forall \mathbf{z} \neq 0, \forall n \in \mathcal{S}_1 : \mathbf{z}^\top [\mathbf{u}_1, \ldots, \mathbf{u}_K]^\top \mathbf{x}_n \neq 0\,, \tag{103}$$

while

$$\forall i > K : \forall n \in \mathcal{S}_1 : \mathbf{u}_i^\top \mathbf{x}_n = 0\,. \tag{104}$$

In other words, $\mathbf{u}_1$ is in the direction of $\hat{\mathbf{w}}_1$, $[\mathbf{u}_1, \ldots, \mathbf{u}_K]$ are in the space spanned by the columns of $\mathbf{X}_{\mathcal{S}_1}$, and $[\mathbf{u}_{K+1}, \ldots, \mathbf{u}_d]$ are orthogonal to the columns of $\mathbf{X}_{\mathcal{S}_1}$.

We define $\mathbf{v}_n = \mathbf{U}^\top \mathbf{x}_n$ and $\mathbf{s} = \mathbf{U}^\top \tilde{\mathbf{w}}_1$. Note that $\forall i > K : v_{i,n} = 0\, \forall n \in \mathcal{S}_1$ from eq. 104, and $\forall i > K : s_i = 0$, since for $\forall \mathbf{z} \in \mathbb{R}^d$ such that $\mathbf{z}^\top \mathbf{X}_{\mathcal{S}_1} = 0$ we would have $\tilde{\mathbf{w}}_1^\top \mathbf{z} = 0$. Lastly, equation 102 becomes

$$\sum_{n\in\mathcal{S}_1} \mathbf{x}_n \beta_n \exp\left(-\sum_{j=1}^K s_j v_{j,n}\right) = \hat{\mathbf{w}}_1\,. \tag{105}$$

Multiplying by $\mathbf{U}^\top$ from the left, we obtain

$$\forall i \leq K : \sum_{n\in\mathcal{S}_1} v_{i,n}\beta_n \exp\left(-\sum_{j=1}^K s_j v_{j,n}\right) = \mathbf{u}_i^\top \hat{\mathbf{w}}_1\,.$$

Since $\mathbf{u}_1 = \hat{\mathbf{w}}_1 / \|\hat{\mathbf{w}}_1\|$, we have that

$$\forall i \leq K : \sum_{n\in\mathcal{S}_1} v_{i,n}\beta_n \exp\left(-\sum_{j=1}^K s_j v_{j,n}\right) = \|\hat{\mathbf{w}}_1\|\, \delta_{i,1}\,. \tag{106}$$

We recall that $v_{1,n} = \hat{\mathbf{w}}_1^\top \mathbf{x}_n / \|\hat{\mathbf{w}}_1\| = 1/\|\hat{\mathbf{w}}_1\|, \forall n \in \mathcal{S}_1$. Given $\{s_j\}_{j=2}^K$, we examine eq. 106 for $i = 1$,

$$\exp\left(-\frac{s_1}{\|\hat{\mathbf{w}}_1\|}\right)\left[\sum_{n\in\mathcal{S}_1} \beta_n \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right)\right] = \|\hat{\mathbf{w}}_1\|^2\,.$$

This equation always has the unique solution

$$s_1 = \|\hat{\mathbf{w}}_1\| \log\left[\|\hat{\mathbf{w}}_1\|^{-2} \sum_{n\in\mathcal{S}_1} \beta_n \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right)\right], \tag{107}$$

given $\{s_j\}_{j=2}^K$. Next, we similarly examine eq. 106 for $2 \leq i \leq K$ as a function of s_i

$$\sum_{n\in\mathcal{S}_1} \beta_n v_{i,n} \exp\left(-s_1/\|\hat{\mathbf{w}}_1\| - \sum_{j=2}^K s_j v_{j,n}\right) = 0\,. \tag{108}$$

multiplying by $\exp(s_1/\|\hat{\mathbf{w}}_1\|)$ we obtain

$$0 = \sum_{n\in\mathcal{S}_1} \beta_n v_{i,n} \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right) = -\frac{\partial}{\partial s_i}\left[E(s_2, \ldots, s_K)\right]\,,$$

where we defined

$$E(s_2, \ldots, s_K) = \sum_{n\in\mathcal{S}_1} \beta_n \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right)\,.$$

Therefore, any critical point of $E(s_2, \ldots, s_K)$ would be a solution of eq. 108 for $2 \leq i \leq K$, and substituting this solution into eq. 107 we obtain s_1. Since $\beta_n > 0$, $E(s_2, \ldots, s_K)$ is a convex function, as positive linear combination of convex function (exponential). Therefore, any finite critical point is a global minimum. All that remains is to show that a finite minimum exists and that it is unique.

From the definition of $\mathcal{S}_1$, $\exists \boldsymbol{\alpha} \in \mathbb{R}_{>0}^{|\mathcal{S}_1|}$ such that $\hat{\mathbf{w}}_1 = \sum_{n\in\mathcal{S}_1} \alpha_n \mathbf{x}_n$. Multiplying this equation by $\mathbf{U}^\top$ we obtain that $\exists \boldsymbol{\alpha} \in \mathbb{R}_{>0}^{|\mathcal{S}_1|}$ such that $2 \leq i \leq K$

$$\sum_{n\in\mathcal{S}_1} v_{i,n}\alpha_n = 0\,. \tag{109}$$

Therefore, $\forall (s_2, \ldots, s_K) \neq \mathbf{0}$ we have that

$$\sum_{n\in\mathcal{S}_1} \left(\sum_{j=2}^K s_j v_{j,n}\right) \alpha_n = 0\,. \tag{110}$$

Recall, from eq. 103 that $\forall (s_2, \ldots, s_K) \neq \mathbf{0}, \exists n \in \mathcal{S}_1 : \sum_{j=2}^K s_j v_{j,n} \neq 0$, and that $\alpha_n > 0$. Therefore, eq. 110 implies that $\exists n \in \mathcal{S}_1$ such that $\sum_{j=2}^K s_j v_{j,n} > 0$ and also $\exists m \in \mathcal{S}_1$ such that $\sum_{j=2}^K s_j v_{j,m} < 0$.

Thus, in any direction we take a limit in which $|s_i| \to \infty\ \forall 2 \leq i \leq K$, we obtain that $E(s_2, \ldots, s_K) \to \infty$, since at least one exponent in the sum diverge. Since $E(s_2, \ldots, s_K)$, is a continuous function, it implies it has a finite global minimum. This proves the existence of a finite solution. To prove uniqueness we will show the function is strictly convex, since the hessian is (strictly) positive definite, *i.e.*, that the following expression is strictly positive:

$$\begin{aligned}
&\sum_{i=2}^K \sum_{k=2}^K q_i q_k \frac{\partial}{\partial s_i}\frac{\partial}{\partial s_k} E(s_2, \ldots, s_K)\,. \\
&= \sum_{n\in\mathcal{S}_1} \beta_n \left(\sum_{i=2}^K q_i v_{i,n}\right)\left(\sum_{k=2}^K q_k v_{k,n}\right) \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right) \\
&= \sum_{n\in\mathcal{S}_1} \beta_n \left(\sum_{i=2}^K q_i v_{i,n}\right)^2 \exp\left(-\sum_{j=2}^K s_j v_{j,n}\right)\,.
\end{aligned}$$

the last expression is indeed strictly positive since $\forall \mathbf{q} \neq \mathbf{0}, \exists n \in \mathcal{S}_1 : \sum_{j=2}^K q_j v_{j,n} \neq 0$, from eq. 103. Thus, there exists a unique solution $\tilde{\mathbf{w}}_1$. ■

C.5. Proof of the existence and uniqueness of the solution to eqs. 64-65

Lemma 18 *For $\forall m > k \geq 1$, the equations*

$$\sum_{n\in\mathcal{S}_m} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{P}_{m-1}\mathbf{x}_n = \sum_{k=1}^{m-1}\left[\sum_{n\in\mathcal{S}_k} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{x}_n\mathbf{x}_n^\top\right] \check{\mathbf{w}}_{k,m} \tag{111}$$

under the constraints

$$\mathbf{P}_{k-1}\check{\mathbf{w}}_{k,m} = 0 \,\text{and}\, \bar{\mathbf{P}}_k\check{\mathbf{w}}_{k,m} = 0 \tag{112}$$

have a unique solution $\check{\mathbf{w}}_{k,m}$.

Proof For this proof we denote $\mathbf{X}_{\mathcal{S}_k}$ as the matrix which columns are $\{\mathbf{x}_n | n \in \mathcal{S}_k\}$, the orthogonal projection matrix $\mathbf{Q}_k = \mathbf{P}_k\bar{\mathbf{P}}_{k-1}$where $\mathbf{Q}_k\mathbf{Q}_m = 0\ \forall k \neq m$, $\mathbf{Q}_k\bar{\mathbf{P}}_m = 0\ \forall k < m$, and

$$\forall m : \mathbf{I} = \mathbf{P}_m + \bar{\mathbf{P}}_m = \sum_{k=1}^{m} \mathbf{Q}_k + \bar{\mathbf{P}}_m \tag{113}$$

We will write $\check{\mathbf{w}}_{k,m} = \mathbf{W}_{k,m}\mathbf{u}_{k,m}$, where $\mathbf{u}_{k,m} \in \mathbb{R}^{d_k}$ and $\mathbf{W}_{k,m} \in \mathbb{R}^{d\times d_k}$ is a full rank matrix such that $\mathbf{Q}_k\mathbf{W}_{k,m} = \mathbf{W}_{k,m}$, so

$$\check{\mathbf{w}}_{k,m} = \mathbf{Q}_k\check{\mathbf{w}}_{k,m} = \mathbf{Q}_k\mathbf{W}_{k,m}\mathbf{u}_{k,m}\,. \tag{114}$$

and, furthermore,

$$\operatorname{rank}\left[\mathbf{X}_{\mathcal{S}_k}^\top\mathbf{Q}_k\mathbf{W}_{k,m}\right] = \operatorname{rank}\left(\mathbf{X}_{\mathcal{S}_k}^\top\mathbf{Q}_k\right) = d_k\,. \tag{115}$$

Recall that $\forall m : \bar{\mathbf{P}}_m\mathbf{P}_m = \mathbf{0}$ and $\forall k \geq 1$, $\forall n \in \mathcal{S}_m\ \bar{\mathbf{P}}_{m+k}\mathbf{x}_n = \mathbf{0}$. Therefore, $\forall \mathbf{v} \in \mathbb{R}^d$, $\mathbf{P}_{k-1}\mathbf{Q}_k\mathbf{v} = \mathbf{0}$, $\bar{\mathbf{P}}_k\mathbf{Q}_k\mathbf{v} = \mathbf{0}$. Thus, $\check{\mathbf{w}}_{k,m}$ eq. 114 implies the constraints in eq. 112 hold.

Next, we prove the existence and uniqueness of the solution $\check{\mathbf{w}}_{k,m}$ for each $k = 1,\ldots,m$ separately. We multiply eq. 111 from the left by the identity matrix, decomposed to orthogonal projection matrices as in eq. 113. Since each matrix projects to an orthogonal subspace, we can solve each product separately.

The product with $\bar{\mathbf{P}}_m$ is equal to zero for both sides of the equation. The product with $\mathbf{Q}_k$ is equal to

$$\sum_{n\in\mathcal{S}_m} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{Q}_k\mathbf{P}_{m-1}\mathbf{x}_n = \left[\sum_{n\in\mathcal{S}_k} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{Q}_k\mathbf{x}_n\mathbf{x}_n^\top\right] \check{\mathbf{w}}_{k,m}\,.$$

Substituting eq. 114, and multiplying by $\mathbf{W}_{k,m}^\top$ from the right, we obtain

$$\sum_{n\in\mathcal{S}_m} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{W}_{k,m}^\top\mathbf{Q}_k\mathbf{P}_{m-1}\mathbf{x}_n = \left[\sum_{n\in\mathcal{S}_k} \exp\left(-\tilde{\mathbf{w}}^\top \mathbf{x}_n\right) \mathbf{W}_{k,m}^\top\mathbf{Q}_k\mathbf{x}_n\mathbf{x}_n^\top\mathbf{Q}_k\mathbf{W}_{k,m}\right] \mathbf{u}_{k,m}\,. \tag{116}$$

Denoting $\mathbf{E}_k \in \mathbb{R}^{|\mathcal{S}_k|\times|\mathcal{S}_k|}$ as diagonal matrix for which $E_{nn,k} = \exp\left(-\frac{1}{2}\tilde{\mathbf{w}}^\top\mathbf{x}_n\right)$, the matrix in the square bracket in the left hand side can be written as

$$\mathbf{W}_{k,m}^\top\mathbf{Q}_k\mathbf{X}_{\mathcal{S}_k}\mathbf{E}_k\mathbf{E}_k\mathbf{X}_{\mathcal{S}_k}^\top\mathbf{Q}_k\mathbf{W}_{k,m}\,. \tag{117}$$

Since $\operatorname{rank}\left(\mathbf{A}\mathbf{A}^\top\right) = \operatorname{rank}\left(\mathbf{A}\right)$ for any matrix $\mathbf{A}$, the rank of this matrix is equal to

$$\operatorname{rank}\left[\mathbf{E}\mathbf{X}_{\mathcal{S}_k}\mathbf{Q}_k\mathbf{W}_{k,m}\right] \overset{(1)}{=} \operatorname{rank}\left[\mathbf{X}_{\mathcal{S}_k}\mathbf{Q}_k\mathbf{W}_{k,m}\right] \overset{(2)}{=} d_k$$

where in (1) we used that $\mathbf{E}_k$ is diagonal and non-zero, and in (2) we used eq. 115. This implies that the $d_k \times d_k$ matrix in eq. 117 is full rank, and so eq. 116 has a unique solution $\mathbf{u}_{k,m}$. Therefore, there exists a unique solution $\check{\mathbf{w}}_{k,m}$. ■

C.6. Proof of Lemma 15

Lemma 15 *Let $\phi(t), h(t), z(t)$ be three functions from $\mathbb{N}$ to $\mathbb{R}_{\geq 0}$, and C_1, C_2, C_3 be three positive constants. Then, if $\sum_{t=1}^{\infty} h(t) \leq C_1 < \infty$, and*

$$\phi^2(t+1) \leq z(t) + h(t)\,\phi(t) + \phi^2(t) \tag{74}$$

we have

$$\phi^2(t+1) \leq C_2 + C_3\sum_{u=1}^{t} z(u) \tag{75}$$

Proof We define $\psi(t) = z(t) + h(t)$, and start from eq. 74

$$\begin{aligned}
&\phi^2(t+1)\\
&\leq z(t) + h(t)\,\phi(t) + \phi^2(t)\\
&\leq z(t) + h(t)\max\left[1, \phi^2(t)\right] + \phi^2(t)\\
&\leq z(t) + h(t) + h(t)\,\phi^2(t) + \phi^2(t)\\
&\leq \psi(t) + (1+h(t))\,\phi^2(t)\\
&\leq \psi(t) + (1+h(t))\,\psi(t-1) + (1+h(t))\,(1+h(t-1))\,\phi^2(t-1)\\
&\leq \psi(t) + (1+h(t))\,\psi(t-1) + (1+h(t))\,(1+h(t-1))\,\psi(t-2)\\
&+ (1+h(t))\,(1+h(t-1))\,(1+h(t-2))\,\phi^2(t-2)
\end{aligned}$$

we keep iterating eq. 74, until we obtain

$$\begin{aligned}
&\leq \left[\prod_{m=1}^{t-1}\left(1+h\left(t-m\right)\right)\right]\phi\left(t_1\right)+\sum_{k=0}^{t-t_1}\left[\prod_{m=0}^{k-1}\left(1+h\left(t-m\right)\right)\right]\psi\left(t-k\right)\\
&\leq \left[\exp\left(\sum_{m=1}^{t-1}h\left(t-m\right)\right)\right]\phi\left(t_1\right)+\sum_{k=0}^{t-1}\left[\exp\left(\sum_{m=1}^{k-1}h\left(t-m\right)\right)\right]\psi\left(t-k\right)\\
&\leq \exp\left(C\right)\left[\phi\left(1\right)+\sum_{k=0}^{t-1}\psi\left(t-k\right)\right]\\
&\leq \exp\left(C\right)\left[\phi\left(1\right)+\sum_{u=1}^{t}\psi\left(u\right)\right]\\
&\leq \exp\left(C\right)\left[\phi\left(1\right)+\sum_{u=1}^{t}\left(z\left(u\right)+h\left(u\right)\right)\right]\\
&\leq \exp\left(C\right)\left[\phi\left(1\right)+C+\sum_{u=1}^{t}z\left(u\right)\right]
\end{aligned}$$

Therefore, the Lemma holds with $C_2=\left(\phi\left(1\right)+C\right)\exp\left(C\right)$ and $C_3=\exp\left(C\right)$. ■

Appendix D. Calculation of convergence rates

In this section we calculate the various rates mentioned in section 3.

D.1. Proof of Theorem 5

From Theorems 4 and 13, we can write $\mathbf{w}\left(t\right)=\hat{\mathbf{w}}\log t+\boldsymbol{\rho}\left(t\right)$, where $\boldsymbol{\rho}\left(t\right)$ has a bounded norm for almost all datasets, while in zero measure case $\boldsymbol{\rho}\left(t\right)$ contains additional $O(\log\log(t))$ components which are orthogonal to the support vectors in $\mathcal{S}_1$, and, asymptotically, have a positive angle with the other support vectors. In this section we first calculate the various convergence rates for the non-degenerate case of Theorem 4, and then write the correction in the zero measure cases, if there is such a correction.

First, we calculated of the normalized weight vector (eq. 8), for almost every dataset:

$$\begin{aligned}
&\frac{\mathbf{w}\left(t\right)}{\left\|\mathbf{w}\left(t\right)\right\|}\\
&=\frac{\boldsymbol{\rho}\left(t\right)+\hat{\mathbf{w}}\log t}{\sqrt{\boldsymbol{\rho}\left(t\right)^{\top}\boldsymbol{\rho}\left(t\right)+\hat{\mathbf{w}}^{\top}\hat{\mathbf{w}}\log^2 t+2\boldsymbol{\rho}\left(t\right)^{\top}\hat{\mathbf{w}}\log t}}\\
&=\frac{\boldsymbol{\rho}\left(t\right)/\log t+\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|\sqrt{1+2\boldsymbol{\rho}\left(t\right)^{\top}\hat{\mathbf{w}}/\left(\left\|\hat{\mathbf{w}}\right\|^2\log t\right)+\left\|\boldsymbol{\rho}\left(t\right)\right\|^2/\left(\left\|\hat{\mathbf{w}}\right\|^2\log^2 t\right)}}\\
&=\frac{1}{\left\|\hat{\mathbf{w}}\right\|}\left(\boldsymbol{\rho}\left(t\right)\frac{1}{\log t}+\hat{\mathbf{w}}\right)\left[1-\frac{\boldsymbol{\rho}\left(t\right)^{\top}\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|^2\log t}+\left[\frac{3}{2}\left(\frac{\boldsymbol{\rho}\left(t\right)^{\top}\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|^2}\right)^2-\frac{\left\|\boldsymbol{\rho}\left(t\right)\right\|^2}{2\left\|\hat{\mathbf{w}}\right\|^2}\right]\frac{1}{\log^2 t}+O\left(\frac{1}{\log^3 t}\right)\right] \qquad (118)\\
&=\frac{\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|}+\left(\frac{\boldsymbol{\rho}\left(t\right)}{\left\|\hat{\mathbf{w}}\right\|}-\frac{\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|}\frac{\boldsymbol{\rho}\left(t\right)^{\top}\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|^2}\right)\frac{1}{\log t}+O\left(\frac{1}{\log^2 t}\right)\\
&=\frac{\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|}+\left(\mathbf{I}-\frac{\hat{\mathbf{w}}\hat{\mathbf{w}}^{\top}}{\left\|\hat{\mathbf{w}}\right\|^2}\right)\frac{\boldsymbol{\rho}\left(t\right)}{\left\|\hat{\mathbf{w}}\right\|}\frac{1}{\log t}+O\left(\frac{1}{\log^2 t}\right),
\end{aligned}$$

where to obtain eq. 118 we used $\frac{1}{\sqrt{1+x}}=1-\frac{1}{2}x+\frac{3}{4}x^2+O\left(x^3\right)$, and in the last line we used the fact that $\boldsymbol{\rho}\left(t\right)$ has a bounded norm for almost every dataset. Thus, in this case

$$\left\|\frac{\mathbf{w}\left(t\right)}{\left\|\mathbf{w}\left(t\right)\right\|}-\frac{\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|}\right\|=O\left(\frac{1}{\log t}\right).$$

For the measure zero cases, we instead have from eq. 61, $\mathbf{w}(t)=\sum_{m=1}^{M}\hat{\mathbf{w}}\log^{\circ m}(t)+\rho(t)$, where $\|\rho(t)\|$ is bounded (Theorem 3). Let $\tilde{\rho}(t)=\sum_{m=2}^{M}\hat{\mathbf{w}}\log^{\circ m}(t)+\rho(t)$, such that $\mathbf{w}(t)=\hat{\mathbf{w}}\log(t)+\tilde{\rho}(t)$ with $\tilde{\rho}(t)=O(\log\log(t))$. Repeating the same calculations as above, we have for the degenerate cases,

$$\left\|\frac{\mathbf{w}\left(t\right)}{\left\|\mathbf{w}\left(t\right)\right\|}-\frac{\hat{\mathbf{w}}}{\left\|\hat{\mathbf{w}}\right\|}\right\|=O\left(\frac{\log\log t}{\log t}\right)$$

Next, we use eq. 118 to calculate the angle (eq. 9)

$$\begin{aligned}
&\frac{\mathbf{w}(t)^\top \hat{\mathbf{w}}}{\|\mathbf{w}(t)\|\|\hat{\mathbf{w}}\|} \\
=&\frac{\hat{\mathbf{w}}^\top}{\|\hat{\mathbf{w}}\|^2}\left(\boldsymbol{\rho}(t)\frac{1}{\log t}+\hat{\mathbf{w}}\right)\left(1-\frac{1}{\log t}\frac{\boldsymbol{\rho}(t)^\top\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|^2}+\left[\frac{3}{4}\left(2\frac{\boldsymbol{\rho}(t)^\top\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|^2}\right)^2-\frac{\|\boldsymbol{\rho}(t)\|^2}{2\|\hat{\mathbf{w}}\|^2}\right]\frac{1}{\log^2 t}+O\left(\frac{1}{\log^3 t}\right)\right) \\
=&1+\frac{2\|\boldsymbol{\rho}(t)\|^2}{\|\hat{\mathbf{w}}\|^2}\left[\left(\frac{\boldsymbol{\rho}(t)^\top\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|\|\boldsymbol{\rho}(t)\|}\right)^2-\frac{1}{4}\right]\frac{1}{\log^2 t}+O\left(\frac{1}{\log^3 t}\right)
\end{aligned}$$

for almost every dataset. Thus, in this case

$$\frac{\mathbf{w}(t)^\top \hat{\mathbf{w}}}{\|\mathbf{w}(t)\|\|\hat{\mathbf{w}}\|}=O\left(\frac{1}{\log^2 t}\right)$$

Repeating the same calculation for the measure zero case, we have instead

$$\frac{\mathbf{w}(t)^\top \hat{\mathbf{w}}}{\|\mathbf{w}(t)\|\|\hat{\mathbf{w}}\|}=O\left(\left(\frac{\log\log t}{\log t}\right)^2\right)$$

Next, we calculate the margin (eq. 10)

$$\begin{aligned}
&\min_n \frac{\mathbf{x}_n^\top \mathbf{w}(t)}{\|\mathbf{w}(t)\|}-\frac{1}{\|\hat{\mathbf{w}}\|} \\
&=\min_n \mathbf{x}_n^\top\left[\left(\frac{\boldsymbol{\rho}(t)}{\|\hat{\mathbf{w}}\|}-\frac{\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|}\frac{\boldsymbol{\rho}(t)^\top\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|^2}\right)\frac{1}{\log t}+O\left(\frac{1}{\log^2 t}\right)\right] \\
&=\frac{1}{\|\hat{\mathbf{w}}\|}\left(\min_n \mathbf{x}_n^\top\boldsymbol{\rho}(t)-\frac{\boldsymbol{\rho}(t)^\top\hat{\mathbf{w}}}{\|\hat{\mathbf{w}}\|^2}\right)\frac{1}{\log t}+O\left(\frac{1}{\log^2 t}\right) \qquad (119)
\end{aligned}$$

for almost every dataset, where in eq. 119 we used eq. 19. Interestingly the measure zero case has a similar convergence rate, since after a sufficient number of iterations, the $O(\log\log(t))$ correction is orthogonal to $\mathbf{x}_k$, where $k=\mathrm{argmin}_n \mathbf{x}_n^\top \mathbf{w}(t)$. Thus, for all datasets,

$$\min_n \mathbf{x}_n^\top \mathbf{w}(t)-\frac{1}{\|\hat{\mathbf{w}}\|}=O\left(\frac{1}{\log t}\right) \qquad (120)$$

Calculation of the training loss (eq. 11):

$$\begin{aligned}
\mathcal{L}(\mathbf{w}(t)) &\leq \sum_{n=1}^N\left(1+\exp\left(-\mu_+\mathbf{w}(t)^\top\mathbf{x}_n\right)\right)\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_n\right) \\
&=\sum_{n=1}^N\left(1+\exp\left(-\mu_+\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_n\right)\right)\exp\left(-\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_n\right) \\
&=\sum_{n=1}^N\left(1+t^{-\mu_+\hat{\mathbf{w}}^\top\mathbf{x}_n}\exp\left(-\mu_+\boldsymbol{\rho}(t)^\top\mathbf{x}_n\right)\right)\exp\left(-\boldsymbol{\rho}(t)^\top\mathbf{x}_n\right)t^{-\hat{\mathbf{w}}^\top\mathbf{x}_n} \\
&=\frac{1}{t}\sum_{n\in\mathcal{S}}e^{-\boldsymbol{\rho}(t)^\top\mathbf{x}_n}+O\left(t^{-\max(\theta,1+\mu_+)}\right).
\end{aligned}$$

Thus, for all datasets $\mathcal{L}(\mathbf{w}(t))=O(t^{-1})$. Note that the zero measure case has the same behavior, since after a sufficient number of iterations, the $O(\log\log(t))$ correction has a non-negative angle with all the support vectors.

Next, we give an example demonstrating the bounds above, for the non-degenerate case, are strict. Consider optimization with and exponential loss $\ell(u)=e^{-u}$, and a single data point $\mathbf{x}=(1,0)$. In this case $\hat{\mathbf{w}}=(1,0)$ and $\|\hat{\mathbf{w}}\|=1$. We take the limit $\eta\to 0$, and obtain the continuous time version of GD:

$$\dot{w}_1(t)=\exp(-w(t))\ ;\ \dot{w}_2(t)=0.$$

We can analytically integrate these equations to obtain

$$w_1(t)=\log(t+\exp(w_1(0)))\ ;\ w_2(t)=w_2(0).$$

Using this example with $w_2(0)>0$, it is easy to see that the above upper bounds are strict in the non-degenerate case. ■

D.2. Validation error lower bound

Lastly, recall that $\mathcal{V}$ is a set of indices for validation set samples. We calculate of the validation loss for logistic loss, if the error of the L_2 max margin vector has some classification errors on the validation, *i.e.,* $\exists k\in\mathcal{V}:\ \hat{\mathbf{w}}^\top\mathbf{x}_k<0$*:*

$$\begin{aligned}
\mathcal{L}_{\text{val}}(\mathbf{w}(t)) &= \sum_{n\in\mathcal{V}}\log\left(1+\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_n\right)\right) \\
&\geq \log\left(1+\exp\left(-\mathbf{w}(t)^\top\mathbf{x}_k\right)\right) \\
&= \log\left(1+\exp\left(-\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_k\right)\right) \\
&= \log\left(\exp\left(-\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_k\right)\left(1+\exp\left(\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_k\right)\right)\right) \\
&\geq -\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_k+\log\left(1+\exp\left(\left(\boldsymbol{\rho}(t)+\hat{\mathbf{w}}\log t\right)^\top\mathbf{x}_k\right)\right) \\
&\geq -\log t\hat{\mathbf{w}}^\top\mathbf{x}_k+\boldsymbol{\rho}(t)^\top\mathbf{x}_k
\end{aligned}$$

Thus, for all datasets $\mathcal{L}_{\text{val}}(\mathbf{w}(t))=\Omega(\log(t))$.

Appendix E. Softmax output with cross-entropy loss

We examine multiclass classification. In the case the labels are the class index $y_n\in\{1,\ldots,K\}$ and we have a weight matrix $\mathbf{W}\in\mathbb{R}^{K\times d}$ with $\mathbf{w}_k$ being the k-th row of $\mathbf{W}$.

Furthermore, we define $\mathbf{w}=\mathrm{vec}(\mathbf{W}^\top)$, a basis vector $\mathbf{e}_k\in\mathbb{R}^K$ so that$(\mathbf{e}_k)_i=\delta_{ki}$, and the matrix $\mathbf{A}_k\in\mathbb{R}^{dK\times d}$ so that $\mathbf{A}_k=\mathbf{e}_k\otimes\mathbf{I}_d$, where $\otimes$ is the Kronecker product and $\mathbf{I}_d$ is the d-dimension identity matrix. Note that $\mathbf{A}_k^\top\mathbf{w}=\mathbf{w}_k$.

Consider the cross entropy loss with softmax output

$$\mathcal{L}(\mathbf{W})=-\sum_{n=1}^N\log\left(\frac{\exp\left(\mathbf{w}_{y_n}^\top\mathbf{x}_n\right)}{\sum_{k=1}^K\exp\left(\mathbf{w}_k^\top\mathbf{x}_n\right)}\right)$$

Using our notation, this loss can be re-written as

$$\mathcal{L}(\mathbf{w}) = -\sum_{n=1}^{N} \log\left(\frac{\exp\left(\mathbf{w}^\top \mathbf{A}_{y_n}\mathbf{x}_n\right)}{\sum_{k=1}^{K}\exp\left(\mathbf{w}^\top \mathbf{A}_k\mathbf{x}_n\right)}\right) = \sum_{n=1}^{N}\log\left(\sum_{k=1}^{K}\exp\left(\mathbf{w}^\top\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n\right)\right) \tag{121}$$

Therefore

$$\nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}\frac{\sum_{k=1}^{K}\exp\left(\mathbf{w}^\top\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n\right)\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n}{\sum_{r=1}^{K}\exp\left(\mathbf{w}^\top\left(\mathbf{A}_r-\mathbf{A}_{y_n}\right)\mathbf{x}_n\right)} = \sum_{n=1}^{N}\sum_{k=1}^{K}\frac{1}{\sum_{r=1}^{K}\exp\left(\mathbf{w}^\top\left(\mathbf{A}_r-\mathbf{A}_k\right)\mathbf{x}_n\right)}\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n\,.$$

If, again, we make the assumption that the data is linearly separable, *i.e.*, in our notation

Assumption 4 $\exists\mathbf{w}_*$ *such that* $\mathbf{w}_*^\top\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n<0\;\forall k\neq y_n$.

then the expression

$$\mathbf{w}_*^\top\nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\frac{\mathbf{w}_*^\top\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n}{\sum_{r=1}^{K}\exp\left(\mathbf{w}^\top\left(\mathbf{A}_r-\mathbf{A}_k\right)\mathbf{x}_n\right)}\,.$$

is strictly negative for any finite $\mathbf{w}$. However, from Lemma 10, in gradient descent with an appropriately small learning rate, we have that $\nabla L(\mathbf{w}(t))\to\mathbf{0}$. This implies that: $\|\mathbf{w}(t)\|\to\infty$, and $\forall k\neq y_n, \exists r:\ \mathbf{w}(t)^\top\left(\mathbf{A}_r-\mathbf{A}_k\right)\mathbf{x}_n\to\infty$, which implies $\forall k\neq y_n, \max_k\mathbf{w}(t)^\top\left(\mathbf{A}_k-\mathbf{A}_{y_n}\right)\mathbf{x}_n\to-\infty$. Examining the loss (eq. 121) we find that $\mathcal{L}(\mathbf{w}(t))\to\mathbf{0}$ in this case. Thus, we arrive to an equivalent Lemma to Lemma 1, for this case:

Lemma 19 *Let* $\mathbf{w}(t)$ *be the iterates of gradient descent (eq. 2) with an appropriately small learning rate, for cross-entropy loss operating on a softmax output, under the assumption of strict linear separability (Assumption 4), then: (1)* $\lim_{t\to\infty}\mathcal{L}(\mathbf{w}(t))=0$*, (2)* $\lim_{t\to\infty}\|\mathbf{w}(t)\|=\infty$*, and (3)* $\forall n,k\neq y_n:\ \lim_{t\to\infty}\mathbf{w}(t)^\top\left(\mathbf{A}_{y_n}-\mathbf{A}_k\right)\mathbf{x}_n=\infty$.

Using Lemma 10 and Lemma 19, we prove the following Theorem (equivalent to Theorem 3) in the next section:

Theorem 7 *For almost all multiclass datasets (*i.e.*, except for a measure zero) which are linearly separable (i.e. the constraints in eq. 15 below are feasible), any starting point* $\mathbf{w}(0)$ *and any small enough stepsize, the iterates of gradient descent on 13 will behave as:*

$$\mathbf{w}_k(t) = \hat{\mathbf{w}}_k\log(t)+\boldsymbol{\rho}_k(t), \tag{14}$$

where the residual $\boldsymbol{\rho}_k(t)$ *is bounded and* $\hat{\mathbf{w}}_k$ *is the solution of the K-class SVM:*

$$\text{argmin}_{\mathbf{w}_1,\ldots,\mathbf{w}_k}\sum_{k=1}^{K}||\mathbf{w}_k||^2 \;s.t.\;\forall n,\forall k\neq y_n:\mathbf{w}_{y_n}^\top\mathbf{x}_n\geq\mathbf{w}_k^\top\mathbf{x}_n+1. \tag{15}$$

E.1. Notations and Definitions

To prove Theorem 7 we require additional notation. we define $\tilde{\mathbf{x}}_{n,k}\triangleq\left(\mathbf{A}_{y_n}-\mathbf{A}_k\right)\mathbf{x}_n$. Using this notation, we can re-write eq. 15 (K-class SVM) as

$$\arg\min_{\mathbf{w}}\|\mathbf{w}\|^2 \text{ s.t. } \forall n,\forall k\neq y_n:\mathbf{w}^\top\tilde{\mathbf{x}}_{n,k}\geq 1 \tag{122}$$

From the KKT optimality conditions, we have for some $\alpha_{n,k}\geq 0$,

$$\hat{\mathbf{w}} = \sum_{n=1}^{N}\sum_{k=1}^{K}\alpha_{n,k}\tilde{\mathbf{x}}_{n,k}\mathbf{1}_{\{n\in\mathcal{S}_k\}} \tag{123}$$

In addition, for each of the K classes, we define $\mathcal{S}_k=\arg\min_n(\hat{\mathbf{w}}_{y_n}-\hat{\mathbf{w}}_k)^\top\mathbf{x}_n$ (the k'th class support vectors).
Using this definition, we define $\mathbf{X}_{\mathcal{S}_k}\in\mathcal{R}^{dK\times|S_k|}$ as the matrix which columns are $\tilde{\mathbf{x}}_{n,k},\ \forall n\in\mathcal{S}_k$.
We also define $\mathcal{S}\triangleq\bigcup_{k=1}^{K}\mathcal{S}_k$ and $\tilde{\mathbf{X}}_{\mathcal{S}}\triangleq\bigcup_{k=1}^{K}\mathbf{X}_{\mathcal{S}_k}$.
We recall that we defined $\mathbf{W}\in\mathbb{R}^{K\times d}$ with $\mathbf{w}_k$ being the k-th row of $\mathbf{W}$ and $\mathbf{w}=\text{vec}(\mathbf{W}^\top)$. Similarly, we define:
1. $\hat{\mathbf{W}}\in\mathbb{R}^{K\times d}$ with $\hat{\mathbf{w}}_k$ being the k-th row of $\hat{\mathbf{W}}$
2. $\mathbf{P}\in\mathbb{R}^{K\times d}$ with $\boldsymbol{\rho}_k$ being the k-th row of $\mathbf{P}$
3. $\tilde{\mathbf{W}}\in\mathbb{R}^{K\times d}$ with $\tilde{\mathbf{w}}_k$ being the k-th row of $\tilde{\mathbf{W}}$
and $\hat{\mathbf{w}}=\text{vec}(\hat{\mathbf{W}}^\top)$, $\boldsymbol{\rho}=\text{vec}(\mathbf{P}^\top)$, $\tilde{\mathbf{w}}=\text{vec}(\tilde{\mathbf{W}}^\top)$.
Using our notations, eq. 14 can be re-written as $\mathbf{w}=\hat{\mathbf{w}}\log(t)+\boldsymbol{\rho}(t)$ when $\boldsymbol{\rho}(t)$ is bounded.
For any solution $\mathbf{w}(t)$, we define

$$\mathbf{r}(t)=\mathbf{w}(t)-\hat{\mathbf{w}}\log t-\tilde{\mathbf{w}}, \tag{124}$$

where $\hat{\mathbf{w}}$ is the concatenation of $\hat{\mathbf{w}}_1,\ldots,\hat{\mathbf{w}}_k$ which are the K-class SVM solution, so

$$\forall k,\ \forall n\in\mathcal{S}_k:\tilde{\mathbf{x}}_{n,k}^\top\hat{\mathbf{w}}=1\ ;\ \theta=\min_k\left[\min_{n\notin\mathcal{S}_k}\tilde{\mathbf{x}}_{n,k}^\top\hat{\mathbf{w}}\right]>1 \tag{125}$$

and $\tilde{\mathbf{w}}$ satisfies the equation:

$$\forall k,\ \forall n\in\mathcal{S}_k\ :\ \eta\exp((\tilde{\mathbf{w}}_k-\tilde{\mathbf{w}}_{y_n})^\top\mathbf{x}_n)=\alpha_{n,k} \tag{126}$$

This equation has a unique solution for almost every data set according to Lemma 12.
For each of the K classes, we define $\mathbf{P}_1^k\in\mathcal{R}^{d\times d}$ as the orthogonal projection matrix to the subspace spanned by the support vector of the k'th class, and $\bar{\mathbf{P}}_1^k=\mathbf{I}-\mathbf{P}_1^k$ as the complementary projection.
Finally, we define $\mathbf{P}_1\in\mathcal{R}^{Kd\times Kd}$ and $\bar{\mathbf{P}}_1\in\mathcal{R}^{Kd\times Kd}$ as follows:

$$\mathbf{P}_1=\text{diag}(\mathbf{P}_1^1,\mathbf{P}_1^2,\ldots,\mathbf{P}_1^K)\ ,\ \bar{\mathbf{P}}_1=\text{diag}(\bar{\mathbf{P}}_1^1,\bar{\mathbf{P}}_1^2,\ldots,\bar{\mathbf{P}}_1^K)$$

$$(\mathbf{P}_1+\bar{\mathbf{P}}_1=\mathbf{I}\in\mathcal{R}^{Kd\times Kd})$$

In the following section we will also use $\mathbf{1}_{\{A\}}$, the indicator function, which is 1 if A is satisfied and 0 otherwise.

E.2. Auxiliary Lemma

Lemma 20 *We have*

$$\exists C_1, t_1 \ : \ \forall t > t_1 : (\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq C_1 t^{-\theta} + C_2 t^{-2} \tag{127}$$

Additionally, $\forall \epsilon_1 > 0$, $\exists C_2, t_2$, such that $\forall t > t_2$, such that if

$$||\mathbf{P}_1 \mathbf{r}(t)|| > \epsilon_1 \tag{128}$$

then we can improve this bound to

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq -C_3 t^{-1} < 0 \tag{129}$$

We prove the Lemma below, in appendix section E.4

E.3. Proof of Theorem 7

Our goal is to show that $||\mathbf{r}(t)||$ is bounded, and therefore $\boldsymbol{\rho}(t) = \mathbf{r}(t) + \tilde{\mathbf{w}}$ is bounded.
To show this, we will upper bound the following equation

$$\|\mathbf{r}(t+1)\|^2 = \|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 + 2(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) + \|\mathbf{r}(t)\|^2 \tag{130}$$

First, we note that first term in this equation can be upper-bounded by

$$\begin{aligned}
&\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 \\
&\overset{(1)}{=} \|\mathbf{w}(t+1) - \hat{\mathbf{w}}\log(t+1) - \tilde{\mathbf{w}} - \mathbf{w}(t) + \hat{\mathbf{w}}\log(t) + \tilde{\mathbf{w}}\|^2 \\
&\overset{(2)}{=} \|-\eta\nabla\mathcal{L}(\mathbf{w}(t)) - \hat{\mathbf{w}}[\log(t+1) - \log(t)]\|^2 \\
&= \eta^2\|\nabla\mathcal{L}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2\log^2\left(1+t^{-1}\right) + 2\eta\hat{\mathbf{w}}^\top\nabla\mathcal{L}(\mathbf{w}(t))\log\left(1+t^{-1}\right) \\
&\overset{(3)}{\leq} \eta^2\|\nabla\mathcal{L}(\mathbf{w}(t))\|^2 + \|\hat{\mathbf{w}}\|^2 t^{-2},
\end{aligned} \tag{131}$$

where in (1) we used eq. 124, in (2) we used eq 2.2, and in (3) we used $\forall x > 0 : x \geq \log(1+x) > 0$, and also that

$$\hat{\mathbf{w}}^\top\nabla\mathcal{L}(\mathbf{w}) = \sum_{n=1}^{N}\sum_{k=1}^{K}\frac{\hat{\mathbf{w}}^\top(\mathbf{A}_{y_n} - \mathbf{A}_k)\mathbf{x}_n}{\sum_{r=1}^{K}\exp(\mathbf{w}^\top(\mathbf{A}_r - \mathbf{A}_k)\mathbf{x}_n)} < 0 \tag{132}$$

since $\hat{\mathbf{w}}^\top(\mathbf{A}_r - \mathbf{A}_k)\mathbf{x}_n = (\hat{\mathbf{w}}_r - \hat{\mathbf{w}}_{y_n})\mathbf{x}_n < 0, \forall k \neq y_n$ (we recall that $\hat{\mathbf{w}}_k$ is the K-class SVM solution).
Also, from Lemma 10 we know that

$$\|\nabla\mathcal{L}(\mathbf{w}(t))\|^2 = o(1) \text{ and } \sum_{t=0}^{\infty}\|\nabla\mathcal{L}(\mathbf{w}(t))\|^2 < \infty\,. \tag{133}$$

Substituting eq. 133 into eq. 131, and recalling that a $t^{-\nu}$ power series converges for any $\nu > 1$, we can find C_0 such that

$$\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 = o(1) \text{ and } \sum_{t=0}^{\infty}\|\mathbf{r}(t+1) - \mathbf{r}(t)\|^2 = C_0 < \infty\,. \tag{134}$$

Note that this equation also implies that $\forall \epsilon_0$

$$\exists t_0 : \forall t > t_0 : |\|\mathbf{r}(t+1)\| - \|\mathbf{r}(t)\|| < \epsilon_0\,. \tag{135}$$

Next, we would like to bound the second term in eq. 130. From eq. 127 in Lemma 20, we can find t_1, C_1 such that $\forall t > t_1$:

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq C_1 t^{-\theta} + C_2 t^{-2} \tag{136}$$

Thus, by combining eqs. 136 and 134 into eq. 130, we find:

$$\begin{aligned}
&||\mathbf{r}(t)||^2 - ||\mathbf{r}(t_1)||^2 \\
&= \sum_{u=t_1}^{t-1}\left[||\mathbf{r}(u+1)||^2 - ||\mathbf{r}(u)||^2\right] \\
&\leq C_0 + 2\sum_{u=t_1}^{t-1}\left[C_1 u^{-\theta} + C_2 u^{-2}\right]
\end{aligned}$$

which is bounded, since $\theta > 1$ (eq. 125). Therefore, $||\mathbf{r}(t)||$ is bounded.

E.4. Proof of Lemma 20

Lemma 20 *We have*

$$\exists C_1, t_1 \ : \ \forall t > t_1 : (\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq C_1 t^{-\theta} + C_2 t^{-2} \tag{127}$$

Additionally, $\forall \epsilon_1 > 0$, $\exists C_2, t_2$, such that $\forall t > t_2$, such that if

$$||\mathbf{P}_1 \mathbf{r}(t)|| > \epsilon_1 \tag{128}$$

then we can improve this bound to

$$(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) \leq -C_3 t^{-1} < 0 \tag{129}$$

We wish to bound $(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t)$. First, we recall we defined $\tilde{\mathbf{x}}_{n,k} \triangleq (\mathbf{A}_{y_n} - \mathbf{A}_k)\mathbf{x}_n$.

$$\begin{aligned}
&(\mathbf{r}(t+1) - \mathbf{r}(t))^\top \mathbf{r}(t) = (-\eta\nabla\mathcal{L}(\mathbf{w}(t)) - \hat{\mathbf{w}}[\log(t+1) - \log(t)])^\top \mathbf{r}(t) \\
&= \left(\eta\sum_{n=1}^{N}\frac{\sum_{k=1}^{K}\exp(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k})\tilde{\mathbf{x}}_{n,k}}{\sum_{r=1}^{K}\exp(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r})} - \hat{\mathbf{w}}\log(1+t^{-1})\right)^\top \mathbf{r}(t) \\
&= \hat{\mathbf{w}}^\top\mathbf{r}(t)[t^{-1} - \log(1+t^{-1})] \qquad (137) \\
&+ \eta\sum_{n=1}^{N}\sum_{k=1}^{K}\left[\frac{\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)}{\sum_{r=1}^{K}\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r}\right)} - t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right], \qquad (138)
\end{aligned}$$

where in the last line we used eqs. 123 and 126 to obtain

$$\hat{\mathbf{w}} = \eta\sum_{n=1}^{N}\sum_{k=1}^{K}\alpha_{n,k}\tilde{\mathbf{x}}_{n,k}\mathbf{1}_{\{n\in\mathcal{S}_k\}} = \eta\sum_{n=1}^{N}\sum_{k=1}^{K}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}\mathbf{1}_{\{n\in\mathcal{S}_k\}},$$

where $\mathbf{1}_{\{A\}}$ is the indicator function which is 1 if A is satisfied and 0 otherwise.

The first term can be upper bounded by

$$\begin{aligned}
&\hat{\mathbf{w}}^\top \mathbf{r}\left(t\right)\left[t^{-1}-\log\left(1+t^{-1}\right)\right]\\
\leq&\max\left[\hat{\mathbf{w}}^\top \mathbf{r}\left(t\right),0\right]\left[t^{-1}-\log\left(1+t^{-1}\right)\right]\\
\overset{(1)}{\leq}&\max\left[\hat{\mathbf{w}}^\top \mathbf{P}_1\mathbf{r}\left(t\right),0\right]t^{-2}\\
\overset{(2)}{\leq}&\begin{cases}\|\hat{\mathbf{w}}\|\epsilon_1 t^{-2} & ,\text{ if }\left\|\mathbf{P}_1\mathbf{r}\left(t\right)\right\|\leq\epsilon_1\\ o\left(t^{-1}\right) & ,\text{ if }\left\|\mathbf{P}_1\mathbf{r}\left(t\right)\right\|>\epsilon_1\end{cases}
\end{aligned}\tag{139}$$

where in (1) we used that $\mathbf{P}_2\hat{\mathbf{w}}=0$, and in (2) we used that $\hat{\mathbf{w}}^\top\mathbf{r}\left(t\right)=o\left(t\right)$, since

$$\begin{aligned}
\hat{\mathbf{w}}^\top\mathbf{r}\left(t\right)&=\hat{\mathbf{w}}^\top\left(\mathbf{w}\left(0\right)-\eta\sum_{u=0}^{t}\nabla\mathcal{L}\left(\mathbf{w}\left(u\right)\right)-\hat{\mathbf{w}}\log\left(t\right)-\tilde{\mathbf{w}}\right)\\
&\leq\hat{\mathbf{w}}^\top\left(\mathbf{w}\left(0\right)-\tilde{\mathbf{w}}-\hat{\mathbf{w}}\log\left(t\right)\right)-\eta t\min_{0\leq u\leq t}\hat{\mathbf{w}}^\top\nabla\mathcal{L}\left(\mathbf{w}\left(u\right)\right)=o\left(t\right)
\end{aligned}$$

where in the last line we used that $\nabla\mathcal{L}\left(\mathbf{w}\left(t\right)\right)=o\left(1\right)$, from Lemma 10.
Next, we wish to upper bound the second term in eq. 137:

$$\eta\sum_{n=1}^{N}\sum_{k=1}^{K}\left[\frac{\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)}{\sum_{r=1}^{K}\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r}\right)}-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right]\tag{140}$$

We examine each term n in the sum:

$$\begin{aligned}
&\sum_{k=1}^{K}\left[\frac{\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)}{\sum_{r=1}^{K}\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r}\right)}-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right]\\
&=\sum_{k=1}^{K}\left[\frac{\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)}{1+\sum_{\substack{r=1\\r\neq y_n}}^{K}\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r}\right)}-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right]\\
&\overset{(1)}{\leq}\sum_{k=1}^{K}\left(\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right)\mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\geq 0\}}\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\\
&+\sum_{k=1}^{K}\left(\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)\left(1-\sum_{\substack{r=1\\r\neq y_n}}^{K}\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,r}\right)\right)\right.\\
&\left.-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right)\mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)<0\}}\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\\
&=\sum_{k=1}^{K}\left(\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\\
&-\sum_{k=1}^{K}\sum_{\substack{r=1\\r\neq y_n}}^{K}\exp\left(-\mathbf{w}(t)^\top(\tilde{\mathbf{x}}_{n,k}+\tilde{\mathbf{x}}_{n,r})\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)<0\}}\\
&\overset{(2)}{\leq}\sum_{k=1}^{K}\left(\exp\left(-\mathbf{w}(t)^\top\tilde{\mathbf{x}}_{n,k}\right)-t^{-1}\exp\left(-\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\mathbf{1}_{\{n\in\mathcal{S}_k\}}\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\\
&-K^2\exp\left(-\mathbf{w}(t)^\top(\tilde{\mathbf{x}}_{n,k_1}+\tilde{\mathbf{x}}_{n,r_1})\right)\tilde{\mathbf{x}}_{n,k_1}^\top\mathbf{r}(t)\mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top\mathbf{r}(t)<0\}},
\end{aligned}\tag{141}$$

where in (1) we used $\forall x\geq 0:\ 1-x\leq\frac{1}{1+x}\leq 1$ and in (2) we defined:

$$(k_1,r_1)=\underset{k,r}{\operatorname{argmax}}\left|\exp\left(-\mathbf{w}(t)^\top(\tilde{\mathbf{x}}_{n,k}+\tilde{\mathbf{x}}_{n,r})\right)\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)\mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}(t)<0\}}\right|$$

Recalling that $\mathbf{w}(t) = \hat{\mathbf{w}}\log(t) + \tilde{\mathbf{w}} + \mathbf{r}(t)$, eq. 141 can be upper bounded by

$$\sum_{k=1}^{K} t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \geq 0\ ,\ n \notin \mathcal{S}_k\}}$$
$$+ \sum_{k=1}^{K} t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \geq 0\ ,\ n \in \mathcal{S}_k\}}$$
$$+ \sum_{k=1}^{K} t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0\ ,\ n \notin \mathcal{S}_k\}}$$
$$+ \sum_{k=1}^{K} t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0\ ,\ n \in \mathcal{S}_k\}}$$
$$- K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}$$
$$\overset{(1)}{\leq} K t^{-\theta} \exp\left(-\min_{n,k} \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) + \phi(t), \tag{142}$$

where in (1) we used $xe^{-x} < 1,\ \forall x:\ (e^{-x}-1)x < 0, \theta = \min_k \left[\min_{n \notin \mathcal{S}_k} \tilde{\mathbf{x}}_{n,k}^\top \hat{\mathbf{w}}\right] > 1$ (eq. 125) and denoted:

$$\phi(t) = \sum_{k=1}^{K} t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0,\ n \notin \mathcal{S}_k\}}$$
$$+ \sum_{k=1}^{K} t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0, n \in \mathcal{S}_k\}}$$
$$- K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}.$$

We use the fact that $\forall x:\ (e^{-x}-1)x < 0$ and therefore $\forall (n,k)$:

$$t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0\}} < 0$$
$$t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0\}} < 0, \tag{143}$$

to show that $\phi(t)$ is strictly negative. If $\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r} \geq 0$ then from the last two equations:

$$\phi(t) = \sum_{k=1}^{K} t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0,\ n \notin \mathcal{S}_k\}}$$
$$+ \sum_{k=1}^{K} t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) < 0,\ n \in \mathcal{S}_k\}} < 0 \tag{144}$$

If $\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r} < 0$ then we note that $-\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) \leq -\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)$ since:
1. If $\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) \geq 0$ then this is immediate since $-\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) \leq 0 \leq -\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)$.
2. If $\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) < 0$ then from (k_1, r_1) definition:

$$\left|\exp\left(-\mathbf{w}(t)^\top (\tilde{\mathbf{x}}_{n,k_1} + \tilde{\mathbf{x}}_{n,r_1})\right) \tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t)\right| \leq \left|\exp\left(-\mathbf{w}(t)^\top (\tilde{\mathbf{x}}_{n,k_1} + \tilde{\mathbf{x}}_{n,r_1})\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)\right|,$$

and therefore

$$-\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) = \left|\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t)\right| \leq \left|\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)\right| = -\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t).$$

We divide into cases:
1. If $n \notin \mathcal{S}_{k_1}$ then we examine the sum

$$t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}$$
$$- K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}$$

The first term is negative and the second is positive. From Lemma 19 $\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1} \to \infty$. Therefore $\exists t_3$ so that $\forall t > t_3:\ \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) < K^2$ and therefore this sum is strictly negative since

$$\left|\frac{K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right)\right) \exp\left(-\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}}{t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}}\right|$$
$$= \left|K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1})\right| < 1,\ \forall t > t_3$$

2. If $n \in \mathcal{S}_{k_1}$ then we examine the sum

$$t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) - 1\right] \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}$$
$$- K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}$$

a. If $|\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)| > C_0$ then $\exists t_4$ such that $\forall t > t_4$ this sum can be upper bounded by zero since

$$\left|\frac{K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1}) t^{-\hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right)\right) \exp\left(-\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)\right) \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}}{t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k_1}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right) - 1\right] \tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \mathbf{1}_{\{\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) < 0\}}}\right|$$
$$= \frac{K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1})}{1 - \exp\left(\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k_1}\right)} \leq \frac{K^2 \exp(-\mathbf{w}(t)^\top \tilde{\mathbf{x}}_{n,r_1})}{1 - \exp\left(-C_0\right)} < 1,\ \forall t > t_4 \tag{145}$$

where in the last transition we used Lemma 19.
b. If $|\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t)| \leq C_0$ then we can find constant C_5 so that eq. 145 can be upper bounded by

$$K^2 t^{-\hat{\mathbf{w}}^\top (\tilde{\mathbf{x}}_{n,k_1} + \tilde{\mathbf{x}}_{n,r_1})} \exp\left(-\tilde{\mathbf{w}}^\top (\tilde{\mathbf{x}}_{n,k_1} + \tilde{\mathbf{x}}_{n,r_1})\right) \exp\left(2C_0\right) C_0 \leq C_5 t^{-2}, \tag{146}$$

since $-\tilde{\mathbf{x}}_{n,r_1}^\top \mathbf{r}(t) \leq -\tilde{\mathbf{x}}_{n,k_1}^\top \mathbf{r}(t) \leq C_0$ and by definition, $\forall (n,k):\ \hat{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k} \geq 1$.
Therefore, eq. 141 can be upper bounded by

$$K t^{-\theta} \exp\left(-\min_{n,k} \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) + C_5 t^{-2} \tag{147}$$

If, in addition, $\exists k, n \in \mathcal{S}_k:\ |\tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t)| > \epsilon_2$ then

$$t^{-1} \exp\left(-\tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(-\mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k}\right) - 1\right] \tilde{\mathbf{x}}_{n,k}^\top \mathbf{r}(t) \tag{148}$$
$$\leq \begin{cases} -t^{-1} \exp\left(-\max_{n,k} \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[1 - \exp\left(-\epsilon_2\right)\right] \epsilon_2 & , \text{ if } \mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k} \geq 0 \\ -t^{-1} \exp\left(-\max_{n,k} \tilde{\mathbf{w}}^\top \tilde{\mathbf{x}}_{n,k}\right) \left[\exp\left(\epsilon_2\right) - 1\right] \epsilon_2 & , \text{ if } \mathbf{r}(t)^\top \tilde{\mathbf{x}}_{n,k} < 0 \end{cases} \tag{149}$$

and we can improve this bound to

$$-C''t^{-1} < 0, \tag{150}$$

where C'' is the minimum between $\exp\left(-\max_{n,k}\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\left[1-\exp\left(-\epsilon_2\right)\right]\epsilon_2$ and $\exp\left(-\max_{n,k}\tilde{\mathbf{w}}^\top\tilde{\mathbf{x}}_{n,k}\right)\left[\exp\left(\epsilon_2\right)-1\right]\epsilon_2$. To conclude:
1. If $\|\mathbf{P}_1\mathbf{r}\left(t\right)\| \geq \epsilon_1$ (as in Eq. 139), we have that

$$\max_{k,n\in\mathcal{S}_k}\left|\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}\left(t\right)\right|^2 \overset{(1)}{\geq} \frac{1}{|\mathcal{S}|}\sum_{k,n\in\mathcal{S}_k}\left|\tilde{\mathbf{x}}_{n,k}^\top\mathbf{P}_1\mathbf{r}\left(t\right)\right|^2 = \frac{1}{|\mathcal{S}|}\left\|\mathbf{X}_\mathcal{S}^\top\mathbf{P}_1\mathbf{r}\left(t\right)\right\|^2 \overset{(2)}{\geq} \frac{1}{|\mathcal{S}|}\sigma_{\min}^2\left(\mathbf{X}_\mathcal{S}\right)\epsilon_1^2 \tag{151}$$

where in (1) we used $\mathbf{P}_1^\top\tilde{\mathbf{x}}_{n,k} = \tilde{\mathbf{x}}_{n,k}\ \forall k,\ n\in\mathcal{S}_k$, in (2) we denoted by $\sigma_{\min}\left(\mathbf{X}_\mathcal{S}\right)$, the minimal non-zero singular value of $\mathbf{X}_\mathcal{S}$ and used eq. 128. Therefore, for some (n,k), $\left|\tilde{\mathbf{x}}_{n,k}^\top\mathbf{r}\right| \geq \epsilon_2 \triangleq |\mathcal{S}|^{-1}\sigma_{\min}^2\left(\mathbf{X}_\mathcal{S}\right)\epsilon_1^2$. If $||\mathbf{P}_1\mathbf{r}(t)|| \geq \epsilon_1$, then combining eq. 139 with eq. 150 we find that eq. 137 can be upper bounded by:

$$\left(\mathbf{r}(t+1)-\mathbf{r}(t)\right)^\top\mathbf{r}(t) \leq -C''t^{-1}+o(t^{-1})$$

This implies that $\exists C_2 < C''$ and $\exists t_2 > 0$ such that eq. 129 holds. This implies also that eq. 127 holds for $||\mathbf{P}_1\mathbf{r}(t)|| \geq \epsilon_1$.
2. If $||\mathbf{P}_1 r(t)|| < \epsilon_1$, we obtain (for some positive constants C_3, C_4):

$$\left(\mathbf{r}(t+1)-\mathbf{r}(t)\right)^\top\mathbf{r}(t) \leq C_3t^{-\theta}+C_4t^{-2}$$

Therefore, $\exists t_1 > 0$ and C_1 such that eq. 127 holds.

Appendix F. An experiment with stochastic gradient descent

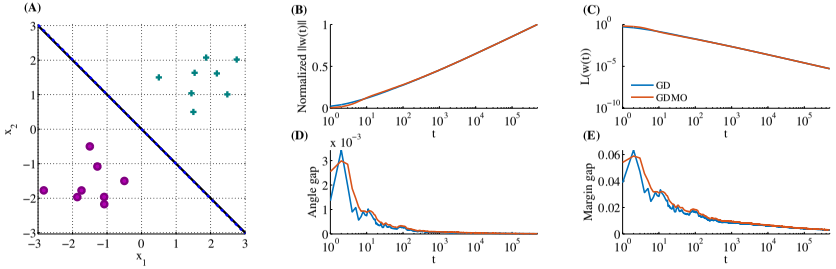


Figure 4: Same as Fig. 1, except stochastic gradient decent is used (with mini-batch of size 4), instead of GD.

References

Mor Shpigel Nacson, Nati Srebro, and Daniel Soudry. Stochastic Gradient Descent on Separable Data Exact Convergence with a Fixed Learning Rate. *arXiv 1806.01796*, 2018.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Implicit Bias of Gradient Descent on Linear Convolutional Networks. *NIPS*, 2018.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

Radha Krishna Ganti. EE6151, Convex optimization algorithms. Unconstrained minimization: Gradient descent algorithm, 2015. URL

Suriya Gunasekar, Blake Woodworth, Srinadh Bhojanapalli, Behnam Neyshabur, and Nathan Srebro. Implicit Regularization in Matrix Factorization. *arXiv*, pages 1–10, 2017.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. *arXiv:1802.08246*, 2018.

Moritz Hardt, Benjamin Recht, and Y Singer. Train faster, generalize better: Stability of stochastic gradient descent. *ICML*, pages 1–24, 2016.

Elad Hoffer, Itay Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In *NIPS*, pages 1–13, may 2017.

I Hubara, M Courbariaux, D. Soudry, R El-yaniv, and Y Bengio. Quantized Neural Networks: Training Neural Networks with Low Precision Weights and Activations. *JMLR*, 2018.

Ziwei Ji and Matus Telgarsky. Risk and parameter convergence of logistic regression. Communicated by the authors, 2018.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. *ICLR*, pages 1–16, 2017.

Diederik P Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. In *ICLR*, pages 1–13, 2015.

Mor Shpigel Nacson, Jason Lee, Suriya Gunasekar, Nathan Srebro, and Daniel Soudry. Convergence of Gradient Descent on Separable Data. *arXiv*, pages 1–45, 2018.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv:1412.6614*, 2014.

Behnam Neyshabur, Ruslan R Salakhutdinov, and Nati Srebro. Path-sgd: Path-normalized optimization in deep neural networks. In *NIPS*, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring Generalization in Deep Learning. *arXiv*, jun 2017.

Saharon Rosset, Ji Zhu, and Trevor J Hastie. Margin Maximizing Loss Functions. In *NIPS*, pages 1237–1244, 2004.

Robert E Schapire, Yoav Freund, Peter Bartlett, Wee Sun Lee, et al. Boosting the margin: A new explanation for the effectiveness of voting methods. *The annals of statistics*, 26(5):1651–1686, 1998.

Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, and N Srebro. The Implicit Bias of Gradient Descent on Separable Data. In *ICLR*, 2018.

Matus Telgarsky. Margins, shrinkage and boosting. In *Proceedings of the 30th International Conference on International Conference on Machine Learning-Volume 28*, pages II–307. JMLR. org, 2013.

Ashia C Wilson, Rebecca Roelofs, Mitchell Stern, Nathan Srebro, and Benjamin Recht. The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv*, pages 1–14, 2017.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *ICLR*, 2017.

Tong Zhang, Bin Yu, et al. Boosting with early stopping: Convergence and consistency. *The Annals of Statistics*, 33(4):1538–1579, 2005.

Journal of Machine Learning Research 19 (2018) 1-36 Submitted 4/18; Revised 9/18; Published 10/18

Optimal Quantum Sample Complexity of Learning Algorithms

Srinivasan Arunachalam * ARUNACHA@MIT.EDU
QuSoft, CWI, Amsterdam, the Netherlands

Ronald de Wolf † RDEWOLF@CWI.NL
QuSoft, CWI and University of Amsterdam, the Netherlands

Editor: Manfred Warmuth

Abstract

In learning theory, the *VC dimension* of a concept class $\mathscr{C}$ is the most common way to measure its "richness." A fundamental result says that the number of examples needed to learn an unknown target concept $c \in \mathscr{C}$ under an unknown distribution D, is tightly determined by the VC dimension d of the concept class $\mathscr{C}$. Specifically, in the PAC model

$$\Theta\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big)$$

examples are necessary and sufficient for a learner to output, with probability $1-\delta$, a hypothesis h that is ε-close to the target concept c (measured under D). In the related *agnostic* model, where the samples need not come from a $c \in \mathscr{C}$, we know that

$$\Theta\Big(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\Big)$$

examples are necessary and sufficient to output an hypothesis $h \in \mathscr{C}$ whose error is at most ε worse than the error of the best concept in $\mathscr{C}$.

Here we analyze *quantum* sample complexity, where each example is a coherent quantum state. This model was introduced by Bshouty and Jackson (1999), who showed that quantum examples are more powerful than classical examples in some fixed-distribution settings. However, Atıcı and Servedio (2005), improved by Zhang (2010), showed that in the PAC setting (where the learner has to succeed for every distribution), quantum examples cannot be much more powerful: the required number of quantum examples is

$$\Omega\Big(\frac{d^{1-\eta}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\Big) \text{ for arbitrarily small constant } \eta > 0.$$

Our main result is that quantum and classical sample complexity are in fact equal up to constant factors in both the PAC and agnostic models. We give two proof approaches. The first is a fairly simple information-theoretic argument that yields the above two classical bounds and yields the same bounds for quantum sample complexity up to a $\log(d/\varepsilon)$ factor. We then give a second approach that avoids the log-factor loss, based on analyzing the behavior of the "Pretty Good Measurement" on the quantum state-identification problems that correspond to learning. This shows classical and quantum sample complexity are equal up to constant factors for every concept class $\mathscr{C}$.

Keywords: Quantum computing, PAC learning, Sample complexity, Lower bounds

∗. Supported by ERC Consolidator Grant 615307 QPROGRESS.
†. Partially supported by ERC Consolidator Grant 615307 QPROGRESS.

1. Introduction

1.1. Sample complexity and VC dimension

Machine learning is one of the most successful parts of AI, with impressive practical applications in areas ranging from image processing and speech recognition, to even beating Go champions. Its theoretical aspects have been deeply studied, revealing beautiful structure and mathematical characterizations of when (efficient) learning is or is not possible in various settings.

1.1.1. The PAC setting

Leslie Valiant (1984)'s Probably Approximately Correct (PAC) learning model gives a precise complexity-theoretic definition of what it means for a concept class to be (efficiently) learnable. For simplicity we will (without loss of generality) focus on concepts that are Boolean functions, $c : \{0,1\}^n \to \{0,1\}$. Equivalently, a concept c is a subset of $\{0,1\}^n$, namely $\{x : c(x) = 1\}$. Let $\mathscr{C} \subseteq \{f : \{0,1\}^n \to \{0,1\}\}$ be a concept class. This could for example be the class of functions computed by disjunctive normal form (DNF) formulas of a certain size, or Boolean circuits or decision trees of a certain depth.

The goal of a learning algorithm (the learner) is to probably approximate some unknown *target concept* $c \in \mathscr{C}$ from random *labeled examples*. Each labeled example is of the form $(x, c(x))$ where x is distributed according to some unknown distribution D over $\{0,1\}^n$. After processing a number of such examples (hopefully not too many), the learner outputs some *hypothesis* h. We say that h is *ε-approximately correct* (w.r.t. the target concept c) if its error probability under D is at most ε: $\Pr_{x\sim D}[h(x) \neq c(x)] \leq \varepsilon$. Note that the learning phase and the evaluation phase (i.e., whether a hypothesis is approximately correct) are according to the same distribution D—as if the learner is taught and then tested by the same teacher. An (ε, δ)-learner for the concept class $\mathscr{C}$ is one whose hypothesis is probably approximately correct:

> For all target concepts $c \in \mathscr{C}$ and distributions D:
> Pr[the learner's output h is ε-approximately correct] $\geq 1-\delta$,

where the probability is over the sequence of examples and the learner's internal randomness. Note that we leave the learner the freedom to output an h which is not in $\mathscr{C}$. If always $h \in \mathscr{C}$, then the learner is called a *proper* PAC-learner.

Of course, we want the learner to be as efficient as possible. Its *sample complexity* is the worst-case number of examples it uses, and its *time complexity* is the worst-case running time of the learner. In this paper we focus on sample complexity. This allows us to ignore technical issues of how the runtime of an algorithm is measured, and in what form the hypothesis h is given as output by the learner.

The sample complexity of a concept class $\mathscr{C}$ is the sample complexity of the most efficient learner for $\mathscr{C}$. It is a function of ε, δ, and of course of $\mathscr{C}$ itself. One of the most fundamental results in learning theory is that the sample complexity of $\mathscr{C}$ is tightly determined by a combinatorial parameter called the *VC dimension* of $\mathscr{C}$, due to and named after Vapnik and Chervonenkis (1971). The VC dimension of $\mathscr{C}$ is the size of the biggest $\mathcal{S} \subseteq \{0,1\}^n$ that can be labeled in all $2^{|\mathcal{S}|}$ possible ways by concepts from $\mathscr{C}$: for each sequence of $|\mathcal{S}|$

binary labels for the elements of $\mathcal{S}$, there is a $c \in \mathscr{C}$ that has that labeling.[1] Knowing this VC dimension (and ε, δ) already tells us the sample complexity of $\mathscr{C}$ up to constant factors. Blumer et al. (1989) proved that the sample complexity of $\mathscr{C}$ is lower bounded by $\Omega(d/\varepsilon + \log(1/\delta)/\varepsilon)$, and they proved an upper bound that was worse by a $\log(1/\varepsilon)$-factor. In very recent work, Hanneke (2016) (improving on Simon (2015)) got rid of this $\log(1/\varepsilon)$-factor for PAC learning,[2] showing that the lower bound of Blumer et al. is in fact optimal: the sample complexity of $\mathscr{C}$ in the PAC setting is

$$\Theta\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big). \tag{1}$$

1.1.2. The agnostic setting

The PAC model assumes that the labeled examples are generated according to a target concept $c \in \mathscr{C}$. However, in many learning situations that is not a realistic assumption, for example when the examples are noisy in some way or when we have no reason to believe there is an underlying target concept at all. The *agnostic* model of learning, introduced by Haussler (1992) and Kearns et al. (1994), takes this into account. Here, the examples are generated according to a distribution D on $\{0,1\}^{n+1}$. The error of a specific concept $c : \{0,1\}^n \to \{0,1\}$ is defined to be $\mathrm{err}_D(c) = \Pr_{(x,b)\sim D}[c(x) \neq b]$. When we are restricted to hypotheses in $\mathscr{C}$, we would like to find the hypothesis that minimizes $\mathrm{err}_D(c)$ over all $c \in \mathscr{C}$. However, it may require very many examples to do that exactly. In the spirit of the PAC model, the goal of the learner is now to output an $h \in \mathscr{C}$ whose error is at most an additive ε worse than that of the best (= lowest-error) concepts in $\mathscr{C}$.

Like in the PAC model, the optimal sample complexity of such agnostic learners is tightly determined by the VC dimension of $\mathscr{C}$: it is

$$\Theta\Big(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\Big), \tag{2}$$

where the lower bound was proven by Vapnik and Chervonenkis (1974) (see also Simon (1996)), and the upper bound was proven by Talagrand (1994). Shalev-Shwartz and Ben-David (2014) call Eq. (1) and Eq. (2) the "Fundamental Theorem of PAC learning."

1.2. Our results

In this paper we are interested in *quantum* sample complexity. Here a *quantum example* for some concept $c : \{0,1\}^n \to \{0,1\}$, according to some distribution D, corresponds to an $(n+1)$-qubit state

$$\sum_{x\in\{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle.$$

In other words, instead of a random labeled example, an example is now given by a coherent quantum superposition where the square-roots of the probabilities become the amplitudes.[3]

1. Such an $\mathcal{S}$ is said to be *shattered* by $\mathscr{C}$.
2. Hanneke's learner is not proper, meaning that its hypothesis h is not always in $\mathscr{C}$. It is known that the extra $\log(1/\varepsilon)$-factor is sometimes necessary for *proper* PAC learners.
3. We could allow more general quantum examples $\sum_{x\in\{0,1\}^n} \alpha_x|x, c(x)\rangle$, where we only require $|\alpha_x|^2 = D(x)$. However, that will not affect our results since our lower bounds apply to quantum examples where we know the amplitudes are square-rooted probabilities. Adding more degrees of freedom to quantum examples does not make learning easier.

This model was introduced by Bshouty and Jackson (1999), who showed that DNF formulas are learnable in polynomial time from quantum examples when D is uniform. For learning DNF under the uniform distribution from *classical* examples, Verbeurgt (1990) gave the best upper bound of quasipolynomial time. With the added power of "membership queries," where the learner can actively ask for the label of any x of his choice, DNF formulas are known to be learnable in polynomial time under uniform D by a result of Jackson (1997), but *without* membership queries polynomial-time learnability is a longstanding open problem (see Daniely and Shalev-Shwartz (2016) for a recent hardness result) .

How reasonable are examples that are given as a coherent superposition rather than as a random sample? They may seem unreasonable a priori because quantum superpositions seem very fragile and are easily collapsed by measurement, but if we accept the "church of the larger Hilbert space" view on quantum mechanics, where the universe just evolves unitarily without any collapses, then they may become more palatable. It is also possible that the quantum examples are generated by some coherent quantum process that acts like the teacher.

How many quantum examples are needed to learn a concept class $\mathscr{C}$ of VC dimension d? Since a learner can just measure a quantum example in order to obtain a classical example, the *upper* bounds on classical sample complexity trivially imply the same upper bounds on quantum sample complexity. But what about the lower bounds? Are there situations where quantum examples are more powerful than classical? Indeed there are. We already mentioned the results of Bshouty and Jackson (1999) for learning DNF under the uniform distribution without membership queries. Another good example is the learnability of the concept class of linear functions over $\mathbb{F}_2$, $\mathscr{C} = \{c(x) = a \cdot x : a \in \{0,1\}^n\}$, again under the uniform distribution D. It is easy to see that a classical learner needs about n examples to learn an unknown $c \in \mathscr{C}$ under this D. However, if we are given one quantum example

$$\sum_{x\in\{0,1\}^n} \sqrt{D(x)}|x, c(x)\rangle = \frac{1}{\sqrt{2^n}} \sum_{x\in\{0,1\}^n} |x, a \cdot x\rangle,$$

then a small modification of the Bernstein and Vazirani (1997) algorithm can recover a (and hence c) with probability $1/2$. Hence $O(1)$ quantum examples suffice to learn c exactly, with high probability, under the uniform distribution. Atıcı and Servedio (2009) used similar ideas to learning *k-juntas* (concepts depending on only k of their n variables) from quantum examples under the uniform distribution. However, PAC learning requires a learner to learn c under *all possible* distributions D, not just the uniform one. The success probability of the Bernstein-Vazirani algorithm deteriorates sharply when D is far from uniform, but that does not rule out the existence of other quantum learners that use $o(n)$ quantum examples and succeed for all D.

Our main result in this paper is that quantum examples are not actually more powerful than classical labeled examples in the PAC model and in the agnostic model: we prove that the lower bounds on classical sample complexity of Eq. (1) and Eq. (2) hold for quantum examples as well. Accordingly, despite several distribution-specific speedups, quantum examples do not significantly reduce sample complexity if we require our learner to work

for all distributions D. This should be contrasted with the situation when considering the *time complexity* of learning. Servedio and Gortler (2004) considered a concept class (already considered in the literature by Kearns and Valiant (1994)) that can be PAC-learned in polynomial time by a quantum computer, even with only classical examples, but that cannot be PAC-learned in polynomial time by a classical learner unless Blum integers can be factored in polynomial time (which is widely believed to be false).

Earlier work on quantum sample complexity had already gotten close to extending the lower bound of Eq. (1) to PAC learning from quantum examples. Atıcı and Servedio (2005) first proved a lower bound of $\Omega(\sqrt{d}/\varepsilon + d + \log(1/\delta)/\varepsilon)$, which was subsequently improved by Zhang (2010) to

$$\Omega\Big(\frac{d^{1-\eta}}{\varepsilon} + d + \frac{\log(1/\delta)}{\varepsilon}\Big) \text{ for arbitrarily small constant } \eta > 0. \tag{3}$$

Here we optimize these bounds, removing the η and achieving the optimal lower bound for quantum sample complexity in the PAC model (Eq. (1)).

We also show that the lower bound (Eq. (2)) for the agnostic model extends to quantum examples. As far as we know, in contrast to the PAC model, no earlier results were known for quantum sample complexity in the agnostic model.

We have two different proof approaches, which we sketch below.

1.2.1. An information-theoretic argument

In Section 3 we give a fairly intuitive information-theoretic argument that gives optimal lower bounds for classical sample complexity, and that gives nearly-optimal lower bounds for quantum sample complexity. Let us first see how we can prove the classical PAC lower bound of Eq. (1). Suppose $\mathcal{S} = \{s_0, s_1, \ldots, s_d\}$ is shattered by $\mathscr{C}$ (we now assume VC dimension $d+1$ for ease of notation). Then we can consider a distribution D that puts probability $1-4\varepsilon$ on s_0 and probability $4\varepsilon/d$ on each of $s_1, \ldots, s_d$.[4] For every possible labeling $(\ell_1 \ldots \ell_d) \in \{0,1\}^d$ of $s_1, \ldots, s_d$ there will be a concept $c \in \mathscr{C}$ that labels s_0 with 0, and labels s_i with ℓ_i for all $i \in \{1, \ldots, d\}$. Under D, most examples will be $(s_0, 0)$ and hence give us no information when we are learning one of those 2^d concepts. Suppose we have a learner that ε-approximates c with high probability under this D using T examples. Informally, our information-theoretic argument has the following three steps:

1. In order to ε-approximate c, the learner has to learn the c-labels of at least $3/4$ of the $s_1, \ldots, s_d$ (since together these have 4ε of the D-weight, and we want an ε-approximation). As all 2^d labelings are possible, the T examples together contain $\Omega(d)$ bits of information about c.

2. T examples give at most T times as much information about c as one example.

3. One example gives only $O(\varepsilon)$ bits of information about c, because it will tell us one of the labels of $s_1, \ldots, s_d$ only with probability 4ε (and otherwise it just gives $c(s_0) = 0$).

4. We remark that the distributions used here (and later in the agnostic setting) for proving lower bounds on quantum sample complexity have been used in the literature before for analyzing classical sample complexity.

Putting these steps together implies $T = \Omega(d/\varepsilon)$.[5] This argument for the PAC setting is similar to an algorithmic-information argument of Apolloni and Gentile (1998) and an information-theoretic argument for variants of the PAC model with noisy examples by Gentile and Helmbold (2001).

As far as we know, this type of reasoning has not yet been applied to the sample complexity of *agnostic* learning. To get good lower bounds there, we consider a set of distributions D_a, indexed by d-bit string a. These distributions still have the property that if a learner gets ε-close to the minimal error, then it will have to learn $\Omega(d)$ bits of information about the distribution (i.e., about a). Hence the first step of the argument remains the same. The second step of our argument also remains the same, and the third step shows an upper bound of $O(\varepsilon^2)$ on the amount of information that the learner can get from one example. This then implies $T = \Omega(d/\varepsilon^2)$. We can also reformulate this for the case where we want the *expected* additional error of the hypothesis over the best classifier in $\mathscr{C}$ to be at most ε, which is how lower bounds are often stated in learning theory. We emphasize that our information-theoretic proof is simpler than the proofs in Anthony and Bartlett (2009); Audibert (2009); Shalev-Shwartz and Ben-David (2014); Kontorovich and Pinelis (2016).

This information-theoretic approach recovers the optimal classical bounds on sample complexity, but also generalizes readily to the quantum case where the learner gets T quantum examples. To obtain lower bounds on quantum sample complexity we use the same distributions D (now corresponding to a coherent quantum state) and basically just need to re-analyze the third step of the argument. In the PAC setting we show that one quantum example gives at most $O(\varepsilon \log(d/\varepsilon))$ bits of information about c, and in the agnostic setting it gives $O(\varepsilon^2 \log(d/\varepsilon))$ bits. This implies lower bounds on sample complexity that are only a logarithmic factor worse than the optimal classical bounds for the PAC setting (Eq. (1)) and the agnostic setting (Eq. (2)). This is not quite optimal yet, but already better than the previous best known lower bound (Eq. (3)). The logarithmic loss in step 3 is actually inherent in this information-theoretic argument: in some cases a quantum example *can* give roughly $\varepsilon \log d$ bits of information about c, for example when c comes from the concept class of linear functions.

1.2.2. A state-identification argument

In order to get rid of the logarithmic factor we then try another proof approach, which views learning from quantum examples as a quantum state-identification problem: we are given T copies of the quantum example for some concept c and need to ε-approximate c from this. In order to render ε-approximation of c equivalent to exact identification of c, we use good linear error-correcting codes, restricting to concepts whose d-bit labeling of the elements of the shattered set $s_1, \ldots, s_d$ corresponds to a codeword. We then have $2^{\Omega(d)}$ possible concepts, one for each codeword, and need to identify the target concept from a quantum state that is the tensor product of T identical quantum examples.

State-identification problems have been well studied, and many tools are available for analyzing them. In particular, the so-called "Pretty Good Measurement" (PGM, also referred to as "square root measurement" by Eldar and Forney Jr (2001)) is a specific measurement

5. The other part of the lower bound of Eq. (1) does not depend on d and is fairly easy to prove.

that one can always use for state identification, and whose success probability is no more than quadratically worse than that of the very best measurement (even better, in our application the PGM *is* the optimal measurement). In Section 4 we use Fourier analysis to give an exact analysis of the average success probability of the PGM on the state-identification problems that come from both the PAC and the agnostic model. This analysis could be useful in other settings as well. Here it implies that the number of quantum examples, T, is lower bounded by Eq. (1) in the PAC setting, and by Eq. (2) in the agnostic setting.

Using the Pretty Good Measurement, we are also able to prove lower bounds for PAC learning under *random classification noise*, which models the real-world situation that the learning data can have some errors. Classically in the random classification noise model (introduced by Angluin and Laird (1988)), instead of obtaining labeled examples $(x, c(x))$ for some unknown $c \in \mathscr{C}$, the learner obtains *noisy examples* (x, b_x), where $b_x = c(x)$ with probability $1-\eta$ and $b_x = 1-c(x)$ with probability η, for some *noise rate* $\eta \in [0, 1/2)$. Similarly, in the quantum learning model we could naturally define a *noisy quantum example* as an $(n+1)$-qubit state

$$\sum_{x\in\{0,1\}^n} \sqrt{(1-\eta)D(x)}|x, c(x)\rangle + \sqrt{\eta D(x)}|x, 1-c(x)\rangle.$$

Using the PGM, we are able to show that the quantum sample complexity of PAC learning a concept class $\mathscr{C}$ under random classification noise is:

$$\Omega\Big(\frac{d}{(1-2\eta)^2\varepsilon} + \frac{\log(1/\delta)}{(1-2\eta)^2\varepsilon}\Big). \tag{4}$$

We remark here that the best known classical sample complexity lower bound (see Simon (1996)) under the random classification noise is equal to the quantum sample complexity lower bound proven in Eq. (4).

1.3. Related work

Here we briefly mention related work on quantum learning, referring to our survey Arunachalam and de Wolf (2017) for more details. In this paper we focus on *sample* complexity, which is a fundamental information-theoretic quantity. Sample complexity concerns a form of "passive" learning: the learner gets a number of examples at the start of the process, and then has to extract enough information about the target concept from these. We may also consider more active learning settings, in particular ones where the learner can make membership queries (i.e., learn the label $c(x)$ for any x of his choice). Servedio and Gortler (2004) showed that in this setting, classical and quantum complexity are polynomially related. They also exhibit an example of a factor-n speed-up from quantum membership queries using the Bernstein-Vazirani algorithm. Jackson et al. (2002) showed how quantum membership queries can improve Jackson (1997)'s classical algorithm for learning DNF with membership queries under the uniform distribution.

For *quantum exact learning* (also referred to as the *oracle identification* problem in the quantum literature), Kothari (2014) resolved a conjecture of Hunziker et al. (2010), that states that for any concept class $\mathscr{C}$, the number of quantum membership queries required to exactly identify a concept $c \in \mathscr{C}$ is $O(\frac{\log|\mathscr{C}|}{\sqrt{\hat{\gamma}^{\mathscr{C}}}})$, where $\hat{\gamma}^{\mathscr{C}}$ is a combinatorial parameter

of the concept class $\mathscr{C}$ which we shall not define here (see Atıcı and Servedio (2005) for a precise definition). Montanaro (2012) showed how low-degree polynomials over a finite field can be identified more efficiently using quantum algorithms.

In many ways the *time* complexity of learning is at least as important as the sample complexity. We already mentioned that Servedio and Gortler (2004) exhibited a concept class based on factoring Blum integers that can be learned in quantum polynomial time but not in classical polynomial time, unless Blum integers can be factored efficiently. Under the weaker (and widely believed) assumption that one-way functions exist, they exhibited a concept class that can be learned exactly in polynomial time using quantum membership queries, but that takes superpolynomial time to learn from classical membership queries. Gavinsky (2012) introduced a model of learning called "Predictive Quantum" (PQ), a variation of quantum PAC learning, and exhibited a *relational* concept class that is polynomial-time learnable in PQ, while any "reasonable" classical model requires an exponential number of classical examples to learn the concept class.

Aïmeur et al. (2006, 2013) considered a number of quantum algorithms in learning contexts such as clustering via minimum spanning tree, divisive clustering, and k-medians, using variants of Grover (1996)'s algorithm to improve the time complexity of the analogous classical algorithms. Recently, there have been some quantum machine learning algorithms based on Harrow et al. (2009)'s algorithm (commonly referred to as the HHL algorithm) for solving (in a weak sense) very well-behaved linear systems. However, these algorithms often come with some fine print that limits their applicability, and their advantage over classical is not always clear. We refer the reader to the fine print by Aaronson (2015) for references and caveats. There has also been some work on quantum training of neural networks by Wiebe et al. (2016a,b).

In addition to learning classical objects such as Boolean functions, one may also study the learnability of *quantum* objects. In particular, Aaronson (2007) studied how well n-qubit quantum states can be learned from measurement results. In general, an n-qubit state ρ is specified by $\exp(n)$ many parameters, and $\exp(n)$ measurement results on equally many copies of ρ are needed to learn a good approximation of ρ (say, in trace distance). However, Aaronson showed an interesting and surprisingly efficient PAC-like result: from $O(n)$ measurement results, with measurements chosen i.i.d. according to an unknown distribution D on the set of all possible two-outcome measurements, we can learn an n-qubit quantum state $\widetilde{\rho}$ that has roughly the same expectation value as ρ for "most" possible two-outcome measurements. In the latter, "most" is again measured under D, just like in the usual PAC learning the error of the learner's hypothesis is evaluated under the same distribution D that generated the learner's examples. Accordingly, $O(n)$ rather than $\exp(n)$ measurement results suffice to approximately learn an n-qubit state for most practical purposes.

The use of Fourier analysis in analyzing the success probability of the Pretty Good Measurement in quantum state identification appears in a number of earlier works. By considering the dihedral hidden subgroup problem (DHSP) as a state-identification problem, Bacon et al. (2006) show that the PGM is the optimal measurement for DHSP and prove a lower bound on the sample complexity of $\Omega(\log|\mathscr{G}|)$ for a dihedral group $\mathscr{G}$ using Fourier analysis. Ambainis and Montanaro (2014) view the "search with wildcard" problem as a state-identification problem. Using ideas similar to ours, they show that the (x, y)-th entry of the Gram matrix for the ensemble depends on the Hamming distance between x

and y, allowing them to use Fourier analysis to obtain an upper bound on the success probability of the state-identification problem using the PGM.

1.4. Organization

In Section 2 we formally define the classical and quantum learning models and introduce the Pretty Good Measurement. In Section 3 we prove our information-theoretic lower bounds both for classical and quantum learning. In Section 4 we prove an optimal quantum lower bound for PAC and agnostic learning by viewing the learning process as a state-identification problem. We conclude in Section 5 with a conclusion of the results and some open questions for further work.

2. Preliminaries

2.1. Notation

Let $[n] = \{1, \ldots, n\}$. For $x, y \in \{0,1\}^d$, the bit-wise sum $x + y$ is over $\mathbb{F}_2$, the *Hamming distance* $d_H(x, y)$ is the number of indices on which x and y differ, $|x + y|$ is the Hamming weight of the string $x + y$ (which equals $d_H(x, y)$), and $x \cdot y = \sum_i x_i y_i$ (where the sum is over $\mathbb{F}_2$). For a vector $z \in \mathbb{R}^d$, the *norm of* z is defined as $\|z\| = (\sum_i z_i^2)^{1/2}$. For an n-dimensional vector space, the standard basis is denoted by $\{e_i \in \{0,1\}^n : i \in [n]\}$, where e_i is the vector with a 1 in the i-th coordinate and 0's elsewhere. We write log for logarithm to base 2, and ln for base e. We will often use the bijection between the sets $\{0,1\}^k$ and $[2^k]$ throughout this paper. Let $1_{[A]}$ be the indicator for an event A, and let $\delta_{x,y} = 1_{[x=y]}$. We denote random variables in bold, such as $\mathbf{A}$, $\mathbf{B}$.

For a Boolean function $f : \{0,1\}^m \to \{0,1\}$ and $M \in \mathbb{F}_2^{m \times k}$ we define $f \circ M : \{0,1\}^k \to \{0,1\}$ as $(f \circ M)(x) := f(Mx)$ (where the matrix-vector product is over $\mathbb{F}_2$) for all $x \in \{0,1\}^k$. For a distribution $D : \{0,1\}^n \to [0,1]$, let $\mathrm{supp}(D) = \{x \in \{0,1\}^n : D(x) \neq 0\}$. By $x \sim D$, we mean x is sampled according to the distribution D, i.e., $\Pr[\mathbf{X} = x] = D(x)$.

If M is a positive semidefinite (psd) matrix, we define $\sqrt{M}$ as the unique psd matrix that satisfies $\sqrt{M} \cdot \sqrt{M} = M$, and $\sqrt{M}(i, j)$ as the (i, j)-th entry of $\sqrt{M}$. For a matrix $A \in \mathbb{R}^{m \times n}$, we denote the singular values of A by $\sigma_1(A) \geq \sigma_2(A) \geq \cdots \geq \sigma_{\min\{m,n\}}(A) \geq 0$. The spectral norm of A is $\|A\| = \max_{x \in \mathbb{R}^n, \|x\|=1} \|Ax\| = \sigma_1$. Given a set of d-dimensional vectors $U = \{u_1, \ldots, u_n\} \in \mathbb{R}^d$, the Gram matrix V corresponding to the set U is the $n \times n$ psd matrix defined as $V(i, j) = u_i^t u_j$ for $i, j \in [n]$, where u_i^t is the row vector that is the transpose of the column vector u_i.

A technical tool used in our analysis of state-identification problems is Fourier analysis on the Boolean cube. We will just introduce the basics of Fourier analysis here, referring to O'Donnell (2014) for more. Define the inner product between functions $f, g : \{0,1\}^n \to \mathbb{R}$ as

$$\langle f, g \rangle = \mathop{\mathbb{E}}_x [f(x) \cdot g(x)]$$

where the expectation is uniform over $x \in \{0,1\}^n$. For $S \subseteq [n]$ (equivalently $S \in \{0,1\}^n$), let $\chi_S(x) := (-1)^{S \cdot x}$ denote the parity of the variables (of x) indexed by the set S. It is easy to see that the set of functions $\{\chi_S\}_{S \subseteq [n]}$ forms an orthonormal basis for the space of

real-valued functions over the Boolean cube. Hence every f can be decomposed as

$$f(x) = \sum_{S \subseteq [n]} \widehat{f}(S)(-1)^{S \cdot x} \qquad \text{for all } x \in \{0,1\}^n,$$

where $\widehat{f}(S) = \langle f, \chi_S \rangle = \mathbb{E}_x[f(x) \cdot \chi_S(x)]$ is called a *Fourier coefficient* of f.

2.2. Learning in general

In machine learning, a concept class $\mathscr{C}$ over $\{0,1\}^n$ is a set of concepts $c : \{0,1\}^n \to \{0,1\}$. We refer to a concept class $\mathscr{C}$ as being *trivial* if either $\mathscr{C}$ contains only one concept, or $\mathscr{C}$ contains two concepts c_0, c_1 with $c_0(x) = 1 - c_1(x)$ for every $x \in \{0,1\}^n$. For $c : \{0,1\}^n \to \{0,1\}$, we will often refer to the tuple $(x, c(x)) \in \{0,1\}^{n+1}$ as a *labeled example*, where $c(x)$ is the *label* of x.

A central combinatorial concept in learning is the Vapnik and Chervonenkis (1971) dimension, also referred to as the *VC dimension*. Fix a concept class $\mathscr{C}$ over $\{0,1\}^n$. A set $\mathcal{S} = \{s_1, \ldots, s_t\} \subseteq \{0,1\}^n$ is said to be *shattered* by a concept class $\mathscr{C}$ if $\{(c(s_1), \ldots, c(s_t)) : c \in \mathscr{C}\} = \{0,1\}^t$. In other words, for every labeling $\ell \in \{0,1\}^t$, there exists a $c \in \mathscr{C}$ such that $(c(s_1), \ldots, c(s_t)) = \ell$. The VC dimension of a concept class $\mathscr{C}$ is the size of the largest $\mathcal{S} \subseteq \{0,1\}^n$ that is shattered by $\mathscr{C}$.

2.3. Classical learning models

In this paper we will be concerned mainly with the PAC (Probably Approximately Correct) model of learning introduced by Valiant (1984), and the agnostic model of learning introduced by Haussler (1992) and Kearns et al. (1994). For further reading, see standard textbooks in computational learning theory such as Kearns and Vazirani (1994); Anthony and Bartlett (2009); Shalev-Shwartz and Ben-David (2014).

In the classical PAC model, a learner $\mathcal{A}$ is given access to a *random example oracle* $\mathrm{PEX}(c, D)$ which generates labeled examples of the form $(x, c(x))$ where x is drawn from an unknown distribution $D : \{0,1\}^n \to [0,1]$ and $c \in \mathscr{C}$ is the *target concept* that $\mathcal{A}$ is trying to learn. For a concept $c \in \mathscr{C}$ and hypothesis $h : \{0,1\}^n \to \{0,1\}$, we define the error of h compared to the target concept c, under D, as $\mathrm{err}_D(h, c) = \Pr_{x \sim D}[h(x) \neq c(x)]$. A learning algorithm $\mathcal{A}$ is an *(ε, δ)-PAC learner* for $\mathscr{C}$, if the following holds:

> For every $c \in \mathscr{C}$ and distribution D, given access to the $\mathrm{PEX}(c, D)$ oracle:
> $\mathcal{A}$ outputs an h such that $\mathrm{err}_D(h, c) \leq \varepsilon$ with probability at least $1 - \delta$.

The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{PEX}(c, D)$ oracle which the learner makes, over all concepts $c \in \mathscr{C}$, distributions D, and the internal randomness of the learner. The *(ε, δ)-PAC sample complexity* of a concept class $\mathscr{C}$ is the minimum sample complexity over all (ε, δ)-PAC learners for $\mathscr{C}$.

Agnostic learning is the following model: for a distribution $D : \{0,1\}^{n+1} \to [0,1]$, a learner $\mathcal{A}$ is given access to an $\mathrm{AEX}(D)$ oracle that generates examples of the form (x, b) drawn from the distribution D. We define the error of $h : \{0,1\}^n \to \{0,1\}$ under D as $\mathrm{err}_D(h) = \Pr_{(x,b) \sim D}[h(x) \neq b]$. When h is restricted to come from a concept class $\mathscr{C}$, the minimal error achievable is $\mathrm{opt}_D(\mathscr{C}) = \min_{c \in \mathscr{C}}\{\mathrm{err}_D(c)\}$. In agnostic learning, a learner $\mathcal{A}$

needs to output a hypothesis h whose error is not much bigger than $\mathrm{opt}_D(\mathscr{C})$. A learning algorithm $\mathcal{A}$ is an (ε,δ)-*agnostic learner* for $\mathscr{C}$ if:

> For every distribution D on $\{0,1\}^{n+1}$, given access to the $\mathrm{AEX}(D)$ oracle:
> $\mathcal{A}$ outputs an $h\in\mathscr{C}$ such that $\mathrm{err}_D(h)\leq \mathrm{opt}_D(\mathscr{C})+\varepsilon$ with probability at least $1-\delta$.

Note that if there is a $c\in\mathscr{C}$ which perfectly classifies every x with label y for $(x,y)\in\mathrm{supp}(D)$, then $\mathrm{opt}_D(\mathscr{C})=0$ and we are in the setting of proper PAC learning. The *sample complexity* of $\mathcal{A}$ is the maximum number of invocations of the $\mathrm{AEX}(c,D)$ oracle which the learner makes, over all distributions D and over the learner's internal randomness. The (ε,δ)-*agnostic sample complexity* of a concept class $\mathscr{C}$ is the minimum sample complexity over all (ε,δ)-agnostic learners for $\mathscr{C}$.

2.4. Quantum information theory

Throughout this paper we will assume the reader is familiar with the following quantum terminology. An n-dimensional *pure state* is $|\psi\rangle=\sum_{i=1}^n\alpha_i|i\rangle$, where $|i\rangle$ is the n-dimensional unit vector that has a 1 only at position i, the α_i's are complex numbers called the *amplitudes*, and $\sum_{i\in[n]}|\alpha_i|^2=1$. An n-dimensional *mixed state* (or *density matrix*) $\rho=\sum_{i=1}^n p_i|\psi_i\rangle\langle\psi_i|$ is a mixture of pure states $|\psi_1\rangle,\ldots,|\psi_n\rangle$ prepared with probabilities $p_1,\ldots,p_n$, respectively. The eigenvalues $\lambda_1,\ldots,\lambda_n$ of ρ are non-negative reals and satisfy $\sum_{i\in[n]}\lambda_i=1$. If ρ is pure (i.e., $\rho=|\psi\rangle\langle\psi|$ for some $|\psi\rangle$), then one of the eigenvalues is 1 and the others are 0.

To obtain classical information from ρ, one could apply a POVM (positive-operator-valued measure) to the state ρ. An m-outcome POVM is specified by a set of positive semidefinite matrices $\{M_i\}_{i\in[m]}$ with the property $\sum_i M_i=\mathrm{Id}$. When this POVM is applied to the mixed state ρ, the probability of the j-th outcome is given by $\mathrm{Tr}(M_j\rho)$.

For a probability vector $(p_1,\ldots,p_k)$ (where $\sum_{i\in[k]}p_i=1$), the entropy function is defined as $H(p_1,\ldots,p_k)=-\sum_{i\in[k]}p_i\log p_i$. When $k=2$, with $p_1=p$ and $p_2=1-p$, we denote the binary entropy function as $H(p)$. For a state ρ_{AB} on the Hilbert space $\mathcal{H}_A\otimes\mathcal{H}_B$, we let ρ_A be the reduced state after taking the partial trace over $\mathcal{H}_B$. The entropy of a quantum state ρ_A is defined as $S(\mathbf{A})=-\mathrm{Tr}(\rho_A\log\rho_A)$. The mutual information is defined as $I(\mathbf{A}:\mathbf{B})=S(\mathbf{A})+S(\mathbf{B})-S(\mathbf{AB})$, and conditional entropy is defined as $S(\mathbf{A}|\mathbf{B})=S(\mathbf{AB})-S(\mathbf{B})$. Classical information-theoretic quantities correspond to the special case where ρ is a diagonal matrix whose diagonal corresponds to the probability distribution of the random variable. Writing ρ_A in its eigenbasis, it follows that $S(\mathbf{A})=H(\lambda_1,\ldots,\lambda_{\dim(\rho_A)})$, where $\lambda_1,\ldots,\lambda_{\dim(\rho_A)}$ are the eigenvalues of ρ. If ρ_A is a pure state, $S(\mathbf{A})=0$.

2.5. Quantum learning models

The quantum PAC learning model was introduced by Bshouty and Jackson (1999). The quantum PAC model is a generalization of the classical PAC model, in which instead of having access to random examples $(x,c(x))$ from the $\mathrm{PEX}(c,D)$ oracle, the learner now has access to superpositions over all $(x,c(x))$. For an unknown distribution $D:\{0,1\}^n\to[0,1]$ and

concept $c\in\mathscr{C}$, a *quantum example oracle* $\mathrm{QPEX}(c,D)$ acts on $|0^n,0\rangle$ and produces a *quantum example* $\sum_{x\in\{0,1\}^n}\sqrt{D(x)}|x,c(x)\rangle$ (we leave QPEX undefined on other basis states). A quantum learner is given access to some copies of the state generated by $\mathrm{QPEX}(c,D)$ and performs a POVM where each outcome is associated with a hypothesis. A learning algorithm $\mathcal{A}$ is an (ε,δ)-*PAC quantum learner* for $\mathscr{C}$ if:

> For every $c\in\mathscr{C}$ and distribution D, given access to the $\mathrm{QPEX}(c,D)$ oracle:
> $\mathcal{A}$ outputs an h such that $\mathrm{err}_D(h,c)\leq\varepsilon$, with probability at least $1-\delta$.

The *sample complexity* of $\mathcal{A}$ is the maximum number invocations of the $\mathrm{QPEX}(c,D)$ oracle, maximized over all $c\in\mathscr{C}$, distributions D, and the learner's internal randomness. The (ε,δ)-*PAC quantum sample complexity* of a concept class $\mathscr{C}$ is the minimum sample complexity over all (ε,δ)-PAC quantum learners for $\mathscr{C}$.

We define quantum agnostic learning now. For a joint distribution $D:\{0,1\}^{n+1}\to[0,1]$ over the set of examples, the learner has access to an $\mathrm{QAEX}(D)$ oracle which acts on $|0^n,0\rangle$ and produces a quantum example $\sum_{(x,b)\in\{0,1\}^{n+1}}\sqrt{D(x,b)}|x,b\rangle$. A learning algorithm $\mathcal{A}$ is an (ε,δ)-*agnostic quantum learner* for $\mathscr{C}$ if:

> For every distribution D, given access to the $\mathrm{QAEX}(D)$ oracle:
> $\mathcal{A}$ outputs an $h\in\mathscr{C}$ such that $\mathrm{err}_D(h)\leq\mathrm{opt}_D(\mathscr{C})+\varepsilon$ with probability at least $1-\delta$.

The *sample complexity* of $\mathcal{A}$ is the maximum number invocations of the $\mathrm{QAEX}(D)$ oracle over all distributions D and over the learner's internal randomness. The (ε,δ)-*agnostic quantum sample complexity* of a concept class $\mathscr{C}$ is the minimum sample complexity over all (ε,δ)-agnostic quantum learners for $\mathscr{C}$.

2.6. Pretty Good Measurement

Consider an ensemble of d-dimensional states, $\mathcal{E}=\{(p_i,|\psi_i\rangle)\}_{i\in[m]}$, where $\sum_{i\in[m]}p_i=1$. Suppose we are given an unknown state $|\psi_i\rangle$ sampled according to the probabilities and we are interested in maximizing the average probability of success to identify the state that we are given. For a POVM specified by positive semidefinite matrices $\mathcal{M}=\{M_i\}_{i\in[m]}$, the probability of obtaining outcome j equals $\langle\psi_i|M_j|\psi_i\rangle$. The average success probability is defined as

$$P_{\mathcal{M}}(\mathcal{E})=\sum_{i=1}^m p_i\langle\psi_i|M_i|\psi_i\rangle.$$

Let $P^{opt}(\mathcal{E})=\max_{\mathcal{M}}P_{\mathcal{M}}(\mathcal{E})$ denote the optimal average success probability of $\mathcal{E}$, where the maximization is over the set of valid m-outcome POVMs.

For every ensemble $\mathcal{E}$, the so-called *Pretty Good Measurement* (PGM) is a specific POVM (depending on the ensemble $\mathcal{E}$), which we shall define shortly, that does *reasonably* well against $\mathcal{E}$.

Theorem 1 *Let $\mathcal{E}=\{(p_i,|\psi_i\rangle)\}_{i\in[m]}$ be an ensemble of d-dimensional states. Suppose $P^{PGM}(\mathcal{E})$ is defined as the average success probability of identifying the states in $\mathcal{E}$ using the PGM, then we have that*

$$P^{opt}(\mathcal{E})^2\leq P^{PGM}(\mathcal{E})\leq P^{opt}(\mathcal{E}).$$

Proof The second inequality in the theorem follows because $P^{opt}(\mathcal{E})$ is a maximization over all valid POVMs and the first inequality was shown by Barnum and Knill (2002). For completeness we give a simple proof of $P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E})$ below (similar to Montanaro (2007)). Let $|\psi'_i\rangle = \sqrt{p_i}|\psi_i\rangle$, and $\mathcal{E}' = \{|\psi'_i\rangle : i \in [m]\}$ be the set of states in $\mathcal{E}$, renormalized to reflect their probabilities. Define $\rho = \sum_{i\in[m]} |\psi'_i\rangle\langle\psi'_i|$. The PGM is defined as the set of measurement operators $\{|\nu_i\rangle\langle\nu_i|\}_{i\in[m]}$ where $|\nu_i\rangle = \rho^{-1/2}|\psi'_i\rangle$ (the inverse square root of ρ is taken over its non-zero eigenvalues). We first verify this is a valid POVM:

$$\sum_{i=1}^{m} |\nu_i\rangle\langle\nu_i| = \rho^{-1/2}\left(\sum_{i=1}^{m} |\psi'_i\rangle\langle\psi'_i|\right)\rho^{-1/2} = \mathrm{Id}\,.$$

Let G be the Gram matrix for the set $\mathcal{E}'$, i.e., $G(i,j) = \langle\psi'_i|\psi'_j\rangle$ for $i,j \in [m]$. It can be verified that $\sqrt{G}(i,j) = \langle\psi'_i|\rho^{-1/2}|\psi'_j\rangle$. Hence

$$\begin{aligned} P^{PGM}(\mathcal{E}) = \sum_{i\in[m]} p_i|\langle\nu_i|\psi_i\rangle|^2 &= \sum_{i\in[m]} |\langle\nu_i|\psi'_i\rangle|^2 \\ &= \sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2 = \sum_{i\in[m]} \sqrt{G}(i,i)^2. \end{aligned}$$

We now prove $P^{opt}(\mathcal{E})^2 \leq P^{PGM}(\mathcal{E})$. Suppose $\mathcal{M}$ is the optimal measurement. Since $\mathcal{E}$ consists of pure states, by a result of Eldar et al. (2003), we can assume without loss of generality that the measurement operators in $\mathcal{M}$ are rank-1, so $M_i = |\mu_i\rangle\langle\mu_i|$ for some $|\mu_i\rangle$. Note that

$$\begin{aligned} 1 = \mathrm{Tr}(\rho) &= \mathrm{Tr}\Big(\sum_{i\in[m]} |\mu_i\rangle\langle\mu_i|\rho^{1/2} \sum_{j\in[m]} |\mu_j\rangle\langle\mu_j|\rho^{1/2}\Big) \\ &= \sum_{i,j\in[m]} |\langle\mu_i|\rho^{1/2}|\mu_j\rangle|^2 \geq \sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle^2. \end{aligned} \tag{5}$$

Then, using the Cauchy-Schwarz inequality, we have

$$\begin{aligned} P^{opt}(\mathcal{E}) = \sum_{i\in[m]} |\langle\mu_i|\psi'_i\rangle|^2 &= \sum_{i\in[m]} |\langle\mu_i|\rho^{1/4}\rho^{-1/4}|\psi'_i\rangle|^2 \\ &\leq \sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle\langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle \\ &\leq \sqrt{\sum_{i\in[m]} \langle\mu_i|\rho^{1/2}|\mu_i\rangle^2}\sqrt{\sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2} \\ &\overset{\text{Eq. (5)}}{\leq} \sqrt{\sum_{i\in[m]} \langle\psi'_i|\rho^{-1/2}|\psi'_i\rangle^2} = \sqrt{P^{PGM}(\mathcal{E})}. \end{aligned}$$

■

The above shows that for all ensembles $\mathcal{E}$, the PGM for that ensemble is not much worse than the optimal measurement. In some cases the PGM *is* the optimal measurement. In particular, an ensemble $\mathcal{E}$ is called *geometrically uniform* if $\mathcal{E} = \{U_i|\varphi\rangle : i \in [m]\}$ for some Abelian group of matrices $\{U_i\}_{i\in[m]}$ and state $|\varphi\rangle$. Eldar and Forney Jr (2001) showed $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$ for such $\mathcal{E}$.

2.7. Known results and required claims

The following theorems characterize the sample complexity of classical PAC and agnostic learning.

Theorem 2 (Blumer et al. (1989); Hanneke (2016)) *Let $\mathscr{C}$ be a concept class with VC-dim($\mathscr{C}$) $= d+1$. In the PAC model, $\Theta\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big)$ examples are necessary and sufficient for a classical (ε,δ)-PAC learner for $\mathscr{C}$.*

Theorem 3 (Vapnik and Chervonenkis (1974); Simon (1996); Talagrand (1994)) *Let $\mathscr{C}$ be a concept class with VC-dim($\mathscr{C}$) $= d$. In the agnostic model, $\Theta\Big(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\Big)$ examples are necessary and sufficient for a classical (ε,δ)-agnostic learner for $\mathscr{C}$.*

We will use the following well-known theorem from the theory of error-correcting codes (which follows immediately from the Gilbert-Varshamov bound):

Theorem 4 *For every sufficiently large integer n, there exists an integer $k \in [n/4, n]$ and a matrix $M \in \mathbb{F}_2^{n\times k}$ of rank k, such that the associated $[n,k,d]_2$ linear code $\{Mx : x \in \{0,1\}^k\}$ has minimal distance $d \geq n/8$.*

We will need the following claims later

Claim 5 *Let $f : \{0,1\}^m \to \mathbb{R}$ and let $M \in \mathbb{F}_2^{m\times k}$. Then the Fourier coefficients of $f \circ M$ are $\widehat{f\circ M}(Q) = \sum_{S\in\{0,1\}^m : M^tS = Q} \widehat{f}(S)$ for all $Q \subseteq [k]$ (where M^t is the transpose of the matrix M).*

Proof Writing out the Fourier coefficients of $f \circ M$

$$\begin{aligned} \widehat{f\circ M}(Q) &= \mathop{\mathbb{E}}_{z\in\{0,1\}^k}[(f\circ M)(z)(-1)^{Q\cdot z}] \\ &= \mathop{\mathbb{E}}_{z\in\{0,1\}^k}\Big[\sum_{S\in\{0,1\}^m} \widehat{f}(S)(-1)^{S\cdot(Mz)+Q\cdot z}\Big] && \text{(Fourier expansion of } f\text{)} \\ &= \sum_{S\in\{0,1\}^m} \widehat{f}(S) \mathop{\mathbb{E}}_{z\in\{0,1\}^k}[(-1)^{(M^tS+Q)\cdot z}] && \text{(using } \langle S, Mz\rangle = \langle M^tS, z\rangle\text{)} \\ &= \sum_{S:M^tS=Q} \widehat{f}(S). && \text{(using } \mathbb{E}_{z\in\{0,1\}^k}(-1)^{(z_1+z_2)\cdot z} = \delta_{z_1,z_2}\text{)} \end{aligned}$$

■

Claim 6 $\max\{(c/\sqrt{t})^t : t \in [1, c^2]\} = e^{c^2/(2e)}$.

Proof The value of t at which the function $\Big(c/\sqrt{t}\Big)^t$ is the largest, is obtained by differentiating the function with respect to t,

$$\frac{d}{dt}\Big(c/\sqrt{t}\Big)^t = (c/\sqrt{t})^t\Big(\ln(c/\sqrt{t}) - 1/2\Big).$$

Equating the derivative to zero we obtain the maxima (the second derivative can be checked to be negative) at $t = c^2/e$. ■

Fact 7 *For all $\varepsilon \in [0, 1/2]$ we have $H(\varepsilon) \leq O(\varepsilon \log(1/\varepsilon))$, and (from the Taylor series)*

$$1 - H(1/2 + \varepsilon) \leq 2\varepsilon^2/\ln 2 + O(\varepsilon^4).$$

Fact 8 *For every positive integer n, we have that $\binom{n}{k} \leq 2^{nH(k/n)}$ for all $k \leq n$ and $\sum_{i=0}^{m}\binom{n}{i} \leq 2^{nH(m/n)}$ for all $m \leq n/2$.*

The following facts are well-known in quantum information theory.

Fact 9 *(Kaye et al., 2006, Appendix A.9) Let binary random variable $\mathbf{b} \in \{0, 1\}$ be uniformly distributed. Suppose an algorithm is given $|\psi_{\mathbf{b}}\rangle$ (for unknown b) and is required to guess whether $\mathbf{b} = 0$ or $\mathbf{b} = 1$. It will guess correctly with probability at most $\frac{1}{2} + \frac{1}{2}\sqrt{1 - |\langle\psi_0|\psi_1\rangle|^2}$.*

Note that if we can distinguish $|\psi_0\rangle$ and $|\psi_1\rangle$ with probability $\geq 1 - \delta$, then $|\langle\psi_0|\psi_1\rangle| \leq 2\sqrt{\delta(1-\delta)}$.

Fact 10 *(Subadditivity of quantum entropy): For an arbitrary bipartite state ρ_{AB} on the Hilbert space $\mathcal{H}_A \otimes \mathcal{H}_B$, it holds that $S(\rho_{AB}) \leq S(\rho_A) + S(\rho_B)$.*

3. Information-theoretic lower bounds

Upper bounds on sample complexity carry over from classical to quantum PAC learning, because a quantum example becomes a classical example if we just measure it. Our main goal is to show that the *lower* bounds also carry over. All our lower bounds will involve two terms, one that is independent of $\mathscr{C}$ and one that is dependent on the VC dimension of $\mathscr{C}$. In Section 3.1 we prove the VC-independent part of the lower bounds for the *quantum* setting (which also is a lower bound for the classical setting), in Section 3.2 we present an information-theoretic lower bound on sample complexity for PAC learning and agnostic learning which yields optimal VC-dependent bounds in the classical case. Using similar ideas, in Section 3.3 we obtain near-optimal bounds in the quantum case.

3.1. VC-independent part of lower bounds

Lemma 11 (Atıcı and Servedio (2005)) *Let $\mathscr{C}$ be a non-trivial concept class. For every $\delta \in (0, 1/2)$, $\varepsilon \in (0, 1/4)$, a (ε, δ)-PAC quantum learner for $\mathscr{C}$ has sample complexity $\Omega(\frac{1}{\varepsilon}\log\frac{1}{\delta})$.*

Proof Since $\mathscr{C}$ is non-trivial, we may assume there are two concepts $c_1, c_2 \in \mathscr{C}$ defined on two inputs $\{x_1, x_2\}$ as follows $c_1(x_1) = c_2(x_1) = 0$ and $c_1(x_2) = 0, c_2(x_2) = 1$. Consider the distribution $D(x_1) = 1 - \varepsilon$ and $D(x_2) = \varepsilon$. For $i \in \{1, 2\}$, the state of the algorithm after T queries to $\mathrm{QPEX}(c_i, D)$ is $|\psi_i\rangle = (\sqrt{1-\varepsilon}|x_1, 0\rangle + \sqrt{\varepsilon}|x_2, c_i(x_2)\rangle)^{\otimes T}$. It follows that $\langle\psi_1|\psi_2\rangle = (1-\varepsilon)^T$. Since the success probability of an (ε, δ)-PAC quantum learner is $\geq 1-\delta$, Fact 9 implies $\langle\psi_1|\psi_2\rangle \leq 2\sqrt{\delta(1-\delta)}$. Hence $T = \Omega(\frac{1}{\varepsilon}\log\frac{1}{\delta})$. ■

Lemma 12 *Let $\mathscr{C}$ be a non-trivial concept class. For every $\delta \in (0, 1/2)$, $\varepsilon \in (0, 1/4)$, a (ε, δ)-agnostic quantum learner for $\mathscr{C}$ has sample complexity $\Omega(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$.*

Proof Since $\mathscr{C}$ is non-trivial, we may assume there are two concepts $c_1, c_2 \in \mathscr{C}$ and there exists an input $x \in \{0,1\}^n$ such that $c_1(x) \neq c_2(x)$. Consider the two distributions D_- and D_+ defined as follows: $D_\pm(x, c_1(x)) = (1 \pm \varepsilon)/2$ and $D_\pm(x, c_2(x)) = (1 \mp \varepsilon)/2$. Let $|\psi_\pm\rangle$ be the state after T queries to $\mathrm{QAEX}(D_\pm)$, i.e., $|\psi_\pm\rangle = (\sqrt{(1 \pm \varepsilon)/2}|x, c_1(x)\rangle + \sqrt{(1 \mp \varepsilon)/2}|x, c_2(x)\rangle)^{\otimes T}$. It follows that $\langle\psi_+|\psi_-\rangle = (1-\varepsilon^2)^{T/2}$. Since the success probability of an (ε, δ)-agnostic quantum learner is $\geq 1 - \delta$, Fact 9 implies $\langle\psi_+|\psi_-\rangle \leq 2\sqrt{\delta(1-\delta)}$. Hence $T = \Omega(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ ■

3.2. Information-theoretic lower bounds on sample complexity: classical case

3.2.1. Optimal lower bound for classical PAC learning

Theorem 13 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d+1$. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-PAC learner for $\mathscr{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon} + \frac{\log(1/\delta)}{\varepsilon}\Big)$.*

Proof Consider an (ε, δ)-PAC learner for $\mathscr{C}$ that uses T examples. The d-independent part of the lower bound, $T = \Omega(\log(1/\delta)/\varepsilon)$, even holds for quantum examples and was proven in Lemma 11. Hence it remains to prove $T = \Omega(d/\varepsilon)$. It suffices to show this for a specific distribution D, defined as follows. Let $\mathcal{S} = \{s_0, s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ be some $(d+1)$-element set shattered by $\mathscr{C}$. Define $D(s_0) = 1 - 4\varepsilon$ and $D(s_i) = 4\varepsilon/d$ for all $i \in [d]$.

Because $\mathcal{S}$ is shattered by $\mathscr{C}$, for each string $a \in \{0,1\}^d$, there exists a concept $c_a \in \mathscr{C}$ such that $c_a(s_0) = 0$ and $c_a(s_i) = a_i$ for all $i \in [d]$. We define two correlated random variables $\mathbf{A}$ and $\mathbf{B}$ corresponding to the concept and to the examples, respectively. Let $\mathbf{A}$ be a random variable that is uniformly distributed over $\{0,1\}^d$; if $\mathbf{A} = a$, let $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T i.i.d. examples from c_a according to D. We give the following three-step analysis of these random variables:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1-\delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.
   Proof. Let random variable $h(\mathbf{B}) \in \{0,1\}^d$ be the hypothesis that the learner produces

(given the examples in $\mathbf{B}$) restricted to the elements $s_1, \ldots, s_d$. Note that the error of the hypothesis $\mathrm{err}_D(h(\mathbf{B}), c_{\mathbf{A}})$ equals $d_H(\mathbf{A}, h(\mathbf{B})) \cdot 4\varepsilon/d$, because each s_i where $\mathbf{A}$ and $h(\mathbf{B})$ differ contributes $D(s_i) = 4\varepsilon/d$ to the error. Let $\mathbf{Z}$ be the indicator random variable for the event that the error is $\leq \varepsilon$. If $\mathbf{Z} = 1$, then $d_H(\mathbf{A}, h(\mathbf{B})) \leq d/4$. Since we are analyzing an (ε, δ)-PAC learner, we have $\Pr[\mathbf{Z} = 1] \geq 1 - \delta$, and $H(\mathbf{Z}) \leq H(\delta)$. Given a string $h(\mathbf{B})$ that is $d/4$-close to $\mathbf{A}$, $\mathbf{A}$ ranges over a set of only $\sum_{i=0}^{d/4} \binom{d}{i} \leq 2^{H(1/4)d}$ possible d-bit strings (using Fact 8), hence

$$H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 1) \leq H(\mathbf{A} \mid h(\mathbf{B}), \mathbf{Z} = 1) \leq H(1/4)d.$$

We now lower bound $I(\mathbf{A} : \mathbf{B})$ as follows:

$$\begin{aligned} I(\mathbf{A} : \mathbf{B}) &= H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}) \\ &\geq H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z}) - H(\mathbf{Z}) \\ &= H(\mathbf{A}) - \Pr[\mathbf{Z} = 1] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 1) - \\ &\qquad \Pr[\mathbf{Z} = 0] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = 0) - H(\mathbf{Z}) \\ &\geq d - (1 - \delta)H(1/4)d - \delta d - H(\delta) \\ &= (1 - \delta)(1 - H(1/4))d - H(\delta). \end{aligned}$$

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.

   Proof. This inequality essentially appeared in (Jain and Zhang, 2009, Lemma 5), we include the proof for completeness.

$$\begin{aligned} I(\mathbf{A} : \mathbf{B}) = H(\mathbf{B}) - H(\mathbf{B} \mid \mathbf{A}) &= H(\mathbf{B}) - \sum_{i=1}^{T} H(\mathbf{B}_i \mid \mathbf{A}) \\ &\leq \sum_{i=1}^{T} H(\mathbf{B}_i) - \sum_{i=1}^{T} H(\mathbf{B}_i \mid \mathbf{A}) = \sum_{i=1}^{T} I(\mathbf{A} : \mathbf{B}_i), \end{aligned}$$

   where the second equality used independence of the $\mathbf{B}_i$'s conditioned on $\mathbf{A}$, and the inequality uses Fact 10. Since $I(\mathbf{A} : \mathbf{B}_i) = I(\mathbf{A} : \mathbf{B}_1)$ for all i, we get the inequality.

3. $I(\mathbf{A} : \mathbf{B}_1) = 4\varepsilon$.

   Proof. View $\mathbf{B}_1 = (\mathbf{I}, \mathbf{L})$ as consisting of an index $\mathbf{I} \in \{0, 1, \ldots, d\}$ and a corresponding label $\mathbf{L} \in \{0, 1\}$. With probability $1 - 4\varepsilon$, $(\mathbf{I}, \mathbf{L}) = (0, 0)$. For each $i \in [d]$, with probability $4\varepsilon/d$, $(\mathbf{I}, \mathbf{L}) = (i, \mathbf{A}_i)$. Note that $I(\mathbf{A} : \mathbf{I}) = 0$ because $\mathbf{I}$ is independent of $\mathbf{A}$; $I(\mathbf{A} : \mathbf{L} \mid \mathbf{I} = 0) = 0$; and $I(\mathbf{A} : \mathbf{L} \mid \mathbf{I} = i) = I(\mathbf{A}_i : \mathbf{L} \mid \mathbf{I} = i) = H(\mathbf{A}_i \mid \mathbf{I} = i) - H(\mathbf{A}_i \mid \mathbf{L}, \mathbf{I} = i) = 1 - 0 = 1$ for all $i \in [d]$. We have

$$I(\mathbf{A} : \mathbf{B}_1) = I(\mathbf{A} : \mathbf{I}) + I(\mathbf{A} : \mathbf{L} \mid \mathbf{I}) = \sum_{i=1}^{d} \Pr[\mathbf{I} = i] \cdot I(\mathbf{A} : \mathbf{L} \mid \mathbf{I} = i) = 4\varepsilon.$$

Combining these three steps implies $T = \Omega(d/\varepsilon)$. ∎

3.2.2. Optimal lower bound for classical agnostic learning

Theorem 14 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d$. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε, δ)-agnostic learner for $\mathscr{C}$ has sample complexity $\Omega\left(\frac{d}{\varepsilon^2} + \frac{\log(1/\delta)}{\varepsilon^2}\right)$.*

Proof The d-independent part of the lower bound, $T = \Omega(\log(1/\delta)/\varepsilon^2)$, even holds for quantum examples and was proven in Lemma 12. For the other part, the proof is similar to Theorem 13, as follows. Assume an (ε, δ)-agnostic learner for $\mathscr{C}$ that uses T examples. We need to prove $T = \Omega(d/\varepsilon^2)$. For shattered set $\mathcal{S} = \{s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ and $a \in \{0,1\}^d$, define distribution D_a on $[d] \times \{0, 1\}$ by $D_a(i, \ell) = (1 + (-1)^{a_i + \ell} 4\varepsilon)/2d$.

Again let random variable $\mathbf{A} \in \{0,1\}^d$ be uniformly random, corresponding to the values of concept c_a on $\mathcal{S}$, and $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T i.i.d. samples from D_a. Note that c_a is the minimal-error concept from $\mathscr{C}$ w.r.t. D_a, and concept $c_{\tilde{a}}$ has additional error $d_H(a, \tilde{a}) \cdot 4\varepsilon/d$. Accordingly, an (ε, δ)-agnostic learner has to produce (from $\mathbf{B}$) an $h(\mathbf{B}) \in \{0,1\}^d$, which, with probability at least $1 - \delta$, is $d/4$-close to $\mathbf{A}$. Our three-step analysis is very similar to Theorem 13; only the third step changes:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1 - \delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.

3. $I(\mathbf{A} : \mathbf{B}_1) = 1 - H(1/2 + 2\varepsilon) = O(\varepsilon^2)$.

   Proof. View the D_a-distributed random variable $\mathbf{B}_1 = (\mathbf{I}, \mathbf{L})$ as index $\mathbf{I} \in [d]$ and label $\mathbf{L} \in \{0, 1\}$. The marginal distribution of $\mathbf{I}$ is uniform; conditioned on $\mathbf{I} = i$, the bit $\mathbf{L}$ equals $\mathbf{A}_i$ with probability $1/2 + 2\varepsilon$. Hence

$$\begin{aligned} I(\mathbf{A} : \mathbf{L} \mid \mathbf{I} = i) = I(\mathbf{A}_i : \mathbf{L} \mid \mathbf{I} = i) &= H(\mathbf{A}_i \mid \mathbf{I} = i) - H(\mathbf{A}_i \mid \mathbf{L}, \mathbf{I} = i) \\ &= 1 - H(1/2 + 2\varepsilon). \end{aligned}$$

   Using Fact 7, we have

$$\begin{aligned} I(\mathbf{A} : \mathbf{B}_1) = I(\mathbf{A} : \mathbf{I}) + I(\mathbf{A} : \mathbf{L} \mid \mathbf{I}) &= \sum_{i=1}^{d} \Pr[\mathbf{I} = i] \cdot I(\mathbf{A} : \mathbf{L} \mid \mathbf{I} = i) \\ &= 1 - H(1/2 + 2\varepsilon) = O(\varepsilon^2). \end{aligned}$$

Combining these three steps implies $T = \Omega(d/\varepsilon^2)$. ∎

In the theorem below, we optimize the constant in the lower bound of the sample complexity in Theorem 14. In learning theory such lower bounds are often stated slightly differently. In order to compare the lower bounds, we introduce the following. We first define an *ε-average agnostic learner* for a concept class $\mathscr{C}$ as a learner that, given access to T samples from an AEX(D) oracle (for some unknown distribution D), needs to output a hypothesis $h_{\mathbf{XY}}$ (where $(\mathbf{X}, \mathbf{Y}) \sim D^T$) that satisfies

$$\mathop{\mathbb{E}}_{(\mathbf{X}, \mathbf{Y}) \sim D^T} [\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathscr{C}) \leq \varepsilon.$$

Lower bounds on the quantity $(\mathbb{E}_{(\mathbf{X},\mathbf{Y})\sim D^T}[\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathscr{C}))$ are generally referred to as *minimax lower bounds* in learning theory. For concept class $\mathscr{C}$, Audibert (2008, 2009) showed that there exists a distribution D, such that if the agnostic learner uses T samples from $\mathrm{AEX}(D)$, then

$$\mathop{\mathbb{E}}_{(\mathbf{X},\mathbf{Y})\sim D^T}[\mathrm{err}_D(h_{\mathbf{XY}})] - \mathrm{opt}_D(\mathscr{C}) \geq \frac{1}{6}\sqrt{\frac{d}{T}}.$$

Equivalently, this is a lower bound of $T \geq \frac{d}{36\varepsilon^2}$ on the sample complexity of an ε-average agnostic learner. We obtain a slightly weaker lower bound that is essentially $T \geq \frac{d}{62\varepsilon^2}$:

Theorem 15 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d$. Then for every $\varepsilon \in (0, 1/10]$, there exists a distribution for which every ε-average agnostic learner has sample complexity at least $\frac{d}{\varepsilon^2} \cdot \left(\frac{1}{62} - \frac{\log(2d+2)}{4d}\right)$.*

Proof The proof is similar to Theorem 14. Assume an ε-average agnostic learner for $\mathscr{C}$ that uses T samples. For shattered set $\mathcal{S} = \{s_1, \ldots, s_d\} \subseteq \{0,1\}^n$ and $a \in \{0,1\}^d$, define distribution D_a on $[d] \times \{0,1\}$ by $D_a(i,\ell) = (1 + (-1)^{a_i+\ell}\beta\varepsilon)/2d$, for some constant $\beta \geq 2$ which we shall pick later.

Again let random variable $\mathbf{A} \in \{0,1\}^d$ be uniformly random, corresponding to the values of concept c_a on $\mathcal{S}$, and $\mathbf{B} = \mathbf{B}_1 \ldots \mathbf{B}_T$ be T i.i.d. samples from D_a. Note that c_a is the minimal-error concept from $\mathscr{C}$ w.r.t. D_a, and concept $c_{\tilde{a}}$ has additional error $d_H(a,\tilde{a}) \cdot \beta\varepsilon/d$. Accordingly, an ε-average agnostic learner has to produce (from $\mathbf{B}$) an $h(\mathbf{B}) \in \{0,1\}^d$, which satisfies $\mathbb{E}_{\mathbf{A},\mathbf{B}}[d_H(\mathbf{A}, h(\mathbf{B}))] \leq d/\beta$.

Our three-step analysis is very similar to Theorem 14; only the first step changes:

1. $I(\mathbf{A} : \mathbf{B}) \geq d(1 - H(1/\beta)) - \log(d+1)$.

   Proof. Define random variable $\mathbf{Z} = d_H(\mathbf{A}, h(\mathbf{B}))$, then $\mathbb{E}[\mathbf{Z}] \leq d/\beta$. Note that given a string $h(\mathbf{B})$ that is ℓ-close to $\mathbf{A}$, $\mathbf{A}$ ranges over a set of only $\binom{d}{\ell} \leq 2^{H(\ell/d)d}$ possible d-bit strings (using Fact 8), hence

   $$H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = \ell) \leq H(\mathbf{A} \mid h(\mathbf{B}), \mathbf{Z} = \ell) \leq H(\ell/d)d.$$

   We now lower bound $I(\mathbf{A} : \mathbf{B})$:

   $$\begin{aligned}
   I(\mathbf{A} : \mathbf{B}) &= H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}) \\
   &\geq H(\mathbf{A}) - H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z}) - H(\mathbf{Z}) \\
   &= d - \sum_{\ell=0}^{d+1} \Pr[\mathbf{Z} = \ell] \cdot H(\mathbf{A} \mid \mathbf{B}, \mathbf{Z} = \ell) - H(\mathbf{Z}) \\
   &\geq d - \mathop{\mathbb{E}}_{\ell \in \{0,\ldots,d\}}[H(\ell/d)d] - \log(d+1) && \text{(since } \mathbf{Z} \in \{0,\ldots,d\}\text{)} \\
   &\geq d - dH\Big(\frac{\mathbb{E}_\ell[\ell]}{d}\Big) - \log(d+1) && \text{(using Jensen's inequality)} \\
   &\geq d - dH(1/\beta) - \log(d+1), && \text{(using } \mathbb{E}[\mathbf{Z}] \leq d/\beta\text{)}
   \end{aligned}$$

   where for the third inequality we used the concavity of the binary entropy function to conclude $\mathbb{E}_\ell[H(\ell/d)] \leq H(\mathbb{E}_\ell[\ell]/d)$, and for the fourth inequality we used that $\beta \geq 2$.

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.

3. $I(\mathbf{A} : \mathbf{B}_1) = 1 - H(1/2 + \beta\varepsilon/2) \overset{\text{Fact 7}}{\leq} \beta^2\varepsilon^2/\ln 4 + O(\varepsilon^4)$.

Combining these three steps implies

$$T \geq \frac{d \ln 4}{\varepsilon^2} \cdot \Big(\frac{1 - H(1/\beta)}{\beta^2 + O(\varepsilon^2)} - \frac{\log(d+1)}{\beta^2 d + O(d\varepsilon^2)}\Big).$$

Using $\varepsilon \leq 1/10$, $\beta = 4$ to optimize this lower bound, we obtain $T \geq \frac{d}{\varepsilon^2} \cdot \left(\frac{1}{62} - \frac{\log(2d+2)}{4d}\right)$. ■

3.3. Information-theoretic lower bounds on sample complexity: quantum case

Here we will "quantize" the above two classical information-theoretic proofs, yielding lower bounds for quantum sample complexity (in both the PAC and the agnostic setting) that are tight up to a logarithmic factor.

3.3.1. Near-optimal lower bound for quantum PAC learning

Theorem 16 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d + 1$. Then, for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/4)$, every (ε,δ)-PAC quantum learner for $\mathscr{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon \log(d/\varepsilon)} + \frac{\log(1/\delta)}{\varepsilon}\Big)$.*

Proof The proof is analogous to Theorem 13. We use the same distribution D, with the $\mathbf{B}_i$ now being quantum samples: $|\psi_a\rangle = \sum_{i\in\{0,1,\ldots,d\}} \sqrt{D(s_i)}|i, c_a(s_i)\rangle$. The $\mathbf{AB}$-system is now in the following classical-quantum state:

$$\frac{1}{2^d} \sum_{a\in\{0,1\}^d} |a\rangle\langle a| \otimes |\psi_a\rangle\langle\psi_a|^{\otimes T}.$$

The first two steps of our argument are identical to Theorem 13. We only need to re-analyze step 3:

1. $I(\mathbf{A} : \mathbf{B}) \geq (1 - \delta)(1 - H(1/4))d - H(\delta) = \Omega(d)$.

2. $I(\mathbf{A} : \mathbf{B}) \leq T \cdot I(\mathbf{A} : \mathbf{B}_1)$.

3. $I(\mathbf{A} : \mathbf{B}_1) \leq H(4\varepsilon) + 4\varepsilon \log(2d) = O(\varepsilon \log(d/\varepsilon))$.

   Proof. Since $\mathbf{AB}$ is a classical-quantum state, we have

   $$I(\mathbf{A} : \mathbf{B}_1) = S(\mathbf{A}) + S(\mathbf{B}_1) - S(\mathbf{AB}_1) = S(\mathbf{B}_1),$$

   where the first equality follows from definition and the second equality uses $S(\mathbf{A}) = d$ since $\mathbf{A}$ is uniformly distributed in $\{0,1\}^d$, and $S(\mathbf{AB}_1) = d$ since the matrix $\sigma = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |a\rangle\langle a| \otimes |\psi_a\rangle\langle\psi_a|$ is block diagonal with 2^d rank-1 blocks on the diagonal.

It thus suffices to bound the entropy of the singular values of the reduced state of $\mathbf{B}_1$, which is

$$\rho = \frac{1}{2^d} \sum_{a\in\{0,1\}^d} |\psi_a\rangle\langle\psi_a|.$$

Let $\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_{2d} \geq 0$ be its singular values. Since ρ is a density matrix, these form a probability distribution. Note that the upper-left entry of the matrix $|\psi_a\rangle\langle\psi_a|$ is $D(s_0) = 1-4\varepsilon$, hence so is the upper-left entry of ρ. This implies $\sigma_0 \geq 1-4\varepsilon$. Consider sampling a number $\mathbf{N} \in \{0,1,\ldots,2d\}$ according to the σ-distribution. Let $\mathbf{Z}$ be the indicator random variable for the event $\mathbf{N} \neq 0$, which has probability $1-\sigma_0 \leq 4\varepsilon$. Note that $H(\mathbf{N} \mid \mathbf{Z} = 0) = 0$, because $\mathbf{Z} = 0$ implies $\mathbf{N} = 0$. Also, $H(\mathbf{N} \mid \mathbf{Z} = 1) \leq \log(2d)$, because if $\mathbf{Z} = 1$ then $\mathbf{N}$ ranges over $2d$ elements. We now have

$$\begin{aligned} S(\rho) &= H(\mathbf{N}) = H(\mathbf{N},\mathbf{Z}) = H(\mathbf{Z}) + H(\mathbf{N} \mid \mathbf{Z}) \\ &= H(\mathbf{Z}) + \Pr[\mathbf{Z}=0]\cdot H(\mathbf{N} \mid \mathbf{Z}=0) + \Pr[\mathbf{Z}=1]\cdot H(\mathbf{N} \mid \mathbf{Z}=1) \\ &\leq H(4\varepsilon) + 4\varepsilon\log(2d) \\ &= O(\varepsilon\log(d/\varepsilon)). \qquad \text{(using Fact 7)} \end{aligned}$$

Combining these three steps implies $T = \Omega\Big(\frac{d}{\varepsilon\log(d/\varepsilon)}\Big)$. ■

3.3.2. Near-optimal lower bound for quantum agnostic learning

Theorem 17 *Let $\mathscr{C}$ be a concept class with VC-dim($\mathscr{C}$) = d. Then for every $\delta \in (0,1/2)$, $\varepsilon \in (0,1/4)$, every (ε,δ)-agnostic quantum learner for $\mathscr{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon^2\log(d/\varepsilon)} + \frac{\log(1/\delta)}{\varepsilon^2}\Big)$.*

Proof The proof is analogous to Theorem 14, with the $\mathbf{B}_i$ now being quantum samples for D_a, $|\psi_a\rangle = \sum_{i\in[d],\ell\in\{0,1\}} \sqrt{D_a(i,\ell)}|i,\ell\rangle$. Again we only need to re-analyze step 3:

1. $I(\mathbf{A}:\mathbf{B}) \geq (1-\delta)(1-H(1/4))d - H(\delta) = \Omega(d)$.
2. $I(\mathbf{A}:\mathbf{B}) \leq T\cdot I(\mathbf{A}:\mathbf{B}_1)$.
3. $I(\mathbf{A}:\mathbf{B}_1) = O(\varepsilon^2\log(d/\varepsilon))$.

   Proof of step 3. As in step 3 of the proof of Theorem 16, it suffices to upper bound the entropy of

   $$\rho = \frac{1}{2^d} \sum_{a\in\{0,1\}^d} |\psi_a\rangle\langle\psi_a|.$$

   We lower bound the largest singular value of ρ. Consider $|\psi\rangle = \frac{1}{\sqrt{2d}} \sum_{i\in[d],\ell\in\{0,1\}} |i,\ell\rangle$.

$$\langle\psi|\psi_a\rangle = \frac{1}{d}\sum_{i\in[d]} \frac{1}{2}\Big(\sqrt{1+4\varepsilon} + \sqrt{1-4\varepsilon}\Big) = \frac{1}{2}\Big(\sqrt{1+4\varepsilon} + \sqrt{1-4\varepsilon}\Big) \geq 1 - 2\varepsilon^2 - O(\varepsilon^4),$$

where the last inequality used the Taylor series expansion of $\sqrt{1+x}$. This implies that the largest singular value of ρ is at least

$$\langle\psi|\rho|\psi\rangle = \frac{1}{2^d}\sum_{a\in\{0,1\}^d} |\langle\psi|\psi_a\rangle|^2 \geq 1 - 4\varepsilon^2 - O(\varepsilon^4).$$

We can now finish as in step 3 of the proof of Theorem 16:

$$I(\mathbf{A}:\mathbf{B}_1) \leq S(\rho) \leq H(4\varepsilon^2) + 4\varepsilon^2\log(2d) \overset{\text{Fact 7}}{=} O(\varepsilon^2\log(d/\varepsilon)).$$

Combining these three steps implies $T = \Omega\Big(\frac{d}{\varepsilon^2\log(d/\varepsilon)}\Big)$. ■

4. A lower bound by analysis of state identification

In this section we present a tight lower bound on quantum sample complexity for both the PAC and the agnostic learning settings, using ideas from Fourier analysis to analyze the performance of the Pretty Good Measurement. The core of both lower bounds is the following theorem.

Theorem 18 *For $m \geq 10$, let $f : \{0,1\}^m \to \mathbb{R}$ be defined as $f(z) = (1-\beta\frac{|z|}{m})^T$ for some $\beta \in (0,1]$ and $T \in [1, m/(e^3\beta)]$. For $k \leq m$, let $M \in \mathbb{F}_2^{m\times k}$ be a matrix with rank k. Suppose $A \in \mathbb{R}^{2^k\times 2^k}$ is defined as $A(x,y) = (f\circ M)(x+y)$ for $x,y \in \{0,1\}^k$, then*

$$\sqrt{A}(x,x) \leq \frac{2\sqrt{e}}{2^{k/2}}\Big(1-\frac{\beta}{2}\Big)^{T/2} e^{11T^2\beta^2/m + \sqrt{Tm\beta}} \qquad \text{for all } x \in \{0,1\}^k.$$

Proof The structure of the proof is to first diagonalize A, relating its eigenvalues to the Fourier coefficients of f. This allows to calculate the diagonal entries of $\sqrt{A}$ exactly in terms of those Fourier coefficients. We then upper bound those Fourier coefficients using a combinatorial argument.

We first observe the well-known relation between the eigenvalues of a matrix P defined as $P(x,y) = g(x+y)$ for $x,y \in \{0,1\}^k$, and the Fourier coefficients of g.

Claim 19 *Suppose $g : \{0,1\}^k \to \mathbb{R}$ and $P \in \mathbb{R}^{2^k\times 2^k}$ is defined as $P(x,y) = g(x+y)$, then the eigenvalues of P are $\{2^k\widehat{g}(Q) : Q \in \{0,1\}^k\}$.*

Proof Let $H \in \mathbb{R}^{2^k\times 2^k}$ be the matrix defined as $H(x,y) = (-1)^{x\cdot y}$ for $x,y \in \{0,1\}^k$. It is easy to see that $H^{-1}(x,y) = (-1)^{x\cdot y}/2^k$. We now show that H diagonalizes P:

$$\begin{aligned} (HPH^{-1})(x,y) &= \frac{1}{2^k}\sum_{z_1,z_2\in\{0,1\}^k} (-1)^{z_1\cdot x + z_2\cdot y} g(z_1+z_2) \\ &= \frac{1}{2^k}\sum_{z_1,z_2,Q\in\{0,1\}^k} (-1)^{z_1\cdot x + z_2\cdot y}\widehat{g}(Q)(-1)^{Q\cdot(z_1+z_2)} \quad \text{(Fourier expansion of } g) \\ &= \frac{1}{2^k}\sum_{Q\in\{0,1\}^k} \widehat{g}(Q) \sum_{z_1\in\{0,1\}^k} (-1)^{(x+Q)\cdot z_1} \sum_{z_2\in\{0,1\}^k} (-1)^{(y+Q)\cdot z_2} \\ &= 2^k\widehat{g}(x)\delta_{x,y}, \end{aligned}$$

where we used $\sum_{z\in\{0,1\}^k}[(-1)^{(a+b)\cdot z}] = 2^k\delta_{a,b}$ in the last equality. The eigenvalues of P are the diagonal entries, $\{2^k\widehat{g}(Q) : Q \in \{0,1\}^k\}$. ■

We now relate the diagonal entries of $\sqrt{A}$ to the Fourier coefficients of f:

Claim 20 *For all $x \in \{0,1\}^k$, we have*

$$\sqrt{A}(x,x) = \frac{1}{2^{k/2}} \sum_{Q\in\{0,1\}^k} \sqrt{\sum_{S\in\{0,1\}^m : M^tS=Q} \widehat{f}(S)}.$$

Proof Since $A(x,y) = (f\circ M)(x+y)$, by Claim 19 it follows that H (as defined in the proof of Claim 19) diagonalizes A and the eigenvalues of A are $\{2^k\widehat{f\circ M}(Q) : Q\in\{0,1\}^k\}$. Hence, we have

$$\sqrt{A} = H^{-1}\cdot \mathrm{diag}\Big(\Big\{\sqrt{2^k\widehat{f\circ M}(Q)} : Q\in\{0,1\}^k\Big\}\Big)\cdot H,$$

and the diagonal entries of $\sqrt{A}$ are

$$\sqrt{A}(x,x) = \frac{1}{2^{k/2}} \sum_{Q\in\{0,1\}^k} \sqrt{\widehat{f\circ M}(Q)} \overset{\text{Claim 5}}{=} \frac{1}{2^{k/2}} \sum_{Q\in\{0,1\}^k} \sqrt{\sum_{S\in\{0,1\}^m : M^tS=Q} \widehat{f}(S)}.$$

■

In the following lemma, we give an upper bound on the Fourier coefficients of f, which in turn (from the claim above) gives an upper bound on the diagonal entries of $\sqrt{A}$.

Lemma 21 *For $\beta \in (0,1]$, the Fourier coefficients of $f : \{0,1\}^m \to \mathbb{R}$ defined as $f(z) = (1-\beta\frac{|z|}{m})^T$, satisfy*

$$0 \le \widehat{f}(S) \le 4e\Big(1-\frac{\beta}{2}\Big)^T\Big(\frac{T\beta}{m}\Big)^q e^{22T^2\beta^2/m}, \quad \textit{for all } S \textit{ such that } |S| = q.$$

Proof In order to see why the Fourier coefficients of f are non-negative, we first define the set $U = \{u_x^{\otimes T}\}_{x\in\{0,1\}^m}$ where $u_x = \sqrt{1-\beta}|0,0\rangle + \sqrt{\beta/m}\sum_{i\in[m]}|i,x_i\rangle$. Let V be the $2^m\times 2^m$ Gram matrix for the set U. For $x,y\in\{0,1\}^m$, we have

$$\begin{aligned} V(x,y) = (u_x^*u_y)^T &= \Big(1-\beta+\frac{\beta}{m}\sum_{i=1}^m\langle x_i|y_i\rangle\Big)^T \\ &= \Big(1-\beta+\frac{\beta}{m}(m-|x+y|)\Big)^T = \Big(1-\beta\frac{|x+y|}{m}\Big)^T = f(x+y). \end{aligned}$$

By Claim 19, the eigenvalues of the Gram matrix V are $\{2^m\widehat{f}(S) : S\in\{0,1\}^m\}$. Since the Gram matrix is psd, its eigenvalues are non-negative, which implies that $\widehat{f}(S)\ge 0$ for all $S\in\{0,1\}^m$.

We now prove the upper bound in the lemma. By definition,

$$\begin{aligned} \widehat{f}(S) &= \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\Big(1-\beta\frac{|z|}{m}\Big)^T(-1)^{S\cdot z}\Big] \\ &= \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\Big(1-\frac{\beta}{2}+\frac{\beta}{2m}\sum_{i=1}^m(-1)^{z_i}\Big)^T(-1)^{S\cdot z}\Big] \qquad (\text{since } |z| = \textstyle\sum_{i\in[m]}\frac{1-(-1)^{z_i}}{2}) \\ &= \sum_{\ell=0}^T\binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^\ell \mathop{\mathbb{E}}_{z\in\{0,1\}^m}\Big[\sum_{i_1,\ldots,i_\ell=1}^m(-1)^{z\cdot(e_{i_1}+\cdots+e_{i_\ell}+S)}\Big] \\ &= \sum_{\ell=0}^T\binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^\ell \sum_{i_1,\ldots,i_\ell=1}^m 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]}, \end{aligned}$$

using $\mathbb{E}_{z\in\{0,1\}^m}[(-1)^{(z_1+z_2)\cdot z}] = \delta_{z_1,z_2}$ in the last equality.

We will use the following claim to upper bound the combinatorial sum in the quantity above.

Claim 22 *Fix $S\in\{0,1\}^m$ with Hamming weight $|S| = q$. For every $\ell\in\{q,\ldots,T\}$, we have*

$$\sum_{i_1,\ldots,i_\ell=1}^m 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \le \begin{cases} \ell!\cdot m^{(\ell-q)/2}\Big/\Big(2^{(\ell-q)/2}((\ell-q)/2)!\Big) & \textit{if } (\ell-q) \textit{ is even} \\ 0 & \textit{otherwise} \end{cases}$$

Proof Since $|S| = q$, we can write $S = e_{r_1}+\cdots+e_{r_q}$ for distinct $r_1,\ldots,r_q\in[m]$. There are $\binom{\ell}{q}$ ways to pick q indices in $(i_1,\ldots,i_\ell)$ (w.l.o.g. let them be $i_1,\ldots,i_q$) and there are $q!$ factorial ways to assign $(r_1,\ldots,r_q)$ to $(i_1,\ldots,i_q)$. It remains to count the number of ways that we can assign values to the remaining indices $i_{q+1},\ldots,i_\ell$ such that $e_{i_{q+1}}+\cdots+e_{i_\ell} = 0$. If $\ell-q$ is odd then this number is 0, so from now on assume $\ell-q$ is even. We upper bound the number of such assignments by partitioning the $\ell-q$ indices into pairs and assigning the same value to both indices in each pair.

We first count the number of ways to partition a set of $\ell-q$ indices into subsets of size 2. This number is exactly $(\ell-q)!\Big(2^{(\ell-q)/2}((\ell-q)/2)!\Big)^{-1}$. Furthermore, there are m possible values that can be assigned to the pair of indices in each of the $(\ell-q)/2$ subsets such that $e_i+e_j = 0$ within each subset. Note that assigning m possible values to each pair of indices in the $(\ell-q)/2$ subsets overcounts, but this rough upper bound is sufficient for our purposes.

Combining the three arguments, we conclude

$$\sum_{i_1,\ldots,i_\ell=1}^d 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \le \binom{\ell}{q}q!\cdot(\ell-q)!\cdot m^{(\ell-q)/2}\Big/\Big(2^{(\ell-q)/2}((\ell-q)/2)!\Big).$$

which yields the claim. ■

Continuing with the evaluation of the Fourier coefficient and using the claim above, we have

$$\begin{aligned}
\widehat{f}(S) &= \sum_{\ell=0}^{T}\binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^{\ell}\sum_{i_1,\ldots,i_\ell=1}^{m} 1_{[e_{i_1}+\cdots+e_{i_\ell}=S]} \\
&\le \sum_{\ell=q}^{T}\binom{T}{\ell}\Big(1-\frac{\beta}{2}\Big)^{T-\ell}\Big(\frac{\beta}{2m}\Big)^{\ell}\ell!\cdot m^{(\ell-q)/2}\Big/\Big(2^{(\ell-q)/2}\Big(\frac{\ell-q}{2}\Big)!\Big) && \text{(by Claim 22)} \\
&= \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{2}{m}\Big)^{q/2}\sum_{\ell=q}^{T}\binom{T}{\ell}\ell!\Big(\frac{\beta}{m(2-\beta)}\Big)^{\ell}\Big(\frac{m}{2}\Big)^{\ell/2}\Big/\Big(\frac{\ell-q}{2}\Big)! \\
&\le \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{2}{m}\Big)^{q/2}\sum_{\ell=q}^{T}\Big(T\cdot\frac{\beta}{m}\cdot\sqrt{\frac{m}{2}}\Big)^{\ell}\Big/\Big(\frac{\ell-q}{2}\Big)! && \text{(since } \beta<1 \text{ and } \tbinom{T}{\ell}\ell!\le T^{\ell}\text{)} \\
&= \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{2}{m}\Big)^{q/2}\Big(\frac{T\beta}{\sqrt{2m}}\Big)^{q}\sum_{r=0}^{T-q}\Big(\frac{T\beta}{\sqrt{2m}}\Big)^{r}\frac{1}{(r/2)!} && \text{(substituting } r\leftarrow(\ell-q)\text{)} \\
&\le \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}\sum_{r=0}^{T-q}\Big(\frac{T\beta}{\sqrt{2m}}\Big)^{r}\frac{e^{r/2}}{(r/2)^{r/2}} && \text{(using } n!\ge(n/e)^{n}\text{)} \\
&= \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}\sum_{r=0}^{T-q}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \\
&\le \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}\sum_{r=0}^{T}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} && \text{(since the summands are } \ge 0\text{)} \\
&= \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}\Big(\sum_{r=0}^{\lceil e^3T^2\beta^2/m\rceil}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r}+\sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r}\Big).
\end{aligned}$$

Note that by the assumptions of the theorem, $T^2e^3\beta^2/m \le T\beta \le T$, which allowed us to split the sum into two pieces in the last equality. At this point, we upper bound both pieces in the last equation separately. For the first piece, using Claim 6 it follows that $\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r}$ is maximized at $r=\lceil T^2\beta^2/m\rceil$. Hence we get

$$\sum_{r=0}^{\lceil e^3T^2\beta^2/m\rceil}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \le \Big(2+\frac{e^3T^2\beta^2}{m}\Big)e^{\lceil T^2\beta^2/m\rceil/2} \le 2e^{22T^2\beta^2/m+1}, \tag{6}$$

where the first inequality uses Claim 6 and the second inequality uses $2+x\le 2e^{x}$ for $x\ge 0$ and $e^3+1/2\le 22$. For the second piece, we use

$$\sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T}\Big(\frac{\sqrt{e}T\beta}{\sqrt{mr}}\Big)^{r} \le \sum_{r=\lceil e^3T^2\beta^2/m\rceil+1}^{T}\Big(\frac{1}{e}\Big)^{r} \le \sum_{r=1}^{T}\Big(\frac{1}{e}\Big)^{r} = \frac{1-e^{-T}}{e-1} \le 2/3. \tag{7}$$

So we finally get

$$\begin{aligned}
\widehat{f}(S) &\le \Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}\Big(2e^{22T^2\beta^2/m+1}+2/3\Big) && \text{(using Eq. (6), (7))} \\
&\le 4e\Big(1-\frac{\beta}{2}\Big)^{T}\Big(\frac{T\beta}{m}\Big)^{q}e^{22T^2\beta^2/m} && \text{(since } 22T^2\beta^2/m>0\text{)}
\end{aligned}$$

■

The theorem follows by putting together Claim 20 and Lemma 21:

$$\begin{aligned}
\sqrt{A}(x,x) &= \frac{1}{2^{k/2}}\sum_{Q\in\{0,1\}^k}\sqrt{\sum_{S\in\{0,1\}^m:M^tS=Q}\widehat{f}(S)} && \text{(using Claim 20)} \\
&\le \frac{1}{2^{k/2}}\sum_{Q\in\{0,1\}^k}\sum_{S\in\{0,1\}^m:M^tS=Q}\sqrt{\widehat{f}(S)} && \text{(using lower bound from Lemma 21)} \\
&= \frac{1}{2^{k/2}}\sum_{S\in\{0,1\}^m}\sqrt{\widehat{f}(S)} && (\cup_Q\{S: M^tS=Q\}=\{0,1\}^m \text{ since rank}(M)=k) \\
&= \frac{1}{2^{k/2}}\sum_{q=0}^{m}\sum_{S\in\{0,1\}^m:|S|=q}\sqrt{\widehat{f}(S)} \\
&\le \frac{2\sqrt{e}}{2^{k/2}}\Big(1-\frac{\beta}{2}\Big)^{T/2}e^{11T^2\beta^2/m}\sum_{q=0}^{m}\binom{m}{q}\Big(\frac{T\beta}{m}\Big)^{q/2} && \text{(using Lemma 21)} \\
&= \frac{2\sqrt{e}}{2^{k/2}}\Big(1-\frac{\beta}{2}\Big)^{T/2}e^{11T^2\beta^2/m}\Big(1+\sqrt{\frac{T\beta}{m}}\Big)^{m} && \text{(using binomial theorem)} \\
&\le \frac{2\sqrt{e}}{2^{k/2}}\Big(1-\frac{\beta}{2}\Big)^{T/2}e^{11T^2\beta^2/m+\sqrt{Tm\beta}}. && \text{(using } (1+x)^t\le e^{xt} \text{ for } x,t\ge 0\text{)}
\end{aligned}$$

■

4.1. Optimal lower bound for quantum PAC learning

We can now prove our tight lower bound on quantum sample complexity in the PAC model:

Theorem 23 *Let $\mathscr{C}$ be a concept class with VC-dim($\mathscr{C}$) $= d+1$, for sufficiently large d. Then for every $\delta\in(0,1/2)$ and $\varepsilon\in(0,1/20)$, every (ε,δ)-PAC quantum learner for $\mathscr{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon}+\frac{1}{\varepsilon}\log\frac{1}{\delta}\Big)$.*

Proof The d-independent part of the lower bound is Lemma 11. To prove the d-dependent part, define a distribution D on a set $\mathcal{S}=\{s_0,\ldots,s_d\}\subseteq\{0,1\}^n$ that is shattered by $\mathscr{C}$ as follows: $D(s_0)=1-20\varepsilon$ and $D(s_i)=20\varepsilon/d$ for all $i\in[d]$.

Now consider a $[d,k,r]_2$ linear code (for $k\ge d/4$, distance $r\ge d/8$) as shown to exist in Theorem 4 with the generator matrix $M\in\mathbb{F}_2^{d\times k}$ of rank k. Let $\{Mx: x\in\{0,1\}^k\}\subseteq\{0,1\}^d$ be the set of codewords in this linear code; these satisfy $d_H(Mx,My)\ge d/8$ whenever $x\ne y$. For each $x\in\{0,1\}^k$, let c^x be a concept defined on the shattered set as: $c^x(s_0)=0$ and

$c^x(s_i) = (Mx)_i$ for all $i \in [d]$. The existence of such concepts in $\mathscr{C}$ follows from the fact that $\mathcal{S}$ is shattered by $\mathscr{C}$. From the distance property of the code, we have $\Pr_{s\sim D}[c^x(s) \neq c^y(s)] \geq \frac{20\varepsilon}{d}\frac{d}{8} = 5\varepsilon/2$. This in particular implies that an (ε,δ)-PAC quantum learner that tries to ε-approximate a concept from $\{c^x : x \in \{0,1\}^k\}$ should successfully *identify* that concept with probability at least $1-\delta$.

We now consider the following state-identification problem: for $x \in \{0,1\}^k$, denote $|\psi_x\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, c^x(s_i)\rangle$. Let the (ε,δ)-PAC quantum sample complexity be T. Assume $T \leq d/(20e^3\varepsilon)$, since otherwise $T \geq \Omega(d/\varepsilon)$ and the theorem follows. Suppose the learner has knowledge of the ensemble $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$, and is given $|\psi_x\rangle^{\otimes T} \in \mathcal{E}$ for a uniformly random x. The learner would like to maximize the average probability of success to identify the given state. For this problem, we prove a lower bound on T using the PGM defined in Section 2.6. In particular, we show that using the PGM, if a learner successfully identifies the states in $\mathcal{E}$, then $T = \Omega(d/\varepsilon)$. Since the PGM is the optimal measurement[6] that the learner could have performed, the result follows. The following lemma makes this lower bound rigorous and will conclude the proof of the theorem.

Lemma 24 *For every $x \in \{0,1\}^k$, let $|\psi_x\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, c^x(s_i)\rangle$, and $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$. Then*

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{2^{d/4+T\varepsilon}} e^{8800T^2\varepsilon^2/d+4\sqrt{5Td\varepsilon}}.$$

Before we prove the lemma, we first show why it implies the theorem. Since we observed above that $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$, a good learner satisfies $P^{PGM}(\mathcal{E}) = \Omega(1)$ (say for $\delta = 1/4$), which in turn implies (by taking logarithms) that $\Omega(d + T\varepsilon) \leq O(T^2\varepsilon^2/d + \sqrt{Td\varepsilon})$. Hence, it follows that

$$\Omega(\max\{d, T\varepsilon\}) \leq O(\min\{T^2\varepsilon^2/d, \sqrt{Td\varepsilon}\}).$$

Note that if $T\varepsilon$ maximizes the left-hand side, then $d \leq T\varepsilon$ and hence $T \geq \Omega(d/\varepsilon)$. The remaining cases are $\Omega(d) \leq T^2\varepsilon^2/d$ and $\Omega(d) \leq \sqrt{Td\varepsilon}$. Both these statements give us $T \geq \Omega(d/\varepsilon)$.[7] Hence the theorem follows, and it remains to prove Lemma 24:

Proof Let $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ and G be the $2^k \times 2^k$ Gram matrix for $\mathcal{E}'$. As we saw in Section 2.6, the success probability of identifying the states in the ensemble $\mathcal{E}$ using the PGM is

$$P^{PGM}(\mathcal{E}) = \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2.$$

6. For $x \in \{0,1\}^k$, define unitary $U_{c^x} : |s_i, b\rangle \to |s_i, b + c^x(s_i)\rangle$ for all $i \in \{0,\ldots,d\}$. The ensemble $\mathcal{E}$ is generated by applying $\{U_{c^x}\}_{x\in\{0,1\}^k}$ to $|\varphi\rangle = \sum_{i\in\{0,\ldots,d\}} \sqrt{D(s_i)}|s_i, 0\rangle$. View $c^x = (0, Mx) \in \{0,1\}^{d+1}$ as a concatenated string where Mx is a codeword of the $[d,k,r]_2$ code. Since the 2^k codewords of the $[d,k,r]_2$ code form a linear subspace, $\{U_{c^x}\}_{x\in\{0,1\}^k}$ is an Abelian group. From the discussion in Section 2.6, we conclude that the PGM is the optimal measurement for this state-identification problem.

7. We made no attempt to optimize the constants here. Also, we remark that this tight lower bound on sample complexity implies that Lemma 21 is tight up to constant factors in the exponent.

For all $x, y \in \{0,1\}^k$, the entries of the Gram matrix G can be written as:

$$\begin{aligned} G(x,y) = \frac{1}{2^k}\langle\psi_x|\psi_y\rangle^T &= \frac{1}{2^k}\Big((1-20\varepsilon) + \frac{20\varepsilon}{d}\sum_{i=1}^{d}\langle c^x(s_i)|c^y(s_i)\rangle\Big)^T \\ &= \frac{1}{2^k}\Big((1-20\varepsilon) + \frac{20\varepsilon}{d}(d - d_H(Mx, My))\Big)^T \\ &= \frac{1}{2^k}\Big(1 - \frac{20\varepsilon}{d}d_H(Mx, My)\Big)^T, \end{aligned}$$

where Mx, $My \in \{0,1\}^d$ are codewords in the linear code defined earlier. Define $f : \{0,1\}^d \to \mathbb{R}$ as $f(z) = (1 - \frac{20\varepsilon}{d}|z|)^T$, and let $A(x,y) = (f \circ M)(x+y)$ for $x, y \in \{0,1\}^k$. Note that $G = A/2^k$. Since we assumed $T \leq d/(20e^3\varepsilon)$, we can use Theorem 18 (by choosing $m = d$ and $\beta = 20\varepsilon$) to upper bound the success probability of successfully identifying the states in the ensemble $\mathcal{E}$ using the PGM.

$$\begin{aligned} P^{PGM}(\mathcal{E}) &= \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2 \\ &= \frac{1}{2^k}\sum_{x\in\{0,1\}^k} \sqrt{A}(x,x)^2 && \text{(since } G = A/2^k\text{)} \\ &\leq \frac{4e}{2^k}\Big(1 - \frac{\beta}{2}\Big)^T e^{22T^2\beta^2/d+2\sqrt{Td\beta}} && \text{(using Theorem 18)} \\ &= \frac{4e}{2^k}\Big(1 - 10\varepsilon\Big)^T e^{8800T^2\varepsilon^2/d+4\sqrt{5Td\varepsilon}} && \text{(substituting } \beta = 20\varepsilon\text{)} \\ &\leq \frac{4e}{2^{k+T\varepsilon}} e^{8800T^2\varepsilon^2/d+4\sqrt{5Td\varepsilon}} && \text{(using } (1-10\varepsilon)^T \leq e^{-10\varepsilon T} \leq 2^{-\varepsilon T}\text{)} \end{aligned}$$

The lemma follows by observing that $k \geq d/4$. ■

■

4.2. Optimal lower bound for quantum agnostic learning

We now use the same approach to obtain a tight lower bound on quantum sample complexity in the *agnostic* setting.

Theorem 25 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d$, for sufficiently large d. Then for every $\delta \in (0, 1/2)$ and $\varepsilon \in (0, 1/10)$, every (ε,δ)-agnostic quantum learner for $\mathscr{C}$ has sample complexity $\Omega\Big(\frac{d}{\varepsilon^2} + \frac{1}{\varepsilon^2}\log\frac{1}{\delta}\Big)$.*

Proof The d-independent part of the lower bound is Lemma 12. For the d-dependent term in the lower bound, consider a $[d,k,r]_2$ linear code (for $k \geq d/4$, distance $r \geq d/8$) as shown to exist in Theorem 4, with generator matrix $M \in \mathbb{F}_2^{d\times k}$ of rank k. Let $\{Mx : x \in \{0,1\}^k\} \subseteq$

$\{0,1\}^d$ be the set of 2^k codewords in this linear code; these satisfy $d_H(Mx,My) \geq d/8$ whenever $x \neq y$. To each codeword $x \in \{0,1\}^k$ we associate a distribution D_x as follows:

$$D_x(s_i,b) = \frac{1}{d}\Big(\frac{1}{2} + \frac{1}{2}(-1)^{(Mx)_i+b}\alpha\Big), \qquad \text{for } (i,b) \in [d]\times\{0,1\},$$

where $\mathcal{S} = \{s_1,\ldots,s_d\}$ is a set that is shattered by $\mathscr{C}$, and α is a parameter which we shall pick later. Let $c^x \in \mathscr{C}$ be a concept that labels $\mathcal{S}$ according to $Mx \in \{0,1\}^d$. The existence of such $c^x \in \mathscr{C}$ follows from the fact that $\mathcal{S}$ is shattered by $\mathscr{C}$. Note that c^x is the minimal-error concept in $\mathscr{C}$ w.r.t. D_x. A learner that labels $\mathcal{S}$ according to some string $\ell \in \{0,1\}^d$ has additional error $d_H(Mx,\ell)\cdot\alpha/d$ compared to c^x. This in particular implies that an (ε,δ)-agnostic quantum learner has to find (with probability at least $1-\delta$) an ℓ such that $d_H(Mx,\ell) \leq d\varepsilon/\alpha$. We pick $\alpha = 20\varepsilon$ and we get $d_H(Mx,\ell) \leq d/20$. However, since Mx was a codeword of a $[d,k,r]_2$ code with distance $r \geq d/8$, finding an ℓ satisfying $d_H(Mx,\ell) \leq d/20$ is equivalent to *identifying* Mx, and hence x.

Now consider the following state-identification problem: for $x \in \{0,1\}^k$, let $|\psi_x\rangle = \sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(s_i,b)}|s_i,b\rangle$. Let the (ε,δ)-agnostic quantum sample complexity be T. Assume $T \leq d/(100e^3\varepsilon^2)$, since otherwise $T \geq \Omega(d/\varepsilon^2)$ and the theorem follows. Suppose the learner has knowledge of the ensemble $\mathcal{E} = \{(2^{-k},|\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$, and is given $|\psi_x\rangle^{\otimes T} \in \mathcal{E}$ for uniformly random x. The learner would like to maximize the average probability of success to identify the given state. For this problem, we prove a lower bound on T using the PGM defined in Section 2.6. In particular, we show that using the PGM, if a learner successfully identifies the states in $\mathcal{E}$, then $T = \Omega(d/\varepsilon^2)$. Since the PGM is the optimal measurement[8] that the learner could have performed, the result follows. The following lemma makes this lower bound rigorous and will conclude the proof of the theorem.

Lemma 26 *For $x \in \{0,1\}^k$, let $|\psi_x\rangle = \sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(s_i,b)}|s_i,b\rangle$, and consider the ensemble $\mathcal{E} = \{(2^{-k},|\psi_x\rangle^{\otimes T}) : x \in \{0,1\}^k\}$. Then*

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{e^{(d\ln 2)/4+25T\varepsilon^2}} e^{220000T^2\varepsilon^4/d+20\sqrt{Td\varepsilon^2}}.$$

Before we prove the lemma, we first show why it implies the theorem. Since we observed above that $P^{opt}(\mathcal{E}) = P^{PGM}(\mathcal{E})$, a good learner satisfies $P^{PGM}(\mathcal{E}) = \Omega(1)$ (say for $\delta = 1/4$), which in turn implies

$$\Omega(\max\{d, T\varepsilon^2\}) \leq O(\min\{T^2\varepsilon^4/d, \sqrt{Td\varepsilon^2}\}).$$

Like in the proof of Theorem 23, this implies a lower bound of $T = \Omega(d/\varepsilon^2)$ and proves the theorem. It remains to prove Lemma 26:

8. For $x \in \{0,1\}^k$, define unitary $U_{c^x} = \sum_{i\in[d]} |s_i\rangle\langle s_i| \otimes X^{(Mx)_i}$, where X is the NOT-gate, so $X^{(Mx)_i}|b\rangle = |b+(Mx)_i\rangle$ for $b \in \{0,1\}$. The ensemble $\mathcal{E}$ is generated by applying $\{U_{c^x}\}_{x\in\{0,1\}^k}$ to $|\varphi\rangle = \frac{1}{\sqrt{d}}\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{\frac{1}{2}+\frac{1}{2}(-1)^b\alpha}|s_i,b\rangle$. Since the 2^k codewords of the $[d,k,r]_2$ code form a linear subspace, $\{U_{c^x}\}_{x\in\{0,1\}^k}$ is an Abelian group. From the discussion in Section 2.6, we conclude that the PGM is the optimal measurement for this state-identification problem.

Proof Let $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ and G be the $2^k \times 2^k$ Gram matrix for the set $\mathcal{E}'$. As we saw in Section 2.6, the success probability of identifying the states in the ensemble $\mathcal{E}$ using the PGM is

$$P^{PGM}(\mathcal{E}) = \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2.$$

For all $x,y \in \{0,1\}^k$, the entries of G can be written as:

$$\begin{aligned}
2^k\cdot G(x,y) &= \langle\psi_x|\psi_y\rangle^T \\
&= \Big(\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{D_x(i,b)D_y(i,b)}\Big)^T \\
&= \Big(\frac{1}{2d}\sum_{(i,b)\in[d]\times\{0,1\}} \sqrt{(1+10\varepsilon(-1)^{(Mx)_i+b})(1+10\varepsilon(-1)^{(My)_i+b})}\Big)^T \\
&= \Big(\frac{1}{2d}\sum_{\substack{(i,b):\\(Mx)_i=(My)_i}} (1+10\varepsilon(-1)^{(Mx)_i+b}) + \frac{1}{2d}\sum_{\substack{(i,b):\\(Mx)_i\neq(My)_i}} \sqrt{1-100\varepsilon^2}\Big)^T \\
&= \Big(\frac{d-d_H(Mx,My)}{d} + \frac{\sqrt{1-100\varepsilon^2}}{d} d_H(Mx,My)\Big)^T \\
&= \Big(1 - \frac{1-\sqrt{1-100\varepsilon^2}}{d} d_H(Mx,My)\Big)^T.
\end{aligned}$$

where we used $\alpha = 20\varepsilon$ in the third equality.

Let $\beta = 1-\sqrt{1-100\varepsilon^2}$, which is at most 1 for $\varepsilon \leq 1/10$. Define $f : \{0,1\}^d \to \mathbb{R}$ as $f(z) = (1-\frac{\beta}{d}|z|)^T$, and let $A(x,y) = (f\circ M)(x+y)$ for $x,y \in \{0,1\}^k$. Then $G = A/2^k$. Note that $T \leq d/(100e^3\varepsilon^2) \leq d/(e^3\beta)$ (the first inequality is by assumption and the second inequality follows for $\varepsilon \leq 1/10$ and $\beta \leq 1$). Since we assumed $T \leq d/(100e^3\varepsilon^2)$, we can use Theorem 18 (by choosing $m = d$ and $\beta = 1-\sqrt{1-100\varepsilon^2}$) to upper bound the success probability of identifying the states in the ensemble $\mathcal{E}$:

$$\begin{aligned}
P^{PGM}(\mathcal{E}) &= \sum_{x\in\{0,1\}^k} \sqrt{G}(x,x)^2 \\
&= \frac{1}{2^k}\sum_{x\in\{0,1\}^k} \sqrt{A}(x,x)^2 && \text{(since } G = A/2^k\text{)} \\
&\leq \frac{4e}{2^k}\Big(1-\frac{\beta}{2}\Big)^T e^{22T^2\beta^2/d+2\sqrt{Td\beta}} && \text{(using Theorem 18)} \\
&\leq \frac{4e}{2^k}\Big(1-\frac{\beta}{2}\Big)^T e^{220000T^2\varepsilon^4/d+20\sqrt{Td\varepsilon^2}} && \text{(using } \beta = 1-\sqrt{1-100\varepsilon^2} \leq 100\varepsilon^2\text{)} \\
&\leq \frac{4e}{2^k}\Big(1-25\varepsilon^2\Big)^T e^{220000T^2\varepsilon^4/d+20\sqrt{Td\varepsilon^2}} && \text{(using } \sqrt{1-100\varepsilon^2} \leq 1-50\varepsilon^2\text{)} \\
&\leq \frac{4e}{e^{k\ln 2+25T\varepsilon^2}} e^{220000T^2\varepsilon^4/d+20\sqrt{Td\varepsilon^2}}. && \text{(using } (1-x)^t \leq e^{-xt} \text{ for } x,t \geq 0\text{)}
\end{aligned}$$

The lemma follows by observing that $k \geq d/4$. ■

■

4.3. Additional results

In this section we mention two additional results that can also be obtained using Theorem 18.

4.3.1. Quantum PAC sample complexity under random classification noise

In the theorem below, we show a lower bound on the quantum PAC sample complexity under the random classification noise model with noise rate η. Recall that in this model, for every $c \in \mathscr{C}$ and distribution D, $\varepsilon, \delta > 0$, given access to copies of the η-noisy state,

$$\sum_{x \in \{0,1\}^n} \sqrt{(1-\eta)D(x)}|x, c(x)\rangle + \sqrt{\eta D(x)}|x, 1-c(x)\rangle,$$

a (ε, δ)-PAC quantum learner is required to output an hypothesis h such that $\mathrm{err}_D(c,h) \leq \varepsilon$ with probability at least $1-\delta$.

Theorem 27 *Let $\mathscr{C}$ be a concept class with VC-dim$(\mathscr{C}) = d+1$, for sufficiently large d. Then for every $\delta \in (0, 1/2)$, $\varepsilon \in (0, 1/20)$ and $\eta \in (0, 1/2)$, every (ε, δ)-PAC quantum learner for $\mathscr{C}$ in the PAC setting with random classification noise rate η, has sample complexity $\Omega\Big(\frac{d}{(1-2\eta)^2\varepsilon} + \frac{\log(1/\delta)}{(1-2\eta)^2\varepsilon}\Big)$.*

One can use exactly the same proof technique as in Lemma 11 and Theorem 23 to prove this, with only the additional inequality $1 - 2\sqrt{\eta(1-\eta)} \leq (1-2\eta)^2$, which holds for $\eta \leq 1/2$. We omit the details of the calculation.

4.3.2. Distinguishing codeword states

Ashley Montanaro (personal communication) alerted us to the following interesting special case of our PGM-based result.

Consider an $[n,k,d]_2$ linear code $\{Mx : x \in \{0,1\}^k\}$, where $M \in \mathbb{F}_2^{n \times k}$ is the rank-k generator matrix of the code, $k = \Omega(n)$, and distinct codewords have Hamming distance at least d.[9] For every $x \in \{0,1\}^k$, define a *codeword state* $|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i, (Mx)_i\rangle$. These states form an example of a *quantum fingerprinting* scheme considered earlier by Buhrman et al. (2001): 2^k states whose pairwise inner products are bounded away from 1. How many copies do we need to identify one such fingerprint?

Let $\mathcal{E} = \{(2^{-k}, |\psi_x\rangle) : x \in \{0,1\}^k\}$ be an ensemble of codeword states. Consider the following task: given T copies of an unknown state drawn uniformly from $\mathcal{E}$, we are required to identify the state with probability $\geq 4/5$. From Holevo's theorem one can easily obtain a lower bound of $T = \Omega(k/\log n)$ copies, since the learner should obtain $\Omega(k)$ bits of information (i.e., identify k-bit string x with probability $\geq 4/5$), while each copy of the codeword state gives at most $\log n$ bits of information. In the theorem below, we improve that $\Omega(k/\log n)$ to the optimal $\Omega(k)$ for constant-rate codes.

Theorem 28 *Let $\mathcal{E} = \{|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_{i \in [n]} |i, (Mx)_i\rangle : x \in \{0,1\}^k\}$, where $M \in \mathbb{F}_2^{n \times k}$ is the generator matrix of an $[n,k,d]_2$ linear code with $k = \Omega(n)$. Then $\Omega(k)$ copies of an unknown state from $\mathcal{E}$ (drawn uniformly at random) are necessary to be able to identify that state with probability at least $4/5$.*

9. Note that throughout this paper $\mathscr{C}$ was a concept class in $\{0,1\}^n$ and d was the VC dimension of $\mathscr{C}$. The use of n, d in this section has been changed to conform to the convention in coding theory.

One can use exactly the proof technique of Theorem 23 to prove the theorem. Suppose we are given T copies of the unknown codeword state. Assume $T \leq n$, since otherwise $T \geq n \geq \sqrt{kn}$ and the theorem follows. Observe that the Gram matrix G for $\mathcal{E}' = \{2^{-k/2}|\psi_x\rangle^{\otimes T} : x \in \{0,1\}^k\}$ can be written as $G(x,y) = \frac{1}{2^k}\Big(1 - \frac{|M(x+y)|}{n}\Big)^T$ for $x, y \in \{0,1\}^k$. Using Theorem 18 (choosing $\beta = 1$ and $m = n$) to upper bound the success probability of successfully identifying the states in the ensemble $\mathcal{E}$ using the PGM, we obtain

$$P^{PGM}(\mathcal{E}) \leq \frac{4e}{2^{k+T}} e^{22T^2/n + 2\sqrt{Tn}}.$$

As in the proof of Theorem 23, this implies the lower bound of Theorem 28. We omit the details of the calculation.

5. Conclusion

The main result of this paper is that quantum examples give no significant improvement over the usual random examples in passive, distribution-independent settings. Of course, these negative results do not mean that quantum machine learning is useless. In our introduction we already mentioned improvements from quantum examples for learning under the uniform distribution; improvements from using quantum membership queries; and improvements in time complexity based on quantum algorithms like Grover's and HHL. Quantum machine learning is still in its infancy, and we hope for many more positive results.

We end by identifying a number of open questions for future work:

- We gave lower bounds on sample complexity for the rather benign random classification noise. What about other noise models, such as *malicious* noise?
- What is the quantum sample complexity for learning concepts whose range is $[k]$ rather than $\{0,1\}$, for some $k > 2$? Even the *classical* sample complexity is not fully determined yet (Shalev-Shwartz and Ben-David, 2014, Section 29.2).
- In the introduction we mentioned a few examples of learning under the *uniform* distribution where quantum examples are significantly more powerful than classical examples. Can we find more such examples of quantum improvements in sample complexity in fixed-distribution settings?
- Can we find more examples of quantum speed-up in *time* complexity of learning?

Acknowledgments

We thank Shalev Ben-David, Dmitry Gavinsky, Robin Kothari, Nishant Mehta, Ashley Montanaro, Henry Yuen for helpful comments and pointers to the literature. We thank Ashley Montanaro for suggesting the additional remark in Section 4.3.2. We also thank the anonymous referees from QIP, CCC, and JMLR for their helpful comments.

References

S. Aaronson. The learnability of quantum states. *Proceedings of the Royal Society of London*, 463(2088), 2007. quant-ph/0608142.

S. Aaronson. Quantum machine learning algorithms: Read the fine print. *Nature Physics*, 11(4):291–293, 2015.

E. Aïmeur, G. Brassard, and S. Gambs. Machine learning in a quantum world. In *Proceedings of Advances in Artificial Intelligence, 19th Conference of the Canadian Society for Computational Studies of Intelligence*, volume 4013, pages 431–442, 2006.

E. Aïmeur, G. Brassard, and S. Gambs. Quantum speed-up for unsupervised learning. *Machine Learning*, 90(2):261–287, 2013.

A. Ambainis and A. Montanaro. Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Information & Computation*, 14(5-6):439–453, 2014. arXiv:1210.1148.

D. Angluin and P. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1988.

M. Anthony and P. L. Bartlett. *Neural network learning: Theoretical foundations*. Cambridge University Press, 2009.

B. Apolloni and C. Gentile. Sample size lower bounds in PAC learning by algorithmic complexity theory. *Theoretical Computer Science*, 209:141–162, 1998.

S. Arunachalam and R. de Wolf. Guest column: A survey of quantum learning theory. *SIGACT News*, 48(2):41–67, 2017. arxiv:1606.08920.

A. Atıcı and R. Servedio. Improved bounds on quantum learning algorithms. *Quantum Information Processing*, 4(5):355–386, 2005. quant-ph/0411140.

A. Atıcı and R. Servedio. Quantum algorithms for learning and testing juntas. *Quantum Information Processing*, 6(5):323–348, 2009. arXiv:0707.3479.

J. Audibert. Fast learning rates in statistical inference through aggregation, 2008. Research Report 06-20, CertisEcole des Ponts. math/0703854.

J. Audibert. Fast learning rates in statistical inference through aggregation. *The Annals of Statistics*, 37(4):1591–1646, 2009. arXiv:0909.1468v1.

D. Bacon, A. Childs, and W. van Dam. Optimal measurements for the dihedral hidden subgroup problem. *Chicago Journal of Theoretical Computer Science*, 2006. Earlier version in FOCS'05. quant-ph/0504083.

H. Barnum and E. Knill. Reversing quantum dynamics with near-optimal quantum and classical fidelity. *Journal of Mathematical Physics*, 43:2097–2106, 2002. quant-ph/0004088.

E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

N. H. Bshouty and J. C. Jackson. Learning DNF over the uniform distribution using a quantum example oracle. *SIAM Journal on Computing*, 28(3):1136–1153, 1999. Earlier version in COLT'95.

H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf. Quantum fingerprinting. *Physical Review Letters*, 87(16), 2001. quant-ph/0102001.

A. Daniely and S. Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Proceedings of the 29th Conference on Learning Theory (COLT'16)*, 2016.

Y. C. Eldar and G. D. Forney Jr. On quantum detection and the square-root measurement. *IEEE Transactions and Information Theory*, 47(3):858–872, 2001. quant-ph/0005132.

Y. C. Eldar, A. Megretski, and G. C. Verghese. Designing optimal quantum detectors via semidefinite programming. *IEEE Transactions Information Theory*, 49(4):1007–1012, 2003. quant-ph/0205178.

D. Gavinsky. Quantum predictive learning and communication complexity with single input. *Quantum Information and Computation*, 12(7-8):575–588, 2012. Earlier version in COLT'10. arXiv:0812.3429.

C. Gentile and D. P. Helmbold. Improved lower bounds for learning from noisy examples: An information-theoretic approach. *Information and Computation*, 166:133–155, 2001.

L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

S. Hanneke. The optimal sample complexity of PAC learning. *Journal of Machine Learning Research*, 17(38):1–15, 2016. arXiv:1507.00473.

A. Harrow, A. Hassidim, and S. Lloyd. Quantum algorithm for solving linear systems of equations. *Physical Review Letters*, 103(15):150502, 2009. arXiv:0811.3171.

D. Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992.

M. Hunziker, D. A. Meyer, J. Park, J. Pommersheim, and M. Rothstein. The geometry of quantum learning. *Quantum Information Processing*, 9(3):321–341, 2010. quant-ph/0309059.

J. C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997. Earlier version in FOCS'94.

J. C. Jackson, C. Tamon, and T. Yamakami. Quantum DNF learnability revisited. In *Proceedings of 8th COCOON*, pages 595–604, 2002. quant-ph/0202066.

R. Jain and S. Zhang. New bounds on classical and quantum one-way communication complexity. *Theoretical Computer Science*, 410(26):2463–2477, 2009. arXiv:0802.4101.

P. Kaye, R. Laflamme, and M. Mosca. *An Introduction to Quantum Computing*. Oxford University Press, 2006.

M. J. Kearns and L. G. Valiant. Cryptographic limitations on learning Boolean formulae and finite automata. *Journal of the ACM*, 41(1):67–95, 1994.

M. J. Kearns and U. V. Vazirani. *An introduction to computational learning theory*. MIT Press, 1994.

M. J. Kearns, R. E. Schapire, and L. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994. Earlier version in COLT'92.

A. Kontorovich and I. Pinelis. Exact lower bounds for the agnostic probably-approximately-correct (PAC) machine learning model, 2016. Preprint at arxiv:1606.08920.

R. Kothari. An optimal quantum algorithm for the oracle identification problem. In *31st International Symposium on Theoretical Aspects of Computer Science (STACS 2014)*, pages 482–493, 2014. arXiv:1311.7685.

A. Montanaro. On the distinguishability of random quantum states. *Communications in Mathematical Physics*, 273(3):619–636, 2007. quant-ph/0607011.

A. Montanaro. The quantum query complexity of learning multilinear polynomials. *Information Processing Letters*, 112(11):438–442, 2012. arXiv:1105.3310.

R. O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014.

R. Servedio and S. Gortler. Equivalences and separations between quantum and classical learnability. *SIAM Journal on Computing*, 33(5):1067–1092, 2004. Combines earlier papers from ICALP'01 and CCC'01. quant-ph/0007036.

S. Shalev-Shwartz and S. Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge University Press, 2014.

H. U. Simon. General bounds on the number of examples needed for learning probabilistic concepts. *Journal of Computer and System Sciences*, 52(2):239–254, 1996. Earlier version in COLT'93.

H. U. Simon. An almost optimal PAC algorithm. In *Proceedings of the 28th Conference on Learning Theory (COLT)*, pages 1552–1563, 2015.

M. Talagrand. Sharper bounds for Gaussian and empirical processes. *The Annals of Probability*, pages 28–76, 1994.

L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

V. Vapnik and A. Chervonenkis. On the uniform convergence of relative frequencies of events to their probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.

V. Vapnik and A. Chervonenkis. Theory of pattern recognition. 1974. In Russian.

K. A. Verbeurgt. Learning DNF under the uniform distribution in quasi-polynomial time. In *Proceedings of the 3rd Annual Workshop on Computational Learning Theory (COLT'90)*, pages 314–326, 1990.

N. Wiebe, A. Kapoor, and K. M. Svore. Quantum deep learning. *Quantum Information & Computation*, 16(7&8):541–587, 2016a. arXiv:1412.3489.

N. Wiebe, A. Kapoor, and K. M. Svore. Quantum perceptron models. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems*, pages 3999–4007, 2016b. arXiv:1602.04799.

C. Zhang. An improved lower bound on query complexity for quantum PAC learning. *Information Processing Letters*, 111(1):40–45, 2010.

Journal of Machine Learning Research 19 (2018) 1-5 Submitted 4/18; Published 10/18

Scikit-Multiflow: A Multi-output Streaming Framework

Jacob Montiel JACOB.MONTIEL@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay
Paris, FRANCE

Jesse Read JESSE.READ@POLYTECHNIQUE.EDU
LIX, École Polytechnique
Palaiseau, FRANCE

Albert Bifet ALBERT.BIFET@TELECOM-PARISTECH.FR
LTCI, Télécom ParisTech, Université Paris-Saclay
Paris, FRANCE

Talel Abdessalem TALEL.ABDESSALEM@ENST.FR
LTCI, Télécom ParisTech, Université Paris-Saclay
Paris, FRANCE
UMI CNRS IPAL, National University of Singapore

Editor: Balazs Kegl

Abstract

scikit-multiflow is a framework for learning from data streams and multi-output learning in Python. Conceived to serve as a platform to encourage the democratization of stream learning research, it provides multiple state-of-the-art learning methods, data generators and evaluators for different stream learning problems, including single-output, multi-output and multi-label. scikit-multiflow builds upon popular open source frameworks including scikit-learn, MOA and MEKA. Development follows the FOSS principles. Quality is enforced by complying with PEP8 guidelines, using continuous integration and functional testing. The source code is available at `https://github.com/scikit-multiflow/scikit-multiflow`.

Keywords: Machine Learning, Stream Data, Multi-output, Drift Detection, Python

1. Introduction

Recent years have witnessed the proliferation of Free and Open Source Software (FOSS) in the research community. Specifically, in the field of Machine Learning, researchers have benefited from the availability of different frameworks that provide tools for faster development, allow replicability and reproducibility of results and foster collaboration. Following the FOSS principles, we introduce scikit-multiflow, a Python framework to implement algorithms and perform experiments in the field of Machine Learning on Evolving Data Streams. scikit-multiflow is inspired in the popular frameworks scikit-learn, MOA and MEKA.

scikit-learn (Pedregosa et al., 2011) is the most popular open source software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forest, gradient boosting, k-means and DBSCAN, and is designed to inter-operate with the Python numerical and scientific packages NumPy and SciPy.

MOA (Bifet et al., 2010) is the most popular open source framework for data stream mining, with a very active growing community. It includes a collection of machine learning algorithms (classification, regression, clustering, outlier detection, concept drift detection and recommender systems) and tools for evaluation. Related to the WEKA project (Hall et al., 2009), MOA is also written in Java, while scaling to more demanding problems.

The MEKA project (Read et al., 2016) provides an open source implementation of methods for multi-label learning and evaluation. In multi-label classification, the aim is to predict multiple output variables for each input instance. This different from the 'standard' case (binary, or multi-class classification) which involves only a single target variable.

As a multi-output streaming framework, scikit-multiflow serves as a bridge between research communities that have flourished around the aforementioned popular frameworks, providing a common ground where they can thrive. scikit-multiflow assists on the democratization of Stream Learning by bringing this research field closer to the Machine Learning community, given the increasing popularity of the Python programing language. The objective is two-folded: First, fills the void in Python for a stream learning framework which can also interact with available tools such as scikit-learn and extends the set of available

Table 1: Available methods in scikit-multiflow. Methodologies on the left, and frameworks on the right of the vertical bar.

						Java		Python		
Algorithm	**Classification**	**Regression**	**Single-Output**	**Multi-Output**	**Drift Detection**	**MOA**	**MEKA**†	**scikit-learn**†	**scikit-multiflow**	**Reference**
kNN	✓		✓			✓	✓	✓	✓	Bishop (2006)
kNN + ADWIN	✓		✓			✓			✓	Bifet et al. (2018)
SAM kNN	✓		✓		✓	✓			✓	Losing et al. (2017)
Hoeffding Tree	✓		✓			✓			✓	Hulten et al. (2001)
Hoeffding Adaptive Tree	✓		✓		✓	✓			✓	Bifet et al. (2018)
FIMT-DD		✓	✓		✓	✓			✓	Bifet et al. (2018)
Adaptive Random Forest	✓		✓		✓	✓			✓	Gomes et al. (2017)
Oza Bagging	✓		✓		*	✓			✓	Oza (2005)
Leverage Bagging	✓		✓		✓	✓			✓	Bifet et al. (2018)
Multi-output Learner	✓	✓	✓	✓	*	✓	✓	✓	✓	Bishop (2006)
Classifier Chains	✓		✓	✓	*		✓	✓	✓	Read et al. (2016)
Regressor Chains		✓	✓	✓	*		✓	✓	✓	Read et al. (2016)
SGD	✓	✓	✓			✓	✓	✓	✓	Bishop (2006)
Naive Bayes	✓		✓			✓	✓	✓	✓	Bishop (2006)
MLP	✓	✓	✓				✓	✓	✓	Bishop (2006)
ADWIN					✓	✓			✓	Bifet et al. (2018)
DDM					✓	✓			✓	Gama et al. (2004)
EDDM					✓	✓			✓	Bifet et al. (2018)
Page Hinkley					✓	✓			✓	Page (1954)

* Depending on the base learner.
† We have only listed incremental methods for data-streams; MEKA and scikit-learn have many other batch-learning models available. MEKA in particular, has many problem-transformation methods which may be incremental depending on the base learner (it is able to use those from the MOA framework).

state-of-the-art methods on this platform. Second, provides a set of tools to facilitate the development of stream learning research, an example is (Montiel et al., 2018).

It is important to notice that scikit-multiflow complements scikit-learn, whose primary focus is batch learning, expanding the set of free and open source tools for Stream Learning. In addition, scikit-multiflow can be used within Jupyter Notebooks, a popular interface in the Data Science community. Special focus in the design of scikit-multiflow is to make it friendly to new users and familiar to experienced ones.

scikit-multiflow contains stream generators, learning methods, change detectors and evaluation methods. Stream generators include: Agrawal, Hyperplane, Led, Mixed, Random-RBF, Random-RBF with drift, Random Tree, SEA, SINE, SEA, STAGGER, Waveform, Multi-label, Regression and Concept-Drift. Available evaluators correspond to Prequential and Hold-Out evaluations, both supporting multiple performance metrics for *Classification* (Accuracy, Kappa Coefficient, Kappa T, Kappa M), *Multi-Output Classification* (Hamming Score, Hamming Loss, Exact Match, Jaccard Index) *Regression* (Mean Squared Error, Mean Absolute Error) and *Multi-Output Regression* (Average Mean Squared Error, Average Mean Absolute Error, Average Root Mean Squared Error). Learning methods and change detectors are listed in Table 1. This table also serves to outline the position of scikit-multiflow with respect to other open source frameworks.

2. Stream Data Mining Notation and Background

Consider a continuous stream of data $A = \{(\vec{x}_t, y_t)\}|t = 1, \ldots, T$ where $T \to \infty$. Input $\vec{x}_t$ is a feature vector and y_t the corresponding target where y is continuous in the case of regression and discrete for classification. The objective is to predict the target y for an unknown $\vec{x}$. In traditional single-output models, we deal with a single target variable for which one corresponding output is produced per test instance. Multi-output models can produce multiple outputs to assign to multiple target variables $\vec{y}$ for each test instance.

Different to batch learning, where all data (X, y) is available for training; in stream learning, training is performed incrementally as new data $(\vec{x}_i, y_i)$ is available. Performance P of a given model is measured according to some loss function that evaluates the difference between the set of expected labels Y and the predicted ones $\hat{Y}$. Hold-out evaluation is a popular performance evaluation method for batch and stream settings, where tests are performed in a separate test set. Prequential-evaluation (Dawid, 1984) or interleaved-test-then-train evaluation, is a popular performance evaluation method for the stream setting only, where tests are performed on new data before using it to train the model.

3. Architecture

The base class in scikit-multiflow is StreamModel which contains the following abstract methods to be implemented by its subclasses:

- fit — Trains a model in a batch fashion. Works as an interface to batch methods that implement a fit() function such as scikit-learn methods.
- partial_fit — Incrementally trains a stream model.
- predict — Predicts the target's value in supervised learning methods.
- predict_proba — Calculates per-class probabilities in classification problems.

A StreamModel object interacts with two other objects: a Stream object and (optionally) a StreamEvaluator object. The Stream object provides a continuous flow of data on request. The StreamEvaluator performs multiple tasks: queries the stream for data, trains and tests the model on the incoming data and continuously tracks the model's performance. The sequence to train a Stream Model and track its performance using prequential evaluation in scikit-multiflow is outlined in Figure 1.

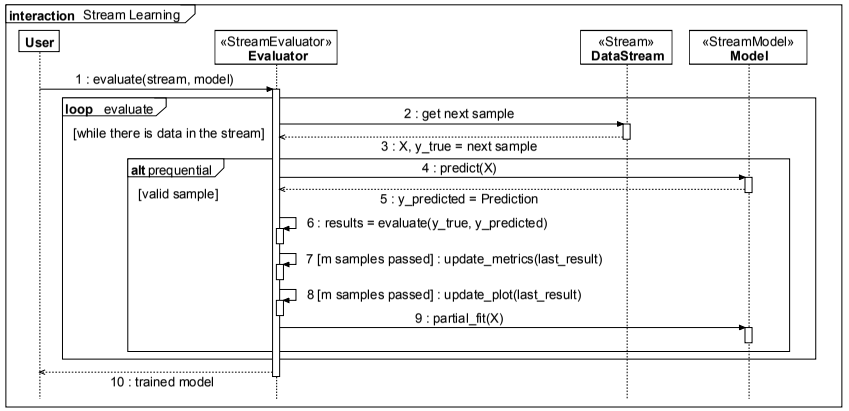


Figure 1: Training and testing a stream model using scikit-multiflow. This sequence corresponds to prequential evaluation.

4. Development

The scikit-multiflow package is distributed under the BSD License. Development follows the FOSS principles and encompasses:

- A webpage, `https://scikit-multiflow.github.io/`, including documentation and user guide. Both, documentation and user guide, are living documents that are continuously updated to reflect the current stage of scikit-multiflow.
- Version control via git. The source code of the package is publicly available on Github at `https://github.com/scikit-multiflow/scikit-multiflow`
- Package deployment and software quality are enforced via continuous integration and functional testing, `https://travis-ci.org/scikit-multiflow/scikit-multiflow`

References

Albert Bifet, Geoff Holmes, Richard Kirkby, and Bernhard Pfahringer. MOA: Massive online analysis. *Journal of Machine Learning Research*, 11(May):1601–1604, 2010.

Albert Bifet, Ricard Gavalda, Geoff Holmes, and Bernhard Pfahringer. *Machine Learning for Data Streams with Practical Examples in MOA*. MIT Press, 2018. `https://moa.cms.waikato.ac.nz/book/`.

Christopher M Bishop. Pattern recognition and machine learning. 2006.

A Philip Dawid. Present position and potential developments: Some personal views: Statistical theory: The prequential approach. *Journal of the Royal Statistical Society. Series A (General)*, pages 278–292, 1984.

João Gama, Pedro Medas, Gladys Castillo, and Pedro Rodrigues. Learning with Drift Detection. pages 286–295, 2004. ISSN 0302-9743. doi: 10.1007/978-3-540-28645-5_29.

Heitor M. Gomes, Albert Bifet, Jesse Read, Jean Paul Barddal, Fabrício Enembreck, Bernhard Pfharinger, Geoff Holmes, and Talel Abdessalem. Adaptive random forests for evolving data stream classification. *Machine Learning*, 106(9-10):1469–1495, 2017. ISSN 15730565. doi: 10.1007/s10994-017-5642-8.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11 (1):10–18, November 2009. ISSN 1931-0145. doi: 10.1145/1656274.1656278.

Geoff Hulten, Laurie Spencer, and Pedro Domingos. Mining time-changing data streams. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '01*, 18:97–106, 2001. ISSN 10844627. doi: 10.1145/502512.502529.

Viktor Losing, Barbara Hammer, and Heiko Wersing. KNN classifier with self adjusting memory for heterogeneous concept drift. *Proceedings - IEEE International Conference on Data Mining, ICDM*, 1:291–300, 2017. ISSN 15504786. doi: 10.1109/ICDM.2016.141.

Jacob Montiel, Albert Bifet, Viktor Losing, Jesse Read, and Talel Abdessalem. Learning fast and slow: A unified batch/stream framework. In *Big Data (Big Data), 2018 IEEE International Conference on*. IEEE, 2018.

N.C. Oza. Online Bagging and Boosting. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2340–2345. IEEE, 2005. ISBN 0-7803-9298-1. doi: 10.1109/ICSMC.2005.1571498.

Ewan S Page. Continuous inspection schemes. *Biometrika*, 41(1/2):100–115, 1954.

Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Jesse Read, Peter Reutemann, Bernhard Pfahringer, and Geoff Holmes. MEKA: A multi-label/multi-target extension to Weka. *Journal of Machine Learning Research*, 17(21):1–5, 2016.

Journal of Machine Learning Research 19 (2018) 1-22 Submitted 5/18; Revised 9/18; Published 10/18

Optimal Bounds for Johnson-Lindenstrauss Transformations

Michael Burr BURR2@CLEMSON.EDU
Shuhong Gao SGAO@CLEMSON.EDU
Department of Mathematical Sciences
Clemson University, Clemson SC 29634, USA

Fiona Knoll FKNOLL@G.CLEMSON.EDU
Department of Mathematical Sciences
University of Cincinnati, Cincinnati, OH 45221, USA

Editor: Kenji Fukumizu

Abstract

In 1984, Johnson and Lindenstrauss proved that any finite set of data in a high-dimensional space can be projected to a lower-dimensional space while preserving the pairwise Euclidean distances between points up to a bounded relative error. If the desired dimension of the image is too small, however, Kane, Meka, and Nelson (2011) and Jayram and Woodruff (2013) proved that such a projection does not exist. In this paper, we provide a precise asymptotic threshold for the dimension of the image, above which, there exists a projection preserving the Euclidean distance, but, below which, there does not exist such a projection.

Keywords: Johnson-Lindenstrauss transformation, Dimension reduction, Phase transition, Asymptotic threshold

1. Introduction

In 1984, Johnson and Lindenstrauss (1984), in establishing a bound on the Lipschitz constant for the Lipschitz extension problem, proved that any finite set of data in a high-dimensional space can be projected into a lower-dimensional space while preserving the pairwise Euclidean distance within any desired relative error. In particular, for any finite set of vectors $x_1, \ldots, x_N \in \mathbb{R}^d$ and for any error factor $0 < \epsilon < \frac{1}{2}$, there exists an absolute constant c such that for all $k \geq c\epsilon^{-2} \log N$, there exists a linear map $A : \mathbb{R}^d \to \mathbb{R}^k$ such that for all pairs $1 \leq i, j \leq N$,

$$(1-\epsilon)\|x_i - x_j\|_2 \leq \|Ax_i - Ax_j\|_2 \leq (1+\epsilon)\|x_i - x_j\|_2,$$

where $\|\cdot\|_2$ denotes the Euclidean norm. These inequalities are implied by the following theorem (by setting $\delta = \frac{1}{N^2}$ and using the union bound):

Theorem 1 (Johnson and Lindenstrauss (1984)) *For any real numbers $0 < \epsilon, \delta < \frac{1}{2}$, there exists an absolute constant $c > 0$ such that for any integer $k \geq c\epsilon^{-2} \log \frac{1}{\delta}$, there exists a probability distribution $\mathcal{D}$ on $k \times d$ real matrices such that for any fixed $x \in \mathbb{R}^d$,*

$$\text{Prob}_{A \sim \mathcal{D}} \left[(1-\epsilon)\|x\|_2^2 \leq \|Ax\|_2^2 \leq (1+\epsilon)\|x\|_2^2 \right] > 1 - \delta, \tag{1}$$

where $A \sim \mathcal{D}$ indicates that the matrix A is a random matrix with distribution $\mathcal{D}$.

Note that, in order to project a large number of vectors, δ must be sufficiently small. For instance, suppose we wish to project a set of $N = 2^{20}$ vectors to a smaller dimensional space. To apply the union bound to Inequality (1), we use any $\delta < 2^{-39}$. In this case, Inequality (1) implies that the probability of preserving all pairwise distances between N points (up to a relative error of ϵ) is at least $1 - \delta N^2/2 > 0$. Since the probability is nonzero, such a projection exists.

A probability distribution $\mathcal{D}$ satisfying Inequality (1) is called an (ϵ, δ)-JL distribution, or simply a JL distribution. Since these transformations are linear, without loss of generality, we assume for the rest of the paper that $\|x\|_2 = 1$. When a JL distribution is specified via an explicit construction, we may call a random projection $x \mapsto Ax$ generated in this way a JL transformation.

Since the introduction of JL distributions, there has been considerable work on explicit constructions of JL distributions, see, e.g., Johnson and Lindenstrauss (1984); Frankl and Maehara (1988); Indyk and Motwani (1998); Achlioptas (2003); Ailon and Chazelle (2006); Matousek (2008); Dasgupta et al. (2010); Kane and Nelson (2014); and the references therein. A simple and easily described JL distribution is that of Achlioptas (2003). In this construction, the entries of A are distributed as follows:

$$a_{ij} = \begin{cases} k^{-1/2}, & \text{with probability } 1/2, \\ -k^{-1/2}, & \text{with probability } 1/2. \end{cases}$$

The recent constructions in Ailon and Chazelle (2006); Matousek (2008); Dasgupta et al. (2010); Kane and Nelson (2014) have focused on the complexity of computing a projection for the purpose of applications. We note that the ability to project a vector to a smaller dimensional space, independent of the original dimension, while preserving the Euclidean norm up to a prescribed relative error, is highly desirable. In particular, dimension reduction has applications in many fields, including machine learning (Arriaga and Vempala (2006); Weinberger et al. (2009)), low rank approximation (Clarkson and Woodruff (2013); Nguyen et al. (2009); Ubaru et al. (2017)), approximate nearest neighbors (Ailon and Chazelle (2006); Indyk and Motwani (1998)), data storage (Candès (2008); Cormode and Indyk (2016)), and document similarity (Bingham and Mannila (2001); Lin and Gunopulos (2003)).

For both practical and theoretical purposes, it is important to know the smallest possible dimension k of a potential image space for any given ϵ and δ. Note that, for any $d_1 < d$, each (ϵ, δ)-JL distribution $\mathcal{D}$ on $\mathbb{R}^{k \times d}$ induces an (ϵ, δ)-JL distribution $\mathcal{D}_1$ on $\mathbb{R}^{k \times d_1}$ in a natural way: the matrices of $\mathcal{D}_1$ are obtained from $\mathcal{D}$ by deleting the last $d - d_1$ columns of a random matrix, together with the induced probability distribution. This construction is a JL distribution since $\mathbb{R}^{d_1}$ can be naturally embedded into $\mathbb{R}^d$ by extending a vector in $\mathbb{R}^{d_1}$ by $d - d_1$ zeros. Hence, if there exists an (ϵ, δ)-JL distribution on $\mathbb{R}^{k \times d}$, then there is an (ϵ, δ)-JL distribution on $\mathbb{R}^{k \times d_1}$ for all $1 \leq d_1 \leq d$. Similarly, if an (ϵ, δ)-JL distribution does not exist on $\mathbb{R}^{k \times d}$, then, for any $k_1 < k$, there cannot be an (ϵ, δ)-JL distribution on $\mathbb{R}^{k_1 \times d}$. In particular, since $\mathbb{R}^{k_1}$ can be naturally embedded into $\mathbb{R}^k$ by extending a vector in $\mathbb{R}^{k_1}$ by $k - k_1$ zeros, if an (ϵ, δ)-JL distribution existed for $\mathbb{R}^{k_1 \times d}$, it could be extended to an (ϵ, δ)-JL distribution for $\mathbb{R}^{k \times d}$.

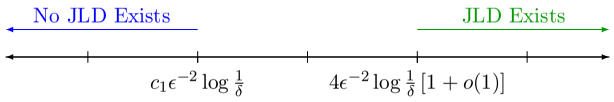


Figure 1: For fixed ϵ and δ, there exists a JL distribution for $k \geq 4\epsilon^{-2}\log(1/\delta)\,[1+o(1)]$. For $k < c_1\epsilon^{-2}\log(1/\delta)$, for some absolute constant $c_1 > 0$, there is no JL distribution. In this paper, we close this gap in the limit.

For any ϵ and δ, we define

$$k_0(\epsilon,\delta) = \min\{k : \text{there exists an } (\epsilon,\delta)\text{-JL distribution on } \mathbb{R}^{k\times d} \text{ for every } d \geq 1\}.$$

By our definition, $k_0 = k_0(\epsilon,\delta)$ is independent of d, and, by Theorem 1, we have that $k_0 \leq c\epsilon^{-2}\log(1/\delta)$ for some absolute constant $c > 0$. Frankl and Maehara (1988) show that $c \leq 9$. Achlioptas (2003) further improves this bound by providing a JL distribution with

$$k > 2\log(2/\delta)\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1},$$

resulting in the following upper bound:

$$k_0 \leq 2\log(2/\delta)\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} = 4\epsilon^{-2}\log(1/\delta)\,[1+o(1)]\,,$$

where $o(1)$ approaches zero as both ϵ and δ approach zero.

A lower bound on k_0 was not given until 2003 when Alon (2003) proved that

$$k_0 \geq c\epsilon^{-2}\log(1/\delta)\Big/\log(1/\epsilon)$$

for some absolute constant $c > 0$. Improving Alon's work, Jayram and Woodruff (2013) and Kane et al. (2011) showed, through different methods, that, for some absolute constant $c_1 > 0$, there is no (ϵ,δ)-JL distribution for $k \leq c_1\epsilon^{-2}\log\frac{1}{\delta}$. Hence, there is a lower bound of the form $k_0 \geq c_1\epsilon^{-2}\log\frac{1}{\delta}$. This situation is summarized in Figure 1.

The goal of the current paper is to close the gap between the upper and lower bounds in the limit. In particular, we prove an optimal lower bound that asymptotically matches the known upper bound when ϵ and δ approach 0, see Theorem 2. This means that $4\epsilon^{-2}\log(1/\delta)$ is an asymptotic threshold for k_0 where a phase change phenomenon occurs. The main result of this paper is captured in the following theorem:

Theorem 2 *For ϵ and δ sufficiently small, $k_0 \approx 4\epsilon^{-2}\log(1/\delta)$. More precisely,*

$$\lim_{\epsilon,\delta\to 0} \frac{k_0(\epsilon,\delta)}{4\epsilon^{-2}\log(1/\delta)} = 1.$$

The rest of the paper is organized as follows: To prove Theorem 2, we follow the approach of Kane et al. (2011). To make their constant c_1 explicit, however, we must use a more careful argument. In Section 2, we provide explicit conditions under which we prove the main result, Theorem 2. We delay the proofs of these conditions until Section 3 in order to make the main result more accessible, since only the statements of these results are needed and not their more technical proofs. In Section 3, we prove probabilistic bounds on $s = x_1^2 + \cdots + x_k^2$ where $x = (x_1, \cdots, x_k, \cdots, x_d)$ is a random variable uniformly distributed on S^{d-1}. These bounds explicitly determine the hidden constants in the bounds on $\text{Prob}\,[s < s_0(1-\epsilon)]$ and $\text{Prob}\,[s > s_0(1+\epsilon)]$ from (Kane et al., 2011, Theorem 20). These bounds can be viewed as explicit bounds for concentration theorems for laws of large numbers from probability theory. In the Appendix, we provide an alternate proof of a result in Kane et al. (2011), showing that if $d\Omega_{d-1}$ is the surface area for the $(d-1)$-dimensional sphere S^{d-1}, then for any $1 \leq k \leq d$,

$$d\Omega_{d-1} = \frac{1}{2}f(s)ds\,d\Omega_{k-1}d\Omega_{d-k-1},$$

where $s \in [0,1]$ and $f(s) = s^{\frac{k-2}{2}}(1-s)^{\frac{d-k-2}{2}}$.

2. Asymptotic Threshold Bound

In this section, we prove the asymptotic threshold bound for JL transformations. In particular, we provide specific conditions that result in the asymptotic threshold bound of $4\epsilon^{-2}\log(1/\delta)$. In Section 3, we prove that these conditions hold, but the details of these proofs are more technical and are unnecessary to understand the main results of this paper.

2.1. The Uniform Distribution on S^{d-1}

There is a unique probability distribution, called the uniform distribution, on the $(d-1)$-dimensional sphere S^{d-1} that is invariant under the orthonormal group. We express the uniform distribution on S^{d-1} in terms of the surface area differential form $d\Omega_{d-1}$, which means that, for any measurable subset $V \subset S^{d-1}$, the $(d-1)$-dimensional surface area of V is equal to the integral with respect to $d\Omega_{d-1}$, i.e., $\text{Vol}_{d-1}(V) = \int_V d\Omega_{d-1}$.[1] Thus, the uniform distribution on S^{d-1} corresponds to the measure $d\Omega_{d-1}/\text{Vol}_{d-1}\left(S^{d-1}\right)$.

As we are interested in reducing a d-dimensional vector to a k-dimensional vector for $1 \leq k < d$, we derive a relationship between the uniform distribution on S^{d-1} and the uniform distributions on S^{k-1} and S^{d-k-1}. Following the approach of Kane et al. (2011), for $1 \leq k < d$, we define an injective map

$$\Psi : S^{d-1} \to [0,1] \times S^{k-1} \times S^{d-k-1}$$

as follows: For any $x = (x_1, x_2, \ldots, x_d)^t \in S^{d-1}$, we define s in $\Psi(x) = (s,u,v)$ as $s = x_1^2 + \cdots + x_k^2$. In the case where $0 < s < 1$, we define

$$u = (x_1, \ldots, x_k)^t/\sqrt{s} \quad \text{and} \quad v = (x_{k+1}, \ldots, x_d)^t/\sqrt{1-s}.$$

1. In this paper, we suppress the pullback maps on equalities for differential forms since there is a unique (almost) bijective map under consideration in each case. We leave the details to the interested reader.

When $s = 0$, i.e., $x_1 = \cdots = x_k = 0$, we define $u = (1, 0, \ldots, 0)^t$ (or any point in S^{k-1}) and $v = (x_{k+1}, \ldots, x_d)^t$. Similarly, for $s = 1$, we define $u = (x_1, \ldots, x_k)^t$ and $v = (1, 0, \ldots, 0)^t$ (or any point in S^{d-k-1}). It is straight-forward to check that Ψ is injective. In addition, the complement of the image of Ψ is a subset of $\{0, 1\} \times S^{k-1} \times S^{d-k-1}$, which has, in turn, $(d-1)$-dimensional surface area equal to 0. Therefore, when convenient, we assume that $s \in (0, 1)$.

For $s \in [0, 1]$, we define

$$f(s) = s^{\frac{k-2}{2}}(1-s)^{\frac{d-k-2}{2}}.$$

In the Appendix, we provide alternative proof to the computation in Kane et al. (2011) which shows that, via the map Ψ,

$$d\Omega_{d-1} = \frac{1}{2} f(s) ds \, d\Omega_{k-1} d\Omega_{d-k-1}.$$

Equivalently, in term of probability distributions,

$$\frac{d\Omega_{d-1}}{\mathrm{Vol}_{d-1}\left(S^{d-1}\right)} = Bf(s)ds \frac{d\Omega_{k-1}}{\mathrm{Vol}_{k-1}\left(S^{k-1}\right)} \frac{d\Omega_{d-k-1}}{\mathrm{Vol}_{d-k-1}\left(S^{d-k-1}\right)}, \tag{2}$$

where

$$B = \frac{1}{2} \frac{\mathrm{Vol}_{k-1}\left(S^{k-1}\right) \mathrm{Vol}_{d-k-1}\left(S^{d-k-1}\right)}{\mathrm{Vol}_{k-1}\left(S^{k-1}\right)} = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{k}{2})\Gamma(\frac{d-k}{2})}.$$

is an appropriate scaling constant, depending on the gamma function, for more details, see the Appendix. Moreover, in this situation, we observe that $Bf(s)$ is a probability distribution on $[0, 1]$. This implies that the uniform distribution on S^{d-1} is a direct product of the distributions on the factors. In other words, a uniformly distributed random variable X_{d-1} on S^{d-1} can be decomposed into three random variables $\Psi(X_{d-1}) = (S, X_{k-1}, X_{d-k-1})$ with the following properties:

(i) S is a random variable on $[0, 1]$ with density function $Bf(s)$,

(ii) X_{k-1} and X_{d-k-1} are uniformly distributed on S^{k-1} and S^{d-k-1}, and

(iii) The random variables S, X_{k-1}, and X_{d-k-1} are *independent*.

The independence of these three random variables is a key property in our proof as it allows us to study the three spaces separately.

2.2. Upper Bound: Explicit JL Distribution

We recall that Achlioptas (2003) proved that

$$k_0(\epsilon, \delta) \leq 2\log(2/\delta) \left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)^{-1} = 4\epsilon^{-2}\log(1/\delta)\left[1 + o(1)\right].$$

In this section, we give an alternate proof of this result using the approach and bounds from this paper.

We recall the following construction by Dasgupta and Gupta (2003): A distribution $\mathcal{D}$ on $k \times d$ matrices is formed by picking a $d \times d$ orthonormal matrix $V = (v_1, \ldots, v_d)^t$ uniformly

at random with respect to the Haar measure on orthonormal matrices and then letting $A = \frac{1}{\sqrt{s_0}}(v_1, \ldots, v_k)^t$, where $s_0 = k/d$. The following proposition shows that $k_0(\epsilon, \delta) \leq 4\epsilon^{-2}\log(1/\delta)\left[1 + o(1)\right]$, which, in turn, implies that the limit appearing in Theorem 2 (if it exists) is at most 1:

Proposition 3 *Let $0 < \epsilon, \delta < \frac{1}{2}$ and $s_0 = k/d$. Suppose that there is some constant C so that*

$$\max\{\mathrm{Prob}_{x\sim S^{d-1}}\left[s < s_0(1-\epsilon)\right], \mathrm{Prob}_{x\sim S^{d-1}}\left[s > s_0(1+\epsilon)\right]\} \leq Ce^{-\frac{k-2}{4}\epsilon^2\left(1-\frac{2}{3}\epsilon\right)},$$

where $s = x_1^2 + \cdots + x_k^2$ is defined as in $\Psi(x) = (s, u, v)$. Then, there exists an $o(1)$ function, which approaches zero as both ϵ and δ approach zero so that if $k > 4\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\left[1 + o(1)\right]$, then the distribution on $k \times d$ random matrices defined as above is an (ϵ, δ)-JL distribution, that is, for any $w \in S^{d-1}$,

$$\mathrm{Prob}_{A\sim\mathcal{D}}\left[\left|\|Aw\|_2^2 - 1\right| < \epsilon\right] \geq 1 - \delta.$$

Proof Let V be the random orthogonal matrix as defined above, and let $x = (x_1, \ldots, x_d)^t = Vw$. Then $Aw = \sqrt{s_0^{-1}}(x_1, \ldots, x_k)^t$, and

$$||Aw||_2^2 = \frac{1}{s_0}(x_1^2 + \cdots + x_k^2).$$

Since V is orthonormal and $||w||_2 = 1$, we have that $||x||_2 = 1$, and, hence, $x \in S^{d-1}$. We observe that since V is a random orthogonal matrix, for fixed $w \in S^{d-1}$, $x = Vw$ is a random variable, uniformly distributed on S^{d-1}. Hence,

$$\mathrm{Prob}_{A\sim\mathcal{D}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] = \mathrm{Prob}_{x\sim S^{d-1}}\left[\left|\frac{1}{s_0}\sum_{i=1}^{k} x_i^2 - 1\right| > \epsilon\right],$$

where $x \sim S^{d-1}$ means that x is a random variable uniformly distributed on S^{d-1}. Let $s = \sum_{i=1}^{k} x_i^2$. Then, $s \in [0, 1]$ and the probability above becomes

$$\mathrm{Prob}_{x\sim S^{d-1}}\left[s < s_0(1-\epsilon)\right] + \mathrm{Prob}_{x\sim S^{d-1}}\left[s > s_0(1+\epsilon)\right] \leq 2Ce^{-\frac{k-2}{4}\epsilon^2\left(1-\frac{2}{3}\epsilon\right)}, \tag{3}$$

by assumption. We observe that when

$$k > 4\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\left[1 + \frac{2\epsilon}{3-2\epsilon} + \frac{\log(2C)}{\log\left(\frac{1}{\delta}\right)}\frac{1}{1-2\epsilon/3} + \frac{2\epsilon^2}{4\log\left(\frac{1}{\delta}\right)}\right] = 4\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\left[1 + o(1)\right], \tag{4}$$

the right-hand-side of Inequality (3) is less than δ. In this case, the $o(1)$ term needed in the theorem statement appears in Inequality (4). Therefore, when $k > 4\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\left[1 + o(1)\right]$, the distribution $\mathcal{D}$ is an (ϵ, δ)-JL distribution. ■

We observe that this result also follows from work in Dasgupta and Gupta (2003). In particular, (Dasgupta and Gupta, 2003, Lemma 2.2) can be used to derive bounds similar to the assumed bounds in this statement.

2.3. Lower Bound for Arbitrary Distributions

In this section, we prove an optimal lower bound on the limit in Theorem 2 that matches the upper bound from the previous section. The proof of this lower bound is the main challenge in this paper. We begin with the following key lemma:

Lemma 4 *Let $x = (x_1, \ldots, x_d)^t$ be a random variable, uniformly distributed on S^{d-1}, $\Psi(x) = (s, u, v)$, and $s_0 = k/d$. Suppose that for a fixed $\epsilon > 0$,*

$$\min\{\text{Prob}\left[s > s_0(1+\epsilon)\right], \text{Prob}\left[s < s_0(1-\epsilon)\right]\} \geq L,$$

where s is a random variable with probability distribution $Bf(s)$ on $[0,1]$. For any function $c(u,v) > 0$ depending only on $u \in S^{k-1}$ and $v \in S^{d-k-1}$ (i.e., independent of s), we have

$$\text{Prob}_{x \sim S^{d-1}}\left[|sc - 1| > \epsilon\right] \geq L.$$

Proof By the equality of differential forms in Equation (2),

$$\begin{aligned}\text{Prob}\left[|sc-1| > \epsilon\right] &= \int_{|sc-1|>\epsilon} Bf(s)ds \frac{d\Omega_{k-1}}{\text{Vol}_{k-1}\left(S^{k-1}\right)} \frac{d\Omega_{d-k-1}}{\text{Vol}_{d-k-1}\left(S^{d-k-1}\right)} \\ &= \int_{S^{k-1}\times S^{d-k-1}} \left(\int_{|sc-1|>\epsilon} Bf(s)ds\right) \frac{d\Omega_{k-1}}{\text{Vol}_{k-1}\left(S^{k-1}\right)} \frac{d\Omega_{d-k-1}}{\text{Vol}_{d-k-1}\left(S^{d-k-1}\right)}.\end{aligned}$$

Our goal is to find a lower bound on the integral $\int_{|sc-1|>\epsilon} Bf(s)ds$. Due to the independence of u, v, and s, $c(u,v)$ is a fixed positive constant within this integral. We observe that $|sc-1| > \epsilon$ consists of two intervals, $s < (1-\epsilon)/c$ and $s > (1+\epsilon)/c$, and we consider two cases depending on the value of c.

We begin by recalling that

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] = \int_{s>s_0(1+\epsilon)} Bf(s)ds \quad \text{and} \quad \text{Prob}\left[s < s_0(1-\epsilon)\right] = \int_{s<s_0(1-\epsilon)} Bf(s)ds.$$

If $c \geq s_0$, then $(1+\epsilon)/c \leq (1+\epsilon)/s_0$, and, hence

$$\int_{|sc-1|>\epsilon} Bf(s)ds \geq \int_{s>(1+\epsilon)/c} Bf(s)ds \geq \int_{s>(1+\epsilon)/s_0} Bf(s)ds \geq L.$$

On the other hand, if $c < s_0$, then $(1-\epsilon)/s_0 < (1-\epsilon)/c$, then

$$\int_{|sc-1|>\epsilon} Bf(s)ds \geq \int_{s<(1-\epsilon)/c} Bf(s)ds \geq \int_{s<(1-\epsilon)/s_0} Bf(s)ds \geq L.$$

Therefore, the integral $\int_{|sc-1|>\epsilon} Bf(s)ds$ is bounded from below by L, and

$$\text{Prob}\left[|sc-1| > \epsilon\right] \geq \int_{S^{k-1}\times S^{d-k-1}} L \frac{d\Omega_{k-1}}{\text{Vol}_{k-1}\left(S^{k-1}\right)} \frac{d\Omega_{d-k-1}}{\text{Vol}_{d-k-1}\left(S^{d-k-1}\right)} = L. \quad \blacksquare$$

We now show that when $k \leq \eta\epsilon^{-2}\log(1/\delta)$ with $\eta < 4$, and ϵ and δ are sufficiently small, there does not exist an (ϵ,δ)-JL distribution on $\mathbb{R}^{k\times d}$. This fact, combined with the results in Section 2.2, shows that the limit appearing in Theorem 2 exists and equals 1. In order to show this, we consider the following related problem: By definition, for a probability distribution $\mathcal{D}$ on $\mathbb{R}^{k\times d}$ to be an (ϵ,δ)-JL distribution, the following inequality must hold for every $w \in S^{d-1}$:

$$\text{Prob}_{A\sim\mathcal{D}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] < \delta.$$

Hence,

$$\text{Prob}_{A\sim\mathcal{D},\ w\sim S^{d-1}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] < \delta, \tag{5}$$

where $w \in S^{d-1}$ is a random variable distributed uniformly on S^{d-1}. Following the approach of Kane et al. (2011), our goal is to prove that, for every $A \in \mathbb{R}^{k\times d}$,

$$\text{Prob}_{w\sim S^{d-1}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] > \delta. \tag{6}$$

When Inequality (6) holds for all A, then Inequality (5) cannot hold for any distribution $\mathcal{D}$ on $\mathbb{R}^{k\times d}$. Therefore, an (ϵ,δ)-JL distribution does not exist. We make this precise in the following theorem:

Theorem 5 *Suppose that $\eta < 4$ and let $k(\epsilon,\delta) = \left\lfloor \eta\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\right\rfloor$. Let $s_0 = k/d$, and suppose that, for every ϵ, δ, and s_0 sufficiently small (to make s_0 sufficiently small, d must be sufficiently large),*

$$\min\{\text{Prob}\left[s > s_0(1+\epsilon)\right], \text{Prob}\left[s < s_0(1-\epsilon)\right]\} \geq C\delta^{\frac{\eta}{4}\gamma},$$

where $C > 0$ is an absolute constant, and γ approaches 1 as ϵ, δ, and s_0 approach 0. Then, by decreasing ϵ, δ, and s_0 as needed, for every matrix $A \in \mathbb{R}^{k(\epsilon,\delta)\times d}$,

$$\text{Prob}_{w\sim S^{d-1}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] > \delta.$$

Proof We assume that A has rank $k = k(\epsilon,\delta)$ since, if not, we may reduce k (and decrease η correspondingly) to the rank of A. Let $A = U\Sigma V^t$ be the singular value decomposition of A where U is a $k\times k$ orthonormal matrix, $V = (v_1,\ldots,v_d)$ is a $d\times d$ orthonormal matrix, and Σ is a $k\times d$ diagonal matrix with $\lambda_i > 0$ its entry at $\Sigma_{i,i}$ for $1 \leq i \leq k$. Let

$$x = (x_1,\ldots,x_d)^t = V^t w.$$

Since V is orthonormal, we have $x \in S^{d-1}$. We observe that since w is a uniformly distributed random variable on S^{d-1}, $V^t w$ is also a uniformly distributed random variable on S^{d-1}. Therefore, since U is orthonormal, we have

$$\|Aw\|_2^2 = \|U\Sigma x\|_2^2 = \|\Sigma x\|_2^2 = \sum_{i=1}^k \lambda_i^2 x_i^2.$$

We recall the definition of the function $\Psi(x) = (s,u,v)$ where $s = x_1^2 + \cdots + x_k^2$. Moreover, we restrict our attention to the case where $s \in (0,1)$ since the complement has zero measure. Let

$$c = \sum_{i=1}^k \lambda_i^2 x_i^2/s = \|\Sigma_{k\times k} u\|_2^2,$$

where $\Sigma_{k\times k}$ denotes the $k \times k$ principal submatrix of Σ, then

$$\text{Prob}_{w\sim S^{d-1}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] = \text{Prob}_{x\sim S^{d-1}}\left[|sc - 1| > \epsilon\right].$$

Due to the independence of u, v, and s, it follows that c depends only on u. Therefore, by Lemma 4, it follows that

$$\text{Prob}_{w\sim S^{d-1}}\left[\left|\|Aw\|_2^2 - 1\right| > \epsilon\right] \geq C\delta^{\frac{\eta}{4}\gamma}.$$

It follows that for ϵ, δ, and s_0 sufficiently small, $C\delta^{\frac{\eta}{4}\gamma} > \delta$. ■

Therefore, by selecting ϵ, δ, and s_0 sufficiently small, it follows from Theorem 5 that there is no (ϵ, δ)-JL distribution when $k < \eta\epsilon^{-2}\log\left(\frac{1}{\delta}\right)$ for $\eta < 4$. Therefore, $k_0(\epsilon, \delta) \geq \eta\epsilon^{-2}\log\left(\frac{1}{\delta}\right)$. We collect the results of Proposition 3 and Theorem 5 in the following corollary:

Corollary 6 *Assume the hypotheses on* $\text{Prob}\left[s > s_0(1+\epsilon)\right]$ *and* $\text{Prob}\left[s < s_0(1-\epsilon)\right]$ *from Proposition 3 and Theorem 5 hold.*

(a) There exists an $o(1)$ *function that approaches* 0 *as* ϵ *and* δ *approach zero such that if* $k > 4\epsilon^{-2}\log\left(\frac{1}{\delta}\right)[1 + o(1)]$*, then there exists a JL distribution.*

(b) If $k(\epsilon, \delta) = \left\lfloor \eta\epsilon^{-2}\log\left(\frac{1}{\delta}\right)\right\rfloor$*, then, by decreasing* ϵ *and* δ*, and increasing* d*, there is no* (ϵ, δ)*-JL distribution for any* $k' \leq k(\epsilon, \delta)$*.*

This proves the main result in the paper. In the following section, we provide the more technical results that verify the assumptions in Proposition 3 and Theorem 5.

3. Explicit Concentration Bounds

Throughout this section, we assume that s is a random variable with probability distribution $Bf(s)$. We define $s_0 = \frac{k}{d}$, and we further assume that $0 \leq \epsilon, \delta \leq 1/2$, $k - 4 \geq \epsilon^{-2}$ and $s_0 < 0.4$. We derive lower and upper bounds for the following probabilities:

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] \quad \text{and} \quad \text{Prob}\left[s < s_0(1-\epsilon)\right],$$

using the probability density $Bf(s)$ for s and $f(s) = s^{(k-2)/2}(1-s)^{(d-k-2)/2}$. These probabilities appear in (Kane et al., 2011, Theorem 20) without the explicit constants that are derived in this paper. These bounds are instances of explicit concentration theorems (or explicit laws of large numbers) from probability theory. Our goal is to formulate these bounds as precisely as possible so that the lower and upper bounds are asymptotically the same when ϵ and δ approach 0.

3.1. Bounds for B

We recall that $\Gamma(1/2) = \sqrt{\pi}$, $\Gamma(1) = 1$, and $\Gamma(1+z) = z\Gamma(z)$. Hence

$$\Gamma\left(\frac{d}{2}\right) = \left(\frac{d}{2} - 1\right)! \text{ if } d \text{ is even, and } \Gamma\left(\frac{d}{2}\right) = \left(\frac{d}{2} - 1\right)\left(\frac{d}{2} - 2\right)\cdots\frac{3}{2}\frac{1}{2}\sqrt{\pi} \text{ if } d \text{ is odd.}$$

In this section, we derive lower and upper bounds for B, see Equation (38), by using the following form of Stirling's approximation of $n!$ due to Robbins (1955):

$$\sqrt{2\pi}n^{n+1/2}e^{-n}e^{\frac{1}{12n+1}} < \Gamma(n+1) = n! < \sqrt{2\pi}n^{n+1/2}e^{-n}e^{\frac{1}{12n}}.$$

Since we are interested in the asymptotic behavior, we focus on the case where d is even. This choice does not affect the asymptotic results of our paper, but the calculations are more straight-forward in this case. We leave the details for the case where d is odd to the interested reader.

Lemma 7 *Suppose* k *and* d *are both even. Then we have the following inequality:*[2]

$$\frac{e^{-2}}{2\sqrt{\pi}}\frac{(d-2)^{(d-1)/2}}{(k-2)^{(k-1)/2}(d-k-2)^{(d-k-1)/2}} \leq B \leq \frac{e^{-1}}{2\sqrt{\pi}}\frac{(d-2)^{(d-1)/2}}{(k-2)^{(k-1)/2}(d-k-2)^{(d-k-1)/2}}.$$

Proof Using the bound on $n!$ from Robbins (1955), we obtain

$$C_0\frac{(d-2)^{(d-1)/2}}{(k-2)^{(k-1)/2}(d-k-2)^{(d-k-1)/2}} \leq B \leq C_1\frac{(d-2)^{(d-1)/2}}{(k-2)^{(k-1)/2}(d-k-2)^{(d-k-1)/2}},$$

where

$$C_0 = \frac{1}{2\sqrt{\pi}}e^{-1}e^{\frac{1}{6(d-2)+1}}e^{\frac{-1}{6(k-2)}}e^{\frac{-1}{6(d-k-2)}} \geq \frac{e^{-2}}{2\sqrt{\pi}}, \quad \text{and}$$

$$C_1 = \frac{1}{2\sqrt{\pi}}e^{-1}e^{\frac{1}{6(d-2)}}e^{\frac{-1}{6(k-2)+1}}e^{\frac{-1}{6(d-k-2)+1}} \leq \frac{e^{-1}}{2\sqrt{\pi}}.$$ ■

Corollary 8 *With* $s_0 = k/d$*, we have*

$$\frac{e^{-2}}{2\sqrt{\pi}}\sqrt{k} \leq Bs_0f(s_0) \leq \frac{9e^{-1}}{\sqrt{2\pi}}\sqrt{k}.$$

Proof By evaluating f at s_0 and replacing B by its lower bound found in Lemma 7, we obtain the lower bound

$$Bs_0f(s_0) \geq \frac{e^{-2}}{2\sqrt{\pi}}\sqrt{k}\left(\frac{d-2}{d-k}\right)^{\frac{1}{2}}\left(\frac{k(d-2)}{d(k-2)}\right)^{\frac{k-1}{2}}\left(\frac{(d-k)(d-2)}{d(d-k-2)}\right)^{\frac{d-k-1}{2}} \geq \frac{e^{-2}}{2\sqrt{\pi}}\sqrt{k}.$$

Similarly, by using the upper bound in Lemma 7, we obtain the upper bound

$$Bs_0f(s_0) \leq \frac{e^{-1}}{2\sqrt{\pi}}\sqrt{k}\left(\frac{d-2}{d}\right)^{\frac{d-1}{2}}\left(\frac{k}{k-2}\right)^{\frac{k-1}{2}}\left(\frac{d-k}{d-k-2}\right)^{\frac{d-k-1}{2}}\frac{\sqrt{d}}{\sqrt{d-k}} \leq \frac{9e^{-1}}{\sqrt{2\pi}}\sqrt{k},$$

where the last inequality follows from $\frac{d}{d-k} \leq 2$ since $s_0 < 0.4$, and $\left(\frac{x}{x-2}\right)^{\frac{x-1}{2}} \leq 3$ for $x \geq 3$. ■

2. Throughout this section, it is possible to derive tighter bounds on constants in the following inequalities, but the ones appearing here are sufficient for our proofs. We leave the details of the tighter bounds to the interested reader.

3.2. Bounds on Prob $[s > s_0(1+\epsilon)]$

We begin by mentioning the following inequalities which are used in our arguments below:

$$\log(1+x) \geq x - \frac{x^2}{2} \quad \text{for } 0 < x < 1, \tag{7}$$

$$\log(1-x) \geq -x - x^2 \quad \text{for } 0 < x < 0.68, \tag{8}$$

$$\log(1+x) \leq x \quad \text{for } x > -1, \quad \text{and} \tag{9}$$

$$\log(1+x) \leq x - \frac{x^2}{2} + \frac{x^3}{3} \quad \text{for } x > -1. \tag{10}$$

These bounds can be verified by employing basic calculus techniques (e.g., derivatives and Taylor expansions) as well as sufficiently accurate approximations. Using these inequalities, we derive the following bounds:

Lemma 9

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] \geq \frac{e^{-2}}{4\sqrt{\pi}} e^{-\frac{1}{4}(\sqrt{k}\epsilon+1)^2 \frac{1+s_0}{1-s_0}}.$$

Moreover, when $k < \eta\epsilon^{-2}\log\frac{1}{\delta}$,

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] \geq \frac{e^{-2}}{4\pi}\delta^{\frac{\eta}{4}\gamma_1},$$

where

$$\gamma_1 = \left(1 + (\eta\log(1/\delta))^{-1/2}\right)^2 \left(\frac{1+s_0}{1-s_0}\right).$$

Additionally, γ_1 *approaches* 1 *as* ϵ, δ, *and* s_0 *approach* 0.

Proof Note that

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] = B\int_{s>s_0(1+\epsilon)} f(s)ds = Bs_0\int_{\epsilon}^{\frac{1}{s_0}-1} f(s_0(1+x))dx, \tag{11}$$

via the substitution $s = s_0(1+x)$.

Let $g(s) = s^{k/2}(1-s)^{(d-k)/2}$, then $f(s_0(1+x))$ can be expressed in terms of $g(s)$, namely,

$$f(s_0(1+x)) = \frac{g(s_0(1+x))}{s_0(1+x)\left(1-s_0(1+x)\right)}. \tag{12}$$

To find a lower bound on Prob $[s > s_0(1+\epsilon)]$, we compute a bound on $g(s_0(1+x))$ from below. Taking the logarithm of $g(s_0(1+x))$, we find

$$\log\left(g(s_0(1+x))\right) = \log g(s_0) + \frac{d}{2}\left(s_0\log(1+x) + (1-s_0)\log\left(1 - \frac{s_0}{1-s_0}x\right)\right). \tag{13}$$

Restricting x to the interval $0 \leq x < 1$, it then follows that $0 < \frac{s_0}{1-s_0}x < 0.68$ from the assumption that $s_0 < 0.4$. We now bound the second term in Equation (13) using Inequalities (7) and (8), as follows:

$$s_0\log(1+x) + (1-s_0)\log\left(1 - \frac{s_0}{1-s_0}x\right) \geq -\left(\frac{s_0(1+s_0)}{2(1-s_0)}\right)x^2. \tag{14}$$

Hence, by substituting Inequality (14) into Equation (13) and exponentiating, we obtain the following lower bound for $g(s_0(1+x))$:

$$g(s_0(1+x)) \geq g(s_0)e^{-\frac{k}{4}x^2\frac{1+s_0}{1-s_0}}. \tag{15}$$

We also observe that since $s_0 < 0.4$ and $0 \leq x < 1$, the denominator of Equation (12) is bounded from below as follows:

$$\frac{1}{s_0(1+x)(1-s_0(1+x))} \geq \frac{1}{2s_0(1-s_0)}. \tag{16}$$

Therefore, by substituting Inequalities (15) and (16) into Equation (12) when $0 \leq x < 1$, we have

$$f(s_0(1+x)) \geq \frac{g(s_0)}{2s_0(1-s_0)}e^{-\frac{k}{4}x^2\frac{1+s_0}{1-s_0}} = \frac{1}{2}f(s_0)e^{-\frac{k}{4}x^2\frac{1+s_0}{1-s_0}}. \tag{17}$$

Since $s_0 < 0.4$, we observe that $\frac{1}{s_0} - 1 = \frac{d-k}{k} > 1.5$. Since we assumed that $\epsilon < \frac{1}{2}$ and $k \geq 4 + \epsilon^{-2} > 4$, it follows that $\epsilon + k^{-1/2} < 1$ and so $\epsilon + k^{-1/2} < 1 < \frac{1}{s_0} - 1$. Therefore, we further restrict x to the interval $(\epsilon, \epsilon + k^{-1/2})$ and observe that Inequality (17) applies in this range. Therefore, since f is a positive function,

$$Bs_0\int_{\epsilon}^{\frac{1}{s_0}-1} f(s_0(1+x))dx \geq Bs_0\int_{\epsilon}^{\epsilon+k^{-1/2}} f(s_0(1+x))dx \geq \frac{1}{2}Bs_0f(s_0)\int_{\epsilon}^{\epsilon+k^{-1/2}} e^{-\frac{k}{4}x^2\frac{1+s_0}{1-s_0}}dx. \tag{18}$$

Replacing $Bs_0f(s_0)$ with its lower bound given in Corollary 8 and observing that the integrand is decreasing over an interval of width $k^{-1/2}$, Inequality (18) is bounded from below by

$$\frac{1}{2}Bs_0f(s_0)\int_{\epsilon}^{\epsilon+k^{-1/2}} e^{-\frac{k}{4}x^2\frac{1+s_0}{1-s_0}}dx \geq \frac{e^{-2}}{4\sqrt{\pi}}e^{-\frac{1}{4}(\sqrt{k}\epsilon+1)^2\frac{1+s_0}{1-s_0}},$$

which completes the first inequality. The second inequality follows by replacing k by the given upper bound and simplifying. ■

Lemma 10

$$\text{Prob}\left[s > s_0(1+\epsilon)\right] \leq \frac{27e^{-1}}{\sqrt{2\pi}}e^{-\frac{k-2}{4}\epsilon^2(1-\frac{2}{3}\epsilon)}.$$

Proof To derive an upper bound, we start with the expression for Prob $[s > (1+\epsilon)s_0]$ from Equation (11). We first find an upper bound on $f(s_0(1+x))$. Then, we bound $f(s_0(1+x))$ using the inequality $1 - x \leq e^{-x}$ for all x as follows:

$$f(s_0(1+x)) = f(s_0)(1+x)^{\frac{k-2}{2}}\left(1 - \frac{s_0}{1-s_0}x\right)^{\frac{d-k-2}{2}} \leq f(s_0)(1+x)^{\frac{k-2}{2}}\left(e^{\frac{-s_0}{1-s_0}x}\right)^{\frac{d-k-2}{2}}, \tag{19}$$

Moreover, since $s_0 < 0.4$,

$$\frac{s_0}{1-s_0}\frac{d-k-2}{2} = \frac{k}{d-k}\frac{d-k-2}{2} > \frac{k-2}{2}. \tag{20}$$

By applying Inequality (20) to Inequality (19), we derive the upper bound

$$f(s_0)(1+x)^{\frac{k-2}{2}}e^{-\frac{k-2}{2}x}.$$

Therefore, by extending the region of integration in Inequality (11), we find the following upper bound on the probability:

$$Bs_0\int_\epsilon^{\frac{1}{s_0}-1} f(s_0(1+x))dx \le Bs_0 f(s_0)\int_\epsilon^\infty (1+x)^{\frac{k-2}{2}}e^{-\frac{k-2}{2}x}dx. \tag{21}$$

By integrating by parts, we observe that for any ℓ and m with $1 \le \ell \le m$,

$$\int_\epsilon^\infty (1+x)^\ell e^{-mx}dx \le \frac{1}{m}(1+\epsilon)^\ell e^{-m\epsilon} + \int_\epsilon^\infty (1+x)^{\ell-1}e^{-mx}dx. \tag{22}$$

Applying Inequality (22) $\frac{k-2}{2}$ times to the integral in Inequality (21) and bounding the resulting geometric series from above gives

$$\int_\epsilon^\infty (1+x)^{\frac{k-2}{2}}e^{-\frac{k-2}{2}x}dx \le \frac{2e^{-\frac{k-2}{2}\epsilon}}{k-2}\sum_{i=0}^{\frac{k-2}{2}}(1+\epsilon)^i \le \frac{2(1+\epsilon)}{\epsilon(k-2)}(1+\epsilon)^{\frac{k-2}{2}}e^{-\frac{k-2}{2}\epsilon}.$$

By applying Inequality (10) to $(1+\epsilon)^{\frac{k-2}{2}} = e^{\frac{k-2}{2}\log(1+\epsilon)}$, we obtain the upper bound

$$\frac{2(1+\epsilon)}{\epsilon(k-2)}(1+\epsilon)^{\frac{k-2}{2}}e^{-\frac{k-2}{2}\epsilon} \le \frac{2(1+\epsilon)}{\epsilon(k-2)}e^{-\frac{k-2}{4}\epsilon^2(1-\frac{2}{3}\epsilon)}.$$

Since $k-4 \ge \epsilon^{-2}$, it follows that $\frac{(k-2)^2}{k} \ge k-4 \ge \epsilon^{-2}$, and, hence, that $\epsilon(k-2) \ge \sqrt{k}$. Therefore, we can further simplify our bound to

$$\frac{2(1+\epsilon)}{\epsilon(k-2)}e^{-\frac{k-2}{4}\epsilon^2(1-\frac{2}{3}\epsilon)} \le \frac{2(1+\epsilon)}{\sqrt{k}}e^{-\frac{k-2}{4}\epsilon^2(1-\frac{2}{3}\epsilon)}. \tag{23}$$

By combining Inequalities (21) and (23), we find an upper bound on the probability as follows:

$$Bs_0\int_\epsilon^{\frac{1}{s_0}-1} f(s_0(1+x))dx \le Bs_0 f(s_0)\frac{2(1+\epsilon)}{\sqrt{k}}e^{-\frac{k-2}{4}\epsilon^2(1-\frac{2}{3}\epsilon)}.$$

Applying the upper bound on $Bs_0f(s_0)$ from Corollary 8 and the assumption that $\epsilon \le \frac{1}{2}$ completes the inequality. ■

3.3. Bounds on Prob $[s < s_0(1-\epsilon)]$

We begin this section by including the following two additional inequalities on $\log(1-x)$:

$$\log(1-x) \ge -x - \frac{x^2}{2} - x^3 \quad \text{for } 0 < x < 0.815, \tag{24}$$

$$\log(1-x) \le -x - \frac{x^2}{2} \quad \text{for } 0 \le x < 1. \tag{25}$$

These bounds can be justified using a similar approach as for Inequalities (7-10). Using these inequalities, we derive the following bounds:

Lemma 11

$$\text{Prob}\left[s < s_0(1-\epsilon)\right] \ge \frac{e^{-2}}{2\sqrt{\pi}}e^{-\frac{1}{4}\left(\frac{(\sqrt{k}\epsilon+1)^2}{1-s_0}+2(\sqrt[3]{k}\epsilon+k^{-1/6})^3\right)}.$$

Moreover, when $k < \eta\epsilon^{-2}\log\frac{1}{\delta}$,

$$\text{Prob}\left[s < s_0(1-\epsilon)\right] \ge \frac{e^{-2}}{2\sqrt{\pi}}\delta^{\frac{\eta}{4}\gamma_2},$$

where

$$\gamma_2 = \frac{1}{1-s_0}\left(1+\frac{1}{\sqrt{\eta\log\frac{1}{\delta}}}\right)^2 + 2\left(\epsilon^{1/3}+\frac{1}{\sqrt[3]{\eta\log\frac{1}{\delta}}}\right)^3.$$

Additionally, γ_2 *approaches* 1 *as* ϵ, δ, *and* s_0 *approach* 0.

Proof The proof of this lemma is very similar to the proof of Lemma 9, so we focus on the new details. The probability can be rewritten, using the substitution $s = s_0(1-x)$, as

$$\text{Prob}\left[s < (1-\epsilon)s_0\right] = B\int_{s<s_0(1-\epsilon)} f(s)ds = Bs_0\int_\epsilon^1 f(s_0(1-x))dx. \tag{26}$$

Using $g(s)$ as in Lemma 9, it follows that

$$f(s_0(1-x)) = \frac{g(s_0(1-x))}{s_0(1-x)(1-s_0(1-x))} \tag{27}$$

and

$$\log g(s_0(1-x)) = \log g(s_0) + \frac{d}{2}\left(s_0\log(1-x) + (1-s_0)\log\left(1+\frac{s_0}{1-s_0}x\right)\right). \tag{28}$$

Since $0 < x \le 1$, it follows that $0 < \frac{s_0}{1-s_0}x < 0.68$ from the assumption that $s_0 < 0.4$. Therefore, we can bound the second term in Equation (28) using Inequalities (7) and (24), as follows:

$$s_0\log(1-x) + (1-s_0)\log\left(1+\frac{s_0}{1-s_0}x\right) \ge \frac{-s_0}{2(1-s_0)}x^2 - s_0x^3. \tag{29}$$

Substituting Inequality (29) into Expression (28) and exponentiating, we get

$$g(s_0(1-x)) \ge g(s_0)e^{-\frac{k}{4}\left(\frac{x^2}{1-s_0}+2x^3\right)}. \tag{30}$$

Since $s_0 < 0.4$, it follows that $\frac{s_0}{1-s_0} < 1$, and, hence, that $(1-x)\left(1+\frac{s_0}{1-s_0}x\right) < 1$. Therefore, the denominator in Equation (27) can be bounded from above by

$$s_0(1-x)\left(1-s_0(1-x)\right) = s_0(1-s_0)(1-x)\left(1+\frac{s_0x}{1-s_0}\right) \le s_0(1-s_0). \tag{31}$$

Therefore, by substituting Inequalities (30) and (31) into Expression (27), when $\epsilon < x < 1$, we have

$$f(s_0(1-x)) \geq f(s_0)e^{-\frac{k}{4}\left(\frac{x^2}{1-s_0}+2x^3\right)}. \tag{32}$$

Since $\epsilon < \frac{1}{2}$ and $k \geq 4+\epsilon^{-2} > 4$, it follows that $\epsilon + k^{-1/2} < 1$. Therefore, we restrict x to the interval $(\epsilon, \epsilon + k^{-1/2})$, and observe that Inequality (32) applies in this range. Therefore,

$$Bs_0 \int_\epsilon^1 f(s_0(1-x))dx \geq Bs_0 f(s_0) \int_\epsilon^{\epsilon+k^{-1/2}} e^{-\frac{k}{4}\left(\frac{x^2}{1-s_0}+2x^3\right)}dx. \tag{33}$$

Replacing $Bs_0 f(s_0)$ with its lower bound given in Corollary 8 and observing that the integrand is decreasing on an interval of width $k^{-1/2}$, Inequality (33) is bounded from below by

$$Bs_0 f(s_0) \int_\epsilon^{\epsilon+k^{-1/2}} e^{-\frac{k}{4}\left(\frac{x^2}{1-s_0}+2x^3\right)}dx \geq \frac{e^{-2}}{2\sqrt{\pi}} e^{-\frac{1}{4}\left(\frac{(\sqrt{k}\epsilon+1)^2}{1-s_0}+2(\sqrt[3]{k}\epsilon+k^{-1/6})^3\right)},$$

which completes the first inequality. The second inequality follows by replacing $k \geq 1$ by the given upper bound and simplifying. ■

Lemma 12

$$\text{Prob}\left[s < s_0(1-\epsilon)\right] \leq \frac{18\sqrt{2}e^{1/2}}{\sqrt{\pi}} e^{-\left(\frac{k}{4}\right)\epsilon^2} \leq \frac{18\sqrt{2}e^{1/2}}{\sqrt{\pi}} e^{-\left(\frac{k-2}{4}\right)\epsilon^2\left(1-\frac{2}{3}\epsilon\right)}.$$

Proof The proof of this lemma is very similar to the proof of Lemma 10, so we focus on the new details. To prove an upper bound, we start with the bound on $\text{Prob}\left[s < s_0(1-\epsilon)\right]$ from Equation (26). We first observe that

$$f(s_0(1-x)) = f(s_0)(1-x)^{\frac{k-2}{2}} \left(1 + \frac{s_0}{1-s_0}x\right)^{\frac{d-k-2}{2}}. \tag{34}$$

We now bound the logarithm of the second and third factors in Equation (34) using Inequalities (9) and (25) as follows:

$$\log\left((1-x)^{\frac{k-2}{2}} \left(1 + \frac{s_0}{1-s_0}x\right)^{\frac{d-k-2}{2}}\right) \leq \frac{k-2}{2}\left(-x - \frac{x^2}{2}\right) + \frac{d-k-2}{2}\left(\frac{s_0}{1-s_0}x\right) \tag{35}$$

Since $\frac{s_0}{1-s_0} = \frac{k}{d-k}$ and $\epsilon < x < 1$, Inequality (35) further simplifies to

$$\frac{k-2}{2}\left(-x - \frac{x^2}{2}\right) + \frac{d-k-2}{2}\left(\frac{k}{d-k}x\right) \leq x - \left(\frac{k-2}{4}\right)x^2 \leq \frac{3}{2} - \left(\frac{k}{4}\right)x^2.$$

Hence, for $\epsilon \leq x < 1$, we have

$$f(s_0(1-x)) \leq f(s_0)e^{3/2}e^{-\left(\frac{k}{4}\right)x^2} \leq f(s_0)e^{3/2}e^{-\left(\frac{k}{4}\right)\epsilon x}. \tag{36}$$

Substituting Inequality (36) into the integral of Equation (26), we find

$$\begin{aligned} Bs_0 \int_\epsilon^1 f(s_0(1-x))dx &\leq Bs_0 f(s_0)e^{3/2} \int_\epsilon^1 e^{-\left(\frac{k}{4}\right)\epsilon x}dx \\ &\leq Bs_0 f(s_0)e^{3/2} \int_\epsilon^\infty e^{-\left(\frac{k}{4}\right)\epsilon x}dx \leq Bs_0 f(s_0)\frac{4e^{3/2}}{k\epsilon}e^{-\frac{k}{4}\epsilon^2}. \end{aligned} \tag{37}$$

Since $k \geq \epsilon^{-2}$, Inequality (37) can be further simplified to

$$Bs_0 f(s_0)\frac{4e^{3/2}}{\sqrt{k}}e^{-\frac{k}{4}\epsilon^2}$$

Applying the upper bound on $Bs_0 f(s_0)$ from Corollary 8 completes the first inequality of the proof. The final inequality follows from the fact that $k \geq (k-2)\left(1-\frac{2}{3}\epsilon\right)$. ■

This completes the proof all of the conditions in Section 2, and, therefore, completes the proof of our main theorem, Theorem 2.

Acknowledgments

This work was supported by a grant from the Simons Foundation (#282399 to Michael Burr) and National Science Foundation Grants CCF-1527193, CCF-1407623, DMS-1403062, and DMS-1547399.

Appendix A. Uniform Distributions on Unit Spheres in High Dimensions

In this section, we study uniform distributions and surface areas (or hypervolumes) on high-dimensional spheres. More precisely, for any $d \geq 1$, let S^{d-1} denote the unit sphere of dimension $d-1$ and $d\Omega_{d-1}$ denote the surface area measure for S^{d-1}. We show that, for any $1 \leq k \leq d$,

$$d\Omega_{d-1} = \frac{1}{2}f(s)ds\, d\Omega_{k-1}d\Omega_{d-k-1},$$

where $s \in [0,1]$ and $f(s) = s^{\frac{k-2}{2}}(1-s)^{\frac{d-k-2}{2}}$. This is a more precise version of a result in Kane et al. (2011), replacing an unspecified constant by 1/2. We include the proof of this result here in order to make this result more accessible to a larger population of researchers. This formula is of independent interest since it shows that the uniform distribution on S^{d-1} is a product of uniform distributions on S^{k-1} and S^{d-k-1} with a distribution on $[0,1]$.

Theorem 13 *Under the almost bijective map $\Psi : S^{d-1} \to [0,1] \times S^{k-1} \times S^{d-k-1}$, we have equality of the surface area differential forms on S^{d-1}, S^{k-1}, and S^{d-k-1}, i.e.,*

$$d\Omega_{d-1} = \frac{1}{2}f(s)ds\, d\Omega_{k-1}d\Omega_{d-k-1},$$

where $f(s) = s^{(k-2)/2}(1-s)^{(d-k-2)/2}$. Equivalently, in terms of probability distribution measures,

$$\frac{d\Omega_{d-1}}{Vol_{d-1}\left(S^{d-1}\right)} = Bf(s)ds\frac{d\Omega_{k-1}}{Vol_{k-1}\left(S^{k-1}\right)}\frac{d\Omega_{d-k-1}}{Vol_{d-k-1}\left(S^{d-k-1}\right)}.$$

Hence, the uniform distribution on S^{d-1} can be identified with the product distribution on $[0,1] \times S^{k-1} \times S^{d-k-1}$ where the distribution of s on $[0,1]$ has density function $Bf(s)$ and

$$B = \frac{1}{2} \frac{Vol_{k-1}\left(S^{k-1}\right) \; Vol_{d-k-1}\left(S^{d-k-1}\right)}{Vol_{k-1}\left(S^{k-1}\right)} = \frac{\Gamma(\frac{d}{2})}{\Gamma(\frac{k}{2})\Gamma(\frac{d-k}{2})}. \tag{38}$$

This theorem is based on the following lemma, which is well-known to experts, but is included here for completeness.

Lemma 14 *Let $x = (x_1, \ldots, x_d)$ with $x_d > 0$ be a point on the upper hemisphere of S^{d-1}. Then the surface area measure of the unit sphere S^{d-1} at x is*

$$d\Omega_{d-1} = \frac{1}{x_d} dx_1 \ldots dx_{d-1}.$$

Before we begin the proof, we recall the approach for S^2 in 3-dimensional space. We consider the upper hemisphere of S^2 as the graph of a function over D^2, where D^{d-1} denotes $(d-1)$-dimensional disk, namely

$$D^{d-1} = \left\{ \hat{x} \in \mathbb{R}^{d-1} : \sum_{i=1}^{d-1} \hat{x}_i^2 \leq 1 \right\}.$$

We then integrate over the disk D^2 to calculate the surface area of S^2. In particular, the integrand is the limit of the ratios of the area of a square in D^2 to the area of the corresponding parallelogram above the square in the tangent space of S^2 as the square shrinks a point. In the case of the sphere, the parallelogram's area is calculated using the cross product, but we must replace the use of the cross product in higher dimensions.

Proof In d-dimensional space, we consider the upper hemisphere of S^{d-1} as the graph of a function over the $(d-1)$-dimensional disk D^{d-1}. We construct a pair of $(d-1)$-dimensional parallelepipeds as follows: P^{d-1} is in the tangent space of D^{d-1} and Q^{d-1} is in the tangent space of S^{d-1}. Then, we take the limit of their $(d-1)$-dimensional volumes as P^{d-1} approaches a point. Due to complications in taking the $(d-1)$-dimensional volume in d-dimensional space, we extend both P^{d-1} and Q^{d-1} to associated, full-dimensional parallelepipeds.

Let $(\hat{x}_1, \ldots, \hat{x}_{d-1}) \in D^{d-1}$ and define $\phi : D^{d-1} \to \mathbb{R}_{\geq 0}$ as

$$\phi(\hat{x}_1, \ldots, \hat{x}_{d-1}) = \sqrt{1 - \sum_{i=1}^{d-1} \hat{x}_i^2}.$$

We observe that the graph of this function is the upper hemisphere of S^{d-1}. We now extend this map to $D^{d-1} \times \mathbb{R}$ as $\Phi_d : D^{d-1} \times \mathbb{R} \to \mathbb{R}^d$ defined by

$$(\hat{x}_1, \ldots, \hat{x}_{d-1}, \hat{x}_d) \mapsto ((1+\hat{x}_d)\hat{x}_1, (1+\hat{x}_d)\hat{x}_2, \ldots, (1+\hat{x}_d)\hat{x}_{d-1}, (1+\hat{x}_d)\phi(\hat{x}_1, \ldots, \hat{x}_{d-1})).$$

We observe that $\Phi_d|_{D^{d-1}\times\{0\}}$ maps the disk $D^{d-1} \times \{0\}$ surjectively onto the graph of ϕ, i.e., the upper hemisphere of S^{d-1}, see Figure 2.

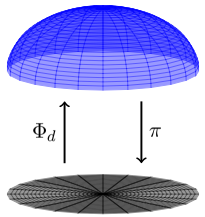


Figure 2: Φ_d maps the disk $D^{d-1} \times \{0\}$ surjectively onto the upper hemisphere of the $(d-1)$-dimensional unit sphere S^{d-1}. We observe that $(\Phi_d)^{-1} = \pi$ is the projection map onto the first $d-1$ coordinates.

We recall that, at $(\hat{x}_1, \ldots, \hat{x}_{d-1}) \in D^{d-1}$, the tangent space is $\mathbb{R}^{d-1}$ and we define the parallelepiped P^{d-1} in the tangent space by the vectors $\Delta\hat{x}_i e_i$ of length $\Delta\hat{x}_i$ in the direction of the i^{th} standard basis vector e_i of $\mathbb{R}^{d-1}$. Similarly, the tangent space at $(\hat{x}_1, \ldots, \hat{x}_{d-1}, 0) \in D^{d-1} \times \mathbb{R}$ is $\mathbb{R}^d$, and we define the parallelepiped P^d in this tangent space by the vectors $\Delta\hat{x}_i e_i$ for $1 \leq i \leq d-1$ and he_d for the final direction. Then, as the vector he_d is perpendicular to the tangent vectors of the disk D^{d-1}, the d-dimensional volume of P^d can be computed in terms of the $(d-1)$-dimensional volume of P^{d-1} and the height h, i.e.,

$$\text{Vol}_d\left(P^d\right) = h\text{Vol}_{d-1}\left(P^{d-1}\right).$$

Next, we let Q^{d-1} and Q^d be the images of P^{d-1} and P^d under the Jacobian of Φ_d, i.e., $\text{Jac}\,\Phi_d$, respectively. Note that the Jacobian of Φ_d, when restricted to $D^{d-1} \times \{0\}$, is

$$(\text{Jac}\,\Phi_d)\,|_{D^{d-1}\times\{0\}} = \left[\begin{array}{ccc|c} & & & \hat{x}_1 \\ & I & & \vdots \\ & & & \hat{x}_{d-1} \\ \hline -\frac{\hat{x}_1}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})} & \cdots & -\frac{\hat{x}_{d-1}}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})} & \phi(\hat{x}_1, \ldots, \hat{x}_{d-1}) \end{array}\right].$$

Since the Jacobian acts on tangent vectors, Q^{d-1} is defined by the vectors

$$\Delta\hat{x}_i \left(f_i - \frac{\hat{x}_i}{\phi(\hat{x}_1, \ldots, \hat{x}_{d-1})} f_d \right),$$

for $1 \leq i \leq d-1$, where the f_i is the i^{th} standard basis vector of $\mathbb{R}^d$. Moreover, Q^d is defined by these vectors as well as the image of he_d, i.e.,

$$h\left(\hat{x}_1, \ldots, \hat{x}_{d-1}, \phi(\hat{x}_1, \ldots, \hat{x}_{d-1})\right).$$

We observe that the vectors $\Delta\hat{x}_i \left(f_i - \frac{\hat{x}_i}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})} f_d\right)$ for $1 \le i \le d-1$ are tangent vectors to the sphere S^{d-1}, and that $h\left(\hat{x}_1,\ldots,\hat{x}_{d-1},\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})\right)$ is the outward pointing surface normal with length h. Since the tangent vectors are perpendicular to the outward pointing normal, the volumes of Q^{d-1} and Q^d have a similar relationship as the volumes of P^{d-1} and P^d, i.e.,

$$\mathrm{Vol}_d\left(Q^d\right) = h\mathrm{Vol}_{d-1}\left(Q^{d-1}\right).$$

Therefore, the ratio between the d-dimensional volumes of Q^d and P^d is the same as the ratio of the $(d-1)$-dimensional volumes of Q^{d-1} and P^{d-1}.

Since Q^d is the image of P^d under the linear map $\mathrm{Jac}\ \Phi_d\big|_{D^{d-1}\times\{0\}}$, it follows that

$$\mathrm{Vol}_d\left(Q^d\right) = \left|\det\left(\mathrm{Jac}\,\Phi_d\right)\big|_{(\hat{x}_1,\ldots,\hat{x}_{d-1},0)}\right| \mathrm{Vol}_d\left(P^d\right).$$

Therefore, the ratio of the volumes of Q^{d-1} and P^{d-1} is $\left|\det\left(\mathrm{Jac}\,\Phi_d\right)\big|_{(\hat{x}_1,\ldots,\hat{x}_{d-1},0)}\right|$. It is straight-forward to compute the determinant of $\mathrm{Jac}(\Phi_d)$ at the point $(\hat{x}_1,\ldots,\hat{x}_{d-1},0)$ via a few row reductions to eliminate the first $d-1$ entries in the last row and turn the matrix into an upper triangular matrix whose lower right corner is $\frac{1}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})}$. Hence, the determinant of $\mathrm{Jac}(\Phi_d)$ is $\frac{1}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})}$, which is the desired scaling factor.

Therefore, $\frac{1}{\phi(\hat{x}_1,\ldots,\hat{x}_{d-1})}$ is the local factor in the stretching of the surface area in the map from D^{d-1} to S^{d-1}. We recall that the coordinates $x_1,\ldots,x_d$ are the coordinates on the upper hemisphere and $\hat{x}_1,\ldots,\hat{x}_{d-1}$ are the coordinates on D^{d-1}. Since, under the map $\Phi_d|_{D^{d-1}\times\{0\}}$, $x_i = \hat{x}_i$ for $1 \le i \le d-1$, it follows that

$$d\hat{x}_1\ldots d\hat{x}_{d-1} = dx_1\ldots dx_{d-1} \qquad \text{and} \qquad \phi(\hat{x}_1,\ldots,\hat{x}_{d-1}) = x_d.$$

From here, the result follows directly. ■

Proof of Theorem 13 Let $(w_1,\ldots,w_d)$, $(x_1,\ldots,x_k)$, and $(y_1,\ldots,y_{d-k})$ be coordinates of points on the $(d-1)$-dimensional unit sphere, the $(k-1)$-dimensional unit sphere, and the $(d-k-1)$-dimensional unit sphere, respectively. Let $(\hat{w}_1,\ldots,\hat{w}_{d-1})$, $(\hat{x}_1,\ldots,\hat{x}_{k-1})$, and $(\hat{y}_1,\ldots,\hat{y}_{d-k-1})$ be the coordinates of points on the disks D^{d-1}, D^{k-1} and D^{d-k-1}, respectively. Let

$$\varphi : [0,1]\times D^{k-1}\times D^{d-k-1} \to D^{d-1}$$

be defined by

$$s\times(\hat{x}_1,\ldots,\hat{x}_{k-1})\times(\hat{y}_1,\ldots,\hat{y}_{d-k-1}) \mapsto \left(\sqrt{s}\hat{x}_1,\ldots,\sqrt{s}\hat{x}_{k-1},\sqrt{s}\sqrt{1-\sum_{i=1}^{k-1}\hat{x}_i^2},\sqrt{1-s}\hat{y}_1,\ldots,\sqrt{1-s}\hat{y}_{d-k-1}\right).$$

We observe that φ maps the disks D^{k-1} and D^{d-k-1} onto the half of the disk D^{d-1} whose k^{th} coordinate is nonnegative. As the measure of the image is the measure of the preimage scaled by the determinant of the Jacobian of φ, the surface area measure of the disk D^{d-1} is

$$d\hat{w}_1\ldots d\hat{w}_{d-1} = |\det\mathrm{Jac}(\varphi)|ds\,d\hat{x}_1\ldots d\hat{x}_{k-1}d\hat{y}_1\ldots d\hat{y}_{d-k-1}. \tag{39}$$

The Jacobian of φ for $s \in (0,1)$ is

$$\mathrm{Jac}\ \varphi = \left[\begin{array}{c|ccc|ccc} \frac{\hat{x}_1}{2\sqrt{s}} & \sqrt{s} & \ldots & 0 & 0 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{\hat{x}_{k-1}}{2\sqrt{s}} & 0 & \ldots & \sqrt{s} & 0 & \ldots & 0 \\ \hline \frac{\sqrt{1-\sum_{i=1}^{k-1}\hat{x}_i^2}}{2\sqrt{s}} & \frac{-\sqrt{s}\hat{x}_1}{\sqrt{1-\sum_{i=1}^{k-1}\hat{x}_i^2}} & \ldots & \frac{-\sqrt{s}\hat{x}_{k-1}}{\sqrt{1-\sum_{i=1}^{k-1}\hat{x}_i^2}} & 0 & \ldots & 0 \\ \hline \frac{-\hat{y}_1}{2\sqrt{1-s}} & 0 & \ldots & 0 & \sqrt{1-s} & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \frac{-\hat{y}_{d-k-1}}{2\sqrt{1-s}} & 0 & \ldots & 0 & 0 & \ldots & \sqrt{1-s} \end{array}\right].$$

Eliminating all but the k^{th} entry of the first column by adding multiples of the other columns to the first column, we obtain

$$\det\mathrm{Jac}(\varphi) = \frac{1}{2\hat{x}_k} s^{\frac{k-2}{2}}(1-s)^{\frac{d-k-1}{2}}.$$

Substituting this value into Expression (39), we have the surface area measure of D^{d-1} in terms of the disks D^{k-1} and D^{d-k-1}. That is,

$$d\hat{w}_1\ldots d\hat{w}_{d-1} = \frac{1}{2\hat{x}_k} s^{(k-2)/2}(1-s)^{(d-k-1)/2} ds\,d\hat{x}_1\ldots d\hat{x}_{k-1}d\hat{y}_1\ldots d\hat{y}_{d-k-1}. \tag{40}$$

We observe that the coordinates of the disk D^{t-1} correspond to the first $t-1$ entries of coordinates of the unit sphere S^{t-1}. Therefore, we may extend φ to the map Ψ, as defined above, where

$$\Psi^{-1} = \Phi_d \circ \varphi \circ (id_s \times (\Phi_k)^{-1} \times (\Phi_{d-k})^{-1}).$$

Employing the results of Lemma 14 in various dimensions, we rewrite the surface measure of a unit sphere in terms of the surface measure of the corresponding disks:

$$\begin{aligned} d\hat{x}_1\ldots d\hat{x}_{k-1} &= dx_1\ldots dx_{k-1} = x_k d\Omega_{k-1} \\ d\hat{y}_1\ldots d\hat{y}_{d-k-1} &= dy_1\ldots dy_{d-k-1} = y_{d-k}d\Omega_{d-k-1} \\ d\hat{w}_1\ldots d\hat{w}_{d-1} &= dw_1\ldots dw_{d-1} = w_d d\Omega_{d-1}. \end{aligned}$$

By applying the Ψ, we can substitute these three equalities into Equation (40) to obtain

$$\begin{aligned} d\Omega_{d-1} &= \frac{1}{w_d}dw_1\ldots dw_{d-1} = \frac{y_{d-k}}{2w_d} s^{(k-2)/2}(1-s)^{(d-k-1)/2} ds\,d\Omega_{k-1}d\Omega_{d-k-1} \\ &= \frac{1}{2}s^{(k-2)/2}(1-s)^{(d-k-2)/2} ds\,d\Omega_{k-1}d\Omega_{d-k-1} = \frac{1}{2}f(s)ds\,d\Omega_{k-1}d\Omega_{d-k-1}, \end{aligned}$$

where the third equality follows from the fact that $w_d = \sqrt{1-s}y_{d-k}$ by the map Ψ. Since the cases where $s = 0$ or $s = 1$ have measure 0, the result follows. ■

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Nir Ailon and Bernard Chazelle. Approximate nearest neighbors and the fast Johnson-Lindenstrauss transform. In *Proceedings of the 38th ACM Symposium on Theory of Computing (STOC)*, pages 557–563, 2006.

Noga Alon. Problems and results in extremal combinatorics. *Discrete Mathematics*, 273: 31–53, 2003.

Rosa I. Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine Learning*, 63:161–182, 2006.

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Knowledge Discovery and Data Mining*, pages 245–250, 2001.

Emmanuel J. Candès. The restricted isometry property and its implications for compressed sensing. *Comptes Rendus Mathématique. Académie des Sciences. Paris*, 346:589–592, 2008.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the 45th ACM Symposium on Theory of Computing (STOC)*, pages 81–90, 2013.

Graham Cormode and Piotr Indyk. Stable distributions in streaming computation. In Garofalakis M., Gehrke J., and Rastoi R., editors, *Data Stream Management*. Springer, Berlin, Heidelberg, 2016.

Anirban Dasgupta, Ravi Kumar, and Tamás Sarlós. A sparse Johnson-Lindenstrauss transform. In *Proceedings of the 42nd ACM Symposium on Theory of Computing (STOC)*, pages 341–350, 2010.

Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structure and Algorithms*, 22(1):60–65, 2003.

Peter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory Series B*, 44(3):355–362, 1988.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the 30th ACM Symposium on Theory of Computing (STOC)*, pages 604–613, 1998.

T. S. Jayram and David P. Woodruff. Optimal bounds for Johnson-Lindenstrauss transforms and streaming problems with subconstant error. *ACM Transactions on Algorithms*, 9(26), 2013.

William B. Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss transforms. *Journal of the ACM*, 61(1), 2014.

Daniel M. Kane, Raghu Meka, and Jelani Nelson. Almost optimal explicit Johnson-Lindenstrauss families. In *Proceedings of the 15th International Workshop on Randomization and Computation (RANDOM)*, pages 628–639, 2011.

Jessica Lin and Dimitrios Gunopulos. Dimensionality reduction by random projection and latent semantic indexing. In *Proceedings of the Text Mining Workshop, at the 3rd SIAM International Conference on Data Mining*, May 2003.

Jiri Matousek. On variants of the Johnson-Lindenstrauss lemma. *Random Structure and Algorithms*, 33(2):142–156, 2008.

Nam H. Nguyen, Thong T. Do, and Trac D. Tran. A fast and efficient algorithm for low-rank approximation of a matrix. In *Proceedings of the 41st ACM Symposium on Theory of Computing (STOC)*, 2009.

Herbert Robbins. A remark on Stirling formula. *American Mathematical Monthly*, 62:26–29, 1955.

Shashanka Ubaru, Arya Mazumdar, and Yousef Saad. Low rank approximation and decomposition of large matrices using error correcting codes. *IEEE Transactions on Information Theory*, 63(9):5544–5558, 2017.

Kilian Q. Weinberger, Anirban Dasgupta, John Langford, Alexander J. Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, volume 1113-1120, 2009.

Journal of Machine Learning Research 19 (2018) 1-37 Submitted 1/15; Revised 4/16; Published 12/18

An efficient distributed learning algorithm based on effective local functional approximations

Dhruv Mahajan DHRUVM@FB.COM
Facebook AI
Menlo Park, CA 94025, USA

Nikunj Agrawal NIKUNJ157@GMAIL.COM
Indian Institute of Technology
Dept. of Computer Science & Engineering
Kanpur, India

S. Sathiya Keerthi KEERTHI@MICROSOFT.COM
Office Data Science Group
Microsoft
Mountain View, CA 94043, USA

Sundararajan Sellamanickam SSRAJAN@MICROSOFT.COM
Microsoft Research
Bangalore, India

Léon Bottou LEON@BOTTOU.ORG
Facebook AI Research
New York, NY, USA

Editor: Inderjit Dhillon

Abstract

Scalable machine learning over big data is an important problem that is receiving a lot of attention in recent years. On popular distributed environments such as Hadoop running on a cluster of commodity machines, communication costs are substantial and algorithms need to be designed suitably considering those costs. In this paper we give a novel approach to the distributed training of linear classifiers (involving smooth losses and L_2 regularization) that is designed to reduce the total communication costs. At each iteration, the nodes minimize locally formed approximate objective functions; then the resulting minimizers are combined to form a descent direction to move. Our approach gives a lot of freedom in the formation of the approximate objective function as well as in the choice of methods to solve them. The method is shown to have $O(\log(1/\epsilon))$ time convergence. The method can be viewed as an iterative parameter mixing method. A special instantiation yields a parallel stochastic gradient descent method with strong convergence. When communication times between nodes are large, our method is much faster than the Terascale method (Agarwal et al., 2011), which is a state of the art distributed solver based on the statistical query model (Chu et al., 2006) that computes function and gradient values in a distributed fashion. We also evaluate against other recent distributed methods and demonstrate superior performance of our method.

Keywords: Distributed learning, Example partitioning, L_2 regularization

1. Introduction

In recent years, machine learning over Big Data has become an important problem, not only in web related applications, but also more commonly in other applications, e.g., in the data mining over huge amounts of user logs. The data in such applications are usually collected and stored in a decentralized fashion over a cluster of commodity machines (nodes) where communication times between nodes are significantly large. Examples of applications involving high communication costs include: Click prediction on advertisement data where the number of examples is huge and features are words which can run in to billions, and Geo distributed data across different countries/continents where cross data-center speeds are extremely slow. In such a settings, it is natural for the examples to be partitioned over the nodes.

Distributed machine learning algorithms that operate on such data are usually iterative. Each iteration involves some *computation* that happens locally in each node. In each iteration, there is also *communication* of information between nodes and this is special to the distributed nature of solution. Distributed systems such as those based on the Map-Reduce framework (Dean and Ghemawat, 2008) involve additional special operations per iteration, such as the loading of data from disk to RAM. Recent frameworks such as Spark (Zaharia et al., 2010) and REEF (Weimer et al., 2015) avoid such unnecessary repeated loading of data from disk. Still, communication between nodes in each iteration is unavoidable and, its cost can be substantial when working with Big Data. Therefore, the development of efficient distributed machine learning algorithms that minimize communication between nodes is an important problem. The key is to come up with algorithms that minimize the number of iterations.

In this paper we consider the distributed batch training of linear classifiers in which: (a) both, the number of examples and the number of features are large; (b) the data matrix is sparse; (c) the examples are partitioned over the nodes; (d) the loss function is convex and differentiable; and, (e) the L_2 regularizer is employed. This problem involves the large scale unconstrained minimization of a convex, differentiable objective function $f(w)$ where w is the weight vector. The minimization is usually performed using an iterative descent method in which an iteration starts from a point w^r, computes a direction d^r that satisfies

$$\textbf{sufficient angle of descent: } \angle(-g^r, d^r) \leq \theta \tag{1}$$

where $g^r = g(w^r)$, $g(w) = \nabla f(w)$, $\angle(a, b)$ is the angle between vectors a and b, and $0 \leq \theta < \pi/2$, and then performs a line search along the direction d^r to find the next point, $w^{r+1} = w^r + td^r$. Let $w^\star = \arg\min_w f(w)$. A key side contribution of this paper is the proof that, when f is convex and satisfies some additional weak assumptions, the method has global linear rate of convergence (*glrc*)[1] and so it finds a point w^r satisfying $f(w^r) - f(w^\star) \leq \epsilon$ in $O(log(1/\epsilon))$ iterations. *The main theme of this paper is that the flexibility offered by this method with strong convergence properties allows us to build a class of useful distributed learning methods with good computation and communication trade-off capabilities.*

Take one of the most effective distributed methods, viz., SQM (Statistical Query Model) (Chu et al., 2006; Agarwal et al., 2011), which is a batch, gradient-based descent method. The gradient is computed in a distributed way with each node computing the gradient

1. We say a method has *glrc* if $\exists\ 0 < \delta < 1$ such that $(f(w^{r+1}) - f(w^\star)) \leq \delta(f(w^r) - f(w^\star))\ \forall r$.

component corresponding to its set of examples. This is followed by an aggregation of the components. We are interested in systems in which the communication time between nodes is large relative to the computation time in each node.[2] For iterative algorithms such as SQM, the total training time is given by

$$\text{Training time} = (T^{cmp} + T^{com})\, T^{iter} \tag{2}$$

where T^{cmp} and T^{com} are respectively, the computation time and the communication time per iteration and T^{iter} is the total number of iterations. When T^{com} is large, it is not optimal to work with an algorithm such as SQM that has T^{cmp} small and due to which, T^{iter} is large. In such a scenario, it is useful to ask: **Q1.** *In each iteration, can we do more computation in each node so that the number of iterations and hence the number of communication passes are decreased, thus reducing the total computing time?*

There have been some efforts in the literature to reduce the amount of communication. In one class of such methods, the current w^r is first passed on to all the nodes. Then, each node p forms an approximation $\tilde{f}_p$ of f using only its examples, followed by several optimization iterations (local passes over its examples) to decrease $\tilde{f}_p$ and reach a point w_p. The w_p $\forall p$ are averaged to form the next iterate w^{r+1}. One can stop after just one major iteration (going from $r = 0$ to $r = 1$); such a method is referred to as *parameter mixing (PM)* (Mann et al., 2009). Alternatively, one can do many major iterations; such a method is referred to as *iterative parameter mixing (IPM)* (Hall et al., 2010). Convergence theory for such methods is inadequate (Mann et al., 2009; McDonald et al., 2010), which prompts us to ask: **Q2.** *Is it possible to devise an IPM method that produces* $\{w^r\} \to w^\star$*?*

In another class of methods, the dual problem is solved in a distributed fashion (Pechyony et al., 2011; Yang, 2013; Yang et al., 2013; Jaggi et al., 2014). Let α_p denote the dual vector associated with the examples in node p. The basic idea is to optimize $\{\alpha_p\}$ in parallel and then use a combination of the individual directions thus generated to take an overall step. In practice these methods tend to have slow convergence; see Section 4 for details.

We make a novel and simple use of the iterative descent method mentioned at the beginning of this section to design a distributed algorithm that answers Q1-Q2 positively. The main idea is to use distributed computation for generating a good search direction d^r and not just for forming the gradient as in SQM. At iteration r, let us say each node p has the current iterate w^r and the gradient g^r. This information can be used together with the examples in the node to form a function $\hat{f}_p(\cdot)$ that approximates $f(\cdot)$ and satisfies $\nabla \hat{f}_p(w^r) = g^r$. One simple and effective suggestion is:

$$\hat{f}_p(w) = f_p(w) + (g^r - \nabla f_p(w^r)) \cdot (w - w^r) \tag{3}$$

where f_p is the part of f that does not depend on examples outside node p; the second term in (3) can be viewed as an approximation of the objective function part associated with data from the other nodes. In Section 3 we give other suggestions for forming $\hat{f}_p$. Now $\hat{f}_p$ can be optimized within node p using any method $\mathcal{M}$ which has *glrc*, e.g., Trust region method, L-BFGS, etc. There is no need to optimize $\hat{f}_p$ fully. We show (see Section 3) that, in a constant number of local passes over examples in node p, an approximate minimizer

2. This is the case when feature dimension is huge. Many applications gain performance when the feature space is expanded, say, via feature combinations, explicit expansion of nonlinear kernels etc.

w_p of $\hat{f}_p$ can be found such that the direction $d_p = w_p - w^r$ satisfies the sufficient angle of descent condition, (1). A convex combination of the set of directions generated in the nodes, $\{d_p\}$ forms the overall direction d^r for iteration r. Note that d^r also satisfies (1). The result is an overall distributed method that finds a point w satisfying $f(w) - f(w^\star) \leq \epsilon$ in $O(\log(1/\epsilon))$ time. This answers **Q2**.

The method also reduces the number of communication passes over the examples compared with SQM, thus also answering **Q1**. The intuition here is that, if each $\hat{f}_p$ is a good approximation of f, then d^r will be a good global direction for minimizing f at w^r, and so the method will move towards $w^\star$ much faster than SQM.

In summary, the paper makes the following contributions. First, for convex f we establish *glrc* for a general iterative descent method. Second, and more important, we propose a distributed learning algorithm that: (a) converges in $O(\log(1/\epsilon))$ time, thus leading to an IPM method with strong convergence; (b) is more efficient than SQM when communication costs are high; and (c) flexible in terms of the local optimization method $\mathcal{M}$ that can be used in the nodes.

There is also another interesting side contribution associated with our method. It is known that example-wise methods such as stochastic gradient descent (SGD) are inherently sequential and hard to parallelize (Zinkevich et al., 2010). By employing SGD as $\mathcal{M}$, the local optimizer for $\tilde{f}_p$ in our method, we obtain a parallel SGD method with good performance as well as strong convergence properties. This contribution is covered in detail in Mahajan et al. (2013b); we give a summarized view in Subsection 3.5.

Experiments (Section 4) validate our theory as well as show the benefits of our method for large dimensional datasets where communication is the bottleneck. We give a discussion on unexplored possibilities for extending our distributed learning method in Section 5 and conclude the paper in Section 6.

2. Basic descent method

Let $f \in \mathcal{C}^1$, the class of continuously differentiable functions[3], f be convex, and the gradient g satisfy the following assumptions.

A1. g is Lipschitz continuous, i.e., $\exists\ L > 0$ such that $\|g(w) - g(\tilde{w})\| \leq L\|w - \tilde{w}\| \quad \forall\ w, \tilde{w}$.

A2. $\exists\ \sigma > 0$ such that $(g(w) - g(\tilde{w})) \cdot (w - \tilde{w}) \geq \sigma\|w - \tilde{w}\|^2 \quad \forall\ w, \tilde{w}$.

A1 and A2 are essentially second order conditions: if f happens to be twice continuously differentiable, then L and σ can be viewed as upper and lower bounds on the eigenvalues of the Hessian of f. A convex function f is said to be σ- strongly convex if $f(w) - \frac{\sigma}{2}\|w\|^2$ is convex. In machine learning, all convex risk functionals in $\mathcal{C}^1$ having the L_2 regularization term, $\frac{\lambda}{2}\|w\|^2$ are σ- strongly convex with $\sigma = \lambda$. It can be shown (Smola and Vishwanathan, 2008) that, if f is σ-strongly convex, then f satisfies assumption A2.

Let $f^r = f(w^r)$, $g^r = g(w^r)$ and $w^{r+1} = w^r + td^r$. Consider the following standard line search conditions.

$$\textbf{Armijo:} \quad f^{r+1} \leq f^r + \alpha g^r \cdot (w^{r+1} - w^r) \tag{4}$$

$$\textbf{Wolfe:} \quad g^{r+1} \cdot d^r \geq \beta g^r \cdot d^r \tag{5}$$

3. It would be interesting future work to extend all the theory developed in this paper to non-differentiable convex functions, using sub-gradients.

where $0 < \alpha < \beta < 1$.

Algorithm 1: Descent method for f

Choose w^0;
for $r = 0, 1 \ldots$ **do**
1. Exit if $g^r = 0$;
2. Choose a direction d^r satisfying (1);
3. Do line search to choose $t > 0$ so that $w^{r+1} = w^r + td^r$ satisfies the Armijo-Wolfe conditions (4) and (5);

end

Let us now consider the general descent method in Algorithm 1 for minimizing f. The following result shows that the algorithm is well-posed. A proof is given in the appendix B.

Lemma 1. Suppose $g^r \cdot d^r < 0$. Then $\{t : (4) \text{ and } (5) \text{ hold for } w^{r+1} = w^r + td^r\} = [t_\beta, t_\alpha]$, where $0 < t_\beta < t_\alpha$, and t_β, t_α are the unique roots of

$$g(w^r + t_\beta d^r) \cdot d^r = \beta g^r \cdot d^r, \tag{6}$$

$$f(w^r + t_\alpha d^r) = f^r + t_\alpha \alpha g^r \cdot d^r, \;\; t_\alpha > 0. \tag{7}$$

Theorem 2. Let $w^\star = \arg\min_w f(w)$ and $f^\star = f(w^\star)$.[4] Then $\{w^r\} \to w^\star$. Also, we have *glrc*, i.e., $\exists\ \delta$ satisfying $0 < \delta < 1$ such that $(f^{r+1} - f^\star) \leq \delta\,(f^r - f^\star)\ \forall\ r \geq 0$, and, $f^r - f^\star \leq \epsilon$ is reached after at most $\frac{\log((f^0 - f^\star)/\epsilon)}{\log(1/\delta)}$ iterations. An upper bound on δ is $(1 - 2\alpha(1-\beta)\frac{\sigma^2}{L^2}\cos^2\theta)$.

A proof of Theorem 2 is given in the appendix B. If one is interested only in proving convergence, it is easy to establish under the assumptions made; such theory goes back to the classical works of Wolfe (Wolfe, 1969, 1971). But proving *glrc* is harder. There exist proofs for special cases such as the gradient descent method (Boyd and Vandenberghe, 2004). The *glrc* result in Wang and Lin (2013) is only applicable to descent methods that are "close" (see equations (7) and (8) in Wang and Lin (2013)) to the gradient descent method. Though Theorem 2 is not entirely surprising, as far as we know, such a result does not exist in the literature.

Remark 1. It is important to note that the rate of convergence indicated by the upper bound on δ given in Theorem 2 is pessimistic since it is given for a very general descent algorithm that includes plain batch gradient descent which is known to have a slow rate of convergence. *Therefore, it should not be misconstrued that the indicated slow convergence rate would hold for all methods falling into the class of methods covered here.* Depending on the method used for choosing d^r, the actual rate of convergence can be a lot better. For example, we observe very good rates for our distributed method, FADL; see the empirical results in Section 4 and also the comments made in Remark 2 of Section 3.

3. Distributed training

In this section we discuss full details of our distributed training algorithm. Let $\{x_i, y_i\}$ be the training set associated with a binary classification problem ($y_i \in \{1, -1\}$). Consider a

4. Assumption A2 implies that $w^\star$ is unique.

linear classification model, $y = \text{sgn}(w^T x)$. Let $l(w \cdot x_i, y_i)$ be a continuously differentiable loss function that has Lipschitz continuous gradient. This allows us to consider loss functions such as least squares ($l(s, y) = (s - y)^2$), logistic loss ($l(s, y) = \log(1 + \exp(-sy))$ and squared hinge loss ($l(s, y) = \max\{0, 1 - ys\}^2$). Hinge loss is not covered by our theory since it is non-differentiable.

Suppose the training examples are distributed in P nodes. Let: I_p be the set of indices i such that (x_i, y_i) sits in the p-th node; $L_p(w) = \sum_{i \in I_p} l(w; x_i, y_i)$ be the total loss associated with node p; and, $L(w) = \sum_p L_p(w)$ be the total loss over all nodes. Our aim is to minimize the regularized risk functional $f(w)$ given by

$$f(w) = \frac{\lambda}{2}\|w\|^2 + L(w) = \frac{\lambda}{2}\|w\|^2 + \sum_p L_p(w), \tag{8}$$

where $\lambda > 0$ is the regularization constant. It is easy to check that $g = \nabla f$ is Lipschitz continuous.

3.1. Our approach

Our distributed method is based on the descent method in Algorithm 1. We use a master-slave architecture.[5] Let the examples be partitioned over P slave nodes. Distributed computing is used to compute the gradient g^r as well as the direction d^r. In the r-th iteration, let us say that the master has the current w^r and gradient g^r. One can communicate these to all P (slave) nodes. The direction d^r is formed as follows. Each node p constructs an approximation of $f(w)$ using only information that is available in that node, call it $\hat{f}_p(w)$, and (approximately) optimizes it (starting from w^r) to get the point w_p. Let $d_p = w_p - w^r$. Then d^r is chosen to be any convex combination of $d_p\ \forall p$. Doing line search along the d^r direction completes the r-th iteration. Line search involves distributed computation, but it is inexpensive; we give details in Subsection 3.4.

We want to point out that $\hat{f}_p$ can change with r, i.e., one is allowed to use a different $\hat{f}_p$ in each outer iteration. We just don't mention it as $\hat{f}_p^r$ to avoid clumsiness of notation. In fact, all the choices for $\hat{f}_p$ that we discuss below in Subsection 3.2 are such that $\hat{f}_p$ depends on the current iterate, w^r.

3.2. Choosing $\hat{f}_p$

Our method offers great flexibility in choosing $\hat{f}_p$ and the method used to optimize it. We only require $\hat{f}_p$ to satisfy the following.

A3. $\hat{f}_p$ is σ-strongly convex, has Lipschitz continuous gradient and satisfies *gradient consistency at* w^r: $\nabla \hat{f}_p(w^r) = g^r$.

Below we give several ways of forming $\hat{f}_p$. The σ-strongly convex condition is easily taken care of by making sure that the L_2 regularizer is always a part of $\hat{f}_p$. This condition implies that

$$\hat{f}_p(w_p) \geq \hat{f}_p(w^r) + \nabla \hat{f}_p(w^r) \cdot (w_p - w^r) + \frac{\sigma}{2}\|w_p - w^r\|^2. \tag{9}$$

5. An *AllReduce* arrangement of nodes (Agarwal et al., 2011) may also be used.

The gradient consistency condition is motivated by the need to satisfy the angle condition (1). Since w_p is obtained by starting from w^r and optimizing $\hat{f}_p$, it is reasonable to assume that $\hat{f}_p(w_p) < \hat{f}_p(w^r)$. Using these in (9) gives $-g^r \cdot d_p > 0$. Since d^r is a convex combination of the d_p it follows that $-g^r \cdot d^r > 0$. Later we will formalize this to yield (1) precisely.

A general way of choosing the approximating functional $\hat{f}_p$ is

$$\hat{f}_p(w) = \frac{\lambda}{2}\|w\|^2 + \tilde{L}_p(w) + \hat{L}_p(w), \tag{10}$$

where $\tilde{L}_p$ is an approximation of L_p and $\hat{L}_p(w)$ is an approximation of $L(w) - L_p(w) = \sum_{q \neq p} L_q(w)$. A natural choice for $\tilde{L}_p$ is L_p itself since it uses only the examples within node p; but there are other possibilities too. To maintain communication efficiency, we would like to design an $\hat{L}_p$ such that it does not explicitly require any examples outside node p. To satisfy A3 we need $\hat{L}_p$ to have Lipschitz continuous gradient. Also, to aid in satisfying gradient consistency, appropriate linear terms are added. We now suggest five choices for $\hat{f}_p$.

Linear Approximation. Set $\tilde{L}_p = L_p$ and choose $\hat{L}_p$ based on the first order Taylor series. Thus,

$$\tilde{L}_p(w) = L_p(w), \qquad \hat{L}_p(w) = (\nabla L(w^r) - \nabla L_p(w^r)) \cdot (w - w^r). \tag{11}$$

(The zeroth order term needed to get $f(w^r) = \hat{f}(w^r)$ is omitted everywhere because it is a constant that plays no role in the optimization.) Note that $\nabla L(w^r) = g^r - \lambda w^r$ and so it is locally computable in node p; this comment also holds for the methods below.

Hybrid approximation. This is an improvement over the linear approximation where we add a quadratic term to $\hat{L}_p$. Since the loss term in (8) is total loss (and not averaged loss), this is done by using $(P-1)$ copies of the quadratic term of L_p to approximate the quadratic term of $\sum_{q \neq p} L_q$.

$$\tilde{L}_p(w) = L_p(w), \tag{12}$$

$$\hat{L}_p(w) = (\nabla L(w^r) - \nabla L_p(w^r)) \cdot (w - w^r) + \frac{P-1}{2}(w - w^r)^T H_p^r (w - w^r), \tag{13}$$

where H_p^r is the Hessian of L_p at w^r. This corresponds to using subsampling to approximate the Hessian of $L(w) - L_p(w)$ at w^r utilizing only the local examples. Subsampling based Hessian approximation is known to be very effective in optimization for machine learning (Byrd et al., 2012).

Quadratic approximation. This is a pure quadratic variant where a second order approximation is used for $\tilde{L}_p$ too.

$$\tilde{L}_p(w) = \nabla L_p(w^r) \cdot (w - w^r) + \frac{1}{2}(w - w^r)^T H_p^r (w - w^r), \tag{14}$$

$$\hat{L}_p(w) = (\nabla L(w^r) - \nabla L_p(w^r)) \cdot (w - w^r) + \frac{P-1}{2}(w - w^r)^T H_p^r (w - w^r). \tag{15}$$

The comment made earlier on the goodness of subsampling based Hessian for the Hybrid approximation applies here too.

Nonlinear approximation. Here the idea is to use $P-1$ copies of L_p to approximate $\sum_{q \neq p} L_q$.

$$\tilde{L}_p(w) = L_p(w), \tag{16}$$

$$\hat{L}_p(w) = (\nabla L(w^r) - P \nabla L_p(w^r)) \cdot (w - w^r) + (P-1) L_p(w). \tag{17}$$

A somewhat similar approximation is used in Sharir et al. (2014). But the main algorithm where it is used does not have deterministic monotone descent like our algorithm. The gradient consistency condition, which is essential for establishing function descent, is not respected in that algorithm. In Section 4 we compare our methods against the method in Sharir et al. (2014).

BFGS approximation. For $\tilde{L}_p$ we can either use L_p or a second order approximation, like in the approximations given above. For $\hat{L}_p$ we can use a second order term, $\frac{1}{2}(w - w^r) \cdot H(w - w^r)$ where H is a positive semi-definite matrix; for H we can use a diagonal approximation or keep a limited history of gradients and form a BFGS approximation of $L - L_p$.

Remark 2. The distributed method described above is an instance of Algorithm 1 and so Theorem 2 can be used. In Theorem 2 we mentioned a convergence rate, δ. For $\cos\theta = \sigma/L$ this yields the rate $\delta = (1 - 2\alpha(1-\beta)(\frac{\sigma}{L})^4)$. This rate is obviously pessimistic given that it applies to general choices of $\hat{f}_p$ satisfying minimal assumptions. Actual rates of convergence depend a lot on the choice made for $\hat{f}_p$. Suppose we choose $\hat{f}_p$ via Hybrid, Quadratic or Nonlinear approximation choices mentioned in Subsection 3.2 and minimize $\hat{f}_p$ exactly in the inner optimization. These approximations are invariant to coordinate transformations such as $w' = Bw$, where B is a positive definite matrix. Note that Armijo-Wolfe line search conditions are also unaffected by such transformations. What this means is that, at each iteration, we can, without changing the algorithm, choose for analysis a coordinate transformation that gives the best rate. The Linear approximation choice for $\hat{f}_p$ does not enjoy this property. This explains why the Hybrid, Quadratic and Nonlinear approximations perform so well and give great rates of convergence in practice; see the experiments in Section 4.7 and Subsection 4.9.1. Proving much better convergence rates for FADL using these approximations would be interesting future work.

In Section 4 we evaluate some of these approximations in detail.

3.3. Convergence theory

In practice, exactly minimizing $\hat{f}_p$ is infeasible. For convergence, it is not necessary for w_p to be the minimizer of $\hat{f}_p$; we only need to find w_p such that the direction $d_p = w_p - w^r$ satisfies (1). The angle θ needs to be chosen right. Let us discuss this first. Let $\hat{w}_p^\star$ be the minimizer of $\hat{f}_p$. It can be shown (see appendix B) that $\angle(\hat{w}_p^\star - w^r, -g^r) \leq \cos^{-1}\frac{\sigma}{L}$. To allow for w_p being an approximation of $\hat{w}_p^\star$, we choose θ such that

$$\frac{\pi}{2} > \theta > \cos^{-1}\frac{\sigma}{L}. \tag{18}$$

The following result shows that if an optimizer with *glrc* is used to minimize $\hat{f}_p$, then, only a constant number of iterations is needed to satisfy the sufficient angle of descent condition.

Lemma 3. Assume $g^r \neq 0$. Suppose we minimize $\hat{f}_p$ using an optimizer $\mathcal{M}$ that starts from $v^0 = w^r$ and generates a sequence $\{v^k\}$ having *glrc*, i.e., $\hat{f}_p(v^{k+1}) - \hat{f}_p^\star \leq \delta(\hat{f}_p(v^k) - \hat{f}_p^\star)$, where $\hat{f}_p^\star = \hat{f}_p(\hat{w}_p^\star)$. Then, there exists $\hat{k}$ (which depends only on σ and L) such that $\angle(-g^r, v^k - w^r) \leq \theta \ \ \forall k \geq \hat{k}$.

Lemma 3 can be combined with Theorem 2 to yield the following convergence theorem.

Theorem 4. Suppose θ satisfies (18), $\mathcal{M}$ is as in Lemma 3 and, in each iteration r and for each p, $\hat{k}$ or more iterations of $\mathcal{M}$ are applied to minimize $\hat{f}_p$ (starting from w^r) and get w_p. Then the distributed method converges to a point w satisfying $f(w) - f(w^\star) \leq \epsilon$ in $O(\log(1/\epsilon))$ time.

Proofs of Lemma 3 and Theorem 4 are given in appendix B.

3.4. Practical implementation.

We refer to our method by the acronym, FADL - Function Approximation based Distributed Learning. Going with the practice in numerical optimization, we replace (1) by the condition, $-g^r \cdot d^r > 0$ and use $\alpha = 10^{-4}$, $\beta = 0.9$ in (4) and (5). In actual usage, Algorithm 1 can be terminated when $\|g^r\| \leq \epsilon_g \|g^0\|$ is satisfied at some r. Let us take line search next. On $w = w^r + td^r$, the loss has the form $l(z_i + te_i, y_i)$ where $z_i = w^r \cdot x_i$ and $e_i = d^r \cdot x_i$. Once we have computed $z_i \ \forall i$ and $e_i \ \forall i$, the distributed computation of $f(w^r + td^r)$ and its derivative with respect to t is cheap as it does not involve any computation involving the data, $\{x_i\}$. Thus, many t values can be explored cheaply. Since d^r is determined by approximate optimization, $t = 1$ is expected to give a decent starting point. We first identify an interval $[t_1, t_2] \subset [t_\beta, t_\alpha]$ (see Lemma 1) by starting from $t = 1$ and doing forward and backward stepping. Then we check if t_1 or t_2 is the minimizer of $f(w^r + td^r)$ on $[t_1, t_2]$; if not, we do several bracketing steps in (t_1, t_2) to locate the minimizer approximately. Finally, when using method $\mathcal{M}$, we terminate it after a fixed number of steps, $\hat{k}$; we have found that, setting $\hat{k}$ to match computation and communication costs works effectively. Algorithm 2 gives all the steps of FADL while also mentioning the distributed communications and computations involved.

Choices for $\mathcal{M}$. There are many good methods having (deterministic) *glrc*: L-BFGS, TRON (Lin et al., 2008), Primal coordinate descent (Chang et al., 2008), etc. One could also use methods with *glrc* in the expectation sense (in which case, the convergence in Theorem 4 should be interpreted in some probabilistic sense). This nicely connects our method with recent literature on parallel SGD. We discuss this in the next subsection only briefly as it is outside the scope of the current paper. See our related work (Mahajan et al., 2013b) for details. For the experiments of this paper we do the following. For optimizing the quadratic approximation in (10) with (14)-(15), we used the conjugate-gradient method (Shewchuk, 1994). For all other (non-quadratic) approximations of FADL as well as all nonlinear solvers needed by other methods, we used TRON.

3.5. Connections with parallel SGD

For large scale learning on a single machine, example-wise methods[6] such as stochastic gradient descent (SGD) and its variations (Bottou, 2010; Johnson and Zhang, 2013) and dual

6. These methods update w after scanning each example.

Algorithm 2: FADL - Function Approximation based Distributed Learning. *com:* communication; *cmp:* = computation; *agg:* aggregation. $\mathcal{M}$ is the optimizer used for minimizing $\hat{f}_p$.

Choose w^0;
for $r = 0, 1 \ldots$ **do**
  1. Compute g^r (*com:* w^r; *cmp:* Two passes over data; *agg:* g^r); By-product: $\{z_i = w^r \cdot x_i\}$;
  2. Exit if $\|g^r\| \leq \epsilon_g \|g^0\|$;
  3. **for** $p = 1, \ldots, P$ *(in parallel)* **do**
    4. Set $v^0 = w^r$;
    5. **for** $k = 0, 1, \ldots, \hat{k}$ **do**
      6. Find v^{k+1} using one iteration of $\mathcal{M}$;
    end
    7. Set $w_p = v^{\hat{k}+1}$;
  end
  8. Set d^r as any convex combination of $\{w_p\}$ (*agg:* w_p);
  9. Compute $\{e_i = d^r \cdot x_i\}$ (*com:* d^r; *cmp:* One pass over data);
  10. Do line search to find t (for each t: *com:* t; *cmp:* l and $\partial l/\partial t$ *agg:* $f(w^r + td^r)$ and its derivative wrt t);
  11. Set $w^{r+1} = w^r + td^r$;
end

coordinate ascent (Hsieh et al., 2008) perform quite well. However, example-wise methods are inherently sequential. If one employs a method such as SGD as $\mathcal{M}$, the local optimizer for $\hat{f}_p$, the result is, in essence, a parallel SGD method. However, with parameter mixing and iterative parameter mixing methods (Mann et al., 2009; Hall et al., 2010; McDonald et al., 2010) (we briefly discussed these methods in Section 1) that do not do line search, convergence theory is limited, even that requiring a complicated analysis (Zinkevich et al., 2010); see also Mann et al. (2009) for some limited results. Thus, the following has been an unanswered question: **Q3.** *Can we form a parallel SGD method with strong convergence properties such as glrc?*

As one special instantiation of our distributed method, we can use, for the local optimization method $\mathcal{M}$, any variation of SGD with *glrc* (in expectation), e.g., the one in Johnson and Zhang (2013). For this case, in a related work of ours (Mahajan et al., 2013b) we show that our method has $O(\log(1/\epsilon))$ time convergence in a probabilistic sense. The result is a strongly convergent parallel SGD method, which answers **Q3**. An interesting side observation is that, the single machine version of this instantiation is very close to the variance-reducing SGD method in Johnson and Zhang (2013). We discuss this next.

Connection with SVRG. Let us take the $\hat{f}_p$ in (11). Let $n_p = |I_p|$ be the number of examples in node p. Define $\psi_i(w) = n_p l(w \cdot x_i, y_i) + \frac{\lambda}{2}\|w\|^2$. It is easy to check that

$$\nabla \hat{f}_p(w) = \frac{1}{n_p} \sum_{i \in I_p} (\nabla \psi_i(w) - \nabla \psi_i(w^r) + g^r). \tag{19}$$

Thus, plain SGD updates applied to $\hat{f}_p$ has the form

$$w = w - \eta(\nabla\psi_i(w) - \nabla\psi_i(w^r) + g^r), \tag{20}$$

which is precisely the update in SVRG. In particular, the single node ($P = 1$) implementation of our method using plain SGD updates for optimizing $\hat{f}_p$ is very close to the SVRG method.[7] While Johnson and Zhang (2013) motivate the update in terms of variance reduction, we derive it from a functional approximation viewpoint.

3.6. Computation-Communication tradeoff

In this subsection we do a rough analysis to understand the conditions under which our method (FADL) is faster than the SQM method (Chu et al., 2006; Agarwal et al., 2011) (see Section 1). This analysis is only for understanding the role of various parameters and not for getting any precise comparison of the speed of the two methods.

Compared to the SQM method, FADL does a lot more computation (optimize $\hat{f}_p$) in each node. On the other hand FADL reaches a good solution using a much smaller number of outer iterations. Clearly, FADL will be attractive for problems with high communication costs, e.g., problems with a large feature dimension. For a given distributed computing environment and specific implementation choices, it is easy to do a rough analysis to understand the conditions in which FADL will be more efficient than SQM. Consider a distributed grid of nodes in an *AllReduce* tree. Let us use a common method such as TRON for implementing SQM as well as for $\mathcal{M}$ in FADL. Assuming that $T^{\text{outer}}_{\text{SQM}} > 3.0T^{outer}_{\text{FADL}}$ (where $T^{\text{outer}}_{\text{FADL}}$ and T^{outer}_{SQM} are the number of outer iterations required by SQM and FADL), we can do a rough analysis of the costs of SQM and FADL (see appendix A for details) to show that FADL will be faster when the following condition is satisfied.

$$\frac{nz}{m} < \frac{\gamma P}{2\hat{k}} \tag{21}$$

where: nz is the number of nonzero elements in the data, i.e., $\{x_i\}$; m is the feature dimension; γ is the relative cost of communication to computation (e.g. $100 - 1000$); P is the number of nodes; and $\hat{k}$ is the number of inner iterations of FADL. Thus, the larger the dimension (m) is, and the higher the sparsity in the data is, FADL will be better than SQM.

4. Experiments

In this section, we demonstrate the effectiveness of our method by comparing it against several existing distributed training methods on five large data sets. We first discuss our experimental setup. We then briefly list each method considered and then do experiments to decide the best overall setting for each method. This applies to our method too, for which the setting is mainly decided by the choice made for the function approximation, $\hat{f}_p$; see Subsection 3.2 for details of these choices. Finally, we compare, in detail, all the methods under their best settings. This study clearly demonstrates scenarios under which our method performs better than other methods.

7. Note the subtle point that applying SVRG method on $\hat{f}_p$ is different from doing (20), which corresponds to plain SGD. It is the former that assures *glrc* (in expectation).

4.1. Experimental Setup

We ran all our experiments on a Hadoop cluster with 379 nodes and 10 Gbit interconnect speed. Each node has Intel (R) Xeon (R) E5-2450L (2 processors) running at 1.8 GHz. Since iterations in traditional *MapReduce* are slower (because of job setup and disk access costs), as in Agarwal et al. (Agarwal et al., 2011), we build an *AllReduce* binary tree between the mappers[8]. The communication bandwidth is *1 Gbps* (gigabits per sec).

Dataset	#Examples (n)	#Features (m)	#Non-zeros (nz)	λ/n
kdd2010	8.41×10^6	20.21×10^6	0.31×10^9	1.25×10^{-6}
url	1.91×10^6	3.23×10^6	0.22×10^9	0.11×10^{-6}
webspam	0.35×10^6	16.6×10^6	0.98×10^9	1.0×10^{-4}
mnist8m	8.1×10^6	784	6.35×10^9	1.0×10^{-4}
rcv	0.5×10^6	47236	0.50×10^8	1.0×10^{-4}

Table 1: Properties of datasets.

Data Sets. We consider the following publicly available datasets having a large number of examples:[9] *kdd2010*, *url*, *webspam*, *mnist8m* and *rcv*. Table 1 shows the numbers of examples, features, nonzero in data matrix and the values of regularizer λ used. The regularizer for each dataset is chosen to be the optimal value that gives the best performance on a small validation set. We use these datasets mainly to illustrate the validity of theory, and its utility to distributed machine learning. In real scenarios of Big data, the datasets are typically much larger. Note that *kdd2010*, *url* and *webspam* are large dimensional (m is large) while *mnist8m* and *rcv* are low/medium dimensional (m is not high). This division of the datasets is useful because communication cost in example-partitioned distributed methods is mainly dependent on m (see Appendix A) and so these datasets somewhat help to see the effect of communications cost.

We use the *squared-hinge loss* function for all the experiments. Unless stated differently, for all numerical optimizations we use the Trust Region Newton method (TRON) proposed in Lin et al. (2008).

Evaluation Criteria. We use the relative difference to the optimal function value and the Area under Precision-Recall Curve (AUPRC) (Sonnenburg and Franc, 2010; Agarwal et al., 2013)[10] as the evaluation criteria. The former is calculated as $(f - f^*)/f^*$ in log scale, where f^* is the optimal function value obtained by running the *TERA* algorithm (see below) for a very large number of iterations.

4.2. Methods for comparison

We compare the following methods.

8. Note that we do not use the pipelined version and hence we incur an extra multiplicative $logP$ cost in communication.
9. These datasets are available at: `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. For *mnist8m* we solve the binary problem of separating class "3" from others.
10. We employed AUPRC instead of AUC because it differentiates methods more finely.

- *TERA:* The Terascale method (TERA) (Agarwal et al., 2011) is the best representative method from the SQM class (Chu et al., 2006). It can be considered as the state-of-the-art distributed solver and therefore an important baseline.

- *ADMM:* We use the example partitioning formulation of the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011; Zhang et al., 2012). ADMM is a dual method which is very different from our primal method; however, like our method, it solves approximate problems in the nodes and iteratively reaches the full batch solution.

- *CoCoA:* This method (Jaggi et al., 2014) represents the class of distributed dual methods (Pechyony et al., 2011; Yang, 2013; Yang et al., 2013; Jaggi et al., 2014) that, in each outer iteration, solve (in parallel) several local dual optimization problems.

- *DANE:* This is the Newton based method described in Sharir et al. (2014) that uses a function approximation similar to FADL.

- *DiSCO:* This (Zhang and Xiao, 2015) is a distributed Newton method designed with communication-efficiency in mind.

- *Our method (FADL):* This is our method described in detail in Section 3 and more specifically, in Algorithm 2.

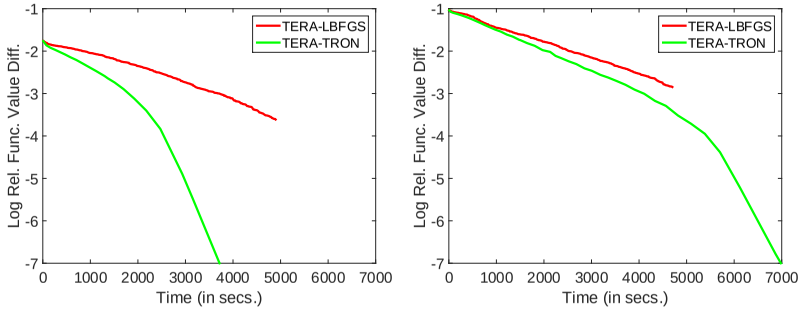
4.3. Study of TERA

A key attractive property of TERA is that the number of outer iterations pretty much remains constant with respect to the number of distributed nodes used. As recommended by Agarwal et al. (2011), we find a local weight vector per node by minimizing the local objective function (based only on the examples in that node) using five epochs of SGD (Bottou, 2010). (The optimal step size is chosen by running SGD on a subset of data.) We then average the weights from all the nodes (on a per-feature basis as explained in Agarwal et al. (2011)) and use the averaged weight vector to warm start *TERA*.[11] Agarwal et al. (2011) use the LBFGS method as the trainer whereas we use TRON. To make sure that this does not lead to bias, we try both, TERA-LBFGS and TERA-TRON. Figure 1 compares the progress of objective function for these two choices. Clearly, TERA-TRON is superior. We observe similar behavior on the other datasets also. Given this, we restrict ourselves to TERA-TRON and simply refer to it as TERA.

4.4. Study of ADMM

The ADMM objective function (Boyd et al., 2011) has a quadratic proximal term called augmented Lagrangian with a penalty parameter ρ multiplying it. In general, the performance of ADMM is very sensitive to the value of ρ and hence making a good choice for it is crucial. We consider three methods for choosing ρ.

11. We use this inexpensive initialization for FADL and ADMM too. It is not applicable to CoCoA. Because of this, CoCoA starts with a different primal objective function value than others.



(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

Figure 1: Plots showing the time efficiency of TERA methods for *kdd2010*.

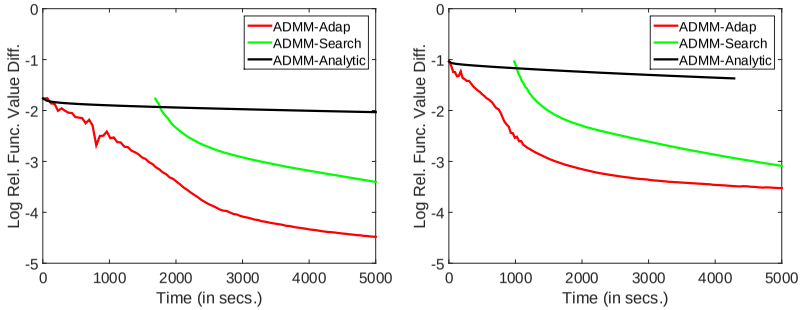
Even though there is no supporting theory, Boyd et al (Boyd et al., 2011) suggest an approach by which ρ is adapted in each iteration; see Equation (3.13) in Section 3.4.1 of that paper. We will refer to this choice as *Adap*.

Recently, Deng and Yin (2012) proved a linear rate of convergence for ADMM under assumptions A1 and A2 (see Section 2) on ADMM functions. As a result, their analysis also hold for the objective function in (8). They also give an analytical formula to set ρ in order to get the best theoretical linear rate constant. We will refer to this choice of ρ as *Analytic*.

We also consider a third choice, *ADMM-Search* in which, we start with the value of ρ given by *Analytic*, choose several values of ρ in its neighborhood and select the best ρ by running ADMM for 10 iterations and looking at the objective function value. Note that this step takes additional time and causes a late start of ADMM.

Figure 2 compares the progress of the training objective function for the three choices on *kdd2010* for $P = 8$ and $P = 128$. *Analytic* is an order of magnitude slower than the other two choices. *Search* works well. However, a significant amount of initial time is spent on finding a good value for ρ, thus making the overall approach slow. *Adap* comes out to be the best performer among the three choices. Similar observations hold for other datasets and other choices of P. So, for ADMM, we will fix *Adap* as the way of choosing ρ and refer to ADMM-*Adap* simply as ADMM.

It is also worth commenting on methods related to ADMM. Apart from ADMM, Bertsekas and Tsitsiklis (1997) discuss several other classic optimization methods for separable convex programming, based on proximal and Augmented Lagrangian ideas which can be used for distributed training of linear classifiers. ADMM represents the best of these methods. Also, Gauss-Seidel and Jacobi methods given in Bertsekas and Tsitsiklis (1997) are related to feature partitioning, which is very different from the example partitioning scenario studied in this paper. Therefore we do not consider these methods.



(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

Figure 2: Plots showing the time efficiency of ADMM methods for *kdd2010*.

4.5. Study of CoCoA

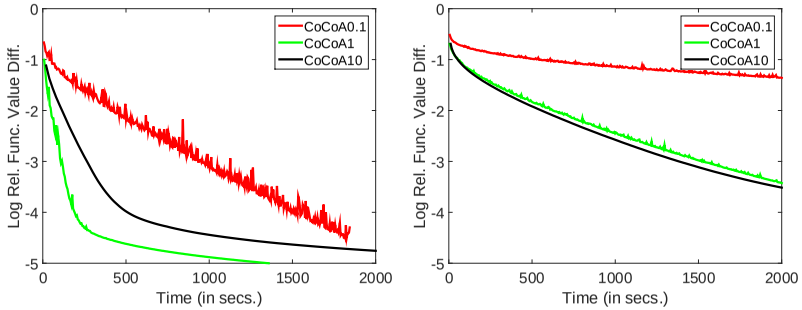
In CoCoA (Jaggi et al., 2014) the key parameter is the approximation level of the inner iterations used to solve each projected dual sub-problem. The number of epochs of coordinate dual ascent inner iterations plays a crucial role. We try the following choices for it: 0.1, 1 and 10.[12] Figure 3 compares the progress of the objective function on *kdd2010* for two choices of nodes, $P = 8$ and $P = 128$. We find the choice of 1 epoch to work well reasonably consistently over all the five datasets and varying number of nodes. So we fix this choice and refer to the resulting method simply as CoCoA. Note in Figure 3 that the (primal) objective function does not decrease continuously with time. This is because it is a dual method and so monotone descent of the objective function is not assured.[13]

4.6. Study of DANE

In our detailed comparison study later in this section, we will also include another method called DANE, which has some resemblance to FADL. This is the Newton based method described in Sharir et al. (2014). Even though this method is very different in spirit from our method, it uses a function approximation similar to our Nonlinear approximation idea; we briefly discussed this in Subsection 3.2. DANE is a non-monotone method that is based on fixed step sizes, with a probabilistic convergence theory to support it.

DANE has two parameters, μ and η in the function approximation; μ is the coefficient for the proximal term and η is used for defining the direction. Sharir et al. (2014) do not prove convergence for any possible choices of μ and η values; their practical recommendation is to use $\mu = 3\lambda$ and $\eta = 1$. The choice of μ in particular, turns out to be quite sub-optimal. We found that there was no single value of μ that is good for all datasets, and so it needs to be tuned for each dataset. So, to improve DANE, we included an initial μ-tuning step.

12. The examples were randomly shuffled for presentation. When the number of epochs is a fraction, e.g., 0.1, the inner iteration was stopped after 10% of the examples were presented.
13. The same comment holds for ADMM which is also a dual method; see for example, the jumps in objective function values for ADMM in Figures 6 and 8 for *kdd2010*.



(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

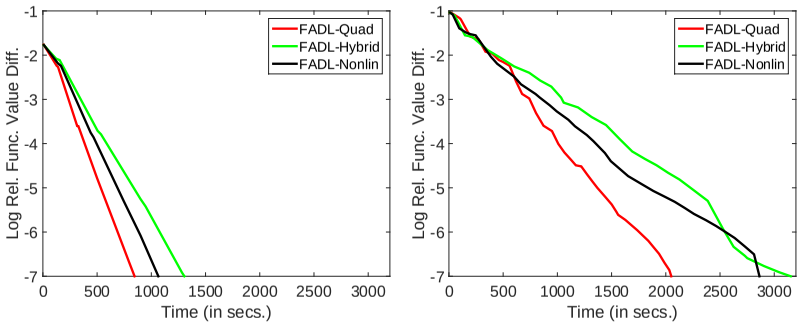
Figure 3: Plots showing the time efficiency of CoCoA settings for *kdd2010*.

We start with $\mu = 3 * \lambda$ and use an initial set of four outer iterations to choose the μ value that gives the best improvement in objective function. Starting from $\mu = 3 * \lambda$ we try several values of μ in the direction of μ change that leads to improvement. After the first four outer iterations, we fix μ at the chosen best value for the remaining iterations. (Note that the cost associated with tuning μ has to be included in the overall cost of DANE.) As we will see later in Subsection 4.9, in spite of all this tuning, DANE did not converge in several situations. Essentially, the issue is that μ has to be adaptive - DANE needs to use different choices of μ in the early, middle and end stages of one training. But then, this is nearly akin to doing a kind of line search in each outer iteration. As opposed to DANE, note that FADL is a monotone method directly based on line search, and it does not need the proximal term to restrict step-sizes. Also, unlike DANE, all the parameters of FADL are fixed to default values for all datasets.

The choice of inner optimizer for DANE is crucial for its efficiency. We tried SVRG (Johnson and Zhang, 2013) and also TRON (Lin et al., 2008). Both did well, but SVRG was better, especially for small number of nodes. So we did a more detailed study of SVRG. We tried two choices. (a) Fix the number of epochs for the local (inner) optimization to some good value, e.g., 10. (b) Always choose the number of epochs for the local (inner) optimization to be such that the cost of local computation in a node is matched with the communications cost. Choice (b) gave a better solution. So we have used it for DANE.

4.7. Study of FADL

Recall from Subsection 3.2 the various choices that we suggested for $\hat{f}_p$. We are yet to implement and study the BFGS approximation; we leave it out for future work. Our original work (Mahajan et al., 2013a) focused on the Linear approximation, but we found the Quadratic, Hybrid and Nonlinear choices to work much better. So we study only these three methods. The implementation of these methods is as described in Subsection 3.4 and Algorithm 2.



(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

Figure 4: Plots showing the time efficiency of the three function approximations of FADL for *kdd2010*.

Figure 4 compares the progress of the training objective function for various choices of $\hat{f}_p$. Among the choices, the quadratic approximation for $\hat{f}_p$ gives the best performance, although the Hybrid and Nonlinear approximations also do quite well. We observe this reasonably consistently in other datasets too. Hence, from the set of methods considered in this subsection we choose FADL-Quadratic approximation as the only method for further analysis, and simply refer to this method as FADL hereafter.

Why does the quadratic approximation do better than hybrid and nonlinear approximations? We do not have a precise answer to this question, but we give some arguments in support. In each outer iteration, the function approximation idea is mainly used to get a good direction. Recall from Subsection 3.2 that different choices use different approximations for $\tilde{L}_p$ and $\hat{L}_p$. Using the same "type" (meaning linear, nonlinear or quadratic) for both, $\tilde{L}_p$ and $\hat{L}_p$ is possibly better for direction finding. Second, the direction finding could be more sensitive to the nonlinear approximation compared to the quadratic approximation; this could become more severe as the number of nodes becomes larger. Literature shows that quadratic approximations have good robustness properties; for example, subsampling in Hessian computation (Byrd et al., 2012) doesn't worsen direction finding much.

4.8. Comparison of FADL against DiSCO

DiSCO (Zhang and Xiao, 2015) is an inexact damped Newton method, where the inexact Newton steps are computed using a distributed preconditioned conjugate gradient method. Like TERA, it belongs to the SQM class (Chu et al., 2006). The DiSCO method as described in Zhang and Xiao (2015) is not designed for squared hinge loss. Therefore, in this subsection we separately compare FADL against DiSCO using the logistic loss. With communication efficiency in mind we compare the two methods in terms of the number of communication passes. Figure 5 gives the progress in objective function as a function of the number of communication passes for the two methods. FADL is clearly better than DiSCO, the

difference being quite large on datasets such as *kdd2010*. In terms of total computing time, the superiority of FADL turns out to be even better than what is seen in the plots of communication passes. This is because DiSCO requires more extensive computational steps than FADL within one communication pass.

4.9. Comparison of FADL against TERA, ADMM, CoCoA and DANE

Having made the best choice of settings for the methods, we now evaluate FADL against TERA, ADMM, CoCoA and DANE in more detail. We do this using three sets of plots. We give details only for $P = 8$ and $P = 128$ to give an idea of how performance varies for small and large number of nodes.

1. *Communication passes.* We plot the variation of the training objective function as a function of the number of communication passes. For the x-axis we prefer the number of communication passes instead of the number of outer iterations since the latter has a different meaning for different methods while the former is quite uniform for all methods. Figures 6 and 7 give the plots respectively for the large dimensional (m large) datasets (*kdd2010*, *url* and *webspam*) and medium/small dimensional (m medium/small) datasets (*mnist8m* and *rcv*).

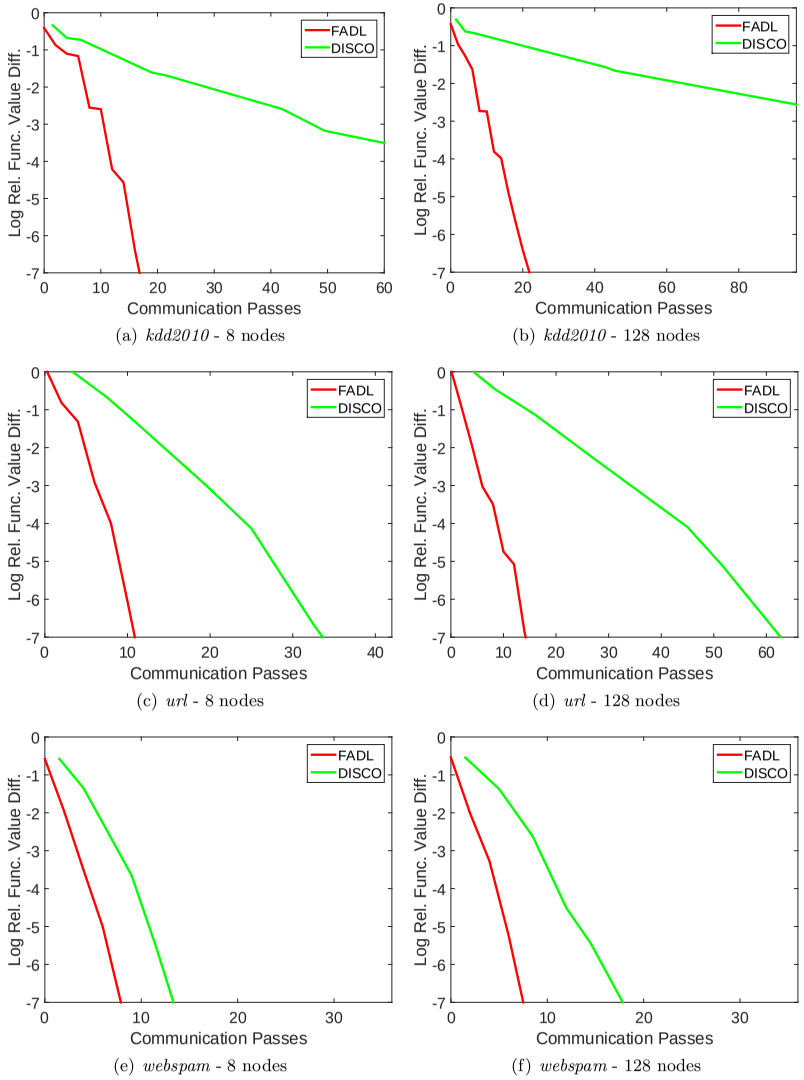
2. *Time.* We plot the variation of the training objective function as a function of the actual solution time. Figures 8 and 9 give the plots respectively for the large dimensional and medium/small dimensional datasets.

3. *Speed-up over TERA.* TERA is an established strong baseline method. So it is useful to ask how other methods fare relative to TERA and study this as a function of the number of nodes. For doing this we need to represent each method by one or two real numbers that indicate performance. Since generalization performance is finally the quantity of interest, we stop a method when it reaches within 0.1% of the steady state AUPRC value achieved by full, perfect training of (8) and record the following two measures: the total number of communication passes and the total time taken. For each measure, we plot the ratio of the measure's value for TERA to the corresponding measure's value for a method, as a function of the number of nodes, and repeat this for each method. Larger this ratio, better is a method; also, ratio greater than one means a method is faster than TERA. Figures 10 and 11 give the plots for all the five datasets.

Let us now use these plots to compare the methods. In the plots, DANE starts later than other methods due to the extra work needed for tuning the proximal parameter μ.

4.9.1. Rate of Convergence

Analysis of the rate of convergence is better done by studying the behavior of the training objective function with respect to the number of communication passes. So it is useful to look at Figures 6 and 7. Clearly, as predicted by theory, the rate of convergence is linear for all methods.

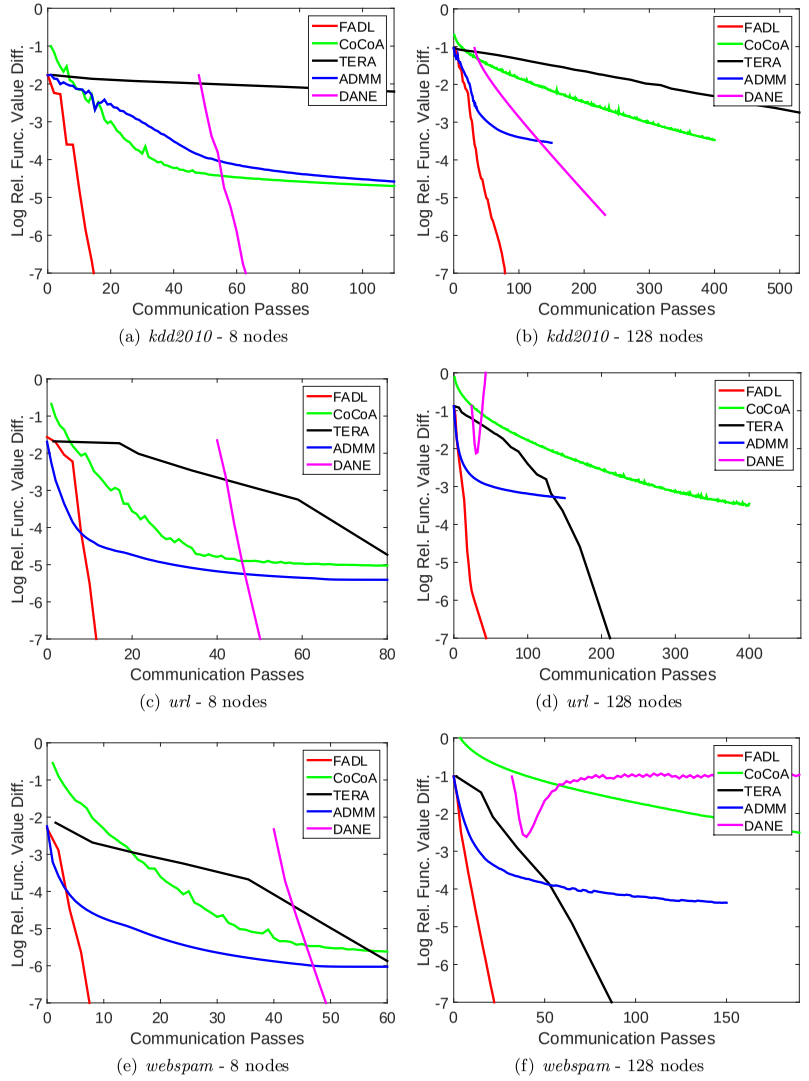


(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

(c) *url* - 8 nodes (d) *url* - 128 nodes

(e) *webspam* - 8 nodes (f) *webspam* - 128 nodes

Figure 5: Plots comparing the number of communication passes of DiSCO and FADL on three datasets.

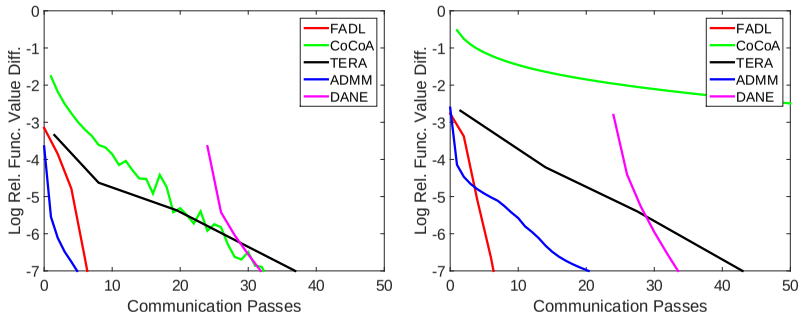


(a) *kdd2010* - 8 nodes (b) *kdd2010* - 128 nodes

(c) *url* - 8 nodes (d) *url* - 128 nodes

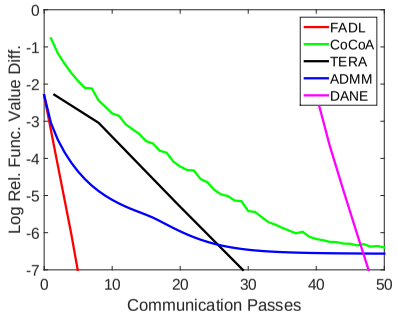
(e) *webspam* - 8 nodes (f) *webspam* - 128 nodes

Figure 6: Plots showing the rate of convergence of various methods for the three high dimensional datasets.

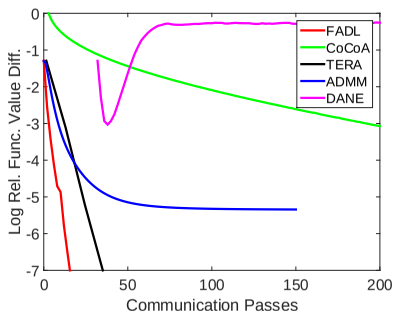


(a) *mnist8m* - 8 nodes

(b) *mnist8m* - 128 nodes



(c) *rcv* - 8 nodes



(d) *rcv* - 128 nodes

Figure 7: Plots showing the linear convergence of various methods for the two low/medium dimensional datasets.

TERA uses distributed computation only to compute the gradient and so the plots should be unaffected by P. But in the plots we do see differences between the plots for $P = 8$ and $P = 128$. This is because of their different initialization (average of one pass SGD solutions from nodes): the initialization with lower number of nodes is better due its smaller variance; note also the better starting objective function value at the start (left most point) for $P = 8$.

For FADL, the rate is steeper for $P = 8$ than for $P = 128$. This steeper behavior for lower number of nodes is expected because the functional approximation in each node becomes better as the number of nodes decreases.

Recall from Section 1 that, our main aim behind the design of the function approximation based methods is to reduce the number of communication passes significantly. The plots in Figures 6 and 7 clearly confirm such a reduction.

Even though the end convergence rate of ADMM is slow, it generally shows good rates of convergence in the initial stages of training. This is a useful behavior because generalization measures such as AUPRC tend to achieve steady state values quickly in the early stages. This usefulness is seen in Figure 10 too.

DANE has good rates of convergence when P is small ($P = 8$), but the cost associated with tuning μ makes it poor; note that, if μ is not tuned, that will affect the rate of convergence. DANE tends to be unstable for large P values.

Overall, FADL gives much better rates of convergence (both, in the early training stage as well as in the end stage) compared to TERA, CoCoA, ADMM and DANE methods.[14] FADL shows a large reduction in the number of communication passes over TERA, especially when the number of nodes is small. Against CoCoA the trend is the other way: FADL needs a much smaller number of communication passes than CoCoA especially when the number of nodes is large. These observations can also be seen from Figure 10. Clearly CoCoA seems to be very slow with increasing number of nodes.

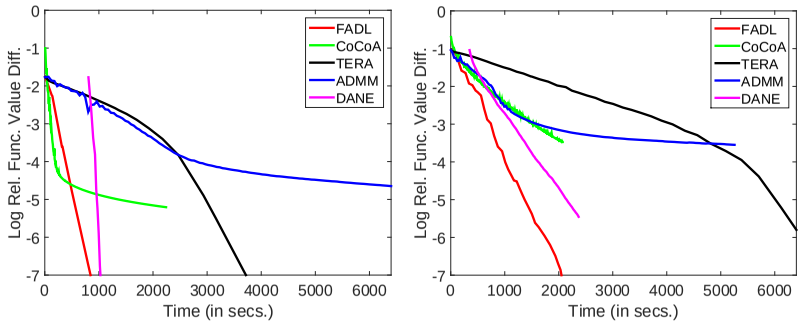
4.9.2. Time Taken

In the previous analysis we ignored computation costs within each iteration. But these costs play a key role when we analyze overall efficiency in terms of the actual time taken. We study this next. Figures 8 and 9 are relevant for this study. FADL, ADMM, CoCoA and DANE involve much more extensive computations in the inner iterations than TERA; this is especially true when the number of nodes is small because of the large amount of local data in each node. TERA fares much better in the time analysis than what we saw while studying using communication passes only. Thus the gap between TERA and other methods becomes a lot smaller in the time analysis. Compare, for example, TERA and ADMM with respect to communication passes and time. Although ADMM is much more efficient than TERA with respect to the number of communication passes, TERA catches up nicely with ADMM on the time taken.

CoCoA does well sometimes; for example, on *kdd2010* and *url*, when the number of nodes is small, say $P = 8$. But it is slow otherwise, especially when the number of nodes is large.

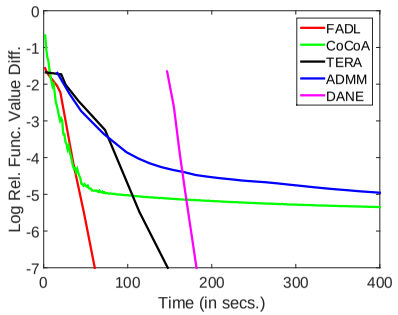
FADL is uniformly better than ADMM with respect to the total time taken. Overall, FADL shows the best performance, performing equally or much better than other methods in different situations. With medium/low dimensional datasets (see Figure 9), communication time is less of an issue and so the expectation is that FADL is less of value for them. Even on these datasets, FADL does equally or better than TERA. DANE is not competitive due to the time involved for tuning μ.

14. Here, it is useful to recall the comments made in Remark 2 of Section 3 on the goodness of the convergence rate of FADL.

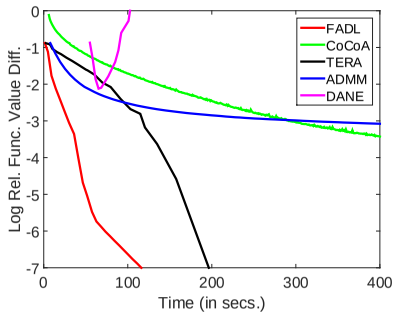


(a) *kdd2010* - 8 nodes

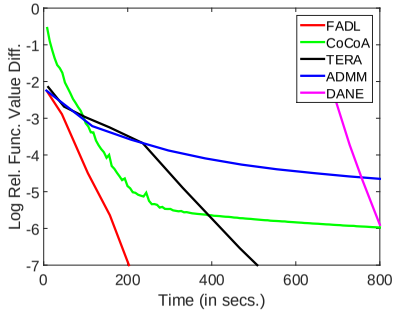
(b) *kdd2010* - 128 nodes



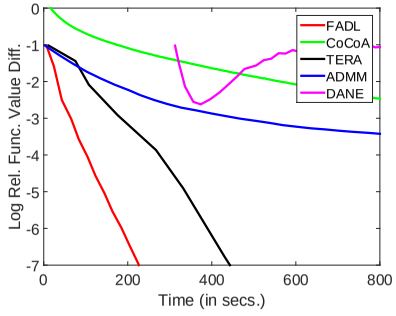
(c) *url* - 8 nodes



(d) *url* - 128 nodes

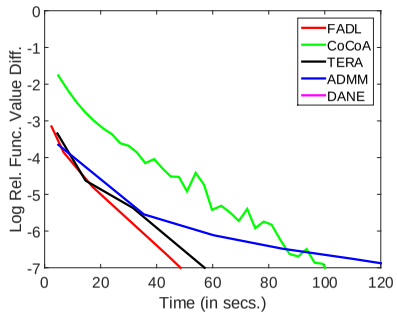


(e) *webspam* - 8 nodes

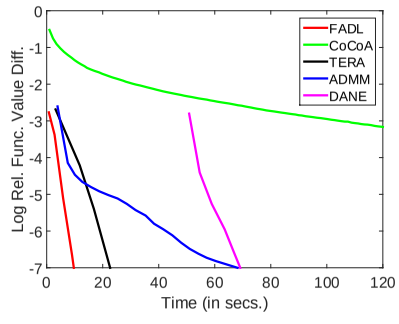


(f) *webspam* - 128 nodes

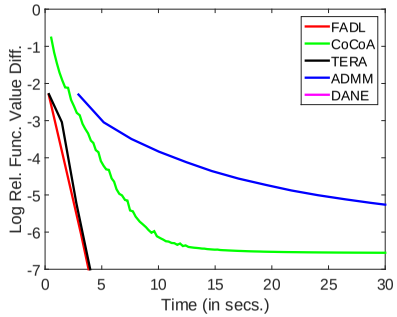
Figure 8: Plots showing the time efficiency of various methods for the three high dimensional datasets.



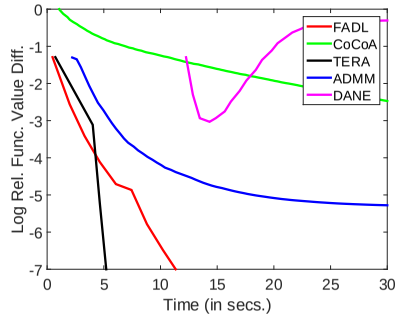
(a) *mnist8m* - 8 nodes



(b) *mnist8m* - 128 nodes



(c) *rcv* - 8 nodes

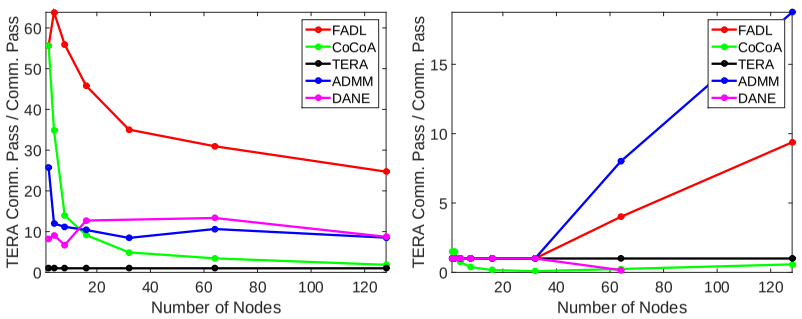


(d) *rcv* - 128 nodes

Figure 9: Plots showing the time efficiency of various methods for the two low/medium dimensional datasets. For *rcv* and *mnist8m* - 8 nodes, the plot of DANE is invisible since the time taken for tuning μ is larger than the time window displayed.

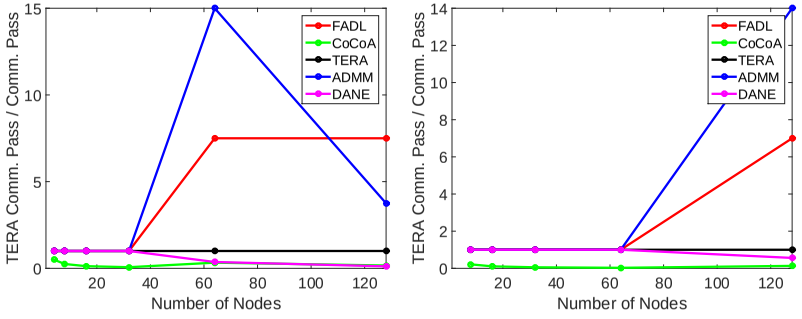
4.9.3. Relative performance of the methods

Figure 11 is relevant for this study. CoCoA shows impressive speed-up over TERA on *kdd2010* but it is much slower on all the other datasets. It is unclear why CoCoA fares so well on *kdd2010* but not on the other datasets. ADMM gives an overall decent performance when compared to TERA. FADL is consistently faster than TERA, with speed-ups ranging anywhere from 1-10. In communication-heavy scenarios where reducing the number of communication passes is most important, methods such as FADL and ADMM have great value (see Figure 10), with the possibility of getting even higher speed-ups over TERA. Except for *kdd2010* for which FADL is slower than CoCoA for small number of nodes, it is generally the fastest method. DANE does not seem to have any special scenario where it is better than others.



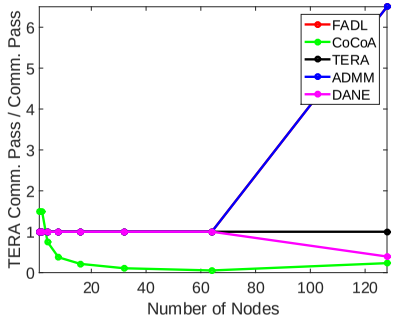
(a) kdd2010

(b) url



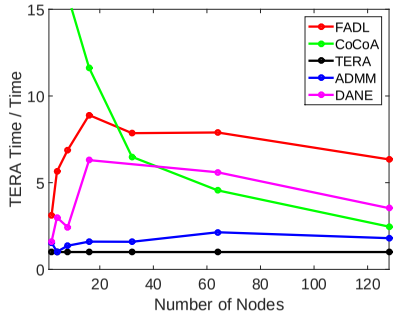
(c) webspam

(d) mnist8m

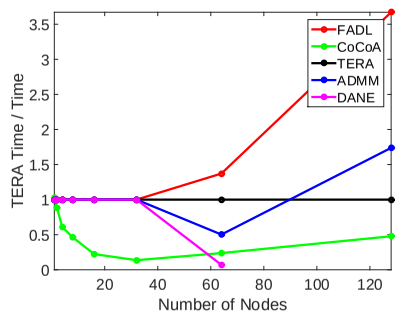


(e) rcv

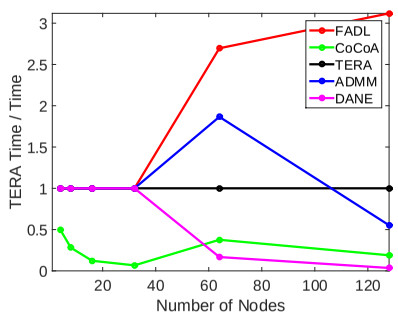
Figure 10: Plots showing communication passes (relative to TERA) as a function of the number of nodes. Each method was terminated when it reached within 0.1% of the steady state AUPRC value achieved by full, perfect training of (8). For *rcv*, the FADL and ADMM curves coincide.



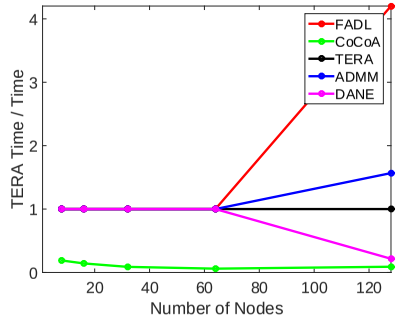
(a) kdd2010



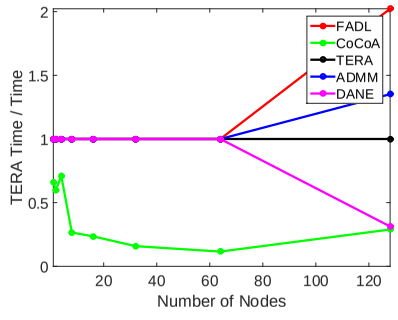
(b) url



(c) webspam



(d) mnist8m



(e) rcv

Figure 11: Plots showing time (relative to TERA) as a function of the number of nodes. Each method was terminated when it reached within 0.1% of the steady state AUPRC value achieved by full, perfect training of (8).

4.9.4. Speed-up as a function of P

Let us revisit Figure 8 and look at the plots corresponding to *kdd2010* for FADL.[15] It can be observed that the time needed for reaching a certain tolerance, say *Log Rel. Func. Value Diff. = -3*, is two times smaller for $P = 8$ than for $P = 128$. This means that using a large number of nodes is not useful, which prompts the question: *Is a distributed solution really necessary?* There are two answers to this question. First, as we already mentioned, when the training data is huge[16] and *the data is generated and forced to reside in distributed nodes (moving data between machines is not an efficient option)*, the right question to ask is not whether we get great speed-up, but to ask which method is the fastest. Second, for a given dataset and method, if the time taken to reach a certain approximate stopping tolerance (e.g., based on AUPRC) is plotted as a function of P, it usually has a minimum at a value $P > 1$. Given this, it is appropriate to choose a P optimally to minimize training time. A large fraction of Big data machine learning applications involve periodically repeated model training involving newly added data. For example, in Advertising, logistic regression based click probability models are retrained on a daily basis on incrementally varying datasets. In such scenarios it is worthwhile to spend time to tune P in an early deployment phase to minimize time, and then use this choice of P for future runs.

4.9.5. Computation and Communication Costs

Table 2 shows the ratio of computational cost to communication cost for the three high dimensional datasets for all the methods.[17] Note that the ratio is small for *TERA* and so communication cost dominates the time for it. On the other hand, both the costs are well balanced for FADL. Note that ratio varies in the range of $0.625 - 2.845$. This clearly shows that FADL trades-off computation with communication, while significantly reducing the number of communication passes (Figures 6 and 7) and time (Figures 8 and 9).

	FADL	CoCoA	TERA	ADMM
kdd2010	1.6333	0.1416	0.1422	1.8499
url	1.3650	0.1040	0.2986	3.4886
webspam	1.2082	0.1570	0.2423	1.2543

Table 2: Ratio of the total computation cost to the total communication cost for various methods which were terminated when AUPRC reached within 0.1% of the AUPRC value for 128 nodes.

15. We choose FADL as an example, but the comments made in the discussion apply to other methods too.
16. The datasets, *kdd2010*, *url* and *webspam* are really not huge in the *Big data* sense. In this paper we used them only because of lack of availability of much bigger public datasets.
17. For the medium/low dimensional datasets *rcv* and *mnist8m*, communication latencies, line search cost etc. also play a key role and an analysis of computation cost versus communication cost does not provide any great insight.

4.10. Experiment on a much larger dataset

To verify the goodness of FADL, we also did an experiment evaluating FADL against other methods, on a large dataset that is more than an order of magnitude bigger than the largest dataset in Table 1. The dataset is the Splice site recognition dataset from the bioinformatics domain (Sonnenburg and Franc, 2010). In this dataset each example is a sequence; we considered all positional features upto 9 grams. This led to a dataset of 49 million features and 50 million examples. The size of the dataset is larger than 0.65 Terabytes. We employed a cluster of 100 nodes to solve this problem. Figure 12 compares the various methods on (a) the reduction of the objective function as a function of communication passes; (b) the reduction of the objective function as a function of time; and (c) the improvement of generalization performance (AUPRC) as a function of time. It is clear that FADL makes great reductions over other methods, on the number of communication passes. FADL is also the best performer when we measure by the time taken. On clusters with slower communication speeds and iteration set-up times, the value of FADL over other methods will be even higher. Interestingly, CoCoA comes out as the next best performer. CoCoA has slower end convergence than FADL, but it shows up equally well in plot (c) on the improvement of generalization performance. As we saw earlier with other datasets, the value of CoCoA varies a lot; for the current scenario of solving the splice site recognition on 100 nodes it seems to be well-suited. FADL, on the other hand, is uniformly good in varying scenarios of several datasets, number of nodes etc.

4.11. Summary

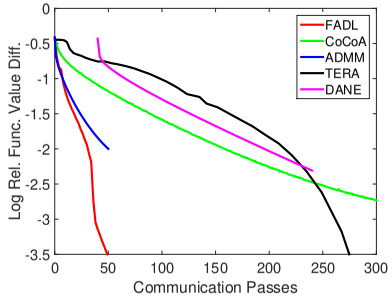
It is useful to summarize the findings of the empirical study.

- FADL gives a great reduction in the number of communication passes, making it clearly superior to other methods in communication heavy settings.
- In spite of higher computational costs per iteration FADL shows the overall best performance on the total time taken. This is true even for medium and low dimensional datasets.
- FADL shows a speed-up of 1-10 over TERA, the actual speed-up depending on the dataset and the setting.
- FADL nicely balances computation and communication costs.

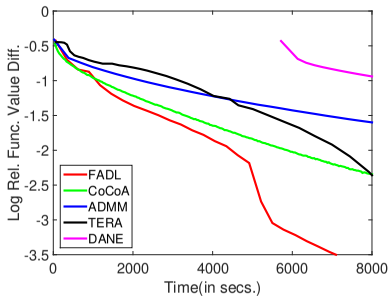
5. Discussion

In this section, we discuss briefly, other different distributed settings made possible by our algorithm. The aim is to show the flexibility and generality of our approach while ensuring *glrc*.

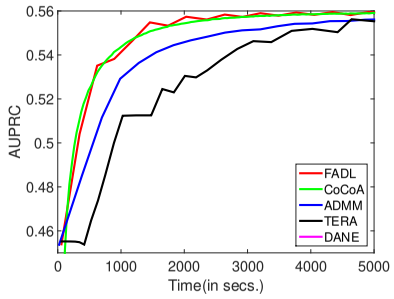
Section 3 considered example partitioning where examples are distributed across the nodes. First, it is worth mentioning that, due to the gradient consistency condition, *partitioning* is not a necessary constraint; our theory allows examples to be resampled, i.e., each example is allowed to be a part of any number of nodes arbitrarily. For example, to reduce the number of outer iterations, it helps to have more examples in each node.



(a) Communication efficiency (objective function)



(b) Time efficiency (objective function)



(c) Time efficiency (AUPRC)

Figure 12: Plots comparing various methods on the *Splice site recognition* dataset.

Second, the theory proposed in Section 3 holds for feature partitioning also. Suppose, in each node p we restrict ourselves to a subset of features, $J_p \subset \{1, \ldots, d\}$, i.e., include the constraint, $w_p \in \{w : w(j) = w^r(j) \;\; \forall r \notin J_p\}$, where $w(j)$ denotes the weight of the j^{th} feature. Note that we do not need $\{J_p\}$ to form a partition. This is useful since important features can be included in all the nodes.

Gradient sub-consistency. Given w^r and J_p we say that $\hat{f}_p(w)$ has gradient sub-consistency with f at w^r on J_p if $\frac{\partial f}{\partial w(j)}(w^r) = \frac{\partial \hat{f}}{\partial w(j)}(w^r) \;\; \forall\, j \in J_p$.

Under the above condition, we can modify the algorithm proposed in Section 3 to come up with a feature decomposition algorithm with *glrc*.

Several feature decomposition based approaches (Richtárik and Takác, 2012; Patriksson, 1998b) have been proposed in the literature. The one closest to our method is the work by Patrikkson on a synchronized parallel algorithm (Patriksson, 1998b) which extends a generic cost approximation algorithm (Patriksson, 1998a) that is similar to our functional approximation. The sub-problems on the partitions are solved in parallel. Although the objective function is not assumed to be convex, the cost approximation is required to satisfy

a monotone property, implying that the approximation is convex. The algorithm only has asymptotic linear rate of convergence and it requires the feature partitions to be disjoint. In contrast, our method has *glrc* and works even if features overlap in partitions. Moreover, there does not exist any counterpart of our example partitioning based distributed algorithm discussed in Section 3.

Recently Mairal (2013) has developed an algorithm called MISO. The main idea of MISO (which is in the spirit of the EM algorithm) is to build majorization approximations with good properties so that line search can be avoided, which is interesting. MISO is a serial method. Developing a distributed version of MISO is an interesting future direction; but, given that line search is inexpensive communication-wise, it is unclear if such a method would give great benefits.

Our approach can be easily generalized to joint example-feature partitioning as well as non-convex settings.[18] The exact details of all the extensions mentioned above and related experiments are left for future work.

Recently, a powerful divide and conquer approach Hsieh et al. (2014) has been suggested for training kernel methods. The idea is to partition the input space such that the restrictions of training on the partitioned input spaces are as decoupled as possible. If, in FADL, we had the ability to choose the parts of data that are placed in the nodes, then we would also gain by choosing decoupled partitions. However, in the distributed case, this requires pre-processing as well as shuffling of data, which are expensive.

6. Conclusion

To conclude, we have proposed FADL, a novel functional approximation based distributed algorithm with provable global linear rate of convergence. The algorithm is general and flexible in the sense of allowing different local approximations at the node level, different algorithms for optimizing the local approximation, early stopping and general data usage in the nodes. We also established the superior efficiency of FADL by evaluating it against key existing distributed methods. We believe that FADL has great potential for solving machine learning problems arising in Big data.

Appendix A: Complexity analysis

Let us use the notations of section 3 given around (21). We define the overall cost of any distributed algorithm as

$$[(c_1 \frac{nz}{P} + c_2 m) T^{\text{inner}} + c_3 \gamma m] T^{\text{outer}}, \tag{22}$$

where T^{outer} is the number of outer iterations, T^{inner} is the number of inner iterations at each node before communication happens and c_1 and c_2 denote the number of passes over the data and m-dimensional dot products per inner iteration respectively. For communication, we assume an *AllReduce* binary tree as described in Agarwal et al. (2011) with

18. For non-convex settings *glrc* is hard to establish, but proving a simpler convergence theory is quite possible.

pipelining. As a result, we do not have a multiplicative factor of $log_2 P$ in our cost[19]. γ is the relative computation to communication speed in the given distributed system; more precisely, it is the ratio of the times associated with communicating a floating point number and performing one floating point operation; γ is usually much larger than 1. c_3 is the number of m-dimensional vectors (gradients, Hessian-vector computations etc.) we need to communicate.

Method	c_1	c_2	c_3	T^{inner}
SQM	2	$\approx 5-10$	1	1
FADL	2	$\approx 5-7$	2	$\hat{k}$

Table 3: Value of cost parameters

The values of different parameters for SQM and FADL are given in Table 3. $T^{\text{outer}}_{\text{SQM}}$ is the number of overall conjugate gradient iterations plus gradient computations. $\hat{k}$ is the average number of conjugate gradient iterations (for the inner minimization of $\hat{f}_p$ using TRON) required per outer iteration in FADL. Typically $\hat{k}$ is between 5 and 20.

Since dense dot products are extremely fast $c_2 m$ is small compared to $c_1 nz/P$ for both the approaches, we ignore it from (22) for simplicity. Now for FADL to have lesser cost than TERA, we can use (22) to get the condition,

$$2.0(\hat{k}T^{\text{outer}}_{\text{FADL}} - T^{\text{outer}}_{\text{SQM}})\frac{nz}{P} \leq (T^{\text{outer}}_{\text{SQM}} - 2T^{\text{outer}}_{\text{FADL}})\gamma m \tag{23}$$

Let us ignore $T^{\text{outer}}_{\text{SQM}}$ on the left side of this inequality (in favor of SQM) and rearrange to get the looser condition,

$$\frac{nz}{m} \leq \frac{\gamma P}{\hat{k}}\frac{1}{2.0}(\frac{T^{\text{outer}}_{\text{SQM}}}{T^{\text{outer}}_{\text{FADL}}} - 2) \tag{24}$$

Assuming $T^{\text{outer}}_{\text{SQM}} > 3.0 T^{\text{outer}}_{\text{FADL}}$, we arrive at the final condition in (21).

Appendix B: Proofs

Proofs of the results in section 2

Let us now consider the establishment of the convergence theory given in section 2.

Proof of Lemma 1. Let $\rho(t) = f(w^r + td^r)$ and $\gamma(t) = \rho(t) - \rho(0) - \alpha t\rho'(0)$. Note the following connections with quantities involved in Lemma 1: $\rho(t) = f^{r+1}$, $\rho(0) = f^r$, $\rho'(t) = g^{r+1} \cdot d^r$ and $\gamma(t) = f^{r+1} - f^r - \alpha g^r \cdot (w^{r+1} - w^r)$. (4) corresponds to the condition $\gamma(t) \leq 0$ and (5) corresponds to the condition $\rho'(t) \geq \beta\rho'(0)$.

$\gamma'(t) = \rho'(t) - \alpha\rho'(0)$. $\rho'(0) < 0$. ρ' is strictly monotone increasing because, by assumption A2,

$$\rho'(t) - \rho'(\tilde{t}) \geq \sigma(t - \tilde{t})\|d^r\|^2 \;\; \forall\, t, \tilde{t} \tag{25}$$

This implies that γ' is also strictly monotone increasing and, all four, ρ, ρ', γ' and γ tend to infinity as t tends to infinity.

19. Actually, there is another communication term, $\gamma b\, log_2 P$, where b is the size of first block of communicated doubles in the pipeline. However, typically $b \ll m$ and hence we ignore it.

Let t_β be the point at which $\rho'(t) = \beta\rho'(0)$. Since $\rho'(0) < 0$ and ρ' is strictly monotone increasing, t_β is unique and $t_\beta > 0$. This validates the definition in (6). Monotonicity of ρ' implies that (5) is satisfied iff $t \geq t_\beta$.

Note that $\gamma(0) = 0$ and $\gamma'(0) < 0$. Also, since γ' is monotone increasing and $\gamma(t) \to \infty$ as $t \to \infty$, there exists a unique $t_\alpha > 0$ such that $\gamma(t_\alpha) = 0$, which validates the definition in (7). It is easily checked that $\gamma(t) \leq 0$ iff $t \in [0, t_\alpha]$.

The properties also imply $\gamma'(t_\alpha) > 0$, which means $\rho'(t_\alpha) \geq \alpha\rho'(0)$. By the monotonicity of ρ' we get $t_\alpha > t_\beta$, proving the lemma.

Proof of Theorem 2. Using (5) and A1,

$$(\beta - 1)g^r \cdot d^r \leq (g^{r+1} - g^r) \cdot d^r \leq Lt\|d^r\|^2 \tag{26}$$

This gives a lower bound on t:

$$t \geq \frac{(1-\beta)}{L\|d^r\|^2}(-g^r \cdot d^r) \tag{27}$$

Using (4), (27) and (1) we get

$$f^{r+1} \leq f^r + \alpha t g^r \cdot d^r \leq f^r - \frac{\alpha(1-\beta)}{L\|d^r\|^2}(-g^r \cdot d^r)^2 \leq f^r - \frac{\alpha(1-\beta)}{L}\cos^2\theta\|g^r\|^2 \tag{28}$$

Subtracting $f^\star$ gives

$$(f^{r+1} - f^\star) \leq (f^r - f^\star) - \frac{\alpha(1-\beta)}{L}\cos^2\theta\|g^r\|^2 \tag{29}$$

A2 together with $g(w^\star) = 0$ implies $\|g^r\|^2 \geq \sigma^2\|w^r - w^\star\|^2$. Also A1 implies $f^r - f^\star \leq \frac{L}{2}\|w^r - w^\star\|^2$ Smola and Vishwanathan (2008). Using these in (29) gives

$$\begin{aligned}(f^{r+1} - f^\star) &\leq (f^r - f^\star) - 2\alpha(1-\beta)\frac{\sigma^2}{L^2}\cos^2\theta(f^r - f^\star) \\ &\leq (1 - 2\alpha(1-\beta)\frac{\sigma^2}{L^2}\cos^2\theta)(f^r - f^\star)\end{aligned} \tag{30}$$

Let $\delta = (1 - 2\alpha(1-\beta)\frac{\sigma^2}{L^2}\cos^2\theta)$. Clearly $0 < \delta < 1$. Theorem 2 follows.

Proofs of the results in section 3

Let us now consider the establishment of the convergence theory given in section 3. We begin by establishing that the exact minimizer of $\hat{f}_p$ makes a sufficient angle of descent at w^r.

Lemma 5. Let $\hat{w}^\star_p$ be the minimizer of $\hat{f}_p$. Let $d_p = (\hat{w}^\star_p - w^r)$. Then

$$-g^r \cdot d_p \geq (\sigma/L)\|g^r\|\|d_p\| \tag{31}$$

Proof. First note, using gradient consistency and $\nabla f_p(\hat{w}^\star_p) = 0$ that

$$\|g^r\| = \|\nabla\hat{f}_p(w^r) - \nabla\hat{f}_p(\hat{w}^\star_p)\| \leq L\|d_p\| \tag{32}$$

Now,

$$-g^r \cdot d_p = (\nabla \hat{f}_p(w^r) - \nabla \hat{f}_p(\hat{w}_p^\star))^T (w^r - \hat{w}_p^\star) \geq \sigma \|d_p\|^2 = \quad \sigma \|g^r\| \|d_p\| \frac{\|d_p\|}{\|g^r\|} \geq \frac{\sigma}{L} \|g^r\| \|d_p\| \quad (33)$$

where the second line comes from σ-strong convexity and the fourth line follows from (32).

Proof of Lemma 3. Let us now turn to the question of approximate stopping and establish Lemma 3. Given θ satisfying (18) let us choose $\zeta \in (0, 1)$ such that

$$\frac{\pi}{2} > \theta > \cos^{-1} \frac{\sigma}{L} + \cos^{-1} \zeta \quad (34)$$

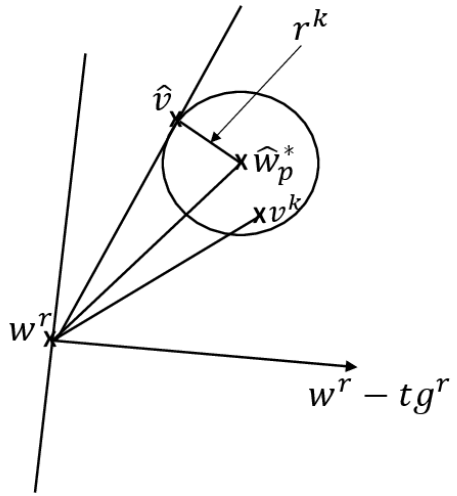


Figure 13: Construction used in the proof of Lemma 3.

By A3 and equations (3.16) and (3.22) in Smola and Vishwanathan (2008), we get

$$\frac{\sigma}{2} \|v - \hat{w}_p^\star\|^2 \leq \hat{f}_p(v) - \hat{f}_p^\star \leq \frac{L}{2} \|v - \hat{w}_p^\star\|^2 \quad (35)$$

After k iterations we have

$$\hat{f}_p(v^k) - \hat{f}_p^\star \leq \delta^k (\hat{f}_p(w^r) - \hat{f}_p^\star) \quad (36)$$

We can use these to get

$$\|v^k - \hat{w}_p^\star\|^2 \leq \frac{2(\hat{f}_p(v^k) - \hat{f}_p^\star)}{\sigma} \leq \frac{2\delta^k (\hat{f}_p(w^r) - \hat{f}_p^\star)}{\sigma} \leq \frac{\delta^k L}{\sigma} \|w^r - \hat{w}_p^\star\|^2 \stackrel{\text{def}}{=} (r^k)^2 \quad (37)$$

For now let us assume the following:

$$\|v^k - \hat{w}_p^\star\|^2 \leq \|w^r - \hat{w}_p^\star\|^2 \quad (38)$$

Using (37) note that (38) holds if

$$\frac{\delta^k L}{\sigma} \leq 1 \quad (39)$$

Let S^k be the sphere, $S^k = \{v : \|v - \hat{w}_p^\star\|^2 \leq (r^k)^2\}$. By (37) we have $v^k \in S^k$. See Figure 6. Therefore,

$$\phi^k \leq \max_{v \in S^k} \phi(v) \quad (40)$$

where ϕ^k is the angle between $\hat{w}_p^\star - w^r$ and $v^k - w^r$, and $\phi(v)$ is the angle between $v - w^r$ and $\hat{w}_p^\star - w^r$. Given the simple geometry, it is easy to see that $\max_{v \in S^k} \phi(v)$ is attained by a point $\hat{v}$ lying on the boundary of S^k (i.e., $\|\hat{v} - \hat{w}_p^\star\|^2 = (r^k)^2$) and satisfying $(\hat{v} - \hat{w}_p^\star) \perp (\hat{v} - w^r)$. This geometry yields

$$\cos^2 \phi(\hat{v}) = \frac{\|\hat{v} - w^r\|^2}{\|\hat{w}_p^\star - w^r\|^2} = \frac{\|\hat{w}_p^\star - w^r\|^2 - (r^k)^2}{\|\hat{w}_p^\star - w^r\|^2} = 1 - \frac{(r^k)^2}{\|\hat{w}_p^\star - w^r\|^2} = 1 - \frac{\delta^k L}{\sigma} \quad (41)$$

Since $\phi^k \leq \phi(\hat{v})$,

$$\cos^2 \phi^k \geq 1 - \frac{\delta^k L}{\sigma} \quad (42)$$

Thus, if

$$1 - \frac{\delta^k L}{\sigma} \geq \zeta^2 \quad (43)$$

then

$$\cos \phi^k \geq \zeta \quad \forall k \geq \hat{k} \quad (44)$$

holds. By (34) this yields $\angle(-g^r, v^k - w^r) \leq \theta$, the result needed in Lemma 3. Since $\zeta > 0$, (43) implies (39), so (38) holds and there is no need to separately satisfy it. Now (43) holds if

$$k \geq \hat{k} \stackrel{\text{def}}{=} \frac{\log(L/(\sigma(1 - \zeta^2)))}{\log(1/\delta)} \quad (45)$$

which proves the lemma.

Proof of Theorem 4. It trivially follows from a combination of Lemma 3 and Theorem 2.

References

A. Agarwal, O. Chapelle, M. Dudik, and J. Langford. A reliable effective terascale linear learning system. In *arXiv:1140.4198*, 2011.

A. Agarwal, O Chapelle, M. Dudik, and J. Langford. A reliable effective terascale linear system. *arXiv:1110.4198*, 2013.

D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and distributed computation: Numerical methods.* Athena Scientific, Cambridge, MA, 1997.

L. Bottou. Large-scale machine learning with stochastic gradient descent. In *COMPSTAT'2010*, pages 177–187, 2010.

S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, UK, 2004.

S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, pages 1–122, 2011.

R. H. Byrd, G. M. Chin, W. Neveitt, and J. Nocedal. On the use of stochastic Hessian information in optimization methods for machine learning. *SIAM Journal of Optimization*, pages 977–995, 2012.

K.W. Chang, C.J. Hsieh, and C.J. Lin. Coordinate descent method for large-scale l2-loss linear SVM. *JMLR*, pages 1369–1398, 2008.

C.T. Chu, S.K. Kim, Y.A. Lin, Y.Y. Yu, G. Bradski, A.Y. Ng, and K. Olukotun. Map-reduce for machine learning on multicore. *NIPS*, pages 281–288, 2006.

J. Dean and S. Ghemawat. Mapreduce: Simplified data processing on large clusters. *Commun. ACM*, pages 107–113, 2008.

W. Deng and W. Yin. On the global linear convergence of the generalized alternating direction method of multipliers. *Rice University CAAM Technical Report*, TR12-14, 2012.

K.B. Hall, S. Gilpin, and G. Mann. Mapreduce/bigtable for distributed optimization. In *NIPS Workshop on Leaning on Cores, Clusters, and Clouds*, 2010.

C.J. Hsieh, K.W. Chang, C.J. Lin, S.S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *ICML*, pages 408–415, 2008.

C.J. Hsieh, S. Si, and I.S. Dhillon. A divide-and-conquer solver for kernel support vector machines. *ICML*, 2014.

M. Jaggi, V. Smith, M. Takáč, J. Terhorst, S. Krishnan, T. Hofmann, and M.I. Jordan. Communication-efficient distributed dual coordinate ascent. *arXiv:1409.1458*, 2014.

R. Johnson and T. Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *NIPS*, 2013.

C.J. Lin, R.C. Weng, and S.S. Keerthi. Trust region newton method for large-scale logistic regression. *JMLR*, pages 627–650, 2008.

D. Mahajan, S. S. Keerthi, S. Sundararajan, and L. Bottou. A functional approximation based distributed learning algorithm. *arXiv:1310.8418*, 2013a.

D. Mahajan, S. S. Keerthi, S. Sundararajan, and L. Bottou. A parallel SGD method with strong convergence. *NIPS Workshop on Optimization in Machine Learning*, 2013b.

J. Mairal. Optimization with first order surrogate functions. *ICML*, 2013.

G. Mann, R.T. McDonald, M. Mohri, N. Silberman, and D. Walker. Efficient large-scale distributed training of conditional maximum entropy models. In *NIPS*, pages 1231–1239, 2009.

R.T. McDonald, K. Hall, and G. Mann. Distributed training strategies for the structured perceptron. In *HLT-NAACL*, pages 456–464, 2010.

M. Patriksson. Cost approximation: A unified framework of descent algorithms for nonlinear programs. *SIAM J. on Optimization*, 8:561–582, 1998a.

M. Patriksson. Decomposition methods for differentiable optimization problems over cartesian product sets. *Comput. Optim. Appl.*, 9:5–42, 1998b.

D. Pechyony, L. Shen, and R. Jones. Solving large scale linear SVM with distributed block minimization. *NIPS workshop on Big Learning*, 2011.

P. Richtárik and M. Takác. Parallel coordinate descent methods for big data optimization. *CoRR*, abs/1212.0873, 2012.

O. Sharir, N. Srebro, and T. Zhang. Communication efficient distributed optimization using an approximate newton-type method. *arXiv:1312.7853v4*, 2014.

J.R. Shewchuk. An introduction to the conjugate gradient method without the agonizing pain. Technical report, 1994.

A. Smola and S.V.N. Vishwanathan. *Introduction to Machine Learning*. Cambridge University Press, Cambridge, UK, 2008.

S. T. Sonnenburg and V. Franc. COFFIN: a computational framework for linear SVMs. *ICML*, 2010.

P.W. Wang and C.J. Lin. Iteration complexity of feasible descent methods for convex optimization. *Technical Report, National Taiwan University*, 2013.

M. Weimer, Y. Chen, B.G. Chun, T. Condie, C. Curino, C. Douglas, Y. Lee, T. Majestro, D. Malkhi, S. Matusevych, et al. Reef: Retainable evaluator execution framework. In *ACM SIGMOD*, pages 1343–1355, 2015.

P. Wolfe. Convergence conditions for ascent methods. *SIAM Review*, 11:226–235, 1969.

P. Wolfe. Convergence conditions for ascent methods: II: Some corrections. *SIAM Review*, 13:185–188, 1971.

T. Yang. Trading computation for communication: distributed stochastic dual coordinate ascent. *NIPS*, 2013.

T. Yang, S. Zhu, R. Jin, and Y. Lin. Analysis of distributed stochastic dual coordinate ascent. *arXiv:1312.1031*, 2013.

M. Zaharia, M. Chowdhury, M.J. Franklin, S. Shenker, and I. Stoica. Spark: Cluster computing with working sets. In *USENIX Conference*, 2010.

C. Zhang, H. Lee, and K.G. Shin. Efficient distributed linear classification algorithms via the alternating direction method of multipliers. *CIKM*, 2012.

Y. Zhang and L. Xiao. DiSCO: Communication-efficient distributed optimization of self-concordant loss. *ICML*, 2015.

M. Zinkevich, M. Weimer, A. Smola, and L. Li. Parallelized stochastic gradient descent. In *NIPS*, pages 2595–2603, 2010.

Journal of Machine Learning Research 19 (2018) 1-26 Submitted 11/16; Revised 11/18; Published 11/18

Sparse Estimation in Ising Model via Penalized Monte Carlo Methods

Błażej Miasojedow bmia@mimuw.edu.pl
Institute of Applied Mathematics and Mechanics
University of Warsaw
ul. Banacha 2, 02-097, Warszawa, Poland
and
Institute of Mathematics
Polish Academy of Sciences
ul. Śniadeckich 8, 00-656 Warszawa

Wojciech Rejchel wrejchel@gmail.com
Faculty of Mathematics and Computer Science
Nicolaus Copernicus University
ul. Chopina 12/18, 87-100 Toruń, Poland
and
Institute of Applied Mathematics and Mechanics
University of Warsaw
ul. Banacha 2, 02-097, Warszawa, Poland

Editor: Sara van de Geer

Abstract

We consider a model selection problem in high-dimensional binary Markov random fields. The usefulness of the Ising model in studying systems of complex interactions has been confirmed in many papers. The main drawback of this model is the intractable norming constant that makes estimation of parameters very challenging. In the paper we propose a Lasso penalized version of the Monte Carlo maximum likelihood method. We prove that our algorithm, under mild regularity conditions, recognizes the true dependence structure of the graph with high probability. The efficiency of the proposed method is also investigated via numerical studies.

Keywords: Ising model, Monte Carlo Markov chain, Markov random field, model selection, Lasso penalty

1. Introduction

A Markov random field is an undirected graph (V, E), where $V = \{1, \ldots, d\}$ is a set of vertices and $E \subset V \times V$ is a set of edges. The structure of this graph describes conditional independence among subsets of a random vector $Y = (Y(1), \ldots, Y(d))$, where a random variable $Y(s)$ is associated with a vertex $s \in V$. Finding interactions between random variables is a central element of many branches of science, for example biology, genetics, physics or social network analysis. The goal of the current paper is to recognize the structure of a graph on the basis of a sample consisting of n independent observations. We consider the high-dimensional setting, i.e. the number of vertices d can be comparable or larger than the sample size n. It is motivated by contemporary applications of Markov random fields in the above-mentioned places, for instance gene microarray data.

The Ising model (Ising, 1925) is an important example of a mathematical model that is often used to explain relations between discrete random variables. In the literature one can find many papers that argue for its effectiveness in recognizing the structure of a graph (Ravikumar et al., 2010; Höfling and Tibshirani, 2009; Guo et al., 2010; Xue et al., 2012; Jalali et al., 2011). This model also plays a key role in our paper. On the other hand, the Ising model is an example of an intractable constant model that is the joint distribution of Y is known only up to a norming constant and this constant cannot be calculated in practice.

Thus, there are two main difficulties in the considered model. The first one is the high-dimensionality of the problem. The second one is the intractable norming constant. To overcome the first obstacle we apply a well-known Lasso method (Tibshirani, 1996). The properties of this method in model selection are deeply studied in many papers that mainly investigate linear models or generalized linear models (Bickel et al., 2009; Bühlmann and van de Geer, 2011; Huang and Zhang, 2012; van de Geer, 2008; Ye and Zhang, 2010; Zhao and Yu, 2006; Zhou, 2009). However, it is not difficult to find papers that describe properties of Lasso estimators in more complex models, for instance Markov random fields (Banerjee et al., 2008; Bühlmann and van de Geer, 2011; Ravikumar et al., 2010; Höfling and Tibshirani, 2009; Guo et al., 2010; Xue et al., 2012) that are considered in this paper.

There are many approaches trying to overcome the second obstacle that is the intractable norming constant. For instance, in Ravikumar et al. (2010) one proposes to perform d regularized logistic regression problems. This idea is based on the fact that the norming constant reduces, if one considers the conditional distribution instead of the joint distribution in the Ising model. This simple fact is at the heart of the pseudolikelihood approach (Besag, 1974) that is replacing the likelihood (that contains the norming constant) by the product of conditionals (that do not contain the norming constant). This idea is widely applied in the literature (Höfling and Tibshirani, 2009; Guo et al., 2010; Xue et al., 2012; Jalali et al., 2011) to study model selection properties of high-dimensional Ising models. However, this approach works well only if the pseudolikelihood is a good approximation of the likelihood. In general, it depends on the true structure of a graph. Namely, if this structure of the graph is sufficiently simple (examples of different structures can be found in section 5.1), then the product of conditionals should be close to the joint distribution. However, in practice this knowledge is unavailable. Another approach is described in Banerjee et al. (2008). It adapts the method that estimates the precision matrix in gaussian graphical models to the binary case. In the current paper we propose the approach to the norming constant problem that relates to Markov chain Monte Carlo (MCMC) methods. Namely, the norming constant is approximated using the importance sampling technique. This method is independent of the unknown complexity of the estimated graph. It is sufficient that the size of a sample used in importance sampling is sufficiently large to have good approximation of the likelihood.

The MCMC method is a well-known approach to overcome the problem with the intractable norming constant in classical (low-dimensional) estimation of graphs. For instance, its properties are investigated in influential papers Geyer and Thompson (1992); Geyer (1994). In the high-dimensional Ising model these algorithms were also studied. For

example, Honorio (2012) and Atchadé et al. (2017) analyzed stochastic versions of proximal gradient algorithms. Both papers derive nonasymptotic bounds between the output of the algorithm and the true minimizer of the cost function. However, in the current paper we focus on model selection properties of MCMC methods. We investigate them in the high-dimensional scenario and compare to the existing methods that are mentioned above. Model selection for undirected graphical models means finding the existing edges in the "sparse" graph that is a graph having relatively few edges (comparing to the total number of possible edges $\frac{d(d-1)}{2}$ and the sample size n).

The paper is organized as follows: in the next section we describe the Ising model and our approach to the problem that relates to minimization of the penalized MCMC approximation of the likelihood. The literature concerning this topic is also discussed. In section 3 we state main theoretical results. Details of efficient implementation are given in section 4, while the results of numerical studies are presented in section 5. The conclusions can be found in section 6. Finally, the proofs are postponed to appendices A and B.

2. Model description and related works

In this section we introduce the Ising model and the proposed method. It also contains a review of the literature relating to this problem.

2.1. Ising Model and undirected graphs

Let (V, E) be an undirected graph that consists of a set of vertices V and a set of edges E. The random vector $Y = (Y(1), Y(2), \ldots, Y(d))$, that takes values in $\mathcal{Y}$, is associated with this graph. In the paper we consider a special case of the Ising model that $Y(s) \in \{-1, 1\}$ and the joint distribution of Y is given by the formula

$$p(y|\theta^\star) = \frac{1}{C(\theta^\star)} \exp\left(\sum_{r<s} \theta^\star_{rs} y(r) y(s)\right), \tag{1}$$

where the sum in (1) is taken over such pairs of indices $(r, s) \in \{1, \ldots, d\}^2$ that $r < s$. The vector $\theta^\star \in \mathbb{R}^{d(d-1)/2}$ is a true parameter and $C(\theta^\star)$ is a norming constant, i.e.

$$C(\theta^\star) = \sum_{y \in \mathcal{Y}} \exp\left(\sum_{r<s} \theta^\star_{rs} y(r) y(s)\right).$$

The norming constant is a finite sum but it consists of 2^d elements that makes it intractable even for a moderate size of d.

For convenience, we denote $J(y) = (y(r)y(s))_{r<s}$, so

$$p(y|\theta^\star) = \frac{1}{C(\theta^\star)} \exp\left[(\theta^\star)' J(y)\right].$$

Remark 1 *The model* (1) *is a simplified version of the Ising model, for instance we omit an external field in* (1). *We have decided to restrict to the model containing only parameters* $\theta^\star_{rs}$, *because interactions between random variables is what we focus on in the current paper. However, our results can be relatively easy extended.*

The Ising model has the following property: vertices r and s are not connected by an edge (i.e. $\theta^\star_{rs} = 0$) means that variables $Y(r)$ and $Y(s)$ are conditionally independent given the other vertices. Therefore, we recognize the structure of the graph (its edges) by estimating the parameter $\theta^\star$. Assume that $Y_1, \ldots, Y_n$ are independent random vectors from the model (1). Then the negative log-likelihood is

$$\ell_n(\theta) = -\frac{1}{n} \sum_{i=1}^n \theta' J(Y_i) + \log C(\theta). \tag{2}$$

The second term in (2) contains the norming constant so we cannot use (2) to estimate $\theta^\star$. To overcome this problem one usually replaces the negative log-likelihood by its approximation and estimates $\theta^\star$ using the minimizer of this approximation. In the current paper the approximation of (2) is based on Monte Carlo (MC) methods. Suppose that $h(y)$ is an importance sampling distribution and note that

$$C(\theta) = \sum_{y \in \mathcal{Y}} \exp\left[\theta' J(y)\right] = \sum_{y \in \mathcal{Y}} \frac{\exp\left[\theta' J(y)\right]}{h(y)} h(y) = \mathbb{E}_{Y \sim h} \frac{\exp\left[\theta' J(Y)\right]}{h(Y)} \tag{3}$$

for each θ. An MC approximation of the norming constant is

$$\frac{1}{m} \sum_{k=1}^m \frac{\exp\left[\theta' J(Y^k)\right]}{h(Y^k)}, \tag{4}$$

where $Y^1, \ldots, Y^m$ is a sample drawn from h or, which is more realistic and is considered in the current paper, $Y^1, \ldots, Y^m$ is a Markov chain with h being a density of its stationary distribution. Thus, the MCMC approximation of (2) is

$$\ell^m_n(\theta) = -\frac{1}{n} \sum_{i=1}^n \theta' J(Y_i) + \log\left(\frac{1}{m} \sum_{k=1}^m \frac{\exp\left[\theta' J(Y^k)\right]}{h(Y^k)}\right). \tag{5}$$

A natural choice of the importance sampling distribution is $h(y) = p(y|\psi)$ for some parameter ψ. It leads to

$$\ell^m_n(\theta) = -\frac{1}{n} \sum_{i=1}^n \theta' J(Y_i) + \log\left(\frac{1}{m} \sum_{k=1}^m \exp\left[(\theta - \psi)' J(Y^k)\right]\right) + \log(C(\psi)). \tag{6}$$

The last term in (6), that contains the unknown constant $C(\psi)$, does not depend on θ, so it can be ignored while minimizing (6).

Our goal is selecting the true model (recognizing edges of a graph) in the high-dimensional setting. It means that the number of vertices d can be large. In fact, it can be greater than the sample size, i.e. $d = d_n \gg n$. To estimate the vector $\theta^\star$ we use penalized empirical risk minimization. The natural choice of the penalty would be the l_0-penalty, but it makes the procedure nonconvex and computationally expensive even for moderate values of d. To avoid such problems, we use the Lasso penalty and minimize a function

$$\ell^m_n(\theta) + \lambda^m_n |\theta|_1, \tag{7}$$

where $|\theta|_1 = \sum_{r<s} |\theta_{rs}|$ and $\lambda_n^m > 0$ is a smoothing parameter, that is a balance between minimizing the MCMC approximation and the penalty. We denote the minimizer of (7) by $\hat{\theta}_n^m$. Notice that the function (7), that we minimize, is convex in θ, because the Lasso penalty as well as the MCMC approximation (6) are convex function in θ. The latter follows from the fact that the Hessian of $\ell_n^m(\theta)$, that is given explicitly in (19), is a weighted covariance matrix with positive weights that sum up to one. Convexity of the problem is important from the practical and theoretical point of view. First, every minimum of a convex function is the global minimum, so there are no local minimum problems. Second, convexity is also utilized in the proofs of the results contained in the paper. In further parts of the paper we study properties of $\hat{\theta}_n^m$ in model selection.

2.2. Related works

Model selection in the high-dimensional Ising model is a popular topic and many papers investigating this problem using different methods can be found in the literature (Banerjee et al., 2008; Bresler et al., 2008; Lee et al., 2006; Ravikumar et al., 2010; Höfling and Tibshirani, 2009; Guo et al., 2010; Xue et al., 2012; Jalali et al., 2011). The significant part of them uses the pseudolikelihood approximation with the Lasso penalty. For instance, Ravikumar et al. (2010) applies it by considering d logistic regression problems. They prove that this algorithm is model selection consistent, if some regularity conditions are satisfied. These conditions are similar to the "irrepresentable conditions" (Zhao and Yu, 2006), that are sufficient to prove an analogous property in the linear model. The pseudolikelihood method with the Lasso as a "joint" procedure is proposed in Höfling and Tibshirani (2009). Moreover, in the same paper one also proposes an "exact" algorithm, that minimizes the negative log-likelihood with the Lasso penalty. However, this procedure also bases on the pseudolikelihood approximation. Model selection consistency of the latter algorithm has not been studied yet. The former procedure has this property, that is showed in Guo et al. (2010) provided that conditions similar to Ravikumar et al. (2010) are satisfied. In Xue et al. (2012) the Lasso penalty is replaced by the SCAD penalty (Fan and Li, 2001) and theoretical properties of this algorithm are studied. In Jalali et al. (2011) one replaces restrictive irrepresentable conditions by weaker restricted strong convexity and smoothness conditions (Negahban et al., 2009) and proves model selection consistency of an algorithm that joints ideas from Ravikumar et al. (2010) and Zhang (2009). Namely, it performs d separate logistic regression problems with the forward-backward greedy approach. The algorithm described in Banerjee et al. (2008) is also based on the likelihood approximation with the Lasso penalty. However, it does not apply the pseudolikelihood method. Using the determinant relaxation (Wainwright and Jordan, 2006) it treats the problem of model selection in discrete Markov random fields analogously to the continuous case.

In the current paper we apply the MCMC method to overcome the intractable norming constant problem. Our experimental study (presented in section 5) confirms that estimators based on the MCMC approximation usually perform comparably or better in model selection than their competitors from Banerjee et al. (2008); Höfling and Tibshirani (2009); Ravikumar et al. (2010). Our theoretical results are similar to those described in the previous paragraph, that is we prove model selection consistency. But, in general, our assumptions

are weaker than their analogs from the above-mentioned papers. The detailed comparison is given after Corollary 3 in section 3.

Moreover, the advantage of our algorithm is that the MCMC approach allows us to approximate the norming constant with an arbitrary precision. The approximation error of other methods is given by the problem/data. It depends on the unknown structure of a graph and a user cannot improve it. In our approach a user can improve approximation by increasing the length of simulation, using the MCMC algorithm tailored to the problem. The obvious drawback of our approach is the need of additional simulations to obtain the MCMC sample. It makes our procedure computationally more complex, but at the same time more accurate in selecting the true model.

2.3. Notation

In further parts of the paper we need few notations. Most of them are collected in this subsection.

For simplicity, we write $\hat{\theta}$ and λ instead of $\hat{\theta}_n^m$ and λ_n^m, respectively. Besides, we denote the number of estimated parameters in the model by $\bar{d} = d(d-1)/2$. Nonzero coordinates of $\theta^\star$ are collected in the set T, and T^c is a completion of T. Besides, $\bar{d}_0 = |T|$ denotes the number of elements in the set T.

For a vector a we denote its l_∞-norm by $|a|_\infty = \max_k |a_k|$ and $a^{\otimes 2} = aa'$. The vector a_T is the same as the vector a on T and zero otherwise. The l_∞-norm of a matrix Σ is denoted by $|\Sigma|_\infty = \max_{k,l} |\Sigma_{kl}|$.

Let us consider a Markov chain on a space $\mathcal{S}$ with a transition kernel $P(x, \cdot)$ and a stationary distribution π. We define the Hilbert space $L^2(\pi)$ as a space of functions that $\pi(f^2) < \infty$ and the inner product is given as $\langle f, g \rangle = \int_{\mathcal{S}} f(x) g(x) \pi(dx)$. The linear operator P on $L^2(\pi)$ associated with the transition kernel $P(x, \cdot)$ is defined as follows

$$Pf(x) = \int_{\mathcal{S}} f(y) P(x, dy) \,.$$

We say that the Markov chain has a spectral gap $1 - \kappa$ if and only if

$$\kappa = \sup\{|\rho| \colon \rho \in Spec(P) \setminus \{1\}\} \,,$$

where $Spec(\cdot)$ denotes the spectrum of an operator in $L^2(\pi)$. For reversible chains the spectral gap property is equivalent to geometric ergodicity of the chain (Kontoyiannis and Meyn, 2012; Roberts and Rosenthal, 1997).

In the paper we focus on the Gibbs sampler for the Ising model. However, theoretical results remain true for other MCMC algorithms as long as the spectral gap property is satisfied. The random scan Gibbs sampler for the Ising model with a joint distribution $p(y|\psi)$ is defined as follows: given Y^{k-1}, first we sample uniformly index r and we draw $Y^k(r)$ from the distribution

$$\mathbb{P}(Y^k(r) = 1) = \frac{\exp\{\psi' J(Y^+)\}}{\exp\{\psi' J(Y^+)\} + \exp\{\psi' J(Y^-)\}} \,, \tag{8}$$

where $Y^+(s) = Y^-(s) = Y^{k-1}(s)$ for $s \neq r$ and $Y^+(r) = 1$, $Y^-(r) = -1$. For $s \neq r$ we set $Y^k(s) = Y^{k-1}(s)$.

Suppose that $Y^1, \ldots, Y^m$ is a Markov chain on $\mathcal{Y}$ generated by a random scan Gibbs sampler defined as above. By construction the chain is irreducible, aperiodic and $h(y) = p(y|\psi)$ is the density of its stationary distribution. Therefore, the stationary distribution is defined uniquely and the chain is ergodic for any initial measure ν with the density q. Moreover, there exists a spectral gap $1-\kappa$, because the state space is finite and the chain is reversible. Actually, κ is the second greatest absolute value of eigenvalues of the transition matrix. We will need three quantities related to this Markov chain :

$$\beta_1 = \sqrt{\sum_{y\in\mathcal{Y}} \frac{q^2(y)}{h(y)}}, \quad \beta_2 = \frac{1-\kappa}{1+\kappa}, \quad M = \max_{y\in\mathcal{Y}} \frac{\exp((\theta^\star)' J(y))}{h(y) C(\theta^\star)}\,. \tag{9}$$

Roughly speaking, these three values can be viewed as: β_1 – how close the initial density is to the stationary one, β_2 – how fast the chain "mixes", M – how close the importance sampling density is to the true density (1).

3. Main results

In this section we state key results of the paper. In the first one (Theorem 2) we show that the estimation error of the minimizer of the MCMC approximation with the Lasso penalty can be controlled. In the second result (Corollary 3) we prove model selection consistency for the thresholded Lasso estimator (Zhou, 2009).

First, we introduce the cone invertibility factor that plays an important role in investigating properties of Lasso estimators. It is defined analogously to Ye and Zhang (2010); Huang and Zhang (2012); Huang et al. (2013) that concerns linear regression, generalized linear models and the Cox model, respectively. It is also closely related to the compatibility condition (van de Geer, 2008) or the restricted eigenvalue condition (Bickel et al., 2009). Thus, for $\xi > 1$ and the set T we define a cone as

$$\mathcal{C}(\xi, T) = \{\theta : |\theta_{T^c}|_1 \le \xi |\theta_T|_1\}\,.$$

For a nonnegative definite matrix Σ the cone invertibility factor is

$$F(\xi, T, \Sigma) = \inf_{0\neq\theta\in\mathcal{C}(\xi,T)} \frac{\theta'\Sigma\theta}{|\theta_T|_1 |\theta|_\infty}\,.$$

Cone invertibility factors of Hessians of two functions are crucial in our argumentation. The first function is the expectation of the negative log-likelihood (2), i.e.

$$\mathbb{E}\ell_n(\theta) = -\theta'\mathbb{E}J(Y) + \log C(\theta) \tag{10}$$

and the second one is the MCMC approximation (5). We denote them as

$$F(\xi, T) = \inf_{0\neq\theta\in\mathcal{C}(\xi,T)} \frac{\theta'\nabla^2 \log C(\theta^\star)\theta}{|\theta_T|_1 |\theta|_\infty} \tag{11}$$

and

$$\bar{F}(\xi, T) = \inf_{0\neq\theta\in\mathcal{C}(\xi,T)} \frac{\theta'\nabla^2 \ell_n^m(\theta^\star)\theta}{|\theta_T|_1 |\theta|_\infty}\,, \tag{12}$$

respectively. Notice that only the values of $\nabla^2 \log C(\theta)$ and $\nabla^2 \ell_n^m(\theta)$ at the true parameter $\theta^\star$ are taken into consideration in (11) and (12).

Now we can state main results of the paper.

Theorem 2 *Let $\varepsilon > 0, \xi > 1$ and $\alpha(\xi) = 2 + \frac{e}{\xi - 1}$. If*

$$n \ge \frac{8(1+\xi)^4\, \alpha^2(\xi)\, \bar{d}_0^2\, \log(2\bar{d}/\varepsilon)}{F^2(\xi, T)}\,, \tag{13}$$

$$m \ge \frac{64(1+\xi)^4\, \alpha^2(\xi)\, \bar{d}_0^2\, M^2\, \log\left[2\bar{d}(\bar{d}+1)\beta_1/\varepsilon\right]}{F^2(\xi, T)\beta_2}\,, \tag{14}$$

then with probability at least $1 - 4\varepsilon$ we have the inequality

$$\left|\hat{\theta} - \theta^\star\right|_\infty \le \frac{2e\,\xi\,\alpha(\xi)\lambda}{(\xi+1)[\alpha(\xi)-2]F(\xi, T)}\,, \tag{15}$$

where

$$\lambda = \frac{\xi+1}{\xi-1} \max\left(2\sqrt{\frac{2\log(2\bar{d}/\varepsilon)}{n}}, 8M\sqrt{\frac{\log\left[(2\bar{d}+1)\beta_1/\varepsilon\right]}{m\beta_2}}\right)\,. \tag{16}$$

Corollary 3 *Suppose that conditions (13) and (14) are satisfied. Let $\theta^\star_{min} = \min_{(r,s)\in T} |\theta^\star_{rs}|$ and R_n^m denote the right-hand side of the inequality (15). Consider the Lasso estimator with a threshold $\delta > 0$ that is the set of nonzero coordinates of the final estimator is defined as $\hat{T} = \{(r,s) : |\hat{\theta}_{rs}| > \delta\}$. If $\theta^\star_{min}/2 > \delta \ge R_n^m$, then*

$$P\left(\hat{T} = T\right) \ge 1 - 4\varepsilon\,.$$

The main results of the paper describe properties of estimators, that are obtained by minimization of the MCMC approximation (5) with the Lasso penalty. Theorem 2 states that the estimation error of the Lasso estimator can be controlled. Roughly speaking, the estimation error is small, if the initial sample size and the MCMC sample size are large enough, the model is sparse and the cone invertibility factor $F(\xi, T)$ is not too close to zero. The influence of the model parameters $(n, d, \bar{d}_0)$ as well as Monte Carlo parameters (m, β_1, β_2, M) on the results are explicitly stated. It is worth to emphasize that our results work in the high-dimensional scenario, i.e. the number of vertices d can be greater than the sample size n provided that the model is sparse. Indeed, the condition (13) is satisfied even if $\bar{d} \sim O\left(e^{n^{c_1}}\right)$, $\bar{d}_0 \sim O(n^{c_2})$ and $c_1 + 2c_2 < 1$. The condition (14), that relates to the MCMC sample size, is also reasonable. The number β_1 depends on the initial and stationary distributions. In general, its relation to the number of vertices is exponential. However, in (14) it appears with the logarithm. Moreover, β_1 is also reduced using so called burn-in time, i.e. the beginning of the Markov chain trajectory is discarded. Next, the number β_2 is related to the spectral gap of a Markov chain. Under mild conditions the inverse of β_2 depends polynomially on d, and under strong regularity conditions it can be reduced to $O(d \log d)$ as in Mossel and Sly (2013). Finally, there is also the number M in the condition (14) that relates to the distance between the stationary distribution $h(\cdot)$ and $p(\cdot|\theta^\star)$. Stating

the explicit relation between M and the model seems to be difficult. However, the algorithm, that we propose to calculate $\hat{\theta}$, is designed in such a way to minimize the impact of M on the results. The detailed implementation of the algorithm is given in section 4.

The estimation error of the Lasso estimator in Theorem 2 is measured in l_∞-norm. Similarly to Huang et al. (2013), it can be extended to the general l_q-norm, $q \geq 1$. We omit it, because (15) is sufficient to obtain the second main result of the paper (Corollary 3). It states that the thresholded Lasso estimator is model selection consistent, if, additionally to (13) and (14), the nonzero parameters are not too small and the threshold is appropriately chosen. It is a consequence of the fact, which follows from Theorem 2, that the Lasso seperates significant parameters from irrelevant ones, i.e. for each $(r,s) \in T$ and $(r',s') \notin T$ we have $|\hat{\theta}_{rs}| > |\hat{\theta}_{r's'}|$ with high probability. However, Corollary 3 does not give a way of choosing the threshold δ, because both endpoints of the interval $[R_n^m, \theta^\star_{min}/2]$ are unknown. It is not a surprising fact and has been already observed, for instance, in linear models (Ye and Zhang, 2010, Theorem 8). In section 4 we propose a method of choosing a threshold, that relates to information criteria.

We have already mentioned that there are many approaches to the high-dimensional Ising model. Now we compare conditions, that are sufficient to prove model selection consistency in the current paper to those basing on the likelihood approximation. If we simplify regularity conditions in Theorem 2, Corollary 3 and forget about Monte Carlo parameters in (14), then we have:

(a) the cone invertibility factor condition is satisfied,

(b) the sample size should be sufficiently large, that is $n > \bar{d}_0^2 \log d$,

(c) the nonzero parameters should be sufficiently large, that is $\theta^\star_{min} > \sqrt{\frac{\log d}{n}}$.

In Ravikumar et al. (2010, Corollary 1) one needs stronger irrepresentable condition in (a). Their analog of (b) is $n \geq v^3 \log d$, where v is the maximum neighbourhood size. Since v is smaller than $\bar{d}_0$, their condition is less restrictive. However, in (c) they require the minimum signal strength to be higher than ours, because it has to be larger than $\sqrt{\frac{v \log d}{n}}$ as distinct from $\sqrt{\frac{\log d}{n}}$ in our paper.

Assumptions in Guo et al. (2010, Theorem 2) are stronger than ours. Indeed, they need irrepresentable condition in (a), $\bar{d}_0$ in the third power in (b) and additional factor $\sqrt{\bar{d}_0}$ in (c).

In Xue et al. (2012, Corollary 3.1 (2)) model selection consistency of Lasso estimators is also proved with more restrictive conditions than ours. Namely, they are similar to Ravikumar et al. (2010) and Guo et al. (2010) but $\bar{d}_0$ is reduced in the condition (c). Moreover, they also consider the pseudolikelihood approximation with the SCAD penalty and shows that the condition (a) seems to be superfluous in this case, see Xue et al. (2012, Corollary 3.1 (1)). However, using the SCAD penalty they minimize a nonconvex function to obtain an estimator, so they have to prove that the computed (local) minimizer is the desired theoretic local solution. Their approach can be viewed as a sequence of weighted Lasso problems, so they need auxiliary Lasso procedures to behave well. Therefore, the irrepresentable condition is assumed (Xue et al., 2012, Corollary 3.2).

The conditions sufficient for model selection consistency that are stated in Jalali et al. (2011, Theorem 2) are comparable to ours but also more restrictive. Instead of the condition (a) they consider a similar requirement called the restricted strong convexity condition. It is completed by the restricted strong smoothness condition. Moreover, in the the lower bound in the condition (c) they need an additional factor $\sqrt{\bar{d}_0}$ as well as the upper bound for $\theta^\star_{min}$.

In the proof of Theorem 2 we use methods that are well-known while investigating properties of Lasso estimators as well as some new argumentation. The main novelty (and difficulty) is the use of the Monte Carlo sample, that contains dependent vectors. The first part of our argumentation consists of two steps:

(i) the first step can be viewed as "deterministic". We apply methods that were developed in Ye and Zhang (2010); Huang and Zhang (2012); Huang et al. (2013) and strongly exploit convexity of the considered problem. These auxiliary results are stated in Lemma 5 and Lemma 6 in the appendix A,

(ii) the second step is "stochastic". We state a probabilistic inequality that bounds the l_∞-norm of the derivative of the MCMC approximation (5) at $\theta^\star$, that is

$$\nabla \ell_n^m(\theta^\star) = -\frac{1}{n}\sum_{i=1}^n J(Y_i) + \frac{\sum_{k=1}^m w_k(\theta^\star) J(Y^k)}{\sum_{k=1}^m w_k(\theta^\star)} \tag{17}$$

where

$$w_k(\theta) = \frac{\exp\left[\theta' J(Y^k)\right]}{h(Y^k)}\,, \quad k = 1, \ldots, m\,. \tag{18}$$

Notice that (17) contains independent random variables $Y_1, \ldots, Y_n$ from the initial sample and the Markov chain $Y^1, \ldots, Y^m$ from the MC sample. Therefore, to obtain the exponential inequalities for the l_∞-norm of (17), which are given in Lemma 7 and Corollary 8 in the appendix A, we apply the MCMC theory. In particular, we frequently use the following Hoeffding's inequality for Markov chains (Miasojedow, 2014, Theorem 1.1).

Theorem 4 *Let $Y^1, \ldots, Y^m$ be a reversible Markov chain with a stationary distribution with a density h and a spectral gap $1-\kappa$. Moreover, let $g : \mathcal{Y} \to \mathbb{R}$ be a bounded function and $\mu = \mathbb{E}_{Y\sim h} g(Y)$ be a stationary mean value. Then for every $t > 0, m \in \mathbb{N}$ and an initial distribution q*

$$P\left(\left|\frac{1}{m}\sum_{k=1}^m g(Y^k) - \mu\right| > t\right) \leq 2\beta_1 \exp\left(-\frac{\beta_2 m t^2}{|g|_\infty^2}\right)\,,$$

where $|g|_\infty = \sup_{y\in\mathcal{Y}} |g(y)|$ and β_1, β_2 are defined in (9).

The next part of our argumentation relates to the fact that the Hessian of the MCMC approximation at $\theta^\star$, that is

$$\nabla^2 \ell_n^m(\theta^\star) = \frac{\sum_{k=1}^m w_k(\theta^\star) J(Y^k)^{\otimes 2}}{\sum_{k=1}^m w_k(\theta^\star)} - \left[\frac{\sum_{k=1}^m w_k(\theta^\star) J(Y^k)}{\sum_{k=1}^m w_k(\theta^\star)}\right]^{\otimes 2}, \tag{19}$$

is random variable. Similar problems were considered in several papers investigating properties of Lasso estimators in the high-dimensional Ising model (Ravikumar et al., 2010; Guo et al., 2010; Xue et al., 2012) or the Cox model (Huang et al., 2013). We overcome this difficulty by bounding from below the cone invertibility factor $\bar{F}(\xi, T)$ by nonrandom $F(\xi, T)$. Therefore, we need to prove that Hessians of (10) and (5) are close. It is obtained again using the MCMC theory in Lemma 9 and Corollary 10 in the appendix A.

Finally, the proofs of Theorem 2 and Corollary 3 are stated in the appendix B.

4. Details of implementation

In this section we describe in details practical implementation of the algorithm analyzed in the previous section.

The solution of the problem (7) depends on the choice of λ in the penalty term and the parameter ψ in the instrumental distribution. Finding "optimal" λ and ψ is difficult in practice. To overcome this problem we compute a sequence of minimizers $(\hat{\theta}_i)_i$ such that $\hat{\theta}_i$ corresponds to $\lambda = \lambda_i$ and the sequence $(\lambda_i)_i$ is decreasing. In the second step we use the Bayesian Information Criterion (BIC) to choose the final penalty λ. More precisely, we start with the greatest value λ_0 for which the entire vector $\hat{\theta}$ is zero. For each value of λ_i, $i \geq 1$, we set $\psi = \hat{\theta}_{i-1}$ and use the MCMC approximation (7) with $Y^1, \ldots, Y^m$ given by a Gibbs sampler with the stationary distribution $p(\cdot|\hat{\theta}_{i-1})$. This scheme exploits warm starts and leads to more stable algorithm. Next, the estimator $\hat{\theta}$ is chosen using BIC that is a popular method of choosing λ in the literature, for instance in Xue et al. (2012). Notice that the function $\ell_n^m(\theta)$ is convex, so we can use proximal gradient algorithms to compute $\hat{\theta}_i$ as a solution of (7) for a given λ_i and ψ. In the current paper we use the FISTA algorithm with backtracking from Beck and Teboulle (2009). The whole procedure is summarized in Algorithm 1.

Algorithm 1 MCMC Lasso for Ising model

Let $\lambda_0 > \lambda_1 > \cdots > \lambda_{100}$ and $\psi = 0$.
for $i = 1$ to 100 **do**
Simulate $Y^1, \ldots, Y^m$ using a Gibbs sampler with the stationary distribution $p(y|\psi)$.
Run the FISTA algorithm to compute $\hat{\theta}_i$ as

$$\arg\min_\theta \left\{\ell_n^m(\theta) + \lambda_i |\theta|_1\right\}.$$

Set $\psi = \hat{\theta}_i$.
end for
Next, set $\hat{\theta} = \hat{\theta}_{i^*}$, where

$$i^* = \arg\min_{1 \leq i \leq 100} \left\{ n\ell_n^m(\hat{\theta}_i) + \log(n)\|\hat{\theta}_i\|_0 \right\}$$

and $\|\theta\|_0$ denotes the number of non-zero elements of θ.

In Algorithm 1 we use 100 values of λ uniformly spaced on the log scale, starting from the largest λ, which corresponds to the empty model. We use $m = 10^3 d$ iteration of the Gibbs sampler. To compute $\ell_n^m(\hat{\theta}_i)$ for $i = 1, \ldots, 100$ in BIC we generate one more sample of the size $m = 10^4 d$ using the Gibbs sampler with the stationary distribution $p(\cdot|\hat{\theta}_{50})$.

The important property of our implementation is that the chosen $\psi = \hat{\theta}_{i-1}$ is usually close to $\hat{\theta}_i$, because differences between consecutive λ_i's are small. In our studies the final estimator $\hat{\theta}$ is the element of the sequence $(\hat{\theta}_i)_i$, that recognizes the true model in the best way, i.e. it minimizes the MCMC approximation and is sparse simultaneously. One believes that the final estimator $\hat{\theta} = \hat{\theta}_i$ is close to $\theta^\star$, therefore the chosen $\psi = \hat{\theta}_{i-1}$ should be also similar to $\theta^\star$, that makes M in (9) close to one. Finally, notice that conditionally on the previous step our algorithm fits to the framework described in subsection 2.1.

Note that in the first iteration in Algorithm 1 we use an uniform distribution on $\{-1, +1\}^d$ as an instrumental distribution and we can use i.i.d sample $Y^1, \ldots, Y^m$. Therefore, for λ_1 we get $\beta_1 = \beta_2 = 1$. When we compute estimators for λ_i with $i \geq 2$ we use the last sample from the previous step as an initial point, so since the Markov chain generated by the Gibbs sampler is ergodic its initial distribution should be close to $p(y|\hat{\theta}_{i-2})$. Therefore, even without the burn-in time β_1 should be approximately equal to the L^2-distance between $p(y|\hat{\theta}_{i-1})$ and $p(y|\hat{\theta}_{i-2})$, which seems to be small, because differences between consecutive λ_i's are small. So, after discarding initial iterations β_1 is further reduced. Our choice of stationary distributions also leads to relatively small variances of importance sampling weights given in (18).

The bounds on β_2 are challenging problem itself and the sharp bounds are available only in very specific cases. Due to that, we do not have the explicit control on β_2. However, in our procedure the stationary distributions are typically given by sparse Ising models and by Mossel and Sly (2013) the spectral gap depends mostly on the number of existing edges in the graph. Therefore β_2 should not vanish to rapidly with dimensionality of the problem.

Finally, the thresholded estimator is obtained using the Generalized Information Criterion (GIC). For a prespecified set of thresholds Δ we calculate

$$\delta^* = \arg\min_{\delta\in\Delta}\left\{n\ell_n^m(\hat\theta^\delta) + \log(\bar d)\|\hat\theta^\delta\|_0\right\},$$

where $\hat\theta^\delta$ is the Lasso estimator $\hat\theta$ after thresholding with the level δ. To compute $\ell_n^m(\hat\theta^\delta)$ for $\delta\in\Delta$ in GIC we generate the last sample of the size $m = 10^4 d$ using the Gibbs sampler with the stationary distribution $p(\cdot|\hat\theta)$. To find the optimal threshold we apply GIC that uses larger penalty than BIC. Choosing the threshold in this way should be better in model selection and is recommended, for instance, in Pokarowski and Mielniczuk (2015).

In the rest of this section we discuss computational complexity of our method. The computational cost of a single step of the Gibbs sampler is dominated by computing probability (8), which is of the order $O(r)$, where r is the maximal degree of vertices in a graph related to the stationary distribution. In the paper we focus on estimation of sparse graphs, so the proposed λ_i's have to be sufficiently large to make $\hat\theta_i$'s sparse. Therefore, the degree r is rather small and the computational cost of generating $Y^1,\dots,Y^m$ is of order $O(m)$. Next, we need to compute $\ell_n^m(\theta)$ and its gradient. For an arbitrary Markov chain the cost of these computation is of the order $O(d^2m)$. But when we use single site updates as in the Gibbs sampler we can reduce it to $O(dm)$ by remembering which coordinate of Y^k where updated. Indeed, if we know that only the r coordinates are updated in the step $Y^k\to Y^{k+1}$, then

$$\theta' J(Y^{k+1}) = \theta' J(Y^k) + \sum_{s:\ s<r}\Big(\theta_{sr}[Y^{k+1}(s)Y^{k+1}(r) - Y^k(s)Y^k(r)]\Big) + \sum_{s:\ s>r}\Big(\theta_{rs}[Y^{k+1}(s)Y^{k+1}(r) - Y^k(s)Y^k(r)]\Big).$$

Finally, it is well-known that FISTA (Beck and Teboulle, 2009) achieves accuracy ϵ in $O(\epsilon^{\frac12})$ steps. So, the total cost of computing the solution for single λ_i with precision ϵ is of order $O(\epsilon^{\frac12} md)$. The further reduction of the cost can be obtained using sparsity of $\hat\theta_{i-1}$ in computing $\ell_n^m(\theta)$ and its gradient, and introducing active variables inside the FISTA algorithm.

5. Numerical experiments

In this section we present efficiency of the proposed method via numerical studies. First, we compare our method to three algorithms, which we have mentioned previously, using simulated data sets. Next we apply our method to the real data example.

5.1. Simulated data

To illustrate the performance of the proposed method we simulate data sets in two scenarios:

M1: The first 6 vertices are correlated, while the remaining vertices are independent: $\theta^\star_{rs} = \pm 2$ for $r<s$ and $s = 2,3,4,5,6$, other $\theta^\star_{rs} = 0$. Thus, the model dimension in this problem is 15. The signs are chosen randomly.

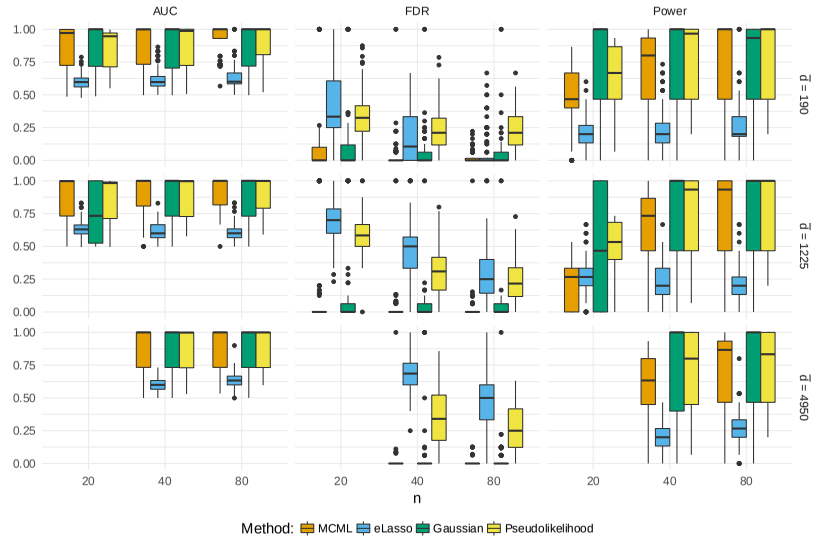


Figure 1: The results for $M1$ model

M2: The first 20 vertices have the "chain structure" and the rest are independent: $\theta^\star_{r-1,r} = \pm 1$ for $r\le 20$. Again the signs are chosen randomly and the model dimension is 19.

The model $M1$ corresponds to a dense structure on a small subset of vertices. The model $M2$ is a simple structure which involves relatively large subset of vertices.

We consider the following cases: $d = 20, 50, 100$. So, the considered number of possible edges (parameters of the model) is $\bar d = 190, 1225, 4950$, respectively. For $\bar d = 190, 1225$ we use $n = 20, 40, 80$ and for $\bar d = 4950$ we use $n = 40, 80$.

For each configuration of the model, the number of vertices d and the number of observation n we sample 100 replications of data sets. For sampling a data set we use a final configuration of independent Gibbs samplers of the length 10^6.

In simulation study we compare our methods to the following methods: the pseudo-likelihood approach, eLasso proposed by van Borkulo et al. (2014) and the Gaussian approximation from Banerjee et al. (2008). The eLasso method is based on separate logistic models and it is similar to the method proposed by Ravikumar et al. (2010). For all methods except eLasso we use two stage procedures, which are analogous to Algorithm 1. Namely, we choose λ by BIC and in the second step we choose the threshold δ by GIC. For the Gaussian approximation we use the same approximate likelihood as in Viallon et al. (2013). For eLasso for every node we use BIC as in van Borkulo et al. (2014).

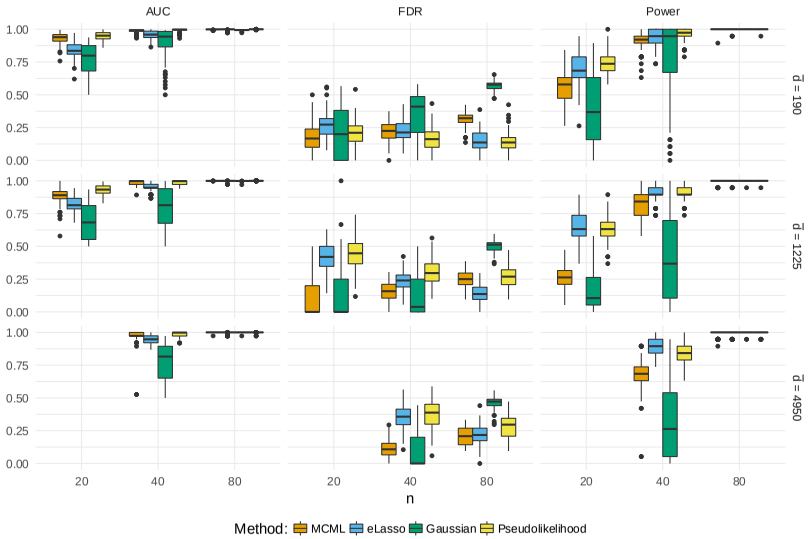


Figure 2: The results for $M2$ model

In the comparison we use the following measures of the accuracy. First, to observe the ability of methods to separate true and false edges we compute AUC, where the ROC curve is computed as the threshold δ varies. The estimates are also compared using the false discovery rate (FDR) and the ability of recognizing true edges (denoted by "Power"). The results are summarized in Figure 1 and Figure 2 for models $M1$ and $M2$, respectively.

In the first model we can observe that our algorithm and the Gaussian approximation work very well and comparably. The latter has slightly larger power, but at the price of slightly larger FDR. The dominance of these two methods over the pseudolikehood estimator and eLasso is evident. The pseudolikelihood method finds well true edges, but is not able to discard false edges. eLasso works very poorly in this model.

The second model has a simple structure, so both methods based on the pseudolikelihood approach work much better than in $M1$. The accuracy of the Gaussian approximation is weak in this model. It has substantial problem with finding true edges, especially when d is large and n small. For $n = 20, 40$ our estimator has relatively small FDR and large Power. In this model we can observe that FDR increases as n increases for the Gaussian approximation and for $n = 80$ it reaches about 0.5. The MCML approximation has also "increasing FDR", but this behaviour is less conspicuous. In fact, for $n = 80$ FDR of our algorithm is still comparable to those of eLasso and Pseudolikelihood.

Summarizing, it is difficult to indicate the winner algorithm. We can observe that the quality of our algorithm in selecting the true model is satisfactory. Moreover, only this procedure works on a good level in both models and avoids making noticeable mistakes. The Gaussian approximation works well in $M1$, but seems to be the weakest in $M2$. The eLasso completely fails in $M1$, but its quality in $M2$ is high, especially for large sample sizes. The pseudolikelihood approach in all examples well separates true and false edges, has good power but in comparison with other methods its FDR is too high, so models that it chooses contains many irrelevant edges.

Clearly, by construction the computational cost of our method is larger than its three competitors. However, the whole time that is needed to compute our estimator is reasonable. For instance, for a single data set and $n = 40$ computing the estimator on 3.4 GHZ CPU takes about 15 seconds for $\bar{d} = 190$, 60 seconds for $\bar{d} = 1225$ and 5 minutes for $\bar{d} = 4950$. Since our algorithm uses only sufficient statistics the dependence on n of its computational cost is negligible. Thus, the computational times for $n = 80$ are almost the same.

5.2. CAL500 dataset

We apply our method to "CAL500" dataset (Turnbull et al., 2008). Working with a real data set we consider the Ising model (1) with a linear term (an external field). This modification is motivated by the fact that in practice marginal probabilities of $Y(s)$ being $+1$ or -1 are unknown and should be estimated. The adaptation of our algorithm to this case is straightforward.

For model selection in the Ising model there are no natural measures of the quality of estimates. One would try to compare the prediction ability of obtained estimators, but prediction for the Ising model is challenging itself and results will be biased by the method used to approximate predicted states. Moreover, all considered methods optimize different loss functions, so these loss functions also cannot be used to the honest comparison of the methods. Due to that, we decide to show only the results of our method for the real data example.

The considered data set consists of 174 binary features and 68 numeric features for 502 songs. We skipped the numeric features and apply our method to find the dependence structure between labels. These labels concerning genre, mood or instrument are annotated to songs. We run our algorithm analogously to the case of simulated data and as the result we obtain a sparse graph with 181 edges, see Figure 3. We observe that founded edges are rather intuitive. For instance, among the most positively correlated labels we have labels denoted by 3 and 14 (,,Emotion-Arousing-Awakening" and ,,Emotion-Exciting-Thrilling"), 57 and 61 (,,Song-Like" and ,, Song-Recommend") or 22 and 34 (,,Emotion-Loving-Romantic" and ,,Emotion-Touching-Loving"). On the other hand, the most negatively correlated labels are: 44 and 45 (,,Instrument-Female Lead Vocals" and ,,Instrument - Male Lead Vocals") or 63 and 64 (,,Song-Texture Acoustic" and ,,Song-Texture Electric").

6. Conclusions

In the paper we consider a problem of structure learning for binary Markov random fields. We base estimation of model parameters on the Lasso penalized Monte Carlo approximation of the likelihood. In the theoretical part of the paper we show that the proposed procedure

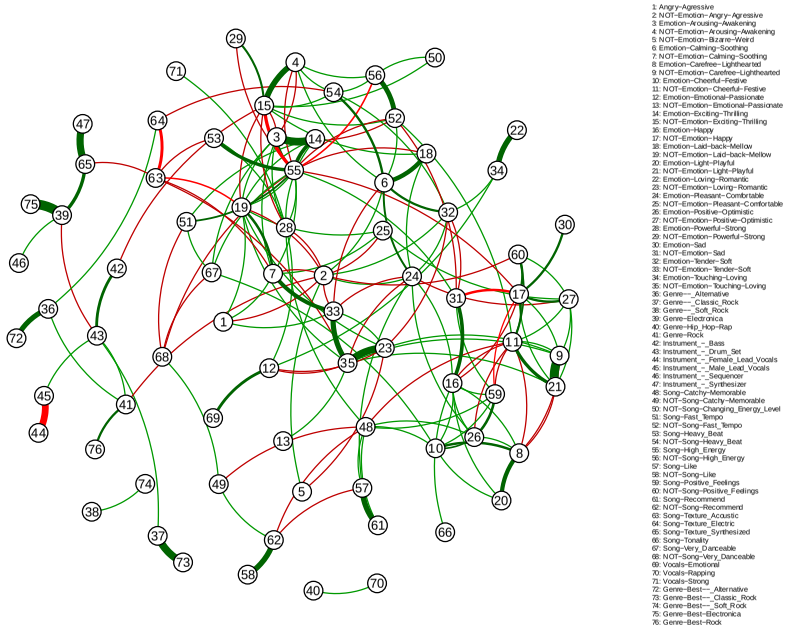


Figure 3: The obtained graph for CAL500 dataset. The width of the edges corresponds to the magnitude of $|\hat{\theta}^{\delta^*}_{rs}|$. The edges with higher absolute value are wider. The color denotes the sign of $\hat{\theta}^{\delta^*}_{rs}$: green – positive, red – negative. To improve clarity we do not show edges corresponding to $|\hat{\theta}^{\delta^*}_{rs}| < 0.001$.

reveals the true dependence structure with high probability. The regularity conditions that we need are not restrictive and are weaker than assumptions used in the other approaches based on the likelihood approximation. Moreover, the theoretical results are completed by numerical experiments. They confirm that the MCMC approximation is able to find the true model in a satisfactory way and its quality is comparable or higher than competing algorithms.

The results of the current paper can be easily extended to other discrete Markov random fields. There are also some non-trivial issues that are not discussed in the paper, for instance

investigating the model (1) with predictors (covariates). The evaluation of the prediction error of the procedure is also a difficult task. Clearly, these problems need detailed studies.

Acknowledgments

We would like to thank the associate editor and reviewers for their comments, that have improved the paper. Błażej Miasojedow was supported by the Polish National Science Center grant no. 2015/17/D/ST1/01198. Wojciech Rejchel was supported by the Polish National Science Center grants no. 2014/12/S/ST1/00344 and no. 2015/17/B/ST6/01878.

Appendix A. Auxiliary results

In this section we formulate lemmas that are needed to prove main results of the paper. The roles, that they play, are described in detail at the end of section 3. The first lemma is borrowed from Huang et al. (2013, Lemma 3.1).

Lemma 5 *Let* $\tilde{\theta} = \hat{\theta} - \theta^\star$, $z^* = |\nabla \ell^m_n(\theta^\star)|_\infty$ *and*

$$D(\hat{\theta}, \theta) = (\hat{\theta} - \theta)' \left[\nabla \ell^m_n(\hat{\theta}) - \nabla \ell^m_n(\theta) \right] .$$

Then

$$(\lambda - z^*)|\tilde{\theta}_{T^c}|_1 \leq D(\hat{\theta}, \theta^\star) + (\lambda - z^*)|\tilde{\theta}_{T^c}|_1 \leq (\lambda + z^*)|\tilde{\theta}_T|_1 . \tag{20}$$

Besides, for arbitrary $\xi > 1$ *on the event*

$$\Omega_1 = \left\{ |\nabla \ell^m_n(\theta^\star)|_\infty \leq \frac{\xi - 1}{\xi + 1} \lambda \right\} \tag{21}$$

the random vector $\tilde{\theta}$ *belongs to the cone* $\mathcal{C}(\xi, T)$.

Proof The proof is the same as the proof of Huang et al. (2013, Lemma 3.1). It is quoted here to make the paper complete.
Convexity of the MCMC approximation $\ell^m_n(\theta)$ easily implies the first inequality in (20). The same property combined with convexity of the Lasso penalty gives us that zero has to belong to the subgradient of (7) at the minimizer $\hat{\theta}$, i.e.

$$\begin{cases} \nabla_{rs} \ell^m_n(\hat{\theta}) = -\lambda \text{sign}(\hat{\theta}_{rs}), & \text{if } \hat{\theta}_{rs} \neq 0 \\ |\nabla_{rs} \ell^m_n(\hat{\theta})| \leq \lambda, & \text{if } \hat{\theta}_{rs} = 0 , \end{cases} \tag{22}$$

where we use $\nabla \ell^m_n(\theta) = \left(\nabla_{rs} \ell^m_n(\hat{\theta}) \right)_{r<s}$ and $\text{sign}(t) = 1$ for $t > 0$, $\text{sign}(t) = -1$ for $t < 0$, $\text{sign}(t) = 0$ for $t = 0$. Using (22) and properties of the l_1-norm we obtain that

$$\begin{aligned} D(\hat{\theta}, \theta^\star) &= \sum_{(r,s) \in T} \tilde{\theta}_{rs} \nabla_{rs} \ell^m_n(\theta^\star + \tilde{\theta}) + \sum_{(r,s) \in T^c} \hat{\theta}_{rs} \nabla_{rs} \ell^m_n(\theta^\star + \tilde{\theta}) - \tilde{\theta}' \nabla \ell^m_n(\theta^\star) \\ &\leq \lambda \sum_{(r,s) \in T} |\tilde{\theta}_{rs}| - \lambda \sum_{(r,s) \in T^c} |\hat{\theta}_{rs}| + |\tilde{\theta}|_1 z^* \\ &\leq \lambda |\tilde{\theta}_T|_1 - \lambda |\tilde{\theta}_{T^c}|_1 + z^* |\tilde{\theta}_T|_1 + z^* |\tilde{\theta}_{T^c}|_1 \\ &= (\lambda + z^*)|\tilde{\theta}_T|_1 + (z^* - \lambda)|\tilde{\theta}_{T^c}|_1 . \end{aligned}$$

Thus, the second inequality in (20) is also established. To prove the last claim of the lemma notice that on the event Ω_1 we obtain from (20)

$$|\tilde{\theta}_{T^c}|_1 \leq \frac{\lambda + z^*}{\lambda - z^*}|\tilde{\theta}_T|_1 \leq \xi|\tilde{\theta}_T|_1 \,.$$

■

The second lemma is an adaptation of Huang et al. (2013, Theorem 3.1) to our problem.

Lemma 6 *Let $\xi > 1$. Moreover, let us denote $\tau = \frac{(\xi+1)\bar{d}_0\lambda}{\bar{F}(\xi,T)}$ and an event*

$$\Omega_2 = \left\{\tau < e^{-1}\right\} \,. \tag{23}$$

Then $\Omega_1 \cap \Omega_2 \subset A$, where

$$A = \left\{|\hat{\theta} - \theta^\star|_\infty \leq \frac{2\xi e^\eta \lambda}{(\xi+1)\bar{F}(\xi,T)}\right\} \,, \tag{24}$$

where $\eta < 1$ is the smaller solution of the equation $\eta e^{-\eta} = \tau$.

Proof Suppose we are on the event $\Omega_1 \cap \Omega_2$. Denote again $\tilde{\theta} = \hat{\theta} - \theta^\star$ and notice that $\theta = \frac{\tilde{\theta}}{|\tilde{\theta}|_1} \in \mathcal{C}(\xi, T)$ by Lemma 5. Consider the function

$$g(t) = \theta' \nabla \ell_n^m(\theta^\star + t\theta) - \theta' \nabla \ell_n^m(\theta^\star)$$

for each $t \geq 0$. This function is nondecreasing, because $\ell_n^m(\cdot)$ is convex. Thus, we obtain $g(t) \leq g(|\tilde{\theta}|_1)$ for every $t \in (0, |\tilde{\theta}|_1)$. On the event Ω_1 and from Lemma 5 we have that

$$\theta' \left[\nabla \ell_n^m(\theta^\star + t\theta) - \nabla \ell_n^m(\theta^\star)\right] + \frac{2\lambda}{\xi+1}|\theta_{T^c}|_1 \leq \frac{2\lambda\xi}{\xi+1}|\theta_T|_1 \,. \tag{25}$$

In further argumentation we consider all nonnegative t satisfying (25), that is an interval $[0, \tilde{t}]$ for some $\tilde{t} > 0$. Proceeding similarly to the proof of Huang et al. (2013, Lemma 3.2) we obtain

$$t\theta' \left[\nabla \ell_n^m(\theta^\star + t\theta) - \nabla \ell_n^m(\theta^\star)\right] \geq t^2 \exp(-\gamma_{t\theta})\theta' \nabla^2 \ell_n^m(\theta^\star)\theta \,, \tag{26}$$

where $\gamma_{t\theta} = t \max_{k,l} |\theta' J(Y^k) - \theta' J(Y^l)| \leq 2t$, because $J(Y^k) = \left(Y^k(r)Y^k(s)\right)_{rs}$ and $|\theta|_1 = 1$. Therefore, the right-hand side in (26) can be lower bounded by

$$t^2 \exp(-2t)\theta' \nabla^2 \ell_n^m(\theta^\star)\theta \,. \tag{27}$$

Using the definition of $\bar{F}(\xi, T)$, the fact that $\theta \in \mathcal{C}(\xi, T)$, the bound (27) and (25) we obtain

$$\begin{aligned} t\exp(-2t)\frac{\bar{F}(\xi,T)|\theta_T|_1^2}{\bar{d}_0} &\leq t\exp(-2t)\theta' \nabla^2 \ell_n^m(\theta^\star)\theta \\ &\leq \theta' \left[\nabla \ell_n^m(\theta^\star + t\theta) - \nabla \ell_n^m(\theta^\star)\right] \\ &\leq \frac{2\lambda\xi}{\xi+1}|\theta_T|_1 - \frac{2\lambda}{\xi+1}|\theta_{T^c}|_1 \\ &\leq \lambda(\xi+1)|\theta_T|_1^2/2 \,. \end{aligned}$$

So, every t satisfying (25) has to fulfill the inequality $2t\exp(-2t) \leq \tau$. In particular, $2\tilde{t}\exp(-2\tilde{t}) \leq \tau$. We are on Ω_2, so it implies that $2\tilde{t} \leq \eta$, where η is the smaller solution of the equation $\eta\exp(-\eta) = \tau$. We know also that $|\tilde{\theta}|_1 \leq \tilde{t}$, so

$$\begin{aligned} |\tilde{\theta}|_1 \exp(-\eta) \leq \tilde{t}\exp(-2\tilde{t}) &\leq \frac{\tilde{t}\exp(-2\tilde{t})\theta' \nabla^2 \ell_n^m(\theta^\star)\theta}{\bar{F}(\xi,T)|\theta_T|_1|\theta|_\infty} \\ &\leq \frac{\theta'\left[\nabla \ell_n^m(\theta^\star + \tilde{t}\theta) - \nabla \ell_n^m(\theta^\star)\right]}{\bar{F}(\xi,T)|\theta_T|_1|\theta|_\infty} \\ &\leq \frac{2\lambda\xi}{(\xi+1)\bar{F}(\xi,T)|\theta|_\infty} \,, \end{aligned}$$

where we have used bounds (27) and (25). Using the equality $|\theta|_\infty = \frac{|\tilde{\theta}|_\infty}{|\tilde{\theta}|_1}$, we finish the proof. ■

Lemma 7 *For every natural n, m and positive t*

$$P\left(|\nabla \ell_n^m(\theta^\star)|_\infty \leq t\right) \geq 1 - 2\bar{d}\exp\left(-nt^2/8\right) - \beta_1 \exp\left(-\frac{m\beta_2}{4M^2}\right) - 2\bar{d}\beta_1 \exp\left(-\frac{t^2 m\beta_2}{64M^2}\right) \,.$$

Proof We can rewrite $\nabla \ell_n^m(\theta^\star)$ as

$$\nabla \ell_n^m(\theta^\star) = -\left[\frac{1}{n}\sum_{i=1}^n J(Y_i) - \frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right] + \frac{\frac{1}{m}\sum_{k=1}^m w_k(\theta^\star)\left[J(Y^k) - \frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right]}{\frac{1}{m}\sum_{k=1}^m w_k(\theta^\star)} \tag{28}$$

Notice that the first therm in (28) depends only on the initial sample $Y_1, \ldots, Y_n$ and is an average of i.i.d random variables. The second term depends only on the MCMC sample $Y^1, \ldots, Y^m$. We start the analysis with the former one. Using Hoeffding's inequality we obtain for each natural n, positive t and a pair of indices $r < s$

$$P\left(\left|\frac{1}{n}\sum_{i=1}^n J_{rs}(Y_i) - \frac{\nabla_{rs} C(\theta^\star)}{C(\theta^\star)}\right| > t/2\right) \leq 2\exp\left(-nt^2/8\right) \,.$$

Therefore, by the union bound we have

$$P\left(\left|\frac{1}{n}\sum_{i=1}^n J(Y_i) - \frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right|_\infty > t/2\right) \leq 2\bar{d}\exp\left(-nt^2/8\right) \,. \tag{29}$$

Next, we investigate the second expression in (28). Its denominator is an average that depends on the Markov chain. To handle it we can use Theorem 4 in section 3. Notice that $\frac{\exp[(\theta^\star)' J(y)]}{h(y)} \leq MC(\theta^\star)$ for every y and $\mathbb{E}_{Y\sim h}\frac{\exp[(\theta^\star)' J(Y)]}{h(Y)} = C(\theta^\star)$. Therefore, for every n, m

$$P\left(\frac{1}{m}\sum_{k=1}^m w_k(\theta^\star) \geq C(\theta^\star)/2\right) \geq 1 - \beta_1 \exp\left(-\frac{m\beta_2}{4M^2}\right) \,. \tag{30}$$

Finally, we bound the l_∞-norm of the numerator of the second term in (28). We fix a pair of indices $r < s$. It is not difficult to calculate that

$$\mathbb{E}_{Y\sim h}\left[\frac{\exp\left[(\theta^\star)'J(Y)\right]}{h(Y)}\left(J_{rs}(Y)-\frac{\nabla_{rs}C(\theta^\star)}{C(\theta^\star)}\right)\right]=0$$

and for every y

$$\frac{\exp((\theta^\star)'J(y)}{h(y)}\left|J_{rs}(y)-\frac{\nabla_{rs}C(\theta^\star)}{C(\theta^\star)}\right|\le 2MC(\theta^\star)\,.$$

From Theorem 4 we obtain for every positive t

$$\left|\frac{1}{m}\sum_{k=1}^m w_k(\theta^\star)\left[J_{rs}(Y^k)-\frac{\nabla_{rs}C(\theta^\star)}{C(\theta^\star)}\right]\right|\ge tC(\theta^\star)/4$$

with probability at least $1-2\beta_1\exp\left(-\frac{t^2m\beta_2}{64M^2}\right)$. Using union bounds we estimate the numerator of the second expression in (28). This fact and (30) imply that for every positive t and natural n, m with probability at least $1-\beta_1\exp\left(-\frac{m\beta_2}{4M^2}\right)-2\bar{d}\beta_1\exp\left(-\frac{t^2m\beta_2}{64M^2}\right)$ we have

$$\left|\frac{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta^\star)\left[J(Y^k)-\frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right]}{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta^\star)}\right|_\infty\le t/2\,. \tag{31}$$

Taking (29) and (31) together we finish the proof. ■

Corollary 8 *Let* $\varepsilon>0, \xi>1$ *and*

$$\lambda=\frac{\xi+1}{\xi-1}\max\left(2\sqrt{\frac{2\log(2\bar{d}/\varepsilon)}{n}}, 8M\sqrt{\frac{\log\left[(2\bar{d}+1)\beta_1/\varepsilon\right]}{m\beta_2}}\right)\,.$$

Conditions (13) *and* (14) *imply*

$$P(\Omega_1)\ge 1-2\varepsilon\,.$$

Proof We take $t=\frac{\xi-1}{\xi+1}\lambda$ in Lemma 7. ■

Lemma 9 *For every* n, m *and positive* t

$$\begin{aligned}P\left(\left|\nabla^2\ell_n^m(\theta^\star)-\nabla^2\log C(\theta^\star)\right|_\infty\le t\right) &\ge 1-2\beta_1\exp\left(-\frac{m\beta_2}{4M^2}\right)\\ &- 2\bar{d}(\bar{d}+1)\beta_1\exp\left(-\frac{t^2m\beta_2}{256M^2}\right)\,.\end{aligned} \tag{32}$$

Proof To estimate the l_∞-norm of the matrix difference in (32) we bound l_∞-norms of two matrices:

$$\frac{\sum\limits_{k=1}^m w_k(\theta^\star)J(Y^k)^{\otimes 2}}{\sum\limits_{k=1}^m w_k(\theta^\star)}-\frac{\nabla^2C(\theta^\star)}{C(\theta^\star)}=\frac{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta^\star)\left[J(Y^k)^{\otimes 2}-\frac{\nabla^2C(\theta^\star)}{C(\theta^\star)}\right]}{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta^\star)} \tag{33}$$

and

$$\left[\frac{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta)J(Y^k)}{\frac{1}{m}\sum\limits_{k=1}^m w_k(\theta)}\right]^{\otimes 2}-\left[\frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right]^{\otimes 2}\,. \tag{34}$$

The denominator of the right-hand side of (33) has been estimated in the proof of Lemma 7, so we bound the numerator. We can calculate that

$$\mathbb{E}_{Y\sim h}\left[\frac{\exp\left[(\theta^\star)'J(Y)\right]}{h(Y)}\left(J(Y)^{\otimes 2}-\frac{\nabla^2C(\theta^\star)}{C(\theta^\star)}\right)\right]=0$$

and for every y and two pairs of indices $r<s, r'<s'$ we have

$$\frac{\exp((\theta^\star)'J(y)}{h(y)}\left|J_{rs}(y)J_{r's'}(y)-\frac{\nabla^2_{rs,r's'}C(\theta^\star)}{C(\theta^\star)}\right|\le 2MC(\theta^\star)\,.$$

Using the union bound and (30) we upper-bound the l_∞-norm of the right-hand side of (33) by $t/2$ with probability at least

$$1-\beta_1\exp\left(-\frac{m\beta_2}{4M^2}\right)-2\bar{d}^2\beta_1\exp\left(-\frac{t^2m\beta_2}{64M^2}\right)\,.$$

The last step of the proof is handling with the l_∞-norm of (34). This expression can be upper-bounded by

$$\left|\frac{\sum\limits_{k=1}^m w_k(\theta)J(Y^k)}{\sum\limits_{k=1}^m w_k(\theta)}-\frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right|_\infty\left(\left|\frac{\sum\limits_{k=1}^m w_k(\theta)J(Y^k)}{\sum\limits_{k=1}^m w_k(\theta)}\right|_\infty+\left|\frac{\nabla C(\theta^\star)}{C(\theta^\star)}\right|_\infty\right)_\infty\,. \tag{35}$$

The first term in (35) has been bounded with high probability in the proof of Lemma 7. The remaining two can be easily estimated by one. Therefore, for every positive t the l_∞-norm of (34) is not greater than $t/2$ with probability at least

$$1-\beta_1\exp\left(-\frac{m\beta_2}{4M^2}\right)-2\bar{d}\beta_1\exp\left(-\frac{t^2m\beta_2}{256M^2}\right)\,.$$

Putting together this fact and the bound of (33) we finish the proof. ■

Corollary 10 *If* (14)*, then for every* $\varepsilon > 0$ *the following inequality*

$$\bar{F}(\xi, T) \geq F(\xi, T) - 16\bar{d}_0(1+\xi)^2 M\sqrt{\frac{\log\left[2\bar{d}(\bar{d}+1)\beta_1/\varepsilon\right]}{m\beta_2}}$$

has probability at least $1 - 2\varepsilon$*.*

Proof We take

$$t = 16M\sqrt{\frac{\log\left[2\bar{d}(\bar{d}+1)\beta_1/\varepsilon\right]}{m\beta_2}}$$

in Lemma 9 and use Huang et al. (2013, Lemma 4.1 (ii)). ■

Appendix B. Proofs of main results

Proof [Proof of Theorem 2] Fix $\varepsilon > 0, \xi > 1$ and denote $\gamma = \gamma(\xi) = \frac{\alpha(\xi)-2}{\alpha(\xi)} \in (0,1)$. First, from Corollary 8 we know that $P(\Omega_1) \geq 1 - 2\varepsilon$. Using the condition (14) we obtain that

$$F(\xi, T) - 16\bar{d}_0(1+\xi)^2 M\sqrt{\frac{\log\left[2\bar{d}(\bar{d}+1)\beta_1/\varepsilon\right]}{m\beta_2}} \geq \gamma F(\xi, T)\,.$$

Therefore, from Corollary 10 we have that $P(\bar{F}(\xi, T) \geq \gamma F(\xi, T)) \geq 1 - 2\varepsilon$. It is not difficult to calculate that

$$\frac{(1+\xi)\bar{d}_0\lambda}{\gamma F(\xi, T)} \leq e^{-1}\,,$$

so we have also $P(\Omega_2) \geq 1 - 2\varepsilon$. To finish the proof we use Lemma 6 (with $\eta = 1$ for simplicity) and again bound $\bar{F}(\xi, T)$ from below by $\gamma F(\xi, T)$ in the event A defined in (24). ■

Proof [Proof of Corollary 3] The proof is a simple consequence of the uniform bound (15) obtained in Theorem 2. Indeed, for an arbitrary pair of indices $(r, s) \notin T$ we obtain

$$|\hat{\theta}_{rs}| = |\hat{\theta}_{rs} - \theta^\star_{rs}| \leq R^m_n\,,$$

so $(r, s) \notin \hat{T}$. Analogously, if we take a pair $(r, s) \in T$, then

$$|\hat{\theta}_{rs}| \geq |\theta^\star_{rs}| - |\hat{\theta}_{rs} - \theta^\star_{rs}| > 2\delta - R^m_n \geq \delta\,.$$

■

References

Yves F. Atchadé, Gersende Fort, and Eric Moulines. On perturbed proximal gradient algorithms. *Journal of Machine Learning Research*, 18:310–342, 2017.

Onureena Banerjee, Laurent El Ghaoui, and Alexandre d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate gaussian or binary data. *Journal of Machine Learning Research*, 9:485–516, 2008.

Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2:183–202, 2009.

Julian Besag. Spacial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–236, 1974.

Peter J Bickel, Ya'acov Ritov, and Alexandre B Tsybakov. Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, 37:1705–1732, 2009.

Guy Bresler, Elchanan Mossel, and Allan Sly. Reconstruction of Markov random fields from samples: Some observations and algorithms. In *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*, pages 343–356. Springer, 2008.

Peter Bühlmann and Sara van de Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer Series in Statistics, New York: Springer, 2011.

Jianqing Fan and Runze Li. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, 96:1348–1360, 2001.

Charles J Geyer. On the convergence of Monte Carlo maximum likelihood calculations. *Journal of the Royal Statistical Society. Series B (Methodological)*, 56:261–274, 1994.

Charles J Geyer and Elizabeth A Thompson. Constrained Monte Carlo maximum likelihood for dependent data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 54:657–699, 1992.

Jian Guo, Elizaveta Levina, George Michailidis, and Ji Zhu. Joint structure estimation for categorical Markov networks. *Technical report*, 2010.

Holger Höfling and Robert Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudolikelihoods. *Journal of Machine Learning Research*, 10:883–906, 2009.

Jean Honorio. Convergence Rates of Biased Stochastic Optimization for Learning Sparse Ising Models. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1099–1106, 2012.

Jian Huang and Cun-Hui Zhang. Estimation and selection via absolute penalized convex minimization and its multistage adaptive applications. *Journal of Machine Learning Research*, 13:1839–1864, 2012.

Jian Huang, Tingni Sun, Zhiliang Ying, Yi Yu, and Cun-Hui Zhang. Oracle inequalities for the lasso in the Cox model. *Annals of Statistics*, 41(3):1142–1165, 2013.

Ernst Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift für Physik A Hadrons and Nuclei*, 31(1):253–258, 1925.

Ali Jalali, Christopher C Johnson, and Pradeep K Ravikumar. On learning discrete graphical models using greedy methods. In *Advances in Neural Information Processing Systems*, pages 1935–1943, 2011.

Ioannis Kontoyiannis and Sean P Meyn. Geometric ergodicity and the spectral gap of non-reversible markov chains. *Probability Theory and Related Fields*, 154:327–339, 2012.

Su-In Lee, Varun Ganapathi, and Daphne Koller. Efficient Structure Learning of Markov Networks using l_1-Regularization. In *Advances in Neural Information Processing Systems*, pages 817–824, 2006.

Błażej Miasojedow. Hoeffding's inequalities for geometrically ergodic markov chains on general state space. *Statistics & Probability Letters*, 87:115–120, 2014.

Elchanan Mossel and Allan Sly. Exact thresholds for Ising–Gibbs samplers on general graphs. *The Annals of Probability*, 41:294–328, 2013.

Sahand Negahban, Bin Yu, Martin J Wainwright, and Pradeep K Ravikumar. A unified framework for high-dimensional analysis of M-estimators with decomposable regularizers. In *Advances in Neural Information Processing Systems*, pages 1348–1356, 2009.

Piotr Pokarowski and Jan Mielniczuk. Combined l_1 and greedy l_0 penalized least squares for linear model selection. *Journal of Machine Learning Research*, 16:961–992, 2015.

Pradeep Ravikumar, Martin J Wainwright, and John Lafferty. High-dimensional Ising model selection using l_1-regularized logistic regression. *The Annals of Statistics*, 38:1287–1319, 2010.

Gareth O Roberts and Jeffrey S Rosenthal. Geometric ergodicity and hybrid Markov chains. *Electronic Communications in Probability*, 2:13–25, 1997.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58:267–288, 1996.

Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Semantic annotation and retrieval of music and sound effects. *IEEE Transactions on Audio, Speech, and Language Processing*, 16:467–476, 2008.

Claudia D. van Borkulo, Denny Borsboom, Sacha Epskamp, Tessa F. Blanken, Lynn Boschloo, Robert A. Schoevers, and Lourens J. Waldorp. A new method for constructing networks from binary data. *Scientific Reports*, 4, 2014.

Sara van de Geer. High-dimensional generalized linear models and the lasso. *The Annals of Statistics*, 36:614–645, 2008.

Vivian Viallon, Onureena Banerjee, Eric Jougla, Grégoire Rey, and Joel Coste. Empirical comparison study of approximate methods for structure selection in binary graphical models. *Biometrical Journal*, 56:307–331, 2013.

Martin J. Wainwright and Michael I. Jordan. Log-determinant relaxation for approximate inference in discrete markov random fields. *IEEE Transactions on Signal Processing*, 54: 2099–2109, 2006.

Lingzhou Xue, Hui Zou, and Tianxi Cai. Nonconcave penalized composite conditional likelihood estimation of sparse Ising models. *The Annals of Statistics*, 40:1403–1429, 2012.

Fei Ye and Cun-Hui Zhang. Rate Minimaxity of the Lasso and Dantzig Selector for the l_q loss in l_r Balls. *Journal of Machine Learning Research*, 11:3519–3540, 2010.

Tong Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 10:555–568, 2009.

Peng Zhao and Bin Yu. On model selection consistency of Lasso. *Journal of Machine Learning Research*, 7:2541–2563, 2006.

Shuheng Zhou. Thresholding procedures for high dimensional variable selection and statistical estimation. In *Advances in Neural Information Processing Systems*, pages 2304–2312, 2009.

Journal of Machine Learning Research 19 (2018) 1-35 Submitted 03/17; Revised 05/18; Published 12/18

Using Side Information to Reliably Learn Low-Rank Matrices from Missing and Corrupted Observations

Kai-Yang Chiang KYCHIANG@CS.UTEXAS.EDU
Inderjit S. Dhillon INDERJIT@CS.UTEXAS.EDU
Department of Computer Science
University of Texas at Austin
Austin, TX 78701, USA
Cho-Jui Hsieh CHOHSIEH@UCDAVIS.EDU
Department of Statistics and Computer Science
University of California at Davis
Davis, CA 95616, USA

Editor: Sanjiv Kumar

Abstract

Learning a low-rank matrix from missing and corrupted observations is a fundamental problem in many machine learning applications. However, the role of *side information* in low-rank matrix learning has received little attention, and most current approaches are either ad-hoc or only applicable in certain restrictive cases. In this paper, we propose a general model that exploits side information to better learn low-rank matrices from missing and corrupted observations, and show that the proposed model can be further applied to several popular scenarios such as matrix completion and robust PCA. Furthermore, we study the effect of side information on sample complexity and show that by using our model, the efficiency for learning can be improved given sufficiently informative side information. This result thus provides theoretical insight into the usefulness of side information in our model. Finally, we conduct comprehensive experiments in three real-world applications—relationship prediction, semi-supervised clustering and noisy image classification, showing that our proposed model is able to properly exploit side information for more effective learning both in theory and practice.

Keywords: Side information, low-rank matrix learning, learning from missing and corrupted observations, matrix completion, robust PCA

1. Introduction

Learning a low-rank matrix from noisy, high-dimensional complex data is an important research challenge in modern machine learning. In particular, in the recent big data era, assuming that the observations come from a model with implicit low-rank structure is one of the most prevailing approaches to avoid the curse of dimensionality. While various low-rank matrix learning problems arise from different contexts and domains, the primary challenge is rather similar: namely to reliably learn a low-rank matrix L_0 based only on missing and corrupted observations from L_0. This generic framework includes many well-known machine learning problems such as matrix completion (Candès and Tao, 2009), robust PCA (Wright et al., 2009) and matrix sensing (Zhong et al., 2015), and is shown to be useful in many important real-world applications including recommender systems (Koren et al., 2009), social network analysis (Hsieh et al., 2012) and image processing (Wright et al., 2009).

Among research related to low-rank matrix learning, one promising direction is to further exploit *side information*, or *features*, to help the learning process. [1] The notion of side information appears naturally in many applications. For example, in the famous Netflix problem where the goal is movie recommendation based on users' ratings, a popular approach is to assume that the given user-movie rating pairs are sampled from a low-rank matrix (Koren et al., 2009). However, besides rating history, profiles of users and/or genres of movies may also be provided, and one can possibly leverage such side information for better recommendation. Since such additional features are available in many applications, designing a model to better incorporate features into low-rank matrix learning problems becomes an important issue with both theoretical and practical interests.

Motivated by the above realization, we study the effect of side information on learning low-rank matrices from missing and corrupted observations in this paper. Our general problem setting can be formally described as follows. Let $L_0 \in \mathbb{R}^{n_1 \times n_2}$ be the low-rank modeling matrix, yet due to various reasons we can only observe a matrix $R \in \mathbb{R}^{n_1 \times n_2}$ which contains missing and/or corrupted observations of L_0. In addition, suppose we are also given additional feature matrices $X \in \mathbb{R}^{n_1 \times d_1}$ and/or $Y \in \mathbb{R}^{n_2 \times d_2}$ as side information, where each row $\mathbf{x}_i \in \mathbb{R}^{d_1}$ (or $\mathbf{y}_i \in \mathbb{R}^{d_2}$) denotes a feature representation of the i-th row (or column) entity of X (or Y). Then, instead of just using R to recover L_0, our hope is to leverage side information X and Y to learn L_0 more effectively. Below, we further list some important applications where the side information naturally comes in as the form of X and/or Y in this framework:

- *Collaborative filtering.* Collaborative filtering is one of the most popular machine learning applications in industry where we aim to predict the preferences of users to any products based on limited rating history (e.g. the Netflix problem we mentioned previously). A traditional approach is to complete the partial user-product rating matrix R via matrix completion. However, one could also collect per-user features x_i and per-product features y_j as possible information to leverage, and the assembled feature representation for users and products becomes X and Y in this framework.

- *Link prediction.* The link prediction problem in online social network analysis is to predict and recommend the implicit friendships of users given the current network snapshot. One approach is to think of the network snapshot as a user-to-user relationship matrix R, and thus any missing relationships in the snapshot can be inferred by conducting matrix completion on R (Liben-Nowell and Kleinberg, 2007). Similarly, if user-specific information (like user profile) is collected, these user features can be deemed as both X and Y.

- *Image denoising.* Another low-rank matrix learning application is image denoising. It is known that same types of images (e.g. images of human face, digits, or images with same scene) often share a common low-rank structure, and learning that low-dimensional space can be useful for many applications such as image recognition

1. We will use terms 'side information' and 'features' interchangeably throughout the paper.

and background subtraction. Yet in the realistic setting, images may be corrupted by sparse noise such as shadowing or brightness saturation, making the learning of that low-dimensional space much more difficult. A popular approach, known as robust PCA, is to construct an observed matrix R where each column is a vector representation of an image, and further learn the underlying low-rank subspace by separating it from the sparse noise in R. In Section 4, we will show that if features of clean images X and/or label-relevant features Y are also given, one can learn the underlying low-dimensional subspace more accurately.

Organization of the paper. To study the effect of side information in low-rank matrix learning with missing and corrupted observations, we focus on answering the following important questions in a systematical manner:

- What type of side information can benefit learning?
- What model should we use for incorporating side information?
- How can we further quantify the merits of side information in learning?

Regarding the first question, in Section 2, we start with the case of *"perfect" side information* (defined in equation 2) as an idealized case where the given features are fully informative, and further generalize to the case of *noisy side information* where the given features are only partially correlated to L_0. We will see that while information from perfect features is extremely useful, certain noisy features can also be quite effective to benefit learning.

The model for incorporating side information can also be constructed subsequently once the type of side information is identified. Precisely, in Section 2, we argue that for perfect features, one can directly transform the low-rank modeling matrix into a bilinear form with respect to features X and Y. However, the validity of such an embedding becomes questionable if features are noisy. Therefore, for noisy features, we propose to break the low-rank matrix into two parts—one that captures information from features and one that captures information outside the feature space—resulting in a general model (problem 4) that learns the low-rank matrix by jointly balancing information from noisy features and observations. In addition, we discuss the connections between our model and several well-known models, such as low-rank matrix completion and robust PCA. We also show that our proposed model can be efficiently solved by well-established optimization procedures.

Furthermore, in Section 3, we provide a theoretical analysis to justify the merits of side information in the proposed model (4). To start with, in Section 3.1, we quantify the quality of features and the noise level of corruption using Rademacher model complexity in the generalization analysis. As a result, a tighter error bound can be derived given better quality of features and/or lower noise level in observations. We further derive sample complexity guarantees for the case of matrix completion in Section 3.2 and for the case where observations are both missing and corrupted in Section 3.3. For the case of matrix completion, our sample complexity result suggests that the proposed model requires asymptotically fewer observations to recover the low-rank matrix compared to standard matrix completion, as long as the given features are sufficiently informative. This result substantially generalizes the previous study of side information in matrix completion in Jain and Dhillon (2013) which only guarantees improved complexity given perfect features. On the other hand, for

the case where observations are both missing and corrupted, our resulting sample complexity guarantee implies that better quality of side information is useful for learning missing entries of the low-rank matrix provided that the corruption is not too severe. These results thus justify the usefulness of side information in the proposed model in theory.

Finally, in Section 4, we verify the effectiveness of the proposed model experimentally on various synthetic data sets, and additionally apply it to three machine learning applications—relationship prediction, semi-supervised clustering and noisy image classification. We show that each of them can be tackled by learning a low-rank modeling matrix from missing or corrupted observations given certain additional features, and therefore, by employing our model to exploit side information, we can achieve better performance in these applications compared to other state-of-the-art methods. These results demonstrate that our proposed model indeed exploits side information for various low-rank matrix learning problems.

Here are the key contributions of this paper:

- We study the effect of side information and provide a general treatment to incorporate side information for learning low-rank matrices from missing and corrupted observations.
- In particular, given perfect side information, we propose to transform the estimated low-rank matrix to a bilinear form with respect to features. Moreover, given noisy side information, we propose to further break the low-rank matrix into a part capturing feature information plus a part capturing information outside the feature space, and therefore, learning can be conducted efficiently by balancing information between features and observations.
- We theoretically justify the usefulness of side information in the proposed model in various scenarios by first quantifying the effectiveness of features and then showing that the sample complexity can be asymptotically improved provided sufficiently informative features.
- We provide comprehensive experimental results to confirm that the proposed model properly embeds both perfect and noisy side information for learning low-rank matrices more effectively compared to other state-of-the-art approaches.

Parts of this paper have previously appeared in Chiang et al. (2015) and Chiang et al. (2016), in which we exclusively studied the effect of noisy side information in matrix completion and the effect of perfect side information in robust PCA, respectively. In this paper, we consider a much more general setting and propose a general model to exploit side information for a broader class of low-rank matrix learning problems. In particular, given this general model, we can further exploit noisy side information for the robust PCA problem and for the case where observations are *both* missing and corrupted as we will discuss in Section 2.3. We also provide much more comprehensive theoretical and experimental results to demonstrate the effectiveness of the proposed treatment.

2. Exploiting Side Information for Learning Low-Rank Matrices

In this section, we discuss how to incorporate side information for learning low-rank matrices from missing and corrupted observations. We first introduce the problem formulation in Section 2.1. We then start with exploiting perfect, noiseless side information in Section 2.2 and introduce the proposed model which can further exploit noisy side information in Section 2.3. We finally describe the optimization for solving the proposed model in Section 2.4.

2.1. Learning from Missing and Corrupted Observations

The problem of learning a low-rank matrix from missing and corrupted observations can be formally stated as follows. Let $L_0 \in \mathbb{R}^{n_1 \times n_2}$ be the underlying rank-r matrix where $r \ll \min(n_1, n_2)$ so that L_0 is low-rank, and S_0 be a noise matrix whose support (denoted as Ω) and magnitude is unknown but the structure is known to be *sparse*. Furthermore, let Ω_{obs} be a set of observed entries with cardinality m, and $\mathcal{P}_{\Omega_{obs}}$ be the orthogonal projection operator defined by:

$$\mathcal{P}_{\Omega_{obs}}(X)_{ij} = \begin{cases} X_{ij}, & \text{if } (i,j) \in \Omega_{obs}, \\ 0, & \text{otherwise.} \end{cases}$$

Then, given the observed data matrix R which is in the form of:

$$R = \mathcal{P}_{\Omega_{obs}}(L_0 + S_0) = \mathcal{P}_{\Omega_{obs}}(L_0) + S_0',$$

the goal is to accurately estimate the underlying matrix L_0 given R. Without loss of generality, we assume that S_0 is supported on Ω_{obs}, i.e. $\Omega \subseteq \Omega_{obs}$ and $S_0' = S_0$. Note that this problem can be viewed as an extension of the matrix completion problem, which only assumes the given observations to be undersampled yet noiseless (Ω is the empty set).

An intuitive way to approach this problem is to estimate the low-rank matrix based on the given structural information of the problem. Specifically, Candès et al. (2011) proposed to solve this problem via the following convex program:

$$\min_{L,S} \ \|L\|_* + \lambda \|S\|_1 \quad \text{s.t.} \quad L_{ij} + S_{ij} = R_{ij}, \ \forall \ (i,j) \in \Omega_{obs}, \tag{1}$$

where $\|L\|_*$ is the nuclear norm of L defined by the sum of singular values of L, and $\|S\|_1 := \sum_{i,j} |S_{ij}|$ is the element-wise one norm of S. These two regularizations are known to be useful for enforcing low rank structure and sparse structure, respectively.

Although problem (1) has been shown to enjoy theoretical and empirical success (Candès et al., 2011), it cannot directly leverage side information for recovery if it is provided. A tailored model is thus required to resolve this issue.

2.2. Idealized Case: Perfect Side Information

Suppose in addition to the data matrix R, we are also given features of row and column entities $X \in \mathbb{R}^{n_1 \times d_1}$ and $Y \in \mathbb{R}^{n_2 \times d_2}$, $d_1 < n_1$ and $d_2 < n_2$ as side information. Then, the goal of low-rank matrix learning with side information is to exploit X and Y in addition to the observations R to better estimate L_0. A concrete example is the Netflix problem where R corresponds to the partial user-movie rating matrix and, X and Y correspond to user and movie features; the hope is to further leverage additional features X and Y along with rating history R to better predict the unknown user-movie ratings.

In principle, not all types of side information will be useful. For instance, if the given X and Y are simply two random matrices, then there is no information gain from the provided side information, and therefore, *any* method incorporating such X and Y is expected to perform the same as methods only using structural information. That being said, to explore the advantage of side information, a condition on side information to ensure its informativeness is required. To begin with, we consider an ideal scenario where the side information is "perfect" in the sense that it implicitly describes the full latent space of L_0.

Definition 1 (Perfect side information) *The side information X and Y is called perfect side information, or noiseless side information, w.r.t. L_0 if X and Y satisfy:*

$$col(X) \supseteq col(L_0), \quad col(Y) \supseteq col(L_0^T), \tag{2}$$

where $col(X)$ and $col(Y)$ denotes the column space of X and Y.

Then, consider $L_0 = U\Sigma V^T$ to be the SVD of L_0, a set of perfect side information will also satisfy $\text{col}(X) \supseteq \text{col}(U)$ and $\text{col}(Y) \supseteq \text{col}(V)$, which further indicates that there exists a matrix $M_0 \in \mathbb{R}^{d_1 \times d_2}$ such that $L_0 = XM_0Y^T$. This fact leads us to expressing the target low-rank matrix as a bilinear form with respect to features X and Y, and as a result, one can cast problem (1) with features as:

$$\min_{M,S} \ \|M\|_* + \lambda \|S\|_1 \quad \text{s.t.} \ \mathbf{x}_i^T M \mathbf{y}_j + S_{ij} = R_{ij}, \ \forall \ (i,j) \in \Omega_{obs}, \tag{3}$$

in which the problem is reduced to learning a smaller $d_1 \times d_2$ low-rank matrix M. The bilinear embedding with respect to perfect features for the low-rank matrix has already been proposed in matrix completion. Indeed, by casting $L = XMY^T$ as matrix completion, one can obtain a so-called "inductive matrix completion" (IMC) model which is able to learn the underlying matrix with much fewer samples given perfect side information (Jain and Dhillon, 2013; Xu et al., 2013; Zhong et al., 2015). We will discuss the improved sample complexity result of IMC in detail in Section 3.2.

However, an obvious weakness of the bilinear embedding in problem (3) is that it assumes the given side information to be perfect. Unfortunately, in real applications, most given features X and Y will not be perfect, and could be in fact noisy or only weakly correlated to the latent space of L_0. In such cases, L_0 can no longer be expressed as XMY^T and thus the translated objective (3) becomes questionable to use. This weakness will also be empirically shown in Section 4 in which we observe that the recovered matrix XM^*Y^T of problem (3) will diverge from L_0 given noisy side information in experiments. Nevertheless, it is arguable that certain noisy features should still be helpful for learning L_0. For example, given the SVD of $L_0 = U\Sigma V^T$, a small perturbation of a single entry of U (or V) makes the perturbed U, V to be imperfect features, yet such U and V should still be very informative. This observation thus motivates us to design a more general model to exploit noisy side information.

2.3. The Proposed Model: Exploiting Noisy Side Information

We now introduce an improved model to further exploit imperfect, noisy side information. The key idea of our model is to balance both feature information and observations when learning the low-rank matrix. Specifically, we propose to learn L_0 jointly in two parts, one part captures information from the feature space as XMY^T, and the other part N captures the information outside the feature space. Thus, even if the given features are noisy and fail to cover the full latent space of L_0, we can still capture missing information using N learned from pure observations.

However, there is an identifiability issue if we simply learn L_0 with the expression XMY^T+N, since there are infinitely many solutions of (M,N) that satisfy $XMY^T+N=L_0$. Although in theory they all perfectly recover the underlying matrix, some of the solutions shall be more preferred than others if we further consider the efficiency of learning. Intuitively, since the underlying L_0 is low-rank, a natural thought is to prefer both XMY^T and N to be low-rank so that the L_0 can be recovered with fewer parameters. This preference leads us to pursue a low-rank M as well, which conceptually means that only a small subspace of X and a subspace of Y are expected to be effective in jointly forming a low-rank estimate XMY^T. Pursuing low-rank solutions of M and N enables us to accurately estimate L_0 with fewer samples because fewer parameters need to be learned compared to other solutions. This advantage will be formally justified later in Section 3.

Therefore, putting this all together, to incorporate noisy side information and learn the low-rank matrix L_0 from missing and corrupted observations, we propose to solve the following problem:

$$\min_{M,N,S} \sum_{(i,j)\in\Omega_{obs}} \ell((XMY^T+N+S)_{ij}, R_{ij}) + \lambda_M\|M\|_* + \lambda_N\|N\|_* + \lambda_S\|S\|_1 \qquad (4)$$

with some convex surrogate loss ℓ, and the underlying matrix L_0 can be estimated by $XM^*Y^T+N^*$, where (M^*,N^*,S^*) is the optimal solution of problem (4). Note that to force M and N to be low-rank, in the proposed objective we add nuclear norm regularization on *both* variables M and N. It is known that nuclear norm regularization is one of the most popular heuristic to pursue low-rank structure as it is the tightest convex relaxation of the rank function (Fazel et al., 2001). In particular, given a low-rank matrix $\text{rank}(R)\le r$ and $\max_{ij}|R_{ij}|\le C_L$, we always have:

$$\|R\|_* \le \sqrt{r}\|R\|_F \le C_L\sqrt{rn_1n_2},$$

and thus, a nuclear norm regularized constraint $\|R\|_*\le t$ can be thought of as a relaxed condition of $\text{rank}(R)\le r$ and $\max_{ij}|R_{ij}|\le t/\sqrt{rn_1n_2}$.

The proposed problem (4) is also a general formulation to better exploit side information for learning low-rank matrices from missing and corrupted observations. This fact can be seen by considering the following equivalent form of problem (4) which converts the loss term to hard constraints:

$$\min_{M,N,S} \alpha\|M\|_* + \beta\|N\|_* + \lambda\|S\|_1 \quad \text{s.t. } (XMY^T+N+S)_{ij} = R_{ij}, \forall (i,j)\in\Omega_{obs}. \qquad (5)$$

Then, it is easy to see that by setting $\alpha=\infty$ or $\beta=\infty$, problem (5) will become problem (1) or problem (3), which learns the low-rank matrix from missing and corrupted observations

either without any side information or using perfect side information, respectively. This suggests that our model (4) is more general as it can exploit both perfect and noisy side information in learning.

The parameters λ_M, λ_N and λ_S of the model are crucial for controlling the contributions from features, observations and corruption. Intuitively, λ_S controls the ratio of corrupted observations. The relative weight between λ_M and λ_N further controls the contributions from XMY^T and N in forming the low-rank estimate. Therefore, with an appropriate ratio between λ_M,λ_N, the proposed model can leverage a (informative) part of the features XMY^T, yet also be robust to feature noise by learning the remaining part N from pure observations. Below, we further discuss the connections between our model (4) and other well-known models for solving various low-rank matrix learning problems.

2.3.1. Connections to models for matrix completion

First, consider the matrix completion case where the partially observed entries are not corrupted. Then, λ_S can be set to ∞ to force $S^*=0$, and therefore, our proposed problem (4) reduces to the following objective:

$$\min_{M,N} \sum_{(i,j)\in\Omega_{obs}} \ell((XMY^T+N)_{ij}, R_{ij}) + \lambda_M\|M\|_* + \lambda_N\|N\|_*, \qquad (6)$$

which is a general model for solving matrix completion problem. For example, when $\lambda_M=\infty$, M^* will be forced to 0 so features are disregarded, and problem (6) becomes a standard matrix completion objective. On the other hand, when $\lambda_N=\infty$, N^* will be forced to 0 and problem (6) becomes the IMC model (Jain and Dhillon, 2013; Xu et al., 2013) where the estimation of the low-rank matrix is completely from XM^*Y^T. However, problem (6) is more general than both problems, since by appropriately setting the weights of λ_M and λ_N, it can better estimate the low-rank matrix jointly from (noisy) features XM^*Y^T and pure observations N^*. Therefore, problem (6) can be thought of as an improved model which exploits noisy side information in matrix completion problem. We thus refer to problem (6) as "IMC with Noisy Features" (IMCNF) and will justify its effectiveness for matrix completion in Section 4.

2.3.2. Connections to models for robust PCA

Another special case is to consider the well-known "robust PCA" setting, in which Ω_{obs} is assumed to be the set of all $n_1\times n_2$ entries, i.e. observations are full without any missing entries but few of them are corrupted. In this scenario, our proposed problem (4) can be used for solving robust PCA problem with side information by again converting the loss term to hard constraints:

$$\min_{M,N,S} \alpha\|M\|_* + \beta\|N\|_* + \lambda\|S\|_1 \quad \text{s.t. } XMY^T+N+S=R. \qquad (7)$$

Problem (7) can be further reduced to several robust PCA models. For example, if $\alpha=\infty$, problem (7) will be equivalent to the well-known PCP method (Candès et al., 2011) which solves robust PCA problem purely using a structural prior. On the other hand, suppose side information is perfect, then one can set $\beta=\infty$ in (7) to derive the following "PCP

Model	Corresponding setting in our proposed model (4)
problem (1) (Candès et al., 2011)	$\lambda_M = \infty$
problem (3)	$\lambda_N = \infty$
MC	$\lambda_S = \infty, \lambda_M = \infty$
IMC (Jain et al., 2013)	$\lambda_S = \infty, \lambda_N = \infty$
IMCNF	$\lambda_S = \infty$
LRR (Liu et al., 2013)	Ω_{obs} = all entries, $\lambda_N = \infty$, $Y = I$
PCP (Candès et al., 2011)	Ω_{obs} = all entries, $\lambda_M = \infty$
PCPF	Ω_{obs} = all entries, $\lambda_N = \infty$
PCPNF	Ω_{obs} = all entries

Table 1: Settings of several low-rank matrix learning models in the form of our proposed problem (4).

with (perfect) Features" (PCPF) objective:

$$\min_{M,S} \ \alpha\|M\|_* + \lambda\|S\|_1 \quad \text{s.t. } XMY^T + S = R, \tag{8}$$

in which L_0 can be directly estimated by the bilinear embedding XM^*Y^T as discussed in Section 2.2. However, problem (7) is more general than both PCP and PCPF as it can exploit noisy side information for recovery. We thus refer to (7) as "PCP with Noisy Features" (PCPNF) and will examine its effectiveness to leverage noisy side information in robust PCA in Section 4.

Table 1 summarizes several well-known low-rank matrix learning models in terms of the proposed model (4). [2] From the above discussion, it shall be convincing that problem (4) is a general treatment for solving various matrix learning problems with side information. In particular, we have provided sufficient intuitions on how parameters λ_M, λ_N and λ_S play important roles in learning under various circumstances. In Section 3, we will further analytically show that by properly setting these parameters based on the quality of features and noise level of corruption, the proposed model is able to achieve more efficient learning. As a remark, in practical applications, feature quality and noise level may not be known a priori. Therefore, in this case, we recommend to set these parameters via validation, i.e. choosing parameters such that the learned low-rank model best estimates the entries in the validation set.

2.4. Optimization

We propose an alternative minimization scheme to solve the proposed problem (4). The algorithm is shown in Algorithm 1 in which we alternatively update one of the variables (M, N or S) by fixing the others in each iteration, [3] and update of each variable can thus be done via solving a single variable minimization (sub)problem. This algorithm can be viewed as applying a block coordinate descent algorithm on a convex and continuous function, and in

2. Some models are originally proposed in hard-constrained forms, yet their equivalent forms in soft constraints become instances of our proposed problem (4).
3. For simplicity, we choose $\ell(t, y) = (t - y)^2$ to be the squared loss.

Algorithm 1: Alternative Minimization for Problem (4) with Squared Loss

Input: R: observed matrix, X, Y: feature matrices, t_{max}: max iteration
Output: L^*: estimated low-rank matrix
$M \leftarrow 0, \quad N \leftarrow 0, \quad S \leftarrow 0, \quad t \leftarrow 0$
do
  $M \leftarrow \arg\min_M \sum_{(i,j)\in\Omega_{obs}} ((XMY^T)_{ij} - (R - N - S)_{ij})^2 + \lambda_M \|M\|_*$.
  $N \leftarrow \arg\min_N \sum_{(i,j)\in\Omega_{obs}} (N_{ij} - (R - XMY^T - S)_{ij})^2 + \lambda_N \|N\|_*$.
  $S \leftarrow \arg\min_S \sum_{(i,j)\in\Omega_{obs}} (S_{ij} - (R - XMY^T - N)_{ij})^2 + \lambda_S \|S\|_1$.
  $t \leftarrow t + 1$.
while *not converged* **and** $t < t_{\max}$
$L^* \leftarrow XMY^T + N$

such case the cyclic block coordinate descent algorithm is guaranteed to converge to global minimums (see Tseng, 2001). The condition required in Tseng (2001) is that the level set has to be compact, which is satisfied when $\lambda_M, \lambda_N, \lambda_S > 0$.

We now briefly discuss the optimization for solving three subproblems in Algorithm 1. Let $\mathcal{S}_x(A) := \text{sign}(A) \circ \max(|A| - x, 0)$ be the soft thresholding operator on elements of A, where $\circ$ denotes the element-wise product. Similarly, let $\mathcal{D}_x(A)$ be the thresholding operator on singular values of A, i.e. $\mathcal{D}_x(A) := U_A S_x(\Sigma_A) V_A^T$ where $U_A \Sigma_A V_A^T$ is the SVD of A. Then, when fixing N and S, the minimization problem over M becomes a standard IMC objective with observed matrix to be $R' := R - N - S$. We then solve for M using typical proximal gradient descent update $M \leftarrow \mathcal{D}_{\lambda_M}(M - \eta X^T(R' - XMY^T)Y)$, where η is the learning rate. Notice that in our setting, feature dimensions (d_1, d_2) are much smaller than number of entities (n_1, n_2). Therefore, it is relatively inexpensive to compute a full SVD for a $d_1 \times d_2$ matrix in each proximal step.

On the other hand, when fixing M and S, the subproblem of solving over N becomes standard matrix completion problem where the observed matrix is $R - XMY^T - S$. In principle, any algorithm for matrix completion with nuclear norm regularization can be used to solve this subproblem (e.g. the singular value thresholding algorithm (Cai et al., 2010) using proximal gradient descent). In our experiment, we apply the active subspace selection algorithm (Hsieh and Olsan, 2014) to solve the matrix completion problem more efficiently.

Finally, the solution of minimizing over S given fixed M, N can be written in a simple closed form, $\mathcal{S}_{\lambda_S}(\mathcal{P}_{\Omega_{obs}}(R - XMY^T - N))$. The resulting S^*, therefore, will be always supported on Ω_{obs}.

3. Theoretical Analysis on the Effect of Side Information

In this section, we provide a theoretical analysis to justify the usefulness of side information in our model (4). We will focus on the *sample complexity* analysis of the model, in which we aim to show that by exploiting side information, learning can be accomplished with fewer number of (possibly corrupted) observations. The high-level idea of the analysis is to consider the generalization error of the estimated entries, which is associated to both

number of samples and a model complexity term. We further show that model complexity can be related to the quality of features and the noise level of sparse error, and as a result, better feature quality will lead to a smaller generalization error and also a better sample complexity guarantee, provided a small enough noise level. To concentrate on the whole picture of the analysis, we leave detailed proofs of theorems, corollaries and lemmas in Appendix A.

3.1. Generalization Bound of the Proposed Model

To begin with, we consider the equivalent hard-constrained form of problem (4):

$$\min_{M,N,S} \sum_{(i,j)\in\Omega_{obs}} \ell((XMY^T + N + S)_{ij}, R_{ij}), \quad s.t.\ \|M\|_* \le \mathcal{M}, \|N\|_* \le \mathcal{N}, \|S\|_1 \le \mathcal{S}. \quad (9)$$

In the analysis, we assume that each entry $(i,j) \in \Omega_{obs}$ is sampled i.i.d. from an *unknown* distribution $\mathcal{D}$ with index set $\{(i_\alpha, j_\alpha)\}_{\alpha=1}^m$, [4] and each entry of L_0 is upper bounded by a constant C_L (so $\|L_0\|_* = O(\sqrt{n_1 n_2})$). Such a circumstance is consistent with real scenarios such as Netflix problem where users can rate movies with scale up to 5. Let $\theta := (M, N, S)$ be any feasible solution and $\Theta := \{(M,N,S) \mid \|M\|_* \le \mathcal{M}, \|N\|_* \le \mathcal{N}, \|S\|_1 \le \mathcal{S}\}$ be the set of feasible solutions. Also, let $f_\theta \in [n_1] \times [n_2] \to \mathbb{R}$, $f_\theta(i,j) := \mathbf{x}_i^T M \mathbf{y}_j + \mathbf{e}_i^T N \mathbf{e}_j + \mathbf{e}_i^T S \mathbf{e}_j$ be the estimation function (parameterized by θ) where $\mathbf{e}_t$ is the unit vector on the t-th axis, and let $F_\Theta := \{f_\theta \mid \theta \in \Theta\}$ be the set of feasible functions. We are interested in both expected and empirical "ℓ-risk" quantities, $R_\ell(f)$ and $\hat{R}_\ell(f)$, defined by:

$$R_\ell(f) := \mathbb{E}_{(i,j)\sim\mathcal{D}}\big[\ell(f(i,j), \mathbf{e}_i^T(L_0 + S_0)\mathbf{e}_j)\big], \qquad \hat{R}_\ell(f) := \frac{1}{m}\sum_{(i,j)\in\Omega_{obs}} \ell(f(i,j), R_{ij}).$$

Under this context, our model (problem 9) is to solve for θ^* that parameterizes $f^* = \arg\min_{f\in F_\Theta} \hat{R}_\ell(f)$. Classic generalization error bounds have shown that the expected risk $R_\ell(f)$ can be controlled by $\hat{R}_\ell(f)$ along with a measurement on the complexity of the model. The following lemma is a typical result to bound $R_\ell(f)$:

Lemma 2 (Bound on Expected ℓ-risk, Bartlett and Mendelson, 2003) *Let ℓ be a Lipschitz loss function and is bounded by $\mathcal{B}$ with respect to its first argument, and δ be a constant where $0 < \delta < 1$. Let $\mathfrak{R}(F_\Theta)$ be the Rademacher model complexity of the function class F_Θ (w.r.t. Ω_{obs}) defined by:*

$$\mathfrak{R}(F_\Theta) := \mathbb{E}_\sigma\big[\sup_{f\in F_\Theta} \frac{1}{m}\sum_{\alpha=1}^m \sigma_\alpha \ell(f(i_\alpha, j_\alpha), R_{i_\alpha j_\alpha})\big],$$

where each σ_α takes values $\{\pm 1\}$ with equal probability. Then with probability at least $1-\delta$, for all $f \in F_\Theta$ we have:

$$R_\ell(f) \le \hat{R}_\ell(f) + 2\mathbb{E}_{\Omega_{obs}}\big[\mathfrak{R}(F_\Theta)\big] + \mathcal{B}\sqrt{\frac{\log\frac{1}{\delta}}{2m}}.$$

4. In other words, we consider the observations to be sampled under a sampling with replacement model which is similar to Recht (2011); Shamir and Shalev-Shwartz (2014). There are also studies that consider other sampling procedures such as Bernoulli model (Candès and Tao, 2009; Candès and Recht, 2012).

Therefore, to guarantee a small enough R_ℓ, not only $\hat{R}_\ell$, but also the Rademacher model complexity $\mathbb{E}_{\Omega_{obs}}\big[\mathfrak{R}(F_\Theta)\big]$ has to be carefully controlled. We further introduce a key lemma to show that the model complexity is related to both the feature quality and the sparse noise level, where better quality of features and lower noise level will lead to a smaller model complexity. The intuition of the goodness of feature quality can be motivated as follows. Consider any imperfect side information which violates (2). One can imagine such a feature set is perturbed by some misleading noise which is not correlated to the true latent space. However, features should still be effective if noise does not weaken the true latent space information too much. Thus, if a large portion of true latent space lies on the informative part of the feature spaces X and Y, they should still be somewhat informative and helpful for recovering the matrix L_0.

More formally, for F_Θ in problem (9), its model complexity $\mathbb{E}_{\Omega_{obs}}\big[\mathfrak{R}(F_\Theta)\big]$ can be bounded in terms of $\mathcal{M}$, $\mathcal{N}$ and $\mathcal{S}$ by the following lemma:

Lemma 3 *Let $\mathcal{X} = \max_i \|\mathbf{x}_i\|_2$, $\mathcal{Y} = \max_i \|\mathbf{y}_i\|_2$, $n = \max(n_1, n_2)$ and $d = \max(d_1, d_2)$. Suppose ℓ is a convex surrogate loss satisfying conditions in Lemma 2 with the Lipschitz constant L_ℓ. Then for F_Θ in problem (9), its model complexity $\mathbb{E}_{\Omega_{obs}}\big[\mathfrak{R}(F_\Theta)\big]$ is upper bounded by:*

$$2L_\ell\mathcal{M}\mathcal{X}\mathcal{Y}\sqrt{\frac{\log 2d}{m}} + \min\left\{2L_\ell\mathcal{N}\sqrt{\frac{\log 2n}{m}}, \sqrt{\frac{9CL_\ell\mathcal{B}\mathcal{N}(\sqrt{n_1}+\sqrt{n_2})}{m}}\right\} + L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}},$$

where L_ℓ and $\mathcal{B}$ are constants appearing in Lemma 2.

Thus, from Lemma 2 and 3, one should carefully construct a feasible solution set (by setting $\mathcal{M}$, $\mathcal{N}$ and $\mathcal{S}$) such that both $\hat{R}_\ell(f^*)$ and $\mathbb{E}_{\Omega_{obs}}\big[\mathfrak{R}(F_\Theta)\big]$ are controlled to be reasonably small. We now suggest a witness setting of $(\mathcal{M}, \mathcal{N}, \mathcal{S})$ as follows. Let $\mathcal{T}_\mu(\cdot) : \mathbb{R}^+ \to \mathbb{R}^+$ be the thresholding operator where $\mathcal{T}_\mu(x) = x$ if $x \ge \mu$ and $\mathcal{T}_\mu(x) = 0$ otherwise. In addition, let $X = \sum_{i=1}^{d_1}\sigma_i\mathbf{u}_i\mathbf{v}_i^T$ be the reduced SVD of X, and $X_\mu = \sum_{i=1}^{d_1}\sigma_1\mathcal{T}_\mu(\sigma_i/\sigma_1)\mathbf{u}_i\mathbf{v}_i^T$ be the "μ-informative" part of X. The ν-informative part of Y, denoted as Y_ν, can also be defined similarly. We then propose to set:

$$\mathcal{M} = \|\hat{M}\|_* \qquad \mathcal{N} = \|L_0 - X_\mu\hat{M}Y_\nu^T\|_* \qquad \mathcal{S} = \|S_0\|_1, \quad (10)$$

where $\hat{M} := \arg\min_M \|X_\mu M Y_\nu^T - L_0\|_F^2 = (X_\mu^T X_\mu)^\dagger X_\mu^T L_0 Y_\nu (Y_\nu^T Y_\nu)^\dagger$ is the optimal solution for approximating L_0 under the informative feature spaces X_μ and Y_ν. The following lemma further shows that the trace norm of $\hat{M}$ will not grow as a function of n.

Lemma 4 *Fix $\mu, \nu \in (0, 1]$, and let γ be a constant defined by*

$$\gamma := \min\left(\frac{\min_i\|\mathbf{x}_i\|}{\mathcal{X}}, \frac{\min_i\|\mathbf{y}_i\|}{\mathcal{Y}}\right)$$

where $\mathcal{X}, \mathcal{Y}$ are constants defined in Lemma 3. Then the trace norm of $\hat{M}$ is upper bounded by:

$$\|\hat{M}\|_* \le \frac{C_L d^2}{\mu^2\nu^2\gamma^2\mathcal{X}\mathcal{Y}},$$

where $C_L \ge \max_{i,j}|\mathbf{e}_i^T L_0 \mathbf{e}_j|$ is the constant upper bounding the entries of L_0.

Therefore, by combining Lemma 2-4, we derive a generalization error bound on $R_\ell(f^*)$ of problem (9) as follows.

Theorem 5 *Suppose ℓ is a convex surrogate loss function with Lipschitz constant L_ℓ bounded by $\mathcal{B}$ with respect to its first argument and assume that $\ell(t,t) = 0$. Consider problem (9) where the constraints $(\mathcal{M}, \mathcal{N}, \mathcal{S})$ are set as (10) with some fixed $\mu, \nu \in (0,1]$. Then with probability at least $1-\delta$, the expected ℓ-risk of the optimal solution $R_\ell(f^*)$ is bounded by:*

$$R_\ell(f^*) \le \min\left\{4L_\ell\mathcal{N}\sqrt{\frac{\log 2n}{m}}, \sqrt{36CL_\ell\mathcal{B}\frac{\mathcal{N}(\sqrt{n_1}+\sqrt{n_2})}{m}}\right\} + 2L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}} + \frac{4L_\ell C_L d^2}{\mu^2\nu^2\gamma^2}\sqrt{\frac{\log 2d}{m}} + \mathcal{B}\sqrt{\frac{\log\frac{1}{\delta}}{2m}},$$

where C, C_L and γ are constants appearing in Lemma 3 and 4.

As a result, Theorem 5 leads us to deem $\mathcal{N}$ and $\mathcal{S}$ in (10) to be the measurement of feature quality and noise level respectively, where features with better quality (or observations with less corruption) lead to a smaller $\mathcal{N}$ (or $\mathcal{S}$) and thus a smaller risk quantity. Note that the measurement $\mathcal{N}$ is consistent with the stated intuition of feature quality, since given a good feature set such that most true latent space of L_0 lies on the informative part of the feature spaces, $X_\mu\hat{M}Y_\nu^T$ will absorb most of L_0, resulting in a small $\mathcal{N}$. Given Theorem 5, we can further discuss the effect of side information in the proposed model (9) on the sample complexity in several important scenarios. To make the comparison more clear, we fix $d = O(1)$ so the feature dimensions do not grow as a function of n in the following discussion.

3.2. Sample Complexity for Matrix Completion

First, consider the matrix completion case where the observations are partial yet not corrupted, i.e. $S_0 = 0$. Then, as mentioned, our model can be further reduced to IMCNF (problem (6), or equivalently problem (9) with $\mathcal{S} = 0$) which exploits noisy side information to solve the matrix completion problem. In addition, from Theorem 5, we can derive the sample complexity of IMCNF as follows.

Corollary 6 *Suppose we aim to (approximately) recover L_0 from partial observations $R = \mathcal{P}_{\Omega_{obs}}(L_0)$ in the sense that $\mathbb{E}_{(i,j)\sim\mathcal{D}}\left[\ell((XM^*Y^T+N^*)_{ij}, \mathbf{e}_i^TL_0\mathbf{e}_j)\right] < \epsilon$ given an arbitrary $\epsilon > 0$. Then by solving problem (9) with constraints to be set as (10), $O(\min(\mathcal{N}\sqrt{n}, \mathcal{N}^2\log n)/\epsilon^2)$ samples are sufficient to guarantee that the estimated low-rank matrix $XM^*Y^T+N^*$ recovers L_0 with high probability, provided a sufficiently large n.*

Corollary 6 suggests that the sample complexity of IMCNF can be lowered with the aid of (sufficiently informative) noisy side information. The significance of this result can be further explained by comparing with the sample complexity of other models. First, if features are perfect ($\mathcal{N} = O(1)$), Corollary 6 suggests that our IMCNF model only requires $O(\log n)$ samples for recovery. This result coincides with the sample complexity of IMC, in which researchers have shown that given perfect features, $O(\log n)$ observations are enough

for exact recovery (Xu et al., 2013; Zhong et al., 2015). However, IMC does not guarantee recovery when features are not perfect, while Corollary 6 suggests that recovery is still attainable by IMCNF with $O(\min(\mathcal{N}\sqrt{n}, \mathcal{N}^2\log n)/\epsilon^2)$ samples.

On the other hand, our analysis suggests that sample complexity of IMCNF is at most $O(n^{3/2})$ given any features by applying the following inequality to Corollary 6:

$$\mathcal{N} \le \|L_0\|_* \le C_L\|E\|_* \le C_L\sqrt{\text{rank}(E)}\|E\|_F = C_L\sqrt{n_1n_2} = O(n),$$

where $E \in \mathbb{R}^{n_1\times n_2}$ is the matrix with all entries to be one. To explain the result, we compare this result to the sample complexity of standard matrix completion where no side information is considered. At the first glance, it may appear that the result is worse than pure matrix completion in the worst case, since many well-known matrix completion guarantees showed that under certain spikiness and distributional conditions, one can achieve $O(n\,\text{polylog}\,n)$ sample complexity for both approximate recovery (Srebro and Shraibman, 2005; Negahban and Wainwright, 2012) and exact recovery (Candès and Recht, 2012). However, all of the above $O(n\,\text{polylog}\,n)$ results require additional distributional assumptions on observed entries, while our analysis does not make distributional assumptions. To make a fairer comparison, Shamir and Shalev-Shwartz (2014) have shown that for pure matrix completion, $O(n^{3/2})$ entries are sufficient for approximate recovery without any distributional assumptions, and furthermore, the bound is *tight* if no further distributional assumptions on observed entries is allowed. Therefore, Corollary 6 indicates that IMCNF is at least as good as pure matrix completion even *in the worst case* under the distribution-free setting. Notice that it is reasonable to meet the matrix completion lower bound $\Omega(n^{3/2})$ even given features, since for completely useless feature case (e.g. X, Y are random matrices), the given information is exactly the same as that in standard matrix completion, so any method cannot beat the matrix completion lower bound even by taking features into account.

However, in most applications, the given features are expected to be far from random, and Corollary 6 provides a theoretical insight to show that even noisy features can be useful in matrix completion. Indeed, as long as features are informative enough such that $\mathcal{N} = o(n)$, sample complexity of the IMCNF model will be asymptotically lower than standard matrix completion. Here we provide a concrete example for such a scenario. We consider the rank-r matrix L_0 to be generated from random orthogonal model (Candès and Recht, 2012) as follows:

Corollary 7 *Let $L_0 \in \mathbb{R}^{n\times n}$ be generated from random orthogonal model, where $U = \{\mathbf{u}_i\}_{i=1}^r$, $V = \{\mathbf{v}_i\}_{i=1}^r$ are random orthogonal bases, and $\sigma_1 \dots \sigma_r$ are singular values with arbitrary magnitude. Let σ_t be the largest singular value such that $\lim_{n\to\infty}\sigma_t/\sqrt{n} = 0$. Then, given the noisy features X, Y where $X_{:i} = \mathbf{u}_i$ (and $Y_{:i} = \mathbf{v}_i$) if $i < t$ and $X_{:i}$ (and $V_{:i}$) be any basis orthogonal to U (and V) if $i \ge t$, $o(n)$ samples are sufficient for IMCNF to achieve recovery of L_0.*

Corollary 7 suggests that, under random orthogonal model, if features are not too noisy in the sense that noise only perturbs the true subspace associated with smaller singular values, the sample complexity of IMCNF can be asymptotically lower than the lower bound of standard matrix completion (which is $\Omega(n^{3/2})$).

All in all, for the matrix completion case where observations are partial yet uncorrupted, our proposed problem (4) reduces to the IMCNF model (6) and moreover, Corollary 6

suggests that it can attain recovery more efficiently than other existing models by exploiting noisy yet informative side information.

3.3. Sample Complexity given Partial and Corrupted Observations

We now further consider the case where observations are *both* missing and corrupted. In the presence of corruption, Theorem 5 results in the following Corollary 8 which shows that the learned matrix $XM^*Y^T + N^* + S^*$ will be close to $L_0 + S_0$ with sufficient observations, where the number of required samples depends on both the quality of features and the noise level of sparse error. Since there always exists a solution of problem (9) with $\mathcal{P}_{\Omega_{obs}^{\perp}}(S^*) = 0$ and the generalization bound in Theorem 5 holds for any solution, the result in Corollary 8 implies that $XM^*Y^T + N^*$ is close to L_0 on *missing entries* $(i,j) \notin \Omega_{obs}$, which means we can recover the missing entries of the underlying low-rank matrix with small error. Moreover, if we apply the proposed Algorithm 1 to solve the soft-constrained form (4), the solution S^* will satisfy $\mathcal{P}_{\Omega_{obs}^{\perp}}(S^*) = 0$ automatically. In the following, we formally state the recovery guarantee for partial and corrupted observations:

Corollary 8 *Suppose we are given a data matrix $R = \mathcal{P}_{\Omega_{obs}}(L_0 + S_0)$ containing both missing and corrupted observations of L_0 along with side information X, Y. Then for problem* (9) *with constraints to be set as* (10)*, if we apply Algorithm 1 to solve its equivalent form in* (4)*, $O(\{\min(\mathcal{N}\sqrt{n}, \mathcal{N}^2 \log n) + \mathcal{S}^2 \log n\}/\epsilon^2)$ samples are sufficient to guarantee that with high probability, $\mathbb{E}_{(i,j)\sim\mathcal{D}}\big[\ell((XM^*Y^T + N^* + S^*)_{ij}, R_{ij})\big] < \epsilon$ for any $\epsilon > 0$ provided a sufficiently large n, where S^* satisfies $\mathcal{P}_{\Omega_{obs}^{\perp}}(S^*) = 0$.*

Corollary 8 suggests that if observations are both missing and corrupted, then to guarantee the learned low-rank matrix $XM^*Y^T + N^*$ is accurate on missing entries, the number of required samples depends not only on the quality of features $\mathcal{N}$, but also on the noise level of corruption $\mathcal{S}$. In addition, larger $\mathcal{S}$ results in a higher complexity guarantee. The reasoning behind this result is intuitive: compared to the matrix completion setting in Section 3.2, allowing observed samples to be corrupted makes the problem harder, and therefore may increase sample complexity. However, suppose the corruption is not too severe as the total magnitude of error $\mathcal{S}$ is in the order of $o(n/\sqrt{\log n})$, Corollary 8 still provides a non-trivial bound on required samples for learning the missing entries accurately. Furthermore, better quality of features becomes helpful for faster learning if corruption is small enough. For example, suppose the allowed corruption budget is upper bounded as $\mathcal{S} = O(1)$, then the sample complexity will again be $O(\min(\mathcal{N}\sqrt{n}, \mathcal{N}^2 \log n)/\epsilon^2)$. As discussed, it implies that the number of samples can be $o(n^{3/2})$ provided sufficiently good features, while the required samples will be $O(n^{3/2})$ if no features are given.

Remark. Overall, we provide sample complexity analysis to justify that our model (4) is able to learn the missing information of L_0 more effectively by leveraging side information. The analysis is based on the generalization bounds of the missing values, where more informative side information (and less corruption) results in fewer required samples for accurate estimation, justifying the usefulness of side information.

Again, we emphasize that our results are relatively loose compared to those exact recovery guarantees in both matrix completion (Candès and Tao, 2009; Candès and Recht, 2012) and robust PCA (Chandrasekaran et al., 2011; Candès et al., 2011) as we only consider an approximate recovery on missing entries. However, it is important to note that those stronger recovery guarantees require additional assumptions, such as incoherence of the underlying low-rank matrices and distributional assumptions, to ensure the sampled observations are sufficiently representative. On the other hand, our analysis does not require distributional or incoherence assumptions, since in generalization analysis we only need to ensure the average loss of the missing entries are sufficiently small, and therefore, the average loss can still be controlled even if few spots are wrongly estimated in a high incoherence L_0.

However, in some circumstances, it is in fact possible to provide a stronger argument to justify the usefulness of side information in the exact recovery context. For example, in the robust PCA setting where observations are grossly corrupted yet full, one can further show that by exploiting perfect side information, a large amount of low-rank matrices L_0, which cannot be recovered by standard robust PCA without features, can be exactly recovered using our proposed model. Interested readers can consult Chiang et al. (2016) for such a result in detail. A theoretical analysis on how much the side information can improve the exact recovery guarantees in general low-rank matrix learning would be an interesting research direction to explore in the future.

4. Experimental Results

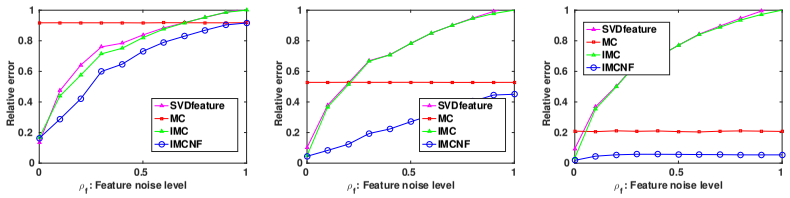
We now present experimental results on exploiting side information using the proposed model (4) for various low-rank matrix learning problems. For synthetic experiments, we show that our model performs better with the aid of side information given observations are either only missing (i.e. matrix completion setting), only corrupted (i.e. robust PCA setting) or both missing and corrupted. For real-world applications, we consider three machine learning applications—relationship prediction, semi-supervised clustering and noisy image classification—and show that each of them can be viewed as a problem of learning a low-rank modeling matrix from missing/corrupted entries with side information. As a result, by applying our model, we can achieve better performance compared to other state-of-the-art methods in these applications.

4.1. Synthetic Experiments

To begin with, we show the usefulness of (both perfect and noisy) side information in our model under different synthetic settings.

4.1.1. Experiments on Matrix Completion Setting

We first examine the effect of side information in our model in the case of matrix completion. We create a low rank matrix $L_0 = UV^T$ where the true latent row/column space $U, V \in \mathbb{R}^{200\times 20}$, $U_{ij}, V_{ij} \sim \mathcal{N}(0,1)$. We then randomly sample ρ_{obs} of entries Ω_{obs} from L_0 to form the observed matrix $R = \mathcal{P}_{\Omega_{obs}}(L_0)$. In addition, we construct perfect side information $X^*, Y^* \in \mathbb{R}^{200\times 40}$ satisfying (2), from which we generate different quality of features X, Y with a noise parameter $\rho_f \in [0,1]$, where X and Y are derived by replacing ρ_f of bases



(a) $\rho_{obs} = 0.1$ (b) $\rho_{obs} = 0.25$ (c) $\rho_{obs} = 0.4$

(d) $\rho_f = 0.1$ (e) $\rho_f = 0.5$ (f) $\rho_f = 0.9$

Figure 1: Performance of various methods for matrix completion under certain fixed sparsity of observations ρ_{obs} (upper figures) and fixed feature quality ρ_f (lower figures). We observe that all feature-based methods perform better than standard matrix completion (MC) given perfect features ($\rho_f = 0$). However, IMCNF is less sensitive to feature noise as ρ_f increases, indicating that it better exploits information from noisy features.

in X^* (and Y^*) with bases orthogonal to X^* (and Y^*). We then consider recovering the underlying matrix L_0 given R, X and Y.

In this experiment, we consider the proposed IMCNF model (problem 6) which is an instance of the general problem (4) for exploiting noisy side information in matrix completion case. We compare IMCNF with standard trace-norm regularized matrix completion (MC), IMC (Jain and Dhillon, 2013) and SVDfeature (Chen et al., 2012). The recovered matrix L^* from each algorithm is evaluated by the standard relative error:

$$\frac{\|L^* - L_0\|_F}{\|L_0\|_F}. \tag{11}$$

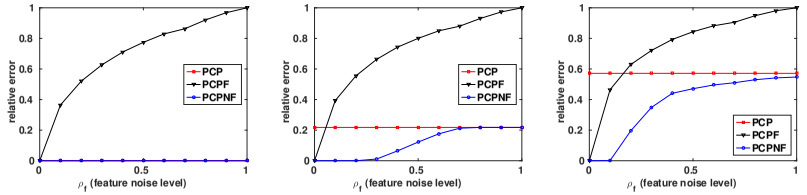
For each method, we select parameters from the set $\{10^\alpha\}_{\alpha=-3}^{2}$ and report the one with the best recovery. All results are averaged over 5 random trials.

Figure 1 shows results of each method under different $\rho_{obs} = 0.1, 0.25, 0.4$ and $\rho_f = 0.1, 0.5, 0.9$. We can first observe in upper figures that IMC and SVDfeature perform similarly under each ρ_{obs}, and moreover, their performance mainly depends on feature quality and will not be affected much by the number of observations. Although their performance is comparable to IMCNF given perfect features ($\rho_f = 0$), their performance quickly drops when features become noisy. This phenomenon is more clear in figure 1c and 1f where we see that given noisy features, IMC and SVDfeature will be easily trapped by feature noise and perform even worse than pure MC. Another interesting finding is that even if feature quality is as good as $\rho_f = 0.1$ (Figure 1d), IMC (and SVDfeature) still fails to achieve 0 relative error as the number of observations increases, suggesting that IMC is sensitive to feature noise and cannot guarantee recoverability when features are not perfect. On the other hand, we see that performance of IMCNF can be improved by both better features and more observations. In particular, it makes use of informative features to achieve lower error compared to MC and is also less sensitive to feature noise compared to IMC and SVDfeature. These results empirically support the analysis presented in Section 3.

4.1.2. Experiments on Robust PCA Setting

In this experiment, we examine the effect of both perfect and noisy side information in the proposed model for robust PCA as follows. We create a low-rank matrix $L_0 = UV^T$, where $U, V \in \mathbb{R}^{n\times 40}$, $U_{ij}, V_{ij} \sim N(0, 1/n)$ with $n = 200$. We also form a sparse noise matrix S_0 where each entry will be a non-zero entry with probability ρ_s, and each non-zero entry will take values from $\{\pm 1\}$ with equal probability. We then construct noisy features $X, Y \in \mathbb{R}^{n\times 50}$ with a noise parameter ρ_f using the same construction in the previous experiment, i.e. features X/Y will only span $40 \times (1 - \rho_f)$ true bases of U/V. We then consider to recover the low-rank matrix given the fully observed matrix $R = L_0 + S_0$ along with noisy side information X and Y.

We consider the following three methods: PCP (Candès et al., 2011) which does not exploit features, PCPF (problem 8) which theoretically exploits perfect features using bilinear embedding, and PCPNF (problem 7) for incorporating noisy side information. Note that PCPF and PCPNF are instances of our proposed model (4) that exploits side information for robust PCA problem as discussed in Section 2.3. The same relative error criterion (11) is used for evaluation.



(a) $\rho_s = 0.1$ (b) $\rho_s = 0.2$ (c) $\rho_s = 0.3$

Figure 2: Performance of various methods for robust PCA given different feature noise level ρ_f and sparsity of corruption ρ_s. These results show that PCPNF can make use of noisy yet informative features for better recovery.

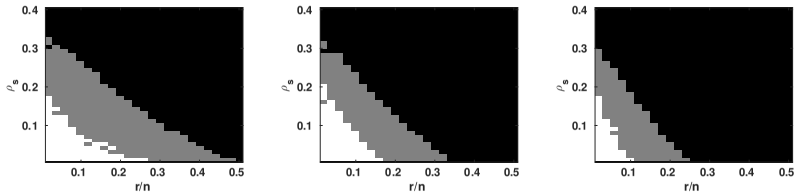
Figure 2 shows the performance of each method given different feature quality under $\rho_s = 0.1, 0.2, 0.3$. We first see that when features are perfect ($\rho_f = 0$), both PCPF and PCPNF can exactly recover the underlying matrix, while pure PCP fails to recover L_0 if $\rho_s \geq 0.2$. This result confirms that both PCPNF and PCPF can leverage perfect features for better recovery. However, as features become noisy (larger ρ_f), we see that PCPF quickly performs worse as it is misled by noise in features, while PCPNF can better exploit noisy features for recovery. In particular, in Figure 2b, we observe that PCPNF still recovers L_0 given noisy yet reasonably good features ($0 < \rho_f < 0.4$), whereas PCP and PCPF fail to recover L_0. These results show that PCPNF can take advantage of noisy side information for learning L_0 given corrupted observations.

4.1.3. Experiments on Learning with Missing and Corrupted Observations

We now further examine to what extent can side information help the learning using our model when observations are both missing and corrupted. We consider the same construction of L_0 and S_0 as in the previous experiment, and generate perfect feature matrices $X, Y \in \mathbb{R}^{n \times d}$ with $d = r + 10$. We then form the observation set Ω_{obs} by randomly sampling ρ_{obs} of entries from all n^2 indexes, and take $R = \mathcal{P}_{\Omega_{obs}}(L_0 + S_0)$ as the observed matrix. The goal is therefore to recover L_0 given R along with side information X and Y.

To exploit the advantage of side information, we consider the proposed model in form (5) where we further set $\alpha = 1$ and $\beta = \infty$ to force N^* to be zero for better exploiting perfect features, and compare it with the problem (1) which tries to recover L_0 only using structural information. Notice that when $\rho_{obs} = 1.0$, the given problem becomes a robust PCA problem where R is a fully observed matrix, in which case problem (1) reduces to PCP method and our model reduces to PCPF objective (problem 8), respectively. From this aspect, we refer to problem (1) as "PCP with partial observations" (PCP-part) and our model as "PCPF with partial observations" (PCPF-part). The relative error criterion (11) is again used to evaluate the recovered matrix. Here, we regard the recovery to be successful if the error is less than 10^{-4}. The parameter λ in both methods are set to be $1/\sqrt{\rho_{obs} n}$.

We compare the recoverability of PCP-part and PCPF-part by varying rank of L_0 (r) and sparsity of S_0 (ρ_s) under different $\rho_{obs} = 1.0, 0.7$ and 0.5. For each pair of (r, ρ_s),



(a) $\rho_{obs} = 1.0$ (b) $\rho_{obs} = 0.7$ (c) $\rho_{obs} = 0.5$

Figure 3: Performance of PCP-part and PCPF-part with perfect features for recovering L_0 from missing and corrupted observations (controlled by ρ_{obs} and ρ_s respectively). Both methods achieve recovery in white region and fail in black region, yet there is a gray region where only PCPF-part achieves recovery. This shows that by leveraging perfect features, PCPF-part can recover a much larger class of L_0 given both missing and corrupted observations are present.

we apply both methods to obtain the estimated low-rank matrix L^*. We then mark the grid point (r, ρ_s) to be white if recovery is attained by both methods and black if both fail. We also observe that in several cases recovery cannot be attained by PCP-part but can be attained by PCPF-part, and these grid points are marked as gray. The results are shown in Figure 3. We observe that for each ρ_{obs}, there exists a substantial gray region where matrices in such a region can be recovered only by PCPF-part. This result shows that in the case where both missing and corrupted entries are present, by exploiting side information, the proposed model is able to further recover a large amount of matrices which cannot be recovered if no side information is provided.

4.2. Real-world Applications

We now consider three applications—relationship prediction in signed networks, semi-supervised clustering and noisy image classification—which can be cast to problems of low-rank matrix learning from missing/corrupted entries with additional side information. As a consequence, we show that by learning the low-rank modeling matrix using our proposed model, we can achieve better performance compared to other methods for these applications as our model can better exploit side information in learning.

4.2.1. Relationship Prediction in Signed Networks

We first consider relationship prediction problem in an online review website Epinions (Massa and Avesani, 2006), where people can write product reviews and choose to trust or distrust others based on their reviews. Such a social network can be modeled as a *signed network* where each person is treated as an entity and trust/distrust relationships between people are modeled as positive/negative edges between entities (Leskovec et al., 2010). The relationship prediction problem in signed network is to predict unknown relationship between any two users given the current network snapshot. While several methods

Method	IMCNF	IMC	MF-ALS	HOC-3	HOC-5
Accuracy	**0.9474**±0.0009	0.9139±0.0016	0.9412±0.0011	0.9242±0.0010	0.9297±0.0011
AUC	**0.9506**	0.9109	0.9020	0.9432	0.9480

Table 2: Relationship prediction on Epinions network. We see that given noisy user features, IMC performs worse even than methods without features (MF-ALS and HOCs), while IMCNF outperforms others by successfully exploiting noisy features.

are proposed, a state-of-the-art approach is the low-rank model (Hsieh et al., 2012; Chiang et al., 2014) which first conducts matrix completion on the adjacency matrix and then uses the sign of the completed matrix for prediction. However, these methods are developed based only on network structure. Therefore, if features of users are available, we can also extend the low-rank model by incorporating user features in the completion step.

The experiment setup is described as follows. In this data set, there are about $n = 105$K users and $m = 807$K observed relationship pairs where 15% relationships are distrust. In addition to the who-trust-to-whom information, we are also given a user feature matrix $Z \in \mathbb{R}^{n\times 41}$ where for each user a 41-dimensional feature is collected based on the user's review history, such as number of positive/negative reviews the user gave or received. We consider the following prediction methods: walk and cycle-based methods including HOC-3 and HOC-5 (Chiang et al., 2014), and the original low-rank model with matrix factorization for the completion step (LR-ALS) (Hsieh et al., 2012). These methods make the prediction based on network structure without considering user features. We further consider the extended low-rank model where the completion step is replaced by IMCNF and IMC (Jain et al., 2013), both of which thus incorporate user features implicitly for prediction. Since row and column entities are both users, $X = Y = Z$ is set for both IMCNF and IMC methods. We randomly divide the edges of the network into 10 folds and conduct the experiment using 10-fold cross validation, in which 8 folds are used for training, one fold for validation and the other for testing. Parameters for validation in each method are chosen from the set $\sqcup_{\alpha=-3}^{2}\{10^\alpha, 5\times 10^\alpha\}$.

The averaged accuracy and AUC of each method are reported in Table 2. We first observe that IMC performs worse than LR-ALS even though IMC takes features into account. It is because these user features are only partially related to the relationship matrix, and IMC is misled by such noisy features. On the other hand, IMCNF performs the best among all prediction methods, as it performs slightly better than LR-ALS in terms of accuracy and much better in terms of AUC. This result shows that IMCNF can exploit weakly informative features to make better prediction without being trapped by feature noise.

4.2.2. Semi-supervised Clustering

Semi-supervised clustering is another application which can be translated to learning a low-rank matrix with partial observations. Given a feature matrix $Z \in \mathbb{R}^{n\times d}$ of n items and m pairwise constraints specifying whether item i and j are similar or dissimilar, the goal is to find a clustering of items such that most similar items are within the same cluster.

First, note that the problem can be sub-optimally solved by dropping either constraint or feature information. For example, traditional clustering algorithms (such as k-means) can solve the problem based purely on features of items. On the other hand, one can also

obtain a clustering purely from the pairwise constraints using matrix completion as follows. Let $S \in \mathbb{R}^{n\times n}$ be the (signed) similarity matrix constructed from the constraint set where $S_{ij} = 1$ if item i and j are similar, -1 if dissimilar and 0 if similarity is unknown. Then finding a clustering of n items becomes equivalent to finding a clustering on the signed graph S, where the goal is to put items (denoted as nodes) into k groups so that most edges within the same group are positive and most edges between groups are negative (Chiang et al., 2014). As a result, one can apply a matrix completion approach proposed in Chiang et al. (2014) to solve the signed graph clustering problem, which first conducts matrix completion on S and runs k-means on the top-k eigenvectors of completed S to obtain a clustering of nodes.

Apparently, either dropping features or constraint set is not optimal for semi-supervised clustering problem. Thus, many algorithms are proposed to take both item features and constraints into account, such as metric-learning-based approaches (Davis et al., 2007), spectral kernel learning (Li and Liu, 2009) and MCCC algorithm (Yi et al., 2013). Among many of them, MCCC algorithm is a cutting edge approach which essentially solves semi-supervised clustering using IMC objective. Observing that each pairwise constraint can be viewed as a sampled entry from the matrix $L_0 = UU^T$ where $U \in \mathbb{R}^{n\times k}$ is the clustering membership matrix, MCCC tries to complete L_0 back as ZMZ^T using IMC objective. Furthermore, since the completed matrix is ideally L_0 whose subspace spans U, it thus conducts k-means on the top-k eigenvectors of the completed matrix to obtain a clustering.

However, since MCCC is based on IMC, its performance thus heavily depends on the quality of features. Therefore, we propose to replace IMC with IMCNF in the matrix completion step of MCCC, and then run k-means on the top-k eigenvectors of the completed matrix to obtain a clustering. Both X and Y are again set to be Z as the target low-rank matrix describes the similarity between items. This algorithm can be viewed as an improved version of MCCC to handle noisy features Z.

We now compare our algorithm with k-means, signed graph clustering with matrix completion (Chiang et al., 2014) (SignMC) and MCCC (Yi et al., 2013) on three real-world data sets: Mushrooms, Segment and Covtype. [5] All of them are classification benchmarks where features and ground-truth labels of items are both available, and the ground-truth cluster of each item is defined by its ground-truth label. The statistics of data sets are summarized in Table 3. For each data set, we randomly sample $m = [1, 5, 10, 15, 20, 25, 30]\times n$ clean pairwise constraints and input both constraints and features to each algorithm to obtain a clustering π, where π_i is the cluster index of item i. We then evaluate π using the following pairwise error:

$$\frac{n(n-1)}{2}\left(\sum_{(i,j):\pi_i^*=\pi_j^*} \mathbf{1}\left[\pi_i \neq \pi_j\right] + \sum_{(i,j):\pi_i^*\neq\pi_j^*} \mathbf{1}\left[\pi_i = \pi_j\right]\right)$$

where π_i^* is the ground-truth cluster of item i.

Figure 4 shows the clustering result of each method given various numbers of constraints on each data set. We first see that for the Mushrooms data set where features are perfect (100% training accuracy can be attained by a linear-SVM for classification), both MCCC

5. All data sets are available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`. For Covtype, we subsample from the entire data set to make each cluster has balanced size.

	number of items n	feature dimension d	number of clusters k
Mushrooms	8124	112	2
Segment	2319	19	7
Covtype	11455	54	7

Table 3: Statistics of semi-supervised clustering data sets.

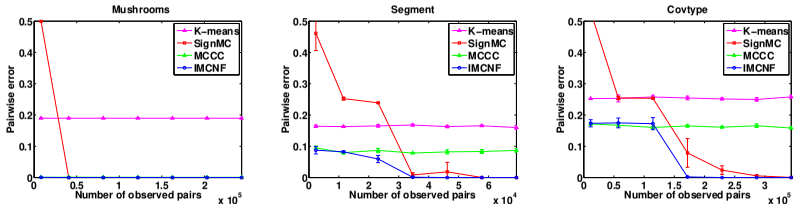


Figure 4: Performance of various semi-supervised clustering methods on real-world data sets. For the Mushrooms data set where features are perfect, both MCCC and IMCNF can output the ground-truth clustering with 0 error rate. For Segment and Covtype where features are more noisy, IMCNF model outperforms MCCC as its error decreases given more constraints.

and IMCNF can obtain the ground-truth clustering with 0 error rate, which indicates that MCCC (and IMC) is indeed effective with perfect features. For the Segment and Covtype data sets, we observe that the performance of k-means and MCCC is dominated by feature quality. Although MCCC is still benefited from constraint information as it outperforms k-means, it clearly does not make the best use of constraints since its performance is not improved even if number of constraints increases. On the other hand, the error rate of SignMC can always decrease down to 0 by increasing m; however, since it disregards features, it suffers from a much higher error rate than other methods when constraints are few. Finally, we see that IMCNF combines advantages from both MCCC and SignMC, as it not only makes use of features when few constraints are observed, but also leverages constraint information to avoid being trapped by feature noise. Therefore, the experiment shows that by carefully handling side information using IMCNF model, we can further improve the state-of-the-art semi-supervised clustering algorithm.

4.2.3. Noisy Image Classification

Finally, we consider noisy image classification problem as an application of low-rank matrix learning with corrupted observations. In this problem, we are given a set of correlated images in which a few of pixels are corrupted, and the task is to denoise the images so that one can classify the images correctly. Since the underlying clean images are correlated and thus share an implicit low-rank structure, standard robust PCA could be used to identify sparse noise and recover the (low-rank approximation of) clean images. However, in certain cases, low-dimensional features of images may also be available from other sources. For example, suppose the set of images are human faces, then the principal components of

	linear SVM classifiers					kernel SVM classifiers			
ρ_s	Clean	Noisy	PCP	PCPF	ρ_s	Clean	Noisy	PCP	PCPF
0.1		59.63	86.33	**87.88**	0.1		18.47	94.85	**95.89**
0.2	91.96	38.16	85.94	**87.48**	0.2	98.33	10.32	94.55	**95.48**
0.3		25.63	78.52	**79.84**	0.3		10.32	87.00	**87.78**

Table 4: Digit classification accuracy of PCP and PCPF with Eigendigit features. The column Clean shows the accuracy on L_0 and the column Noisy shows the accuracy on R. Denoised images from both PCP and PCPF achieve much higher accuracy than noisy images, and PCPF further outperforms PCP by incorporating Eigendigit features.

general human faces—known as Eigenface (Turk and Pentland, 1991)—could be used as features, and such features could be helpful in the denoising process.

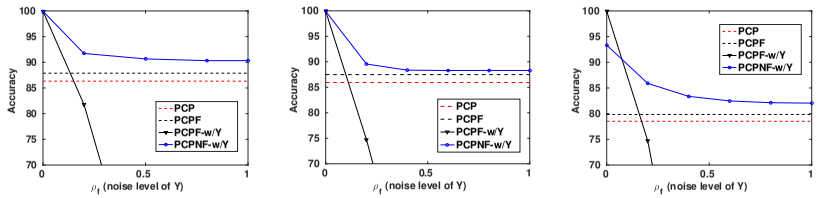
Motivated by the above realization, here we consider multiclass classification on a set of noisy images from the MNIST data set. The data set includes 50,000 training images and 10,000 testing images, and each image is a 28×28 handwritten digit represented as a 784-dimensional vector. We first pre-train both multiclass linear and kernel SVM classifiers on the clean training images, and perturb the testing image set to generate noisy images R. Precisely, let $L_0 \in \mathbb{R}^{784\times 10000}$ be the set of (clean) testing images, where each row denotes a pixel and each column denotes an image. We then construct a sparse noise matrix $S_0 \in \mathbb{R}^{784\times 10000}$ where ρ_s of entries are randomly picked to be corrupted by setting their values to be 255. The observed noisy images is thus given by $R = \min(L_0 + S_0, 255)$. In the following, we show that by exploiting features of row and column entities in this problem, we can better denoise the noisy images for classification.

Exploiting Eigendigit Features. We first exploit "Eigendigit" features to help denoising. We take the training image set to produce the Eigendigit features $X \in \mathbb{R}^{784\times 300}$ using PCA and simply set $Y = I$ as we do not consider any column features here. We then input R into PCP to derive a set of denoised images L^*_{pcp} and input R, X and Y (which is I) into PCPF (problem 8) to derive another set of denoised images $L^*_{pcpf} = XM^*$. Both L^*_{pcp} and L^*_{pcpf} will be low rank approximations of the clean images. Note that although the Eigendigit features X will not satisfy (2) which is assumed in the derivation of PCPF, we could heuristically incorporate it using PCPF in this circumstance because X is still expected to contain unbiased information of the low-rank approximation of the clean digits. [6]

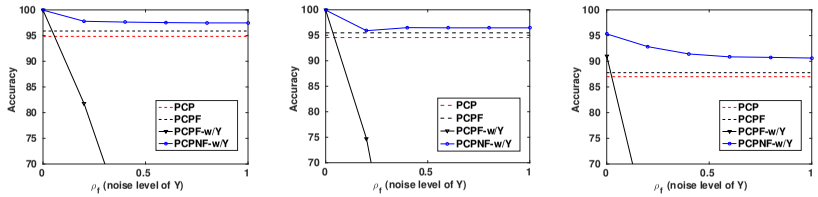
To compare the quality of denoised images of PCP and PCPF, we input L^*_{pcp} and L^*_{pcpf} to pre-trained SVMs for digit classification and report the results in Table 4. Both methods are somehow effective for denoising sparse noise, since accuracies achieved by the denoised images are much closer to the clean images compared to the noisy images. Furthermore, PCPF consistently achieves better accuracies than PCP under different ρ_s, showing that incorporating Eigendigit features using PCPF is helpful on denoising process for classification.

Exploiting both Eigendigit and Label-relevant Features In addition to the Eigendigit features X, now we further exploit features for column entities. Ideally, the

6. Rigorously speaking, the ground-truth L_0 is not even low-rank, but only approximately low-rank.



(a) Linear SVM, $\rho_s = 0.1$ (b) Linear SVM, $\rho_s = 0.2$ (c) Linear SVM, $\rho_s = 0.3$



(d) Kernel SVM, $\rho_s = 0.1$ (e) Kernel SVM, $\rho_s = 0.2$ (f) Kernel SVM, $\rho_s = 0.3$

Figure 5: Digit classification accuracy of various methods with both Eigendigit and label-relevant features. For each ρ_s, we construct the label-relevant features Y with different quality by varying ρ_f. The results show that PCPNF-w/Y is able to better exploit noisy label-relevant features Y.

column features Y may describe the relevant information between images, which could be extremely useful for classification. Thus, we generate the "label-relevant" features Y for column entities as follows. Let $Y^* \in \mathbb{R}^{10000\times 10}$ be a perfect column feature matrix where the i-th column of Y^* is the indicator vector of digit $i-1$ (so Y^* contains ground-truth label information). We then randomly pick ρ_f of rows in Y^* and shuffle these rows to form $\tilde{Y}$, which correspondingly means that $10,000 \times \rho_f$ images have noisy relevant information in $\tilde{Y}$. Finally, we form the column feature $Y \in \mathbb{R}^{10000\times 50}$ which spans $\tilde{Y}$. Thus, the quality of Y depends on the parameter $\rho_f \in [0,1]$ and smaller ρ_f results in better label-relevant features.

We consider four approaches for denoising in the following experiment. The first two baseline methods are PCP and PCPF with only Eigendigit features X. Both methods are the ones we considered in the previous experiment which do not take label-relevant features into account. Moreover, we consider using PCPF and PCPNF to incorporate *both* the Eigendigit features X and the label-relevant features Y for denoising, and we name them as "PCPF-w/Y" and "PCPNF-w/Y" to emphasize that they embed the label-relevant features Y. We apply each method to denoise noisy images under different ρ_f and ρ_s and examine the quality of denoised images by testing the accuracies they achieve in pre-trained SVMs.

The results are shown in Figure 5. In each figure, we fix the sparsity of noise ρ_s and try to recover the clean images using each method with different quality of Y. We can see that the perfect label-relevant features are extremely useful, as when $\rho_f = 0$, recovered images

from both PCPF-w/Y and PCPNF-w/Y achieve even much higher accuracies compared to the clean images (reported in Table 4). However, once ρ_f increases, PCPF-w/Y quickly fails as its accuracy drops drastically (accuracies become much lower than 70 for $\rho_f > 0.5$ and thus are not shown in figures). On the other hand, we see that PCPNF-w/Y performs much better than PCPF-w/Y on exploiting noisy label-relevant features, as it still achieves better accuracies compared to both PCPF and PCP when $\rho_f > 0$. The results again demonstrate the effectiveness of our proposed model on exploiting noisy side information.

5. Related Work

Learning a low-rank matrix from imperfect observations is an expansive domain in machine learning including many fundamental problems, such as Principal Component Analysis (PCA) (Hotelling, 1933), matrix completion (Candès and Tao, 2009), low-rank matrix sensing (Zhong et al., 2015) and robust PCA (Wright et al., 2009). While each of the above topics is an independent research area burgeoning in recent years, our main focus is to study the usefulness of side information in low-rank matrix learning where the observations are partial and/or corrupted in both theoretical and practical aspects.

Learning a low-rank matrix from partial observations is well-known as matrix completion problem, which has been successfully applied to many machine learning tasks including recommender systems (Koren et al., 2009), social network analysis (Hsieh et al., 2012; Chiang et al., 2014) and clustering (Chen et al., 2014). Several theoretical foundations have also been established. One of the most striking results is the exact recovery guarantee provided by Candès and Tao (2009) and Candès and Recht (2012) where the authors showed that $O(n \operatorname{polylog} n)$ observations are sufficient for exact recovery with high probability, provided that entries are uniformly sampled at random. Several works also study recovery under non-uniform distributional assumptions (Negahban and Wainwright, 2012), distribution-free settings (Shamir and Shalev-Shwartz, 2014) and noisy observations (Keshavan et al., 2010; Candès and Plan, 2010).

A few research papers also consider side information in the matrix completion setting (Menon and Elkan, 2011; Chen et al., 2012; Natarajan and Dhillon, 2014; Shin et al., 2015). Although most of them found that features are helpful in certain applications (Menon and Elkan, 2011; Shin et al., 2015) and in the cold-start setting (Natarajan and Dhillon, 2014), they mainly focus on the non-convex matrix factorization formulation without any theoretical analysis on the effect of side information. More recently, Jain et al. (2013) studied Inductive Matrix Completion (IMC) objective to incorporate side information, and several follow-up works also consider IMC with trace norm regularization (Xu et al., 2013; Zhong et al., 2015). All of them showed that recovery can be achieved by IMC with much lower sample complexity provided perfect features. However, as we have discussed in the paper, given imperfect features, IMC cannot recover the underlying matrix and may even suffer from poor performance in practice. This observation leads us to further develop an improved model which better exploits noisy side information in learning (see Section 2.3).

Robust PCA is another prominent instance of low-rank matrix learning from imperfect observations, where the goal is to recover a low-rank matrix from a full matrix in which a few of entries are arbitrarily corrupted by sparse noise. This sparse structure of noise is common in many applications such as image processing and bioinformatics (Wright et al.,

2009). Researchers have also investigated several approaches to robust PCA with theoretical guarantees (Chandrasekaran et al., 2011; Candès et al., 2011). Perhaps the most remarkable milestone is the strong guarantee provided by Candès et al. (2011), in which the authors showed that under mild conditions, low-rank and sparse structure are exactly distinguishable. Several extensions of robust PCA have also been considered, such as robust PCA with column-sparse errors (Xu et al., 2010), with missing data (Candès et al., 2011; Chen et al., 2013) and with compressed data (Ha and Barber, 2015).

However, unlike matrix completion, there is little research that directly exploits side information in the robust PCA problem, leaving the advantage of side information in robust PCA unexplored. Though it may appear that one can extend the analysis of side information in matrix completion to robust PCA as both problems share certain similarities, the robust PCA problem is still essentially different—in fact harder—from matrix completion in many aspects. In particular, matrix completion has been mostly used for *missing value estimation*, where the emphasis is to accurately recover the missing entries given trustable, partial observations, while robust PCA is a *matrix separation problem* where one has to identify the corrupted entries given full yet untrustable observations. This difference naturally precludes a direct extension from the analyses of matrix completion to robust PCA. Nevertheless, Chiang et al. (2016) has recently shown that given perfect features, exact recovery of higher-rank matrices becomes attainable in the robust PCA problem, indicating that side information in robust PCA can be exploited. In this paper, we extend Chiang et al. (2016) and develop a more general model which can further exploit noisy side information to help solve the robust PCA problem.

Another model that shares certain similarities to robust PCA with side information is Low-Rank Representation (LRR), which emerged from the subspace clustering problem (Liu et al., 2010, 2013). Given that the observed data matrix is corrupted by sparse errors, LRR model assumes that the underlying low-rank matrix can be represented by a linear combination of a provided dictionary. Interestingly, LRR can be thought of as a special case of the proposed PCPF model (see Section 2.3) where the given dictionary serves as the row features X. Our problem setting is also more general than LRR as we consider incorporating both row and column features to help recovery.

6. Conclusions

In this paper, we study the effectiveness of side information for low-rank matrix learning from missing and corrupted observations. We propose a general model (problem (4)) which incorporates both perfect and noisy side information by balancing information from features and observations simultaneously, from which we can derive several instances of the model, including IMCNF and PCPNF, that better solve matrix completion and robust PCA by leveraging side information. In addition, we provide a formal analysis to justify the effectiveness of side information in the proposed model, in which we quantify the quality of features and show that the sample complexity of learning can be asymptotically improved given sufficiently informative features, provided a small enough noise level. This analysis therefore quantifies the merits of side information in our model for low-rank matrix learning in theory. Finally, we verify our model in several synthetic experiments as well as in real-world machine learning applications including relationship prediction, semi-supervised

clustering and noisy image classification. By viewing each application as a low-rank matrix learning problem from missing or corrupted observations given certain additional features, we show that employing our model results in competitive algorithms whose performance is comparable to or better than other state-of-the-art approaches. All of our results consistently demonstrate that the proposed model learns the low-rank matrix from missing and corrupted observations more effectively by properly exploiting side information.

Acknowledgments

We would like to acknowledge support for this research from CCF-1320746, IIS-1546452 and CCF-1564000.

Appendix A. Proofs

Preliminary Lemmas

We first introduce two lemmas required in the proof of Lemma 3. These two lemmas provide bounds on the Rademacher complexity of the function class with bounded trace norm and $\ell 1$ norm respectively.

Lemma 9 *Let* $S_w = \{W \in \mathbb{R}^{n\times n} \mid \|W\|_* \leq \mathcal{W}\}$ *and* $\mathcal{A} = \max_i \|A_i\|_2$, *where each* $A_i \in \mathbb{R}^{n\times n}$, *then:*

$$\mathbb{E}_\sigma\big[\sup_{W\in S_w} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \text{trace}(WA_i)\big] \leq 2\mathcal{A}\mathcal{W}\sqrt{\frac{\log 2n}{m}}.$$

Proof This Lemma is directly from the Lemma 3 in Hsieh et al. (2015). ■

Lemma 10 *Let* $S_w = \{W \in \mathbb{R}^{n_1\times n_2} \mid \|W\|_1 \leq \mathcal{W}\}$, *and each* E_i *is in the form of* $E_i = \mathbf{e}_x\mathbf{e}_y^T$, *where* $\mathbf{e}_x \in \mathbb{R}^{n_2}$, $\mathbf{e}_y \in \mathbb{R}^{n_1}$ *are two unit vectors. Then:*

$$\mathbb{E}_\sigma\big[\sup_{W\in S_w} \frac{1}{m}\sum_{i=1}^{m} \sigma_i \text{trace}(WE_i)\big] \leq \mathcal{W}\sqrt{\frac{2\log(2n_1n_2)}{m}}.$$

Proof This Lemma is a special case of Theorem 1 in Kakade et al. (2008) with the fact that $\|E_i\|_\infty := \max_{a,b} |(E_i)_{ab}| = 1$. ■

Proof of Lemma 3

Proof First, we can use a standard Rademacher contraction principle (e.g. Lemma 5 in Meir and Zhang, 2003) to bound $\mathfrak{R}(F_\Theta)$ to be:

$$\begin{aligned}\mathfrak{R}(F_\Theta) \leq & L_\ell \mathbb{E}_\sigma\big[\sup_{\theta\in\Theta}\frac{1}{m}\sum_{\sigma=1}^{m}\sigma_\alpha(XMY^T+N+S)_{i_\alpha j_\alpha}\big]\\ =& L_\ell\mathbb{E}_\sigma\big[\sup_{\|M\|_*\leq\mathcal{M}}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha \mathrm{trace}(M\mathbf{y}_{j_\alpha}\mathbf{x}_{i_\alpha}^T)\big]+L_\ell\mathbb{E}_\sigma\big[\sup_{\|N\|_*\leq\mathcal{N}}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha\mathrm{trace}(N\mathbf{e}_{j_\alpha}\mathbf{e}_{i_\alpha}^T)\big]\\ &+L_\ell\mathbb{E}_\sigma\big[\sup_{\|S\|_1\leq\mathcal{S}}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha\mathrm{trace}(S\mathbf{e}_{j_\alpha}\mathbf{e}_{i_\alpha}^T)\big]\\ \leq & 2L_\ell\mathcal{M}\max_{i,j}\|\mathbf{y}_j\mathbf{x}_i^T\|_2\sqrt{\frac{\log 2d}{m}}+2L_\ell\mathcal{N}\sqrt{\frac{\log 2n}{m}}+L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}}\end{aligned}$$

where the last inequality is derived by applying Lemma 9 and Lemma 10. Moreover, since $\max_{i,j}\|\mathbf{y}_j\mathbf{x}_i^T\|_2=\max_j\|\mathbf{y}_j\|_2\max_i\|\mathbf{x}_i\|_2$, we can thus upper bound $\mathfrak{R}(F_\Theta)$ by:

$$\mathfrak{R}(F_\Theta)\leq 2L_\ell\mathcal{M}\mathcal{X}\mathcal{Y}\sqrt{\frac{\log 2d}{m}}+2L_\ell\mathcal{N}\sqrt{\frac{\log 2n}{m}}+L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}}. \tag{12}$$

However, in some circumstances, the above bound (12) is too loose for sample complexity analysis. To deal with these cases, we follow Shamir and Shalev-Shwartz (2014) to derive a tighter bound on the trace norm of residual (i.e. $\mathcal{N}$). To begin with, we rewrite $\mathfrak{R}(F_\Theta)$ as:

$$\mathfrak{R}(F_\Theta)=\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha\ell(f(i_\alpha,j_\alpha),R_{i_\alpha,j_\alpha}))\big]=\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\frac{1}{m}\sum_{(i,j)}\Gamma_{ij}\ell(f(i,j),R_{ij})\big],$$

where $\Gamma\in\mathbb{R}^{n_1\times n_2}$, $\Gamma_{ij}=\sum_{\alpha:i_\alpha=i,j_\alpha=j}\sigma_\alpha$. Now, using the same trick in Shamir and Shalev-Shwartz (2014), we can divide Γ based on the "hit-time" of each $(i,j)\in\Omega_{obs}$, with some threshold $p>0$ whose value will be set later. Formally, let $h_{ij}=|\{\alpha:i_\alpha=i,j_\alpha=j\}|$, and let $A,B\in\mathbb{R}^{n_1\times n_2}$ be defined by:

$$A_{ij}=\begin{cases}\Gamma_{ij}, & \text{if } h_{ij}>p,\\ 0, & \text{otherwise.}\end{cases}\qquad B_{ij}=\begin{cases}0, & \text{if } h_{ij}>p,\\ \Gamma_{ij}, & \text{otherwise.}\end{cases}$$

By construction, $\Gamma=A+B$. Therefore, we can separate $\mathfrak{R}(F_\Theta)$ to be:

$$\mathfrak{R}(F_\Theta)=\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\frac{1}{m}\sum_{(i,j)}A_{ij}\ell(f(i,j),R_{ij})\big]+\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\frac{1}{m}\sum_{(i,j)}B_{ij}\ell(f(i,j),R_{ij})\big]. \tag{13}$$

For the first term in (13), since $|\ell(f(i,j),R_{ij})|\leq\mathcal{B}$, it can be upper bounded by:

$$\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\frac{1}{m}\sum_{(i,j)}A_{ij}\ell(f(i,j),R_{ij})\big]\leq\frac{\mathcal{B}}{m}\mathbb{E}_\sigma\big[\sum_{(i,j)}|A_{ij}|\big]\leq\frac{\mathcal{B}}{\sqrt{p}}$$

where the last inequality is derived by applying Lemma 10 in Shamir and Shalev-Shwartz (2014). Now consider the second term of (13). Again, by Rademacher contraction principle, it can be upper bounded by:

$$\begin{aligned}&\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{f\in F_\Theta}\sum_{(i,j)}B_{ij}f(i,j)\big]\\ =&\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{\|M\|_*\leq\mathcal{M}}\sum_{(i,j)}B_{ij}\mathbf{x}_i^TM\mathbf{y}_j\big]+\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{\|N\|_*\leq\mathcal{N}}\sum_{(i,j)}B_{ij}N_{ij}\big]+\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{\|S\|_1\leq\mathcal{S}}\sum_{(i,j)}B_{ij}S_{ij}\big].\end{aligned} \tag{14}$$

We can again upper bound the first and third term of (14) using Lemma 9 and Lemma 10. Precisely, the first term can be upper bounded by:

$$\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{\|M\|_*\leq\mathcal{M}}\sum_{\alpha=1}^{m}\sigma_\alpha\mathbf{x}_{i_\alpha}^TM\mathbf{y}_{j_\alpha}\big]=L_\ell\mathbb{E}_\sigma\big[\sup_{\|M\|_*\leq\mathcal{M}}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha\mathrm{trace}(M\mathbf{y}_{j_\alpha}\mathbf{x}_{i_\alpha}^T)\big]\leq 2L_\ell\mathcal{M}\mathcal{X}\mathcal{Y}\sqrt{\frac{\log 2d}{m}},$$

and the third term of (14) is upper bounded by:

$$L_\ell\mathbb{E}_\sigma\big[\sup_{\|S\|_1\leq\mathcal{S}}\frac{1}{m}\sum_{\alpha=1}^{m}\sigma_\alpha\mathrm{trace}(S\mathbf{e}_{j_\alpha}\mathbf{e}_{i_\alpha}^T)\big]\leq L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}}.$$

In addition, by applying Hölder's inequality, the second term of (14) is upper bounded by:

$$\frac{L_\ell}{m}\mathbb{E}_\sigma\big[\sup_{\|N\|_*\leq\mathcal{N}}\sum_{(i,j)}B_{ij}N_{ij}\big]\leq\frac{L_\ell}{m}\sup_{N:\|N\|_*\leq\mathcal{N}}\|B\|_2\|N\|_*=\frac{L_\ell\mathcal{N}}{m}\mathbb{E}_\sigma\big[\|B\|_2\big]\leq\frac{2.2CL_\ell\mathcal{N}\sqrt{p}(\sqrt{n_1}+\sqrt{n_2})}{m},$$

where the last inequality is derived by applying Lemma 11 in Shamir and Shalev-Shwartz (2014). Therefore, putting all of the above upper bounds to (13) and choosing p to be $m\mathcal{B}/(2.2CL_\ell\mathcal{N}(\sqrt{n_1}+\sqrt{n_2}))$, we obtain another upper bound on $\mathfrak{R}(F_\Theta)$ as:

$$\mathfrak{R}(F_\Theta)\leq 2L_\ell\mathcal{M}\mathcal{X}\mathcal{Y}\sqrt{\frac{\log 2d}{m}}+\sqrt{\frac{9CL_\ell\mathcal{B}\mathcal{N}(\sqrt{n_1}+\sqrt{n_2})}{m}}+L_\ell\mathcal{S}\sqrt{\frac{2\log(2n_1n_2)}{m}}. \tag{15}$$

Lemma 3 thus follows by combining (12) and (15). ■

Proof of Lemma 4

Proof To begin with, we have:

$$\|X_\mu^TL_0Y_\nu\|_2\leq\|X_\mu\|_2\|L_0\|_2\|Y_\nu\|_2\leq\sigma_x\sigma_y\|L_0\|_*,$$

where σ_x (or σ_y) is the largest singular value of X_μ (or Y_ν). Therefore, by the definition of $\hat{M}$, we have:

$$\|\hat{M}\|_*\leq d\|\hat{M}\|_2=d\|(X_\mu^TX_\mu)^\dagger X_\mu^TL_0Y_\nu(Y_\nu^TY_\nu)^\dagger\|_2\leq\frac{\sigma_x\sigma_yd\|L_0\|_*}{\sigma_{xm}^2\sigma_{ym}^2}, \tag{16}$$

where σ_{xm} (or σ_{ym}) is the smallest non-zero singular value of X_μ (or Y_ν). Furthermore, by the construction of X_μ and Y_ν, we have $\sigma_{xm} \geq \mu\sigma_x$ and $\sigma_{ym} \geq \nu\sigma_y$. We can further lower bound σ_x (and σ_y) by:

$$\sigma_x^2 = \|X_\mu\|_2^2 = \|X\|_2^2 \geq \frac{\|X\|_F^2}{d} \geq \frac{n \min \|\mathbf{x}_i\|^2}{d} \geq \frac{n\gamma^2\mathcal{X}^2}{d}.$$

Therefore, from (16), we can further bound $\|\hat{M}\|_*$ by:

$$\|\hat{M}\|_* \leq \frac{d\|L_0\|_*}{\mu^2\nu^2\sigma_x\sigma_y} \leq \frac{d^2\|L_0\|_*}{\mu^2\nu^2\gamma^2\mathcal{X}\mathcal{Y}\sqrt{n_1 n_2}}.$$

The lemma is thus concluded by the fact that $\|L_0\|_* \leq C_L\sqrt{n_1 n_2}$. ■

Proof of Theorem 5

Proof The claim is directly proved by plugging Lemma 3 - 4 to Lemma 2, in which $\hat{R}_\ell(f^*) = 0$ because $(\hat{M}, L_0 - X\hat{M}Y^T, S_0) \in \Theta$ and such an instance makes $\hat{R}_\ell = 0$. ■

Proof of Theorem 6

Proof Note that since $S_0 = S^* = 0$ in matrix completion case, we have:

$$R_\ell(f^*) = \mathbb{E}_{(i,j)\sim\mathcal{D}}\big[\ell(XM^*Y_{ij}^T, \mathbf{e}_i^T L_0 \mathbf{e}_j)\big].$$

The claim therefore directly follows from Theorem 5 by setting $R_\ell(f^*) < \epsilon$. ■

Proof of Theorem 7

Proof By the construction of X and Y, we can rewrite them as follows:

$$X = \sum_{i=1}^{t-1} \mathbf{u}_i\mathbf{e}_i^T + \sum_{i=t}^{d} \tilde{\mathbf{u}}_i\mathbf{e}_i^T, \qquad Y = \sum_{i=1}^{t-1} \mathbf{v}_i\mathbf{e}_i^T + \sum_{i=t}^{d} \tilde{\mathbf{v}}_i\mathbf{e}_i^T, \tag{17}$$

where for each $\tilde{\mathbf{u}}_i$, $\tilde{\mathbf{u}}_i^T\mathbf{u}_j = 0, \forall j$. Therefore, we can upper bound $\mathcal{N}$ by:

$$\begin{aligned}\|L_0 - X\hat{M}Y^T\|_* &= \|\tilde{U}\tilde{U}^T L_0 + L_0\tilde{V}\tilde{V}^T - \tilde{U}\tilde{U}^T L_0\tilde{V}\tilde{V}^T\|_* \\ &\leq 2\|\tilde{U}\tilde{U}^T U\Sigma V^T\|_* + \|U\Sigma V^T\tilde{V}\tilde{V}^T\|_* \\ &\leq 3\sum_{i=t}^{k}\sigma_i,\end{aligned}$$

where $\tilde{U}$, $\tilde{V}$ are the second term of X and Y in (17). Moreover, we have $\sigma_i = o(\sqrt{n})$ for all $i \geq t$. To see this, suppose $\sigma_p = \Omega(\sqrt{n})$ for any $t \leq p \leq k$, then:

$$\lim_{n\to\infty}\frac{\sigma_t}{\sqrt{n}} \geq \lim_{n\to\infty}\frac{\sigma_p}{\sqrt{n}} > 0,$$

leading a contradiction to the definition of σ_t. Therefore we can conclude:

$$\mathcal{N} = \|L_0 - X\hat{M}Y^T\|_* \leq 3\sum_{i=t}^{k}\sigma_i \leq 3k \times o(\sqrt{n}) = o(\sqrt{n}),$$

and the Theorem is thus proved by plugging the above bound on $\mathcal{N}$ to Theorem 6. ■

Proof of Theorem 8

Proof The sample complexity claim directly follows from Theorem 5 by setting $R_\ell(f^*) < \epsilon$, and the claim of $\mathcal{P}_{\Omega_{obs}^\perp}(S^*) = 0$ is directly from the construction of Algorithm 1 as discussed in Section 2.4. ■

References

P. L. Bartlett and S. Mendelson. Rademacher and Gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2003.

J.-F. Cai, E. J. Candès, and Z. Shen. A singular value thresholding algorithm for matrix completion. *Society for Industrial and Applied Mathematics*, 20(4):1956–1982, 2010.

E. J. Candès and Y. Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6): 925–936, 2010.

E. J. Candès and B. Recht. Exact matrix completion via convex optimization. *Communications of the ACM*, 55(6):111–119, 2012.

E. J. Candès and T. Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transaction of Information Theory*, 56(5):2053–2080, 2009.

E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of ACM*, 58(3):11:1–11:37, 2011.

V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2), 2011.

T. Chen, W. Zhang, Q. Lu, K. Chen, Z. Zheng, and Y. Yu. SVDFeature: A toolkit for feature-based collaborative filtering. *Journal of Machine Learning Research*, 13:3619–3622, 2012.

Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transaction of Information Theory*, 59(7):4324–4337, 2013.

Y. Chen, A. Jalali, S. Sanghavi, and H. Xu. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15(1):2213–2238, 2014.

K.-Y. Chiang, C.-J. Hsieh, N. Natarajan, I. S. Dhillon, and A. Tewari. Prediction and clustering in signed networks: A local to global perspective. *Journal of Machine Learning Research*, 15:1177–1213, 2014.

K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon. Matrix completion with noisy side information. In *Advances in Neural Information Processing Systems*, 2015.

K.-Y. Chiang, C.-J. Hsieh, and I. S. Dhillon. Robust principal component analysis with side information. In *International Conference on Machine Learning*, 2016.

J. V. Davis, B. Kulis, P. Jain, S. Sra, and I. S. Dhillon. Information-theoretic metric learning. In *International Conference on Machine Learning*, pages 209–216, 2007.

M. Fazel, H. Hindi, and S. P. Boyd. A rank minimization heuristic with application to minimum order system approximation. In *American Control Conference*, volume 6, pages 4734–4739, 2001.

W. Ha and R. F. Barber. Robust PCA with compressed data. In *Advances in Neural Information Processing Systems*, 2015.

H. Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, 24(6):417–441, 1933.

C.-J. Hsieh and P. A. Olsan. Nuclear norm minimization via active subspace selection. In *International Conference on Machine Learning*, 2014.

C.-J. Hsieh, K.-Y. Chiang, and I. S. Dhillon. Low rank modeling of signed networks. In *International Conference on Knowledge Discovery and Data Mining*, pages 507–515, 2012.

C.-J. Hsieh, N. Natarajan, and I. S. Dhillon. PU learning for matrix completion. In *International Conference on Machine Learning*, 2015.

P. Jain and I. S. Dhillon. Provable inductive matrix completion. *CoRR*, abs/1306.0626, 2013.

P. Jain, P. Netrapalli, and S. Sanghavi. Low-rank matrix completion using alternating minimization. In *Symposium on Theory of Computing*, pages 665–674, 2013.

S. M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Advances in Neural Information Processing Systems*, pages 793 – 800, 2008.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from noisy entries. *Journal of Machine Learning Research*, 11:2057–2078, 2010.

Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42:30–37, 2009.

J. Leskovec, D. Huttenlocher, and J. Kleinberg. Predicting positive and negative links in online social networks. In *International Conference on World Wide Web*, pages 641–650, 2010.

Z. Li and J. Liu. Constrained clustering by spectral kernel learning. In *International Conference on Computer Vision*, 2009.

D. Liben-Nowell and J. Kleinberg. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.*, 58(7):1019–1031, 2007.

G. Liu, Z. Lin, and Y. Yu. Robust subspace segmentation by low-rank representation. In *International Conference on Machine Learning*, 2010.

G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma. Robust recovery of subspace structures by low-rank representation. *IEEE Transactions on Pattern Aanalysis and Machine Intelligence*, 35(1):171–184, 2013.

P. Massa and P. Avesani. Trust-aware bootstrapping of recommender systems. In *ECAI Workshop on Recommender Systems*, pages 29–33, 2006.

R. Meir and T. Zhang. Generalization error bounds for bayesian mixture algorithms. *Journal of Machine Learning Research*, 4:839–860, 2003.

A. K. Menon and C. Elkan. Link prediction via matrix factorization. *European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452, 2011.

N. Natarajan and I. S. Dhillon. Inductive matrix completion for predicting gene-disease associations. *Bioinformatics*, 30(12):60–68, 2014.

S. Negahban and M. J. Wainwright. Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *Journal of Machine Learning Research*, 13(1): 1665–1697, 2012.

B. Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12:3413–3430, 2011.

O. Shamir and S. Shalev-Shwartz. Matrix completion with the trace norm: Learning, bounding, and transducing. *Journal of Machine Learning Research*, 15(1):3401–3423, 2014.

D. Shin, S. Cetintas, K.-C. Lee, and I. S. Dhillon. Tumblr blog recommendation with boosted inductive matrix completion. In *International Conference on Information and Knowledge Management*, pages 203–212, 2015.

N. Srebro and A. Shraibman. Rank, trace-norm and max-norm. In *Annual Conference on Learning Theory*, pages 545–560, 2005.

Paul Tseng. Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of optimization theory and applications*, 109(3):475–494, 2001.

M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3 (1):71–86, 1991.

J. Wright, A. Ganesh, S. Rao, Y. Peng, and Y. Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in Neural Information Processing Systems*, 2009.

H. Xu, C. Caramanis, and S. Sanghavi. Robust PCA via outlier pursuit. In *Advances in Neural Information Processing Systems*, pages 2496–2504, 2010.

M. Xu, R. Jin, and Z.-H. Zhou. Speedup matrix completion with side information: Application to multi-label learning. In *Advances in Neural Information Processing Systems*, 2013.

J. Yi, L. Zhang, R. Jin, Q. Qian, and A. Jain. Semi-supervised clustering by input pattern assisted pairwise similarity matrix completion. In *International Conference on Machine Learning*, 2013.

K. Zhong, P. Jain, and I. S. Dhillon. Efficient matrix sensing using rank-1 Gaussian measurements. In *International Conference on Algorithmic Learning Theory*, 2015.

Journal of Machine Learning Research 19 (2018) 1-13 Submitted 3/17; Revised 12/17; Published 11/18

A Note on Quickly Sampling a Sparse Matrix with Low Rank Expectation

Karl Rohe KARLROHE@STAT.WISC.EDU
Statistics Department
University of Wisconsin-Madison
Madison, WI 53706, USA

Jun Tao ITAOJUN@PKU.EDU.CN
School of Mathematical Sciences
Peking University
Beijing, 100871, China

Xintian Han HANXINTIAN@PKU.EDU.CN
Yuanpei College
Peking University
Beijing, 100871, China

Norbert Binkiewicz NORBERTBIN@GMAIL.COM
Statistics Department
University of Wisconsin-Madison
Madison, WI 53706, USA

Editor: Inderjit Dhillon

Abstract

Given matrices $X, Y \in \mathbb{R}^{n\times K}$ and $S \in \mathbb{R}^{K\times K}$ with positive elements, this paper proposes an algorithm `fastRG` to sample a sparse matrix A with low rank expectation $E(A) = XSY^T$ and independent Poisson elements. This allows for quickly sampling from a broad class of stochastic blockmodel graphs (degree-corrected, mixed membership, overlapping) all of which are specific parameterizations of the generalized random product graph model defined in Section 2.2. The basic idea of `fastRG` is to first sample the number of edges m and then sample each edge. The key insight is that because of the the low rank expectation, it is easy to sample individual edges. The naive "element-wise" algorithm requires $O(n^2)$ operations to generate the $n \times n$ adjacency matrix A. In sparse graphs, where $m = O(n)$, ignoring log terms, `fastRG` runs in time $O(n)$. An implementation in `R` is available on github. A computational experiment in Section 2.4 simulates graphs up to $n = 10,000,000$ nodes with $m = 100,000,000$ edges. For example, on a graph with $n = 500,000$ and $m = 5,000,000$, `fastRG` runs in less than one second on a 3.5 GHz Intel i5.

Keywords: Random Dot Product Graph, Edge exchangeable, Simulation

1. Introduction

The random dot product graph (RDPG) model serves as a test bed for various types of clustering and statistical inference algorithms. This model generalizes the Stochastic Blockmodel (SBM), the Overlapping SBM, the Mixed Membership SBM, and the Degree-Corrected SBM (Holland et al., 1983; Latouche and Ambroise, 2011; Airoldi et al., 2008; Karrer and Newman, 2011). Under the RDPG, each node i has a (latent) node feature $x_i \in \mathbb{R}^K$ and the probability that node i and j share an edge is parameterized by $\langle x_i, x_j\rangle$ (Young and Scheinerman, 2007).

While many network analysis algorithms only require $O(m)$ operations, where m is the number of edges in the graph, sampling RDPGs with the naive "element-wise" algorithm takes $O(n^2)$ operations, where n is the number of nodes. In particular, sparse eigensolvers compute the leading k eigenvectors of such graphs in $O(km)$ operations (e.g. in ARPACK Lehoucq et al. 1995). As such, sampling an RDPG is a computational bottleneck in simulations to examine many network analysis algorithms.

The organization of this paper is as follows. Section 2 gives `fastRG`. Section 2.1 relates `fastRG` to xlr, a new class of edge-exchangeable random graphs with low-rank expectation. Section 2.2 presents a generalization of the random dot product graph. Theorem 4 shows that `fastRG` samples Poisson-edge graphs from this model. Then, Theorem 7 in Section 2.3 shows how `fastRG` can be used to approximate a certain class of Bernoulli-edge graphs. Section 2.4 describes our implementation of `fastRG` (available at `https://github.com/karlrohe/fastRG`) and assesses the empirical run time of the algorithm.

1.1. Notation

Let $G = (V, E)$ be a graph with the node set $V = \{1, \ldots, n\}$ and the edge set E contains edge (i, j) if node i is connected to node j. In a directed graph, each edge is an ordered pair of nodes while in an undirected graph, each edge is an unordered pair of nodes. A multi-edge graph allows for repeated edges. In any graph, a self-loop is an edge that connects a node to itself. Let the adjacency matrix $A \in \mathbb{R}^{n\times n}$ contain the number of edges from i to j in element A_{ij}. For column vectors $x \in \mathbb{R}^a$, $z \in \mathbb{R}^b$ and $S \in \mathbb{R}^{a\times b}$, define $\langle x, z\rangle_S = x^T S z$; this function is not necessarily a proper inner product because it does not require that S is non-negative definite. We use standard big-O and little-o notations, i.e. for sequence x_n, y_n; $x_n = o(y_n)$ when y_n is nonzero implies $\lim_{n\to\infty} x_n/y_n = 0$; $x_n = O(y_n)$ implies there exists a positive real number M and an integer N such that $|x_n| \le M|y_n|$, $\forall n \ge N$.

2. `fastRG`

`fastRG` is motivated by the wide variety of low rank graph models that specify the expectation of the adjacency matrix as $E(A) = XSX^T$ for some matrix (or vector) X and some matrix (or value) S.

Types of low rank models	$X \in$	In each row...
SBM	$\{0,1\}^{n\times K}$	a single one
Degree-Corrected SBM	$\mathbb{R}^{n\times K}$	a single non-zero positive entry
Mixed Membership SBM	$\mathbb{R}^{n\times K}$	non-negative and sum to one
Overlapping SBM	$\{0,1\}^{n\times K}$	a mix of 1s and 0s
Erdős-Rényi	$\{1\}^n$	a single one
Chung-Lu	$\mathbb{R}^n$	a single value

Table 1: Restrictions on the matrix X create different types of well known low rank models. There are further differences between these models that are not emphasized by this table.

Given $X \in \mathbb{R}^{n\times K_x}$, $Y \in \mathbb{R}^{d\times K_y}$ and $S \in \mathbb{R}^{K_x\times K_y}$, `fastRG` samples a random graph. Define A as the $n \times d$ matrix where A_{ij} counts the number of times edge (I,J) was sampled by `fastRG`. In expectation A is XSY^T. Importantly, `fastRG` requires that the elements of X,Y, and S are non-negative. This condition holds for all of the low rank models in the above table. Each of those models set $Y = X$ and enforce different restrictions on the matrix X.

As stated below, `fastRG` samples a (i) directed graph with (ii) multiple edges and (iii) self-loops. After sampling, these properties can be modified to create a simple graph (undirected, no repeated edges, and no self-loops); see Remarks 5 and 6 in Section 2.2 and Theorem 7 in Section 2.3.

Algorithm 1 `fastRG`(X,S,Y)

Require: $X \in \mathbb{R}^{n\times K_x}$, $S \in \mathbb{R}^{K_x\times K_y}$, and $Y \in \mathbb{R}^{d\times K_y}$ with all matrices containing non-negative entries.
  Compute diagonal matrix $C_X \in \mathbb{R}^{K_x\times K_x}$ with $C_X = diag(\sum_i X_{i1}\ ,\ \ldots\ ,\ \sum_i X_{iK_x})$.
  Compute diagonal matrix $C_Y \in \mathbb{R}^{K_y\times K_y}$ with $C_Y = diag(\sum_i Y_{i1}\ ,\ \ldots\ ,\ \sum_i Y_{iK_y})$.
  Define $\tilde{X} = XC_X^{-1}$, $\tilde{S} = C_X S C_Y$, and $\tilde{Y} = YC_Y^{-1}$.
  Sample the number of edges $m \sim Poisson(\sum_{u,v}\tilde{S}_{uv})$.
  for $\ell = 1 : m$ **do**
    Sample $U \in \{1,\ \ldots,\ K_x\}, V \in \{1,\ \ldots,\ K_y\}$ with $\mathbb{P}(U = u, V = v) \propto \tilde{S}_{uv}$.
    Sample $I \in \{1,\ \ldots,\ n\}$ with $\mathbb{P}(I = i) = \tilde{X}_{iU}$.
    Sample $J \in \{1,\ \ldots,\ d\}$ with $\mathbb{P}(J = j) = \tilde{Y}_{jV}$.
    Add edge (I,J) to the graph, allowing for multiple edges (I,J).
  end for

An implementation in `R` is available at `https://github.com/karlrohe/fastRG`. As discussed in Section 2.4, in order to make the algorithm more efficient, the implementation is slightly different from the statement of the algorithm above.

There are two model classes that can help to interpret the graphs generated from `fastRG` and those model classes are explored in the next two subsections. Throughout all of the discussion, the key fact that is exploited by `fastRG` is given in the next Theorem.

Theorem 1 *Suppose that $X \in \mathbb{R}^{n\times K_x}$, $Y \in \mathbb{R}^{d\times K_y}$ and $S \in \mathbb{R}^{K_x\times K_y}$ all contain non-negative entries. Define $x_i \in \mathbb{R}^{K_x}$ as the ith row of X. Define $y_j \in \mathbb{R}^{K_y}$ as the jth row of Y. Let (I,J) be a single edge sampled inside the for loop in `fastRG`(X,S,Y), then*

$$\mathbb{P}((I,J) = (i,j)) \propto \langle x_i, y_j\rangle_S.$$

Proof

$$\begin{aligned}\mathbb{P}\big((I,J) = (i,j)\big) &= \sum_{u,v}\mathbb{P}((I,J) = (i,j)|(U,V) = (u,v))\,\mathbb{P}((U,V) = (u,v)))\\ &= \frac{\sum_{u,v}\tilde{X}_{iu}\tilde{Y}_{jv}\tilde{S}_{uv}}{\sum_{u,v}\tilde{S}_{u,v}} = \frac{\sum_{u,v}X_{iu}Y_{jv}S_{uv}}{\sum_{u,v}\tilde{S}_{u,v}} = \frac{x_i^T S y_j}{\sum_{a,b}x_a^T S y_b}\end{aligned}$$

■

2.1. `fastRG` samples from xlr: a class of edge-exchangeable random graphs

There has been recent interest in edge exchangeable graph models with blockmodel structure (e.g. Crane and Dempsey 2016; Cai et al. 2016; Herlau et al. 2016; Todeschini and Caron 2016). To characterize a broad class of such models, we propose xlr. For notational simplicity, the rest of the paper will suppose that $Y = X \in \mathbb{R}^{n\times K}$ and `fastRG`(X,S) = `fastRG`(X,S,X).

Definition 2 *[xlr] An xlr graph on n nodes and K dimensions is generated as follows,*

1. *Sample $(X,S) \in \mathbb{R}^{n\times K}\times\mathbb{R}^{K\times K}$ from some distribution and define x_i as the ith row of X.*
2. *Initialize the graph to be empty.*
3. *Add independent edges $e_1, e_2, \ldots$ to the graph, where*

$$\mathbb{P}(e_\ell = (i,j)) = \frac{\langle x_i, x_j\rangle_S}{\sum_{a,b}\langle x_a, x_b\rangle_S}.$$

From Theorem 1, `fastRG` samples the edges in xlr.

An xlr graph is both (i) edge-exchangeable as defined by Crane and Dempsey (2016) and (ii) conditional on X and S, its expected adjacency matrix is low rank. By sampling X to satisfy one set of restrictions specified in Table 1, xlr provides a way to sample edge exchangeable blockmodels. xlr stands for *edge-exchangeable and low rank* because it characterizes all edge-exchangeable and low rank random graph models on a finite number of nodes. In particular, by Theorem 4.2 in Crane and Dempsey (2016) if a random undirected graph with an infinite number of edges is edge exchangeable, then the edges are drawn iid from some randomly chosen distribution on edges f. Moreover, let B be the adjacency matrix of a single edge drawn from f. Under the assumption that $E(B|f)$ is rank K, there exist matrices $X \in \mathbb{R}^{n\times K}$ and $S \in \mathbb{R}^{K\times K}$ that are a function of f and give the eigendecomposition $E(B|f) = XSX^T$. This implies that $\mathbb{P}(e_1 = (i,j)|f) \propto \langle x_i, x_j\rangle_S$, where x_i is the ith row of X.

2.2. `fastRG` samples from a generalization of the RDPG

Under the RDPG as described in Young and Scheinerman (2007), the expectation of the adjacency matrix is XX^T for some matrix $X \in \mathbb{R}^{n\times K}$. This implies that the expected adjacency matrix is always non-negative definite (i.e. its eigenvalues are non-negative). However, some parameterizations of the SBM (and other blockmodels) lead to an expected adjacency matrix with negative eigenvalues (i.e. it is not non-negative definite); for example, if the off-diagonal elements of S are larger than the diagonal elements, then XSX^T could have negative eigenvalues. Moreover, even if the elements of X and S are positive, as is the case for the low rank models in Table 1 and as is required for `fastRG`, it is still possible for XSX^T to have negative eigenvalues. By modifying the RDPG to incorporate a matrix S, the model class below incorporates all types of blockmodels.

Definition 3 *[Generalized Random Product Graph (gRPG) model] For n nodes in K dimensions, the gRPG is parameterized by $X \in \mathbb{R}^{n\times K}$ and $S \in \mathbb{R}^{K\times K}$, where each node i is assigned the ith row of X, $x_i = (X_{i1},\ \ldots,\ X_{iK})^T \in \mathbb{R}^K$. For $i,j \in V$, define*

$$\lambda_{ij} = \langle x_i, x_j\rangle_S = \sum_k^K\sum_l^K X_{ik}S_{kl}X_{jl}.$$

Under the gRPG, the adjacency matrix $A \in \mathbb{R}^{n\times n}$ contains independent elements and the distribution of A_{ij} (i.e. the number of edge from i to j) is fully parameterized by $f(\lambda_{ij})$, where f is some mean function.

Below, we will use the fact that the gRPG only requires that the λ_{ij}s specify the distribution of A_{ij}, allowing for A_{ij} to be non-binary (as in multi-graphs and weighted graphs) or to have edge probabilities which are a function of λ_{ij}.

Theorem 4 *For $X \in \mathbb{R}^{n\times K}$ and $S \in \mathbb{R}^{K\times K}$, each with non-negative elements, if $\tilde{A}$ is the adjacency matrix of a graph sampled with `fastRG`(X,S), then $\tilde{A}$ is a Poisson gRPG with $\tilde{A}_{ij} \sim Poisson(\langle x_i, x_j\rangle_S)$.*

The proof is contained in the appendix.

Remark 5 (Simulating an undirected graph) *As defined, both the gRPG model and `fastRG` generate directed graphs. An "undirected gRPG" should add a constraint to Definition 3 that $A_{ij} = A_{ji}$ for all i, j. To sample such a graph with `fastRG`, input $S/2$ instead of S, then after sampling a directed graph with `fastRG`, symmetrize each edge by removing its direction (this doubles the probability of an edge, hence the need to input $S/2$). Theorem 4 can be easily extended to show this is an undirected gRPG.*

Remark 6 (Simulating a graph without self-loops) *As defined, both the gRPG model and `fastRG` generate graphs with self-loops. A "gRPG without self-loops" should add a constraint to Definition 3 that $A_{ii} = 0$ for all i. A graph from `fastRG` can be converted to a gRPG without self-loops by simply (1) sampling $m \sim Poisson(\sum_{u,v}\tilde{S}_{uv} - \sum_i \langle x_i, x_i\rangle_S)$ and (2) resampling any edge that is a self-loop. The proof of Theorem 4 can be extended to show that this is equivalent.*

2.3. Approximate Bernoulli-edges

To create a simple graph with `fastRG` (i.e. no multiple edges, no self-loops, and undirected), first sample a graph with `fastRG`. Then, perform the modifications described in Remarks 5 and 6. Then, keep an edge between i and j if there is at least one edge in the multiple edge graph; define the threshold function, $t(A_{ij}) = \mathbb{1}(A_{ij} > 0)$, where $t(A)$ applies element-wise.

If $\tilde{A}$ is a Poisson gRPG, then $t(\tilde{A})$ is a Bernoulli gRPG with mean function $f(\lambda_{ij}) = 1 - exp(-\lambda_{ij})$. Let B be distributed as Bernoulli gRPG(X,S) with identity mean function,

$$B_{ij} \sim Bernoulli(\lambda_{ij}).$$

Theorem 7 shows that in the sparse setting, there is a coupling between $t(\tilde{A})$ and B such that $t(\tilde{A})$ is approximately equal to B. The theorem is asymptotic in n; a superscript of n is suppressed on $\tilde{A}, B$ and λ.

Theorem 7 *Let $\tilde{A}$ be a Poisson gRPG and let B be a Bernoulli gRPG using the same set of λ_{ij}s, with $\tilde{A}_{ij} \sim Poisson(\lambda_{ij})$ and $B_{ij} \sim Bernoulli(\lambda_{ij})$. Let $t(\cdot)$ be the thresholding function for $\tilde{A}$.*

Let α_n be a sequence. If $\lambda_{ij} = O(\alpha_n/n)$ for all i, j and there exists some constant $c > 0$ and $N > 0$ such that $\sum_{ij}\lambda_{ij} > c\alpha_n n$ for all $n > N$, then there exists a coupling between $t(\tilde{A})$ and B such that

$$\frac{E\|t(\tilde{A}) - B\|_F^2}{E\|B\|_F^2} = O(\alpha_n/n).$$

For example, in the sparse graph setting where $\lambda_{ij} = O(1/n)$ and $\sum_{ij}\lambda_{ij} = O(n)$, $\alpha_n = 1$. Under this setting and the coupling defined in the proof, all of the $O(n)$ edges in $t(\tilde{A})$ are contained in B and B has an extra $O(1)$ more edges than $t(\tilde{A})$.

The condition $\alpha_n = o(n)$ implies that all edge probabilities decay. If one is interested in models where some λ_{ij}'s are constant (e.g. certain models with heavy tailed degree distributions), then there are three possible paths forward.

1. Segment the pairs i, j into two sets (large and small λ_{ij}'s) and use two different sampling techniques on each set.

2. Use `fastRG` as a proposal distribution for rejection sampling (if for some $\varepsilon > 0$, $\lambda_{ij} < 1 - \varepsilon$ for all i, j, then rejection sampling would still be $O(m)$ operations).

3. Use `fastRG` and expect edge attenuation (no greater than 37%) for high probability edges.

Regarding the third point, consider the coupling in the proof of Theorem 7 without any condition on λ_{ij}. The coupling ensures that every edge in $t(\tilde{A})$ is also contained in B. Conversely, conditioned on edge i, j appearing in B, then the probability that this edge is included in $t(\tilde{A})$ is a greater than $1 - exp(-\lambda_{ij}) \geq 1 - exp(-1) > .63$.

2.4. Implementation of `fastRG`

Code at `https://github.com/karlrohe/fastRG` gives an implementation of `fastRG` in `R`. It also provides wrappers that simulate the SBM, Degree-Corrected SBM, Overlapping SBM, and Mixed Membership SBM. The code for these models first generates the appropriate X and then calls `fastRG`. In order to help control the edge density of the graph, `fastRG` and its wrappers can be given an additional argument avgDeg. If avgDeg is given, then the matrix S is scaled so that `fastRG` simulates a graph with expected average degree equal to avgDeg. Without this, parameterizations can easily produce very dense graphs.

To accelerate the running time of `fastRG`, the implementation is slightly different than the statement of the algorithm above. The difference can be thought of as sampling all of the (U,V) pairs before sampling any of the Is or Js. In particular, the implementation samples $\varpi \in R^{K\times K}$ as multinomial$(m, \tilde{S}/\sum_{u,v}\tilde{S}_{uv})$. Then, for each $u \in \{1,\dots K\}$, it samples $\sum_v \varpi_{u,v}$-many Is from the distribution $\tilde{X}_{u\cdot}$. Similarly, for each $v \in \{1,\dots K\}$, it samples $\sum_u \varpi_{u,v}$-many Js from the distribution $\tilde{X}_{v\cdot}$. Finally, the indexes are appropriately arranged so that there are $\varpi_{u,v}$-many edges (I,J) where $I \sim X_{u\cdot}$ and $J \sim X_{\cdot v}$. Recall that the statement of `fastRG` above allows for X and Y, where those matrices can have different numbers of rows and/or columns; the implementation also allows for this.

Under the SBM, it is possible to use `fastRG` to sample from the Bernoulli gRPG with the *identity* mean function instead of the mean function $1 - \exp(-\langle x_i, x_j\rangle_S)$ that is created by the thresholding function t from Section 2.3. The wrapper for the SBM does this by first transforming each element of S as $-\ln(1 - S_{ij})$ and then calling `fastRG`. The others models are not amenable to this trick; by default, they sample from the Poisson gRPG with identity mean function.

3. Experiments

3.1. Running time of `fastRG` on large and sparse graphs

To examine the running time of `fastRG`, we simulated a range of different values of n and $E(m)$, where $E(m)$ is the expected number of edges. In all simulations $X = Y$ and $K = 5$. The elements of X are independent $Poisson(1)$ random variables and the elements of S are independent $Uniform[0,1]$ random variables. To specify $E(m)$, the parameter avgDeg is set to $E(m)/n$. The values of n range from $10,000$ to $10,000,000$ and the values of $E(m)$ range from $100,000$ to $100,000,000$. The graph was taken to be directed, with self-loops and multiple edges. Moreover, the reported times are only to generate the edge list of the random graph; the edge list is not converted into a sparse adjacency matrix, which in some experiments would have more than doubled the running time. Each pair of n and $E(m)$ is simulated one time; deviations around the trend lines indicate the variability in run time.

In Figure 1, the vertical axes present the running time in `R` on a Retina 5K iMac, 27-inch, Late 2014 with 3.5 GHz Intel i5 and 8GB of 1600 MHz DDR3 memory. In the left panel of Figure 1, each line corresponds to a single value of n and $E(m)$ increases along the horizontal axis. In the right panel of Figure 1, each line corresponds to a single value of $E(m)$ and n increases along the horizontal axis. All axes are on the log_{10} scale. The solid black line has a slope of 1. Because the data aligns with this black line, this suggests that `fastRG` runs in linear time.

The computational bottleneck is sampling the Is and Js. The implementation uses Walker's Alias Method (Walker, 1977) (via `sample` in `R`). To take m samples from a distribution over n elements, Walker's Alias Method requires $O(m + \ln(n)n)$ operations (Vose, 1991). However, the log dependence is not clearly evident in the right plot of Figure 1; perhaps it would be visible for larger values of n.

3.2. Comparison to a previous technique

Previously, Hagberg and Lemons (2015) studied a fast technique to generate sparse random kernel graphs. Under the random kernel graph model, nodes i and j connect with probability $\kappa(i/n, j/n)$, where the function κ is non-negative, bounded, symmetric, measurable, and almost surely continuous. This model class includes the SBM and the Degree-Corrected SBM. It is difficult to see how a more general low rank model could be parameterized as a random kernel graph with an almost surely continuous κ. For example, we suspect that Mixed Membership SBMs could not be parameterized as such.

The algorithm proposed in Hagberg and Lemons (2015) is fast when it is fast to compute (i) the integral $F(y,a,b) = \int_a^b \kappa(x,y)dx$ and (ii) its roots, that is for any y,a,r, solve for $F(y,a,b) = r$. Their software, which we will refer to as *fast*-κ is in python and generates a NetworkX graph.

For a simple benchmark to compare the running times of `fastRG` and *fast*-κ, Figure 2 repeats the run time experiment that was performed in Hagberg and Lemons (2015). This simulation is for an Erdős-Rényi graph with expected degree 10, for n ranging between $5,000$ and $5M$. Speed comparisons are troubled by the fact that our code returns an edge list in `R` and *fast*-κ returns a NetworkX graph in python. Converting from an edge list to other data types takes longer than sampling the edge list with `fastRG`. For example, converting the edge list to a sparse matrix (a type that is convenient for spectral estimators) takes about as long as sampling the edge list with `fastRG`. Converting the edge list to an igraph takes about 10x longer than sampling the edge list

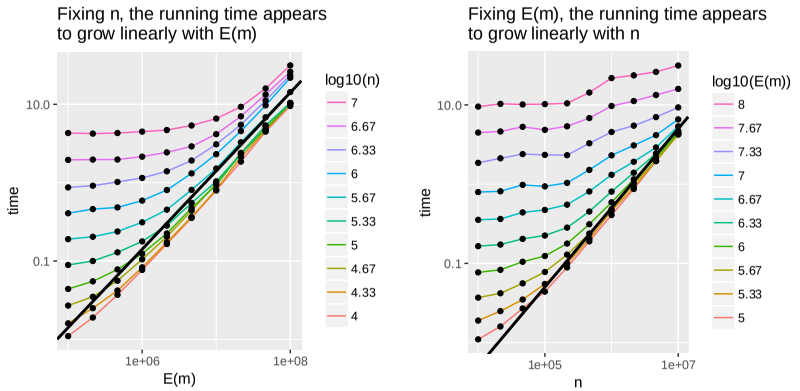


Figure 1: Both plots present the same experimental data. In the left plot, each line corresponds to a different value of n and they are presented as a function of $E(m)$. In the right plot, each line corresponds to a different value of $E(m)$ and they are presented as a function of n. On the right side of both plots, the lines start to align with the solid black line, suggesting a linear dependence on $E(m)$ and n.

with `fastRG`. There are three lines in Figure 2 for `fastRG`, one line for each data type (edge list, sparse adjacency matrix, and igraph). The speed comparison in Figure 2 corresponds to the average running time over 10 simulations performed on a 2015 MacBook Pro, 2.8 GHz Intel Core i7, with 16 GB 1600 MHz DDR3 running Python 3.5.2 and `R` 3.3.2. The slope of the blue line corresponds to the running time $O(n \log n)$. While none of these packages have been optimized for speed, they are all sufficiently fast for a wide range of purposes.

3.3. Simulating small and dense graphs with `fastRG`

This section investigates the graph density at which `fastRG` becomes slower than simulating each element A_{ij} as a Bernoulli random variable. In Figure 3, the reported time to compute this naive (element-wise) algorithm includes both (i) the time it takes to compute the probabilities `eA = X%*%B%*%t(X)` and (ii) sample the edges `z = rbinom(length(eA), 1, eA)`. The time for `fastRG` is for a directed graph, represented as a sparse matrix. The time to compute `fastRG` also includes the time it takes to construct X and B.

Figure 3 compares these two approaches on a set of Stochastic Blockmodels for values $K \in \{2,5,10\}$, $n \in \{500,1000,5000,10000\}$, and graph density $\rho = n^{-2}E(m)$ varying between .02 and .35. In all simulations, the S matrix is proportional to $I_K + J_K \in \mathbb{R}^{K \times K}$, where I_K is the identity matrix and J_K is the matrix of ones. The scale of S is adjusted to ensure the correct density ρ. For each model, both `fastRG` and the naive simulation are used to simulate two graphs. The line is a quadratic fit with ordinary least squares. In each panel, the vertical line is at $\rho = .25$, which is

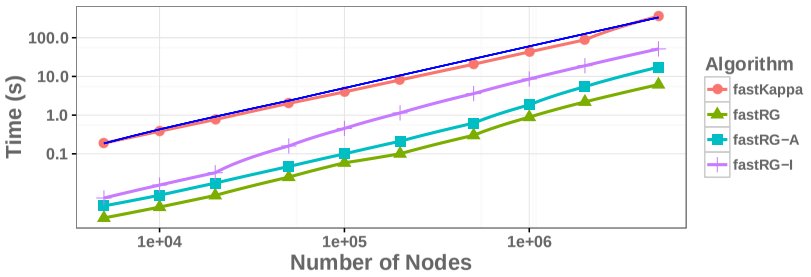


Figure 2: As the number of nodes increases (horizontal axis), all of the running times increase in parallel to the solid blue line which gives the rate $O(n \log n)$. The bottom three lines all correspond to `fastRG`, outputting three different graph types (edge list, sparse adjacency matrix, and igraph). For example, in roughly 8 seconds, *fast-*κ generates a graph with 20k nodes and `fastRG` generates an igraph with 1M nodes. To generate the random edge list on 1M nodes with `fastRG` takes less than 1 second

approximately the crossover point in all simulations. For scale, $\rho = .25$ corresponds to expected degrees 125, 250, 1250, and 2,500 respectively for n values 500, 1000, 5000, and 10000.

Acknowledgments

This research was supported by NSF grant DMS-1612456 and ARO grant W911NF-15-1-0423.

Appendix A. Proofs

For an integer d, define $1_d \in \mathbb{R}^d$ as a vector of ones. The proof of Theorem 4 requires the following lemma, which says that a vector (or matrix) of independent Poisson entries becomes multinomial when you condition on the sum of the vector (or matrix).

Lemma 8 *Let $A \in \mathbb{R}^{n \times n}$ be the random matrix whose i,jth element $A_{ij} \overset{i.d.}{\sim} Pois(\lambda_{ij})$ $i,j = 1, \ldots, n$. Then conditioned on $\sum_{i,j} A_{ij} = 1_n^T A 1_n = m$,*

$$(A_{11}, A_{12}, \ldots, A_{nn}) \sim Multinomial(m, \lambda / \sum_{ij} \lambda_{ij})$$

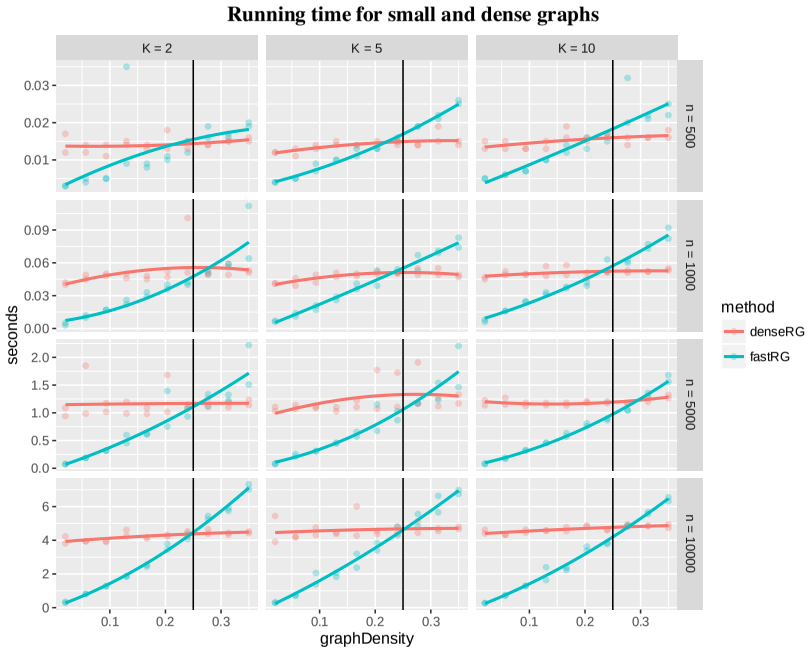


Figure 3: This figure compares the run time of `fastRG` to the run time of simulating each A_{ij} as a Bernoulli random variable (`denseRG` in the legend). Note that the number of edges grows quadratically with the edge density. So, as the density of the graph increases (horizontal axis), the running time of `fastRG` grows quadratically, whereas the running time of the naive algorithm does not depend on the density of the graph. In this simulation, the crossover is around $\rho = .25$ (black line). This figure shows that even for relatively small n and in the dense regime, `fastRG` has potential to be faster than the naive approach.

where $\lambda = (\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nn})$. *That is, let* $a \in \mathbb{R}^{n\times n}$ *be a fixed matrix of integers with* $1_n^T a 1_n = m$, *then*

$$\begin{aligned}\mathbb{P}(A = a | 1_n^T A 1_n = m) &= \mathbb{P}(A_{11} = a_{11}, A_{12} = a_{12}, \ldots, A_{nn} = a_{nn} | 1_n^T A 1_n = m) \\ &= \frac{m!}{\prod_{i,j} a_{ij}!} \prod_{i,j} \left(\frac{\lambda_{ij}}{\lambda_{11} + \lambda_{12} + \cdots + \lambda_{nn}} \right)^{a_{ij}}.\end{aligned}$$

For completeness, a proof of this classical result is given at the end of the paper. The next proof is a proof of Theorem 4.

Proof Let A come from the Poisson gRPG with X and S and identity mean function. Let $\tilde{A}$ be a sample from `fastRG`. For any fixed adjacency matrix a, we will show that $\mathbb{P}(A = a) = \mathbb{P}(\tilde{A} = a)$.

Define $m = 1_n^T a 1_n$ and decompose the probabilities,

$$\mathbb{P}(A = a) = \mathbb{P}(1_n^T A 1_n = m)\mathbb{P}(A = a | 1_n^T A 1_n = m) \quad (1)$$
$$\mathbb{P}(\tilde{A} = a) = \mathbb{P}(1_n^T \tilde{A} 1_n = m)\mathbb{P}(\tilde{A} = a | 1_n^T \tilde{A} 1_n = m). \quad (2)$$

The proof will be divided into two parts. The first part shows that $\mathbb{P}(1_n^T A 1_n = m) = \mathbb{P}(1_n^T \tilde{A} 1_n = m)$ and the second part will show that $\mathbb{P}(A = a | 1_n^T A 1_n = m) = \mathbb{P}(\tilde{A} = a | 1_n^T \tilde{A} 1_n = m)$.

Part 1: The sum of independent Poisson variables is still Poisson,

$$\sum_{ij} A_{ij} \sim Poisson(\sum_{ij} \lambda_{ij}).$$

So, we must only show that $1_n^T A 1_n$ and $1_n^T \tilde{A} 1_n$ have the same Poisson parameter:

$$\sum_{ij} \lambda_{ij} = 1_n^T X S X 1_n = 1_n^T X C C^{-1} S C^{-1} C X 1_n = 1_n^T \tilde{X} \tilde{S} \tilde{X} 1_n = 1_n^T \tilde{X} \tilde{S} \tilde{X} 1_n = 1_K^T \tilde{S} 1_K = \sum_{u,v} \tilde{S}_{uv}.$$

Part 2: After conditioning on $1_n^T A 1_n = m$, Lemma 8 shows that A has the multinomial distribution. In `fastRG`, we first sample $1_n^T \tilde{A} 1_n$ and then add edges with the multinomial distribution. So, we must only show that the multinomial edge probabilities are equal for A and $\tilde{A}$. From Lemma 8, the multinomial edge probabilities for A are $\lambda_{ij} / \sum_{a,b} \lambda_{ab}$. To compute the multinomial edge probabilities for $\tilde{A}$, recall that (I, J) is a single edge added to the graph in `fastRG`. By Theorem 1,

$$\mathbb{P}(\tilde{A}_{ij} = 1 | 1_n^T \tilde{A} 1_n = 1) = \mathbb{P}\big((I, J) = (i, j)\big) = \frac{\langle x_i, x_j \rangle_S}{\sum_{a,b} \langle x_a, x_b \rangle_S} = \frac{\lambda_{ij}}{\sum_{a,b} \lambda_{ab}}$$

This concludes the proof. ■

Proof [Proof of Theorem 7] Let $U_{ij} \overset{i.i.d}{\sim} Uniform(0,1)$. Define $\mathscr{A}$ and $\mathscr{B}$:

$$\mathscr{A}_{ij} = \mathbf{1}(U_{ij} > 1 - e^{-\lambda_{ij}}),$$
$$\mathscr{B}_{ij} = \mathbf{1}(U_{ij} > \lambda_{ij}).$$

Note that $\mathscr{A}$ and $\mathscr{B}$ are equal in distribution to $t(\tilde{A})$ and B respectively. By Taylor expansion,

$$E\|\mathscr{A} - \mathscr{B}\|_F^2 = \sum_{i,j} (\lambda_{ij} - (1 - e^{-\lambda_{ij}})) = \sum_{i,j} \sum_{k=2}^{\infty} (-\lambda_{ij})^k / k! = \sum_{i,j} O(\lambda_{ij}^2) = \sum_{i,j} O((\alpha_n / n)^2) = O(\alpha_n^2).$$

Then, $E\|B\|_F^2 = \sum_{ij} \lambda_{ij} > c\alpha_n n$. So, defining $t(\tilde{A})$ and B with the above coupling yields the result. ■

Proof [proof of Lemma 1]

$$\begin{aligned}\mathbb{P}(A = a | 1_n^T A 1_n = m) = \frac{\mathbb{P}(A = a)}{\mathbb{P}(1_n^T A 1_n = m)} &= \frac{\prod_{i,j} \frac{\lambda_{ij}^{a_{ij}}}{a_{ij}!} e^{-\lambda_{ij}}}{\frac{(\lambda_{11} + \lambda_{12} + \cdots + \lambda_{nn})^m}{m!} e^{-(\lambda_{11} + \lambda_{12} + \cdots + \lambda_{nn})}} \\ &= \frac{m!}{\prod_{i,j} a_{ij}!} \prod_{i,j} \left(\frac{\lambda_{ij}}{\lambda_{11} + \lambda_{12} + \cdots + \lambda_{nn}} \right)^{a_{ij}}\end{aligned}$$

■

References

E Airoldi, E Blei, S Fienberg, and E Xing. Mixed membership stochastic blockmodels. *Journal of Machine Learning Research*, 9(5):1981–2014, 2008.

D Cai, T Campbell, and T Broderick. Edge-exchangeable graphs and sparsity. In *Advances in Neural Information Processing Systems*, pages 4242–4250, 2016.

H Crane and W Dempsey. Edge exchangeable models for network data. *arXiv preprint arXiv:1603.04571*, 2016.

A Hagberg and N Lemons. Fast generation of sparse random kernel graphs. *PloS one*, 10(9): e0135177, 2015.

T Herlau, M Schmidt, and M Mø rup. Completely random measures for modelling block-structured sparse networks. In *Advances in Neural Information Processing Systems 29*. 2016.

P Holland, K Laskey, and S Leinhardt. Stochastic blockmodels: First steps. *Social Networks*, 5(2): 109–137, 1983.

B Karrer and M Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011.

P Latouche and C Ambroise. Overlapping stochastic block models with application to the french political blogosphere. *Annals of Applied Statistics*, 5(1):309–336, 2011.

R Lehoucq, D Sorensen, and P Vu. Arpack: An implementation of the implicitly re-started arnoldi iteration that computes some of the eigenvalues and eigenvectors of a large sparse matrix. *Available from netlib@ ornl. gov under the directory scalapack*, 1995.

A Todeschini and F Caron. Exchangeable Random Measures for Sparse and Modular Graphs with Overlapping Communities. *arXiv preprint arXiv:1602.02114*, February 2016.

M Vose. A linear algorithm for generating random numbers with a given distribution. *IEEE Transactions on software engineering*, 17(9):972–975, 1991.

A Walker. An efficient method for generating discrete random variables with general distributions. *ACM Transactions on Mathematical Software*, 3(3):253–256, 1977.

S Young and E Scheinerman. *Random Dot Product Graph Models for Social Networks*, pages 138–149. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

Journal of Machine Learning Research 19 (2018) 1-21 Submitted 6/17; Published 12/18

Online Bootstrap Confidence Intervals for the Stochastic Gradient Descent Estimator

Yixin Fang YIXIN.FANG@NJIT.EDU
Department of Mathematical Sciences
New Jersey Institute of Technology

Jinfeng Xu XUJF@HKU.HK
Department of Statistics and Actuarial Science
The University of Hong Kong

Lei Yang LY888@NYU.EDU
Department of Population Health
New York University School of Medicine

Editor: Gabor Lugosi

Abstract

In many applications involving large dataset or online learning, stochastic gradient descent (SGD) is a scalable algorithm to compute parameter estimates and has gained increasing popularity due to its numerical convenience and memory efficiency. While the asymptotic properties of SGD-based estimators have been well established, statistical inference such as interval estimation remains much unexplored. The classical bootstrap is not directly applicable if the data are not stored in memory. The plug-in method is not applicable when there is no explicit formula for the covariance matrix of the estimator. In this paper, we propose an online bootstrap procedure for the estimation of confidence intervals, which, upon the arrival of each observation, updates the SGD estimate as well as a number of randomly perturbed SGD estimates. The proposed method is easy to implement in practice. We establish its theoretical properties for a general class of models that includes linear regressions, generalized linear models, M-estimators and quantile regressions as special cases. The finite-sample performance and numerical utility is evaluated by simulation studies and real data applications.

Keywords: Bootstrap, Interval estimation, Generalized linear models, Large datasets, M-estimators, Quantile regression, Resampling methods, Stochastic gradient descent

1. Introduction

Big datasets arise frequently in clinical, epidemiological, financial and sociological studies. In such applications, many classical optimization methods for parameter estimation such as Fisher scoring, the EM algorithm or iterated reweighted least squares (Hastie et al. 2009, Nelder and Baker 1972) do not scale well and are computationally less attractive. Due to its computational and memory efficiency, stochastic gradient descent (Robbins and Monro 1951)[SGD] is a scalable algorithm for parameter estimation and has recently drawn a great deal of attention. Unlike other classical methods that evaluate the objective function involving the entire dataset, the SGD method calculates the gradient of the objective function using only one data point at a time and recursively updates the parameter estimate. This is also numerically appealing and particularly useful in online updating settings such as streaming data where it may not even be feasible to store the entire dataset in memory. Wang et al. (2016) gave a nice review on recent achievements of applying the SGD method to big data and streaming data.

The asymptotic properties of SGD estimators such as consistency and asymptotic normality have been well established; see, for example, Ruppert (1988) and Polyak and Juditsky (1992). However, statistical inference such as confidence interval estimation for SGD estimators has remained largely unexplored. Traditional interval estimation procedures such as the plug-in procedure and the bootstrap are often numerically difficult in the presence of big datasets. The bootstrap repeatedly draws samples from the entire dataset. The plug-in procedure requires an explicit variance-covariance formula. Since the classical bootstrap is not directly applicable if the data are not stored in memory, using the deal from the weighted bootstrap (Rubin 1981), we propose an online bootstrap procedure for the estimation of confidence intervals.

There are only a few papers considering the statistical inference of the SGD method. Chen et al. (2016) proposed a method called the batch-mean procedure. Although computationally efficient and theoretically sound, the batch-means procedure substantially underestimates the variance of the SGD estimator in finite-sample studies, because of the correlations between the batch means. Li et al. (2017) presented a new method for statistical inference in M-estimation problems, based on SGD estimators with a fixed step size. However, this method is limited to M-estimation and fixed step size. Su and Zhu (2018) proposed a new method called HiGrad, short for Hierarchical Incremental GRAdient Descent, which estimates model parameters in an online fashion and provides a confidence interval for the true population value. This method is also computationally efficient and theoretically sound, but it is not applicable to vanilla SGD estimators.

In this paper, we propose an online bootstrap resampling procedure to approximate the distribution of a SGD estimator in a general class of models that includes linear regressions, generalized linear models, M-estimators and quantile regressions as special cases. Our proposal, justified by asymptotic theories, provides a simple way to estimate the covariance matrix and confidence regions. Through numerical experiments, we verify the ability of this procedure to give accurate inference for big datasets.

The rest of the article is organized as follows. In Section 2, we introduce the proposed online bootstrap procedure for constructing confidence regions. In Section 3, we theoretically justify the validity of our proposal for a general class of models, along with some special cases. In Section 4, we demonstrate the performance of the proposed procedures in finite samples via simulation studies and three real data applications. Some concluding remarks are given in Section 5 and all the technical proofs are relegated to the Appendix.

2. The proposed resampling procedure

Parameter estimation by optimizing an objective function is often encountered in statistical practice. Consider the general situation where the optimal model parameter $\theta_0 \in \mathcal{R}^p$ is defined to be the minimizer of the expected loss function,

$$\theta_0 = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{ L(\theta) \triangleq \mathbb{E}[\ell(\theta; Z)] \right\}, \tag{1}$$

where $\ell(\theta; z)$ is some loss function and Z denotes one single observation and Θ is the domain on which the loss function is defined, which is assumed to be open. Suppose that the data consist of independent and identically distributed (i.i.d.) copies of Z, denoted by $\mathcal{D}_N = \{Z_1, \ldots, Z_N\}$. Under mild conditions, θ_0 can be consistently estimated by

$$\widetilde{\theta}_N = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{ \frac{1}{N} \sum_{i=1}^{N} \ell(\theta; Z_i) \right\}. \tag{2}$$

However, the minimization problem (2) for large-scale datasets pose numerical challenges. Furthermore, for applications such as online data where each sample arrives sequentially (e.g., search queries or transactional data), it may not be necessary or feasible to store the entire dataset, leaving alone evaluating the minimand in (2).

As a stochastic approximation method (Robbins and Monro 1951), stochastic gradient descent is a scalable algorithm for parameter estimation with large-scale data. Given an initial estimate $\widehat{\theta}_0$, the SGD method recursively updates the estimate upon the arrival of each data point Z_n, $n = 1, 2, \ldots, N$,

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \gamma_n \nabla \ell(\widehat{\theta}_{n-1}; Z_n), \tag{3}$$

where the learning rates are $\gamma_n = \gamma_1 n^{-\alpha}$ with $\gamma_1 > 0$ and $\alpha \in (0.5, 1)$. As suggested by Ruppert (1988) and Polyak and Juditsky (1992), we consider the averaging estimate,

$$\overline{\theta}_n = \frac{1}{n} \sum_{i=1}^{n} \widehat{\theta}_i, \tag{4}$$

which can also be recursively updated given that $\overline{\theta}_n = (n-1)\overline{\theta}_{n-1}/n + \widehat{\theta}_n/n$.

In order to conduct statistical inference with the averaging SGD estimator $\overline{\theta}_n$ at any stage, we propose an online bootstrap resampling procedure, which recursively updates the SGD estimate as well as a large number of randomly perturbed SGD estimates, upon the arrival of each data point. Specifically, let $\mathcal{W} = \{W_i, i = 1, \ldots, N\}$ be a set of i.i.d. non-negative random variables with mean and variance equal to one. In parallel with (3) and (4), with $\widehat{\theta}_0^* \equiv \widehat{\theta}_0$, upon observing data point Z_n, we recursively updates randomly perturbed SGD estimates,

$$\widehat{\theta}_n^* = \widehat{\theta}_{n-1}^* - \gamma_n W_n \nabla \ell(\widehat{\theta}_{n-1}^*; Z_n), \tag{5}$$

$$\overline{\theta}_n^* = \frac{1}{n} \sum_{i=1}^{n} \widehat{\theta}_i^*. \tag{6}$$

We will show that $\sqrt{n}(\overline{\theta}_n - \theta_0)$ and $\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n)$ converge in distribution to the same limiting distribution. In practice, these results allow us to estimate the distribution of $\sqrt{n}(\overline{\theta}_n - \theta_0)$ by generating a large number, say B, of random samples of $\mathcal{W}$. We obtain $\overline{\theta}_n^{*,b}$ by sequentially updating perturbed SGD estimates for each sample, $b = 1, \ldots, B$,

$$\widehat{\theta}_n^{*,b} = \widehat{\theta}_{n-1}^{*,b} - \gamma_n W_{n,b} \nabla \ell(\widehat{\theta}_{n-1}^{*,b}; Z_n), \tag{7}$$

$$\overline{\theta}_n^{*,b} = \frac{1}{n} \sum_{i=1}^{n} \widehat{\theta}_i^{*,b}, \tag{8}$$

and then approximate the sampling distribution of $\overline{\theta}_n - \theta_0$ using the empirical distribution of $\{\overline{\theta}_n^{*,b} - \overline{\theta}_n, b = 1, ..., B\}$. Specifically, the covariance matrix of $\overline{\theta}_n$ can be estimated by the sample covariance matrix constructed from $\{\overline{\theta}_n^{*,b}, b = 1, ..., B\}$. Estimating the distribution of $\sqrt{n}(\overline{\theta}_n - \theta_0)$ based on the distribution of $\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n)|\mathcal{D}_n$ leads to the construction of $(1-\alpha)100\%$ confidence regions for θ_0. The resulting inferential procedure retains the numerical simplicity of the SGD method, only using one pass over the data. The proposed inferential procedure scales well for datasets with millions of data points or more, and its theoretical validity can be justified for a general class models with mild regularity conditions as shown in the next section.

3. Theoretical Results

3.1. Main theorems

In this section, we derive some theoretical properties of $\overline{\theta}_n^*$, justifying that the conditional distribution of $\overline{\theta}_n^* - \overline{\theta}_n$ given data $\mathcal{D}_n = \{Z_1, Z_2, \ldots, Z_n\}$ can approximate the sampling distribution of $\overline{\theta}_n - \theta_0$, under the following assumptions. Let $\|\cdot\|$ be the Euclidean norm for vectors and the operator norm for matrices. The proofs are presented in the Appendix.

(A1). The objective function $L(\theta)$ is convex, continuously differentiable over $\theta \in \Theta$, and twice continuously differentiable at $\theta = \theta_0$, where θ_0 is the unique minimizer of $L(\theta)$.

(A2). The gradient of $L(\theta)$, $R(\theta) = \nabla L(\theta)$, is Lipschitz continuous with constant $L_1 > 0$; that is, for any θ_1 and θ_2, $\|R(\theta_1) - R(\theta_2)\| \le L_1 \|\theta_1 - \theta_2\|$.

(A3). The Hessian matrix of $L(\theta)$, $S(\theta) = \nabla^2 L(\theta)$, exists and is positive definite at θ_0 with $S_0 = S(\theta_0) > 0$ and is Lipschitz continuous at θ_0 with constant $L_2 > 0$.

(A4). Let $V_0 = \mathbb{E}\{[\nabla \ell(\theta_0; Z)][\nabla \ell(\theta_0; Z)]^{\mathrm{T}}\}$. Assume $\mathbb{E}\|\nabla \ell(\theta; Z)\|^2 \le C(1 + \|\theta\|^2)$ for some C and $\mathbb{E}\|\nabla \ell(\theta; Z) - \nabla \ell(\theta_0; Z)\|^2 \le \delta(\|\theta - \theta_0\|)$ for some $\delta(\cdot)$ with $\delta(x) \to 0$ as $x \to 0$.

(A5). The learning rates are chosen as $\gamma_n = \gamma_1 n^{-\alpha}$ with $\gamma_1 > 0$ and $\alpha \in (0.5, 1)$.

(A6). The perturbation variables, $W_1, W_2, \ldots$, are non-negative i.i.d. random variables satisfying that $\mathbb{E}(W_n) = \mathrm{Var}(W_n) = 1$.

Following similar arguments in Ruppert (1988) and Polyak and Juditsky (1992), we can prove the asymptotic normality of the SGD estimator $\overline{\theta}_n$ under the above assumptions.

Lemma 1 *If Assumptions A1-A5 are satisfied, then we have*

$$\sqrt{n}(\overline{\theta}_n - \theta_0) \Rightarrow \mathcal{N}\left(0, S_0^{-1} V_0 S_0^{-1}\right), \text{ in distribution, as } n \to \infty. \tag{9}$$

From Lemma 1, we can conduct statistical inference based on $\overline{\theta}_n$ provided that we can estimate the covariance matrix $S_0^{-1} V_0 S_0^{-1}$, or we can use some resampling procedure to approximate the sampling distribution of $\sqrt{n}(\overline{\theta}_n - \theta_0)$. We first derive the asymptotically linear representation of $\overline{\theta}_n^*$ for any perturbation variables that are i.i.d. random variables satisfying that $\mathbb{E}(W_n) = 1$.

Theorem 2 *If Assumptions A1-A5 hold, and the perturbation variables, $W_1, W_2, \ldots$, are non-negative i.i.d. random variables satisfying that $\mathbb{E}(W_n) = 1$, then we have,*

$$\sqrt{n}(\overline{\theta}_n^* - \theta_0) = -\frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^{n} W_i \nabla \ell(\theta_0; Z_i) + o_p(1). \tag{10}$$

By Theorem 1, letting $W_n \equiv 1$, we derive the following representation for $\overline{\theta}_n$,

$$\sqrt{n}(\overline{\theta}_n - \theta_0) = -\frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^{n} \nabla \ell(\theta_0; Z_i) + o_p(1). \tag{11}$$

Then, considering the difference between (2) and (11), we have

$$\sqrt{n}(\overline{\theta}_n^* - \theta_n) = -\frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^{n} (W_i - 1) \nabla \ell(\theta_0; Z_i) + o_p(1). \tag{12}$$

Let $\mathbb{P}^*$ and $\mathbb{E}^*$ denote the conditional probability and expectation given the data. Starting from (12), we derive the following theorem.

Theorem 3 *If Assumptions A1-A6 hold, then we have*

$$\sup_{v \in \mathcal{R}^p} \left| \mathbb{P}^* \left(\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n) \le v \right) - \mathbb{P}\Big(\sqrt{n}(\overline{\theta}_n - \theta_0) \le v \Big) \right| \to 0, \textit{in probability.} \tag{13}$$

By Theorem 2, the Kolmogorow-Smirnov distance between $\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n)$ and $\sqrt{n}(\overline{\theta}_n - \theta_0)$ converges to zero in probability. This validates our proposal of the perturbation-based resampling procedure for inference with SGD. In the next section, we consider some special cases where Assumptions A1-A4 are satisfied.

3.2. Special cases

3.2.1. Cases where $\ell(\theta; Z)$ is twice differentiable

If the loss function $\ell(\theta; Z)$ is twice differentiable, we can use the plug-in procedure to estimate the asymptotic covariance matrix of $\sqrt{N}(\overline{\theta}_N - \theta_0)$, $S_0^{-1} V_0 S_0^{-1}$. That is, at the final step, S_0 and V_0 can be estimated respectively by

$$\widehat{S}_N = \frac{1}{N} \sum_{i=1}^{N} \nabla^2 \ell(\overline{\theta}_N; Z_i) \ \text{ and } \ \widehat{V}_N = \frac{1}{N} \sum_{i=1}^{N} [\nabla \ell(\overline{\theta}_N; Z_i)][\nabla \ell(\overline{\theta}_N; Z_i)]^{\mathrm{T}}. \tag{14}$$

However, the above final-step plug-in estimation is impractical for large-scale data or streaming data, because it requires that the whole dataset be stored. To overcome this problem, in practice we can estimate S_0 and V_0 recursively, for $n = 1, 2, \ldots$, using

$$\widehat{S}_n = \frac{1}{n} \sum_{i=1}^{n} \nabla^2 \ell(\overline{\theta}_i; Z_i) \ \text{ and } \ \widehat{V}_n = \frac{1}{n} \sum_{i=1}^{n} [\nabla \ell(\overline{\theta}_i; Z_i)][\nabla \ell(\overline{\theta}_i; Z_i)]^{\mathrm{T}}. \tag{15}$$

In the following, we examine two examples where $\ell(\theta; Z)$ is twice differentiable. Example 1 is linear regression, where the loss function $\ell(\theta; Z)$ is twice differentiable and the objective

function $L(\theta)$ is strongly convex. Example 2 is logistic regression, where $\ell(\theta; Z)$ is twice differentiable but $L(\theta)$ is non-strongly convex. They are two examples of generalized linear models, one for quantitative outcome and the other for binary outcome. In these two examples, both the plug-in procedures and the proposed perturbation resampling procedure are robust to model mis-specification.

Example 1 (Linear regression) Suppose that $Z_n = (Y_n, X_n)$, $n = 1, 2, \ldots$, are i.i.d. copies of $Z = (Y, X)$, where Y is quantitative and X is p-dim with $\mathbb{E}\|X\|^2 < \infty$. Let

$$\theta_0 = \arg\min_{\theta \in \mathcal{R}^p} \mathbb{E}\left(Y - X^{\mathrm{T}}\theta\right)^2, \tag{16}$$

where $\ell(\theta; Z) = (Y - X^{\mathrm{T}}\theta)^2$ is twice differentiable and $L(\theta) = \mathbb{E}\left(Y - X^{\mathrm{T}}\theta\right)^2$ is strongly convex. Moveover, $\nabla \ell(\theta; Z) = -2(Y - X^{\mathrm{T}}\theta)X$, $\nabla^2 \ell(\theta; Z) = 2X^{\mathrm{T}}X$, $\nabla L(\theta) = 2\mathbb{E}\{XX^{\mathrm{T}}\}\theta - 2\mathbb{E}\{XY\}$, and $\nabla^2 L(\theta) = \mathbb{E}\{\nabla l(\theta; Z)\} = 2\mathbb{E}\{XX^{\mathrm{T}}\}$. Letting $V_0 = 4\mathbb{E}\{(Y - X^{\mathrm{T}}\theta_0)^2 XX^{\mathrm{T}}\}$ and $S_0 = 2\mathbb{E}\{XX^{\mathrm{T}}\}$, we can easily verify that Assumptions A1-A4 hold. The SGD and perturbed SGD updates for θ_0, as defined in (3) and (5) respectively, are

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + 2\gamma_n (Y_n - X_n^{\mathrm{T}} \widehat{\theta}_{n-1}) X_n, \tag{17}$$

$$\widehat{\theta}_n^* = \widehat{\theta}_{n-1}^* + 2\gamma_n W_n (Y_n - X_n^{\mathrm{T}} \widehat{\theta}_{n-1}^*) X_n. \tag{18}$$

Example 2 (Logistic regression) Suppose that $Z_n = (Y_n, X_n)$, $n = 1, 2, \ldots$, are i.i.d. copies of $Z = (Y, X)$, where $Y = \pm 1$ and X is p-dim with $\mathbb{E}\|X\|^2 < \infty$. Let

$$\theta_0 = \arg\min_{\theta \in \mathcal{R}^p} \mathbb{E}\left\{ -\log\left(\frac{1}{1 + \exp(-YX^{\mathrm{T}}\theta)} \right) \right\}, \tag{19}$$

where $\ell(\theta; Z) = \log\left(1 + \exp(-YX^{\mathrm{T}}\theta)\right)$ is twice differentiable and $L(\theta) = \mathbb{E}\{\ell(\theta; Z)\}$ is non-strongly convex. Moveover, $\nabla \ell(\theta; Z) = -\frac{1}{1+\exp(YX^{\mathrm{T}}\theta)} XY$, $\nabla^2 \ell(\theta; Z) = \frac{\exp(YX^{\mathrm{T}}\theta)}{[1+\exp(YX^{\mathrm{T}}\theta)]^2} XX^{\mathrm{T}}$, $\nabla L(\theta) = \mathbb{E}\left\{\nabla \ell(\theta; Z)\right\}$, and $\nabla^2 L(\theta) = \mathbb{E}\left\{\nabla^2 \ell(\theta; Z)\right\}$. Letting $V_0 = \mathbb{E}\left\{ \frac{1}{[1+\exp(YX^{\mathrm{T}}\theta)]^2} XX^{\mathrm{T}} \right\}$ and $S_0 = \mathbb{E}\left\{ \frac{\exp(YX^{\mathrm{T}}\theta)}{[1+\exp(YX^{\mathrm{T}}\theta)]^2} XX^{\mathrm{T}} \right\}$, we can easily verify that Assumptions A1-A4 hold. The SGD and perturbed SGD updates for θ_0, as defined in (3) and (5) respectively, are

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} + \gamma_n XY / [1 + \exp(YX^{\mathrm{T}} \widehat{\theta}_{n-1})], \tag{20}$$

$$\widehat{\theta}_n^* = \widehat{\theta}_{n-1}^* + \gamma_n W_n XY / [1 + \exp(YX^{\mathrm{T}} \widehat{\theta}_{n-1}^*)]. \tag{21}$$

We conclude this subsection with some discussion on the strong convexity of objective function $L(\theta)$, which is strongly convex in Example 1 and is non-strongly convex in Example 2. If $L(\theta)$ is strongly convex, i.e. there exists $\mu > 0$ such that $L(\theta_1) \ge L(\theta_2) + \nabla L(\theta_2)^{\mathrm{T}}(\theta_1 - \theta_2) + \mu\|\theta_1 - \theta_2\|^2$ for any θ_1 and θ_2, Moulines and Bach (2011) derived a non-asymptotic bound for $(\mathbb{E}\|\overline{\theta}_n - \theta_0\|)^{1/2}$. The bound of $(\mathbb{E}\|\overline{\theta}_n - \theta_0\|)^{1/2}$ has several terms; the leading term is of order $O(n^{-1})$ and the next two leading terms have order $O(n^{\alpha-2})$ and $O(n^{-2\alpha})$, suggesting the setting $\alpha = 2/3$ to make them equal. If $L(\theta)$ is non-strongly convex, Moulines and Bach (2011) derived a non-asymptotic bound for $\mathbb{E}[L(\widehat{\theta}_n) - L(\theta_0)]$ and a non-asymptotic bound for $\mathbb{E}[L(\overline{\theta}_n) - L(\theta_0)]$. The bound of $\mathbb{E}[L(\widehat{\theta}_n) - L(\theta_0)]$ is $O(\max\{n^{\alpha-1}, n^{-\alpha/2}\})$, also suggesting the setting $\alpha = 2/3$ to achieve optimal rate $O(n^{-1/3})$. Using the Polyak-Ruppert averaging has allowed the bound of $\mathbb{E}[L(\overline{\theta}_n) - L(\theta_0)]$ to go from $O(\max\{n^{\alpha-1}, n^{-\alpha/2}\})$ to $O(n^{-\alpha})$. Therefore, we use $\alpha = 2/3$ in the numerical results.

3.2.2. Cases where $\ell(\theta; Z)$ is not twice-differentiable

If the loss function $\ell(\theta; Z)$ is not twice-differentiable, neither the final-step plug-in estimation (14) nor the recursive plug-in estimation (15) is applicable. Fortunately, our proposal of scalable inference based on perturbation resampling is still applicable because it only depends on the first order derivative $\nabla\ell(\theta; Z)$. To understand this explicitly, consider the following example of robust regression via ψ-type M-estimator where the loss function may be not twice-differentiable.

Example 3 (Robust regression via ψ-type M-estimator) Suppose that $Z_n = (Y_n, X_n)$, $n = 1, 2, \ldots$, are i.i.d. copies of $Z = (Y, X)$, where Y is quantitative and X is p-dim with $\mathbb{E}\|X\|^2 < \infty$. Let $\rho(\cdot)$ be some convex function with $\rho(0) = 0$ and we attempt to estimate

$$\theta_0 = \arg\min_{\theta\in\mathcal{R}^p} \mathbb{E}\rho\left(Y - X^{\mathrm{T}}\theta\right), \tag{22}$$

where $\ell(\theta; Z) = \rho(Y - X^{\mathrm{T}}\theta)$ and $L(\theta) = \mathbb{E}\rho\left(Y - X^{\mathrm{T}}\theta\right)$. This is robust regression via ρ-type M-estimator. If $\rho(\cdot)$ is differentiable with derivative $\dot\rho(\cdot) = \psi(\cdot)$, we can solve it via ψ-type M-estimator, solving the following equation,

$$\mathbb{E}\left\{\psi\left(Y - X^{\mathrm{T}}\theta_0\right) X\right\} = 0, \tag{23}$$

where $\nabla\ell(\theta; Z) = -\psi(Y - X^{\mathrm{T}}\theta)X$ and $\nabla L(\theta) = -\mathbb{E}\left\{\psi\left(Y - X^{\mathrm{T}}\theta\right) X\right\}$. Hence, the SGD and perturbed SGD updates for θ_0, as defined in (3) and (5) respectively, are

$$\widehat\theta_n = \widehat\theta_{n-1} + \gamma_n\psi(Y_n - X_n^{\mathrm{T}}\widehat\theta_{n-1})X_n, \tag{24}$$
$$\widehat\theta_n^* = \widehat\theta_{n-1}^* + \gamma_n W_n\psi(Y_n - X_n^{\mathrm{T}}\widehat\theta_{n-1}^*)X_n. \tag{25}$$

If $\psi(\cdot)$ is not differentiable, neither the final-step plug-in estimation (14) nor the recursive plug-in estimation (15) is applicable. However, if the corresponding $\ell(\theta; Z)$ and $L(\theta)$ satisfy Assumptions A1-A4, the perturbation resampling procedure is applicable. Next we consider a special setting where the following Assumptions B1-B4 hold.

(B1). Assume that $\rho(u)$ is a convex function on $\mathcal{R}$ with the right derivative being $\psi_+(u)$ and left derivative being $\psi_-(u)$. Let $\psi(u)$ be a function such that $\psi_-(u) \le \psi(u) \le \psi_+(u)$. There exists constant $C_1 > 0$ such that $|\psi(u)| \le C_1(1 + |u|)$.

(B2). Let $\varepsilon_n = Y_n - X_n^{\mathrm{T}}\theta_0$. Assume that (X_n, ε_n), $n = 1, 2, ...$, are i.i.d. copies of (X, ε), with $\mathbb{E}\|X\|^4 < \infty$ and $\mathbb{E}\|\varepsilon\|^2 < \infty$. Let $V_0 = \mathbb{E}\{\psi^2(\varepsilon)XX^{\mathrm{T}}\} > 0$.

(B3). Let $\phi(u|X) = \mathbb{E}\{\psi(u + \varepsilon)|X\}$. Assume that $\phi(0|X) = 0$, $u\phi(u|X) > 0$ for any $u \ne 0$, and $\phi(u|X)$ has a derivative at $u = 0$ with $\dot\phi(0|X) \ge \sigma > 0$ uniformly over X. Let $S_0 = \mathbb{E}\{\dot\phi(0|X)XX^{\mathrm{T}}\} > 0$.

(B4). Assume that $\dot\phi(u|X)$ is uniformly Lipschitz at $u = 0$. That is, there exist constants $C_2 > 0$ and $\delta > 0$ such that $\left|\dot\phi(u|X) - \dot\phi(0|X)\right| \le C_2|u|$ for $|u| \le \delta$ uniformly over X.

We derive the asymptotic properties of the ψ-type M-estimator as follows.

Lemma 4 *If Assumptions B1-B4 and A5 are satisfied, then we have*

$$\sqrt{n}(\overline\theta_n - \theta_0) \Rightarrow \mathcal{N}\left(0, S_0^{-1}V_0S_0^{-1}\right), \text{ in distribution, as } n \to \infty. \tag{26}$$

Theorem 5 *If Assumptions B1-B4 and A5-A6 are satisfied, then we have*

$$\sqrt{n}(\overline\theta_n^* - \theta_0) = \frac{1}{\sqrt{n}}S_0^{-1}\sum_{i=1}^n W_i\psi(\varepsilon_i)X_i + o_p(1). \tag{27}$$

From Lemma 2 we see that the plug-in procedures are not applicable for estimating the asymptotic covariance matrix, because although they are applicable for estimating V_0, they are not applicable for estimating S_0, which involves $\dot\phi(0|X)$. Moreover, by Theorem 3, we can show that the Kolmogorov-Smirnov distance between $\sqrt{n}(\overline\theta_n^* - \overline\theta_n)$ and $\sqrt{n}(\overline\theta_n - \theta_0)$ converges to zero in probability, as stated in Theorem 2. This validates our proposal of the perturbation-based resampling procedure for robust regression. To further understand Assumptions B1-B4, we examine the following example of quantile regression, which is a special case of the above robust regression.

Example 4 (Quantile regression). Assume the τ-quantile of Y given X is $X^{\mathrm{T}}\theta_0$, with

$$\theta_0 = \arg\min_{\theta\in\mathcal{R}^p} \mathbb{E}\rho_\tau(Y - X^{\mathrm{T}}\theta), \tag{28}$$

where $\rho_\tau(u) = u(\tau - I(u < 0))$ with a given $0 < \tau < 1$. Let $\varepsilon = Y - X^{\mathrm{T}}\theta_0$ and $\psi_\tau(u) = \tau - I(u < 0)$. Thus $\mathbb{E}\{\psi_\tau(\varepsilon)|X\} = \tau - F_\varepsilon(0|X) = 0$, where $F_\varepsilon(u|X)$ is the conditional distribution function of ε. Let $p_\varepsilon(u|X)$ be the conditional density function of ε. Note that $\phi(u|X) = \mathbb{E}\{\psi_\tau(u + \varepsilon)|X\} = \tau - F_\varepsilon(-u|X)$, $\dot\phi(0|X) = p_\varepsilon(0|X)$, $V_0 = \mathbb{E}\{\psi_\tau^2(\varepsilon)XX^{\mathrm{T}}\} = \tau(1-\tau)\mathbb{E}\{XX^{\mathrm{T}}\}$ and $S_0 = \mathbb{E}\{p_\varepsilon(0|X)XX^{\mathrm{T}}\}$. Therefore, we can easily verify that if $p_\varepsilon(0|X)$ is uniformly bounded away from 0 and $p_\varepsilon(u|X)$ is uniformly Lipschitz continuous at $u = 0$, then Assumptions B1-B4 hold. Thus, the SGD and perturbed SGD updates for θ_0, as defined in (3) and (5) respectively, are

$$\widehat\theta_n = \widehat\theta_{n-1} + \gamma_n\left\{\tau - I(Y_n - X_n^{\mathrm{T}}\widehat\theta_{n-1} < 0)\right\}X_n, \tag{29}$$
$$\widehat\theta_n^* = \widehat\theta_{n-1}^* + \gamma_n W_n\left\{\tau - I(Y_n - X_n^{\mathrm{T}}\widehat\theta_{n-1}^* < 0)\right\}X_n, \tag{30}$$

and the asymptotic results stated in Lemma 2 and Theorem 3 follow directly.

4. Numerical results

4.1. Simulation studies

To assess the performance of the proposed online bootstrap resampling procedure (a.k.a. random weighting procedure; RW) for SGD estimators, we conduct simulation studies for those four examples discussed in Section 3. We compare the proposed procedure with the recursive plug-in procedure (RPI).

Setting 1 (Linear regression): Consider linear regression (16), where covariates $X^{(j)}$, $j = 1, \ldots, p$, and residual $\varepsilon = Y - X^{\mathrm{T}}\theta_0$, are independently generated from standard normal

$N(0,1)$. Here $X^{(j)}$ indicates the j-th dimension of X. Let $\theta_0 = (\mu \mathbf{1}_{q/2}^{\mathrm{T}}, -\mu \mathbf{1}_{q/2}^{\mathrm{T}}, \mathbf{0}_{p-q}^{\mathrm{T}})^{\mathrm{T}}$ (same for the other three settings). Consider the corresponding SGD estimators (17) and (18).

Setting 2 (Logistic regression): Consider logistic regression (19), where covariates $X^{(j)}$ are independently from $N(0,1)$ and response Y from Bernoulli distribution with $\mathrm{logit}\{P(Y=1|X)\} = X^{\mathrm{T}}\theta_0$. Consider the corresponding SGD estimators (20) and (21).

Setting 3 (LAD regression): Consider least absolute deviation (LAD) regression, which is a special case of robust regression (22) with $\rho(x) = |x|$ and quantile regression (28) with $\tau = 0.5$, where covariates $X^{(j)}$, $j = 1, \ldots, p$, are i.i.d. with $N(0,1)$ and residual ε, defined as $Y - X^{\mathrm{T}}\theta_0$, is independently from double exponential distribution $DE(0,1)$. Consider the corresponding SGD estimators (29) and (30) with $\tau = 0.5$.

Setting 4 (LAD regression for data with outliers): Consider LAD regression for the data generated from Setting 1 but contaminated with 10% outliers. The contaminated data are obtained by transforming the outcome variable in the data generated from Setting 1 using $Y \leftarrow Y + 10$ if $|X^{(1)}| \geq 1.96$ and $|X^{(2)}| < 1.96$; and $Y \leftarrow Y - 10$ if $|X^{(1)}| < 1.96$ and $|X^{(2)}| \geq 1.96$. In this setting, covariate vector X and residual $\varepsilon = Y - X^{\mathrm{T}}\theta_0$ are not independent, but $\mathrm{median}\{\varepsilon|X\} = 0$. Consider the corresponding SGD estimators defined in (29) and (30) with $\tau = 0.5$.

For each simulation setting, we consider four scenarios, as described by (N, p, q, μ), where sample size $N = 10,000$ or $20,000$, number of covariates $p = 10$ or 20, number of informative covariates $q = 6$, and effect size $\mu = 0.1$ or 0.2. For each example, we repeat the data generation 1000 times. For each data repetition, we use $W_{nb} \sim \exp(1)$ as random weights and generate $B = 200$ copies of random weights whenever a new data point is read. Then, for each data repetition, we obtain the SGD estimate (4), and apply the following procedures to construct 95% confidence intervals: the proposed random weighting procedure to obtain 2.5% and 97.5% quantile (RW-Q), the proposed random weighting procedure to estimate its standard error (RW-σ) and then construct "estimate $\pm$ 1.96 $\times$ $\widehat{\mathrm{SE}}$", and the recursive plug-in procedures (RPI), if applicable, to estimate its standard error. We consider the learning rate $\alpha = 2/3$. When we calculate the average SGD estimators (4) and (6), the first 2000 and 4000 estimates are excluded for $N = 10,000$ and $N = 20,000$, respectively. We also obtain the empirical standard error based on 1000 repeated SGD estimates, as a benchmark approximation to the true standard error.

The coverage probabilities of the 95% confidence interval estimates are summarized in Table 1 for linear regression (Setting 1), Table 2 for logistic regression (Setting 2) and Table 3 for LAD regression (Settings 3-4), respectively. We only report results corresponding to the first, fourth and seventh covariates (that is, $X^{(1)}$, $X^{(q/2+1)}$ and $X^{(q+1)}$). The plug-in procedures are not applicable for LAD regression in Settings 3-4.

From Tables 1 and 2, we see that, for linear regression and logistic regression, the coverage probabilities using the RW-Q, RW-σ and RPI are close to 95%. Therefore, the proposed random weighting procedures (both RW-Q and RW-σ) perform well for linear regression and logistic regression, and if we choose to use the plug-in procedure, which involves matrix inverse, we can use RPI. Since the point estimate is $\sum_{i=N_0+1}^{N} \widehat{\theta}_i/(N - N_0)$, where N_0 is the number of excluded estimates and which involves a pass of SGD estimates, its standard error should also involve a pass of SGD estimates, instead of involving only the final-step estimate or the true parameter value. We can understand this clearer if we see the "SE" column, where the standard errors from RW-σ and RPI are close to the empirical standard error. Moveover, from Table 3, we see that, for LAD regression and for both Settings 3 and 4, the coverage probabilities using either of the two proposed random weighting procedures are close to 95%, and the standard errors using RW-σ are close to the empirical standard errors.

Table 1: Coverage probabilities of 95% confidence intervals for linear regression

(N, p, q, μ)	Method	Dim 1		Dim $q/2+1$		Dim $q+1$	
		Cover	SE	Cover	SE	Cover	SE
(10000,10,6,0.1)	RW-Q	0.947	–	0.946	–	0.950	–
	RW-σ	0.956	0.012	0.950	0.012	0.959	0.012
	RPI	0.953	0.011	0.946	0.011	0.956	0.011
	Empirical	–	0.011	–	0.011	–	0.011
(10000,10,6,0.2)	RW-Q	0.947	–	0.946	–	0.950	–
	RW-σ	0.956	0.012	0.950	0.012	0.959	0.012
	RPI	0.953	0.011	0.946	0.011	0.956	0.011
	Empirical	–	0.011	–	0.011	–	0.011
(20000,20,6,0.1)	RW-Q	0.939	–	0.951	–	0.945	–
	RW-σ	0.949	0.008	0.953	0.008	0.960	0.008
	RPI	0.956	0.008	0.948	0.008	0.957	0.008
	Empirical	–	0.008	–	0.008	–	0.008
(20000,20,6,0.2)	RW-Q	0.939	–	0.951	–	0.945	–
	RW-σ	0.949	0.008	0.953	0.008	0.960	0.008
	RPI	0.956	0.008	0.948	0.008	0.957	0.008
	Empirical	–	0.008	–	0.008	–	0.008

4.2. Real data applications

In this section, we apply the proposed procedures to conduct inference for linear regression analysis for the individual household electric power consumption data (POWER) and logistic regression analysis for the skin segmentation dataset (SKIN) and gas sensors for the home Activity monitoring data (GAS). All the three datasets are publicly available on UCI machine learning repository.

The POWER dataset contains 2,075,259 observations and we fit a linear model to investigate how the time of a day influences the response variable "sub-metering-1", the energy sub-metering No. 1, in watt-hour of active energy corresponding to kitchen. The observations with missing value are deleted, and the time of a day is divided into 8 categories, "*Time 0-2*", "*Time 3-5*", ..., and "*Time 21-23*". The SKIN dataset contains 245,057 observations, out of which 50,859 is the skin samples and 194,198 is non-skin samples. We fit a logistic model to examine the relationship between the indicator of skin and three predictors, B, G and R. The GAS dataset contains 919,438 observations and we only use a subset containing 652,024 observations with response variable being either "banana" or "wine". We fit a logistic model to examine the association between the response variable and 11 explanatory variables, *Time*, *R1* to *R8*, *Temperature* and *Humidity*.

Table 2: Coverage probabilities of 95% confidence intervals for logistic regression

(N,p,q,μ)	Method	Dim 1		Dim $q/2+1$		Dim $q+1$	
		Cover	SE	Cover	SE	Cover	SE
(10000,10,6,0.1)	RW-Q	0.950	–	0.948	–	0.928	–
	RW-σ	0.954	0.024	0.954	0.024	0.948	0.023
	RPI	0.954	0.023	0.952	0.023	0.942	0.023
	Empirical	–	0.022	–	0.022	–	0.023
(10000,10,6,0.2)	RW-Q	0.945	–	0.945	–	0.944	–
	RW-σ	0.959	0.025	0.954	0.025	0.954	0.024
	RPI	0.955	0.023	0.952	0.023	0.948	0.023
	Empirical	–	0.023	–	0.023	–	0.023
(20000,20,6,0.1)	RW-Q	0.944	–	0.939	–	0.934	–
	RW-σ	0.951	0.016	0.953	0.016	0.952	0.016
	RPI	0.948	0.016	0.953	0.016	0.949	0.016
	Empirical	–	0.016	–	0.016	–	0.016
(20000,20,6,0.2)	RW-Q	0.946	–	0.941	–	0.939	–
	RW-σ	0.958	0.017	0.952	0.017	0.957	0.016
	RPI	0.950	0.016	0.944	0.016	0.952	0.015
	Empirical	–	0.016	–	0.016	–	0.016

Although standard softwares such as SAS and R can fit linear and logistic regression to such datasets without difficulty, for the illustration purpose, we use the SGD as in Examples 1 and 2 to fit linear and logistic regression and use the proposed online boostrap procedure to construct confidence intervals. The point estimates and the 95% confidence intervals of the coefficients are showed in Table 4. From the left-top panel of Table 4, we see that the electronic power consumption from kitchen is relatively high in the evening and night. From the left-bottom panel of Table 4, we see that variable B is positively associated with the response while the other two variables G and R are negatively associated. From the right panel of Table 4, we see that all the variables but *R4* are statistical significantly associated with the response. Furthermore, we display the histogram of $B = 1000$ perturbation-based SGD estimates for each coefficient in Figures 4.2-4.2 for the POWER data, the SKIN data and the GAS data, respectively. The blue triangle in each figure indicates the corresponding point estimate the red triangles indicate 2.5 and 97.5 quantiles. From these figures, we see the the perturbation-based procedure can be used to approximate the sampling distribution of the corresponding SGD estimator, which might be skewed. For example, for the GAS data, the sampling distributions are very skewed, and therefore the proposed resampling procedure is able to display such skewness.

5. Discussion

Online updating is a useful strategy for analyzing big data and streaming data, and recently stochastic gradient decent has become a popular method for doing online updating. Although the asymptotic properties of SGD have been well studied, there is little research

Table 3: Coverage probabilities of 95% confidence intervals for LAD regression

(N,p,q,μ)	Method	Dim 1		Dim $q/2+1$		Dim $q+1$	
		Cover	SE	Cover	SE	Cover	SE
Simulation Setting 3							
(10000,10,6,0.1)	RW-Q	0.965	–	0.965	–	0.969	–
	RW-σ	0.969	0.029	0.965	0.029	0.965	0.029
	Empirical	–	0.026	–	0.027	–	0.026
(10000,10,6,0.2)	RW-Q	0.972	–	0.965	–	0.967	–
	RW-σ	0.973	0.029	0.966	0.029	0.968	0.029
	Empirical	–	0.026	–	0.027	–	0.026
(20000,20,6,0.1)	RW-Q	0.970	–	0.973	–	0.966	–
	RW-σ	0.966	0.010	0.969	0.010	0.965	0.010
	Empirical	–	0.009	–	0.009	–	0.009
(20000,20,6,0.2)	RW-Q	0.967	–	0.974	–	0.966	–
	RW-σ	0.966	0.010	0.969	0.010	0.969	0.010
	Empirical	–	0.009	–	0.009	–	0.009
Simulation Setting 4							
(10000,10,6,0.1)	RW-Q	0.954	–	0.971	–	0.950	–
	RW-σ	0.958	0.035	0.969	0.035	0.960	0.035
	Empirical	–	0.032	–	0.031	–	0.033
(10000,10,6,0.2)	RW-Q	0.960	–	0.966	–	0.955	–
	RW-σ	0.958	0.035	0.966	0.035	0.956	0.035
	Empirical	–	0.032	–	0.031	–	0.033
(20000,20,6,0.1)	RW-Q	0.948	–	0.953	–	0.965	–
	RW-σ	0.963	0.047	0.958	0.047	0.964	0.047
	Empirical	–	0.043	–	0.045	–	0.044
(20000,20,6,0.2)	RW-Q	0.944	–	0.957	–	0.961	–
	RW-σ	0.961	0.047	0.961	0.047	0.961	0.047
	Empirical	–	0.044	–	0.045	–	0.044

on conducting statistical inference based on SGD estimators. In this paper, we propose the perturbation-based resampling procedure, which can be applied to estimate the sampling distribution of an SGD estimator. The offline version of perturbation-based resamping procedure was first proposed by Rubin (1981) and was also discussed in Shao and Tu (2012).

The proposed resampling procedure is in essence an online version of the bootstrap. Recall that the data points, $Z_1, Z_2, \ldots, Z_N$, are arriving one at a time and an SGD estimate updates itself from $\widehat{\theta}_{n-1}$ to $\widehat{\theta}_n$ whenever a new data point Z_n arrives. If we are forced to apply the bootstrap, then we should have many bootstrap samples; the data points of each bootstrap sample, $Z_1^*, Z_2^*, \ldots, Z_N^*$, are assumed to be arriving one at a time and the bootstrapped SGD estimate updates itself from $\widehat{\theta}_{n-1}^*$ to $\widehat{\theta}_n^*$ whenever a new data point Z_n^* arrives. Of course such bootstrap is impractical here because in online updating we will not obtain and store all the data points and then generate bootstrap samples. Now if we rearrange hypothetical bootstrap sample $Z_1^*, Z_2^*, \ldots, Z_N^*$ as

Table 4: Point estimates and 95% confidence intervals for three real datasets; POWER data on the left-top panel, SKIN data on the left-bottom panel and GAS data on the right panel

Variable	Est.	95% CI	Variable	Est.	95% CI
Time 0-2	2.265	(2.254, 2.275)	*Time*	−0.158	(−0.178, −0.139)
Time 3-5	2.045	(2.040, 2.049)	*R1*	−0.202	(−0.215, −0.190)
Time 6-8	2.623	(2.608, 2.639)	*R2*	0.176	(0.160, 0.191)
Time 9-11	3.323	(3.298, 3.347)	*R3*	−0.907	(−0.932, −0.882)
Time 12-14	3.445	(3.420, 3.470)	*R4*	−0.007	(−0.018, 0.004)
Time 15-17	3.059	(3.037, 3.082)	*R5*	−0.450	(−0.467, −0.432)
Time 18-20	4.176	(4.143, 4.208)	*R6*	1.772	(1.759, 1.785)
Time 21-23	4.053	(4.024, 4.082)	*R7*	0.173	(0.139, 0.207)
B	1.501	(1.441, 1.569)	*R8*	0.302	(0.272, 0.332)
G	−0.242	(−0.319, −0.166)	*Temp.*	−0.175	(−0.191, −0.160)
R	−1.956	(−1.999, −1.918)	*Humi.*	−0.551	(−0.560, −0.542)

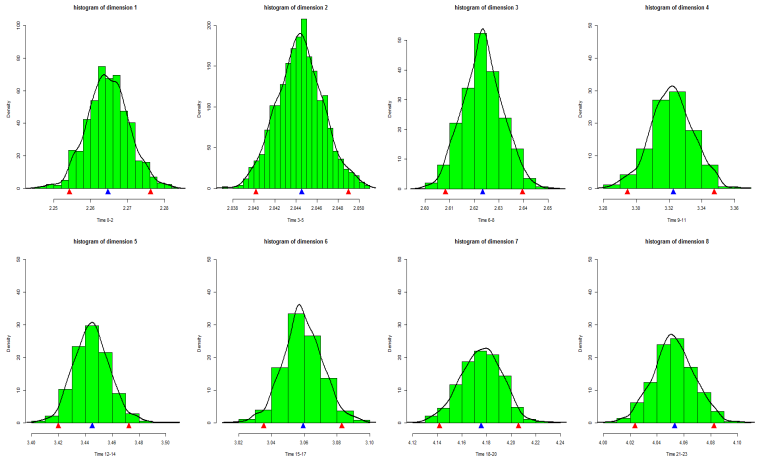


Figure 1: Histograms of $B = 1000$ perturbation-based SGD estimates for the POWER data.

$\{K_1$ copies Z_1, K_2 copies $Z_2, \ldots, K_N$ copies $Z_N\}$, where K_n follows binomial distribution $B(N, 1/N)$, then the SGD estimator updates itself from $\widehat{\theta}^*_{n-1}$ to $\widehat{\theta}^*_n$ whenever a new batch of data points, K_n copies of Z_n, arrives. Noting that binomial distribution $B(N, 1/N)$ approximates to Poisson distribution $P(1)$ as $N \to \infty$, we see that the aforementioned hypothetical

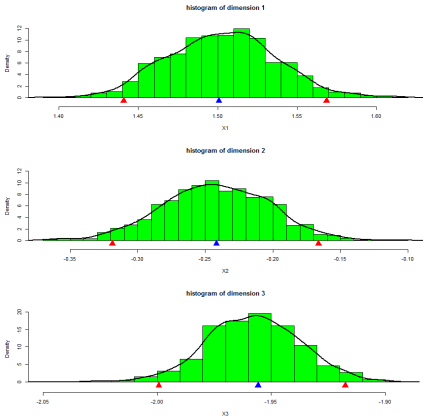


Figure 2: Histograms of $B = 1000$ perturbation-based SGD estimates for the SKIN data.

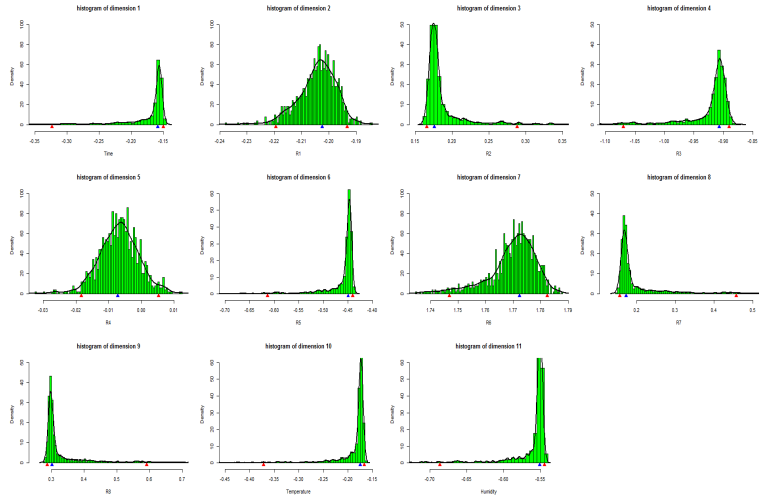


Figure 3: Histograms of $B = 1000$ perturbation-based SGD estimates for the GAS data.

bootstrap is equivalent to our proposed online bootstrap procedure with $W_n \sim P(1)$, whose mean and variance are both equal to one.

Finally, the SGD method considered in this paper is actually the explicit SGD, in contrast with the implicit SGD considered in Toulis et al (2017). We are working on extending

the proposed perturbation-based resampling procedure for conducting statistical inference for the implicit SGD.

Appendix A. technical proofs

For ease exposition of establishing asymptotic normality of SGD and perturbed SGD estimates, we present the following Proposition 1, adapted from Theorem 2 of Polyak and Juditsky (1992; pp.841). Let $R(\theta) : \mathcal{R}^p \to \mathcal{R}^p$ be some unknown function and $R(\theta_0) = 0$. The dataset consists of $Z_n, n = 1, 2, \ldots,$ which are i.i.d. copies of Z. Stochastic gradients are $\widehat{R}(\theta; Z_i)$ and $\mathbb{E}\{\widehat{R}(\theta; Z_i)\} = R(\theta)$. With an initial point $\widehat{\theta}_0$ and learning rates γ_n, the SGD estimate is defined as

$$\widehat{\theta}_n = \widehat{\theta}_{n-1} - \gamma_n \widehat{R}(\widehat{\theta}_{n-1}; Z_n) = \widehat{\theta}_{n-1} - \gamma_n \left(R(\widehat{\theta}_{n-1}) - D_n \right), \tag{31}$$

where $D_n = R(\widehat{\theta}_{n-1}) - \widehat{R}(\widehat{\theta}_{n-1}; Z_n)$ is a martingale-difference process; that is, $\mathbb{E}\{D_n | \mathfrak{F}_{n-1}\} = 0$, where $\mathfrak{F}_{n-1} = \sigma(\mathcal{D}_{n-1})$. The regularity conditions for Proposition 1 are listed as follows.

(C1). There exists a function $U(\theta) : \mathcal{R}^p \to \mathcal{R}$ such that for some $\lambda > 0$, $\delta > 0$, $l_0 > 0$, $L_0 > 0$, and all $\theta, \theta' \in \mathcal{R}^p$, the conditions $U(\theta) \geq \lambda \|\theta\|^2$, $\|\nabla U(\theta) - \nabla U(\theta')\| \leq L_0 \|\theta - \theta'\|$, $U(0) = 0$, $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) > 0$ for $\theta \neq \theta_0$ hold true. Moreover, $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) \geq l_0 U(\theta - \theta_0)$ for all $\|\theta - \theta_0\| \leq \delta$.

(C2). There exists a positive definite matrix $S_0 \in \mathcal{R}^{p \times p}$ such that for some $C > 0$, $0 < \varrho \leq 1$, and $\delta > 0$, the condition $\|R(\theta) - S_0(\theta - \theta_0)\| \leq C\|\theta - \theta_0\|^{1+\varrho}$ for all $\|\theta - \theta_0\| \leq \delta$ holds true.

(C3). $\{D_n\}_{n \geq 1}$ is a martingale difference process with $\mathbb{E}\{D_n | \mathfrak{F}_{n-1}\} = 0$, and for some $C > 0$,

$$\mathbb{E}\left\{ \|D_n\|^2 \,|\, \mathfrak{F}_{n-1} \right\} + \|R(\widehat{\theta}_{n-1})\|^2 \leq C \left(1 + \|\widehat{\theta}_{n-1}\|^2 \right) \text{ a.s.},$$

for all $n \geq 1$. Consider decomposition $D_n = D_n(0) + E_n(\widehat{\theta}_{n-1})$, where $D_n(0) = R(\theta_0) - \widehat{R}(\theta_0; Z_n)$ and $E_n(\widehat{\theta}_{n-1}) = D_n - D_n(0)$. Assume that $\mathbb{E}\{D_n(0) | \mathfrak{F}_{n-1}\} = 0$ a.s.,

$$\mathbb{E}\{D_n(0) D_n(0)^{\mathrm{T}} | \mathfrak{F}_{n-1}\} \xrightarrow{P} V_0 > 0,$$
$$\sup_{n \geq 1} \mathbb{E}\left\{ \|D_n(0)\|^2 I(\|D_n(0)\| > \eta) | \mathfrak{F}_{n-1} \right\} \xrightarrow{P} 0, \text{ as } \eta \to \infty,$$

and there exists $\delta(x) \to 0$ as $x \to 0$ such that, for all n large enough,

$$\mathbb{E}\left\{ \|E_n(\widehat{\theta}_{n-1})\|^2 \,|\, \mathfrak{F}_{n-1} \right\} \leq \delta(\|\widehat{\theta}_{n-1} - \theta_0\|) \text{ a.s..}$$

(C4). It holds that $(\gamma_n - \gamma_{n+1})/\gamma_n = o(\gamma_n)$, $\gamma_n > 0$ for all n, and $\sum_{n=1}^{\infty} \gamma_n^{(1+\varrho)/2} n^{-1/2} < \infty$.

Assumptions C2 and C4 are implied by the following Assumptions C2′ (i.e. Assumption C2 with $\varrho = 1$) and C4′ (i.e. Assumption A5):

(C2′). There exists a positive definite matrix $S \in \mathcal{R}^{p \times p}$ such that for some $C > 0$ and $\delta > 0$, the condition $\|R(\theta) - S_0(\theta - \theta_0)\| \leq C\|\theta - \theta_0\|^2$ for all $\|\theta - \theta_0\| \leq \delta$ holds true.

(C4′). The learning rates are chosen as $\gamma_n = \gamma_1 n^{-\alpha}$ with $\gamma_1 > 0$ and $\alpha \in (0.5, 1)$.

Proposition 1. *If Assumptions C1-C4 are satisfied, then we have* $\overline{\theta}_n \to \theta_0$, *a.s.; and*

$$\sqrt{n}(\overline{\theta}_n - \theta_0) = \frac{1}{\sqrt{n}} S^{-1} \sum_{i=1}^{n} D_i + o_p(1), \tag{32}$$

which implies $\sqrt{n}(\overline{\theta}_n - \theta_0) \Rightarrow \mathcal{N}\left(0, S_0^{-1} V_0 S_0^{-1}\right)$, *in distribution.*

A1. Proof of Lemma 1

Proof. By Proposition 1, it suffices to show that Assumptions C1-C4 hold under Assumptions A1-A5. Because C2 and C4 are implied by C2′ and C4′, it suffices to show that Assumptions C1, C2′, C3 and C4′ hold under Assumptions A1-A5.

Verification of Assumption C1: Recall that $R(\theta) = \nabla L(\theta)$ and $\widehat{R}(\theta; Z_i) = \nabla \ell(\theta; Z_i)$. Define $U(\theta) = L(\theta_0 + \theta) - L(\theta_0) + \lambda\|\theta\|^2$ for a given $\lambda > 0$. By definition of $U(\theta)$ and Assumption A1, we have $U(\theta) \geq \lambda\|\theta\|^2$ and $U(0) = 0$. For any θ and θ', since $\nabla U(\theta) - \nabla U(\theta') = R(\theta + \theta_0) - R(\theta' + \theta_0) + 2\lambda(\theta - \theta')$, letting $L_0 = L_1 + 2\lambda$ and by Assumption A2, we have $\|\nabla U(\theta) - \nabla U(\theta')\| \leq L_1\|\theta - \theta'\|$. Since $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) = \|R(\theta)\|^2 + \lambda(\theta - \theta_0)^{\mathrm{T}} R(\theta)$, by Assumption A1, we have $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) > 0$ for any $\theta \neq \theta_0$. Last, it remains to verify there exist $l_0 > 0$ and $\delta > 0$ such that $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) \geq l_0 U(\theta - \theta_0)$ for all $\|\theta - \theta_0\| \leq \delta$. Noting that $U(\theta - \theta_0) = L(\theta + \theta_0) + \lambda\|\theta - \theta_0\|^2$, by Taylor expansion and Assumption A3, we see that there exist $l_1 > 0$ and δ_1 such that $U(\theta - \theta_0) \leq l_1\|\theta - \theta_0\|^2$ for all $\|\theta - \theta_0\| \leq \delta_1$. On the other hand, noting that $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) = \|R(\theta)\|^2 + \lambda(\theta - \theta_0)^{\mathrm{T}} R(\theta)$, by Assumption A3 also, we see that there exist $l_2 > 0$ and δ_2 such that $(\theta - \theta_0)^{\mathrm{T}} R(\theta) \geq l_2\|\theta - \theta_0\|^2$ for all $\|\theta - \theta_0\| \leq \delta_2$. Selecting $\delta = \min(\delta_1, \delta_2)$ and $l_0 = \lambda l_2 / l_2$, we show that $\nabla U(\theta - \theta_0)^{\mathrm{T}} R(\theta) \geq l_0 U(\theta - \theta_0)$ for all $\|\theta - \theta_0\| \leq \delta$.

Verification of Assumption C2′: Recall that $R(\theta) = \nabla L(\theta)$, $S(\theta) = \nabla R(\theta)$, and $S_0 = S(\theta_0) > 0$. By Assumption A3, there exists $\delta > 0$ such that for any $\|\theta - \theta_0\| < \delta'$, $\|S(\theta) - S(\theta_0)\| < L_2\|\theta - \theta_0\|$. By mean-value theorem, $\|S(\theta) - S_0(\theta - \theta_0)\| = \|S(\widetilde{\theta})(\theta - \theta_0) - S_0(\theta - \theta_0)\|$, where $\widetilde{\theta}$ lies between θ and θ_0. Hence $\|S(\theta) - S_0(\theta - \theta_0)\| \leq L_2\|\theta - \theta_0\|^2$, for any $\|\theta - \theta_0\| \leq \delta$. Letting $C = L_2$, we have verified Assumption C2′.

Verification of Assumption C3: $D_n = R(\widehat{\theta}_{n-1}) - \widehat{R}(\widehat{\theta}_{n-1}; Z_n)$. Consider decomposition $D_n = D_n(0) + E_n(\widehat{\theta}_{n-1})$, where $D_n(0) = -\nabla \ell(\theta_0, Z_n)$ and $E_n(\widehat{\theta}_{n-1}) = [R(\widehat{\theta}_{n-1}) - R(\theta_0)] - [\nabla \ell(\widehat{\theta}_{n-1}; Z_n) - \nabla \ell(\theta_0; Z_n)]$. By Assumption A2, $\|R(\widehat{\theta}_{n-1})\|^2 \leq L_1^2 \|\widehat{\theta}_{n-1} - \theta_0\|^2$. In addition, Cauchy-Schwartz inequality implies that $\mathbb{E}\left\{ \|\nabla \ell(\theta; Z_n) - \nabla \ell(\theta_0; Z_n)\|^2 \right\} = 2\mathbb{E}\|\nabla \ell(\theta; Z_n)\|^2 + 2\mathbb{E}\|\nabla \ell(\theta_0; Z_n)\|^2$; we have $\mathbb{E}\left\{ \|\nabla \ell(\widehat{\theta}_{n-1}; Z_n) - \nabla \ell(\theta_0; Z_n)\|^2 \,| \right\} \leq C'(1 + \|\theta\|^2)$ for some $C' > 0$ by Assumption A4, . Together, we have $\mathbb{E}\left\{ \|D_n\|^2 \,|\, \mathfrak{F}_{n-1} \right\} + \|R(\widehat{\theta}_{n-1})\|^2 \leq C\left(1 + \|\widehat{\theta}_{n-1}\|^2\right)$ for some $C > 0$. Moreover, because $D_n(0)$'s are i.i.d., we have $\mathbb{E}\{D_n(0) D_n(0)^{\mathrm{T}} | \mathfrak{F}_{n-1}\} = V_0 > 0$ and $\sup_{n \geq 1} \mathbb{E}\left\{ \|D_n(0)\|^2 I(\|D_n(0)\| > \eta) | \mathfrak{F}_{n-1} \right\} \xrightarrow{P} 0$, as $\eta \to \infty$. Finally, note that $\mathbb{E}\|E_n(\theta)\|^2 \leq L_1^2\|\theta - \theta_0\|^2 + \mathbb{E}\|\nabla \ell(\theta; Z_n) - \nabla \ell(\theta_0; Z_n)\|^2$. By Assumption A3, $\mathbb{E}\|\nabla \ell(\theta; Z_n) - \nabla \ell(\theta_0; Z_n)\|^2 \leq \delta'(\|\theta - \theta_0\|)$ for some $\delta'(\cdot)$ with $\delta'(x) \to 0$ as $x \to 0$. Define $\delta(x) = L_1^2 x^2 + \delta'(x)$, we show $\mathbb{E}\|E_n(\theta)\|^2 \leq \delta(\|\theta - \theta_0\|)$. This complete the verification of Assumption C3.

Obviously Assumption A5 is the same as Assumption C4′. Therefore, we have verified that Assumptions A1-A5 imply Assumptions C1, C2′, C3 and C4′. By Proposition 1, we complete the proof of Lemma 1. □

A2. Proof of Theorem 1

Proof. Rewrite $\widehat{\theta}_n^*$ as

$$\widehat{\theta}_n^* = \widehat{\theta}_{n-1}^* - \gamma_n R(\widehat{\theta}_{n-1}^*) + \gamma_n D_n^*, \tag{33}$$

where $D_n^* = R(\widehat{\theta}_{n-1}^*) - W_n \nabla \ell(\widehat{\theta}_{n-1}^*; Z_n)$. Then let $\mathfrak{F}_{n-1}^*$ be the Borel field generated by $\{(Z_i, W_i), i \le n-1\}$. Since $\mathbb{E}\{W_n|\mathfrak{F}_{n-1}^*\} = 1$ and $R(\theta) = \mathbb{E}\{\nabla \ell(\theta; Z_n)\}$, we have $\mathbb{E}\{D_n^*|\mathfrak{F}_{n-1}\} = 0$. Thus D_n^* is a martingale-difference process. Let $D_n^*(\theta) = R(\theta) - W_n \nabla \ell(\theta; Z_n)$. Consider decomposition $D_n^* = D_n^*(0) + E_n^*(\widehat{\theta}_{n-1}^*)$, where

$$D_n^*(0) = -W_n \nabla \ell(\theta_0; Z_n) \tag{34}$$

and

$$E_n^*(\theta) = [R(\theta) - R(\theta_0)] - W_n[\nabla \ell(\theta; Z_n) - \nabla \ell(\theta_0; Z_n)]. \tag{35}$$

Noting that $\mathbb{E}\{D_n^*(0)\} = 0$ and $\mathbb{E}(W_n^2) = 2$ under Assumption A6, by Assumption A4, we have

$$\mathbb{E}\{[D_n^*(0)][D_n^*(0)]^{\mathrm{T}}\} = 2\mathbb{E}\left\{[\ell(\theta_0; Z_n)][\ell(\theta_0; Z_n)]^{\mathrm{T}}\right\} = 2V_0. \tag{36}$$

By Cauchy-Schwartz inequality and Assumptions A2 and A4, we have

$$\mathbb{E}\{\|E_n^*(\theta)\|^2\} \le 2\|R(\theta)\|^2 + 4\mathbb{E}\left\{\|\nabla \ell(\theta, Z) - \nabla \ell(\theta_0, Z)\|^2\right\} \le \delta'(\|\theta - \theta_0\|), \tag{37}$$

where $\delta'(x) = 2L_1^2 x^2 + 2\delta(x)$ satisfying that $\delta'(x) \to 0$ as $x \to 0$. Also by Cauchy-Schwartz inequality, $\mathbb{E}\{\|E_n^*(\theta)\|^2\} \le 2\|R(\theta)\|^2 + 2\mathbb{E}\{\|\nabla \ell(\theta, Z)\|^2\}$. Further, by Assumptions A2 and A4,

$$\mathbb{E}\{\|D_n^*\|^2|\mathfrak{F}_{n-1}\} + \|R(\widehat{\theta}_{n-1}^*)\|^2 \le 3L_1^2\|\widehat{\theta}_{n-1}^* - \theta_0\|^2 + 2C\|\widehat{\theta}_{n-1}^*\|^2 \le C'(1 + \|\widehat{\theta}_{n-1}^* - \theta_0\|^2), \tag{38}$$

for some large enough $C' > 0$. Combining results (36)-(38), we have verified Assumption C3. Moreover, noting that $\mathbb{E}W_n = 1$, we can easily verify that Assumptions A1 and A2 imply that Assumption C1 holds, and Assumption A3 implies that Assumption C2 holds. By Proposition 1, we show that $\widehat{\theta}_n^* \to \theta_0$ almost surely, and

$$\begin{aligned}\sqrt{n}(\overline{\theta}_n^* - \theta_0) &= \frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^n D_i^* + o_p(1) \\ &= -\frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^n W_i \nabla \ell(\theta_0; Z_i) + \frac{1}{\sqrt{n}} S_0^{-1} \sum_{i=1}^n E_n^*(\widehat{\theta}_{n-1}^*) + o_p(1). \end{aligned} \tag{39}$$

Note that $\mathbb{E}\{\|E_n(\widehat{\theta}_{n-1}^*)\|^2|\mathfrak{F}_{n-1}\} \le \delta'(\|\widehat{\theta}_{n-1}^* - \theta_0\|)$, following (37). Since $\widehat{\theta}_n^* \to \theta_0$ a.s., we have $\delta(\|\widehat{\theta}_{n-1}^* - \theta_0\|) \to 0$ a.s.. Thus, $S_0^{-1} \sum_{i=1}^n E_n(\widehat{\theta}_{n-1}^*)/\sqrt{n} = o_p(1)$. Therefore, by (39), this completes the proof of Theorem 1. □

A3. Proof of Theorem 2

Proof. Let

$$V_n = -\frac{1}{\sqrt{n}} S_0^{-1}(W_i - 1)\nabla \ell(\theta_0, Z_i). \tag{40}$$

By Theorem 1, we have $\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n) = V_n + o_p(1)$. We first show that, for any $\beta \in \mathbf{B} \triangleq \{\beta \in \mathcal{R}^p : \|\beta\| = 1\}$ and $u \in \mathcal{R}$,

$$\mathbb{P}^*\left(\beta^{\mathrm{T}} V_n \le u\right) \to \Phi(u/\sigma_\beta), \text{ in probability}, \tag{41}$$

where $\Phi(u)$ is the distribution of $\mathcal{N}(0,1)$ and $\sigma_\beta = \beta^{\mathrm{T}} S_0^{-1} V_0 S_0^{-1} \beta$. In fact,

$$\beta^{\mathrm{T}} V_n = \frac{1}{\sqrt{n}} \sum_{i=1}^n (W_i - 1)\xi_i, \tag{42}$$

where $\xi_i = -\beta^{\mathrm{T}} S_0^{-1} \nabla \ell(\theta_0, Z_i)$. Note that, by Assumption A6, $\mathbb{E}W_i = \mathrm{Var}(W_i) = 1$. Hence

$$s_n = \frac{1}{n} \sum_{i=1}^n \mathrm{Var}^*\{(W_i - 1)\xi_i\} = \frac{1}{n} \sum_{i=1}^n \xi_i^2 \to \sigma_\beta^2, \text{ in probability}, \tag{43}$$

and for any $\epsilon > 0$,

$$\frac{1}{n s_n} \sum_{i=1}^n \mathbb{E}^*\left\{(W_i - 1)^2 \xi_i^2 I\left[|(W_i - 1)\xi_i| > \sqrt{n} s_n \epsilon\right]\right\} \to 0, \text{ in probability}. \tag{44}$$

Therefore, the Lindeberg's condition is satisfied. By the central limit theorem, (41) holds, which implies that for any $\beta \in \mathbf{B}$,

$$\sup_{u \in \mathcal{R}} |\mathbb{P}^*\left(\beta^{\mathrm{T}} V_n \le u\right) - \Phi(u/\sigma_\beta)| \to 0, \text{ in probability}. \tag{45}$$

Consider $\mathbf{B}_0 \triangleq \{\beta \in \mathcal{R}^p : \|\beta\| = 1, \text{the components of } \beta \text{ are rational}\}$, which contains only countable many β and is a dense subset of $\mathbf{B}$. For any subsequence $\{n_1\}$, by Cantor's "diagonal method" used in Rao and Zhao (1992), we can show that there exists a subsequence $\{n_2\} \subset \{n_1\}$ such that, with probability one,

$$\sup_{u \in \mathcal{R}} |\mathbb{P}^*\left(\beta^{\mathrm{T}} V_{n-1} \le u\right) - \Phi(u/\sigma_\beta)| \to 0, \text{ for any } \beta \in \mathbf{B}_0. \tag{46}$$

Hence, we show that

$$\sup_{v \in \mathcal{R}^p} \left|\mathbb{P}^*\left(\sqrt{n}(\overline{\theta}_n^* - \overline{\theta}_n) \le v\right) - \mathbb{P}(\zeta \le v)\right| \to 0, \text{ in probability}, \tag{47}$$

where $\zeta \sim \mathcal{N}(0, S_0^{-1} V_0 S_0^{-1})$. By Lemma 1, we can also show that

$$\sup_{v \in \mathcal{R}^p} \left|\mathbb{P}\left(\sqrt{n}(\overline{\theta}_n - \theta_0) \le v\right) - \mathbb{P}(\zeta \le v)\right| \to 0. \tag{48}$$

Combining (47) and (48), we complete the proof of Theorem 2. □

A4. Proof of Lemma 2

Proof. By Proposition 1, it suffices to show that Assumptions C1, C2′ and C3 hold under Assumptions B1-B4.

Verification of Assumption C1: Define $\Delta = \theta - \theta_0$. Let $R(\Delta) = \mathbb{E}\{\phi(\Delta^{\mathrm{T}}X|X)X\}$ and $\widehat{R}(\Delta; Z_n) = \psi(\Delta^{\mathrm{T}}X_n+\varepsilon_n)X_n$. Define $U(\Delta) = \Delta^{\mathrm{T}}\Delta$. By definition of $U(\Delta)$, $U(\Delta) \geq \lambda\|\Delta\|^2$ with $\lambda = 1$, $U(0) = 0$ and $\nabla U(\Delta)$ is Lipschitz continuous. Since $\Delta^{\mathrm{T}}\mathbb{E}\{\phi(\Delta^{\mathrm{T}}X|X)\Delta^{\mathrm{T}}X\} > 0$ by Assumption B3, we see that $\nabla U(\Delta)^{\mathrm{T}}R(\Delta) > 0$. By the mean-value theorem and by Assumption B3, there exists δ such that $\Delta^{\mathrm{T}}\mathbb{E}\{\phi(\Delta^{\mathrm{T}}X|X)\Delta^{\mathrm{T}}X\} \geq \Delta^{\mathrm{T}}\mathbb{E}\{\dot{\phi}(0|X)XX^{\mathrm{T}}\}\Delta/2 \geq \lambda_{\min}(S_0)\|\Delta\|^2/2$, for any $\|\Delta\| \leq \delta$, where $\lambda_{\min}(S_0)$ is the minimum eigenvalue of S_0. Hence $\nabla U(\Delta)^{\mathrm{T}}R(\Delta) \geq \lambda_{\min}(S_0)\|\Delta\|^2$ and we have verified Assumption C1.

Verification of Assumption C2′: Note that

$$\|R(\Delta) - S_0\Delta\| = \|\mathbb{E}\phi(\Delta^{\mathrm{T}}X|X) - \mathbb{E}\dot{\phi}(0|X)XX^{\mathrm{T}}\Delta\|.$$

By the mean-value theorem and Assumption B4, there exists δ such that, for any $\|\Delta\| \leq \delta$, $\|\mathbb{E}\{\phi(\Delta^{\mathrm{T}}X|X)X\} - \mathbb{E}\{\dot{\phi}(0|X)XX^{\mathrm{T}}\Delta\}\| \leq C_2\lambda_{\max}(S_0)\|\Delta\|^2$. This implies Assumption C2′.

Verification of Assumption C3: Let $\widehat{\Delta}_n = \widehat{\theta}_n - \theta_0$. Then $D_n = R(\widehat{\Delta}_{n-1}) - \widehat{R}(\widehat{\Delta}_{n-1}; Z_n)$. Consider decomposition $D_n = D_n(0)+E_n(\widehat{\Delta}_{n-1})$, where $D_n(0) = \psi(\varepsilon_n)X_n$ and $E_n(\widehat{\Delta}_{n-1}) = \mathbb{E}\{\psi(\widehat{\Delta}_{n-1}^{\mathrm{T}}X+\varepsilon_n)X_n\}-[\psi(\widehat{\Delta}_{n-1}^{\mathrm{T}}X+\varepsilon_n)X_n-\psi(\varepsilon_n)X_n]$. By Assumption B1 and the Cauchy-Schwartz inequality, we can show that $\mathbb{E}\left\{\|D_n\|^2 \mid \mathfrak{F}_{n-1}\right\}+\|R(\widehat{\Delta}_{n-1})\|^2 \leq 2C_1\left(1 + \|\widehat{\Delta}_{n-1}\|^2\right)$. Moreover, by Assumption B2, $D_n(0)$'s are i.i.d., so $\mathbb{E}\{D_n(0)D_n(0)^{\mathrm{T}}|\mathfrak{F}_{n-1}\} = V_0 > 0$ and $\sup_{n\geq 1}\mathbb{E}\left\{\|D_n(0)\|^2 I(\|D_n(0)\| > \eta)|\mathfrak{F}_{n-1}\right\} \to 0$, as $\eta \to \infty$. Finally, by Cauchy-Schwartz inequality and Assumptions B2-B3, we can show that $\mathbb{E}\|E_n(\Delta)\|^2 \leq \delta(\|\Delta\|)$ for some $\delta(\cdot)$ with $\delta(x) \to 0$ as $x \to 0$. This complete the verification of Assumption C3.

Obviously Assumption A5 is the same as Assumption C4′. Therefore, we have verified that Assumptions B1-B4 and A5 imply Assumptions C1, C2′, C3 and C4′. By Proposition 1, we complete the proof of Lemma 2. □

A5. Proof of Theorem 3

Proof. Recall that $R(\Delta) = \mathbb{E}\{\phi(\Delta^{\mathrm{T}}X|X)X\}$ and $\widehat{\Delta}_n = \widehat{\theta}_n - \theta_0$. Rewrite $\widehat{\theta}_n^*$ as

$$\widehat{\theta}_n^* = \widehat{\theta}_{n-1}^* + \gamma_n R(\widehat{\Delta}_n) + \gamma_n D_n^*, \tag{49}$$

where $D_n^* = W_n\psi(\widehat{\Delta}_n^{\mathrm{T}}X_n+\varepsilon_n)X_n - R(\widehat{\Delta}_n)$ is a martingale-difference process by Assumption B3 and that $\mathbb{E}\{W_n|\mathfrak{F}_{n-1}\} = 1$. Let $D_n^*(\Delta) = W_n\psi\left(\Delta^{\mathrm{T}}X_n + \varepsilon_n\right)X_n - R(\Delta)$ and $D_n^*(\Delta) = D_n^*(0) + E_n^*(\Delta)$, where

$$E_n^*(\Delta) = W_n\left[\psi(\Delta^{\mathrm{T}}X_n + \varepsilon_n) - \psi(\varepsilon_n)\right]X_n - R(\Delta). \tag{50}$$

Since $D_n^*(0) = W_n\psi(\varepsilon_n)X_n$, $\mathbb{E}\{D_n^*(0)\} = 0$ and $\mathbb{E}\{[D_n^*(0)][D_n^*(0)]^{\mathrm{T}}\} = (1 + \mathrm{Var}(W_1))V_0$ by Assumption B2. By Cauchy-Schwartz inequality and Assumptions B2-B3, we can show that $\mathbb{E}\|E_n^*(\Delta)\|^2 \leq \delta(\|\Delta\|)$ for some $\delta(\cdot)$ with $\delta(x) \to 0$ as $x \to 0$. Therefore, using the similar arguments as those in the proof of Lemma 2, we can verify that, under Assumptions B1-B4, Assumptions C1-C4 are satisfied. By Proposition 1, it follows that $\widehat{\theta}_n^* \to \theta_0$ almost surely, and

$$\sqrt{n}(\overline{\theta}_n^* - \theta_0) = \frac{1}{\sqrt{n}}S_0^{-1}\sum_{i=1}^{n} D_i^* + o_p(1). \tag{51}$$

By the decomposition of D_i^*, we have

$$\sqrt{n}(\overline{\theta}_n^* - \theta_0) = \frac{1}{\sqrt{n}}S_0^{-1}\sum_{i=1}^{n} W_i\psi(\varepsilon_i)X_i + \frac{1}{\sqrt{n}}S_0^{-1}\sum_{i=1}^{n} E_n^*(\widehat{\theta}_{n-1}^* - \theta_0) + o_p(1). \tag{52}$$

By the definition of $\delta(\|\Delta\|)$, $\mathbb{E}\{\|E_n^*(\widehat{\theta}_{n-1}^* - \Delta)\|_2^2|\mathfrak{F}_{n-1}\} = \delta(\|\widehat{\theta}_{n-1}^* - \theta_0\|)$. Since $\widehat{\theta}_n^* \to \theta_0$ a.s., we have $\delta(\|\widehat{\theta}_{n-1}^* - \theta_0\|) \to 0$ a.s.. Thus, $\sum_{i=1}^n E_n^*(\widehat{\theta}_{n-1}^* - \theta_0)/\sqrt{n} = o_p(1)$. By (52), we complete the proof of Theorem 3. □

References

Moulines, Eric and Bach, Francis R. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in Neural Information Processing Systems*, 451–459, 2011.

Chen, Xi and Lee, Jason D and Tong, Xin T and Zhang, Yichen. Statistical Inference for Model Parameters in Stochastic Gradient Descent. *arXiv preprint arXiv:1610.08637*, 2016.

Efron, Bradley. Bootstrap methods: Another look at the jackknife. *Annals of Statistics*, 7, 1-26.

Wang, Chun and Chen, Ming-Hui and Schifano, Elizabeth and Wu, Jing and Yan, Jun. Statistical methods and computing for big data. *Statistics and its interface*, 9, 399.

Hastie, Trevor and Tibshirani, Robert and Friedman, Jerome. The elements of statistical learning 2nd edition. *New York: Springer*, 2009.

Kushner, Harold and Yin, G George. Stochastic approximation and recursive algorithms and applications. *Springer Science & Business Media*, 2003.

Li, Tianyang and Liu, Liu and Kyrillidis, Anastasios and Caramanis, Constantine. Statistical methods and computing for big data. *arXiv preprint arXiv:1705.07477*, 2017.

Nelder, John A and Baker, R Jacob. Generalized linear models. *Encyclopedia of statistical sciences*, 1972.

Polyak, Boris T and Juditsky, Anatoli B. Acceleration of stochastic approximation by averaging. *SIAM Journal on Control and Optimization*, 30, 838–855.

Rao, C Radhakrishna and Zhao, LC. Approximation to the distribution of M-estimates in linear models by randomly weighted bootstrap. *Sankhyā: The Indian Journal of Statistics, Series A*, 323–331.

Robbins, Herbert and Monro, Sutton. A stochastic approximation method. *The annals of mathematical statistics*, 400–407.

Rubin, Donald B. The bayesian bootstrap. *The annals of statistics*, 9, 130–134.

Ruppert, David. Efficient estimations from a slowly convergent Robbins-Monro process. *Cornell University Operations Research and Industrial Engineering*, 1988.

Shao, Jun and Tu, Dongsheng. The jackknife and bootstrap. *Springer Science & Business Media*, 2012.

Su, Weijie and Zhu, Yuancheng. Statistical Inference for Online Learning and Stochastic Approximation via Hierarchical Incremental Gradient Descent. *arXiv preprint arXiv:1802.04876*, 2018.

Toulis, Panos and Airoldi, Edoardo M and others. Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics*, 45, 1694–1727.

Journal of Machine Learning Research 19 (2018) 1-27 Submitted 7/17; Revised 11/18; Published 11/18

A Random Matrix Analysis and Improvement of Semi-Supervised Learning for Large Dimensional Data

Xiaoyi Mai xiaoyi.mai@l2s.centralesupelec.fr
Romain Couillet romain.couillet@centralesupelec.fr
CentraleSupélec
Université Paris-Saclay
Laboratoire des Signaux et Systèmes
3 rue Joliot Curie, 91192 Gif-Sur-Yvette

Editor: Nicolas Vayatis

Abstract

This article provides an original understanding of the behavior of a class of graph-oriented semi-supervised learning algorithms in the limit of large and numerous data. It is demonstrated that the intuition at the root of these methods collapses in this limit and that, as a result, most of them become inconsistent. Corrective measures and a new data-driven parametrization scheme are proposed along with a theoretical analysis of the asymptotic performances of the resulting approach. A surprisingly close behavior between theoretical performances on Gaussian mixture models and on real data sets is also illustrated throughout the article, thereby suggesting the importance of the proposed analysis for dealing with practical data. As a result, significant performance gains are observed on practical data classification using the proposed parametrization.

Keywords: semi-supervised learning, kernel methods, random matrix theory, high dimensional statistics

1. Introduction

Semi-supervised learning consists in classification schemes combining few labelled and numerous unlabelled data. With the advent of the big-data paradigm, where supervised learning implies the impossible pre-labelling of sometimes millions of samples, these so-far marginal methods are attracting a renewed attention. Its appeal also draws on its providing an alternative to unsupervised learning which excludes the possibility to exploit known data. We refer to Chapelle et al. (2006) for an overview.

An important subset of semi-supervised learning methods concerns graph-based approaches. In these, one considers data instances $x_1, \cdots, x_n \in \mathbb{R}^p$ as vertices on a graph with edge weights W_{ij} encoding their similarity, which is usually defined through a kernel function f, as with radial kernels of the type $W_{ij} = f(\|x_i - x_j\|^2/p)$ which we shall focus on in this article. The motivation follows from one's expectation that two instances with a strong edge weight tend to belong to the same class and thus vertices of a common class tend to aggregate. Standard methods for recovering the classes of the unlabelled data then consist in various random walk (Jaakkola and Szummer, 2002) or label propagation (Zhu and Ghahramani, 2002) algorithms on the graph which softly allocate "scores" for each node to belong to a particular class. These scores are then compared for each class in order to obtain a hard decision on the individual unlabelled node class. A popular, and widely recognized as highly performing, example is the PageRank approach (Avrachenkov et al., 2011).

Many of these algorithms also have the particularity of having a closed-form and quite interrelated expression for their stationary points. These stationary points are also often found to coincide with the solutions to optimization problems under constraints, independently established. This is notably the case of Zhu et al. (2003) under equality constraints for the labelled nodes or of Belkin et al. (2004) where a relaxation approach is used instead to allow for modifications of the value of labelled nodes – this ensuring that erroneously labelled data or poorly informative labelled data do not hinder the algorithm performance. As is often the case in graph-related optimization, a proper choice of the matrix representative of the inter-data affinity is at the core of scientific research and debates and mainly defines the differences between any two schemes. In particular, Joachims et al. (2003) suggests the use of a standard Laplacian representative, where Zhou et al. (2004) advises for a normalized Laplacian approach. These individual choices correspondingly lead to different versions of the label propagation methods on the graph, as discussed in Avrachenkov et al. (2011).

There also exists another branch of manifold based semi-supervised learning (Belkin and Niyogi, 2004; Goldberg et al., 2009; Moscovich et al., 2016). In contrast to the methods discussed in this paper, these approaches involve a step of manifold learning, which plays a decisive role in the success of the learning task. While there exist many articles providing theoretical analyses for such methods (Wasserman and Lafferty, 2008; Bickel et al., 2007; Moscovich et al., 2016; Globerson et al., 2017), a comprehensive comparison to the graph-based methods presently discussed is beyond current analytical reach. This being said, while the Gaussian-mixture data model under study in the present article violates the manifold assumption, given appropriate feature (kernel function) mapping, there exists a low dimensional manifold where data demonstrate a clustering behavior, as shown by Couillet and Benaych-Georges (2015); as such, when the classes are very well separated and sufficient data are available to estimate the manifold, manifold-based methods in this setting should lead to competitive performance. While clearly out of our present scope, future investigations might allow for a comparative study of manifold versus graph approaches. Another recent line of alternative works consider SSL from a graph signal processing perspective (Narang et al., 2013a,b; Gadde et al., 2014; Anis et al., 2015), where the classification scores are viewed as smooth signals on the similarity graph and the learning task then consists in recovering a bandlimited (understood in the graph Fourier transform domain) graph signal from its known sample values.

Returning to graph-based SSL, a likely key reason for the open-ended question of a most natural choice for the graph representative arises from these methods being essentially built upon intuitive reasoning arising from low dimensional data considerations rather than from mostly inaccessible theoretical results. Indeed, the non-linear expression of the affinity matrix W as well as the rather involved form assumed by the algorithm output (although explicit) hinder the possibility to statistically evaluate the algorithm performances for all finite n, p, even for simple data assumptions. The present article is placed instead under a large dimensional data assumption, thus appropriate to the present big-data paradigm,

and proposes instead to derive, for the first time to the best of the authors' knowledge, theoretical results on the performance of the aforementioned algorithms in the large n, p limit for a certain class of statistically distributed data $x_1, \ldots, x_n \in \mathbb{R}^p$. Precisely due to the large data assumption, as we shall observe, most of the intuition leading up to the aforementioned algorithms collapse as $n, p \to \infty$ at a similar rate, and we shall prove that few algorithms remain consistent in this regime.

Specifically, recall that the idea behind graph-based semi-supervised learning is to exploit the similarity between data points and thus expect a clustering behavior of close-by data nodes. In the large data assumption (i.e., $p \gg 1$), this similarity-based approach suffers a curse of dimensionality. As the span of $\mathbb{R}^p$ grows exponentially with the data dimension p, when p is large, the data points x_i (if not too structured) are in general so sparsely distributed that their pairwise distances tend to be similar regardless of their belonging to the same class or not. The Gaussian mixture model that we define in Subsection 3 and will work on is a telling example of this phenomenon; as we show, in a regime where the classes ought to be separable (even by unsupervised methods as shown by Couillet and Benaych-Georges 2015), the normalized distance $\|x_i - x_j\|/\sqrt{p}$ of two random different data instances x_i and x_j generated from this model converges to a constant *irrespective of the class of* x_i *and* x_j in the Gaussian mixture and, consequently, the similarity defined by $W_{ij} = f(\|x_i - x_j\|^2/p)$ is asymptotically the same for all pairs of data instances. This behavior should therefore invalidate the intuition behind semi-supervised classification, hence likely render graph-based methods ineffective. As a direct consequence, the scores are *flat* in the sense that they have the same asymptotic values, irrespective of the class. Nonetheless, we will show that sensible classification on data sets generated from this model can still be achieved provided that appropriate amendments to the classification algorithms are enforced, due to the small fluctuations around these flat asymptotic limit of scores. This flat limit is reminiscent of the work by Nadler et al. (2009) where the authors show that the scores indeed share the same limit, irrespective of the class, in the presence of infinitely many unlabelled samples but for $p \geq 2$ fixed. Yet, despite the scores flatness, the authors experimentally observed non-trivial classification in binary tasks thanks to the small difference between scores; they however did not provide any theoretical support for such behavior, for they analysis failed to recover the small fluctuations.

Inspired by Avrachenkov et al. (2011), we generalize here the algorithm proposed in Zhu et al. (2003) by introducing a normalization parameter α in the cost function in order to design a large class of regularized affinity-based methods, among which are found the traditional Laplacian- and normalized Laplacian-based algorithms. The generalized optimization framework is presented in Section 2.

The main contribution of the present work is to provide a quantitative performance study of the generalized graph-based semi-supervised algorithm for large dimensional Gaussian-mixture data and radial kernels, technically following the random matrix approach developed by Couillet and Benaych-Georges (2015). Our main findings are summarized as follows:

- Irrespective of the choice of the data affinity matrix, the classification outcome is strongly biased by *the number of labelled data from each class* and unlabelled data tend to be classified into the class with most labelled nodes; we propose a normalization update of the standard algorithms to correct this limitation.

- Once the aforementioned bias corrected, the choice of the affinity matrix (and thus of the parameter α) strongly impacts the performances; most importantly, within our framework, both *standard Laplacian* ($\alpha = 0$ here) and *normalized Laplacian-based* ($\alpha = -\frac{1}{2}$) methods, although widely discussed in the literature, fail in the large dimensional data regime. Of the family of algorithms discussed above, only the *PageRank* approach ($\alpha = -1$) is shown to provide asymptotically acceptable results.

- The scores of belonging to each class attributed to individual nodes by the algorithms are shown to asymptotically follow a *Gaussian distribution* with mean and covariance depending on the statistical properties of classes, the ratio of labelled versus unlabelled data, and the value of the first derivatives of the kernel function at the limiting value τ of $\frac{1}{p}\|x_i - x_j\|^2$ (which we recall is irrespective of the genuine classes of x_i, x_j). This last finding notably allows one to *predict the asymptotic performances* of the semi-supervised learning algorithms.

- From the latter result, three main outcomes unfold:
  - when three classes or more are considered, there exist Gaussian mixture models for which classification is shown to be *impossible*;
  - despite PageRank's consistency, we further justify that the choice $\alpha = -1$ is not in general optimal. For the case of 2-class learning, we provide a method to approach the optimal value of α; this method is demonstrated on real data sets to convey sometimes *dramatic improvements* in correct classification rates.
  - for a 2-class learning task, necessary and sufficient conditions for asymptotic consistency are: $f'(\tau) < 0$, $f''(\tau) > 0$ and $f''(\tau)f(\tau) > f'(\tau)^2$; in particular, Gaussian kernels, failing to meet the last condition, cannot deal with the large dimensional version of the "concentric spheres" task.

Throughout the article, theoretical results and related discussions are confirmed and illustrated with simulations on Gaussian-mixture data as well as the popular MNIST data (LeCun et al., 1998), which serves as a comparison for our theoretical study on real world data sets. The consistent match of our theoretical findings on MNIST data, despite their departing from the very large dimensional and Gaussian-mixture assumption, suggests that our results have a certain robustness to these assumptions and can be applied to a larger range of data. We indeed believe that, while only the limiting behavior of Gaussian mixture inputs is characterized in this article (mostly for technical reasons), the analysis reveals certain properties inherent to graph-based SSL methods, which extend well beyond the Gaussian hypothesis.

Notations: $\boldsymbol{\delta}_a^b$ is a binary function taking the value of 1 if $a = b$ or that of 0 if not. 1_n is the column vector of ones of size n, I_n the $n \times n$ identity matrix. The norm $\|\cdot\|$ is the Euclidean norm for vectors and the operator norm for matrices. The operator $\mathrm{diag}(v) = \mathrm{diag}\{v_a\}_{a=1}^k$ is the diagonal matrix having $v_1, \ldots, v_k$ as its ordered diagonal elements. $O(\cdot)$ is the same as specified in the work of Couillet and Benaych-Georges (2015): for a random

variable $x \equiv x_n$ and $u_n \geq 0$, we write $x = O(u_n)$ if for any $\eta > 0$ and $D > 0$, we have $n^D \mathrm{P}(x \geq n^\eta u_n) \to 0$. When multidimensional objects are concerned, for a vector (or a diagonal matrix)v, $v = O(u_n)$ means the maximum entry in absolute value is $O(u_n)$ and for a square matrix M, $M = O(u_n)$ means that the operator norm of M is $O(u_n)$.

2. Optimization Framework

Let $x_1, \ldots, x_n \in \mathbb{R}^p$ be n data vectors belonging to K classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$. The class association of the $n_{[l]}$ vectors $x_1, \ldots, x_{n_{[l]}}$ is known (these vectors will be referred to as *labelled*), while the class of the remaining $n_{[u]}$ vectors $x_{n_{[l]}+1}, \ldots, x_n$ ($n_{[l]} + n_{[u]} = n$) is unknown (these are referred to as *unlabelled* vectors). Within both labelled and unlabelled subsets, the data are organized in such a way that the $n_{[l]1}$ first vectors $x_1, \ldots, x_{n_{[l]1}}$ belong to class $\mathcal{C}_1$, $n_{[l]2}$ subsequent vectors to $\mathcal{C}_2$, and so on, and similarly for the $n_{[u]1}, n_{[u]2}, \ldots$ first vectors of the set $x_{n_{[l]}+1}, \ldots, x_n$. Note already that this ordering is for notational convenience and shall not impact the generality of our results.

The affinity relation between the vectors $x_1, \ldots, x_n$ is measured from the weight matrix W defined by

$$W \equiv \left\{ f\left(\frac{1}{p} \|x_i - x_j\|^2 \right) \right\}_{i,j=1}^n$$

for some function f. The matrix W may be seen as the adjacency matrix of the n-node graph indexed by the vectors $x_1, \ldots, x_n$. We further denote by D the diagonal matrix with $D_{ii} \equiv d_i = \sum_{j=1}^n W_{ij}$ the degree of the node associated to x_i.

We next define a score matrix $F \in \mathbb{R}^{n \times K}$ with F_{ik} representing the evaluated score for x_i to belong to $\mathcal{C}_k$. In particular, following the conventions typically used in graph-based semi-supervised learning (Chapelle et al., 2006), we shall affect a unit score $F_{ik} = 1$ if x_i is a labelled data of class $\mathcal{C}_k$ and a null score for all $F_{ik'}$ with $k' \neq k$. In order to attribute classes to the unlabelled data, scores are first affected by means of the resolution of an optimization framework. We propose here

$$\begin{aligned} F &= \mathrm{argmin}_{F \in \mathbb{R}^{n \times k}} \sum_{k=1}^K \sum_{i,j=1}^n W_{ij} \left\| d_i^\alpha F_{ik} - d_j^\alpha F_{jk} \right\|^2 \\ \text{s.t. } F_{ik} &= \begin{cases} 1, & \text{if } x_i \in \mathcal{C}_k \\ 0, & \text{otherwise} \end{cases}, \; 1 \leq i \leq n_{[l]}, \; 1 \leq k \leq K \end{aligned} \tag{1}$$

where $\alpha \in \mathbb{R}$ is a given parameter. The interest of this generic formulation is that it coincides with the standard Laplacian-based approach for $\alpha = 0$ and with the normalized Laplacian-based approach for $\alpha = -\frac{1}{2}$, both discussed in Section 1. Note importantly that Equation (1) is naturally motivated by the observation that large values of W_{ij} enforce close values for F_{ik} and F_{jk} while small values for W_{ij} allow for more freedom in the choice of F_{ik} and F_{jk}.

By denoting

$$F = \begin{bmatrix} F_{[l]} \\ F_{[u]} \end{bmatrix}, \; W = \begin{bmatrix} W_{[ll]} & W_{[lu]} \\ W_{[ul]} & W_{[uu]} \end{bmatrix}, \text{ and } D = \begin{bmatrix} D_{[l]} & 0 \\ 0 & D_{[u]} \end{bmatrix}$$

with $F_{[l]} \in \mathbb{R}^{n_{[l]}}$, $W_{[ll]} \in \mathbb{R}^{n_{[l]} \times n_{[l]}}$, $D_{[l]} \in \mathbb{R}^{n_{[l]} \times n_{[l]}}$, one easily finds (since the problem is a convex quadratic optimization with linear equality constraints) the solution to (1) is explicitly given by

$$F_{[u]} = \left(I_{n_u} - D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha \right)^{-1} D_{[u]}^{-1-\alpha} W_{[ul]} D_{[l]}^\alpha F_{[l]}. \tag{2}$$

Once these scores are affected, a mere comparison between all scores $F_{i1}, \ldots, F_{iK}$ for unlabelled data x_i (i.e., for $i > n_{[l]}$) is performed to decide on its class, i.e., the allocated class index $\hat{\mathcal{C}}_{x_i}$ for vector x_i is given by

$$\hat{\mathcal{C}}_{x_i} = \mathcal{C}_{\hat{k}} \text{ for } \hat{k} = \mathrm{argmax}_{1 \leq k \leq K} F_{ik}.$$

Note in passing that the formulation (2) implies in particular that

$$F_{[u]} = D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha F_{[u]} + D_{[u]}^{-1-\alpha} W_{[ul]} D_{[l]}^\alpha F_{[l]} \tag{3}$$

$$F_{[l]} = \{ \boldsymbol{\delta}_{x_i \in \mathcal{C}_k} \}_{\substack{1 \leq i \leq n_{[l]} \\ 1 \leq k \leq K}} \tag{4}$$

and thus the matrix F is a stationary point for the algorithm constituted of the updating rules (3) and (4) (when replacing the equal signs by affectations). In particular, for $\alpha = -1$, the algorithm corresponds to the standard label propagation method found in the PageRank algorithm for semi-supervised learning as discussed in Avrachenkov et al. (2011), with the major difference that $F_{[l]}$ is systematically reset to its known value while in the study of Avrachenkov et al. (2011), $F_{[l]}$ is allowed to evolve (for reasons related to robustness to pre-labeling errors).

The technical objective of the article is to analyze the behavior of $F_{[u]}$ in the large n, p regime for a Gaussian mixture model for the data $x_1, \ldots, x_n$. To this end, we shall first need to design appropriate growth rate conditions for the Gaussian mixture statistics as $p \to \infty$ (in order to avoid trivializing the classification problem as p grows large) before proceeding to the evaluation of the behavior of W, D, and thus F.

3. Model and Theoretical Results

3.1. Model and Assumptions

In the remainder of the article, we shall assume that the data $x_1, \ldots, x_n$ are extracted from a Gaussian mixture model composed of K classes. Specifically, for $k \in \{1, \ldots, K\}$,

$$x_i \in \mathcal{C}_k \Leftrightarrow x_i \sim \mathcal{N}(\mu_k, C_k).$$

Consistently with the previous section, for each k, there are n_k instances of vectors of class $\mathcal{C}_k$, among which $n_{[l]k}$ are labelled and $n_{[u]k}$ are unlabelled.

As pointed out above, in the regime where $n, p \to \infty$, special care must be taken to ensure that the classes $\mathcal{C}_1, \ldots, \mathcal{C}_K$, the statistics of which evolve with p, remain at a "somewhat constant" distance from each other. This is to ensure that the classification problem does not become asymptotically infeasible nor trivially simple as $p \to \infty$. Based on the earlier work (Couillet and Benaych-Georges, 2015) where similar considerations were made, the behavior of the class means, covariances, and cardinalities will follow the prescription below:

Assumption 1 (Growth Rate) *As* $n \to \infty$, $\frac{p}{n} \to c_0 > 0$ *and* $\frac{n_{[l]}}{n} \to c_{[l]} > 0$, $\frac{n_{[u]}}{n} \to c_{[u]} > 0$. *For each* k, $\frac{n_k}{n} \to c_k > 0$, $\frac{n_{[l]k}}{n} \to c_{[l]k} > 0$, $\frac{n_{[u]k}}{n} \to c_{[u]k} > 0$. *Besides,*

1. For $\mu^\circ \triangleq \sum_{k=1}^K \frac{n_k}{n}\mu_k$ and $\mu_k^\circ \triangleq \mu_k - \mu^\circ$, $\|\mu_k^\circ\| = O(1)$.
2. For $C^\circ \triangleq \sum_{k=1}^K \frac{n_k}{n} C_k$ and $C_k^\circ \triangleq C_k - C^\circ$, $\|C_k\| = O(1)$ and $\mathrm{tr} C_k^\circ = O(\sqrt{p})$.
3. As $n \to \infty$, $\frac{2}{p}\mathrm{tr} C^\circ \to \tau \neq 0$.
4. As $n \to \infty$, $\alpha = O(1)$.

It will also be convenient in the following to define

$$t_k \equiv \frac{1}{\sqrt{p}} \mathrm{tr}\, C_k^\circ$$
$$T_{kk'} \equiv \frac{1}{p} \mathrm{tr}\, C_k C_{k'}$$

as well as the labelled-data centered notations

$$\tilde{\mu}_k \equiv \mu_k - \sum_{k'=1}^K \frac{n_{[l]k'}}{n_{[l]}} \mu_{k'}$$
$$\tilde{C}_k \equiv C_k - \sum_{k'=1}^K \frac{n_{[l]k'}}{n_{[l]}} C_{k'}$$
$$\tilde{t}_k \equiv \frac{1}{\sqrt{p}} \mathrm{tr}\, \tilde{C}_k$$
$$\tilde{T}_{kk'} \equiv \frac{1}{p} \mathrm{tr}\, \tilde{C}_k \tilde{C}_{k'}.$$

A few comments on Assumption 1 are in order. First note that, unlike in the previous works (Nadler et al., 2009; Globerson et al., 2017) where the number of labelled data $n_{[l]}$ and data dimension p are considered fixed and the number of unlabelled data $n_{[u]}$ is supposed to be infinite, we assume a regime where $n_{[l]}, n_{[u]}$ and p are simultaneously large. Letting p large allows us to investigate SSL in the context of large dimensional data. Further imposing that $n_{[l]}, n_{[u]}$ grow at a controlled rate with respect to p (here at the same rate) allows for an *exact characterization* of the limiting SSL performances, as a function of the hyperparameters α, f and data statistics μ_k, C_k, in non-trivial classification scenarios (i.e., when classification is neither asymptotically perfect nor impossible), instead of solely retrieving consistency bounds as a function of growth rates in $p, n_{[l]}, n_{[u]}$. This in turn allows for possible means of precise parameter setting to reach optimal performances (which is not possible with results based on bounds). While it may be claimed that SSL in practice often handles scenarios where $n_{[u]} \gg n_{[l]}$, assuming that $n_{[u]}, n_{[l]}$ are of the same order but that $n_{[u]}$ is multiple times $n_{[l]}$ actually maintains the validity of our results so long that $n_{[l]}$ is not too small. To be more exact, our results are still valid in the limit where $c_{[l]} \to 0$, but then become trivial, as numerically confirmed by Figure 5. To consider the setting where

$n_{[l]}$ is fixed while $p, n_{[u]}$ grow large would demand a change in the statistical assumptions of the input data sets, which goes beyond the scope of the present investigation.

Item 3. of Assumption 1 is mostly a technical convenience that shall simplify our analysis, but our results naturally extend as long as both liminf and limsup of $\frac{2}{p}\mathrm{tr} C^\circ$ are away from zero or infinity. The necessity of Item 1. only appears through a detailed analysis of spectral properties of the weight matrix W for large n, p, carried out later in the article. As for Item 2., note that if $\mathrm{tr}\, C_k^\circ = O(\sqrt{p})$ were to be relaxed, it is easily seen that a mere (unsupervised) comparison of the values of $\|x_i\|^2$ would asymptotically provide an almost surely perfect classification.

As a by-product of imposing the growth constraints on the data to ensure non-trivial classification, Assumption 1 induces the following seemingly unsettling implication, easily justified by a simple concentration of measure argument

$$\max_{1 \leq i,j \leq n} \left| \frac{1}{p}\|x_i - x_j\|^2 - \tau \right| \xrightarrow{\text{a.s.}} 0 \tag{5}$$

as $p \to \infty$. Equation (5) is the cornerstone of our analysis and states that all vector pairs x_i, x_j are essentially at the same distance from one another as p gets large, *irrespective of their classes*. This striking result evidently is in sharp opposition to the very motivation for the optimization formulation (1) as discussed in the introduction. It thus immediately entails that the solution (2) to (1) is bound to produce asymptotically inconsistent results. We shall see that this is indeed the case for all but a short range of values of α.

This being said, Equation (5) has an advantageous side as it allows for a Taylor expansion of $W_{ij} = f(\frac{1}{p}\|x_i - x_j\|^2)$ around $f(\tau)$, provided f is sufficiently smooth around τ, which is ensured by our subsequent assumption.

Assumption 2 (Kernel function) *The function* $f : \mathbb{R}^+ \to \mathbb{R}$ *is three-times continuously differentiable in a neighborhood of* τ.

Note that Assumption 2 does not constrain f aside from its local behavior around τ. In particular, we shall not restrict ourselves to matrices W arising from nonnegative definite kernels as standard machine learning theory would advise (Schölkopf and Smola, 2002).

The core technical part of the article now consists in expanding W, and subsequently all terms intervening in (2), in a Taylor expansion of successive matrices of *non-vanishing operator norm*. Note indeed that the magnitude of the individual entries in the Taylor expansion of W needs not follow the magnitude of the operator norm of the resulting matrices;[1] rather, great care must be taken to only retain those matrices of non-vanishing operator norm. These technical details call for advanced random matrix considerations and are discussed in the appendix and in Couillet and Benaych-Georges (2015).

We are now in position to introduce our main technical results.

3.2. Main Theoretical Results

In the course of this section, we provide in parallel a series of technical results under the proposed setting (notably under Assumption 1) along with simulation results both on a

1. For instance, $\|I_n\| = 1$ while $\|1_n 1_n^\mathrm{T}\| = n$ despite both matrices having entries of similar magnitude.

2-class Gaussian mixture data model with $\mu_1 = [4; 0_{p-1}]$, $\mu_2 = [0; 4; 0_{p-2}]$, $C_1 = I_p$ and $\{C_2\}_{i,j} = .4^{|i-j|}(1 + \frac{3}{\sqrt{p}})$, as well as on real data sets, here images of eights and nines from the MNIST database (LeCun et al., 1998), for $f(t) = \exp(-\frac{1}{2}t)$, i.e., the classical Gaussian (or heat) kernel. For reasons that shall become clear in the following discussion, these figures will depict the (size n) vectors

$$\left[F^\circ_{[u]}\right]_{\cdot k} \equiv \left[F_{[u]}\right]_{\cdot k} - \frac{1}{K}\sum_{k'=1}^{K}\left[F_{[u]}\right]_{\cdot k'}$$

for $k \in \{1, 2\}$. Obviously, the decision rule on $F^\circ_{[u]}$ is the same as that on $F_{[u]}$.

Our first hinging result concerns the behavior of the score matrix F in the large n, p regime, as per Assumption 1, and reads as follows.

Proposition 1 *Let Assumptions 1–2 hold. Then, for $i > n_{[l]}$ (i.e., for x_i an unlabelled vector),*

$$F_{ik} = \frac{n_{[l]k}}{n}\left[1 + \underbrace{(1+\alpha)\frac{f'(\tau)}{f(\tau)}\frac{t_k}{\sqrt{p}} + z_i}_{O(n^{-\frac{1}{2}})} + O(n^{-1})\right] \tag{6}$$

where $z_i = O(n^{-\frac{1}{2}})$ is a random variable, function of x_i, but independent of k.

The proof of Proposition 1 is given as an intermediary result of the proof of Theorem 5 in the appendix.

Proposition 1 provides a clear overview of the outcome of the semi-supervised learning algorithm. First note that $F_{ik} = c_{[l]k} + O(n^{-\frac{1}{2}})$. Therefore, irrespective of x_i, F_{ik} is strongly biased towards $c_{[l]k}$. If the values $n_{[l]1}, \ldots, n_{[l]k}$ differ by $O(n)$, this induces a systematic asymptotic allocation of every x_i to the class having largest $c_{[l]k}$ value. Figure 1 illustrates this phenomenon, observed both on synthetic and real data sets, here for $n_{[l]1} = 3n_{[l]2}$.

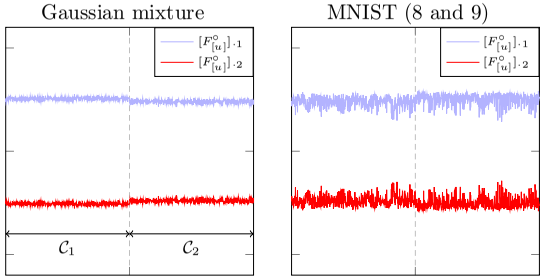


Figure 1: $[F^\circ_{[u]}]_{\cdot 1}$ and $[F^\circ_{[u]}]_{\cdot 2}$ for 2-class data, $n = 1024$, $p = 784$, $n_l/n = 1/16$, $n_{[u]1} = n_{[u]2}$, $n_{[l]1} = 3n_{[l]2}$, $\alpha = -1$, Gaussian kernel.

Pursuing the analysis of Proposition 1 by now assuming that $n_{[l]1} = \ldots = n_{[l]K}$, the comparison between $F_{i1}, \ldots, F_{iK}$ next revolves around the term of order $O(n^{-\frac{1}{2}})$. Since z_i only depends on x_i and not on k, it induces a constant offset to the vector $F_{i\cdot}$, thereby not intervening in the class allocation. On the opposite, the term t_k is independent of x_i but may vary with k, thereby possibly intervening in the class allocation, again an undesired effect. Figure 2 depicts the effect of various choices of α for equal values of $n_{[l]k}$. This deleterious outcome can be avoided either by letting $f'(\tau) = O(n^{-\frac{1}{2}})$ or $\alpha = -1 + O(n^{-\frac{1}{2}})$. But, as discussed in the study of Couillet and Benaych-Georges (2015) and later in the article, the choice of f such that $f'(\tau) \simeq 0$, if sometimes of interest, is generally inappropriate.

The discussion above thus induces two important consequences to adapt the semi-supervised learning algorithm to large data.

1. The final comparison step *must* be made upon the normalized scores

$$\hat{F}_{ik} \equiv \frac{n}{n_{[l]k}}F_{ik} \tag{7}$$

   rather than upon the scores F_{ik} directly.

2. The parameter α *must* be chosen in such a way that

$$\alpha = -1 + O(n^{-\frac{1}{2}}).$$

Under these two amendments of the algorithm, according to Proposition 1, the performance of the semi-supervised learning algorithm now relies upon terms of magnitude $O(n^{-1})$, which are so far left undefined. A thorough analysis of these terms allows for a complete understanding of the asymptotic behavior of the normalized scores $\hat{F}_{i\cdot} = (\hat{F}_{i1}, \ldots, \hat{F}_{iK})$, as presented in our next result.

Theorem 2 *Let Assumptions 1–2 hold. For $i > n_{[l]}$ (i.e., x_i unlabelled) with $x_i \in \mathcal{C}_b$, let $\hat{F}_{ia}$ be given by (7) with F defined in (2) and $\alpha = -1 + \frac{\beta}{\sqrt{p}}$ for $\beta = O(1)$. Then,*

$$p\hat{F}_{i\cdot} = p(1 + z_i)1_K + G_i + o_P(1) \tag{8}$$

where $z_i = O(\sqrt{p})$ is as in Proposition 1 and $G_i \sim \mathcal{N}(m_b, \Sigma_b)$, $i > n_{[l]}$, are independent with

$$[m_b]_a = -\frac{2f'(\tau)}{f(\tau)}\tilde{\mu}_a^\mathsf{T}\tilde{\mu}_b + \left(\frac{f''(\tau)}{f(\tau)} - \frac{f'(\tau)^2}{f(\tau)^2}\right)\tilde{t}_a\tilde{t}_b + \frac{2f''(\tau)}{f(\tau)}\tilde{T}_{ab} + \frac{\beta}{c_{[l]}}\frac{f'(\tau)}{f(\tau)}t_a \tag{9}$$

$$[\Sigma_b]_{a_1a_2} = 2\left(\frac{f''(\tau)}{f(\tau)} - \frac{f'(\tau)^2}{f(\tau)^2}\right)^2 T_{bb}t_{a_1}t_{a_2} + 4\frac{f'(\tau)^2}{f(\tau)^2}\left[\mu_{a_1}^{\circ\mathsf{T}}C_b\mu_{a_2}^\circ + \boldsymbol{\delta}_{a_1a_2}\frac{c_0T_{b,a_1}}{c_{[l]}c_{[l]a_1}}\right]. \tag{10}$$

Besides, there exists $\mathcal{A} \subset \sigma(\{\{x_1, \ldots, x_{n_{[l]}}\}, p = 1, 2, \ldots\})$ (the σ-field induced by the labelled variables) with $P(\mathcal{A}) = 1$ over which (8) also holds conditionally to $\{\{x_1, \ldots, x_{n_{[l]}}\}, p = 1, 2, \ldots\}$.

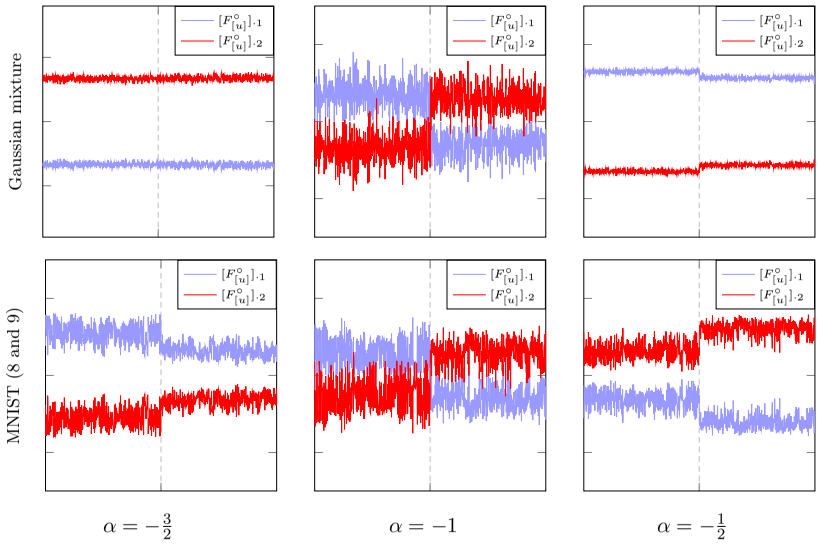


Figure 2: $[F^\circ_{[u]}]_{\cdot 1}$, $[F^\circ_{[u]}]_{\cdot 2}$ for 2-class data, $n = 1024$, $p = 784$, $n_l/n = 1/16$, $n_{[u]1} = n_{[u]2}$, $n_{[l]1} = n_{[l]2}$, Gaussian kernel.

Note that the statistics of G_i are independent of the realization of $x_1, \ldots, x_{[l]}$ when $\alpha = -1 + O(\frac{1}{\sqrt{p}})$. This in fact no longer holds when α is outside this regime, as pointed out by Theorem 5 in the appendix which provides the asymptotic behavior of $\hat{F}_{i\cdot}$ for all values of α (and thus generalizes Theorem 2).

Since the ordering of the entries of $\hat{F}_{i\cdot}$ is the same as that of $\hat{F}_{i\cdot} - (1 + z_i)$, Theorem 2 amounts to saying that the probability of correctly classifying unlabeled vectors x_i genuinely belonging to class $\mathcal{C}_b$ is asymptotically given by the probability of $[G_i]_b$ being the maximal element of G_i, which, as mentioned above, is the same whether conditioned or not on $x_1, \ldots, x_{[l]}$ for $\alpha = -1 + O(\frac{1}{\sqrt{p}})$. This is formulated in the following corollary.

Corollary 3 *Let Assumptions 1–2 hold. Let $i > n_{[l]}$ and $\alpha = -1 + \frac{\beta}{\sqrt{p}}$. Then, under the notations of Theorem 2,*

$$\mathbb{P}\left(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b, x_1, \cdots, x_{n_{[l]}}\right) - \mathbb{P}\left(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b\right) \to 0$$
$$\mathbb{P}\left(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b\right) - \mathbb{P}\left([G_i]_b > \max_{a \neq b} \{[G_i]_a\} | x_i \in \mathcal{C}_b\right) \to 0.$$

In particular, for $K = 2$, and $a \neq b \in \{1, 2\}$,

$$\mathbb{P}\left([G_i]_b > \max_{a \neq b} \{[G_i]_a\} | x_i \in \mathcal{C}_b\right) = \Phi(\theta_b^a), \quad \text{with } \theta_b^a \equiv \frac{[m_b]_b - [m_b]_a}{\sqrt{[\Sigma_b]_{bb} + [\Sigma_b]_{aa} - 2[\Sigma_b]_{ab}}}$$

where $\Phi(u) = \frac{1}{2\pi} \int_{-\infty}^{u} e^{-\frac{t^2}{2}} dt$ is the Gaussian distribution function.

With G_i being independent, Corollary 3 allows us to approach the empirical classification accuracy as it is consistently estimated by the probability of correct classification given in the corollary. As with Theorem 2 which can be appended to Theorem 5 for a large set of values of α, Corollary 3 is similarly generalized by Corollary 6 in the appendix. Using both corollaries, Figure 3 displays a comparison between simulated accuracies from various pairs of digits from the MNIST data against our theoretical results; to apply our results, a 2-class Gaussian mixture model is assumed with means and covariances equal to the empirical means and covariances of the individual digits, evaluated from the full 60 000-image MNIST database. It is quite interesting to observe that, despite the obvious inadequacy of a Gaussian mixture model for this image database, the theoretical predictions are in strong agreement with the practical performances. Also surprising is the strong adequacy of the theoretical prediction of Corollary 3 beyond the range of values of α in the neighborhood of -1.

4. Consequences

4.1. Semi-Supervised Learning beyond Two Classes

An immediate consequence of Corollary 3 is that, for $K > 2$, there exists a Gaussian mixture model for which the semi-supervised learning algorithms under study necessarily fail to classify at least one class. To see this, we consider $K = 3$ and let $\mu_3 = 3\mu_2 = 6\mu_1$, $C_1 = C_2 = C_3$, $n_1 = n_2 = n_3$, $n_{[l]1} = n_{[l]2} = n_{[l]3}$. First, it follows from Corollary 3 that,

$$\mathbb{P}(x_i \to \mathcal{C}_2 | x_i \in \mathcal{C}_2) \leq \mathbb{P}([G_i]_2 > [G_i]_1 | x_i \in \mathcal{C}_2) + o(1) = \Phi(\theta_2^1) + o(1)$$
$$\mathbb{P}(x_i \to \mathcal{C}_3 | x_i \in \mathcal{C}_3) \leq \mathbb{P}([G_i]_3 > [G_i]_1 | x_i \in \mathcal{C}_3) + o(1) = \Phi(\theta_3^1) + o(1)$$

Then, under Assumptions 1–2 and the notations of Corollary 3,

$$\theta_2^1 = \text{sgn}(f'(\tau)) \frac{\mu_1^2}{\sqrt{(\Sigma_2)_{22} + (\Sigma_2)_{11} - 2(\Sigma_2)_{12}}}$$
$$\theta_3^1 = -\text{sgn}(f'(\tau)) \frac{15\mu_1^2}{\sqrt{(\Sigma_3)_{33} + (\Sigma_3)_{11} - 2(\Sigma_3)_{13}}}$$

so that $f'(\tau) < 0 \Rightarrow \theta_2^1 < 0$, $f'(\tau) > 0 \Rightarrow \theta_3^1 < 0$, while $f'(\tau) = 0 \Rightarrow \theta_2^1 = \theta_3^1 = 0$. As such, the correct classification rate of elements of $\mathcal{C}_2$ and $\mathcal{C}_3$ cannot be simultaneously greater than $\frac{1}{2}$, leading to necessarily inconsistent classifications.

It is nonetheless easy to check that this kind of inconsistency cannot occur if μ_1, μ_2 and μ_3 are mutually orthogonal (which is often bound to occur with large dimensional data). Indeed, note that all first three terms at the right-hand side of (9) can be viewed as products of

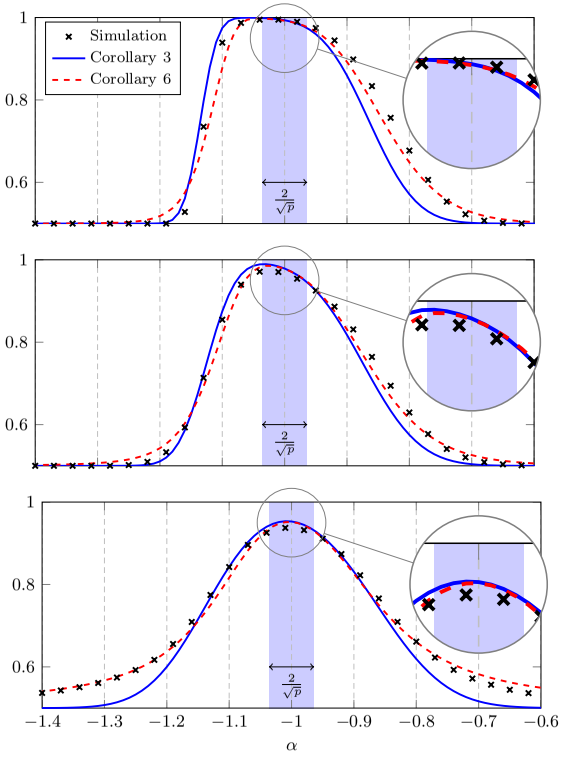


Figure 3: Theoretical and empirical accuracy as a function of α for 2-class MNIST data (top: digits (0,1), middle: digits (1,7), bottom: digits (8,9)), $n = 1024$, $p = 784$, $n_{[l]}/n = 1/16$, $n_{[u]1} = n_{[u]2}$, Gaussian kernel. Averaged over 50 iterations.

some centered vectors $\tilde{v}_k = v_k - \sum_{k'=1}^K \gamma_{k'} v_{k'}$ where $\sum_{k'=1}^K \gamma_{k'} = 1$.[2] Inconsistency occurs to class k if there exist $a, b \neq k$ such that $\tilde{v}_k^T \tilde{v}_b > \tilde{v}_k^T \tilde{v}_k > \tilde{v}_k^T \tilde{v}_a$. To better understand the cause of this inconsistency, let us consider two extreme scenarios: (i) the v_k differ by 'intensity', i.e., $v_k = r_k v$ for $k \in \{1, \cdots, K\}$, or (ii) the v_k differ by 'direction', i.e, $v_k = v + u_k$ with orthogonal u_k's. In scenario (i), let $s_{\min} = \operatorname{argmin}_{k \in \{1,\cdots,K\}} r_k$ and $s_{\max} = \operatorname{argmax}_{k \in \{1,\cdots,K\}} r_k$; then, for $k \neq \{s_{\min}, s_{\max}\}$, $\min\{\tilde{v}_k^T \tilde{v}_{s_{\min}}, \tilde{v}_k^T \tilde{v}_{s_{\max}}\} < \tilde{v}_k^T \tilde{v}_k < \max\{\tilde{v}_k^T \tilde{v}_{s_{\min}}, \tilde{v}_k^T \tilde{v}_{s_{\max}}\}$ and inconsistency is thus observed for classes $k \neq \{s_{\min}, s_{\max}\}$. Contrarily, in scenario (ii), for all

2. The third term of (9) can be seen in this way since for any two symmetric matrices $A = \{a_{ij}\}_{i,j=1}^m$ and $B = \{b_{ij}\}_{i,j=1}^m$ of same dimensions, $\operatorname{tr} AB = \sum_{i,j}^m a_{ij} b_{ij} = a_v^T b_v$ with $a_v = [a_{11}, \cdots, a_{1m}, \cdots, a_{m1}, \cdots, a_{mm}]$, $b_v = [b_{11}, \cdots, b_{1m}, \cdots, b_{m1}, \cdots, b_{mm}]$.

$k \neq k' \in \{1, \cdots, K\}$, $\tilde{v}_k^T \tilde{v}_k \geq \tilde{v}_k^T \tilde{v}_{k'}$ since $\tilde{v}_k^T \tilde{v}_k \geq 0$ and $\tilde{v}_k^T \tilde{v}_{k'} \leq 0$. As such, inconsistency is less likely to occur if the v_k's have very different directions.

4.2. Choice of f and Suboptimality of the Heat Kernel

As a consequence of the previous section, we shall from here on concentrate on the semi-supervised classification of $K = 2$ classes. In this case, it is easily seen that,

$$(K = 2) \quad \forall a \neq b \in \{1, 2\}, \ \|\tilde{\mu}_b\|^2 \geq \tilde{\mu}_b^{\mathsf{T}} \tilde{\mu}_a, \quad \tilde{t}_b^2 \geq \tilde{t}_a \tilde{t}_b, \quad \tilde{T}_{bb} \geq \tilde{T}_{ab}$$

with equalities respectively for $\mu_a = \mu_b$, $t_a = t_b$, and $\operatorname{tr} C_a C_b = \operatorname{tr} C_b^2$. This result, along with Corollary 3, implies the necessity of the conditions

$$f'(\tau) < 0, \quad f''(\tau) f(\tau) > f'(\tau)^2, \quad f''(\tau) > 0$$

to fully discriminate Gaussian mixtures. As such, from Corollary 3, by letting $\alpha = -1$, semi-supervised classification of $K = 2$ classes is always consistent under these conditions.

Since only the first three derivatives of f are involved, one may design a simple kernel for any desired values of $f'(\tau)$, $f''(\tau) f(\tau) - f'(\tau)^2$ and $f''(\tau)$ with a second degree polynomial $f(t) = at^2 + bx + c$ in such a way that $a\tau + b = f'(\tau)$, $a(a\tau^2 + b\tau + c) - (a\tau + b)^2 = f''(\tau) f(\tau) - f'(\tau)^2$ and $a = f''(\tau)$, i.e.,

$$a = f''(\tau) \quad b = f'(\tau) - f''(\tau)\tau \quad c = (f''(\tau) f(\tau) - f'(\tau)\tau^2)/f''(\tau).$$

Since τ can be consistently estimated in practice by (11) (see the discussion in Subsection 5.1), so can a, b, and c.

A quite surprising outcome of the necessary conditions on the derivatives of f is that the widely used Gaussian (or heat) kernel $f(t) = \exp(-\frac{t}{2\sigma^2})$, while fulfilling the condition $f'(t) < 0$ and $f''(t) > 0$ for all t (and thus $f'(\tau) < 0$ and $f''(\tau) > 0$), only satisfies $f''(t) f(t) = f'(t)^2$. This indicates that discrimination over $t_1, \ldots, t_K$, under the conditions of Assumption 1, is asymptotically *not* possible with a Gaussian kernel. This remark is illustrated in Figure 4 for a discriminative task between two centered isotropic Gaussian classes only differing by the trace of their covariance matrices. There, irrespective of the choice of the bandwidth σ, the Gaussian kernel leads to a constant $1/2$ accuracy, where a mere second order polynomial kernel selected upon its derivatives at τ demonstrates good performances. Since p-dimensional isotropic Gaussian vectors tend to concentrate "close to" the surface of a sphere, this thus suggests that Gaussian kernels are not inappropriate to solve the large dimensional generalization of the "concentric spheres" task (for which they are very efficient in small dimensions). In passing, the right-hand side of Figure 4 confirms the need for $f''(\tau) f(\tau) - f'(\tau)^2$ to be positive (there $|f'(\tau)| < 1$) as an accuracy lower than $1/2$ is obtained for $f''(\tau) f(\tau) - f'(\tau)^2 < 0$.

Another interesting fact lies in the choice $f'(\tau) = 0$ (while $f''(\tau) \neq 0$). As already identified by Couillet and Benaych-Georges (2015) and further thoroughly investigated by Couillet and Kammoun (2016), if $t_1 = t_2$ (which can be enforced by normalizing the data set) and $\tilde{T}_{bb} > \tilde{T}_{ba}$ for all $a \neq b \in \{1, 2\}$, then $\Sigma_b = 0$ while $[m_b]_b > [m_b]_a$ for all $b \neq a \in \{1, 2\}$ and thus leading asymptotically to a perfect classification. As such, while Assumption 1 was

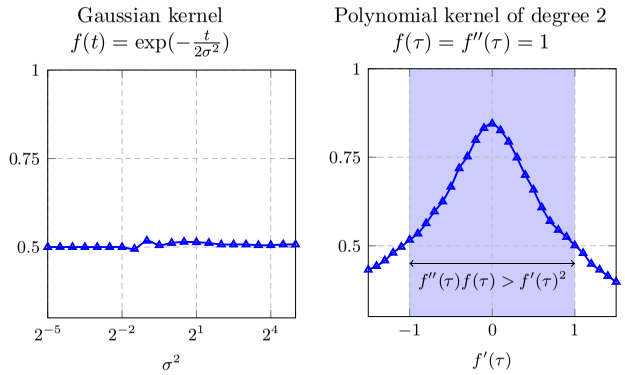


Figure 4: Empirical accuracy for 2-class Gaussian data with $\mu_1 = \mu_2$, $C_1 = I_p$ and $C_2 = (1 + \frac{3}{\sqrt{p}})I_p$, $n = 1024$, $p = 784$, $n_l/n = 1/16$, $n_{[u]1} = n_{[u]2}$, $n_{[l]1} = n_{[l]2}$, $\alpha = -1$.

claimed to ensure a "non-trivial" growth rate regime, asymptotically perfect classification may be achieved by choosing f such that $f'(\tau) = 0$, under the aforementioned statistical conditions. One must nonetheless be careful that this *asymptotic result* does not necessarily entail outstanding performances in practical finite dimensional scenarios. Indeed, note that taking $f'(\tau) = 0$ discards the visibility of differing means $\mu_1 \neq \mu_2$ (from the expression of $[m_b]_a$ in Theorem 2); for finite n, p, cancelling the differences in means (often larger than differences in covariances) may not be compensated for by the reduction in variance. Trials on MNIST particularly emphasize this remark.

4.3. Impact of Class Sizes

A final remark concerns the impact of $c_{[l]}$ and c_0 on the asymptotic performances. Note that $c_{[l]}$ and c_0 only act upon the covariance Σ_b and precisely on its diagonal elements. Both a reduction in c_0 (by increasing n) and an increase in $c_{[l]}$ reduce the diagonal terms in the variance, thereby mechanically increasing the classification performances (if in addition $[m_b]_b > [m_b]_a$ for $a \neq b$). In the opposite case of few labeled data, i.e., $c_{[l]} \to 0$, the variance diverges and the performance tends to that of random classification, as shown in Figure 5.

5. Parameter Optimization in Practice

5.1. Estimation of τ

In previous sections, we have emphasized the importance of selecting the kernel function f so as to meet specific conditions on its derivatives at the quantity τ. In practice however, τ is an unknown quantity. A mere concentration of measure argument nonetheless shows

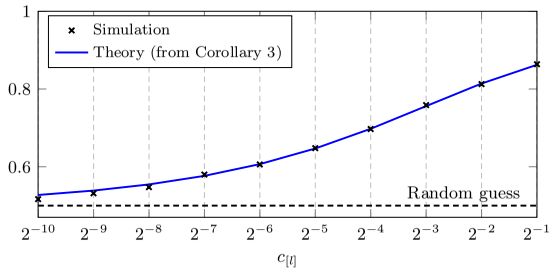


Figure 5: Theoretical and empirical accuracy as a function of $c_{[l]}$ for 2-class Gaussian data with $\mu_2 = 2\mu_1 = [6, 0, \cdots, 0]$, $C_1 = I_p$ and $\{C_2\}_{i,j} = .4^{|i-j|}$, $p = 2048$, $c_0 = 1$, $n_{[u]1} = n_{[u]2}$, $n_{[l]1} = n_{[l]2}$, $\alpha = -1$, Gaussian kernel. Averaged over 50 iterations.

that

$$\hat{\tau} \equiv \frac{1}{n(n-1)} \sum_{i,j=1}^{n} \frac{1}{p} \|x_i - x_j\|^2 \xrightarrow{\text{a.s.}} \tau. \tag{11}$$

As a consequence, the results of Theorem 2 and subsequently of Corollary 3 hold verbatim with $\hat{\tau}$ in place of τ. One thus only needs to design f in such as way that its derivatives at $\hat{\tau}$ meet the appropriate conditions.

5.2. Optimization of α

In Section 4.2, we have shown that the choice $\alpha = -1$, along with an appropriate choice of f, ensures the asymptotic consistency of semi-supervised learning for $K = 2$ classes, in the sense that non-trivial asymptotic accuracy (> 0.5) can be achieved. This choice of α may however not be optimal in general. This subsection is devoted to the optimization of α so as to maximize the average precision, a criterion often used in absence of prior information to favor one class over the other. While not fully able to estimate the optimal $\alpha^\star$ of α, we shall discuss here a heuristic means to select a close-to-optimal α, subsequently denoted α_0.

As per Theorem 2, α must be chosen as $\alpha = -1 + \frac{\beta}{\sqrt{p}}$ for some $\beta = O(1)$. In order to set $\beta = \beta^\star$ in such a way that the classification accuracy is maximized, Corollary 3 further suggests the need to estimate the θ_b^a terms which in turn requires the evaluation of a certain number of quantities appearing in the expressions of m_b and Σ_b. Most of these are however not directly accessible from simple statistics of the data. Instead, we shall propose here a heuristic and simple method to retrieve a reasonable choice β_0 for β, which we claim is often close to optimal and sufficient for most needs.

To this end, first observe from (9) that the mappings $\beta \mapsto [m_b]_a$ satisfy

$$\frac{d}{d\beta}([m_b]_b - [m_b]_a) = \frac{f'(\tau)}{f(\tau)c_{[l]}}(t_b - t_a) = -\frac{d}{d\beta}([m_a]_a - [m_a]_b).$$

Hence, changes in β induce a simultaneous reduction and increase of either one of $[m_b]_b - [m_b]_a$ and $[m_a]_a - [m_a]_b$. Placing ourselves again in the case $K = 2$, we define β_0 to be the value for which both differences (with $a \neq b \in \{1, 2\}$) are the same, leading to the following Proposition–Definition.

Proposition 4 *Let $K = 2$ and $[m_b]_a$ be given by (9). Then*

$$\beta_0 \equiv \frac{f(\tau)}{f''(\tau)} \frac{c_{[l]1} - c_{[l]2}}{t_1 - t_2} \Delta m \tag{12}$$

where

$$\Delta m = -\frac{2f'(\tau)}{f(\tau)} \|\mu_1 - \mu_2\|^2 + \left(\frac{f''(\tau)}{f(\tau)} - \frac{f'(\tau)^2}{f(\tau)^2} \right) (t_1 - t_2)^2 + \frac{2f''(\tau)}{f(\tau)} (T_{11} + T_{22} - 2T_{12})$$

is such that, for $\alpha = -1 + \frac{\beta_0}{\sqrt{p}}$, $[m_1]_1 - [m_1]_2 = [m_2]_2 - [m_2]_1$.

By choosing $\alpha = \alpha_0 \equiv -1 + \frac{\beta_0}{\sqrt{p}}$, one ensures that $\mathbb{E}_{x_i \in \mathcal{C}_1}[\hat{F}_{i1} - \hat{F}_{i2}] = -\mathbb{E}_{x_i \in \mathcal{C}_2}[\hat{F}_{i1} - \hat{F}_{i2}] + o(1)$ ($i > n_{[l]}$), thereby evenly balancing the *average* "resolution" of each class. An even balance typically produces the desirable output of the central displays of Figure 2 (as opposed to the largely undesirable bottom of top displays, there for very offset values of α). Obviously though, since the variances of $\hat{F}_{i1} - \hat{F}_{i2}$ for $x_i \in \mathcal{C}_1$ or $x_i \in \mathcal{C}_2$ are in general not the same, this choice of α may not be optimal. Nonetheless, in most experimental scenarios of practical interest, the score variances tend to be sufficiently similar for the choice of α_0 to be quite appealing.

This heuristic motivation made, note that β_0 is proportional to $c_{[l]b} - c_{[l]a}$. This indicates that the more unbalanced is the labelled data set, the more deviated from zero is β_0. In particular, for $n_{[l]1} = n_{[l]2}$, $\alpha_0 = -1$. As we shall subsequently observe in simulations, this remark is of dramatic importance in practice where taking $\alpha = -1$ (the PageRank method) in place of $\alpha = \alpha_0$ leads to significant performance losses.

Of utmost importance here is the fact that, unlike θ_0^a which are difficult to assess empirically, a consistent estimate of β_0 can be obtained through a rather simple method, which we presently elaborate on.

While an estimate for t_a and T_{ab} can be obtained empirically from the labelled data themselves, $\|\mu_1 - \mu_2\|^2$ is not directly accessible (note indeed that $\frac{1}{n_{[l]a}} \sum_{\mathcal{C}_a} x_i = \mu_a + \frac{1}{n_{[l]a}} \sum_{\mathcal{C}_a} w_i$, for some $w_i \sim \mathcal{N}(0, C_a)$, and the central limit theorem guarantees that $\|\frac{1}{n_{[l]a}} \sum_{\mathcal{C}_a} w_i\| = O(1)$, the same order of magnitude as $\|\mu_a - \mu_b\|$). However, one may access an estimate for Δm by running two instances of the PageRank algorithm ($\alpha = -1$), resulting in the method described in Algorithm 1. It is easily shown that, under Assumptions 1–2,

$$\hat{\beta}_0 - \beta_0 \xrightarrow{\text{a.s.}} 0.$$

Figure 6 provides a performance comparison, in terms of average precision, between the PageRank ($\alpha = -1$) method and the proposed heuristic improvement for $\alpha = \alpha_0$, versus the oracle estimator for which $\alpha = \alpha^\star$, the precision-maximizing value. The curves are here

Algorithm 1 Estimate $\hat{\beta}_0$ of β_0.

1: Let $\hat{\tau}$ be given by (11).
2: Let

$$\widehat{\Delta t} = \frac{1}{2\sqrt{p}} \left(\frac{\sum_{i,j=1}^{n_{[l]1}} \|x_i - x_j\|^2}{n_{[l]1}(n_{[l]1} - 1)} - \frac{\sum_{i,j=n_{[l]1}+1}^{n_{[l]}} \|x_i - x_j\|^2}{n_{[l]2}(n_{[l]2} - 1)} \right)$$

3: Set $\alpha = -1$ and define $J \equiv p \sum_{i=n_{[l]}+1}^{n} \hat{F}_{i1} - \hat{F}_{i2}$.
4: Still for $\alpha = -1$, reduce the set of labelled data to $n'_{[l]1} = n'_{[l]2} = \min\{n_{[l]1}, n_{[l]2}\}$ and, with obvious notations, let $J' \equiv p \sum_{i=n'_{[l]}+1}^{n} \hat{F}'_{i1} - \hat{F}'_{i2}$.
5: Return $\hat{\beta}_0 \equiv \frac{c_{[l]} f(\hat{\tau})}{f'(\hat{\tau}) \widehat{\Delta t}} \frac{J' - J}{n_{[u]}}$.

snapshots of typical classification precision obtained from examples of $n = 4096$ images with $c_{[l]} = 1/16$. As expected, the gain in performance is largest as $|c_{[l]1} - c_{[l]2}|$ is large. More surprisingly, the performances obtained are impressively close to optimal. It should be noted though that simulations revealed more unstable estimates of $\hat{\beta}_0$ for smaller values of n.

Note that the method for estimating β_0 provided in Algorithm 1 implicitly exploits the resolution of two equations (through the observation of J, J' obtained for different values of $n_{[l]1}, n_{[l]2}$) to retrieve the value of Δm defined in Proposition 4. Having access to Δm further allows access to $\|\mu_1 - \mu_2\|^2$, for instance by setting f so that $f''(\tau) = 0$ and $f''(\tau) f(\tau) = f'(\tau)^2$. This in turn allows access to all terms intervening in $[m_b]_a$ (as per (9)), making it possible to choose f so to maximize the distances $|[m_1]_1 - [m_1]_2|$ and $|[m_2]_2 - [m_2]_1|$. However, in addition to the cumbersome aspect of the induced procedure (and the instability implied by multiple evaluations of the scores F under several settings for f and $c_{[l]a}$), such operations also alter the values of the variances in (10) for which not all terms are easily estimated. It thus seems more delicate to derive a simple method to optimize f in addition to α.

6. Concluding Remarks

This article is part of a series of works consisting in evaluating the performance of kernel-based machine learning methods in the large dimensional data regime (Couillet and Benaych-Georges, 2015; Liao and Couillet, 2017; Couillet and Kammoun, 2016). Relying on the derivations of Couillet and Benaych-Georges (2015) that provide a Taylor expansion of radial kernel matrices around the limiting common value τ of $\frac{1}{p}\|x_i - x_j\|^2$ for $i \neq j$ and $p \to \infty$, we observed that the choice of the kernel function f merely affects the classification performances through the successive derivatives of f at τ. In particular, similar to the earlier analyses (Couillet and Benaych-Georges, 2015; Liao and Couillet, 2017; Couillet and Kammoun, 2016), we found that the case $f'(\tau) = 0$ induces a sharp phase transition on normalized data by which the asymptotic classification error rate vanishes. However, unlike the works (Couillet and Benaych-Georges, 2015; Liao and Couillet, 2017), the exact

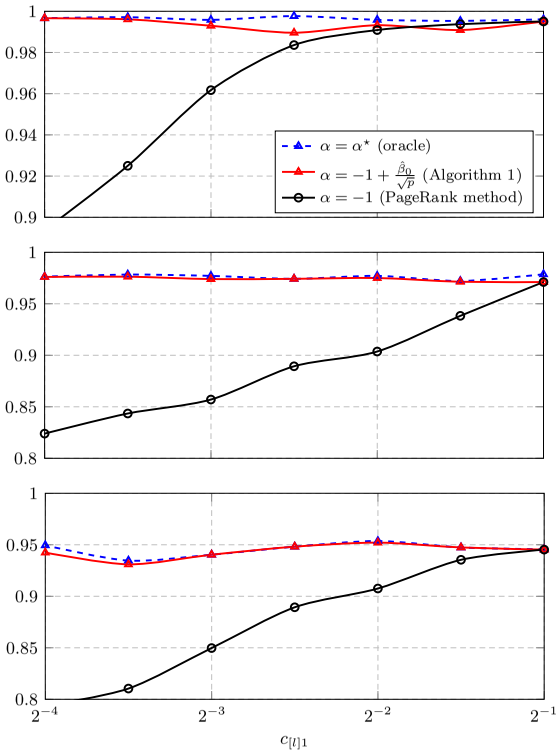


Figure 6: Average precision varying with $c_{[l]1}$ for 2-class MNIST data (**top:** digits (0,1), **middle:** digits (1,7), **bottom:** digits (8,9)), $n = 4096$, $p = 784$, $n_{[l]}/n = 1/16$, $n_{[u]1} = n_{[u]2}$, Gaussian kernel.

expression at the core of the limiting performance assumes a different form. Of importance is the finding that, under a heat kernel assumption $f(t) = \exp(-\frac{t}{2\sigma^2})$, the studied semi-supervised learning method fails to classify Gaussian mixtures of the type $\mathcal{N}(0, C_k)$ with $\operatorname{tr} C_k^\circ = O(\sqrt{p})$ and $\operatorname{tr} C_k C_{k'} - \operatorname{tr} C_k^2 = o(p)$, which unsupervised learning or LS-SVM are able to do (Couillet and Benaych-Georges, 2015; Liao and Couillet, 2017). This paradox may deserve a more structural way of considering together methods on the spectrum from unsupervised to supervised learning.

The very fact that the kernel matrix W is essentially equivalent to the matrix $f(\tau)1_n 1_n^\mathsf{T}$ (the $n \times n$ matrix filled with $f(\tau)$ values), thereby strongly disrupting with the expected

natural behavior of kernels, essentially follows from the Gaussian mixture model we assumed as well as from the decision to compare vectors by means of a mere Euclidean distance. We believe that this simplistic (although widely used) method explains the strong coincidence between performances on the Gaussian mixture model and on real data sets. Indeed, as radial functions are not specially adapted to image vectors (as would be wavelet or convolutional filters), the kernel likely operates on first order statistics of the input vectors, hence similar to its action on a Gaussian-mixture data. It would be interesting to generalize our result, and for that matter the set of works (Couillet and Benaych-Georges, 2015; Liao and Couillet, 2017; Couillet and Kammoun, 2016), to more involved data-oriented kernels, so long that the data contain enough exploitable degrees of freedom.

It is also quite instructive to note that, from the proof of our main results, the terms remaining after the expansion of $D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^{\alpha}$ appearing in (2) almost all vanish, strongly suggesting that similar results would be obtained if the inverse matrix in (2) were discarded altogether. This implies that the intra-unlabelled data kernel $W_{[uu]}$ is of virtually no asymptotic use. Also, the remark of Section 4.3 according to which $c_{[l]} \to 0$ implies a vanishing classification rate suggests that even the (unsupervised) clustering performance obtained by Couillet and Benaych-Georges (2015) is not achieved, despite the presence of possibly numerous unlabelled data. This, we believe, is due to a mismatched scaling in the SSL problem definition. A promising avenue of investigation would consist in introducing appropriate scaling parameters in the label propagation method or the optimization (1) to ensure that $W_{[uu]}$ is effectively used in the algorithm. Early simulations do suggest that elementary amendments to (2) indeed result in possibly striking performance improvements.

These considerations are left to future works.

Acknowledgments

This work is supported by the ANR Project RMT4GRAPH (ANR-14-CE28-0006).

Appendix A. Preliminaries

We begin with some additional notations that will be useful in the proofs.

- For $x_i \in \mathcal{C}_k$, $\omega_i \equiv (x_i - \mu_k)/\sqrt{p}$, and $\Omega \equiv [\omega_1, \cdots, \omega_n]^\mathsf{T}$
- $j_k \in \mathbb{R}^n$ is the canonical vector of $\mathcal{C}_k$, in the sense that its i-th element is 1 if $x_i \in \mathcal{C}_k$ or 0 otherwise. $j_{[l]k}$ and $j_{[u]k}$ are respectively the canonical vectors for labelled and unlabelled data of $\mathcal{C}_k$.
- $\psi_i \equiv \|\omega_i\|^2 - \mathrm{E}[\|\omega_i\|^2]$, $\psi \equiv [\psi_1, \cdots, \psi_n]^\mathsf{T}$ and $(\psi)^2 \equiv [(\psi_1)^2, \cdots, (\psi_n)^2]^\mathsf{T}$.

With these notations at hand, we introduce next the generalized version of Theorem 2 for all $\alpha = O(1)$ (rather than $\alpha = -1 + O(1/\sqrt{n})$).

Theorem 5 *For $x_i \in \mathcal{C}_b$ an unlabelled vector (i.e., $i > n_{[l]}$), let $\hat{F}_{ia}$ be given by (7) with F defined in (2) for $\alpha = O(1)$. Then, under Assumptions 1–2,*

$$p\hat{F}_{i\cdot} = p(1+z_i)1_K + G_i + o_P(1)$$
$$G_i \sim \mathcal{N}(m_b, \Sigma_b)$$

where z_i is as in Theorem 2 and

(i) for $F_{i\cdot}$ considered on the σ-field induced by the random variables $x_{[l]+1}, \ldots, x_n$, $p = 1, 2, \ldots$,

$$[m_b]_a = H_{ab} + \frac{1}{n_{[l]}}\sum_{d=1}^{K}(\alpha n_d + n_{[u]d})H_{ad} + (1+\alpha)\frac{n}{n_{[l]}}\left[\Delta_a + \frac{p}{n_{[l]a}}\frac{f'(\tau)}{f(\tau)}\psi_{[l]}^T j_{[l]a} - \alpha\frac{f'(\tau)^2}{f(\tau)^2}t_a t_b\right] \tag{13}$$

$$[\Sigma_b]_{a_1a_2} = \left(\frac{(-\alpha^2-\alpha)n - n_{[l]}}{n_{[l]}}\frac{f'(\tau)^2}{f(\tau)^2} + \frac{f''(\tau)}{f(\tau)}\right)^2 T_{bb}t_{a_1}t_{a_2} + \boldsymbol{\delta}_{a_1}^{a_2}\frac{f'(\tau)^2}{f(\tau)^2}\frac{4c_0 T_{ba_1}}{c_{[l]a_1}} + \frac{4f'(\tau)^2}{f(\tau)^2}\mu_{a_1}^\circ C_b \mu_{a_2}^\circ \tag{14}$$

where

$$H_{ab} = \frac{f'(\tau)}{f(\tau)}\|\mu_b^\circ - \mu_a^\circ\|^2 + \left(\frac{f''(\tau)}{f(\tau)} - \frac{f'(\tau)^2}{f(\tau)^2}\right)t_a t_b + \frac{2f''(\tau)}{f(\tau)}T_{ab} \tag{15}$$

$$\Delta_a = \frac{\sqrt{p}f'(\tau)}{f(\tau)}t_a + \frac{\alpha f'(\tau)^2 + f(\tau)f''(\tau)}{2f(\tau)^2}\left(2T_{aa} + t_a^2\right) + \frac{1}{n_{[l]}}\left(\frac{f'(\tau)}{f(\tau)}\right)^2\left(\sum_{d=1}^{K} n_{[u]d}t_d\right)t_a. \tag{16}$$

(ii) for $F_{i\cdot}$ considered on the σ-field induced by the random variables $x_1, \ldots, x_n$,

$$[m_b]_a = H_{ab} + \frac{1}{n_{[l]}}\sum_{d=1}^{K}(\alpha n_d + n_{[u]d})H_{ad} + (1+\alpha)\frac{n}{n_{[l]}}\left[\Delta_a - \alpha\frac{f'(\tau)^2}{f(\tau)^2}t_a t_b\right]$$

$$[\Sigma_b]_{a_1a_2} = \left(\frac{(-\alpha^2-\alpha)n - n_{[l]}}{n_{[l]}}\frac{f'(\tau)^2}{f(\tau)^2} + \frac{f''(\tau)}{f(\tau)}\right)^2 T_{bb}t_{a_1}t_{a_2} + \boldsymbol{\delta}_{a_1}^{a_2}\frac{f'(\tau)^2}{f(\tau)^2}\left(\frac{(1+\alpha)^2}{c_{[l]}^2}\frac{2c_0 T_{aa}}{c_{[l]a_1}} + \frac{4c_0 T_{ba_1}}{c_{[l]a_1}}\right) + \frac{4f'(\tau)^2}{f(\tau)^2}\mu_{a_1}^\circ C_b \mu_{a_2}^\circ$$

with H_{ab} given in (15) and Δ_a in (16).

Let $P(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b, x_1, \cdots, x_{n_{[l]}})$ denote the probability of correct classification of $x_i \in \mathcal{C}_b$ unlabelled, conditioned on $x_1, \ldots, x_{n_{[l]}}$, and $P(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b)$ the unconditional probability. Recall that the probability of correct classification of $x_i \in \mathcal{C}_b$ is the same as the probability of $\hat{F}_{ib} > \max_{a \neq b}\hat{F}_{ib}$, which, according to the above theorem, is asymptotically the probability that $[G_i]_b$ is the greatest element of G_i. Particularly for $K = 2$, we have the following corollary.

Corollary 6 *Under the conditions of Theorem 1, and with $K = 2$, we have, for $a \neq b \in \{1, 2\}$,*

(i) Conditionally on $x_1, \cdots, x_{n_{[l]}}$,

$$\mathbb{P}\left(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b, x_1, \cdots, x_{n_{[l]}}\right) - \Phi(\theta_b^a) \to 0$$
$$\theta_b^a = \frac{[m_b]_b - [m_b]_a}{\sqrt{[\Sigma_b]_{bb} + [\Sigma_b]_{aa} - 2[\Sigma_b]_{ab}}}$$

where $\Phi(u) = \frac{1}{2\pi}\int_{-\infty}^{u}\exp(-t^2/2)dt$ and m_b, Σ_b are given in (i) of Theorem 5.

(ii) Unconditionally,

$$\mathbb{P}\left(x_i \to \mathcal{C}_b | x_i \in \mathcal{C}_b\right) - \Phi(\theta_b^a) \to 0$$
$$\theta_b^a = \frac{[m_b]_b - [m_b]_a}{\sqrt{[\Sigma_b]_{bb} + [\Sigma_b]_{aa} - 2[\Sigma_b]_{ab}}}$$

where here m_b, Σ_b are given in (ii) of Theorem 5.

The remainder of the appendix is dedicated to the proof of Theorem 5 and Corollary 6 from which the results of Section 3.2 directly unfold.

Appendix B. Proof of Theorems 5

The proof of Theorem 5 is divided into two steps: first, we Taylor-expand the normalized scores for unlabelled data $\hat{F}_{[u]}$ using the convergence $\frac{1}{p}\|x_i - x_j\|^2 \xrightarrow{\text{a.s.}} \tau$ for all $i \neq j$; this expansion yields a random equivalent $\hat{F}_{[u]}^{\text{eq}}$ in the sense that $p(\hat{F}_{[u]} - \hat{F}_{[u]}^{\text{eq}}) \xrightarrow{\text{a.s.}} 0$. Proposition 1 is directly obtained from $\hat{F}_{[u]}^{\text{eq}}$. We then complete the proof by demonstrating the convergence to Gaussian variables of $\hat{F}_{[u]}^{\text{eq}}$ by means of a central limit theorem argument.

B.1. Step 1: Taylor expansion

In the following, we provide a sketch of the development of $F_{[u]}$; most unshown intermediary steps can be retrieved from simple, yet painstaking algebraic calculus.

Recall from (2) the expression of the unnormalized scores for unlabelled data

$$F_{[u]} = (I_{n_u} - D_{[u]}^{-1-\alpha}W_{[uu]}D_{[u]}^{\alpha})^{-1}D_{[u]}^{-1-\alpha}W_{[ul]}D_{[l]}^{\alpha}F_{[l]}.$$

We first proceed to the development of the terms $W_{[ul]}$, $W_{[uu]}$, subsequently to $D_{[l]}, D_{[u]}$, to then reach an expression for $F_{[u]}$. To this end, owing to the convergence $\|x_i - x_j\|^2/p \xrightarrow{\text{a.s.}} \tau$

for all $i \neq j$, we first Taylor-expand $W_{ij} = f(\|x_i - x_j\|^2/p)$ around $f(\tau)$ to obtain the following expansion for W, already evaluated by Couillet and Benaych-Georges (2015),

$$W = W^{(n)} + W^{(\sqrt{n})} + W^{(1)} + O(n^{-\frac{1}{2}}) \tag{17}$$

where $\|W^{(n)}\| = O(n)$, $\|W^{(\sqrt{n})} = O(\sqrt{n})$ and $\|W^{(1)}\| = O(1)$, with the definitions

$$\begin{aligned}
W^{(n)} &= f(\tau) 1_n 1_n^\mathsf{T} \\
W^{(\sqrt{n})} &= f'(\tau) \left[\psi 1_n^\mathsf{T} + 1_n \psi^\mathsf{T} + \left(\sum_{b=1}^K \frac{t_b}{\sqrt{p}} j_b \right) 1_n^\mathsf{T} + 1_n \sum_{a=1}^K \frac{t_a}{\sqrt{p}} j_a^\mathsf{T} \right] \\
W^{(1)} &= f'(\tau) \left[\sum_{a,b=1}^K \frac{\|\mu_a^\circ - \mu_b^\circ\|^2}{p} j_b j_a^\mathsf{T} - \frac{2}{\sqrt{p}} \Omega \sum_{a=1}^K \mu_a^\circ j_a^\mathsf{T} + \frac{2}{\sqrt{p}} \sum_{b=1}^K \mathrm{diag}(j_b) \Omega \mu_b^\circ 1_n^\mathsf{T} \right. \\
&\quad \left. - \frac{2}{\sqrt{p}} \sum_{b=1}^K j_b \mu_b^{\circ\mathsf{T}} \Omega^\mathsf{T} + \frac{2}{\sqrt{p}} 1_n \sum_{a=1}^K \mu_a^{\circ\mathsf{T}} \Omega^\mathsf{T} \mathrm{diag}(j_a) - 2\Omega\Omega^\mathsf{T} \right] \\
&\quad + \frac{f''(\tau)}{2} \left[(\psi)^2 1_n^\mathsf{T} + 1_n [(\psi)^2]^\mathsf{T} + \sum_{b=1}^K \frac{t_b^2}{p} j_b 1_n^\mathsf{T} + 1_n \sum_{a=1}^K \frac{t_a^2}{p} j_a^\mathsf{T} \right. \\
&\quad + 2 \sum_{a,b=1}^K \frac{t_a t_b}{p} j_b j_a^\mathsf{T} + 2 \sum_{b=1}^K \mathrm{diag}(j_b) \frac{t_b}{\sqrt{p}} \psi 1_n^\mathsf{T} + 2 \sum_{b=1}^K \frac{t_b}{\sqrt{p}} j_b \psi^\mathsf{T} + 2 \sum_{a=1}^K 1_n \psi^\mathsf{T} \mathrm{diag}(j_a) \frac{t_a}{\sqrt{p}} \\
&\quad \left. + 2\psi \sum_{a=1}^K \frac{t_a}{\sqrt{p}} j_a^\mathsf{T} + 4 \sum_{a,b=1}^K \frac{T_{ab}}{p} j_b j_a^\mathsf{T} + 2\psi\psi^\mathsf{T} \right] + (f(0) - f(\tau) + \tau f'(\tau)) I_n.
\end{aligned}$$

As $W_{[ul]}$, $W_{[uu]}$ are sub-matrices of W, their approximated expressions are obtained directly by extracting the corresponding subsets of (17). Applying then (17) in $D = \mathrm{diag}(W 1_n)$, we next find

$$D = n f(\tau) \left[I_n + \frac{1}{n f(\tau)} \mathrm{diag}(W^{(\sqrt{n})} 1_n + W^{(1)} 1_n) \right] + O(n^{-\frac{1}{2}}).$$

Thus, for any $\sigma \in \mathbb{R}$, $(n^{-1}D)^\sigma$ can be Taylor-expanded around $f(\tau)^\sigma I_n$ as

$$\begin{aligned}
(n^{-1}D)^\sigma &= f(\tau)^\sigma \left[I_n + \frac{\sigma 1}{n f(\tau)} \mathrm{diag}((W^{(\sqrt{n})} + W^{(1)}) 1_n) + \frac{\sigma(\sigma-1)}{2n^2 f(\tau)^2} \mathrm{diag}^2(W^{(\sqrt{n})} 1_n) \right] \\
&\quad + O(n^{-\frac{3}{2}})
\end{aligned} \tag{18}$$

where $\mathrm{diag}^2(\cdot)$ stands for the squared diagonal matrix. The Taylor-expansions of $(n^{-1}D_{[u]})^\alpha$ and $(n^{-1}D_{[l]})^\alpha$ are then directly extracted from this expression for $\sigma = \alpha$, and similarly for $(n^{-1}D_{[u]})^{-1-\alpha}$ with $\sigma = -1-\alpha$. Since

$$D_{[u]}^{-1-\alpha} W_{[ul]} D_{[l]}^\alpha = \frac{1}{n} (n^{-1} D_{[u]})^{-1-\alpha} W_{[ul]} (n^{-1} D_{[l]})^\alpha$$

it then suffices to multiply the Taylor-expansions of $(n^{-1}D_{[u]})^\alpha$, $(n^{-1}D_{[l]})^\alpha$, and $W_{[ul]}$, given respectively in (18) and (17), normalize by n and then organize the result in terms of order $O(1)$, $O(1/\sqrt{n})$, and $O(1/n)$.

The term $D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha$ is dealt with in the same way. In particular,

$$D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha = \frac{1}{n} 1_{n_{[u]}} 1_{n_{[l]}} + O(n^{-\frac{1}{2}}).$$

Therefore, $(I_{n_{[u]}} - D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha)^{-1}$ may be simply written as

$$\left(I_{n_{[u]}} - \frac{1}{n} 1_{n_{[u]}} 1_{n_{[u]}} + O(n^{-\frac{1}{2}}) \right)^{-1} = I_{n_{[u]}} + \frac{1}{n_{[l]}} 1_{n_{[u]}} 1_{n_{[u]}} + O(n^{-\frac{1}{2}}).$$

Combining all terms together completes the full linearization of $\hat{F}_{[u]}$.

This last derivation, which we do not provide in full here, is simpler than it appears and is in fact quite instructive in the overall behavior of $F^{[u]}$. Indeed, only product terms in the development of $(I_{n_{[u]}} - D_{[u]}^{-1-\alpha} W_{[uu]} D_{[u]}^\alpha)^{-1}$ and $D_{[u]}^{-1-\alpha} W_{[ul]} D_{[l]}^\alpha F^{[l]}$ of order at least $O(1)$ shall remain, which discards already a few terms. Now, in addition, note that for any vector v, $v 1_{n_{[l]}}^\mathsf{T} F^{[l]} = v 1_k^\mathsf{T}$ so that such matrices are non informative for classification (they have identical score columns); these terms are all placed in the intermediary variable z, the entries z_i of which are irrelevant and thus left as is (these are the z_i's of Proposition 1 and Theorem 2). It is in particular noteworthy to see that *all* terms of $W_{[uu]}^{(1)}$ that remain after taking the product with $D_{[u]}^{-1-\alpha} W_{[ul]} D_{[l]}^\alpha F^{[l]}$ are precisely those multiplied by $f(\tau) 1_{n_{[u]}} 1_{n_{[l]}}^\mathsf{T} F^{[l]}$ and thus become part of the vector z. Since most informative terms in the kernel matrix development are found in $W^{(1)}$, this means that the algorithm under study shall make little use of the *unsupervised* information about the data (those found in $W_{[uu]}^{(1)}$). This is an important remark which, as discussed in Section 6, opens up the path to further improvements of the semi-supervised learning algorithms which would use more efficiently the information in $W_{[uu]}^{(1)}$.

All calculus made, this development finally leads to $F_{[u]} = F_{[u]}^{\mathrm{eq}}$ with, for $a, b \in \{1, \ldots, K\}$ and $x_i \in \mathcal{C}_b$, $i > n_{[l]}$,

$$\begin{aligned}
\hat{F}_{ia}^{\mathrm{eq}} &= 1 + \frac{1}{p} \left[H_{ab} + \frac{1}{n_{[l]}} \sum_{d=1}^K H_{ad} (\alpha n_d + n_{[u]d}) \right] + (1+\alpha) \frac{n}{p n_{[l]}} \left[\Delta_a - \alpha \frac{f'(\tau)^2}{f(\tau)^2} t_a t_b \right] \\
&\quad + \left(\frac{(-\alpha^2 - \alpha) n - n_{[l]}}{n_{[l]}} \frac{f'(\tau)^2}{f(\tau)^2} + \frac{f''(\tau)}{f(\tau)} \right) \frac{t_a}{\sqrt{p}} \psi_i + \frac{2 f'(\tau)}{f(\tau) \sqrt{p}} \mu_a^\circ \omega_i \\
&\quad + \frac{f'(\tau)}{f(\tau)} \left(\frac{(1+\alpha) n}{n_{[l]} n_{[l]a}} \psi_{[l]}^\mathsf{T} j_{[l]a} + \frac{4}{n_{[l]a}} j_{[l]a}^T \Omega_{[l]} \omega_i \right) + z_i
\end{aligned} \tag{19}$$

where H_{ab} is as specified in (15), Δ_a as in (16), and $z_i = O(\sqrt{p})$ is some residual random variable only dependent on x_i. Gathering the terms in successive orders of magnitude, Proposition 1 is then straightforwardly proven from (19).

B.2. Step 2: Central limit theorem

The focus of this step is to examine $\mathcal{G}_i = p(\hat{F}_i^{\mathrm{eq}} - (1 + z_i) 1_K)$. Theorem 5 can be proven by showing that $\mathcal{G}_i = G_i + o_P(1)$.

First consider Item (i) of Theorem 5, which describes the behavior of $\hat{F}_{[u]}$ conditioned on $x_1, \ldots, x_{n_{[l]}}$. Recall that a necessary and sufficient condition for a vector v to be a Gaussian vector is that all linear combinations of the elements of v are Gaussian variables. Thus, for given $x_1, \ldots, x_{n_{[l]}}$ deterministic, according to (19), $\mathcal{G}_i$ is asymptotically Gaussian if, for all $g_1 \in \mathbb{R}$, $g_2 \in \mathbb{R}^p$, $g_1\psi_i + g_2^T\omega_i$ has a central limit.

Letting $\omega_i = \frac{C_b^{\frac{1}{2}}}{\sqrt{p}} r$, with $r \sim \mathcal{N}(0, I_p)$, $g_1\psi_i + g_2\omega_i$ can be rewritten as $r^\mathsf{T} A r + br + c$ with $A = g_1\frac{C_b}{p}$, $b = g_2\frac{C_b}{p}$, $c = -g_1\frac{\mathrm{tr} C_b}{p}$. Since A is symmetric, there exists an orthonormal matrix U and a diagonal Λ such that $A = U^\mathsf{T}\Lambda U$. We thus get

$$r^\mathsf{T} A r + br + c = r^T U^T \Lambda U r + bU^T U r + c = \tilde{r}^T \Lambda \tilde{r} + \tilde{b}\tilde{r} + c$$

with $\tilde{r} = Ur$ and $\tilde{b} = bU^\mathsf{T}$. By unitary invariance, we have $\tilde{r} \sim \mathcal{N}(0, I_p)$ so that $g_1\psi_i + g_2\omega_i$ is thus the sum of the independent but not identically distributed random variables $q_j = \lambda_j \tilde{r}_j^2 + \tilde{b}_j\tilde{r}_j$, $i = 1, \ldots, p$. From Lyapunov's central limit theorem (Billingsley, 1995, Theorem 27.3), it remains to find a $\delta > 0$ such that $\frac{\sum_j \mathbb{E}|q_j - \mathbb{E}[q_j]|^{2+\delta}}{\left(\sum_j \mathrm{Var}[q_j]\right)^{1+\delta/2}} \to 0$ to ensure the central limit theorem. For $\delta = 1$, we have $\mathbb{E}[q_j] = \lambda_j$, $\mathrm{Var}[q_j] = 2\lambda_j^2 + \tilde{b}_j^2$ and $\mathbb{E}\left[(q_j - \mathbb{E}[q_j])^3\right] = 8\lambda_j^3 + 6\lambda_j\tilde{b}_j^2$, so that $\frac{\sum_j \mathbb{E}\left[|q_j - \mathbb{E}[q_j]|^3\right]}{\left(\sum_j \mathrm{Var}[q_j]\right)^{3/2}} = O(n^{-\frac{1}{2}})$.

It thus remains to evaluate the expectation and covariance matrix of $\mathcal{G}_i$ conditioned on $x_1, \ldots, x_{n_{[l]}}$ to obtain (i) of Theorem 5. For $x_i \in \mathcal{C}_b$, we have

$$\begin{aligned}
\mathbb{E}\{[\mathcal{G}_i]_a\} &= H_{ab} + \frac{1}{n_{[l]}} \sum_{d=1}^{K} (\alpha n_d + n_{[u]d}) H_{ad} \\
&\quad + (1+\alpha)\frac{n}{n_{[l]}} \left[\Delta_a + \frac{p}{n_{[l]a}} \frac{f'(\tau)}{f(\tau)} \psi_{[l]}^T j_{[l]a} - \alpha \frac{f'(\tau)^2}{f(\tau)^2} t_a t_b \right] \\
\mathrm{Cov}\{[\mathcal{G}_i]_{a_1}[\mathcal{G}_i]_{a_2}\} &= \left(\frac{(-\alpha^2 - \alpha)n - n_{[l]}}{n_{[l]}} \frac{f'(\tau)^2}{f(\tau)^2} + \frac{f''(\tau)}{f(\tau)} \right)^2 T_{bb} t_{a_1} t_{a_2} \\
&\quad + \boldsymbol{\delta}_{a_1}^{a_2} \frac{f'(\tau)^2}{f(\tau)^2} \frac{4c_0 T_{ba_1}}{c_{[l]a_1}} + \frac{4f'(\tau)^2}{f(\tau)^2} \mu_{a_1}^\circ C_b \mu_{a_2}^\circ + o(1).
\end{aligned}$$

From the above equations, we retrieve the asymptotic expressions of $[m_b]_a$ and $[\Delta_b]_{a_1 a_2}$ given in (13) and (14). This completes the proof of Item (i) of Theorem 5. Item (ii) is easily proved by following the same reasoning.

References

Aamir Anis, Aly El Gamal, Salman Avestimehr, and Antonio Ortega. Asymptotic justification of bandlimited interpolation of graph signals for semi-supervised learning. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5461–5465. IEEE, 2015.

Konstantin Avrachenkov, Paulo Gonçalves, Alexey Mishenin, and Marina Sokol. Generalized optimization framework for graph-based semi-supervised learning. *arXiv preprint arXiv:1110.4278*, 2011.

Mikhail Belkin and Partha Niyogi. Semi-supervised learning on riemannian manifolds. *Machine learning*, 56(1-3):209–239, 2004.

Mikhail Belkin, Irina Matveeva, and Partha Niyogi. Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory (COLT)*, pages 624–638. Springer, 2004.

Peter J Bickel, Bo Li, et al. Local polynomial regression on unknown manifolds. In *Complex Datasets and Inverse Problems*, pages 177–186. Institute of Mathematical Statistics, 2007.

P. Billingsley. *Probability and Measure*. John Wiley and Sons, Inc., Hoboken, NJ, third edition, 1995.

Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. *Semi-Supervised Learning*. MIT press, 2006.

Romain Couillet and Florent Benaych-Georges. Kernel spectral clustering of large dimensional data. *arXiv preprint arXiv:1510.03547*, 2015.

Romain Couillet and Abla Kammoun. Random matrix improved subspace clustering. In *Asilomar Conference on Signals, Systems and Computers*, pages 90–94. IEEE, 2016.

Akshay Gadde, Aamir Anis, and Antonio Ortega. Active semi-supervised learning using sampling theory for graph signals. In *International Conference on Knowledge Discovery and Data Mining*, pages 492–501. ACM, 2014.

Amir Globerson, Roi Livni, and Shai Shalev-Shwartz. Effective semi-supervised learning on manifolds. In *International Conference on Learning Theory (COLT)*, pages 978–1003, 2017.

Andrew B Goldberg, Xiaojin Zhu, Aarti Singh, Zhiting Xu, and Robert Nowak. Multi-manifold semi-supervised learning. 2009.

Martin Szummer Tommi Jaakkola and Martin Szummer. Partially labeled classification with markov random walks. *International Conference in Neural Information Processing Systems*, 14:945–952, 2002.

Thorsten Joachims et al. Transductive learning via spectral graph partitioning. In *International Conference on Machine Learning*, volume 3, pages 290–297, 2003.

Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits, 1998.

Zhenyu Liao and Romain Couillet. A large dimensional analysis of least squares support vector machines. *arXiv preprint arXiv:1701.02967*, 2017.

Amit Moscovich, Ariel Jaffe, and Boaz Nadler. Minimax-optimal semi-supervised regression on unknown manifolds. *arXiv preprint arXiv:1611.02221*, 2016.

Boaz Nadler, Nathan Srebro, and Xueyuan Zhou. Semi-supervised learning with the graph laplacian: the limit of infinite unlabelled data. In *International Conference on Neural Information Processing Systems*, pages 1330–1338, 2009.

Sunil K Narang, Akshay Gadde, and Antonio Ortega. Signal processing techniques for interpolation in graph structured data. In *IEEE International Conference Acoustics, Speech and Signal Processing*, pages 5445–5449. IEEE, 2013a.

Sunil K Narang, Akshay Gadde, Eduard Sanou, and Antonio Ortega. Localized iterative methods for interpolation in graph structured data. In *Global Conference on Signal and Information Processing*, pages 491–494. IEEE, 2013b.

Bernhard Schölkopf and Alexander J Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT press, 2002.

Larry Wasserman and John D Lafferty. Statistical analysis of semi-supervised regression. In *International Conference on Neural Information Processing Systems*, pages 801–808, 2008.

Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. volume 16, pages 321–328, 2004.

Xiaojin Zhu and Zoubin Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, Citeseer, 2002.

Xiaojin Zhu, Zoubin Ghahramani, John Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, volume 3, pages 912–919, 2003.

Journal of Machine Learning Research 19 (2018) 1-39 Submitted 8/17; Revised 10/18; Published 11/18

Robust PCA by Manifold Optimization

Teng Zhang TENG.ZHANG@UCF.EDU
Department of Mathematics
University of Central Florida
4000 Central Florida Blvd
Orlando, FL 32816, USA

Yi Yang YI.YANG6@MCGILL.CA
Department of Mathematics and Statistics
McGill University
805 Sherbrooke Street West
Montreal, QC H3A0B9, Canada

Editor: Michael Mahoney

Abstract

Robust PCA is a widely used statistical procedure to recover an underlying low-rank matrix with grossly corrupted observations. This work considers the problem of robust PCA as a nonconvex optimization problem on the manifold of low-rank matrices and proposes two algorithms based on manifold optimization. It is shown that, with a properly designed initialization, the proposed algorithms are guaranteed to converge to the underlying low-rank matrix linearly. Compared with a previous work based on the factorization of low-rank matrices Yi et al. (2016), the proposed algorithms reduce the dependence on the condition number of the underlying low-rank matrix theoretically. Simulations and real data examples confirm the competitive performance of our method.

Keywords: principal component analysis, low-rank modeling, manifold of low-rank matrices.

1. Introduction

In many problems, the underlying data matrix is assumed to be approximately low-rank. Examples include problems in computer vision Epstein et al. (1995); Ho et al. (2003), machine learning Deerwester et al. (1990), and bioinformatics Price et al. (2006). For such problems, principal component analysis (PCA) is a standard statistical procedure to recover the underlying low-rank matrix. However, PCA is highly sensitive to outliers in the data, and robust PCA Candès et al. (2011); Chandrasekaran et al. (2011); Clarkson and Woodruff (2013); Frieze et al. (2004); Bhojanapalli et al. (2015); Yi et al. (2016); Chen and Wainwright (2015); Gu et al. (2016); Cherapanamjeri et al. (2016); Netrapalli et al. (2014) is hence proposed as a modification to handle grossly corrupted observations. Mathematically, the robust PCA problem is formulated as follows: given a data matrix $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$ that can be written as the sum of a low-rank matrix $\mathbf{L}^*$ (signal) and a sparse matrix $\mathbf{S}^*$ (corruption) with only a few nonzero entries, can we recover both components accurately? Robust PCA has been shown to have applications in many real-life applications

including background detection Li et al. (2004), face recognition Basri and Jacobs (2003), ranking, and collaborative filtering Candès et al. (2011).

Since the set of all low-rank matrices is nonconvex, it is generally difficult to obtain an algorithm with theoretical guarantee since there is no tractable optimization algorithm for the nonconvex problem. Here we review a few carefully designed algorithms such that the theoretical guarantee on the recovery of underlying low-rank matrix exists. The works Candès et al. (2011); Chandrasekaran et al. (2011) consider the convex relaxation of the original problem instead:

$$\min_{\mathbf{L},\mathbf{S}} \|\mathbf{L}\|_* + \|\mathbf{S}\|_1, \text{ s.t. } \mathbf{Y} = \mathbf{L} + \mathbf{S}, \tag{1}$$

where $\|\mathbf{L}\|_*$ represents the nuclear norm (i.e., Schatten 1-norm) of $\mathbf{L}$, defined by the sum of its singular values and $\|\mathbf{S}\|_1$ represents the sum of the absolute values of all entries of $\mathbf{S}$. Since this problem is convex, the solution to (1) can be solved in polynomial time. In addition, it is shown that the solution recovers the correct low-rank matrix when $\mathbf{S}^*$ has at most $\gamma^* = O(1/\mu^2 r)$ fraction of corrupted non-zero entries, where r is the rank of $\mathbf{L}^*$ and μ is the incoherence level of $\mathbf{L}^*$ Hsu et al. (2011). If the sparsity of $\mathbf{S}^*$ is assumed to be random, then Candès et al. (2011) shows that the algorithm succeeds with high probability, even when the percentage of corruption can be in the order of $O(1)$ while the rank $r = O(\min(n_1, n_2)/\mu \log^2 \max(n_1, n_2))$, where μ is a coherence parameter of the low-rank matrix $\mathbf{L}^*$ (this work defines μ slightly differently compared to Candès et al. (2011) and (16) in this work, but the value is comparable).

However, the aforementioned algorithms based on convex relaxation have a computational complexity of $O(n_1 n_2 \min(n_1, n_2))$ per iteration, which could be prohibitive when n_1 and n_2 are very large. Alternatively, some faster algorithms are proposed based on nonconvex optimization. In particular, the work by Kyrillidis and Cevher (2012) proposes a method based on the projected gradient method. However, it assumes that the sparsity pattern of $\mathbf{S}^*$ is random, and the algorithm still has the same computational complexity as the convex methods. Netrapalli et al. (2014) proposes a method based on the alternating projecting, which allows $\gamma^* \le \frac{1}{\mu^2 r}$, with a computational complexity of $O(r^2 n_1 n_2)$ per iteration. Chen and Wainwright (2015) assumes that $\mathbf{L}^*$ is positive semidefinite and applies the gradient descent method on the Cholesky decomposition factor of $\mathbf{L}^*$, but the positive semidefinite assumption is not satisfied in many applications. Gu et al. (2016) factorizes $\mathbf{L}^*$ into the product of two matrices and performs alternating minimization over both matrices. It shows that the algorithm allows $\gamma^* = O(1/\mu^{2/3} r^{2/3} \min(n_1, n_2))$ and has the complexity of $O(r^2 n_1 n_2)$ per iteration. Yi et al. (2016) applies a similar factorization and applies an alternating gradient descent algorithm with a complexity of $O(r n_1 n_2)$ per iteration and allows $\gamma^* = O(1/\kappa^2 \mu r^{3/2})$, where κ is the condition number of the underlying low-rank matrix. There is another line of works that further reduces the complexity of the algorithm by subsampling the entries of the observation matrix $\mathbf{Y}$, including Mackey et al. (2011); Li and Haupt (2015); Rahmani and Atia (2017); Cherapanamjeri et al. (2016) and (Yi et al., 2016, Algorithm 2), which will also be discussed in this paper as the partially observed case.

The common idea shared by Gu et al. (2016) and Yi et al. (2016) is as follows. Since any low-rank matrix $\mathbf{L} \in \mathbb{R}^{n_1 \times n_2}$ with rank r can be written as the product of two low-rank

matrices by $\mathbf{L} = \mathbf{U}\mathbf{V}^T$ with $\mathbf{U} \in \mathbb{R}^{n_1 \times r}$ and $\mathbf{V} \in \mathbb{R}^{n_2 \times r}$, we can optimize the pair $(\mathbf{U}, \mathbf{V})$ instead of $\mathbf{L}$, and a smaller computational cost is expected since $(\mathbf{U}, \mathbf{V})$ has $(n_1 + n_2)r$ parameters, which is smaller than $n_1 n_2$, the number of parameters in $\mathbf{L}$. In fact, such a re-parametrization technique has a long history Ruhe (1974), and has been popularized by Burer and Monteiro Burer and Monteiro (2003, 2005) for solving semi-definite programs (SDPs). The same idea has been used in other low-rank matrix estimation problems such as dictionary learning Sun et al. (2017), phase synchronization Boumal (2016), community detection Bandeira et al. (2016), matrix completion Jain et al. (2013), recovering matrix from linear measurements Tu et al. (2016), and even general problems Chen and Wainwright (2015); Wang et al. (2017); Park et al. (2016); Wang et al. (2017); Park et al. (2017). In addition, the property of associated stochastic gradient descent algorithm is studied in De Sa et al. (2015).

The main contribution of this work is a novel robust PCA algorithm based on the gradient descent algorithm on the manifold of low-rank matrices, with a theoretical guarantee on the exact recovery of the underlying low-rank matrix. Compared with Yi et al. (2016), the proposed algorithm utilizes the tool of manifold optimization, which leads to a simpler and more naturally structured algorithm with a stronger theoretical guarantee. In particular, with a proper initialization, our method can still succeed with $\gamma^* = O(1/\kappa\mu r^{3/2})$, which means that it can tolerate more corruption than Yi et al. (2016) by a factor of κ. In addition, the theoretical convergence rate is also faster than Yi et al. (2016) by a factor of κ. Simulations also verified the advantage of the proposed algorithm over Yi et al. (2016). We remark that while manifold optimization has been applied to robust PCA in Cambier and Absil (2016), our work studies a different algorithm and gives theoretical guarantees. Considering the popularity of the methods based on the factorization of low-rank matrices, it is expected that manifold optimization could be applied to other low-rank matrix estimation problems. In addition, we implement our method in an efficient and user-friendly R package `morpca`, which is available at `https://github.com/emeryyi/morpca`.

The paper is organized as follows. We first present the algorithm in Section 2, and explain how the proposed algorithms are derived in Section 3. Their theoretical properties are studied and compared with previous algorithms in Section 4. In Section 5, simulations and real data analysis on the `Shoppingmall` dataset show that the proposed algorithms are competitive in many scenarios and have superior performances to the algorithm based on matrix factorization. A discussion about the proposed algorithms is then presented in Section 6, followed by the proofs of the results in Appendix.

2. Algorithm

In this work, we consider the robust PCA problem in two settings: fully observed setting and partially observed setting. The problem under the fully observed setting can be formulated as follows: given $\mathbf{Y} = \mathbf{L}^* + \mathbf{S}^*$, where $\mathbf{L}^*$ is a low-rank matrix and $\mathbf{S}^*$ is a sparse matrix, then can we recover $\mathbf{L}^*$ from $\mathbf{Y}$? To recover $\mathbf{L}^*$, we solve the following optimization problem:

$$\widehat{\mathbf{L}} = \underset{\text{rank}(\mathbf{L})=r}{\arg\min}\ f(\mathbf{L}), \text{ where } f(\mathbf{L}) = \frac{1}{2}\|F(\mathbf{L}-\mathbf{Y})\|_F^2, \tag{2}$$

where $F : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ is a hard thresholding procedure defined in (3):

$$F_{ij}(\mathbf{A}) = \begin{cases} 0, & \text{if } |\mathbf{A}_{ij}| > |\mathbf{A}_{i,\cdot}|^{[\gamma]} \text{ and } |\mathbf{A}_{ij}| > |\mathbf{A}_{\cdot,j}|^{[\gamma]} \\ \mathbf{A}_{ij}, & \text{otherwise.} \end{cases} \tag{3}$$

Here $\mathbf{A}_{i,\cdot}$ represents the i-th row of the matrix $\mathbf{A}$, and $\mathbf{A}_{\cdot,j}$ represents the j-th column of $\mathbf{A}$. $|\mathbf{A}_{i,\cdot}|^{[\gamma]}$ and $|\mathbf{A}_{\cdot,j}|^{[\gamma]}$ represent the $(1-\gamma)$-th percentile of the absolute values of the entries of $\mathbf{A}_{i,\cdot}$ and $\mathbf{A}_{\cdot,j}$ for $\gamma \in [0, 1)$. In other words, what are removed are the entries that are simultaneously among the largest γ-fraction in the corresponding row and column of $\mathbf{A}$ in terms of the absolute values. The threshold γ is set by users. If some entries of $\mathbf{A}_{i,\cdot}$ or $\mathbf{A}_{\cdot,j}$ have the entries with identical absolute values, the ties can be broken down arbitrarily.

The motivation is that, if $\mathbf{S}^*$ is sparse in the sense that the percentage of nonzero entries in each row and each column is smaller than γ, then $F(\mathbf{L}^* - \mathbf{Y}) = F(-\mathbf{S}^*)$ is zero by definition thus $f(\mathbf{L}^*)$ is zero. Since f is nonnegative, $\mathbf{L}^*$ is the solution to (2). To solve (2), we propose Algorithm 1 based on manifold optimization, with its derivation deferred to Section 3.3.1.

Algorithm 1 Gradient descent on the manifold under the fully observed setting.

Input: Observation $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$; Rank r; Thresholding value γ; Step size η.
Initialization: Set $k = 0$; Initialize $\mathbf{L}^{(0)}$ using the rank-r approximation to $F(\mathbf{Y})$.
Loop: Iterate Steps 1–4 until convergence:
1: Let $\mathbf{L}^{(k)} = \mathbf{U}^{(k)}\mathbf{\Sigma}^{(k)}\mathbf{V}^{(k)T}$.
2: Let $\mathbf{D}^{(k)} = F(\mathbf{L}^{(k)} - \mathbf{Y})$.
3(a): (Option 1) Let $\mathbf{\Omega}^{(k)} = \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}^{(k)} + \mathbf{D}^{(k)}\mathbf{V}^{(k)}\mathbf{V}^{(k)T} - \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}^{(k)}\mathbf{V}^{(k)}\mathbf{V}^{(k)T}$, and let $\mathbf{U}^{(k+1)} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{\Sigma}^{(k+1)} \in \mathbb{R}^{r \times r}$, and $\mathbf{V}^{(k+1)} \in \mathbb{R}^{n_2 \times r}$ be matrices consist of the top r left singular vectors/singular values/right singular vectors of $\mathbf{L}^{(k)} - \eta\mathbf{\Omega}^{(k)}$.
3(b): (Option 2) Let $\mathbf{Q}_1, \mathbf{R}_1$ be the QR decomposition of $(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})^T\mathbf{U}^{(k)}$ and $\mathbf{Q}_2, \mathbf{R}_2$ be the QR decomposition of $(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})\mathbf{V}^{(k)}$. Then $\mathbf{U}^{(k+1)} = \mathbf{Q}_2$, $\mathbf{V}^{(k+1)} = \mathbf{Q}_1$ and $\mathbf{\Sigma}^{(k+1)} = \mathbf{R}_2[\mathbf{U}^{(k)T}(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})\mathbf{V}^{(k)}]^{-1}\mathbf{R}_1^T$.
4: $k \coloneqq k + 1$.
Output: Estimation of the low-rank matrix $\mathbf{L}^*$, given by $\lim_{k\to\infty} \mathbf{L}^{(k)}$.

Under the partially observed setting, in addition to gross corruption $\mathbf{S}^*$, the observed matrix $\mathbf{Y}$ has a large number of missing values, i.e., many entries of $\mathbf{Y}$ are not observed. We denote the set of all observed entries by $\mathbf{\Phi} = \{(i,j)|\mathbf{Y}_{ij} \text{ is observed}\}$, and define $\tilde{F} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$

$$\tilde{F}_{ij}(\mathbf{A}) = \begin{cases} 0, & \text{if } |\mathbf{A}_{ij}| > |\mathbf{A}_{i,\cdot}|^{[\gamma,\mathbf{\Phi}]} \text{ and } |\mathbf{A}_{ij}| > |\mathbf{A}_{\cdot,j}|^{[\gamma,\mathbf{\Phi}]} \\ \mathbf{A}_{ij}, & \text{otherwise.} \end{cases} \tag{4}$$

Here $|\mathbf{A}_{i,\cdot}|^{[\gamma,\mathbf{\Phi}]}$ and $|\mathbf{A}_{\cdot,j}|^{[\gamma,\mathbf{\Phi}]}$ represent the $(1-\gamma)$-th percentile of the absolute values of the observed entries of $\mathbf{A}_{i,\cdot}$ and $\mathbf{A}_{\cdot,j}$ of the matrix $\mathbf{A}$ respectively.

As a generalization of Algorithm 1, we propose to solve

$$\underset{\text{rank}(\mathbf{L})=r}{\arg\min}\ \tilde{f}(\mathbf{L}), \quad \tilde{f}(\mathbf{L}) = \frac{1}{2}\sum_{(i,j)\in\mathbf{\Phi}} \tilde{F}_{ij}(\mathbf{L}-\mathbf{Y})^2, \tag{5}$$

Algorithm 2 Gradient descent on the manifold under the partially observed setting.

Input: Observation $\mathbf{Y} \in \mathbb{R}^{n_1 \times n_2}$; Set of all observed entries by $\mathbf{\Phi}$; Rank r; Thresholding value γ; Step size η.
Initialization: Set $k = 0$; Initialize $\mathbf{L}^{(0)}$ using the rank-r approximation to $\tilde{F}(\mathbf{Y})$.
Loop: Iterate Steps 1–4 until convergence:
1: Let $\mathbf{L}^{(k)}$ be a sparse matrix with support $\mathbf{\Phi}$, with nonzero entries given by the corresponding entries of $\mathbf{U}^{(k)}\mathbf{\Sigma}^{(k)}\mathbf{V}^{(k)\,T}$.
2: Let $\mathbf{D}^{(k)} = \tilde{F}(\mathbf{L}^{(k)} - \mathbf{Y})$.
3(a): (Option 1) Let $\mathbf{\Omega}^{(k)} = \mathbf{U}^{(k)}\mathbf{U}^{(k)\,T}\mathbf{D}^{(k)} + \mathbf{D}^{(k)}\mathbf{V}^{(k)}\mathbf{V}^{(k)\,T} - \mathbf{U}^{(k)}\mathbf{U}^{(k)\,T}\mathbf{D}^{(k)}\mathbf{V}^{(k)}\mathbf{V}^{(k)\,T}$, and let $\mathbf{U}^{(k+1)} \in \mathbb{R}^{n_1 \times r}$, $\mathbf{\Sigma}^{(k+1)} \in \mathbb{R}^{r \times r}$, and $\mathbf{V}^{(k+1)} \in \mathbb{R}^{n_2 \times r}$ be matrices consists of the top r left singular vectors/singular values/right singular vectors of $\mathbf{L}^{(k)} - \eta\mathbf{\Omega}^{(k)}$.
3(b): (Option 2) Let $\mathbf{Q}_1, \mathbf{R}_1$ be the QR decomposition of $(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})^T\mathbf{U}^{(k)}$ and $\mathbf{Q}_2, \mathbf{R}_2$ be the QR decomposition of $(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})\mathbf{V}^{(k)}$. Then $\mathbf{U}^{(k+1)} = \mathbf{Q}_2$, $\mathbf{V}^{(k+1)} = \mathbf{Q}_1$ and $\mathbf{\Sigma}^{(k+1)} = \mathbf{R}_2[\mathbf{U}^{(k)\,T}(\mathbf{L}^{(k)} - \eta\mathbf{D}^{(k)})\mathbf{V}^{(k)}]^{-1}\mathbf{R}_1^T$.
4: $k \coloneqq k + 1$.
Output: Estimation of the low-rank matrix $\mathbf{L}^*$, given by $\lim_{k\to\infty} \mathbf{L}^{(k)}$.

which is similar to (2) but only the observed entries are considered. The implementation is presented in Algorithm 2 and its derivation is deferred to Section 3.3.2.

For Algorithm 1, its memory usage is $O(n_1 n_2)$ due to the storage of $\mathbf{Y}$. For Algorithm 2, storing $\mathbf{Y}$ and $\mathbf{L}^{(k)}$ requires $O(|\mathbf{\Phi}|)$ and storing $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ requires $O(r(n_1+n_2))$. Adding them together, the memory usage is $O(|\mathbf{\Phi}| + r(n_1 + n_2))$.

For both Algorithm 1 and Algorithm 2 with Option 1, the singular value decomposition is the most computationally intensive step and as a result, the complexity per iteration is $O(rn_1n_2)$. For Algorithm 1 and Algorithm 2 with Option 2, their computational complexities per iteration are in the order of $O(rn_1n_2)$ and $O(r^2(n_1 + n_2) + r|\mathbf{\Phi}|)$ respectively.

3. Derivation of the Proposed Algorithms

This section gives the derivations of Algorithms 1 and 2. Since they are derived from manifold optimization, we first give a review of manifold optimization in Section 3.1 and the geometry of the manifold of low-rank matrices in Section 3.2.

3.1. Manifold optimization

The purpose of this section is to review the framework of the gradient descent method on manifolds. It summarizes mostly the framework used in Vandereycken (2013); Shalit et al. (2012); Absil et al. (2009), and we refer readers to these work for more details.

Given a smooth manifold $\mathcal{M} \subset \mathbb{R}^n$ and a differentiable function $f : \mathcal{M} \to \mathbb{R}$, the procedure of the gradient descent algorithm for solving $\min_{x\in\mathcal{M}} f(x)$ is as follows:

Step 1. Consider $f(x)$ as a differentiable function from $\mathbb{R}^n$ to $\mathbb{R}$ and calculate the Euclidean gradient $\nabla f(x)$.

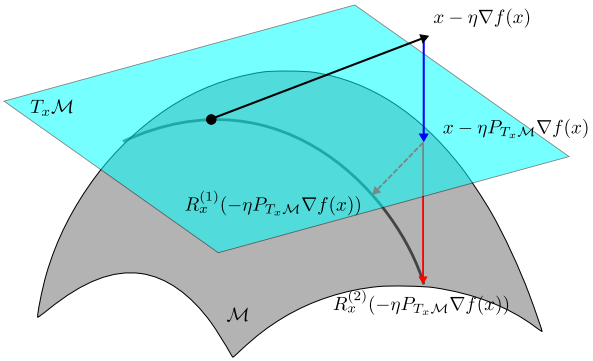


Figure 1: The visualization of gradient descent algorithms on the manifold $\mathcal{M}$. The black solid line is the Euclidean gradient. The blue solid line is the projection of the Euclidean gradient to the tangent space. The red solid line represents the orthographic retraction, while the red dashed line represents the projective retraction.

Step 2. Calculate its Riemannian gradient, which is the direction of steepest ascent of $f(x)$ among all directions in the *tangent space* $T_x\mathcal{M}$. This direction is given by $P_{T_x\mathcal{M}}\nabla f(x)$, where $P_{T_x\mathcal{M}}$ is the projection operator to the tangent space $T_x\mathcal{M}$.

Step 3. Define a *retraction* R_x that maps the tangent space back to the manifold, i.e. $R_x : T_x\mathcal{M} \to \mathcal{M}$, where R_x needs to satisfy the conditions in (Vandereycken, 2013, Definition 2.2). In particular, $R_x(0) = x$, $R_x(y) = x + y + O(\|y\|^2)$ as $y \to 0$, and R_x needs to be smooth. Then the update of the gradient descent algorithm x^+ is defined by

$$x^+ = R_x(-\eta P_{T_x\mathcal{M}}\nabla f(x)), \tag{6}$$

where η is the step size.

We remark that in differential geometry, the standard "retraction" is the exponential map from the tangent space to the manifold. However, in this work (as well as many works on manifold optimization) it is used to represent a generic mapping from the tangent plane to the manifold. As a result, the definition of retraction is not unique in this work. In Figure 1, we visualize the gradient descent method on the manifold $\mathcal{M}$ with two different kinds of retractions (orthographic and projective). We will discuss the details of those two retractions in Section 3.2.

3.2. The geometry of the manifold of low-rank matrices

To apply the gradient descent algorithm in Section 3.1 to the manifold of the low-rank matrices, the projection $P_{T_x\mathcal{M}}$ and the retraction R_x need to be defined. In this section, we let $\mathcal{M}$ be the manifold of all $\mathbb{R}^{n_1\times n_2}$ matrices with rank r and $\mathbf{X}\in\mathcal{M}$ be a matrix of rank r and will find the explicit expressions of $P_{T_x\mathcal{M}}$ and R_x.

The tangent space $T_{\mathbf{X}}\mathcal{M}$ and the retraction $R_{\mathbf{X}}$ of the manifold of the low-rank matrices have been well-studied Absil and Oseledets (2015): Assume that the SVD decomposition of $\mathbf{X}$ is $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, then the tangent space $T_{\mathbf{X}}\mathcal{M}$ can be defined by $T_{\mathbf{X}}\mathcal{M}=\{\mathbf{A}\mathbf{V}\mathbf{V}^T+\mathbf{U}\mathbf{U}^T\mathbf{B}\ :\ \text{for } \mathbf{A}\in\mathbb{R}^{n_1\times n_1},\mathbf{B}\in\mathbb{R}^{n_2\times n_2}\}$ according to Absil and Oseledets (2015). The explicit formula for the projection $P_{T_{\mathbf{X}}\mathcal{M}}$ is given in (Absil and Oseledets, 2015, (9)):

$$P_{T_{\mathbf{X}}\mathcal{M}}(\mathbf{D})=\mathbf{U}\mathbf{U}^T\mathbf{D}+\mathbf{D}\mathbf{V}\mathbf{V}^T-\mathbf{U}\mathbf{U}^T\mathbf{D}\mathbf{V}\mathbf{V}^T,\qquad \mathbf{D}\in\mathbb{R}^{n_1\times n_2}. \tag{7}$$

For completeness, a proof of (7) is presented in Appendix.

There are various ways of defining retractions for the manifold of low-rank matrices, and we refer the reader to Absil and Oseledets (2015) for more details. In this work, we consider two types of retractions. One is called the *projective* retraction Shalit et al. (2012); Vandereycken (2013), Given any $\delta\in T_{\mathbf{X}}\mathcal{M}$, the retraction is defined as the nearest low-rank matrix to $\mathbf{X}+\delta$ in terms of Frobenius norm:

$$R^{(1)}_{\mathbf{X}}(\delta)=\underset{\mathbf{Z}\in\mathcal{M}}{\arg\min}\|\mathbf{X}+\delta-\mathbf{Z}\|_F. \tag{8}$$

The solution is the rank-r approximation of $\mathbf{X}+\delta$ (for any matrix $\mathbf{W}$, its rank-r approximation is given by $\sum_{i=1}^r\sigma_i\mathbf{u}_i\mathbf{v}_i^T$, where $\sigma_i,\mathbf{u}_i,\mathbf{v}_i$ are the ordered singular values and vectors of $\mathbf{W}$).

In order to further improve computation efficiency, we also consider the *orthographic* retraction Absil and Oseledets (2015). Denoted by $R^{(2)}_{\mathbf{X}}(\delta)$, it is the nearest rank-r matrix to $\mathbf{X}+\delta$ that their difference is orthogonal to the tangent space $T_{\mathbf{X}}\mathcal{M}$:

$$R^{(2)}_{\mathbf{X}}(\delta)=\underset{\mathbf{Z}\in\mathcal{M}}{\arg\min}\|\mathbf{X}+\delta-\mathbf{Z}\|_F,\ \text{ s.t. }\ \langle R^{(2)}_{\mathbf{X}}(\delta)-(\mathbf{X}+\delta),\mathbf{Z}\rangle_F=0\ \text{ for all }\ \mathbf{Z}\in T_{\mathbf{X}}\mathcal{M}, \tag{9}$$

and its explicit solution of (9) is given in (Absil and Oseledets, 2015, Section 3.2),

$$R^{(2)}_{\mathbf{X}}(\delta)=(\mathbf{X}+\delta)\mathbf{V}[\mathbf{U}^T(\mathbf{X}+\delta)\mathbf{V}]^{-1}\mathbf{U}^T(\mathbf{X}+\delta), \tag{10}$$

and a proof is given in Appendix.

3.3. Derivation of the proposed algorithms

3.3.1. Derivation of Algorithm 1

The gradient descent algorithm (6) for solving (2) can be written as

$$\mathbf{L}^{(k+1)}=R_{\mathbf{L}^{(k)}}(-\eta P_{T_{\mathbf{L}^{(k)}}}\nabla f(\mathbf{L}^{(k)})), \tag{11}$$

where $P_{T_{\mathbf{L}^{(k)}}}$ is defined in (7) and $R_{\mathbf{L}^{(k)}}$ is defined in (8) or (10). To derive the explicit algorithm, it remains to find the gradient ∇f.

If the absolute values of all entries of $\mathbf{A}$ are different, then we have

$$\nabla f(\mathbf{L})=F(\mathbf{L}-\mathbf{Y}). \tag{12}$$

The proof of (12) is deferred to Appendix. When some entries of $\mathbf{A}$ are equivalent and there is a tie in generating $F(\mathbf{L}-\mathbf{Y})$, the objective function could be non-differentiable. However, it can be shown that by arbitrarily breaking the tie, $F(\mathbf{L}-\mathbf{Y})$ is a subgradient of $f(\mathbf{L})$.

The corresponding gradient descent method (or subgradient method when f is not differentiable) with projective retraction can be written as follows:

$$\mathbf{L}^{(k+1)}:=\text{rank-}r\text{ approximation of }\left[\mathbf{L}^{(k)}-\eta P_{T_{\mathbf{L}^{(k)}}}F(\mathbf{L}^{(k)}-\mathbf{Y})\right], \tag{13}$$

where the rank-r approximation has been defined after (8). This leads to Algorithm 1 with Option 1.

For the orthographic retraction, i.e., $R_{\mathbf{L}^{(k)}}$ defined according to (10), by writing $\mathbf{D}=F(\mathbf{L}^{(k)}-\mathbf{Y})$, the update formula (11) can be simplified to

$$\mathbf{L}^{(k+1)}:=(\mathbf{L}^{(k)}-\eta\mathbf{D})\mathbf{V}^{(k)}[\mathbf{U}^{(k)T}(\mathbf{L}^{(k)}-\eta\mathbf{D})\mathbf{V}^{(k)}]^{-1}\mathbf{U}^{(k)T}(\mathbf{L}^{(k)}-\eta\mathbf{D}), \tag{14}$$

where $\mathbf{U}^{(k)}\in\mathbb{R}^{n_1\times r}$ is any matrix such that its column space is the same as the column space of $\mathbf{L}^{(k)}$; and $\mathbf{V}^{(k)}\in\mathbb{R}^{n_2\times r}$ is any matrix such that its column space is the same as the row space of $\mathbf{L}^{(k)}$. The derivation of (14) is deferred to Appendix, and it can be shown that the implementation of (14) leads to Algorithm 1 with Option 2.

3.3.2. Derivation of Algorithm 2

By a similar argument as in the previous section, we can conclude that when all entries of $|\mathbf{L}-\mathbf{Y}|$ are different from each other, then applying the same procedure of deriving (12), we have

$$\nabla\tilde{f}(\mathbf{L})=\tilde{F}(\mathbf{L}-\mathbf{Y});$$

and $\tilde{F}(\mathbf{L}-\mathbf{Y})$ is a subgradient when $\tilde{f}(\mathbf{L})$ is not differentiable. Based on this observation, the algorithm under the partially observed setting is identical to (13) or (14), with F replaced by $\tilde{F}$. This gives the implementation of Algorithm 2.

3.3.3. Basic convergence properties of Algorithms 1 and 2

An interesting topic is that, can we still expect the algorithm to have reasonable basic properties, such as convergence to a critical point? Unfortunately, it is impossible to have such a theoretical guarantee if a fixed step size η is chosen: in general, the subgradient method with fixed step size does not have the convergence guarantee if the objective function is non-differentiable. However, if we choose step size with line search, then any accumulation point of $\mathbf{L}^{(k)}$, $\widehat{\mathbf{L}}$, would have the property that either the objective function is not differentiable at $\widehat{\mathbf{L}}$, or it is a critical point in the sense that its Riemannian gradient is zero. For example, the line search strategy for Algorithm 1 can be described as follows: start the step size η_k with a relatively large value, and repeatedly shrinks it by a factor of $\beta\in(0,1)$ such that the following condition is satisfied: for $\mathbf{L}^{(k+1)}=R_{\mathbf{L}^{(k)}}(-\eta_k P_{T_{\mathbf{L}^{(k)}}}\nabla f(\mathbf{L}^{(k)}))$,

$$f(\mathbf{L}^{(k)})-f(\mathbf{L}^{(k+1)})>c\eta_k\|P_{T_{\mathbf{L}^{(k)}}}\nabla f(\mathbf{L}^{(k)})\|^2,$$

where $c \in (0,1)$ is prespecified. The argument for convergence follows from the same argument as the proof of (Absil et al., 2009, Theorem 4.3.1).

3.4. Prior works on manifold optimization

The idea of optimization on manifolds has been well investigated in the literature Vandereycken (2013); Shalit et al. (2012); Absil et al. (2009). For example, Absil et al. Absil et al. (2009) give a summary of many advances in the field of optimization on manifolds. Manifold optimization has been applied to many matrix estimation problems, including recovering a low rank matrix from its partial entries, i.e., matrix completion Keshavan et al. (2010); Vandereycken (2013); Wei et al. (2016) and robust matrix completion in Cambier and Absil (2016). In fact, the problem studied in this work can be reformulated to the problem analyzed in Cambier and Absil (2016). In comparison, our work studies a different algorithm and gives additional theoretical guarantees.

In another aspect, while Wei et al. (2016) studies matrix completion, it shares some similarities with this work: both works study manifold optimization algorithms and have theoretical guarantees showing that the proposed algorithms can recover the underlying low-rank matrix exactly. In fact, Wei et al. (2016) can be considered as our problem under the partially observed setting, without corruption $\mathbf{S}^*$. It proposes to solve

$$\underset{\mathbf{L}\in\mathbb{R}^{n_1\times n_2},\mathrm{rank}(\mathbf{L})=r}{\arg\min} \sum_{(i,j)\in\Phi} (\mathbf{Y}_{ij} - \mathbf{L}_{ij})^2,$$

which can be considered as $\tilde{f}$ in (5) when $\gamma = 0$.

4. Theoretical Analysis

In this section, we analyze the theoretical properties of Algorithms 1 and 2 and compare them with previous algorithms. Since the goal is to recover the low-rank matrix $\mathbf{L}^*$ and the sparse matrix $\mathbf{S}^*$ from $\mathbf{Y} = \mathbf{L}^* + \mathbf{S}^*$, to avoid identifiability issues, we need to assume that $\mathbf{L}^*$ can not be both low-rank and sparse. Specifically, we make the following standard assumptions on $\mathbf{L}^*$ and $\mathbf{S}^*$:

Assumption 1 *Each row of $\mathbf{S}^*$ contains at most $\gamma^* n_2$ nonzero entries and each column of $\mathbf{S}^*$ contains at most $\gamma^* n_1$ nonzero entries. In other words, for $\gamma^* \in [0,1)$, assume $\mathbf{S}^* \in \mathcal{S}_{\gamma^*}$ where*

$$\mathcal{S}_{\gamma^*} \coloneqq \left\{ A \in \mathbb{R}^{n_1\times n_2} \mid \|A_{i,\cdot}\|_0 \le \gamma^* n_2, \text{ for } 1 \le i \le n_1; \|A_{\cdot,j}\|_0 \le \gamma^* n_1, \text{ for } 1 \le j \le n_2 \right\}. \tag{15}$$

Assumption 2 *The low-rank matrix $\mathbf{L}^*$ is not near-sparse. To achieve this, we require that $\mathbf{L}^*$ must be μ-coherent. Given the singular value decomposition (SVD) $\mathbf{L}^* = \mathbf{U}^*\mathbf{\Sigma}^*\mathbf{V}^{*T}$, where $\mathbf{U}^* \in \mathbb{R}^{n_1\times r}$ and $\mathbf{V}^* \in \mathbb{R}^{n_2\times r}$, we assume there exists an incoherence parameter μ such that*

$$\|\mathbf{U}^*\|_{2,\infty} \le \sqrt{\frac{\mu r}{n_1}}, \ \ \|\mathbf{V}^*\|_{2,\infty} \le \sqrt{\frac{\mu r}{n_2}}, \tag{16}$$

where the norm $\|\cdot\|_{2,\infty}$ is defined by $\|\mathbf{A}\|_{2,\infty} = \max_{\|\mathbf{z}\|_2=1} \|\mathbf{A}\mathbf{z}\|_\infty$ and $\|\mathbf{x}\|_\infty = \max_i |\mathbf{x}_i|$.

4.1. Analysis of Algorithm 1

With Assumption 1 and 2, we have the following theoretical results regarding the convergence rate, initialization, and stability of Algorithm 1:

Theorem 1 (Linear convergence rate, fully observed case) *Suppose that $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F \le a\sigma_r(\mathbf{L}^*)$, where $\sigma_r(\mathbf{L}^*)$ is the r-th largest singular value of $\mathbf{L}^*$, $a \le 1/2$, $\gamma > 2\gamma^*$ and $C_1 = \sqrt{4(\gamma + 2\gamma^*)\mu r + 4\frac{\gamma^*}{\gamma-\gamma^*} + a^2} < \frac{1}{2}$, then there exists $\eta_0 = \eta_0(C_1, a) > 0$ that does not depend on k, such that for all $\eta \le \eta_0$,*

$$\|\mathbf{L}^{(k)} - \mathbf{L}^*\|_F \le \left(1 - \frac{1-2C_1}{8}\eta\right)^k \|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F.$$

Remark 2 *(Choices of parameters). It is shown in the proof that η_0 can be set to the solution of the equation*

$$\eta_0(1+C_1)^2\left[\frac{1}{2} + \frac{a^2}{1-\eta_0(1+C_1)a}\right] = \frac{1}{8}(1-2C_1).$$

Since the LHS is an increasing function of η_0 and is zero when $\eta_0 = 0$, and its RHS is a positive number.

While $C_1 < 1/2$ requires $\sqrt{4\gamma^*/(\gamma-\gamma^*)} < 1/2$, i.e., $\gamma > 17\gamma^*$. In practice a much smaller γ can be used. In Section 5, $\gamma = 1.5\gamma^*$ is used and works well for a large number of examples. It suggests that some constants in Theorem 1 might be due to the technicalities in the proof and can be potentially improved.

Remark 3 *(Simplified choices of parameters) There exists c_1 and c_2 such that if $a < c_1$, $\gamma^*\mu r < c_2$ and $\gamma = 65\gamma^*$, then one can choose $\eta_0 = 1/8$. In this sense, if the initialization of the algorithm is good, then the algorithm can handle γ^* as large as $O(1/\mu r)$. In addition, it requires $O(\log(1/\epsilon))$ iterations to achieve $\|\mathbf{L}^{(k)} - \mathbf{L}^*\|_F / \|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F < \epsilon$.*

Since the statements require proper initializations (i.e., small a), the question arises as to how to choose proper initializations. The work by Yi et al. (2016) shows that if the rank-r approximation to $F(\mathbf{Y})$ is used as the initialization $\mathbf{L}^{(0)}$, then such initialization has the upper bound $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|$ according to the proofs of (Yi et al., 2016, Theorems 1 and 3) (we borrow this estimation along with the fact that $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F \le \sqrt{2r}\|\mathbf{L}^{(0)} - \mathbf{L}^*\|$).

Theorem 4 (Initialization, fully observed case) *If $\gamma > \gamma^*$ and we initialize $\mathbf{L}^{(0)}$ using the rank-r approximation to $F(\mathbf{Y})$, then*

$$\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F \le 8\gamma\mu r\sqrt{2r}\sigma_1(\mathbf{L}^*).$$

The combination of Theorem 1, 4 and the fact that $\gamma = O(\gamma^*)$ implies that under the fully observed setting, the tolerance of the proposed algorithms to corruption is at most $\gamma^* = O(\frac{1}{\mu r\sqrt{r}\kappa})$, where $\kappa = \sigma_1(\mathbf{L}^*)/\sigma_r(\mathbf{L}^*)$ is the condition number of $\mathbf{L}^*$. We also study the stability of Algorithm 1 in the following statement.

Theorem 5 (Stability, fully observed case) *Let $\mathbf{L}$ be the current value, and let $\mathbf{L}^+$ be the next update by applying Algorithm 1 to $\mathbf{L}$ for one iteration. Assuming $\mathbf{Y} = \mathbf{L}^* + \mathbf{S}^* + \mathbf{N}^*$, where $\mathbf{N}^*$ is a random Gaussian noise i.i.d. sampled from $N(0, \sigma^2)$, $\gamma > 10\gamma^*$ and $(\gamma + 2\gamma^*)\mu r < 1/64$, then there exist $C, a, c, \eta_0 > 0$ such that when $\eta < \eta_0$,*

$$P\left(\|\mathbf{L}^+ - \mathbf{L}^*\|_F \le (1 - c\eta)\,\|\mathbf{L} - \mathbf{L}^*\|_F \text{ for all } \mathbf{L} \in \Gamma\right) \to 1, \quad \text{as } n_1, n_2 \to \infty, \tag{17}$$

where

$$\Gamma = \{\mathbf{L} \in \mathbb{R}^{n_1 \times n_2} : \text{rank}(\mathbf{L}) = r, C\sigma\sqrt{(n_1 + n_2)r\ln(n_1 n_2)} \le \|\mathbf{L} - \mathbf{L}^*\|_F \le a\sigma_r(\mathbf{L}^*)\|,$$

Since $1 - c\eta < 1$, and Theorem 5 shows that when the observation $\mathbf{Y}$ is contaminated with a random Gaussian noise, if $\mathbf{L}^{(0)}$ is properly initialized such that $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F < a\sigma_r(\mathbf{L}^*)$, Algorithms 1 will converge to a neighborhood of $\mathbf{L}^*$ given by

$$\{\mathbf{L} : \|\mathbf{L} - \mathbf{L}^*\|_F \le C\sigma\sqrt{(n_1 + n_2)r\ln(n_1 n_2)}\}$$

in $[-\log(\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F) + \log(C\sigma\sqrt{(n_1 + n_2)r\ln(n_1 n_2)})]/\log(1 - c\eta)$ iterations, with probability goes to 1 as $n_1, n_2 \to \infty$.

4.2. Analysis of Algorithm 2

For the partially observed setting, we assume that each entry of $\mathbf{Y} = \mathbf{L}^* + \mathbf{S}^*$ is observed with probability p. That is, for any $1 \le i \le n_1$ and $1 \le j \le n_2$, $\Pr((i, j) \in \mathbf{\Phi}) = p$. Then we have the following statement on convergence:

Theorem 6 (Linear convergence rate, partially observed case) *There exists $c > 0$ such that for $n = \max(n_1, n_2)$, if $p \ge \max(c\mu r \log(n)/\epsilon^2 \min(n_1, n_2), \frac{56}{3}\frac{\log n}{\gamma \min(n_1, n_2)})$, then with probability $1 - 2n^{-3} - 6n^{-1}$,*

$$\begin{aligned}\frac{\|\mathbf{L}^{(k)} - \mathbf{L}^*\|_F}{\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F} \le &\Big[\sqrt{1 - p^2(1-\epsilon)^2\left(2\eta\left(1 - \tilde{C}_1 - \frac{ap(1+\epsilon)}{2(1-a)}(1 + \tilde{C}_1)\right) - \eta^2(1 + \tilde{C}_1)^2\right)} \\ &+ \frac{\eta^2 a^2 (p + p\epsilon)^2 (1 + \tilde{C}_1)^2}{1 - \eta a(p + p\epsilon)(1 + \tilde{C}_1)}\Big]^k\end{aligned} \tag{18}$$

for

$$\tilde{C}_1 = \frac{1}{p(1-\epsilon)}\Big[6(\gamma + 2\gamma^*)p\mu r + 4\frac{3\gamma^*}{\gamma - 3\gamma^*}(\sqrt{p(1+\epsilon)} + \frac{a}{2})^2 + a^2\Big].$$

Remark 7 *(Choice of parameters) Note that when η is small, the RHS of (18) is in the order of*

$$1 - p^2(1-\epsilon)^2\left(1 - \tilde{C}_1 - \frac{ap(1+\epsilon)}{2(1-a)}(1 + \tilde{C}_1)\right)\eta + O(\eta^2).$$

As a result, to make sure that the RHS of (18) to be smaller than 1 for small η, we assume that

$$1 - \tilde{C}_1 - \frac{ap(1+\epsilon)}{2(1-a)}(1 + \tilde{C}_1) > 0. \tag{19}$$

For example, when $ap(1 + \epsilon) = 4(1 - a)$, it requires that $\tilde{C}_1 < 1/3$. If (19) holds, then there exists $\eta_0 = \eta_0(\tilde{C}_1, p, \epsilon, a)$ such that for all $\eta \le \eta_0$, the RHS of (18) is smaller than 1.

The practical choices of η and γ will be discussed in Section 5.

Remark 8 *(Simplified choice of parameters) There exists $\{c_i\}_{i=1}^4 > 0$ such that when $\epsilon < 1/2$, $a < c_1 p$, $\gamma^*\mu r < c_2$ and $\gamma = c_3\gamma^*$, then when $\eta < 1/8$,*

$$\frac{\|\mathbf{L}^{(k)} - \mathbf{L}^*\|_F}{\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F} \le (1 - c_4\eta p^2)^k.$$

Compared with the result in Theorem 1, the addition parameter p appears in both the initialization requirement $a < c_1 p$ as well as the convergence rate. This makes the result weaker, but we suspect that the dependence on the subsampling ratio p could be improved through a better estimation in (39) and the estimation of $\tilde{C}_1$ in Lemma 16, and we leave it as a possible future direction.

We present a method of obtaining a proper initialization in Theorem 9. Combining it with Theorem 6, Algorithm 2 allows the corruption level γ^* to be in the order of $O(\frac{p}{\mu r\sqrt{r\kappa}})$.

Theorem 9 (Initialization, partially observed case) *There exists $c_1, c_2, c_3 > 0$ such that if $\gamma > 2\gamma^*$, and $p \ge c_2(\frac{\mu r^2}{\epsilon^2} + \frac{1}{\alpha})\log n/\min(n_1, n_2)$, and we initialize $\mathbf{L}^{(0)}$ using the rank-r approximation to $F(\mathbf{Y})$, then*

$$\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F \le 16\gamma\mu r\sigma_1(\mathbf{L}^*)\sqrt{2r} + 2\sqrt{2}c_1\epsilon\sigma_1(\mathbf{L}^*)$$

with probability at least $1 - c_3 n^{-1}$, where $\sigma_1(\mathbf{L}^*)$ is the largest singular value of $\mathbf{L}^*$.

4.3. Comparison with Alternating Gradient Descent

Since our objective functions are equivalent to the objective functions of the alternating gradient descent (AGD) in Yi et al. (2016), it would be interesting to compare these two works. The only difference of these two works lies in the algorithmic implementation: our methods use the gradient descent on the manifold of low-rank matrices, while the methods in Yi et al. (2016) use alternating gradient descent on the factors of the low-rank matrix. In the following we compare the results of both works from four aspects:

1. **Accuracy of initialization.** What is the largest value t that the algorithm can tolerate, such that for any initialization $\mathbf{L}^{(0)}$ satisfying $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F \le t$, the algorithm is guaranteed to converge to $\mathbf{L}^*$?

2. **Convergence rate.** What is the smallest number of iteration steps k such that the algorithm reaches a given convergence criterion ϵ, i.e. $\|\mathbf{L}^{(k)} - \mathbf{L}^*\|_F/\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F < \epsilon$?

3. **Corruption level (perfect initialization).** Suppose that the initialization is in a sufficiently small neighborhood of $\mathbf{L}^*$ (i.e. there exists a very small $\epsilon_0 > 0$ such that $\mathbf{L}^{(0)}$ satisfies $\|\mathbf{L}^{(0)} - \mathbf{L}^*\|_F < \epsilon_0$), what is the maximum corruption level that can be tolerated in the convergence analysis?

4. **Corruption level (proper initialization).** Suppose that the initialization is given by the procedure in Theorem 4 (for the partially observed case) and 9 (for the fully observed case), what is the maximum corruption level that can be tolerated?

These comparisons are summarized in Table 1. We can see that under the full observed setting, our results remove or reduce the dependence on the condition number κ, while keeping other values unchanged. Under the partially observed setting our results still have the advantage of less dependence on κ, but sometimes require an additional dependence on the subsampling ratio p. The simulation results discussed in the next section also verify that when κ is large our algorithms have better performance, while that the slowing effect of p under the partially observed setting is not significant. As discussed after Theorem 6, we suspect that this dependence could be removed after a more careful analysis (or more assumptions).

Criterion	Accuracy of initialization	Convergence rate	Max corruption (perfect init.)	Max corruption (proper init.)
Algorithm 1	$O(\sigma_r(\mathbf{L}^*))$	$O(\log(\frac{1}{\epsilon}))$	$O(\frac{1}{\mu r})$	$O(\frac{1}{\mu r^{1.5}\kappa})$
APG (full)	$O(\frac{\sigma_r(\mathbf{L}^*)}{\sqrt{\kappa}})$	$O(\kappa\log(\frac{1}{\epsilon}))$	$O(\frac{1}{\kappa^2\mu r})$	$O(\frac{1}{\max(\mu r^{1.5}\kappa^{1.5},\kappa^2\mu r)})$
Algorithm 2	$O(\sqrt{p}\sigma_r(\mathbf{L}^*))$	$O(\log(\frac{1}{\epsilon})/p^2)$	$O(\frac{1}{\mu r})$	$O(\frac{\sqrt{p}}{\mu r^{1.5}\kappa})$
APG (partial)	$O(\frac{\sigma_r(\mathbf{L}^*)}{\kappa})$	$O(\kappa\mu r\log(\frac{1}{\epsilon})$	$O(\frac{1}{\kappa^2\mu r})$	$O(\frac{1}{\max(\mu r^{1.5}\kappa^{1.5},\kappa^2\mu r)})$

Table 1: Comparison of the theoretical guarantees in our work and the alternating gradient descent algorithm in Yi et al. (2016). The four criteria are explained in details in Section 4.3.

Here we use a simple example to give some intuition of why our proposed methods work better than gradient descent method based on factorization. Let us consider the following simple optimization problem:

$$\arg\min_{\mathbf{z}\in\mathbb{R}^m} f(\mathbf{z}),\ \text{ where } \mathbf{z}=\mathbf{A}\mathbf{x}+\mathbf{B}\mathbf{y},$$

where $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$ and $\mathbf{A},\mathbf{B}\in\mathbb{R}^{m\times n}$. In this example, $(\mathbf{x},\mathbf{y})$ can be considered as the "factors" of $\mathbf{z}$. The gradient descent method on the factors $(\mathbf{x},\mathbf{y})$ is then given by

$$\mathbf{x}^+=\mathbf{x}-\eta\mathbf{A}^Tf'(\mathbf{z}),\ \ \mathbf{y}^+=\mathbf{y}-\eta\mathbf{B}^Tf'(\mathbf{z}). \tag{20}$$

Writing the update formula (20) in terms of $\mathbf{z}$, it becomes

$$\mathbf{z}^+=\mathbf{A}\mathbf{x}^++\mathbf{B}\mathbf{y}^+=\mathbf{z}-\eta(\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{B}^T)f'(\mathbf{z}).$$

As a result, the "gradient descent on factors $(\mathbf{x},\mathbf{y})$" has a direction of $-(\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{B}^T)f'(\mathbf{z})$. In comparison, the gradient descent on the variable $\mathbf{z}$ has a direction of $-f'(\mathbf{z})$, which is the direction that f decreases fastest. If $\mathbf{A}\mathbf{A}^T+\mathbf{B}\mathbf{B}^T$ is a matrix with a large condition number, then we expect that the gradient descent method on factors $(\mathbf{x},\mathbf{y})$ would not work well. This example shows that generally, compared to applying the gradient descent method to the factors of a variable, it is better to apply it to the variable itself. Similar to this example, our method applies gradient descent on the $\mathbf{L}$ itself while Yi et al. (2016) applies gradient descent to the factors of $\mathbf{L}$.

4.4. Comparison with other robust PCA algorithms

In this section we compare our result with other robust PCA methods and summarize them in Table 2. Some criterion in Table 1 are not included since they do not apply. For example, (Netrapalli et al., 2014, Alternating Projection) only analyzes the algorithm with specific initialization, and the criterion 1 and 3 in Table 1 do not apply to this work. As a result, we only compare the maximum corruption ratio that these methods can handle, and the computational complexity per iteration in Table 2. As for the convergence rate, it depends on assumptions on parameters such as the coherence parameter, rank, and the size of the matrix: The alternating projection Netrapalli et al. (2014) requires $10\log(4n_1\mu^2r\|\mathbf{Y}-\mathbf{L}^{(0)}\|_2/\epsilon\sqrt{n_1n_2})$ iterations to achieve an accuracy ϵ, under the assumptions that $\gamma^*<1/512\mu r^2$ and a tuning parameter is chosen to be $4\mu^2r\sqrt{n_1n_2}$. The alternating minimization method Gu et al. (2016) have the guarantee that if $\|\mathbf{L}^{(1)}-\mathbf{L}^*\|_2\le\sigma_1(\mathbf{L}^*)$, then

$$\|\mathbf{L}^{(k+1)}-\mathbf{L}^*\|_F\le\sigma_1\left(\frac{96\sqrt{2}\nu\mu\sqrt{r}(s^*/d)^{3/2}\kappa\sigma_1}{1-24\sqrt{2}\nu\mu\sqrt{r}(s^*/d)^{3/2}\kappa\sigma_1}\right)^k\|\mathbf{L}^{(1)}-\mathbf{L}^*\|_F,$$

where ν is a parameter concerning the coherence of $\mathbf{L}^*$, s^* is the number of nonzero entries in $\mathbf{S}^*$, $d=\min(n_1,n_2)$. As a result s^*/d is approximately $\gamma^*\max(n_1,n_2)$ in our notation. If ν is in the order of $O(1)$, then this results requires that $\mu\sqrt{r}\kappa\sigma_1\max(n_1,n_2)^{3/2}\gamma^{*3/2}\le O(1)$, which is more restrictive than our assumption in Theorem 1 that $\gamma^*\mu r\le O(1)$. Convex methods usually have convergence rate guarantees based on convexity, for example, the accelerated proximal gradient method Toh and Yun (2010) has a convergence rate of $O(1/k^2)$. While it is a slower convergence rate compared to the result in Theorem 1 in this work or the results in Netrapalli et al. (2014); Gu et al. (2016) and it does not necessarily converge to the correct solution, this result does not depend on any assumption on the low-rank matrix and the corruption ratio.

Criterion	Maximum corruption level	Complexity per iteration
Algorithm 1	$O(1/\kappa\mu r^{3/2})$	$O(rn_1n_2)$
Convex methods	$O(1/\mu^2r)$	$O(n_1n_2\min(n_1,n_2))$
Netrapalli et al. (2014)	$1/512\mu^2r$	$O(r^2n_1n_2)$
Gu et al. (2016)	$O(1/\mu^{2/3}r^{2/3}\min(n_1,n_2))$	$O(r^2n_1n_2)$
Yi et al. (2016)	$O(1/\kappa^2\mu r^{3/2})$	$O(rn_1n_2)$

Table 2: Comparison of the theoretical guarantees in our work and some other robust PCA algorithms.

The stability in Theorem 5 is comparable to analysis in Netrapalli et al. (2014) (the works Gu et al. (2016) and (Yi et al., 2016, Alternating Gradient Descent) do not have stability analysis). The work Netrapalli et al. (2014) assumes that $\|\mathbf{N}^*\|_\infty<\sigma_r(\mathbf{L}^*)/100n_2$ and proves that the output of their algorithm $\widehat{\mathbf{L}}$ satisfies

$$\|\widehat{\mathbf{L}}-\mathbf{L}^*\|_F\le\epsilon+2\mu^2r\left(7\|\mathbf{N}^*\|_2+\frac{8\sqrt{n_1n_2}}{\sqrt{r}}\|\mathbf{N}^*\|_\infty\right),$$

where ϵ is the error of the algorithm when there is no noise. If $\mathbf{N}^*$ is i.i.d. sampled from $N(0,\sigma^2)$, this result suggests that $\|\widehat{\mathbf{L}}-\mathbf{L}^*\|_F$ is bounded above by $\epsilon + O(\mu^2\sqrt{rn_1n_2}\sigma)$. In comparison, Theorem 5 suggests that after a few iterations, $\|\mathbf{L}^{(k)}-\mathbf{L}^*\|_F$ is bounded above by $C\sigma\sqrt{(n_1+n_2)r\ln(n_1n_2)}$ with high probability, which is a tighter upper bound.

5. Simulations

In this section, we test the performance of the proposed algorithms by simulated data sets and real data sets. The MATLAB implementation of our algorithm used in this section is available at `https://sciences.ucf.edu/math/tengz/`. For simulated data sets, we generate $\mathbf{L}^*$ by $\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{U}\in\mathbb{R}^{n_1\times r}$ and $\mathbf{V}\in\mathbb{R}^{n_2\times r}$ are random matrices that i.i.d. sampled from $N(0,1)$, and $\mathbf{\Sigma}\in\mathbb{R}^{n_1\times r}$ is an diagonal matrix. As for $\mathbf{S}^*\in\mathbb{R}^{n_1\times n_2}$, each entry is sampled from $N(0,100)$ with probability q, and is zero with probability $1-q$. That is, q represents the level of sparsity in the sparse matrix $\mathbf{S}^*$. It measures the overall corruption level of $\mathbf{Y}$ and is associated with the corruption level γ^* (γ^* measures the row and column-wise corruption level). For the partially observed case, we assume that each entry of $\mathbf{Y}$ is observed with probability p.

5.1. Choice of parameters

We first investigate the performance of the proposed algorithms, in particular, the dependence on the parameters η and γ. In simulations, we let $[n_1,n_2]=[500,600]$, $r=3$, $\mathbf{\Sigma}=\mathbf{I}$, and $q=0.02$. For the partially observed case, we let $p=0.2$.

The first simulation investigates the following questions:

- Should we use the Algorithms 1 and 2 with Option 1 or Option 2?
- What is the appropriate choice of the step size η?

The simulation results for Option 1 an 2 with various step sizes are visualized in Figure 2, which show that the two options perform similarly. Usually the algorithms converge faster when the step size is larger. However, if the step size is too large then it might diverge. As a result, we use the step size $\eta=0.7$ for Algorithm 1 and $0.7/p$ for Algorithm 2 for the following simulations.

The second simulation concerns the choice of γ. We test $\gamma=c\gamma^*$ for a few choices of c (γ^* can be calculated from the zero pattern of $\mathbf{S}$). Figure 5.1 shows that if γ is too small, for example, $0.5\gamma^*$, then the algorithm fail to converge to the correct solution; and if γ is too large, then the convergence is slow. Following these observations, we use $\gamma=1.5\gamma^*$ as the default choice of the following experiments, which is also used in Yi et al. (2016).

5.2. Performance of the proposed algorithm

In this section, we analyze the convergence behavior as the parameters (overall ratio of corrupted entries q, condition number κ, rank r, subsampling ratio p) changes and visualized the result in Figure 4.

Figure 4(a) shows the simulation for corruptions level q, we use the setting in Section 5.1, but replace the corruption level q by $q=0.1,0.2,0.3,0.4$. Figure 4 shows that the algorithm

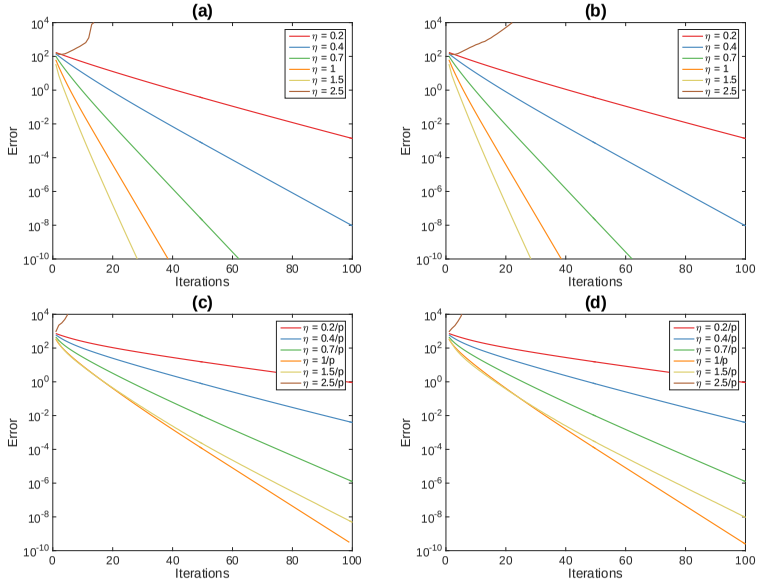


Figure 2: The dependence of the estimation error on the number of iterations for different step sizes η (a) Algorithm 1 (Option 1); (b) Algorithm 1 (Option 2); (c) Algorithm 2 (Option 1); (d) Algorithm 2 (Option 2).

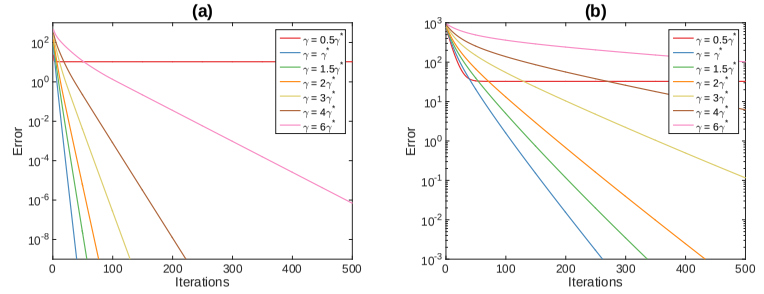


Figure 3: The convergence of the algorithm depending on the choice of γ. (a) fully observed setting; (b) partially observed setting.

converges slower with more corruption, which is expected since there is fewer information available. However, the algorithm still converges even with an overall corruption level at 0.4.

Figure 4(b) shows the simulation for rank r, we use the setting in Section 5.1, but replace r by $r = 3, 10, 30, 100, 300$ respectively. Simulations show that the algorithm works fine for rank $r = 3, 10, 30, 100$ and it converges slower for rank $r = 300$.

Figure 4(c) shows the simulation for condition number κ of $\mathbf{L}$, we use the setting in Section 5.1, but replace Σ by $\Sigma = \mathrm{diag}(1, 1, 1/\kappa)$ and try various values of κ. While the algorithm converges for κ up to 30 in the simulation, for larger κ the algorithm converges slowly at the beginning, and then decreases quickly to zero. We suspect that the initialization is not sufficiently good and it takes a while for the algorithm to reach the "local neighborhood of convergence". We also remark that $\mathbf{L}$ with a very large condition number, e.g. $\kappa = 100$, is generally challenging for any nonconvex optimization algorithm, as shown in Figure 5, Setting 4. It is because that when κ is large, the solution is close to a matrix with rank less than r – a singular point on the manifold of the matrices of rank r, which gives a geometry of manifold that is not "smooth" enough. We observe that when $\kappa = 100$, our algorithm performs well if the rank r is set to 2 (instead of the true value 3)—in fact, when $\kappa = 100$, the underlying matrix is approximately of rank 2 since the third singular value is very small.

We test the algorithm with various matrix sizes using the setting in Section 5.1 and set $[n_1, n_2] = [1000, 1200], [5000, 6000], [10000, 12000]$. Figure 4(d) shows that Algorithm 1 converges quickly for all of the choices within a few iterations.

In the last simulation, we test Algorithm 2 under the setting in Section 5.1 with various choices of the subsampling ratio p. Figure 4(e) shows suggest that the algorithm converges for p as small as 0.1, though the convergence rate is slow for small p.

5.3. Comparison with other robust PCA algorithms

In this section, we compare our algorithm with the accelerated proximal gradient method (APG) and the alternating direction method of multiplier (ADMM) based on convex relaxation (1); the robust matrix completion algorithm (RMC) Cambier and Absil (2016) based on manifold optimization problem

$$\underset{\mathrm{rank}(\mathbf{L})=r}{\arg\min} \sum_{(i,j)\in\Phi} \|\mathbf{L}_{ij} - \mathbf{Y}_{ij}\| + \lambda \sum_{(i,j)\notin\Phi} \mathbf{L}_{ij}^2,$$

as well as the alternating gradient descent method (AGD) in Yi et al. (2016) that solves the same optimization as this work, but with an implementation based on matrix factorization rather than manifold optimization. We use the implementation of APG from Toh and Yun (2010) and the implementation of ADMM from `https://github.com/dlaptev/RobustPCA`. In these two algorithms, we use the choice of parameter $\lambda = 1/\sqrt{\max(n_1, n_2)}$, which is the default choice in the implementation Toh and Yun (2010) and the theoretical analysis in Candès et al. (2011). For ADMM, the augmented Lagrangian parameter is set by default as 10λ. For RMC and GD, we use their default setting of parameters. Since the setting of Algorithm 2 does not apply to the implementations of APG/ADMM, we compare them under the fully observed setting. We compare them in the following four settings:

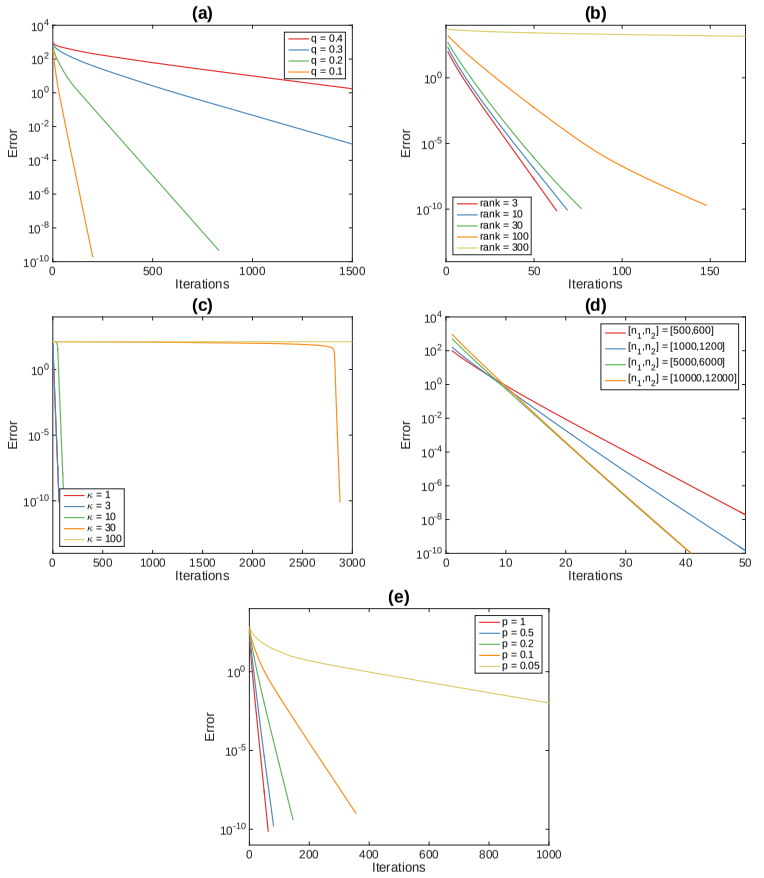


Figure 4: Dependence of the estimation error on the number of iterations for different (a) Overall ratios of corrupted entries q (Algorithm 1); (b) Ranks r (Algorithm 1); (c) Condition numbers κ (Algorithm 1); (d) Matrix sizes $[n_1, n_2]$ (Algorithm 1); (e) Subsampling ratio p (Algorithm 2).

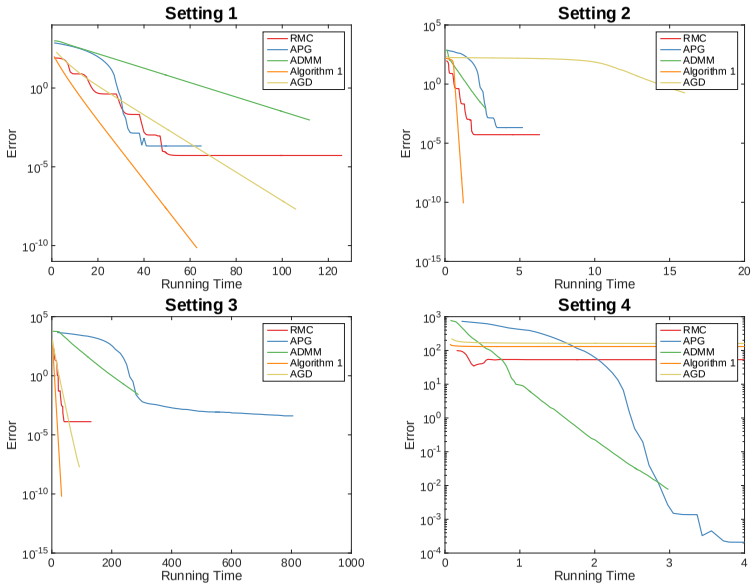


Figure 5: The comparison of the performance of the algorithms under the fully observed setting. The running time is measured in seconds.

- Setting 1: same setting as in Section 5.1.
- Setting 2 (large condition number): replace Σ by $\mathrm{diag}(1, 1, 0.1)$ in Setting 1.
- Setting 3 (large matrix): replace n_1 and n_2 by 3000 and 4000 in Setting 1.
- Setting 4 (large condition number): replace Σ by $\mathrm{diag}(1, 1, 0.01)$ in Setting 1.

Figure 5 shows that under Setting 1, 2 and 3, Algorithm 1 converges faster than other algorithms. In particular, the advantage over the AGD algorithm is very clear under Setting 2, where the condition number is larger. This verifies our theoretical analysis, where the convergence rate is faster than the analysis in Yi et al. (2016) by a factor of $\sqrt{\kappa}$. In Setting 3, the algorithms RMC, AGD and Algorithm 1 converge much faster than APG and ADMM, which verifies the computational advantage of nonconvex algorithms when the matrix size is large. However, in Setting 4, the convex algorithms converge to the correct solution while the nonconvex algorithms converge to a local minimizer that is different than the correct solution. This is due to the fact that the nonconvex algorithms have more than one minimizer, and if it is not initialized well then it could get trapped in local minimizers. In

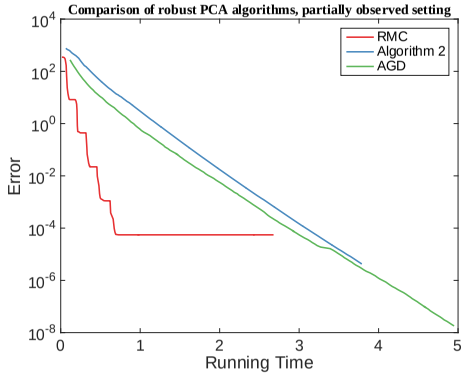


Figure 6: The comparison of the performances of the algorithms under the partially observed setting. The running time is measured in seconds.

practice, we observed that if the initialization is well chosen and close to the true $\mathbf{L}^*$, then Algorithm 1 converges quickly to the correct solution.

We also compare the performance of RMC, AGD and Algorithm 2 under the partially observed setting. We use Setting 1 with $p = 0.3$ and visualize the result in Figure 6. The results are similar to that of the fully observed setting: AGD and Algorithm 2 are comparable and RMC converges faster at the beginning, but then does not achieve higher accuracy, possibly due to their choice of the regularization parameter.

We also test the proposed algorithms in a real data application for video background subtraction. We adopt the public data set `Shoppingmall` studied in Yi et al. (2016),[1] A few frames are visualized in the first column of Figure 7. There are 1000 frames in this video sequence, represented by a matrix of size 81920×1000, where each column corresponds to a frame of the video and each row corresponds to a pixel of the video. We apply our algorithms with $r = 3$ and $\gamma^* = 0.1$, $p = 0.5$ for the partially observed case, the step size $\eta = 0.7$. We stop the algorithm after 100 iterations. Figure 7 shows that our algorithms obtain desirable low-rank approximations within 100 iterations.

In Figure 8, we compare our algorithms with APG in terms of the convergence of the objective function value. In this figure, the relative error is defined as $\|F(\mathbf{L}-\mathbf{Y})\|_F/\|\mathbf{Y}\|_F$, a scaled objective value. A smaller relative error implies a better low-rank approximation. Figure 8 shows out that our algorithms can obtain smaller objective values within 100 iterations under both fully observed and partially observed cases.

1. The data set is originally from `http://perception.i2r.a-star.edu.sg/bk_model/bk_index.html`, and is available at `https://sciences.ucf.edu/math/tengz/`.

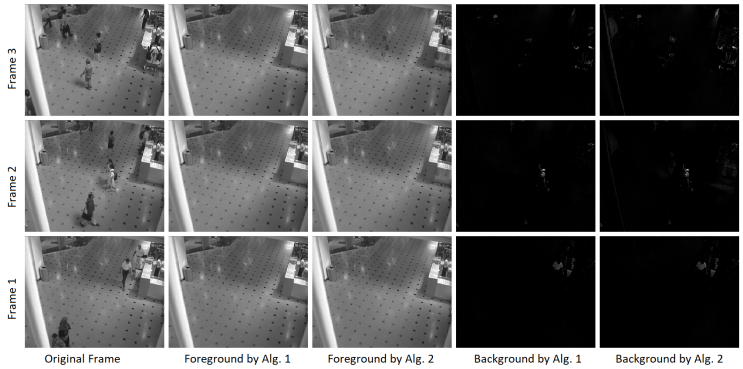


Figure 7: The performance of Algorithms 1 and 2 in video background subtraction, with three rows representing three frames in the video sequence. For Algorithm 2, a subsampling ratio of $p = 0.5$ is used.

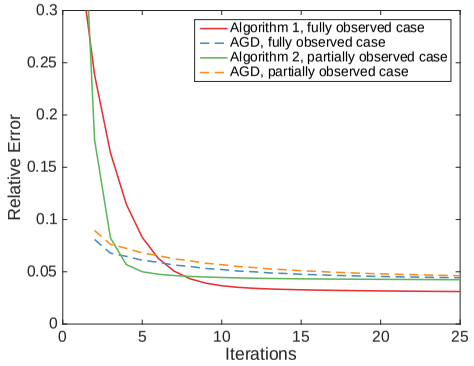


Figure 8: The relative error of Algorithms 1, 2, and AGD with respect to the iterations, for both fully observed case and partially observed case in the experiment of background subtraction.

6. Conclusion

This paper proposes two robust PCA algorithms (one for fully observed case and one for partially observed case) based on the gradient descent algorithm on the manifold of low-rank matrices. Theoretically, compared with the gradient descent algorithm with matrix factorization, our approach has a faster convergence rate, better tolerance of the initialization accuracy and corruption level. The approach removes or reduces the dependence of the algorithms on the condition number of the underlying low-rank matrix. Numerically, the proposed algorithms performance is less sensitive to the choice of step sizes. We also find that under the partially observed setting, the performance of the proposed algorithm is not significantly affected by the presence of the additional dependence on the observation probability. Considering the popularity of the methods based on matrix factorization, it is an interesting future direction to apply manifold optimization to other low-rank matrix estimation problems.

Acknowledgements

The authors thank the editor, an associate editor and referee for their helpful comments and suggestions. The authors also thank David Fleischer for providing helps on the coding. Yang's research is partially supported by NSERC RGPIN-2016-05174. Zhang's research is partially supported by National Science Foundation (NSF) grant CNS-1739736.

Appendix for "Robust PCA by Manifold Optimization"

A. Technical Derivations in Section 3

Verification of (7). Formula (7) can be verified as follows. Let $\langle\cdot\rangle_F$ be the Frobenius inner product of two matrices, then

$$\begin{aligned}&\langle \mathbf{D} - P_{T_{\mathbf{X}}\mathcal{M}}(\mathbf{D}), \mathbf{A}\mathbf{V}\mathbf{V}^T\rangle_F = \langle (\mathbf{I}-\mathbf{U}\mathbf{U}^T)\mathbf{D}(\mathbf{I}-\mathbf{V}\mathbf{V}^T), \mathbf{A}\mathbf{V}\mathbf{V}^T\rangle_F\\ =&\langle (\mathbf{I}-\mathbf{U}\mathbf{U}^T)\mathbf{D}(\mathbf{I}-\mathbf{V}\mathbf{V}^T)\mathbf{V}\mathbf{V}^T, \mathbf{A}\rangle_F = \langle \mathbf{0}, \mathbf{A}\rangle_F = 0\end{aligned}$$

and similarly $\langle \mathbf{D} - P_{T_{\mathbf{X}}\mathcal{M}}(\mathbf{D}), \mathbf{U}\mathbf{U}^T\mathbf{B}\rangle_F = 0$. As a result, $\langle \mathbf{D} - P_{T_{\mathbf{X}}\mathcal{M}}(\mathbf{D}), \mathbf{A}\mathbf{V}\mathbf{V}^T + \mathbf{U}\mathbf{U}^T\mathbf{B}\rangle_F = 0$ for all $\mathbf{A}\in\mathbb{R}^{n_1\times n_1}$ and $\mathbf{B}\in\mathbb{R}^{n_2\times n_2}$, which verifies formula (7) by showing that $\mathbf{D} - P_{T_{\mathbf{X}}\mathcal{M}}(\mathbf{D})$ is orthogonal to $T_{\mathbf{X}}\mathcal{M}$.

Verification of (10). It is clear that $R^{(2)}_{\mathbf{X}}(\delta)$ defined in (10) has rank r; and to show that $\langle R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta), \mathbf{Z}\rangle_F = 0$ for all $\mathbf{Z}\in T_{\mathbf{X}}\mathcal{M}$, we first write this property as $[R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta)] \perp T_{\mathbf{X}}\mathcal{M}$ for simplicity, and since $T_{\mathbf{X}}\mathcal{M} = \{\mathbf{A}\mathbf{V}\mathbf{V}^T + \mathbf{U}\mathbf{U}^T\mathbf{B} : \text{for } \mathbf{A}\in\mathbb{R}^{n_1\times n_1}, \mathbf{B}\in\mathbb{R}^{n_2\times n_2}\}$, we just need to show that $\langle R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta), \mathbf{A}\mathbf{V}\mathbf{V}^T + \mathbf{U}\mathbf{U}^T\mathbf{B}\rangle_F = 0$ for all $\mathbf{A}\in\mathbb{R}^{n_1\times n_1}$ and $\mathbf{B}\in\mathbb{R}^{n_2\times n_2}$. This is easy to verify, because we have $R^{(2)}_{\mathbf{X}}(\delta)\mathbf{V} = (\mathbf{X}+\delta)\mathbf{V}$,

$$\langle R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta), \mathbf{A}\mathbf{V}\mathbf{V}^T\rangle_F = \langle (R^{(2)}_{\mathbf{X}}(\delta)\mathbf{V} - (\mathbf{X}+\delta)\mathbf{V})\mathbf{V}^T, \mathbf{A}\rangle_F = \langle \mathbf{0}, \mathbf{A}\rangle_F = 0, \quad (21)$$

Similarly, we can easily verify that $\mathbf{U}^T R^{(2)}_{\mathbf{X}}(\delta) = \mathbf{U}^T(\mathbf{X}+\delta)$, we have $\langle R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta), \mathbf{U}\mathbf{U}^T\mathbf{B}\rangle_F = 0$, and therefore $[R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta)] \perp T_{\mathbf{X}}\mathcal{M}$. As a result, there exists a unique $R^{(2)}_{\mathbf{X}}$ such that $\text{rank}(R^{(2)}_{\mathbf{X}}) = r$ and $[R^{(2)}_{\mathbf{X}}(\delta) - (\mathbf{X}+\delta)] \perp T_{\mathbf{X}}\mathcal{M}$.

Verification of (12). We first define the operator $S:\mathbb{R}^{n_1\times n_2}\to\mathbb{R}^{n_1\times n_2}$ such that $F(\mathbf{A}) = \mathbf{S}(\mathbf{A})\circ\mathbf{A}$ ($\circ$ represents the elementwise product), i.e.,

$$\mathbf{S}(\mathbf{A}) = \begin{cases}0, & \text{if } |\mathbf{A}_{ij}| > |\mathbf{A}_{i,\cdot}|^{[\gamma]} \text{ and } |\mathbf{A}_{ij}| > |\mathbf{A}_{\cdot,j}|^{[\gamma]},\\ 1, & \text{otherwise.}\end{cases}$$

Then if the absolute values of all entries of $\mathbf{A}$ are different, the sparsity pattern does not change under a small perturbation, i.e., $\mathbf{S}(\mathbf{A}) = \mathbf{S}(\mathbf{A}+\mathbf{\Delta})$. Then by definition of $f(\cdot)$,

$$\begin{aligned}&f(\mathbf{L}+\mathbf{\Delta}) - f(\mathbf{L}) = \frac{1}{2}\|\mathbf{S}(\mathbf{L}-\mathbf{Y}+\mathbf{\Delta})\circ(\mathbf{L}-\mathbf{Y}+\mathbf{\Delta})\|_F^2 - \frac{1}{2}\|\mathbf{S}(\mathbf{L}-\mathbf{Y})\circ(\mathbf{L}-\mathbf{Y})\|_F^2\\ =&\frac{1}{2}\|\mathbf{S}(\mathbf{L}-\mathbf{Y})\circ(\mathbf{L}-\mathbf{Y}+\mathbf{\Delta})\|_F^2 - \frac{1}{2}\|\mathbf{S}(\mathbf{L}-\mathbf{Y})\circ(\mathbf{L}-\mathbf{Y})\|_F^2\\ =&\langle \mathbf{S}(\mathbf{L}-\mathbf{Y})\circ(\mathbf{L}-\mathbf{Y}), \mathbf{\Delta}\rangle_F + O(\|\mathbf{\Delta}\|_F^2),\end{aligned}$$

where $\circ$ represents the Hadamard product, i.e., the elementwise product between matrices.

Verification of (14). It is sufficient to prove the case where $\mathbf{U}^{(k)}$ and $\mathbf{V}^{(k)}$ are given by the SVD decomposition $\mathbf{L}^{(k)} = \mathbf{U}^{(k)}\mathbf{\Sigma}^{(k)}\mathbf{V}^{(k)T}$. Denote $\mathbf{D} = \nabla f(\mathbf{L}^{(k)}) = F(\mathbf{L}^{(k)} - \mathbf{Y})$. Set $\mathbf{X} = \mathbf{L}^{(k)}$ and $\delta = -\eta P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D})$ in (10), we have

$$\begin{aligned}\mathbf{L}^{(k+1)} \;:=\; &(\mathbf{L}^{(k)} - \eta P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D}))\mathbf{V}^{(k)}[\mathbf{U}^{(k)T}(\mathbf{L}^{(k)} - \eta P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D}))\mathbf{V}^{(k)}]^{-1} \quad (22)\\ &\mathbf{U}^{(k)T}(\mathbf{L}^{(k)} - \eta P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D}))\end{aligned}$$

On the other hand, from (7) we have the projection

$$P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D}) = \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D} + \mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T} - \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T}.$$

As a result

$$\begin{aligned}&P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D})\mathbf{V}^{(k)} = [\mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D} + \mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T} - \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T}]\mathbf{V}^{(k)}\\ &= \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}\mathbf{V}^{(k)} + \mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T}\mathbf{V}^{(k)} - \mathbf{U}^{(k)}\mathbf{U}^{(k)T}\mathbf{D}\mathbf{V}^{(k)}\mathbf{V}^{(k)T}\mathbf{V}^{(k)} = \mathbf{D}\mathbf{V}^{(k)} \quad (23)\end{aligned}$$

and similarly,

$$\mathbf{U}^{(k)T}P_{T_{\mathbf{L}^{(k)}}\mathcal{M}}(\mathbf{D}) = \mathbf{U}^{(k)T}\mathbf{D}. \quad (24)$$

Combining (23), (24) with (22), the update formula (14) is verified.

B. Proof of Theorem 1

In this proof, we will investigate $\|\mathbf{L}^+ - \mathbf{L}^*\|_F$, where

$$\mathbf{L}^+ = R_{\mathbf{L}}(-\eta P_{T_{\mathbf{L}}}F(\mathbf{L}-\mathbf{Y})).$$

It is sufficient to prove that when $\|\mathbf{L}-\mathbf{L}^*\| \le a\sigma_r(\mathbf{L}^*)$ with the value a satisfying the conditions in Theorem 1, then

$$\|\mathbf{L}^+ - \mathbf{L}^*\|_F \le \Big(1 - \frac{1-2C_1}{8}\eta\Big)\|\mathbf{L}-\mathbf{L}^*\|_F. \quad (25)$$

To prove (25), we first introduce three auxiliary lemmas.

Lemma 10 **(a)** *Let* $\mathbf{D} = \mathbf{L}-\mathbf{L}^* - F(\mathbf{L}-\mathbf{Y}) = \mathbf{L}-\mathbf{L}^* - F(\mathbf{L}-\mathbf{L}^*-\mathbf{S}^*)$, *then*

$$\|\mathbf{D}\|_F^2 \le C_1^2\|\mathbf{L}-\mathbf{L}^*\|_F^2. \quad (26)$$

(b) *For the noisy setting where* $\mathbf{Y} = \mathbf{L}^* + \mathbf{S}^* + \mathbf{N}^*$, *and* $\mathbf{D}' = \mathbf{L}-\mathbf{L}^*-\mathbf{N}^* - F(\mathbf{L}-\mathbf{Y})$, *we have*

$$\|\mathbf{D}'\|_F^2 \le 2C_1^2\|\mathbf{L}-\mathbf{L}^*\|_F^2 + 2(\gamma+5\gamma^*)N_1, \quad (27)$$

where $N_1 = n_2\sum_{i=1}^{n_1}|\mathbf{N}^*_{i,\cdot}|^{\max} + n_1\sum_{j=1}^{n_2}|\mathbf{N}^*_{\cdot,j}|^{\max}$.

Lemma 11 *If* $\|\mathbf{L}-\mathbf{L}^*\|_F \le a\sigma_r(\mathbf{L}^*)$ *and* $a\le 1$, *then*

$$\|(\mathbf{L}-\mathbf{L}^*) - \mathbf{P}_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F \le \frac{a}{2(1-a)}\|\mathbf{L}-\mathbf{L}^*\|_F, \quad (28)$$

$$\|(\mathbf{L}-\mathbf{L}^*) - \mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F \le \frac{a}{2}\|\mathbf{L}-\mathbf{L}^*\|_F. \quad (29)$$

Lemma 12 *For* $\mathbf{X}\in T_{\mathbf{L}}\mathcal{M}$, *then*

$$\|R^{(i)}_{\mathbf{L}}(\mathbf{X}) - (\mathbf{L}+\mathbf{X})\|_F \le \frac{\|\mathbf{X}\|_F^2}{2(\sigma_r(\mathbf{L}) - \|\mathbf{X}\|)}, \text{ for either } i=1 \text{ or } 2.$$

To prove (25), first we note that

$$\begin{aligned}
&\|\mathbf{L}-\mathbf{L}^*\|_F^2-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F^2\\
=&\|\mathbf{L}-\mathbf{L}^*\|_F^2-\|\mathbf{L}-\mathbf{L}^*\|_F^2+2\eta\langle\mathbf{L}-\mathbf{L}^*,P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\rangle_F-\|\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F^2\\
=&2\eta\langle\mathbf{L}-\mathbf{L}^*,P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\rangle_F-\|\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F^2\\
=&2\eta\langle\mathbf{L}-\mathbf{L}^*,P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)-P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*-F(\mathbf{L}-\mathbf{Y}))\rangle_F-\eta^2\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F^2\\
=&2\eta\langle P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*),P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)-P_{T_\mathbf{L}}\mathbf{D}\rangle_F-\eta^2\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F^2\\
\geq&2\eta(\|P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\|_F^2-\|\mathbf{D}\|_F\|P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\|_F)-\eta^2(\|\mathbf{L}-\mathbf{L}^*\|_F+\|\mathbf{D}\|_F)^2. \qquad (30)
\end{aligned}$$

The fourth line is obtained by $P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*-F(\mathbf{L}-\mathbf{Y}))=\mathbf{L}-\mathbf{L}^*-P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\rangle_F$. The fifth line is because $\mathbf{L}-\mathbf{L}^*=P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)+P_{T_\mathbf{L}}^\perp(\mathbf{L}-\mathbf{L}^*)$. The last line uses Cauchy-Schwarz inequality $\langle P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*),P_{T_\mathbf{L}}\mathbf{D}\rangle_F\leq\|\mathbf{D}\|_F\|P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\|_F$ and triangular inequality $\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F\leq\|\mathbf{L}-\mathbf{L}^*\|_F+\|P_{T_\mathbf{L}}(\mathbf{D})\|_F\leq\|\mathbf{L}-\mathbf{L}^*\|_F+\|\mathbf{D}\|_F$. Lemma 11 and the assumptions $\|\mathbf{L}-\mathbf{L}^*\|_F\leq a\sigma_r(\mathbf{L}^*)$ and $\sqrt{1-(\frac{a}{2(1-a)})^2}>\frac{1}{2}$ imply

$$\|P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\|_F\geq\frac{1}{2}\|\mathbf{L}-\mathbf{L}^*\|_F. \qquad (31)$$

Combining it with the estimation of $\|\mathbf{D}\|_F$ in Lemma 10, we have

$$\begin{aligned}
&\|\mathbf{L}-\mathbf{L}^*\|_F^2-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F^2\\
\geq&\eta(\frac{1}{2}-C_1)\|\mathbf{L}-\mathbf{L}^*\|_F^2-\eta^2(1+C_1)^2\|\mathbf{L}-\mathbf{L}^*\|_F^2. \qquad (32)
\end{aligned}$$

When ths RHS of (32) is positive (i.e., when $(1-2C_1)\geq2\eta(1+C_1)^2$), (32) implies $\|\mathbf{L}-\mathbf{L}^*\|_F>\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F$ and

$$\begin{aligned}
&\|\mathbf{L}-\mathbf{L}^*\|_F-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F\\
\geq&\frac{\eta(\frac{1}{2}-C_1)\|\mathbf{L}-\mathbf{L}^*\|_F^2-\eta^2(1+C_1)^2\|\mathbf{L}-\mathbf{L}^*\|_F^2}{\|\mathbf{L}-\mathbf{L}^*\|_F+\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F}\\
\geq&\frac{1}{2}\left(\eta(\frac{1}{2}-C_1)-\eta^2(1+C_1)^2\right)\|\mathbf{L}-\mathbf{L}^*\|_F. \qquad (33)
\end{aligned}$$

In addition,

$$\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F\leq\|F(\mathbf{L}-\mathbf{Y})\|_F=\|\mathbf{L}-\mathbf{L}^*\|_F+\|\mathbf{D}\|_F\leq(1+C_1)\|\mathbf{L}-\mathbf{L}^*\|_F \qquad (34)$$

and Lemma 12 give

$$\begin{aligned}
&\|\mathbf{L}^+-\mathbf{L}^*\|_F-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F\leq\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^+\|_F\\
\leq&\frac{\eta^2\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F^2}{\sigma_r(\mathbf{L}^*)-\eta\|P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\|_F}\leq\frac{\eta^2a^2(1+C_1)^2}{1-\eta a(1+C_1)}\|\mathbf{L}-\mathbf{L}^*\|_F. \qquad (35)
\end{aligned}$$

Combining (33) and (35),

$$\frac{\|\mathbf{L}-\mathbf{L}^*\|_F-\|\mathbf{L}^+-\mathbf{L}^*\|_F}{\|\mathbf{L}-\mathbf{L}^*\|_F}\geq\frac{1}{4}\eta(1-2C_1)-\eta^2(1+C_1)^2\left[\frac{1}{2}+\frac{a^2}{1-\eta(1+C_1)a}\right].$$

Therefore, Theorem 1 is proved when $C_1<1/2$, and η_0 is chosen such that

$$\eta_0(1+C_1)^2\left[\frac{1}{2}+\frac{a^2}{1-\eta_0(1+C_1)a}\right]\leq\frac{1}{8}(1-2C_1).$$

C. Proof of Theorem 5

The proof of the noisy case also follows similarly from the proofs of Theorem 1 and 6. Note that

$$F(\mathbf{L}-\mathbf{Y})=\mathbf{L}-\mathbf{L}^*-\mathbf{N}^*-\mathbf{D}',$$

and define $\mathbf{Q}=P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)$, then following the proof of Theorem 1 and applying Lemma 10 (b), we have

$$\begin{aligned}
&\|\mathbf{L}-\mathbf{L}^*\|_F^2-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F^2\\
=&2\eta\langle\mathbf{L}-\mathbf{L}^*,P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\rangle_F+O(\eta^2)=2\eta\langle P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*),P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})\rangle_F+O(\eta^2)\\
=&2\eta\langle P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*),P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*-\mathbf{N}^*-\mathbf{D}')\rangle_F+O(\eta^2)\\
\geq&2\eta\left(\|\mathbf{Q}\|_F^2-\langle\mathbf{N}^*,\mathbf{Q}\rangle_F-\|\mathbf{Q}\|_F\sqrt{2C_1^2\|\mathbf{L}-\mathbf{L}^*\|_F^2+2(\gamma+5\gamma^*)N_1}\right)+O(\eta^2).
\end{aligned}$$

In addition, (35) gives

$$\left|\|\mathbf{L}^+-\mathbf{L}\|_F-\|\mathbf{L}-\eta P_{T_\mathbf{L}}F(\mathbf{L}-\mathbf{Y})-\mathbf{L}^*\|_F\right|=O(\eta^2).$$

Combining it with the estimation of C_1, N_1, and $\langle\mathbf{N}^*,\mathbf{Q}\rangle_F$ in Lemma 13 and the fact that $(1-\frac{a}{2(1-a)})\|\mathbf{L}-\mathbf{L}^*\|_F\leq\|\mathbf{Q}\|_F\leq(1+\frac{a}{2(1-a)})\|\mathbf{L}-\mathbf{L}^*\|_F$ (which follows from Lemma 11), the Theorem is proved.

Lemma 13 *If $\mathbf{N}^*\in\mathbb{R}^{n_1\times n_2}$ is elementwisely i.i.d. sampled from $N(0,\sigma^2)$, then*
(a) with probability $1-\frac{4}{n_1^7n_2^7}$, $\sum_{i=1}^{n_1}(|\mathbf{N}^*_{i,\cdot}|^{\max})^2\leq16\sigma^2n_1\ln(n_1n_2)$, and $\sum_{j=1}^{n_2}(|\mathbf{N}^*_{\cdot,j}|^{\max})^2\leq16\sigma^2n_2\ln(n_1n_2)$, and as a result, $N_1\leq32\sigma^2n_1n_2\ln(n_1n_2)$.
(b) There exists $C_6>0$ such that as $n_1+n_2\to\infty$, the probability that

$$\langle\mathbf{N}^*,P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\rangle_F\leq\frac{1}{4}\|P_{T_\mathbf{L}}(\mathbf{L}-\mathbf{L}^*)\|_F^2 \qquad (36)$$

holds for all $\{\mathbf{L}:C_6\sigma\sqrt{(n_1+n_2)r\ln(n_1n_2)}\leq\|\mathbf{L}-\mathbf{L}^*\|_F\leq a\sigma_r(\mathbf{L}^*)\}$ converges to 1.

D. Proof of Theorem 6

This proof borrows two lemmas from (Yi et al., 2016, Lemmas 9, 10) as follows.

Lemma 14 *(Yi et al., 2016, Lemma 9) There exists $c>0$ such that for all $0<\epsilon<1$, if $p\geq c\mu r\log(n)/\epsilon^2\min(n_1,n_2)$, then with probability at least $1-2n^{-3}$, for all $\mathbf{X}$ in the tangent plane $T_{\mathbf{L}^*}$, i.e., all $\mathbf{X}$ that can be written as $\mathbf{L}^*\mathbf{A}+\mathbf{B}\mathbf{L}^*$, where $\mathbf{A}\in\mathbb{R}^{n_2\times n_2}$ and $\mathbf{B}\in\mathbb{R}^{n_1\times n_1}$,*

$$(1-\epsilon)\|\mathbf{X}\|_F^2\leq\frac{1}{p}\|P_\mathbf{\Phi}\mathbf{X}\|_F^2\leq(1+\epsilon)\|\mathbf{X}\|_F^2.$$

Lemma 15 *(Yi et al., 2016, Lemma 10) If $p\geq\frac{56}{3}\frac{\log n}{\gamma\min(n_1,n_2)}$, the with probability at least $1-6n^{-1}$, the number of entries in $\mathbf{\Phi}$ per row is in the interval $[pn_2/2,3pn_2/2]$, and the number of entries in $\mathbf{\Phi}$ per column is in $[pn_1/2,3pn_1/2]$.*

Then we introduce the following lemma parallel to Lemma 10:

Lemma 16 *When the events in Lemmas 14 and 15 hold, for* $\tilde{\mathbf{D}} = P_{\mathbf{\Phi}}[\mathbf{L} - \mathbf{L}^* - \tilde{F}(\mathbf{L} - \mathbf{Y})]$ *we have*

$$\|\tilde{\mathbf{D}}\|_F^2 \leq \tilde{C}_1^2 \|\mathbf{L} - \mathbf{L}^*\|_F^2, \tag{37}$$

with

$$\tilde{C}_1 = \frac{1}{p(1-\epsilon)}\Big[6(\gamma + 2\gamma^*)p\mu r + 4\frac{3\gamma^*}{\gamma - 3\gamma^*}(\sqrt{p(1+\epsilon)} + \frac{a}{2})^2 + a^2\Big].$$

The proof of Theorem 6 is parallel to the proof of Theorem 1, with $\mathbf{L}^+$ defined slightly differently by

$$\mathbf{L}^+ = R_{\mathbf{L}}(-\eta P_{T_{\mathbf{L}}}\tilde{F}(\mathbf{L} - \mathbf{Y})).$$

Defining $P_{\mathbf{\Phi}} : \mathbb{R}^{n_1 \times n_2} \to \mathbb{R}^{n_1 \times n_2}$ by

$$[P_{\mathbf{\Phi}}\mathbf{X}]_{ij} = \begin{cases} \mathbf{X}_{ij}, \text{ if } (i,j) \in \mathbf{\Phi}, \\ 0, \text{ if } (i,j) \notin \mathbf{\Phi}. \end{cases}$$

Then $\tilde{F}(\mathbf{L} - \mathbf{Y}) = P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})$. Following a similar analysis as (30),

$$\begin{aligned}
&\|\mathbf{L} - \mathbf{L}^*\|_F^2 - \|\mathbf{L} - \eta P_{T_{\mathbf{L}}}P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y}) - \mathbf{L}^*\|_F^2 \\
=&2\eta\langle \mathbf{L} - \mathbf{L}^*, P_{T_{\mathbf{L}}}P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})\rangle_F - \|\eta P_{T_{\mathbf{L}}}P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})\|_F^2 \\
\geq&2\eta\langle P_{\mathbf{\Phi}}P_{T_{\mathbf{L}}}(\mathbf{L} - \mathbf{L}^*), P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})\rangle_F - \|\eta P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})\|_F^2 \\
\geq&2\eta\langle P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*) - P_{\mathbf{\Phi}}P_{T_{\mathbf{L}}}^{\perp}(\mathbf{L} - \mathbf{L}^*), P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*) - \tilde{\mathbf{D}}\rangle_F - \eta^2(\|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F + \|\tilde{\mathbf{D}}\|_F)^2,
\end{aligned} \tag{38}$$

here $P_{T_{\mathbf{L}}}^{\perp}$ represents the projector to the subspace orthogonal to $T_{\mathbf{L}}$. Lemma 11 and Lemma 14 imply

$$\frac{\|P_{\mathbf{\Phi}}P_{T_{\mathbf{L}}}^{\perp}(\mathbf{L} - \mathbf{L}^*)\|_F}{\|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F} \leq \frac{\|P_{T_{\mathbf{L}}}^{\perp}(\mathbf{L} - \mathbf{L}^*)\|_F}{\|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F} \leq \frac{ap(1+\epsilon)}{2(1-a)}, \tag{39}$$

and combining it with the estimation of $\tilde{\mathbf{D}}$ in Lemma 16, the RHS of (38) is larger than

$$\|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F^2\left(2\eta\left(1 - \tilde{C}_1 - \frac{ap(1+\epsilon)}{2(1-a)}(1+\tilde{C}_1)\right) - \eta^2(1+\tilde{C}_1)^2\right). \tag{40}$$

In addition, Lemma 14 implies

$$\begin{aligned}
\|P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y})\|_F &\leq \|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F + \|P_{\mathbf{\Phi}}\tilde{\mathbf{D}}\|_F \\
&\leq (1 + \tilde{C}_1)\|P_{\mathbf{\Phi}}(\mathbf{L} - \mathbf{L}^*)\|_F, \\
&\leq (1 + \tilde{C}_1)p(1+\epsilon)\|\mathbf{L} - \mathbf{L}^*\|
\end{aligned}$$

and combining it with Lemma 12,

$$\|\mathbf{L}^+ - \mathbf{L}^*\|_F - \|\mathbf{L} - \eta P_{T_{\mathbf{L}}}P_{\mathbf{\Phi}}\tilde{F}(\mathbf{L} - \mathbf{Y}) - \mathbf{L}^*\|_F \leq \frac{\eta^2 a^2(p+p\epsilon)^2(1+\tilde{C}_1)^2}{1 - \eta a(p+p\epsilon)(1+\tilde{C}_1)}\|\mathbf{L} - \mathbf{L}^*\|_F.$$

Combining it with (40) and Lemma 11, we have

$$\begin{aligned}
\frac{\|\mathbf{L}^+ - \mathbf{L}^*\|_F}{\|\mathbf{L} - \mathbf{L}^*\|_F} \leq&\sqrt{1 - p^2(1-\epsilon)^2\left(2\eta\left(1 - \tilde{C}_1 - \frac{ap(1+\epsilon)}{2(1-a)}(1+\tilde{C}_1)\right) - \eta^2(1+\tilde{C}_1)^2\right)} \\
&+\frac{\eta^2 a^2(p+p\epsilon)^2(1+\tilde{C}_1)^2}{1 - \eta a(p+p\epsilon)(1+\tilde{C}_1)},
\end{aligned}$$

and Theorem 6 is proved.

E. Proof of Lemmas

Lemma 10(a) Proof By the definition of F, for any matrix $\mathbf{A}$, $\mathbf{A} - F(\mathbf{A})$ is a sparse matrix, therefore

$$\mathbf{D} = \mathbf{L} - \mathbf{L}^* - \mathbf{S}^* - F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*) + \mathbf{S}^*$$

is a sparse matrix. Denote the locations of the nonzero entries of $\mathbf{D}$ by $\mathcal{S}$, and divide it into two sets $\mathcal{S}_1 \cup \mathcal{S}_2$ defined as follows:

$$\mathcal{S}_1 = \{(i,j) : |[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{ij}| > |[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{i,\cdot}|^{[\gamma]} \text{ and } |[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{ij}| > |[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{\cdot,j}|^{[\gamma]}\},$$

and

$$\mathcal{S}_2 = \{(i,j) \notin \mathcal{S}_1 : \mathbf{D}_{ij} = [\mathbf{L} - \mathbf{L}^*]_{ij} - F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)_{ij} \neq 0\}.$$

For $(i,j) \in \mathcal{S}_1$, $[F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)]_{ij} = 0$. As a result, $\mathbf{D}_{ij} = [\mathbf{L} - \mathbf{L}^*]_{ij}$. In addition, by definition of $F(\cdot)$, each row or column of $\mathbf{D}$ has at most γ percentage of points in $\mathcal{S}_1$.

For $(i,j) \in \mathcal{S}_2$, since $[F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)]_{ij} = [\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{ij}$, we have $\mathbf{D}_{ij} = \mathbf{S}^*_{ij} \neq 0$. By Assumption 1, therefore, for each row or column of $\mathbf{D}$, at most γ^* percentage of points lie in $\mathcal{S}_2$.

Combine the results

$$|[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{i,\cdot}|^{[\gamma]} \leq \{|[\mathbf{L} - \mathbf{L}^*]_{i,\cdot}| + |[\mathbf{S}^*]_{i,\cdot}|\}^{[\gamma]} \leq |[\mathbf{L} - \mathbf{L}^*]_{i,\cdot}|^{[\gamma - \gamma^*]},$$

$$|[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{j,\cdot}|^{[\gamma]} \leq \{|[\mathbf{L} - \mathbf{L}^*]_{j,\cdot}| + |[\mathbf{S}^*]_{j,\cdot}|\}^{[\gamma]} \leq |[\mathbf{L} - \mathbf{L}^*]_{j,\cdot}|^{[\gamma - \gamma^*]},$$

with $[F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)]_{ij} = [\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{ij}$, we have for $(i,j) \in \mathcal{S}_2$

$$\begin{aligned}
|\mathbf{D}_{ij}| =&|[\mathbf{L} - \mathbf{L}^* - F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)]_{ij}| \\
\leq&|[\mathbf{L} - \mathbf{L}^*]_{ij}| + |F(\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*)]_{ij}| \\
\leq&|[\mathbf{L} - \mathbf{L}^*]_{ij}| + \max(|[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{i,\cdot}|^{[\gamma]}, |[\mathbf{L} - \mathbf{L}^* - \mathbf{S}^*]_{\cdot,j}|^{[\gamma]}) \\
\leq&|[\mathbf{L} - \mathbf{L}^*]_{ij}| + \max(|[\mathbf{L} - \mathbf{L}^*]_{i,\cdot}|^{[\gamma - \gamma^*]}, |[\mathbf{L} - \mathbf{L}^*]_{\cdot,j}|^{[\gamma - \gamma^*]}).
\end{aligned}$$

Applying the estimations above, and repeatedly use the fact that $(x+y)^2 \leq 2x^2 + 2y^2$, we have

$$
\begin{aligned}
\|\mathbf{D}\|_F^2 =& \sum_{(i,j)\in\mathcal{S}} \mathbf{D}_{ij}^2 = \sum_{(i,j)\in\mathcal{S}_1} \mathbf{D}_{ij}^2 + \sum_{(i,j)\in\mathcal{S}_2} \mathbf{D}_{ij}^2 \le \sum_{(i,j)\in\mathcal{S}_1} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 \\
&+ \sum_{(i,j)\in\mathcal{S}_2} \Big\{ |[\mathbf{L}-\mathbf{L}^*]_{ij}| + \max(|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]}, |[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]}) \Big\}^2 \\
\le& \sum_{(i,j)\in\mathcal{S}_1} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} \max\{|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]}, |[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]}\}^2 \\
\le& \sum_{(i,j)\in\mathcal{S}_1} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} \{|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]} + |[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]}\}^2 \\
\le& \sum_{(i,j)\in\mathcal{S}_1} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 4\sum_{(i,j)\in\mathcal{S}_2} \{|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]}\}^2 + \{|[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]}\}^2 \\
\le& \sum_{(i,j)\in\mathcal{S}} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + \sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + 4\frac{\gamma^*}{\gamma-\gamma^*}\|\mathbf{L}-\mathbf{L}^*\|_F^2 \\
\le& 2\sum_{(i,j)\in\mathcal{S}} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 + 4\frac{\gamma^*}{\gamma-\gamma^*}\|\mathbf{L}-\mathbf{L}^*\|_F^2 \\
&+ 2\sum_{(i,j)\in\mathcal{S}} [\mathbf{L}-\mathbf{L}^*-\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*-\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 \\
\le& 2\sum_{(i,j)\in\mathcal{S}} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 + 2\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 + 4\frac{\gamma^*}{\gamma-\gamma^*}\|\mathbf{L}-\mathbf{L}^*\|_F^2 \\
&+ 4\|\mathbf{L}-\mathbf{L}^*-\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F^2. \qquad (41)
\end{aligned}
$$

Note that from line 5 to line 6, we used the fact that for $\mathbf{x}\in\mathbb{R}^n$, and $k\le n$

$$k(x_{(k)})^2 \le (x_{(k)})^2 + (x_{(k+1)})^2 + \cdots + (x_{(n-1)}) + (x_{(n)}) \le (x_{(1)}) + \cdots + (x_{(n)}) = \|x\|_F^2,$$

where $x_{(k)}$ is the k-th order statistics of $x_1,\ldots,x_n$, i.e. the k-th smallest value. This gives us

$$(\gamma-\gamma^*)n_2|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]} \le \|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}\|_2^2; \quad (\gamma-\gamma^*)n_2|[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]} \le \|[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}\|_2^2.$$

Therefore

$$\sum_{(i,j)\in\mathcal{S}_2} |[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]} \le \frac{\gamma^* n_2}{(\gamma-\gamma^*)n_2}\|[\mathbf{L}-\mathbf{L}^*]_{i,\cdot}\|_F^2; \qquad (42)$$

$$\sum_{(i,j)\in\mathcal{S}_2} |[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]} \le \frac{\gamma^* n_1}{(\gamma-\gamma^*)n_1}\|[\mathbf{L}-\mathbf{L}^*]_{\cdot,j}\|_F^2. \qquad (43)$$

The values $\gamma^* n_2$ and $\gamma^* n_1$ in the numerator of the right hand sides in 42 and 43 are due to the fact that, in each row or column of $\mathbf{D}$, at most γ^* percentage of points lie in $\mathcal{S}_2$.

On the other hand, Lemma 11 implies

$$\|\mathbf{L}-\mathbf{L}^*-\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F \le \frac{a}{2}\|\mathbf{L}-\mathbf{L}^*\|_F. \qquad (44)$$

In addition, using the fact that there exists $\mathbf{A}\in\mathbb{R}^{n_1\times r}$ and $\mathbf{B}\in\mathbb{R}^{n_2\times r}$, such that $\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*) = \mathbf{A}\mathbf{V}^T + \mathbf{U}\mathbf{B}^T$ and $\|\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F^2 = \|\mathbf{A}\mathbf{V}^T\|_F^2 + \|\mathbf{U}\mathbf{B}^T\|_F^2$, and that for each row or column, at most $\gamma+\gamma^*$ percentage of points lie in $\mathcal{S}$, we have

$$
\begin{aligned}
&\sum_{(i,j)\in\mathcal{S}} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 \le 2\sum_{(i,j)\in\mathcal{S}} [\|(\mathbf{A}\mathbf{V}^T)_{ij}\|^2 + \|(\mathbf{U}\mathbf{B}^T)_{ij}\|^2] \\
\le& 2(\gamma+\gamma^*)\mu r \sum_{1\le i\le n_1, 1\le j\le n_2} [\|(\mathbf{A}\mathbf{V}^T)_{ij}\|^2 + \|(\mathbf{U}\mathbf{B}^T)_{ij}\|^2] = 2(\gamma+\gamma^*)\mu r\|\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F^2 \\
\le& 2(\gamma+\gamma^*)\mu r\|\mathbf{L}-\mathbf{L}^*\|_F^2. \qquad (45)
\end{aligned}
$$

Similarly, $\|\mathbf{A}\|_{2,\infty} = \max_{\|\mathbf{z}\|_2=1}\|\mathbf{A}\mathbf{z}\|_\infty$

$$\sum_{(i,j)\in\mathcal{S}_2} [\mathbf{P}_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)]_{ij}^2 \le 2\gamma^*\mu r\|\mathbf{L}-\mathbf{L}^*\|_F^2, \qquad (46)$$

Combining (41)-(46), (26) is proved. ■

Lemma 10(b) Proof Let $\mathbf{L}' = \mathbf{L}-\mathbf{N}^*$, then applying the fact that for any $\mathbf{x},\mathbf{y}\in\mathbb{R}^n$,

$$|[\mathbf{x}+\mathbf{y}]|^{[\gamma]} \le |[\mathbf{x}]|^{[\gamma]} + |[\mathbf{x}]|^{\max},$$

where $|[\mathbf{x}]|^{\max}$ represents the largest value of $|[\mathbf{x}]|$. We have

$$
\begin{aligned}
\|\mathbf{D}'\|_F^2 \le& \sum_{(i,j)\in\mathcal{S}} [\mathbf{L}'-\mathbf{L}^*]_{ij}^2 + \sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}'-\mathbf{L}^*]_{ij}^2 \\
&+ 2\sum_{(i,j)\in\mathcal{S}_2} \{(|[\mathbf{L}'-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]})^2 + (|[\mathbf{L}'-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]})^2\} \\
\le& 2\left(\sum_{(i,j)\in\mathcal{S}} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + |\mathbf{N}_{ij}^*|^2 + \sum_{(i,j)\in\mathcal{S}_2} [\mathbf{L}-\mathbf{L}^*]_{ij}^2 + |\mathbf{N}_{ij}^*|^2\right) \\
&+ 4\sum_{(i,j)\in\mathcal{S}_2} \{(|[\mathbf{L}'-\mathbf{L}^*]_{i,\cdot}|^{[\gamma-\gamma^*]})^2 + (|\mathbf{N}_{i,\cdot}^*|^{\max})^2 + (|[\mathbf{L}'-\mathbf{L}^*]_{\cdot,j}|^{[\gamma-\gamma^*]})^2 + (|\mathbf{N}_{\cdot,j}^*|^{\max})^2\} \\
\le& 2C_1^2\|\mathbf{L}-\mathbf{L}^*\|_F^2 + 2(\gamma+5\gamma^*)N_1,
\end{aligned}
$$

where the last inequality follows from the proof of part (a) and the definition of N_1. ■

Lemma 11 Proof Let the SVD decomposition of $\mathbf{L}^*$ be $\mathbf{L}^* = \mathbf{U}\mathbf{\Sigma}\mathbf{V}$, $\mathbf{U}^\perp$ and $\mathbf{V}^\perp$ be orthogonal matrices of sizes $\mathbb{R}^{n_1\times(n_1-r)}$ and $\mathbb{R}^{n_2\times(n_2-r)}$ such that $\mathrm{Col}(\mathbf{U}^\perp)\perp\mathrm{Col}(\mathbf{U})$ and $\mathrm{Col}(\mathbf{V}^\perp)\perp\mathrm{Col}(\mathbf{V})$ (here $\mathrm{Col}(\mathbf{U})$ represents the subspace spanned by the columns of $\mathbf{U}$). Let

$$\mathbf{L}_{(1,1)}^* \equiv \mathbf{U}^T\mathbf{L}^*\mathbf{V}, \qquad \mathbf{L}_{(1,2)}^* \equiv \mathbf{U}^T\mathbf{L}^*\mathbf{V}^\perp,$$

$$\mathbf{L}_{(2,1)}^* \equiv \mathbf{U}^{\perp T}\mathbf{L}^*\mathbf{V}, \qquad \mathbf{L}_{(2,2)}^* \equiv \mathbf{U}^{\perp T}\mathbf{L}^*\mathbf{V}^\perp.$$

Since $\text{rank}(\mathbf{L}^*) = r$, we have

$$\mathbf{L}^*_{(2,2)} = \mathbf{L}^*_{(2,1)}{\mathbf{L}^*_{(1,1)}}^{-1}\mathbf{L}^*_{(1,2)}.$$

Since all singular values of $\mathbf{L}^*_{(1,1)}$ are larger than $(1-a)\sigma_r(\mathbf{L}^*)$, if the singular value decomposition of ${\mathbf{L}^*_{(1,1)}}^{-1}$ is given by

$${\mathbf{L}^*_{(1,1)}}^{-1} = \mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^T,$$

then the $\|\mathbf{\Sigma}_0\| \le 1/(1-a)\sigma_r(\mathbf{L}^*)$. Applying

$$\|\mathbf{AB}\|_F^2 \le \|\mathbf{A}\|_F^2\|\mathbf{B}\|_F^2$$

and the fact that for a square, diagonal matrix Σ, $|[\mathbf{X\Sigma}]_{ij}| = |\mathbf{X}_{ij}\mathbf{\Sigma}_{jj}| \le \|\mathbf{\Sigma}\||\mathbf{X}_{ij}|$, we have

$$\begin{aligned}
\|\mathbf{L}^*_{(2,2)}\|_F &= \|\mathbf{L}^*_{(2,1)}\mathbf{U}_0\mathbf{\Sigma}_0\mathbf{V}_0^T\mathbf{L}^*_{(1,2)}\|_F \\
&\le \|\mathbf{L}^*_{(2,1)}\mathbf{U}_0\mathbf{\Sigma}_0\|_F\|\mathbf{V}_0^T\mathbf{L}^*_{(1,2)}\|_F \\
&\le \frac{1}{(1-a)\sigma_r(\mathbf{L}^*)}\|\mathbf{L}^*_{(2,1)}\mathbf{U}_0\|_F\|\mathbf{V}_0^T\mathbf{L}^*_{(1,2)}\|_F \\
&\le \frac{1}{(1-a)\sigma_r(\mathbf{L}^*)}\|\mathbf{L}^*_{(2,1)}\|_F\|\mathbf{L}^*_{(1,2)}\|_F \\
&\le \frac{1}{(1-a)\sigma_r(\mathbf{L}^*)}\Big(\frac{\|\mathbf{L}^*_{(2,1)}\|_F^2 + \|\mathbf{L}^*_{(1,2)}\|_F^2}{2}\Big) \\
&\le \frac{1}{(1-a)\sigma_r(\mathbf{L}^*)}\Big(\frac{a^2\sigma_r(\mathbf{L}^*)^2}{2}\Big) \\
&\le \frac{a^2}{2(1-a)}\sigma_r(\mathbf{L}^*), \qquad (47)
\end{aligned}$$

and (28) is proved. The proof of (29) is similar. ■

Lemma 12 **Proof** Let the SVD decomposition of $\mathbf{L}$ be $\mathbf{L} = \mathbf{U\Sigma V}$, and

$$\begin{aligned}
\mathbf{L}_{(1,1)} &= \mathbf{U}^T(\mathbf{X}+\mathbf{L})\mathbf{V}, \mathbf{L}_{(1,2)} = \mathbf{U}^T(\mathbf{X}+\mathbf{L})\mathbf{V}^\perp \\
&= \mathbf{U}^T\mathbf{X}\mathbf{V}^\perp, \mathbf{L}_{(2,1)} = \mathbf{U}^{\perp T}(\mathbf{X}+\mathbf{L})\mathbf{V} = \mathbf{U}^{\perp T}\mathbf{X}\mathbf{V},
\end{aligned}$$

then it is clear that

$$R^{(2)}_{\mathbf{L}}(\mathbf{X}) = \mathbf{L} + \mathbf{X} + \mathbf{U}^\perp\mathbf{L}_{(2,1)}{\mathbf{L}_{(1,1)}}^{-1}\mathbf{L}_{(1,2)}\mathbf{V}^{\perp T},$$

and using the same argument as in (47),

$$\begin{aligned}
\|\mathbf{L}_{(2,1)}{\mathbf{L}_{(1,1)}}^{-1}\mathbf{L}_{(1,2)}\|_F \le& \frac{1}{\sigma_r(\mathbf{L}^*_{(1,1)})}\|\mathbf{L}_{(1,2)}\|_F\|\mathbf{L}_{(2,1)}\|_F \\
\le& \frac{1}{\sigma_r(\mathbf{L}) - \|\mathbf{X}\|}\Big(\frac{\|\mathbf{L}_{(2,1)}\|_F^2 + \|\mathbf{L}_{(1,2)}\|_F^2}{2}\Big) \\
\le& \frac{1}{\sigma_r(\mathbf{L}) - \|\mathbf{X}\|}\frac{\|\mathbf{X}\|_F^2}{2}.
\end{aligned}$$

So Lemma 12 is proved for $R^{(2)}_{\mathbf{L}}(\mathbf{X})$.

By definition, $R^{(1)}_{\mathbf{L}}(\mathbf{X})$ is the closest matrix to $\mathbf{L}+\mathbf{X}$ that has rank r, so $\|R^{(1)}_{\mathbf{L}}(\mathbf{X}) - (\mathbf{L}+\mathbf{X})\|_F \le R^{(2)}_{\mathbf{L}}(\mathbf{X}) - (\mathbf{L}+\mathbf{X})\|_F$ and Lemma 12 is also proved for $R^{(1)}_{\mathbf{L}}(\mathbf{X})$. ■

Lemma 13 **Proof** WLOG, we assume $\sigma = 1$ and the generic cases can be proved similarly.

(a) It follows from the estimation of the distribution of the maximum of n_1 i.i.d. Gaussian variables $\{g_i\}_{i=1}^{n_1}$:

$$\begin{aligned}
&\Pr\{\max_{1\le i\le n_1}|g_i| \le 4\sqrt{\ln(n_1n_2)}\} \ge \Big(1 - 2\exp\Big(-\frac{(4\sqrt{\ln(n_1n_2)})^2}{2}\Big)\Big)^{n_1} \\
\ge& 1 - 2n_1\exp\Big(-\frac{(4\sqrt{\ln(n_1n_2)})^2}{2}\Big) = 1 - 2n_1^{-7}n_2^{-8},
\end{aligned}$$

where the first inequality applies the estimation of the cumulative distribution function of the Gaussian distribution (Ledoux and Talagrand, 1991, pg 8).

Combining this estimation for each column of $\mathbf{N}^*$ and applying the union bound, the second inequality in part (a) holds with probability $1-2n_1^{-7}n_2^{-7}$. Similarly, the first inequality in part (a) holds with the same probability.

(b) First, we parameterize $\mathbf{L}$ by $g(\mathbf{L}) = P_{\mathbf{L}^*}(\mathbf{L}-\mathbf{L}^*)$. Then we claim that, for any $\mathbf{L}$ and $\mathbf{L}'$ such that $\|\mathbf{L}-\mathbf{L}^*\|_F, \|\mathbf{L}'-\mathbf{L}^*\|_F \le a\sigma_r(\mathbf{L}^*)$, there exists C_0 depending on a such that

$$\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*) - P_{T_{\mathbf{L}'}}(\mathbf{L}'-\mathbf{L}^*)\|_F \le C_0\|g(\mathbf{L}) - g(\mathbf{L}')\|_F. \qquad (48)$$

To prove (48), apply (29) and obtain

$$\|\mathbf{L}-\mathbf{L}'\|_F \le \frac{1}{1-\frac{a}{2}}\|g(\mathbf{L}) - g(\mathbf{L}')\|_F. \qquad (49)$$

Since $P_{T_{\mathbf{L}}} = \mathbf{U_L}\mathbf{U_L}^T + \mathbf{V_L}\mathbf{V_L}^T - \mathbf{U_L}\mathbf{U_L}^T\mathbf{V_L}\mathbf{V_L}^T$, and using Davis-Kahan theorem Davis and Kahan (1970) and the assumption $\|\mathbf{L}-\mathbf{L}^*\|_F \le a\sigma_r(\mathbf{L}^*)$, there exists c_1, c_2 depending on a such that

$$\|\mathbf{U_L}\mathbf{U_L}^T - \mathbf{U_{L'}}\mathbf{U_{L'}}^T\|_F \le c_1,\ \|\mathbf{V_L}\mathbf{V_L}^T - \mathbf{V_{L'}}\mathbf{V_{L'}}^T\|_F \le c_2,$$

so there exists C' depending on a such that

$$\begin{aligned}
&\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*) - P_{T_{\mathbf{L}'}}(\mathbf{L}'-\mathbf{L}^*)\|_F \qquad (50)\\
=&\|[P_{T_{\mathbf{L}'}}(\mathbf{L}-\mathbf{L}^*) - P_{T_{\mathbf{L}'}}(\mathbf{L}'-\mathbf{L}^*)] + [P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*) - P_{T_{\mathbf{L}'}}(\mathbf{L}-\mathbf{L}^*)]\|_F \\
\le&\|\mathbf{L}-\mathbf{L}'\|_F + C'\|\mathbf{L}-\mathbf{L}'\|_F.
\end{aligned}$$

Combining (49) and (50), (48) is proved.

Second, based on (48), we will apply an ϵ-net covering argument to finish the proof that combines probabilistic estimation for each $\mathbf{L}$ and a union bound (ϵ-net covering argument is a standard argument in probabilistic estimation Vershynin (2012)). Use the estimation of

the cumulative distribution function of the Gaussian distribution (Ledoux and Talagrand, 1991, pg 8), for any $\mathbf{L}'$,

$$\Pr\left\{\langle \mathbf{N}^*, P_{T_{\mathbf{L}'}}(\mathbf{L}'-\mathbf{L}^*)\rangle_F \geq t\|P_{T_{\mathbf{L}'}}(\mathbf{L}'-\mathbf{L}^*)\|_F\right\} \leq \frac{1}{2}\exp\left(-\frac{t^2}{2}\right).$$

For any $\mathbf{L}$ such that $\|g(\mathbf{L}')-g(\mathbf{L})\|_F<\epsilon$, applying (48),

$$\Pr\left\{\langle \mathbf{N}^*, P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\rangle_F \geq t\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F + C_0\epsilon(\|\mathbf{N}^*\|_F+t)\right\} \leq \frac{1}{2}\exp\left(-\frac{t^2}{2}\right).$$

Using union bound, there is an ϵ-net of the set $\{g(\mathbf{L}) : \|g(\mathbf{L})\|_F = x\}$ with at most $(C_5x/\epsilon)^{n_1r+n_2r-r^2}$ points. Therefore, for all $\mathbf{L}$ such that $x-\epsilon \leq \|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F \leq x+\epsilon$,

$$\begin{aligned}&\Pr\left\{\langle \mathbf{N}^*, P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\rangle_F \geq t\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F + 2C_0\epsilon(\|\mathbf{N}^*\|_F+t)\right\}\\ \leq&\frac{1}{2}\exp\left(-\frac{t^2}{2}\right)\cdot\left(\frac{C_5x}{\epsilon}\right)^{n_1r+n_2r-r^2}. \end{aligned} \tag{51}$$

Let $t=x/8$ and $\epsilon = x/16C_0\|\mathbf{N}^*\|_F$, then when $\|\mathbf{N}^*\|_F \geq 1$ (which holds with high probability as n_1n_2 goes to infinity), then using $C_0\geq 1$ we have $\epsilon \leq x/16$, and when $x\geq 4$,

$$\begin{aligned}&t\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F + 2C_0\epsilon(\|\mathbf{N}^*\|_F+t) \leq \frac{x}{8}(x+\epsilon)+\frac{x}{8\|\mathbf{N}^*\|_F}(\|\mathbf{N}^*\|_F+t)\\ =&\frac{x}{8}(x+\epsilon)+\frac{x}{8}+\frac{x^2}{64\|\mathbf{N}^*\|_F} \leq \frac{x^2}{8}\frac{17}{16}+\frac{x}{8}+\frac{x^2}{64} \leq \frac{x^2}{8}\frac{17}{16}+\frac{x^2}{32}+\frac{x^2}{64} \leq \frac{1}{4}(x-\epsilon)^2\\ \leq&\frac{1}{4}\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F^2, \end{aligned} \tag{52}$$

where the last inequality applies the assumption $x-\epsilon \leq \|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F$. Combining (51) and (52) and recall that $t=x/8$, we have that for all $\mathbf{L}$ such that $x-x/16C_0\|\mathbf{N}^*\|_F \leq \|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F \leq x+x/16C_0\|\mathbf{N}^*\|_F$,

$$\begin{aligned}&\Pr\Bigg\{\langle \mathbf{N}^*, P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\rangle_F \geq \frac{1}{4}\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F^2,\\ &\qquad\text{for all } \mathbf{L} \text{ s.t. } \left|\|P_{T_{\mathbf{L}}}(\mathbf{L}-\mathbf{L}^*)\|_F - x\right| \leq \frac{x}{16C_0\|\mathbf{N}^*\|_F}\Bigg\}\\ \leq&\frac{1}{2}\exp\left(-\frac{x^2}{128}\right)\cdot\left(16C_5C_0\|\mathbf{N}^*\|_F\right)^{n_1r+n_2r-r^2}. \end{aligned} \tag{53}$$

Let $x_i = \sqrt{n_1+n_2+128(n_1r+n_2r-r^2)\ln(16C_5C_0\|\mathbf{N}^*\|_F)}(1+1/16C_0\|\mathbf{N}^*\|_F)^i$ with $i=1,2,...$, then

$$\begin{aligned}&\sum_{i=1}^{\infty}\exp\left(-\frac{x_i^2}{128}\right)\cdot\left(16C_5C_0\|\mathbf{N}^*\|_F\right)^{n_1r+n_2r-r^2}\\ \leq&\exp(-\frac{n_1+n_2}{128})\sum_{i=1}^{\infty}\exp(-(1+1/16C_0\|\mathbf{N}^*\|_F)^{2i})\\ \leq&\exp(-\frac{n_1+n_2}{128})\sum_{i=1}^{\infty}\exp(-1-i/8C_0\|\mathbf{N}^*\|_F)\\ =&\exp(-\frac{n_1+n_2}{128}-1)\frac{\exp(-1/8C_0\|\mathbf{N}^*\|_F)}{1-\exp(-1/8C_0\|\mathbf{N}^*\|_F)} \leq 8C_0\|\mathbf{N}^*\|_F\exp(-\frac{n_1+n_2}{128}-1), \end{aligned} \tag{54}$$

where the last inequality uses $\exp(-c)\leq 1-c$ when $c\geq 0$. Clearly, the RHS goes to 0 as $n_1+n_2\to\infty$.

Combining the estimation (53) for $\{x_i\}_{i=1}^{\infty}$, with probability $1-8C_0\|\mathbf{N}^*\|_F\exp(-\frac{n_1+n_2}{128}-1)$, the event (36) holds for all $\mathbf{L}$ such that

$$\|g(\mathbf{L})\|_F \geq \max(\sqrt{n_1+n_2+128(n_1r+n_2r-r^2)\ln(16C_5C_0\|\mathbf{N}^*\|_F)}, 4).$$

Combining it with (29), the event (36) holds for all for all $\mathbf{L}$ such that

$$\begin{aligned}a\sigma_r(\mathbf{L}^*) \geq& \|\mathbf{L}-\mathbf{L}^*\|_F\\ \geq& \frac{1}{1-\frac{a}{2}}\max(\sqrt{n_1+n_2+128(n_1r+n_2r-r^2)\ln(16C_5C_0\|\mathbf{N}^*\|_F)}, 4).\end{aligned}$$

Considering that $\sqrt{n_1+n_2+128(n_1r+n_2r-r^2)\ln(16C_5C_0\|\mathbf{N}^*\|_F)}$ is the dominant term when $n_1,n_2\to\infty$, Lemma 13(b) is proved. ■

Lemma 16 Proof Following (41) and the proof of Lemma 10[a], and note that Lemma 15 means that γ^* and γ are replaced by arbitrary numbers in the intervals $[0.5p\gamma^*, 1.5p\gamma^*]$ and $[0.5p\gamma, 1.5p\gamma]$, we have

$$\|\tilde{\mathbf{D}}\|_F^2 \leq 6(\gamma+2\gamma^*)p\mu r\|\mathbf{L}-\mathbf{L}^*\|_F^2 + 4\frac{3\gamma^*}{\gamma-3\gamma^*}\|P_{\mathbf{\Phi}}(\mathbf{L}-\mathbf{L}^*)\|_F^2 + a^2\|\mathbf{L}-\mathbf{L}^*\|_F^2.$$

Applying Lemma 11 and (44), we have

$$\begin{aligned}\|P_{\mathbf{\Phi}}(\mathbf{L}-\mathbf{L}^*)\|_F \leq& \|P_{\mathbf{\Phi}}P_{T_{\mathbf{L}^*}}(\mathbf{L}-\mathbf{L}^*)\|_F + \|P_{\mathbf{\Phi}}P_{T_{\mathbf{L}^*}}^{\perp}(\mathbf{L}-\mathbf{L}^*)\|_F\\ \leq& \sqrt{p(1+\epsilon)}\|\mathbf{L}-\mathbf{L}^*\|_F + \frac{a}{2}\|\mathbf{L}-\mathbf{L}^*\|_F.\end{aligned}$$

Combining it with the estimation of $\|P_{\mathbf{\Phi}}(\mathbf{L}-\mathbf{L}^*)\|_F$ in Lemma 11, we have $\|\tilde{\mathbf{D}}\|_F \leq \tilde{C}_1\|P_{\mathbf{\Phi}}(\mathbf{L}-\mathbf{L}^*)\|_F$ with

$$\tilde{C}_1 = \frac{1}{p(1-\epsilon)}\Big[6(\gamma+2\gamma^*)p\mu r + 4\frac{3\gamma^*}{\gamma-3\gamma^*}(\sqrt{p(1+\epsilon)}+\frac{a}{2})^2 + a^2\Big].$$

■

References

P.-A. Absil and I V Oseledets. Low-rank retractions: a survey and new results. *Computational Optimization and Applications*, 62(1):5–29, 2015. ISSN 1573-2894. doi: 10.1007/s10589-014-9714-4. URL http://dx.doi.org/10.1007/s10589-014-9714-4.

P.A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, 2009. ISBN 9781400830244. URL http://books.google.com/books?id=NSQGQeLN3NcC.

Afonso S. Bandeira, Nicolas Boumal, and Vladislav Voroninski. On the low-rank approach for semidefinite programs arising in synchronization and community detection. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 361–382, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR. URL http://proceedings.mlr.press/v49/bandeira16.html.

R. Basri and D. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003.

Srinadh Bhojanapalli, Prateek Jain, and Sujay Sanghavi. Tighter low-rank approximation via sampling the leveraged element. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 902–920, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=2722129.2722191.

Nicolas Boumal. Nonconvex phase synchronization. *SIAM Journal on Optimization*, 26 (4):2355–2377, 2016. doi: 10.1137/16M105808X. URL http://dx.doi.org/10.1137/16M105808X.

Samuel Burer and Renato D.C. Monteiro. A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization. *Mathematical Programming*, 95(2):329–357, 2003. ISSN 1436-4646. doi: 10.1007/s10107-002-0352-8. URL http://dx.doi.org/10.1007/s10107-002-0352-8.

Samuel Burer and Renato D.C. Monteiro. Local minima and convergence in low-rank semidefinite programming. *Mathematical Programming*, 103(3):427–444, 2005. ISSN 1436-4646. doi: 10.1007/s10107-004-0564-1. URL http://dx.doi.org/10.1007/s10107-004-0564-1.

Léopold Cambier and P.-A. Absil. Robust low-rank matrix completion by riemannian optimization. *SIAM Journal on Scientific Computing*, 38(5):S440–S460, 2016. doi: 10.1137/15M1025153. URL https://doi.org/10.1137/15M1025153.

Emmanuel J. Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *J. ACM*, 58(3):11:1–11:37, June 2011. ISSN 0004-5411. doi: 10.1145/1970392.1970395. URL http://doi.acm.org/10.1145/1970392.1970395.

Venkat Chandrasekaran, Sujay Sanghavi, Pablo A. Parrilo, and Alan S. Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2): 572–596, 2011. doi: 10.1137/090761793. URL http://dx.doi.org/10.1137/090761793.

Yudong Chen and Martin J. Wainwright. Fast low-rank estimation by projected gradient descent: General statistical and algorithmic guarantees. *CoRR*, abs/1509.03025, 2015.

Yeshwanth Cherapanamjeri, Kartik Gupta, and Prateek Jain. Nearly-optimal robust matrix completion. *CoRR*, abs/1606.07315, 2016. URL http://arxiv.org/abs/1606.07315.

Kenneth L. Clarkson and David P. Woodruff. Low rank approximation and regression in input sparsity time. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 81–90, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2029-0. doi: 10.1145/2488608.2488620. URL http://doi.acm.org/10.1145/2488608.2488620.

Chandler Davis and W. M. Kahan. The rotation of eigenvectors by a perturbation. iii. *SIAM Journal on Numerical Analysis*, 7(1):1–46, 1970. doi: 10.1137/0707001. URL http://dx.doi.org/10.1137/0707001.

Christopher De Sa, Kunle Olukotun, and Christopher Ré. Global convergence of stochastic gradient descent for some non-convex matrix problems. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML'15, pages 2332–2341. JMLR.org, 2015. URL http://dl.acm.org/citation.cfm?id=3045118.3045366.

Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1990. ISSN 1097-4571. doi: 10.1002/(SICI)1097-4571(199009)41:6⟨391::AID-ASI1⟩3.0.CO;2-9. URL http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9.

R. Epstein, P. Hallinan, and A. Yuille. 5 ± 2 eigenimages suffice: An empirical investigation of low-dimensional lighting models. In *IEEE Workshop on Physics-based Modeling in Computer Vision*, pages 108–116, June 1995.

Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *J. ACM*, 51(6):1025–1041, November 2004. ISSN 0004-5411. doi: 10.1145/1039488.1039494. URL http://doi.acm.org/10.1145/1039488.1039494.

Quanquan Gu, Zhaoran Wang, and Han Liu. Low-rank and sparse structure pursuit via alternating minimization. In Arthur Gretton and Christian C. Robert, editors, *AISTATS*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 600–609. JMLR.org, 2016. URL http://dblp.uni-trier.de/db/conf/aistats/aistats2016.html#GuWL16.

J. Ho, M. Yang, J. Lim, K. Lee, and D. Kriegman. Clustering appearances of objects under varying illumination conditions. In *Proceedings of International Conference on Computer Vision and Pattern Recognition*, volume 1, pages 11–18, 2003.

D. Hsu, S. M. Kakade, and T. Zhang. Robust matrix decomposition with sparse corruptions. *IEEE Transactions on Information Theory*, 57(11):7221–7234, Nov 2011. ISSN 0018-9448. doi: 10.1109/TIT.2011.2158250.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the Forty-fifth Annual ACM Symposium on Theory of Computing*, STOC '13, pages 665–674, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2029-0. doi: 10.1145/2488608.2488693. URL http://doi.acm.org/10.1145/2488608.2488693.

R. H. Keshavan, A. Montanari, and S. Oh. Matrix completion from a few entries. *IEEE Transactions on Information Theory*, 56(6):2980–2998, June 2010. ISSN 0018-9448. doi: 10.1109/TIT.2010.2046205.

A. Kyrillidis and V. Cevher. Matrix alps: Accelerated low rank and sparse matrix reconstruction. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 185–188, Aug 2012. doi: 10.1109/SSP.2012.6319655.

M. Ledoux and M. Talagrand. *Probability in Banach Spaces: Isoperimetry and Processes*. A Series of Modern Surveys in Mathematics Series. Springer, 1991. ISBN 9783540520139. URL https://books.google.com/books?id=cyKYDfvxRjsC.

L. Li, W. Huang, I. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Image Processing, IEEE Transactions on*, 13(11):1459 – 1472, nov. 2004. ISSN 1057-7149. doi: 10.1109/TIP.2004.836169.

X. Li and J. Haupt. Identifying outliers in large matrices via randomized adaptive compressive sampling. *IEEE Transactions on Signal Processing*, 63(7):1792–1807, April 2015. ISSN 1053-587X. doi: 10.1109/TSP.2015.2401536.

Lester W. Mackey, Michael I. Jordan, and Ameet Talwalkar. Divide-and-conquer matrix factorization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1134–1142. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4486-divide-and-conquer-matrix-factorization.pdf.

Praneeth Netrapalli, Niranjan U N, Sujay Sanghavi, Animashree Anandkumar, and Prateek Jain. Non-convex robust pca. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 1107–1115. Curran Associates, Inc., 2014. URL http://papers.nips.cc/paper/5430-non-convex-robust-pca.pdf.

Dohyung Park, Anastasios Kyrillidis, Srinadh Bhojanapalli, Constantine Caramanis, and Sujay Sanghavi. Provable burer-monteiro factorization for a class of norm-constrained matrix problems. *arXiv preprint arXiv:1606.01316*, 2016.

Dohyung Park, Anastasios Kyrillidis, Constantine Carmanis, and Sujay Sanghavi. Non-square matrix sensing without spurious local minima via the Burer-Monteiro approach. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 65–74, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL http://proceedings.mlr.press/v54/park17a.html.

Alkes L Price, Nick J Patterson, Robert M Plenge, Michael E Weinblatt, Nancy A Shadick, and David Reich. Principal components analysis corrects for stratification in genome-wide association studies. *Nature genetics*, 38(8):904–909, aug 2006. ISSN 1061-4036 (Print). doi: 10.1038/ng1847.

M. Rahmani and G. K. Atia. High dimensional low rank plus sparse matrix decomposition. *IEEE Transactions on Signal Processing*, 65(8):2004–2019, April 2017. ISSN 1053-587X. doi: 10.1109/TSP.2017.2649482.

A. Ruhe. *Numerical Computation of Principal Components when Several Observations are Missing*. Univ., 1974. URL https://books.google.com/books?id=CgbyjgEACAAJ.

Uri Shalit, Daphna Weinshall, and Gal Chechik. Online learning in the embedded manifold of low-rank matrices. *J. Mach. Learn. Res.*, 13(1):429–458, February 2012. ISSN 1532-4435. URL http://dl.acm.org/citation.cfm?id=2503308.2188399.

J. Sun, Q. Qu, and J. Wright. Complete dictionary recovery over the sphere i: Overview and the geometric picture. *IEEE Transactions on Information Theory*, 63(2):853–884, Feb 2017. ISSN 0018-9448. doi: 10.1109/TIT.2016.2632162.

Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640):15, 2010.

Stephen Tu, Ross Boczar, Max Simchowitz, Mahdi Soltanolkotabi, and Ben Recht. Low-rank solutions of linear matrix equations via procrustes flow. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 964–973, 2016. URL http://jmlr.org/proceedings/papers/v48/tu16.html.

Bart Vandereycken. Low-rank matrix completion by riemannian optimization. *SIAM Journal on Optimization*, 23(2):1214–1236, 2013. doi: 10.1137/110845768. URL http://dx.doi.org/10.1137/110845768.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. In Yonina C. Eldar and GittaEditors Kutyniok, editors, *Compressed Sensing: Theory and Practice*, pages 210–268. Cambridge University Press, 2012. ISBN 9780511794308. doi: 10.1017/CBO9780511794308.006.

Lingxiao Wang, Xiao Zhang, and Quanquan Gu. A Unified Computational and Statistical Framework for Nonconvex Low-rank Matrix Estimation. In Aarti Singh and Jerry Zhu, editors, *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, pages 981–990, Fort Lauderdale, FL, USA, 20–22 Apr 2017. PMLR. URL http://proceedings.mlr.press/v54/wang17b.html.

Ke Wei, Jian-Feng Cai, Tony F. Chan, and Shingyu Leung. Guarantees of riemannian optimization for low rank matrix recovery. *SIAM Journal on Matrix Analysis and Applications*, 37(3):1198–1222, 2016. doi: 10.1137/15M1050525. URL `https://doi.org/10.1137/15M1050525`.

Xinyang Yi, Dohyung Park, Yudong Chen, and Constantine Caramanis. Fast algorithms for robust PCA via gradient descent. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4152–4160, 2016. URL `http://papers.nips.cc/paper/6445-fast-algorithms-for-robust-pca-via-gradient-descent`.

Journal of Machine Learning Research 19 (2018) 1-68 Submitted 11/17; Revised 7/18; Published 12/18

Improved Asynchronous Parallel Optimization Analysis for Stochastic Incremental Methods

Rémi Leblond REMI.LEBLOND@INRIA.FR
INRIA - Sierra Project-Team
École Normale Supérieure, Paris

Fabian Pedregosa F@BIANP.NET
INRIA - Sierra Project-Team
École Normale Supérieure, Paris

Simon Lacoste-Julien SLACOSTE@IRO.UMONTREAL.CA
Department of CS & OR (DIRO)
Université de Montréal, Montréal

Editor: Tong Zhang

Abstract

As data sets continue to increase in size and multi-core computer architectures are developed, asynchronous parallel optimization algorithms become more and more essential to the field of Machine Learning. Unfortunately, conducting the theoretical analysis asynchronous methods is difficult, notably due to the introduction of delay and inconsistency in inherently sequential algorithms. Handling these issues often requires resorting to simplifying but unrealistic assumptions. Through a novel perspective, we revisit and clarify a subtle but important technical issue present in a large fraction of the recent convergence rate proofs for asynchronous parallel optimization algorithms, and propose a simplification of the recently introduced "perturbed iterate" framework that resolves it. We demonstrate the usefulness of our new framework by analyzing three distinct asynchronous parallel incremental optimization algorithms: Hogwild (asynchronous Sgd), Kromagnon (asynchronous Svrg) and Asaga, a novel asynchronous parallel version of the incremental gradient algorithm Saga that enjoys fast linear convergence rates. We are able to both remove problematic assumptions and obtain better theoretical results. Notably, we prove that Asaga and Kromagnon can obtain a theoretical linear speedup on multi-core systems even without sparsity assumptions. We present results of an implementation on a 40-core architecture illustrating the practical speedups as well as the hardware overhead. Finally, we investigate the overlap constant, an ill-understood but central quantity for the theoretical analysis of asynchronous parallel algorithms. We find that it encompasses much more complexity than suggested in previous work, and often is order-of-magnitude bigger than traditionally thought.

Keywords: optimization, machine learning, large scale, asynchronous parallel, sparsity

1. Introduction

We consider the unconstrained optimization problem of minimizing a *finite sum* of smooth convex functions:

$$\min_{x\in\mathbb{R}^d} f(x), \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x), \tag{1}$$

where each f_i is assumed to be convex with L-Lipschitz continuous gradient, f is μ-strongly convex and n is large (for example, the number of data points in a regularized empirical risk minimization setting). We define a condition number for this problem as $\kappa := L/\mu$, as is standard in the finite sum literature.[1] A flurry of randomized incremental algorithms (which at each iteration select i at random and process only one gradient f_i') have recently been proposed to solve (1) with a fast[2] linear convergence rate, such as Sag (Le Roux et al., 2012), Sdca (Shalev-Shwartz and Zhang, 2013), Svrg (Johnson and Zhang, 2013) and Saga (Defazio et al., 2014). These algorithms can be interpreted as variance reduced versions of the stochastic gradient descent (Sgd) algorithm, and they have demonstrated both theoretical and practical improvements over Sgd (for the *finite sum* optimization problem 1).

In order to take advantage of the multi-core architecture of modern computers, the aforementioned optimization algorithms need to be adapted to the asynchronous parallel setting, where multiple threads work concurrently. Much work has been devoted recently in proposing and analyzing asynchronous parallel variants of algorithms such as Sgd (Niu et al., 2011), Sdca (Hsieh et al., 2015) and Svrg (Reddi et al., 2015; Mania et al., 2017; Zhao and Li, 2016). Among the incremental gradient algorithms with fast linear convergence rates that can optimize (1) in its general form, only Svrg had had an asynchronous parallel version proposed.[3] No such adaptation had been attempted for Saga until Leblond et al. (2017), even though one could argue that it is a more natural candidate as, contrarily to Svrg, it is not epoch-based and thus has no synchronization barriers at all. The present paper is an extended journal version of the conference paper from Leblond et al. (2017).

The usual frameworks for asynchronous analysis are quite intricate (see Section 2.2) and thus require strong simplifying assumptions. They are not well suited to the study of complex algorithms such as Saga. We therefore base our investigation on the newly proposed "perturbed iterate" framework introduced in Mania et al. (2017), which we also improve upon in order to properly analyze Saga. Deriving a framework in which the analysis of Saga is possible enables us to highlight the deficiencies of previous frameworks and to define a better alternative. Our new approach is not limited to Saga but can be used to investigate other algorithms and improve their existing bounds.

Contributions. In Section 2, we propose a simplification of the "perturbed iterate" framework from Mania et al. (2017) as a basis for our asynchronous convergence analysis. At the

1. Since we have assumed that each individual f_i is L-smooth, f itself is L-smooth – but its smoothness constant L_f could be much smaller. While the more classical condition number is $\kappa_b := L_f/\mu$, our rates are in terms of this bigger L/μ in this paper.
2. Their complexity in terms of gradient evaluations to reach an accuracy of ϵ is $O((n+\kappa)\log(1/\epsilon))$, in contrast to $O(n\kappa_b \log(1/\epsilon))$ for batch gradient descent in the worst case.
3. We note that Sdca requires the knowledge of an explicit μ-strongly convex regularizer in (1), whereas Sag / Saga are adaptive to any local strong convexity of f (Schmidt et al., 2016; Defazio et al., 2014). The variant of Svrg from Hofmann et al. (2015) is also adaptive (we review this variant in Section 4.1).

same time, through a novel perspective, we revisit and clarify a technical problem present in a large fraction of the literature on randomized asynchronous parallel algorithms (with the exception of Mania et al. 2017, which also highlights this issue): namely, they all assume unbiased gradient estimates, an assumption that is *inconsistent* with their proof technique without further unpractical synchronization assumptions.

In Section 3.1, we present a novel sparse variant of Saga that is more adapted to the parallel setting than the original Saga algorithm. In Section 3.2, we present Asaga, a lock-free asynchronous parallel version of Sparse Saga that does not require consistent read or write operations. We give a tailored convergence analysis for Asaga. Our main result states that Asaga obtains the same geometric convergence rate per update as Saga when the overlap bound τ (which scales with the number of cores) satisfies $\tau \leq \mathcal{O}(n)$ and $\tau \leq \mathcal{O}(\frac{1}{\sqrt{\Delta}} \max\{1, \frac{n}{\kappa}\})$, where $\Delta \leq 1$ is a measure of the sparsity of the problem. This notably implies that a linear speedup is theoretically possible even without sparsity in the well-conditioned regime where $n \gg \kappa$. This result is in contrast to previous analysis which always required some sparsity assumptions.

In Section 4, we revisit the asynchronous variant of Svrg from Mania et al. (2017), Kromagnon, while removing their gradient bound assumption (which was inconsistent with the strongly convex setting).[4] We prove that the algorithm enjoys the same fast rates of convergence as Svrg under similar conditions as Asaga – whereas the original paper only provided analysis for slower rates (in both the sequential and the asynchronous case), and thus less meaningful speedup results.

In Section 5, in order to show that our improved "after read" perturbed iterate framework can be used to revisit the analysis of other optimization routines with correct proofs that do not assume homogeneous computation, we provide the analysis of the Hogwild algorithm first introduced in Niu et al. (2011). Our framework allows us to remove the classic gradient bound assumption and to prove speedups in more realistic settings.

In Section 6, we provide a practical implementation of Asaga and illustrate its performance on a 40-core architecture, showing improvements compared to asynchronous variants of Svrg and Sgd. We also present experiments on the overlap bound τ, showing that it encompasses much more complexity than suggested in previous work.

Related Work. The seminal textbook of Bertsekas and Tsitsiklis (1989) provides most of the foundational work for parallel and distributed optimization algorithms. An asynchronous variant of Sgd with constant step size called Hogwild was presented by Niu et al. (2011); part of their framework of analysis was re-used and inspired most of the recent literature on asynchronous parallel optimization algorithms with convergence rates, including asynchronous variants of coordinate descent (Liu et al., 2015), Sdca (Hsieh et al., 2015), Sgd for non-convex problems (De Sa et al., 2015; Lian et al., 2015), Sgd for stochastic optimization (Duchi et al., 2015) and Svrg (Reddi et al., 2015; Zhao and Li, 2016). These papers make use of an unbiased gradient assumption that is not consistent with the proof technique, and thus suffers from technical problems[5] that we highlight in Section 2.2.

The "perturbed iterate" framework presented in Mania et al. (2017) is to the best of our knowledge the only one that does not suffer from this problem, and our convergence analysis builds heavily from their approach, while simplifying it. In particular, the authors assumed that f was both strongly convex and had a bound on the gradient, two *inconsistent* assumptions in the unconstrained setting that they analyzed. We overcome these difficulties by using tighter inequalities that remove the requirement of a bound on the gradient. We also propose a more convenient way to label the iterates (see Section 2.2). The sparse version of Saga that we propose is also inspired from the sparse version of Svrg proposed by Mania et al. (2017).

Reddi et al. (2015) presents a hybrid algorithm called Hsag that includes Saga and Svrg as special cases. Their asynchronous analysis is epoch-based though, and thus does not handle a fully asynchronous version of Saga as we do. Moreover, they require consistent reads and do not propose an efficient sparse implementation for Saga, in contrast to Asaga.

Pan et al. (2016) proposes a black box mini-batch algorithm to parallelize Sgd-like methods while maintaining serial equivalence through smart update partitioning. When the data set is sparse enough, they obtain speedups over "Hogwild" implementations of Svrg and Saga.[6] However, these "Hogwild" implementations appear to be quite suboptimal, as they do not leverage data set sparsity efficiently: they try to adapt the "lazy updates" trick from Schmidt et al. (2016) to the asynchronous parallel setting – which as discussed in Appendix E is extremely difficult – and end up making several approximations which severely penalize the performance of the algorithms. In particular, they have to use much smaller step sizes than in the sequential version, which makes for worse results.

Pedregosa et al. (2017) extend the Asaga algorithm presented in Section 3.2 to the proximal setting.

Notation. We denote by $\mathbb{E}$ a full expectation with respect to all the randomness in the system, and by $\mathbf{E}$ the *conditional* expectation of a random i (the index of the factor f_i chosen in Sgd and other algorithms), conditioned on all the past, where "past" will be clear from the context. $[x]_v$ represents the coordinate v of the vector $x \in \mathbb{R}^d$. For *sequential* algorithms, x^+ is the updated parameter vector after one algorithm iteration.

2. Revisiting the Perturbed Iterate Framework for Asynchronous Analysis

As most recent parallel optimization contributions, we use a similar hardware model to Niu et al. (2011). We consider multiple cores which all have read and write access to a shared memory. The cores update a central parameter vector in an asynchronous and lock-free fashion. Unlike Niu et al. (2011), we *do not* assume that the vector reads are consistent: multiple cores can read and write different coordinates of the shared vector at the same time. This also implies that a full vector read for a core might not correspond to any consistent state in the shared memory at any specific point in time.

4. Although the authors mention that this gradient bound assumption can be enforced through the use of a thresholding operator, they do not explain how to handle the interplay between this non-linear operator and the asynchrony of the algorithm. Their theoretical analysis relies on the linearity of the operations (e.g. to derive (Mania et al., 2017, Eq. (2.6))), and thus this claim is not currently supported by theory (note that a strongly convex function over an unbounded domain always has unbounded gradients).

5. Except (Duchi et al., 2015) that can be easily fixed by incrementing their global counter *before* sampling.

6. By "Hogwild", the authors mean asynchronous parallel variants where cores independently run the sequential update rule.

2.1. Perturbed Iterate Framework

We first review the "perturbed iterate" framework recently introduced by Mania et al. (2017) which will form the basis of our analysis. In the sequential setting, stochastic gradient descent and its variants can be characterized by the following update rule:

$$x_{t+1} = x_t - \gamma g(x_t, i_t)\,, \tag{2}$$

where i_t is a random variable independent from x_t and we have the unbiasedness condition $\mathbf{E}g(x_t, i_t) = f'(x_t)$ (recall that $\mathbf{E}$ is the relevant-past conditional expectation with respect to i_t).

Unfortunately, in the parallel setting, we manipulate stale, inconsistent reads of shared parameters and thus we do not have such a straightforward relationship. Instead, Mania et al. (2017) proposed to distinguish $\hat{x}_t$, the actual value read by a core to compute an update, from x_t, a "virtual iterate" that we can analyze and is *defined* by the update equation:

$$x_{t+1} := x_t - \gamma g(\hat{x}_t, i_t)\,. \tag{3}$$

We can thus interpret $\hat{x}_t$ as a noisy (perturbed) version of x_t due to the effect of asynchrony.

We formalize the precise meaning of x_t and $\hat{x}_t$ in the next section. We first note that all references mentioned in the related work section that analyzed asynchronous parallel randomized algorithms assumed that the following unbiasedness condition holds:

$$\text{[unbiasedness condition]} \quad \mathbf{E}[g(\hat{x}_t, i_t)|\hat{x}_t] = f'(\hat{x}_t).\ ^{7} \tag{4}$$

This condition is at the heart of most convergence proofs for randomized optimization methods.[8] Mania et al. (2017) correctly pointed out that most of the literature thus made the often implicit assumption that i_t is independent of $\hat{x}_t$. But as we explain below, this assumption is incompatible with a non-uniform asynchronous model in the analysis approach used in most of the recent literature.

2.2. On the Difficulty of Labeling the Iterates

Formalizing the meaning of x_t and $\hat{x}_t$ highlights a subtle but important difficulty arising when analyzing *randomized* parallel algorithms: what is the meaning of t? This is the problem of *labeling* the iterates for the purpose of the analysis, and this labeling can have randomness itself that needs to be taken in consideration when interpreting the meaning of an expression like $\mathbb{E}[x_t]$. In this section, we contrast three different approaches in a unified framework. We notably clarify the dependency issues that the labeling from Mania et al. (2017) resolves and propose a new, simpler labeling which allows for much simpler proof techniques.

7. We note that to be completely formal and define this conditional expectation more precisely, one would need to define another random vector that describes the entire system randomness, including all the reads, writes, delays, etc. Conditioning on $\hat{x}_t$ in (4) is actually a shorthand to indicate that we are conditioning on all the relevant "past" that defines both the value of $\hat{x}_t$ as well as the fact that it was the t^{th} labeled element. For clarity of exposition, we will not go into this level of technical detail, but one could define the appropriate sigma fields to condition on in order to make this equation fully rigorous.
8. A notable exception is Sag (Le Roux et al., 2012) which has biased updates and thus requires a significantly more complex convergence proof. Making Sag unbiased leads to Saga (Defazio et al., 2014), with a much simpler convergence proof.

We consider algorithms that execute in parallel the following four steps, where t is a global labeling that needs to be defined:[9]

1. Read the information in shared memory ($\hat{x}_t$).
2. Sample i_t.
3. Perform some computations using $(\hat{x}_t, i_t)$.
4. Write an update to shared memory.

(5)

The "After Write" Approach. We call the "after write" approach the standard global labeling scheme used in Niu et al. (2011) and re-used in all the later papers that we mentioned in the related work section, with the notable exceptions of Mania et al. (2017) and Duchi et al. (2015). In this approach, t is a (virtual) global counter recording the number of *successful writes* to the shared memory x (incremented after step 4 in 5); x_t thus represents the (true) content of the shared memory after t updates. The interpretation of the crucial equation (3) then means that $\hat{x}_t$ represents the (delayed) local copy value of the core that made the $(t+1)^{\text{th}}$ successful update; i_t represents the factor sampled by this core for this update. Notice that in this framework, the value of $\hat{x}_t$ and i_t is unknown at "time t"; we have to wait to the later time when the next core writes to memory to finally determine that its local variables are the ones labeled by t. We thus see that here $\hat{x}_t$ and i_t are not necessarily independent – they share dependence through the assignment of the t label. In particular, if some values of i_t yield faster updates than others, it will influence the label assignment defining $\hat{x}_t$. We provide a concrete example of this possible dependency in Figure 1.

The only way we can think to resolve this issue and ensure unbiasedness is to assume that the computation time for the algorithm running on a core is independent of the sample i chosen. This assumption seems overly strong in the context of potentially heterogeneous factors f_i's, and is thus a fundamental flaw for analyzing non-uniform asynchronous computation that has mostly been ignored in the recent asynchronous optimization literature.[10]

9. Observe that contrary to most asynchronous algorithms, we choose to read the shared parameter vector *before* sampling the next data point. We made this design choice to emphasize that in order for $\hat{x}_t$ and i_t to be independent – which will prove crucial for the analysis – the reading of the shared parameter has to be independent of the sampled data point. Although in practice one would prefer to only read the necessary parameters *after* sampling the relevant data point, for the sake of the analysis we cannot allow this source of dependence. We note that our analysis could also handle reading the parameter first and then sampling as long as independence is ensured, but for clarity of presentation, we decided to make this independence explicit.

   Mania et al. (2017) make the opposite presentation choice. In their main analysis, they explicitly assume that $\hat{x}_t$ and i_t are independent, although they explain that it is not the case in practical implementations. The authors then propose a scheme to handle the dependency directly in their appendix. However, this "fix" can only be applied in a restricted setup: only for the Hogwild algorithm, with the assumption that the norm of the gradient is uniformly bounded. Furthermore, even in this restricted setup, the scheme leads to worsened theoretical results (the bound on τ is κ^2 worse). Applying it to a more complex algorithm such as Kromagnon or Asaga would mean overcoming several significant hurdles and is thus still an open problem.

   In the absence of a better option, we choose to enforce the independence of $\hat{x}_t$ and i_t with our modified steps ordering.
10. We note that Bertsekas and Tsitsiklis (1989) briefly discussed this issue (see Section 7.8.3), stressing that their analysis for SGD required that the scheduling of computation was independent from the randomness from SGD, but they did not offer any solution if this assumption was not satisfied. Both the "before read" labeling from Mania et al. (2017) and our proposed "after read" labeling resolve this issue.

	f_1	f_2		f_1	f_2		f_1	f_2		f_1	f_2
core 1	×		core 1	×		core 1		×	core 1		×
core 2	×		core 2		×	core 2	×		core 2		×
$f'_{i_0}(\hat{x}_0)$	$f'_1(\hat{x}_0)$		$f'_1(\hat{x}_0)$			$f'_1(\hat{x}_0)$			$f'_2(\hat{x}_0)$		

Figure 1: Suppose that we have two cores and that f has two factors: f_1 which has support on only one variable, and f_2 which has support on 10^6 variables and thus yields a gradient step that is significantly more expensive to compute. x_0 is the initial content of the memory, and we do not know yet whether $\hat{x}_0$ is the local copy read by the first core or the second core, but we are sure that $\hat{x}_0 = x_0$ as no update can occur in shared memory without incrementing the counter. There are four possibilities for the next step defining x_1 depending on which index i was sampled on each core. If any core samples $i = 1$, we know that $x_1 = x_0 - \gamma f'_1(x_0)$ as it will be the first (much faster update) to complete. This happens in 3 out of 4 possibilities; we thus have that $\mathbb{E}x_1 = x_0 - \gamma(\frac{3}{4}f'_1(x_0) + \frac{1}{4}f'_2(x_0))$. We see that this analysis scheme *does not* satisfy the crucial unbiasedness condition (4). To understand this subtle point better, note that in this very simple example, i_0 and i_1 are not independent. We can show that $P(i_1 = 2 \mid i_0 = 2) = 1$. They share dependency *through the labeling assignment*.

The "Before Read" Approach. Mania et al. (2017) address this issue by proposing instead to increment the global t counter just *before* a new core starts to *read* the shared memory (before step 1 in 5). In their framework, $\hat{x}_t$ represents the (inconsistent) read that was made by this core in this computational block, and i_t represents the chosen sample. The update rule (3) represents a *definition* of the meaning of x_t, which is now a "virtual iterate" as it does not necessarily correspond to the content of the shared memory at any point. The real quantities manipulated by the algorithm in this approach are the $\hat{x}_t$'s, whereas x_t is used only for the analysis – consequently, the critical quantity we want to see vanish is $\mathbb{E}\|\hat{x}_t - x^*\|^2$. The independence of i_t with $\hat{x}_t$ can be simply enforced in this approach by making sure that the way the shared memory x is read does not depend on i_t (e.g. by reading all its coordinates in a fixed order). Note that this implies that we have to read all of x's coordinates, regardless of the size of f_{i_t}'s support. This is a much weaker condition than the assumption that all the computation in a block does not depend on i_t as required by the "after write" approach, and is thus more reasonable.

A New Global Ordering: the "After Read" Approach. The "before read" approach gives rise to the following complication in the analysis: $\hat{x}_t$ can depend on i_r for $r > t$. This is because t is a global time ordering only on the assignment of computation to a core, not on when $\hat{x}_t$ was finished being read. This means that we need to consider both the "future" and the "past" when analyzing x_t. To simplify the analysis, we thus propose a third way to label the iterates that we call "after read": $\hat{x}_t$ represents the $(t+1)^{\text{th}}$ *fully completed read* (t incremented after step 1 in 5). As in the "before read" approach, we can ensure that i_t is independent of $\hat{x}_t$ by ensuring that how we read does not depend on i_t. But unlike in the "before read" approach, t here now does represent a global ordering on the $\hat{x}_t$ iterates – and thus we have that i_r is independent of $\hat{x}_t$ for $r > t$. Again using (3) as the definition of the virtual iterate x_t as in the perturbed iterate framework, we then have a very simple form for

the value of x_t and $\hat{x}_t$ (assuming atomic writes, see Property 5 below):

$$\begin{aligned} x_t &= x_0 - \gamma \sum_{u=0}^{t-1} g(\hat{x}_u, \hat{\alpha}^u, i_u)\,; \\ [\hat{x}_t]_v &= [x_0]_v - \gamma \sum_{\substack{u=0 \\ \text{u s.t. coordinate } v \text{ was written} \\ \text{for } u \text{ before } t}}^{t-1} [g(\hat{x}_u, \hat{\alpha}^u, i_u)]_v\,. \end{aligned} \tag{6}$$

This proved crucial for our Asaga proof and allowed us to obtain better bounds for Hogwild and the Kromagnon algorithm presented in Mania et al. (2017).

The main idea of the perturbed iterate framework is to use this handle on $\hat{x}_t - x_t$ to analyze the convergence for x_t. As x_t is a virtual quantity, Mania et al. (2017) supposed that there exists an index T such that x_T lives in shared memory (T is a pre-set final iteration number after which all computation is completed, which means $x_T = \hat{x}_T$) and gave their convergence result for this x_T.

In this paper, we instead state the convergence results directly in terms of $\hat{x}_t$, thus avoiding the need for an unwieldy pre-set final iteration counter, and also enabling guarantees during the entire course of the algorithm.

Remark 1 *As mentioned in footnote 9, Mania et al. (2017) choose to sample a data point first and only then read the shared parameter vector in* (5)*. One advantage of this option is that it allows for reading only the relevant dimensions of the parameter vector, although it means losing the crucial independence property between $\hat{x}_t$ and i_t.*

We can thus consider that their labeling approach is "after sampling" rather than "before read" (both are equivalent given their ordering). If we take this view, then by switching the order of the sampling and the reading steps in their setup, the "after sampling" approach becomes equivalent to our proposed "after read" labeling.

However, the framework in which they place their analysis is the "before read" approach as described above, which results in having to take into account troublesome "future" terms in (6)*. These additional terms make the analysis considerably harder and ultimately lead to worse theoretical results.*

3. Asynchronous Parallel Sparse Saga

We start by presenting Sparse Saga, a sparse variant of the Saga algorithm that is more adapted to the asynchronous parallel setting. We then introduce Asaga, the asynchronous parallel version of Sparse Saga. Finally, we state both convergence and speedup results for Asaga and give an outline of their proofs.

3.1. Sparse Saga

Borrowing our notation from Hofmann et al. (2015), we first present the original Saga algorithm and then describe our novel sparse variant.

Original Saga Algorithm. The standard Saga algorithm (Defazio et al., 2014) maintains two moving quantities to optimize (1): the current iterate x and a table (memory) of historical

gradients $(\alpha_i)_{i=1}^n$.[11] At every iteration, the Saga algorithm samples uniformly at random an index $i \in \{1, \dots, n\}$, and then executes the following update on x and α (for the unconstrained optimization version):

$$x^+ = x - \gamma\big(f_i'(x) - \alpha_i + \bar{\alpha}\big); \qquad \alpha_i^+ = f_i'(x), \tag{7}$$

where γ is the step size and $\bar{\alpha} := {}^1\!/_n \sum_{i=1}^n \alpha_i$ can be updated efficiently in an online fashion. Crucially, $\mathbf{E}\alpha_i = \bar{\alpha}$ and thus the update direction is unbiased ($\mathbf{E}x^+ = x - \gamma f'(x)$). Furthermore, it can be proven (see Defazio et al., 2014) that under a reasonable condition on γ, the update has vanishing variance, which enables the algorithm to converge linearly with a constant step size.

Motivation for a Variant. In its current form, every Saga update is dense even if the individual gradients are sparse due to the historical gradient ($\bar{\alpha}$) term. Schmidt et al. (2016) introduced an implementation technique denoted lagged updates in which each iteration has a cost proportional to the size of the support of $f_i'(x)$. However, this technique involves keeping track of past updates and is not easily adaptable to the parallel setting (see Appendix E). We therefore introduce Sparse Saga, a novel variant which explicitly takes sparsity into account and is easily parallelizable.

Sparse* Saga *Algorithm. As in the Sparse Svrg algorithm proposed in Mania et al. (2017), we obtain Sparse Saga by a simple modification of the parameter update rule in (7) where $\bar{\alpha}$ is replaced by a sparse version equivalent in expectation:

$$x^+ = x - \gamma(f_i'(x) - \alpha_i + D_i\bar{\alpha}), \tag{8}$$

where D_i is a diagonal matrix that makes a weighted projection on the support of f_i'. More precisely, let S_i be the support of the gradient f_i' function (i.e., the set of coordinates where f_i' can be nonzero). Let D be a $d \times d$ diagonal reweighting matrix, with coefficients ${}^1\!/_{p_v}$ on the diagonal, where p_v is the probability that dimension v belongs to S_i when i is sampled uniformly at random in $\{1, ..., n\}$. We then define $D_i := P_{S_i} D$, where P_{S_i} is the projection onto S_i. The reweighting by D ensures that $\mathbf{E}D_i\bar{\alpha} = \bar{\alpha}$, and thus that the update is still unbiased despite the sparsifying projection.

***Convergence Result for (Serial) Sparse* Saga.** For clarity of exposition, we model our convergence result after the simple form of Hofmann et al. (2015, Corollary 3). Note that the rate we obtain for Sparse Saga is the same as the one obtained in the aforementioned reference for Saga.

Theorem 2 *Let $\gamma = \frac{a}{5L}$ for any $a \leq 1$. Then* Sparse Saga *converges geometrically in expectation with a rate factor of at least $\rho(a) = \frac{1}{5}\min\left\{\frac{1}{n}, a\frac{1}{\kappa}\right\}$, i.e., for x_t obtained after t updates, we have $\mathbb{E}\|x_t - x^*\|^2 \leq (1-\rho)^t C_0$, where $C_0 := \|x_0 - x^*\|^2 + \frac{1}{5L^2}\sum_{i=1}^n \|\alpha_i^0 - f_i'(x^*)\|^2$.*

11. For linear predictor models, the memory α_i^0 can be stored as a scalar. Following Hofmann et al. (2015), α_i^0 can be initialized to any convenient value (typically 0), unlike the prescribed $f_i'(x_0)$ analyzed in (Defazio et al., 2014).

Proof outline. We reuse the proof technique from Hofmann et al. (2015), in which a combination of classical strong convexity and Lipschitz inequalities is used to derive the inequality (Hofmann et al., 2015, Lemma 1):

$$\mathbf{E}\|x^+ - x^*\|^2 \leq (1-\gamma\mu)\|x - x^*\|^2 + 2\gamma^2\mathbf{E}\|\alpha_i - f_i'(x^*)\|^2 + (4\gamma^2 L - 2\gamma)\big(f(x) - f(x^*)\big). \tag{9}$$

This gives a contraction term. A Lyapunov function is then defined to control the two other terms. To ensure our variant converges at the same rate as regular Saga, we only need to prove that the above inequality (Hofmann et al., 2015, Lemma 1) is still verified. To prove this, we derive close variants of equations (6) and (9) in their paper. The rest of the proof can be reused without modification. The full details can be found in Appendix A.

Comparison with Lagged Updates. The lagged updates technique in Saga is based on the observation that the updates for component $[x]_v$ need not be applied until this coefficient needs to be accessed, that is, until the next iteration t such that $v \in S_{i_t}$. We refer the reader to Schmidt et al. (2016) for more details.

Interestingly, the expected number of iterations between two steps where a given dimension v is in the support of the partial gradient is p_v^{-1}, where p_v is the probability that v is in the support of the partial gradient at a given step. p_v^{-1} is precisely the term which we use to multiply the update to $[x]_v$ in Sparse Saga. Therefore one may see the updates in Sparse Saga as *anticipated* updates, whereas those in the Schmidt et al. (2016) implementation are *lagged*.

The two algorithms appear to be very close, even though Sparse Saga uses an expectation to multiply a given update whereas the lazy implementation uses a random variable (with the same expectation). Sparse Saga therefore uses a slightly more aggressive strategy, which may explain the result of our experiments (see Section 6.3): both Sparse Saga and Saga with lagged updates had similar convergence in terms of number of iterations, with the Sparse Saga scheme being slightly faster in terms of runtime.

Although Sparse Saga requires the computation of the p_v probabilities, this can be done during a first pass throughout the data (during which constant step size Sgd may be used) at a negligible cost.

3.2. Asynchronous Parallel Sparse Saga

We describe Asaga, a sparse asynchronous parallel implementation of Sparse Saga, in Algorithm 1 in the theoretical form that we analyze, and in Algorithm 2 as its practical implementation. We state our convergence result and analyze our algorithm using the improved perturbed iterate framework.

In the specific case of (Sparse) Saga, we have to add the additional read memory argument $\hat{\alpha}^t$ to our perturbed update (3):

$$\begin{aligned} x_{t+1} &:= x_t - \gamma g(\hat{x}_t, \hat{\alpha}^t, i_t); \\ g(\hat{x}_t, \hat{\alpha}^t, i_t) &:= f_{i_t}'(\hat{x}_t) - \hat{\alpha}_{i_t}^t + D_{i_t}\left({}^1\!/_n \sum_{i=1}^n \hat{\alpha}_i^t\right). \end{aligned} \tag{10}$$

Before stating our convergence result, we highlight some properties of Algorithm 1 and make one central assumption.

Algorithm 1 ASAGA (analyzed algorithm)

1: Initialize shared variables x and $(\alpha_i)_{i=1}^n$
2: **keep doing in parallel**
3: $\quad \hat{x} =$ inconsistent read of x
4: $\quad \forall j$, $\hat{\alpha}_j =$ inconsistent read of α_j
5: $\quad$ Sample i uniformly in $\{1, ..., n\}$
6: $\quad$ Let S_i be f_i's support
7: $\quad [\bar{\alpha}]_{S_i} = 1/n \sum_{k=1}^n [\hat{\alpha}_k]_{S_i}$
8:
9: $\quad [\delta x]_{S_i} = -\gamma(f_i'(\hat{x}) - \hat{\alpha}_i + D_i[\bar{\alpha}]_{S_i})$
10: $\quad$ **for** v **in** S_i **do**
11: $\quad\quad [x]_v \leftarrow [x]_v + [\delta x]_v$ // atomic
12: $\quad\quad [\alpha_i]_v \leftarrow [f_i'(\hat{x})]_v$
13: $\quad\quad$ // '$\leftarrow$' denotes a shared memory update.
14: $\quad$ **end for**
15: **end parallel loop**

Algorithm 2 ASAGA (implementation)

1: Initialize shared x, $(\alpha_i)_{i=1}^n$ and $\bar{\alpha}$
2: **keep doing in parallel**
3: $\quad$ Sample i uniformly in $\{1, ..., n\}$
4: $\quad$ Let S_i be f_i's support
5: $\quad [\hat{x}]_{S_i} =$ inconsistent read of x on S_i
6: $\quad \hat{\alpha}_i =$ inconsistent read of α_i
7: $\quad [\bar{\alpha}]_{S_i} =$ inconsistent read of $\bar{\alpha}$ on S_i
8: $\quad [\delta\alpha]_{S_i} = f_i'([\hat{x}]_{S_i}) - \hat{\alpha}_i$
9: $\quad [\delta x]_{S_i} = -\gamma([\delta\alpha]_{S_i} + D_i[\bar{\alpha}]_{S_i})$
10: $\quad$ **for** v **in** S_i **do**
11: $\quad\quad [x]_v \leftarrow [x]_v + [\delta x]_v$ // atomic
12: $\quad\quad [\alpha_i]_v \leftarrow [\alpha_i]_v + [\delta\alpha]_v$ // atomic
13: $\quad\quad [\bar{\alpha}]_v \leftarrow [\bar{\alpha}]_v + 1/n[\delta\alpha]_v$ // atomic
14: $\quad$ **end for**
15: **end parallel loop**

Property 3 (independence) *Given the "after read" global ordering, i_r is independent of $\hat{x}_t$ $\forall r \geq t$.*

The independence property for $r = t$ is assumed in most of the parallel optimization literature, even though it is not verified in case the "after write" labeling is used. We emulate Mania et al. (2017) and *enforce* this independence in Algorithm 1 by having the core read all the shared data parameters and historical gradients before starting their iterations. Although this is too expensive to be practical if the data is sparse, this is required by the theoretical Algorithm 1 that we can analyze. The independence for $r > t$ is a consequence of using the "after read" global ordering instead of the "before read" one.

Property 4 (unbiased estimator) *The update, $g_t := g(\hat{x}_t, \hat{\alpha}^t, i_t)$, is an unbiased estimator of the true gradient at $\hat{x}_t$, i.e.* (10) *yields* (4) *in conditional expectation.*

This property is crucial for the analysis, as in most related literature. It follows by the independence of i_t with $\hat{x}_t$ and from the computation of $\bar{\alpha}$ on line 7 of Algorithm 1, which ensures that $\mathbb{E}\hat{\alpha}_i = 1/n \sum_{k=1}^n [\hat{\alpha}_k]_{S_i} = [\bar{\alpha}]_{S_i}$, making the update unbiased. In practice, recomputing $\bar{\alpha}$ is not optimal, but storing it instead introduces potential bias issues in the proof (as detailed in Appendix F.3).

Property 5 (atomicity) *The shared parameter coordinate update of $[x]_v$ on line 11 is atomic.*

Since our updates are additions, there are no overwrites, even when several cores compete for the same resources. In practice, this is enforced by using *compare-and-swap* semantics, which are heavily optimized at the processor level and have minimal overhead. Our experiments with non-thread safe algorithms (i.e. where this property is not verified, see Figure 7 of Appendix F) show that compare-and-swap is necessary to optimize to high accuracy.

Finally, as is standard in the literature, we make an assumption on the maximum delay that asynchrony can cause – this is the *partially asynchronous* setting as defined in Bertsekas and Tsitsiklis (1989):

Assumption 6 (bounded overlaps) *We assume that there exists a uniform bound, called τ, on the maximum number of iterations that can overlap together. We say that iterations r and t overlap if at some point they are processed concurrently. One iteration is being processed from the start of the reading of the shared parameters to the end of the writing of its update. The bound τ means that iterations r cannot overlap with iteration t for $r \geq t + \tau + 1$, and thus that every coordinate update from iteration t is successfully written to memory before the iteration $t + \tau + 1$ starts.*

Our result will give us conditions on τ subject to which we have linear speedups. τ is usually seen as a proxy for p, the number of cores (which lowerbounds it). However, though τ appears to depend linearly on p, it actually depends on several other factors (notably the data sparsity distribution) and can be orders of magnitude bigger than p in real-life experiments. We can upper bound τ by $(p-1)R$, where R is the ratio of the maximum over the minimum iteration time (which encompasses theoretical aspects as well as hardware overhead). More details can be found in Section 6.7.

Explicit effect of asynchrony. By using the overlap Assumption 6 in the expression (6) for the iterates, we obtain the following explicit effect of asynchrony that is crucially used in our proof:

$$\hat{x}_t - x_t = \gamma \sum_{u=(t-\tau)_+}^{t-1} G_u^t g(\hat{x}_u, \hat{\alpha}^u, i_u), \tag{11}$$

where G_u^t are $d \times d$ diagonal matrices with terms in $\{0, +1\}$. From our definition of t and x_t, it is clear that every update in $\hat{x}_t$ is already in x_t – this is the 0 case. Conversely, some updates might be late: this is the $+1$ case. $\hat{x}_t$ may be lacking some updates from the "past" in some sense, whereas given our global ordering definition, it cannot contain updates from the "future".

3.3. Convergence and Speedup Results

We now state our main theoretical results. We give a detailed outline of the proof in Section 3.3.2 and its full details in Appendix B.

We first define a notion of problem sparsity, as it will appear in our results.

Definition 7 (Sparsity) *As in Niu et al. (2011), we introduce $\Delta_r := \max_{v=1..d} |\{i : v \in S_i\}|$. Δ_r is the maximum right-degree in the bipartite graph of the factors and the dimensions, i.e., the maximum number of data points with a specific feature. For succinctness, we also define $\Delta := \Delta_r / n$. We have $1 \leq \Delta_r \leq n$, and hence $1/n \leq \Delta \leq 1$.*

3.3.1. Convergence and Speedup Statements

Theorem 8 (Convergence guarantee and rate of Asaga) *Suppose* $\tau < n/10$.[12] *Let*

$$a^*(\tau) := \frac{1}{32\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)} \quad \text{where } \xi(\kappa,\Delta,\tau) := \sqrt{1+\frac{1}{8\kappa}\min\{\frac{1}{\sqrt{\Delta}},\tau\}} \quad \text{(note that } \xi(\kappa,\Delta,\tau)\approx 1 \text{ unless } \kappa < 1/\sqrt{\Delta}\ (\leq\sqrt{n})\text{).} \tag{12}$$

For any step size $\gamma = \frac{a}{L}$ *with* $a \leq a^*(\tau)$*, the inconsistent read iterates of Algorithm 1 converge in expectation at a geometric rate of at least:* $\rho(a) = \frac{1}{5}\min\left\{\frac{1}{n}, a\frac{1}{\kappa}\right\}$*, i.e.,* $\mathbb{E}f(\hat{x}_t) - f(x^*) \leq (1-\rho)^t \tilde{C}_0$*, where* $\tilde{C}_0$ *is a constant independent of* t *(*$\approx \frac{n}{\gamma}C_0$ *with* C_0 *as defined in Theorem 2).*

This result is very close to Saga's original convergence theorem, but with the maximum step size divided by an extra $1+\tau\sqrt{\Delta}$ factor. Referring to Hofmann et al. (2015) and our own Theorem 2, the rate factor for Saga is $\min\{1/n, a/\kappa\}$ up to a constant factor. Comparing this rate with Theorem 8 and inferring the conditions on the maximum step size $a^*(\tau)$, we get the following conditions on the overlap τ for Asaga to have the same rate as Saga (comparing upper bounds).

Corollary 9 (Speedup condition) *Suppose* $\tau \leq \mathcal{O}(n)$ *and* $\tau \leq \mathcal{O}(\frac{1}{\sqrt{\Delta}}\max\{1, \frac{n}{\kappa}\})$*. Then using the step size* $\gamma = a^*(\tau)/L$ *from* (12)*,* Asaga *converges geometrically with rate factor* $\Omega(\min\{\frac{1}{n}, \frac{1}{\kappa}\})$ *(similar to* Saga*), and is thus linearly faster than its sequential counterpart up to a constant factor. Moreover, if* $\tau \leq \mathcal{O}(\frac{1}{\sqrt{\Delta}})$*, then a universal step size of* $\Theta(\frac{1}{L})$ *can be used for* Asaga *to be adaptive to local strong convexity with a similar rate to* Saga *(i.e., knowledge of* κ *is not required).*

Interestingly, in the well-conditioned regime ($n > \kappa$, where Saga enjoys a range of step sizes which all give the same contraction ratio), Asaga enjoys the same rate as Saga even in the non-sparse regime ($\Delta = 1$) for $\tau < \mathcal{O}(n/\kappa)$. This is in contrast to the previous work on asynchronous incremental gradient methods which required some kind of sparsity to get a theoretical linear speedup over their sequential counterpart (Niu et al., 2011; Mania et al., 2017). In the ill-conditioned regime ($\kappa > n$), sparsity is required for a linear speedup, with a bound on τ of $\mathcal{O}(\sqrt{n})$ in the best-case (though degenerate) scenario where $\Delta = 1/n$.

The proof for Corollary 9 can be found in Appendix B.9.

Comparison to related work.

- We give the first convergence analysis for an asynchronous parallel version of Saga (note that Reddi et al. (2015) only covers an epoch based version of Saga with random stopping times, a fairly different algorithm).
- Theorem 8 can be directly extended to the a parallel extension of the Svrg version from Hofmann et al. (2015) which is adaptive to the local strong convexity with similar rates (see Section 4.2).
- In contrast to the parallel Svrg analysis from Reddi et al. (2015, Thm. 2), our proof technique handles inconsistent reads and a non-uniform processing speed across f_i's. Our bounds are similar (noting that Δ is equivalent to theirs), except for the adaptivity to local strong convexity: Asaga does not need to know κ for optimal performance, contrary to parallel Svrg (see Section 4 for more details).
- In contrast to the Svrg analysis from Mania et al. (2017, Thm. 14), we obtain a better dependence on the condition number in our rate ($1/\kappa$ vs. $1/\kappa^2$ on their work) and on the sparsity (they obtain $\tau \leq \mathcal{O}(\Delta^{-1/3})$), while we furthermore remove their gradient bound assumption. We also give our convergence guarantee on $\hat{x}_t$ *during* the algorithm, whereas they only bound the error for the "last" iterate x_T.

12. Asaga can actually converge for any τ, but the maximum step size then has a term of $\exp(\tau/n)$ in the denominator with much worse constants. See Appendix B.7.

3.3.2. Proof Outline of Theorem 8

We give here an extended outline of the proof. We detail key lemmas in Section 3.3.3.

Initial recursive inequality. Let $g_t := g(\hat{x}_t, \hat{\alpha}^t, i_t)$. By expanding the update equation (10) defining the virtual iterate x_{t+1} and introducing $\hat{x}_t$ in the inner product term, we obtain:

$$\begin{aligned}\|x_{t+1} - x^*\|^2 &= \|x_t - \gamma g_t - x^*\|^2 \\ &= \|x_t - x^*\|^2 + \gamma^2\|g_t\|^2 - 2\gamma\langle x_t - x^*, g_t\rangle \\ &= \|x_t - x^*\|^2 + \gamma^2\|g_t\|^2 - 2\gamma\langle \hat{x}_t - x^*, g_t\rangle + 2\gamma\langle \hat{x}_t - x_t, g_t\rangle\,. \end{aligned} \tag{13}$$

Note that we introduce $\hat{x}_t$ in the inner product because g_t is a function of $\hat{x}_t$, not x_t.

In the sequential setting, we require i_t to be independent of x_t to obtain unbiasedness. In the perturbed iterate framework, we instead require that i_t is independent of $\hat{x}_t$ (see Property 3). This crucial property enables us to use the unbiasedness condition (4) to write: $\mathbb{E}\langle \hat{x}_t - x^*, g_t\rangle = \mathbb{E}\langle \hat{x}_t - x^*, f'(\hat{x}_t)\rangle$. Taking the expectation of (13) and using this unbiasedness condition we obtain an expression that allows us to use the μ-strong convexity of f:[13]

$$\langle \hat{x}_t - x^*, f'(\hat{x}_t)\rangle \geq f(\hat{x}_t) - f(x^*) + \frac{\mu}{2}\|\hat{x}_t - x^*\|^2. \tag{14}$$

With further manipulations on the expectation of (13), including the use of the standard inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$ (see Appendix B.1), we obtain our basic recursive contraction inequality:

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + \gamma^2\mathbb{E}\|g_t\|^2 \underbrace{+\gamma\mu\mathbb{E}\|\hat{x}_t - x_t\|^2 + 2\gamma\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle}_{\text{additional asynchrony terms}} - 2\gamma e_t\,, \tag{15}$$

where $a_t := \mathbb{E}\|x_t - x^*\|^2$ and $e_t := \mathbb{E}f(\hat{x}_t) - f(x^*)$.

Inequality (15) is a midway point between the one derived in the proof of Lemma 1 in Hofmann et al. (2015) and Equation (2.5) in Mania et al. (2017), because we use the tighter strong convexity bound (14) than in the latter (giving us the important extra term $-2\gamma e_t$).

In the sequential setting, one crucially uses the negative suboptimality term $-2\gamma e_t$ to cancel the variance term $\gamma^2\mathbb{E}\|g_t\|^2$ (thus deriving a condition on γ). In our setting, we need

13. Note that here is our departure point with Mania et al. (2017) who replaced the $f(\hat{x}_t) - f(x^*)$ term with the lower bound $\frac{\mu}{2}\|\hat{x}_t - x^*\|^2$ in this relationship (see their Equation (2.4)), thus yielding an inequality too loose afterwards to get the fast rates for Svrg.

to bound the additional asynchrony terms using the same negative suboptimality in order to prove convergence and speedup for our parallel algorithm – this will give stronger constraints on the maximum step size.

The rest of the proof then proceeds as follows:

1. By using the expansion (11) for $\hat{x}_t - x_t$, we can bound the additional asynchrony terms in (15) in terms of the past updates $(\mathbb{E}\|g_u\|^2)_{u\leq t}$. This gives Lemma 10 below.
2. We then bound the updates $\mathbb{E}\|g_t\|^2$ in terms of past suboptimalities $(e_v)_{v\leq u}$ by using standard Saga inequalities and carefully analyzing the update rule for α_i^+ (7) in expectation. This gives Lemma 13 below.
3. By applying Lemma 13 to the result of Lemma 10, we obtain a master contraction inequality (27) in terms of a_{t+1}, a_t and $(e_u)_{u\leq t}$.
4. We define a novel Lyapunov function $\mathcal{L}_t = \sum_{u=0}^{t}(1-\rho)^{t-u}a_u$ and manipulate the master inequality to show that $\mathcal{L}_t$ is bounded by a contraction, subject to a maximum step size condition on γ (given in Lemma 14 below).
5. Finally, we unroll the Lyapunov inequality to get the convergence Theorem 8.

3.3.3. Details

We list the key lemmas below with their proof sketch, and pointers to the relevant parts of Appendix B for detailed proofs.

Lemma 10 (Inequality in terms of $g_t := g(\hat{x}_t, \hat{\alpha}^t, i_t)$) *For all $t \geq 0$:*

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + \gamma^2 C_1 \mathbb{E}\|g_t\|^2 + \gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 - 2\gamma e_t\,, \tag{16}$$

$$\textit{where} \quad C_1 := 1 + \sqrt{\Delta}\tau \quad \textit{and} \quad C_2 := \sqrt{\Delta} + \gamma\mu C_1\,. \quad {}^{14} \tag{17}$$

To prove this lemma we need to bound both $\mathbb{E}\|\hat{x}_t - x^*\|^2$ and $\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle$ with respect to $(\mathbb{E}\|g_u\|^2)_{u\leq t}$. We achieve this by crucially using Equation (11), together with the following proposition, which we derive by a combination of Cauchy-Schwartz and our sparsity definition (see Section B.2).

Proposition 11 *For any $u \neq t$,*

$$\mathbb{E}|\langle g_u, g_t\rangle| \leq \frac{\sqrt{\Delta}}{2}(\mathbb{E}\|g_u\|^2 + \mathbb{E}\|g_t\|^2)\,. \tag{18}$$

To derive this essential inequality for both the right-hand-side terms of Eq. (18), we start by proving a relevant property of Δ. We reuse the sparsity constant introduced in Reddi et al. (2015) and relate it to the one we have defined earlier, Δ_r:

14. Note that C_2 depends on γ. In the rest of the paper, we write $C_2(\gamma)$ instead of C_2 when we want to draw attention to that dependency.

Remark 12 *Let D be the smallest constant such that:*

$$\mathbf{E}\|x\|_i^2 = \frac{1}{n}\sum_{i=1}^{n}\|x\|_i^2 \leq D\|x\|^2 \quad \forall x \in \mathbb{R}^d, \tag{19}$$

where $\|.\|_i$ is defined to be the ℓ_2-norm restricted to the support S_i of f_i. We have:

$$D = \frac{\Delta_r}{n} = \Delta\,. \tag{20}$$

Proof We have:

$$\mathbf{E}\|x\|_i^2 = \frac{1}{n}\sum_{i=1}^{n}\|x\|_i^2 = \frac{1}{n}\sum_{i=1}^{n}\sum_{v\in S_i}[x]_v^2 = \frac{1}{n}\sum_{v=1}^{d}\sum_{i|v\in S_i}[x]_v^2 = \frac{1}{n}\sum_{v=1}^{d}\delta_v[x]_v^2\,, \tag{21}$$

where $\delta_v := |(i \mid v \in S_i)|$. This implies:

$$D \geq \frac{1}{n}\sum_{v=1}^{d}\delta_v\frac{[x]_v^2}{\|x\|^2}\,. \tag{22}$$

Since D is the minimum constant satisfying this inequality, we have:

$$D = \max_{x\in\mathbb{R}^d}\frac{1}{n}\sum_{v=1}^{d}\delta_v\frac{[x]_v^2}{\|x\|^2}\,. \tag{23}$$

We need to find x such that it maximizes the right-hand side term. Note that the vector $([x]_v^2/\|x\|^2)_{v=1..d}$ is in the unit probability simplex, which means that an equivalent problem is the maximization over all convex combinations of $(\delta_v)_{v=1..d}$. This maximum is found by putting all the weight on the maximum δ_v, which is Δ_r by definition.

This implies that $\Delta = \Delta_r/n$ is indeed the smallest constant satisfying (19). ■

Proof of Proposition 11 Let $u \neq t$. Without loss of generality, $u < t$.[15] Then:

$$\begin{aligned}
\mathbb{E}|\langle g_u, g_t\rangle| &\leq \mathbb{E}\|g_u\|_{i_t}\|g_t\| && \text{(Sparse inner product; support of } g_t \text{ is } S_{i_t})\\
&\leq \sqrt{\mathbb{E}\|g_u\|_{i_t}^2}\sqrt{\mathbb{E}\|g_t\|^2} && \text{(Cauchy-Schwarz for expectations)}\\
&\leq \sqrt{\Delta\mathbb{E}\|g_u\|^2}\sqrt{\mathbb{E}\|g_t\|^2} && \text{(Remark 12 and } i_t \perp\!\!\!\perp g_u, \forall u < t)\\
&\leq \frac{\sqrt{\Delta}}{2}(\mathbb{E}\|g_u\|^2 + \mathbb{E}\|g_t\|^2)\,. && \text{(AM-GM inequality)}
\end{aligned}$$

All told, we have:

$$\mathbb{E}|\langle g_u, g_t\rangle| \leq \frac{\sqrt{\Delta}}{2}(\mathbb{E}\|g_u\|^2 + \mathbb{E}\|g_t\|^2)\,. \tag{24}$$

15. One only has to switch u and t if $u > t$.

Lemma 13 (Suboptimality bound on $\mathbb{E}\|g_t\|^2$) *For all $t \geq 0$,*

$$\mathbb{E}\|g_t\|^2 \leq 4Le_t + \frac{4L}{n}\sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u + 4L(1-\frac{1}{n})^{(t-\tau)_+}\tilde{e}_0\,, \tag{25}$$

where $\tilde{e}_0 := \frac{1}{2L}\mathbb{E}\|\alpha_i^0 - f_i'(x^*)\|^2$.[16]

From our proof of convergence for Sparse Saga we know that (see Appendix A):

$$\mathbb{E}\|g_t\|^2 \leq 2\mathbb{E}\|f_{i_t}'(\hat{x}_t) - f_{i_t}'(x^*)\|^2 + 2\mathbb{E}\|\hat{\alpha}_{i_t}^t - f_{i_t}'(x^*)\|^2. \tag{26}$$

We can handle the first term by taking the expectation over a Lipschitz inequality (Hofmann et al. (2015, Equations 7 and 8). All that remains to prove the lemma is to express the $\mathbb{E}\|\hat{\alpha}_{i_t}^t - f_{i_t}'(x^*)\|^2$ term in terms of past suboptimalities. We note that it can be seen as an expectation of past first terms with an adequate probability distribution which we derive and bound.

From our algorithm, we know that each dimension of the memory vector $[\hat{\alpha}_i]_v$ contains a partial gradient computed at some point in the past $[f_i'(\hat{x}_{u_{i,v}^t})]_v$[17] (unless $u = 0$, in which case we replace the partial gradient with α_i^0). We then derive bounds on $P(u_{i,v}^t = u)$ and sum on all possible u. Together with clever conditioning, we obtain Lemma 13 (see Section B.3).

Master inequality. Let H_t be defined as $H_t := \sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u$. Then, by setting (25) into Lemma 10, we get (see Section B.5):

$$\begin{aligned} a_{t+1} \leq & (1-\frac{\gamma\mu}{2})a_t - 2\gamma e_t + 4L\gamma^2 C_1\big(e_t + (1-\frac{1}{n})^{(t-\tau)_+}\tilde{e}_0\big) + \frac{4L\gamma^2 C_1}{n}H_t \\ & + 4L\gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1}\big(e_u + (1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0\big) + \frac{4L\gamma^2 C_2}{n}\sum_{u=(t-\tau)_+}^{t-1} H_u\,. \end{aligned} \tag{27}$$

Lyapunov function and associated recursive inequality. We now have the beginning of a contraction with additional positive terms which all converge to 0 as we near the optimum, as well as our classical negative suboptimality term. This is not unusual in the variance reduction literature. One successful approach in the sequential case is then to define a Lyapunov function which encompasses all terms and is a true contraction (see Defazio et al., 2014; Hofmann et al., 2015). We emulate this solution here. However, while all terms in the sequential case only depend on the current iterate, t, in the parallel case we have terms "from the past" in our inequality. To resolve this issue, we define a more involved Lyapunov function which also encompasses past iterates:

$$\mathcal{L}_t = \sum_{u=0}^{t}(1-\rho)^{t-u}a_u, \quad 0 < \rho < 1, \tag{28}$$

where ρ is a target contraction rate that we define later.

16. We introduce this quantity instead of e_0 so as to be able to handle the arbitrary initialization of the α_i^0.
17. More precisely: $\forall t, i, v\ \exists u_{i,v}^t < t$ s.t. $[\hat{\alpha}_i^t]_v = [f_i'(\hat{x}_{u_{i,v}^t})]_v$.

Using the master inequality (27), we get (see Appendix B.6):

$$\begin{aligned} \mathcal{L}_{t+1} &= (1-\rho)^{t+1}a_0 + \sum_{u=0}^{t}(1-\rho)^{t-u}a_{u+1} \\ &\leq (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + \sum_{u=1}^{t} r_u^t e_u + r_0^t\tilde{e}_0\,. \end{aligned} \tag{29}$$

The aim is to prove that $\mathcal{L}_t$ is bounded by a contraction. We have two promising terms at the beginning of the inequality, and then we need to handle the last term. Basically, we can rearrange the sums in (27) to expose a simple sum of e_u multiplied by factors r_u^t.

Under specific conditions on ρ and γ, we can prove that r_u^t is negative for all $u \geq 1$, which coupled with the fact that each e_u is positive means that we can safely drop the sum term from the inequality. The r_0^t term is a bit trickier and is handled separately.

In order to obtain a bound on e_t directly rather than on $\mathbb{E}\|\hat{x}_t - x^*\|^2$, we then introduce an additional γe_t term on both sides of (29). The bound on γ under which the modified $r_t^t + \gamma$ is negative is then twice as small (we could have used any multiplier between 0 and 2γ, but chose γ for simplicity's sake). This condition is given in the following Lemma.

Lemma 14 (Sufficient condition for convergence) *Suppose $\tau < n/10$ and $\rho \leq 1/4n$. If*

$$\gamma \leq \gamma^* = \frac{1}{32L(1+\sqrt{\Delta}\tau)\sqrt{1+\frac{1}{8\kappa}\min(\tau, \frac{1}{\sqrt{\Delta}})}} \tag{30}$$

then for all $u \geq 1$, the coefficients r_u^t from (29) are negative. Furthermore, we have $r_t^t + \gamma \leq 0$ and thus:

$$\gamma e_t + \mathcal{L}_{t+1} \leq (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + r_0^t\tilde{e}_0\,. \tag{31}$$

We obtain this result after carefully deriving the r_u^t terms. We find a second-order polynomial inequality in γ, which we simplify down to (30) (see Appendix B.7).

We can then finish the argument to bound the suboptimality error e_t. We have:

$$\mathcal{L}_{t+1} \leq \gamma e_t + \mathcal{L}_{t+1} \leq (1-\frac{\gamma\mu}{2})\mathcal{L}_t + (1-\rho)^{t+1}(a_0 + A\tilde{e}_0)\,. \tag{32}$$

We have two linearly contracting terms. The sum contracts linearly with the worst rate between the two (the smallest geometric rate factor). If we define $\rho^* := \nu\min(\rho, \gamma\mu/2)$, with $0 < \nu < 1$,[18] then we get:

$$\gamma e_t + \mathcal{L}_{t+1} \leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (1-\rho^*)^{t+1}\frac{a_0 + A\tilde{e}_0}{1-\eta} \tag{33}$$

$$\gamma e_t \leq (1-\rho^*)^{t+1}\big(\mathcal{L}_0 + \frac{1}{1-\eta}(a_0 + A\tilde{e}_0)\big)\,, \tag{34}$$

where $\eta := \frac{1-M}{1-\rho^*}$ with $M := \max(\rho, \gamma\mu/2)$. Our geometric rate factor is thus ρ^* (see Appendix B.8).

18. ν is introduced to circumvent the problematic case where ρ and $\gamma\mu/2$ are too close together.

4. Asynchronous Parallel SVRG with the "After Read" Labeling

ASAGA *vs. asynchronous* SVRG. There are several scenarios in which ASAGA can be practically advantageous over its closely related cousin, asynchronous SVRG (note though that "asynchronous" SVRG still requires one synchronization step per epoch to compute a full gradient).

First, while SAGA trades memory for less computation, in the case of generalized linear models the memory cost can be reduced to $\mathcal{O}(n)$, compared to $\mathcal{O}(d)$ for SVRG (Johnson and Zhang, 2013). This is of course also true for their asynchronous counterparts.

Second, as ASAGA does not require any synchronization steps, it is better suited to heterogeneous computing environments (where cores have different clock speeds or are shared with other applications).

Finally, ASAGA does not require knowing the condition number κ for optimal convergence in the sparse regime. It is thus adaptive to local strong convexity, whereas SVRG is not. Indeed, SVRG and its asynchronous variant require setting an additional hyper-parameter – the epoch size m – which needs to be at least $\Omega(\kappa)$ for convergence but yields a slower effective convergence rate than ASAGA if it is set much bigger than κ. SVRG thus requires tuning this additional hyper-parameter or running the risk of either slower convergence (if the epoch size chosen is much bigger than the condition number) or even not converging at all (if m is chosen to be much smaller than κ).[19]

***Motivation for analyzing asynchronous* SVRG.** Despite the advantages that we have just listed, in the case of complex models, the storage cost of SAGA may become too expensive for practical use. SVRG (Johnson and Zhang, 2013) trades off more computation for less storage and does not suffer from this drawback. It can thus be applied to cases where SAGA cannot (e.g. deep learning models, see Reddi et al., 2016).

Another advantage of KROMAGNON is that the historical gradient term $f'(\tilde{x})$ is fixed during an epoch, while its ASAGA equivalent, $\bar{\alpha}$, has to be updated at each iteration, either by recomputing if from the $\hat{\alpha}$ – which is costly – or by updating a maintained quantity – which is cheaper but may ultimately result in introducing some bias in the update (see Appendix F.3 for more details on this subtle issue).

It is thus worthwhile to carry out the analysis of KROMAGNON (Mania et al., 2017)[20], the asynchronous parallel version of SVRG, although it has to be noted that since SVRG requires regularly computing batch gradients, KROMAGNON will present regular synchronization steps as well as coordinated computation – making it less attractive for the asynchronous parallel setting.

We first extend our ASAGA analysis to analyze the convergence of a variant of SVRG presented in Hofmann et al. (2015), obtaining exactly the same bounds. This variant improves upon the initial algorithm because it does not require tuning the epoch size hyperparameter and is thus adaptive to local strong convexity (see Section 4.1). Furthermore, it allows for a cleaner analysis where – contrary to SVRG – we do not have to replace the final parameters of an epoch by one of its random iterates.

Then, using our "after read" labeling, we are also able to derive a convergence and speedup proof for KROMAGNON, with comparable results to our ASAGA analysis. In particular, we prove that as for ASAGA in the "well-conditioned" regime KROMAGNON can achieve a linear speedup even without sparsity assumptions.

4.1. SVRG Algorithms

We start by describing the original SVRG algorithm, the variant given in Hofmann et al. (2015) and the sparse asynchronous parallel adaptation, KROMAGNON.

Original* SVRG *algorithm. The standard SVRG algorithm (Johnson and Zhang, 2013) is very similar to SAGA. The main difference is that instead of maintaining a table of historical gradients, SVRG uses a "reference" batch gradient $f'(\tilde{x})$, updated at regular intervals (typically every m iterations, where m is a hyper-parameter). SVRG is thus an epoch-based algorithm, where at the beginning of every epoch a reference iterate $\tilde{x}$ is chosen and its gradient is computed. Then, at every iteration in the epoch, the algorithm samples uniformly at random an index $i \in \{1, \dots, n\}$, and then executes the following update on x:

$$x^+ = x - \gamma\big(f_i'(x) - f_i'(\tilde{x}) + f'(\tilde{x})\big)\,. \tag{35}$$

As for SAGA the update direction is unbiased ($\mathbf{E}x^+ = x - \gamma f'(x)$) and it can be proven (see Johnson and Zhang, 2013) that under a reasonable condition on γ and m (the epoch size), the update has vanishing variance, which enables the algorithm to converge linearly with a constant step size.

Hofmann's* SVRG *variant. Hofmann et al. (2015) introduce a variant where the size of the epoch is a random variable. At each iteration t, a first Bernoulli random variable B_t with $p = 1/n$ is sampled. If $B_t = 1$, then the algorithm updates the reference iterate, $\tilde{x} = x_t$ and computes its full gradient as its new "reference gradient". If $B_t = 0$, the algorithm executes the normal SVRG inner update. Note that this variant is adaptive to local strong convexity, as it does not require the inner loop epoch size $m = \Omega(\kappa)$ as a hyperparameter. In that respect it is closer to SAGA than the original SVRG algorithm which is not adaptive.

KROMAGNON. KROMAGNON, introduced in Mania et al. (2017) is obtained by using the same sparse update technique as Sparse SAGA, and then running the resulting algorithm in parallel (see Algorithm 3).

4.2. Extension to the SVRG Variant from Hofmann et al. (2015)

We introduce AHSVRG – a sparse asynchronous parallel version for the SVRG variant from Hofmann et al. (2015) – in Algorithm 4. Every core runs stochastic updates independently as long as they are all sampling inner updates, and coordinate whenever one of them decides to do a batch gradient computation. The one difficulty of this approach is that each core needs to be able to communicate to every other core that they should stop doing inner updates and start computing a synchronized batch gradient instead.

To this end, we introduce a new shared variable, s, which represents the "state" of the computation. This variable is checked by each core c before each update. If $s = 1$, then

19. Note that as SAGA (and contrary to the original SVRG) the SVRG variant from Hofmann et al. (2015) does not require knowledge of κ and is thus adaptive to local strong convexity, which carries over to its asynchronous adaptation that we analyze in Section 4.2.

20. The speedup analysis presented in Mania et al. (2017) is not fully satisfactory as it does not achieve state-of-the-art convergence results for either SVRG or KROMAGNON. Furthermore, we are able to remove their uniform gradient bound assumption, which is inconsistent with strong convexity.

Algorithm 3 Kromagnon (Mania et al., 2017)

Initialize shared x and x_0
while True **do**
  Compute in parallel $g = f'(x_0)$ (synchronously)
  for $i = 1..m$ **do in parallel (asynchronously)**
    Sample i uniformly in $\{1, ..., n\}$
    Let S_i be f_i's support
    $[\hat{x}]_{S_i}$ = inconsistent read of x on S_i
    $[\delta x]_{S_i} = -\gamma([f_i'(\hat{x}_t) - f_i'(x_0)]_{S_i} + D_i[g]_{S_i})$
    for v **in** S_i **do**
      $[x]_v = [x]_v + [\delta x]_v$ // atomic
    end for
  end parallel loop
  $x_0 = x$
end while

Algorithm 4 Ahsvrg

Initialize shared x, s and x_0
while True **do**
  Compute in parallel $g = f'(x_0)$ (synchronously)
  $s = 0$
  while $s = 0$ **do in parallel (asynchronously)**
    Sample B with $p = 1/n$
    if $B = 1$ **then**
      $s = 1$
    else
      Sample i uniformly in $\{1, ..., n\}$
      Let S_i be f_i's support
      $[\hat{x}]_{S_i}$ = inconsistent read of x on S_i
      $[\delta x]_{S_i} = -\gamma([f_i'(\hat{x}_t) - f_i'(x_0)]_{S_i} + D_i[g]_{S_i})$
      for v **in** S_i **do**
        $[x]_v = [x]_v + [\delta x]_v$ // atomic
      end for
    end if
  end parallel loop
  $x_0 = x$
end while

another core has called for a batch gradient computation and core c starts computing its allocated part of this computation. If $s = 0$, core c proceeds to sample a first random variable. Then it either samples and performs an inner update and keeps going, or it samples a full gradient computation, in which case it updates s to 1 and starts computing its allocated

part of the computation. Once a full gradient is computed, s is set to 0 once again and every core resume their loop.

Our Asaga convergence and speedup proofs can easily be adapted to accommodate Ahsvrg since it is closer to Saga than the initial Svrg algorithm. To prove convergence, all one has to do is to modify Lemma 13 very slightly (the only difference is that the $(t - 2\tau - u - 1)_+$ exponent is replaced by $(t - \tau - u - 1)_+$ and the rest of the proof can be used as is). The justification for this small tweak is that the batch steps in Svrg are fully synchronized. More details can be found in Appendix B.4.

4.3. Fast Convergence and Speedup Rates for Kromagnon

We now state our main theoretical results. We give a detailed outline of the proof in Section 4.4 and its full details in Appendix C.

Theorem 15 (Convergence guarantee and rate of Kromagnon) *Suppose the step size γ and epoch size m are chosen such that the following condition holds:*

$$0 < \theta := \frac{\frac{1}{\mu\gamma m} + 2L(1 + 2\sqrt{\Delta}\tau)(\gamma + \tau\mu\gamma^2)}{1 - 2L(1 + 2\sqrt{\Delta}\tau)(\gamma + \tau\mu\gamma^2)} < 1\,. \tag{36}$$

Then the inconsistent read iterates of Kromagnon *converge in expectation at a geometric rate, i.e.*

$$\mathbb{E} f(\tilde{x}_k) - f(x^*) \leq \theta^t (f(x_0) - f(x^*))\,, \tag{37}$$

where $\tilde{x}_k$ is the initial iterate for epoch k, which is obtained by choosing uniformly at random among the inconsistent read iterates from the previous epoch.

This result is similar to the theorem given in the original Svrg paper (Johnson and Zhang, 2013). Indeed, if we remove the asynchronous part (i.e. if we set $\tau = 0$), we get exactly the same rate and condition. It also has the same form as the one given in Reddi et al. (2015), which was derived for dense asynchronous Svrg in the easier setting of consistent read and writes (and in the flawed "after write" framework), and gives essentially the same conditions on γ and m.

In the canonical example presented in most Svrg papers, with $\kappa = n$, $m = \mathcal{O}(n)$ and $\gamma = 1/10L$, Svrg obtains a convergence rate of 0.5. Reddi et al. (2015) get the same rate by setting $\gamma = 1/20\max(1,\sqrt{\Delta}\tau)L$ and $m = \mathcal{O}\big(n(1 + \sqrt{\Delta}\tau)\big)$. Following the same line of reasoning (setting $\gamma = 1/20\max(1,\sqrt{\Delta}\tau)L$, $\tau = \mathcal{O}(n)$ and $\theta = 0.5$ and computing the resulting condition on m), these values for γ and m also give us a convergence rate of 0.5. Therefore, as in Reddi et al. (2015), when $\kappa = n$ we get a linear speedup for $\tau < 1/\sqrt{\Delta}$ (which can be as big as $\sqrt{n}$ in the degenerate case where no data points share any feature with each other). Note that this is the same speedup condition as Asaga in this regime.

Svrg theorems are usually similar to Theorem 15, which does not give an optimal step size or epoch size. This makes the analysis of a parallel speedup difficult, prompting authors to compare rates in specific cases with most parameters fixed, as we have just done. In order to investigate the speedup and step size conditions more precisely and thus derive a more general theorem, we now give Svrg and Kromagnon results modeled on Theorem 8.

Corollary 16 (Convergence guarantee and rate for serial SVRG) *Let $\gamma = \frac{a}{4L}$ for any $a \leq \frac{1}{4}$ and $m = \frac{32\kappa}{a}$. Then* SVRG *converges geometrically in expectation with a rate factor per gradient computation of at least $\rho(a) = \frac{1}{4}\min\left\{\frac{1}{n}, \frac{a}{64\kappa}\right\}$, i.e.*

$$\mathbb{E}f(\tilde{x}_k) - f(x^*) \leq (1-\rho)^{k(2m+n)}\left(f(x_0) - f(x^*)\right) \qquad \forall k \geq 0\,. \tag{38}$$

Due to SVRG's special structure, we cannot write $\mathbb{E}f(x_t) - f(x^*) \leq (1-\rho)^t\left(f(x_0) - f(x^*)\right)$ for all $t \geq 0$. However, expressing the convergence properties of this algorithm in terms of a rate factor per gradient computation (of which there are $2m+n$ per epoch) makes it easier to compare convergence rates, either to similar algorithms such as SAGA or to its parallel variant KROMAGNON – and thus to study the speedup obtained by parallelizing SVRG.

Compared to SAGA, this result is very close. The main difference is that the additional hyper-parameter m has to be set and requires knowledge of μ. This illustrates the fact that SVRG is not adaptive to local strong convexity, whereas both SAGA and Hofmann's SVRG are.

Corollary 17 (Simplified convergence guarantee and rate for KROMAGNON) *Let*

$$a^*(\tau) = \frac{1}{4(1+2\sqrt{\Delta}\tau)(1+\frac{\tau}{16\kappa})}\,. \tag{39}$$

For any step size $\gamma = \frac{a}{4L}$ with $a \leq a^*(\tau)$ and $m = \frac{32\kappa}{a}$, KROMAGNON *converges geometrically in expectation with a rate factor per gradient computation of at least $\rho(a) = \frac{1}{4}\min\left\{\frac{1}{n}, \frac{a}{64\kappa}\right\}$, i.e.*

$$\mathbb{E}f(\tilde{x}_k) - f(x^*) \leq (1-\rho)^{k(2m+n)}\left(f(x_0) - f(x^*)\right) \qquad \forall k \geq 0\,. \tag{40}$$

This result is again quite close to Corollary 16 derived in the serial case. We see that the maximum step size is divided by an additional $(1+2\tau\sqrt{\Delta})$ term, while the convergence rate is the same. Comparing the rates and the maximum allowable step sizes in both settings give us the sufficient condition on τ to get a linear speedup.

Corollary 18 (Speedup condition) *Suppose $\tau \leq \mathcal{O}(n)$ and $\tau \leq \mathcal{O}(\frac{1}{\sqrt{\Delta}}\max\{1, \frac{n}{\kappa}\})$. If $n \geq \kappa$, also suppose $\tau \leq \sqrt{n\Delta^{-1/2}}$. Then using the step size $\gamma = a^*(\tau)/L$ from (39),* KROMAGNON *converges geometrically with rate factor $\Omega(\min\{\frac{1}{n}, \frac{1}{\kappa}\})$ (similar to* SVRG*), and is thus linearly faster than its sequential counterpart up to a constant factor.*

This result is almost the same as ASAGA, with the additional condition that $\tau \leq \mathcal{O}(\sqrt{n})$ in the well-conditioned regime. We see that in this regime KROMAGNON can also get the same rate as SVRG even without sparsity, which had not been observed in previous work.

Furthermore, one has to note that τ is generally smaller for KROMAGNON than for ASAGA since it is reset to 0 at the beginning of each new epoch (where all cores are synchronized once more).

Comparison to related work.

- Corollary 16 provides a rate of convergence *per gradient computation* for SVRG, contrary to most of the literature on this algorithm (including the seminal paper Johnson and Zhang, 2013). This result allows for easy comparison with SAGA and other algorithms (in contrast, Konečný and Richtárik 2013 is more involved).
- In contrast to the SVRG analysis from Reddi et al. (2015, Thm. 2), our proof technique handles inconsistent reads and a non-uniform processing speed across f_i's. While Theorem 15 is similar to theirs, Corollary 16 and 17 are more precise results. They enable a finer analysis of the speedup conditions (Corollary 18) – including the possible speedup without sparsity regime.
- In contrast to the KROMAGNON analysis from Mania et al. (2017, Thm. 14), Theorem 15 gives a better dependence on the condition number in the rate ($1/\kappa$ vs. $1/\kappa^2$ for them) and on the sparsity (they get $\tau \leq \mathcal{O}(\Delta^{-1/3})$), while we remove their gradient bound assumption. Our results are state-of-the-art for SVRG (contrary to theirs) and so our speedup comparison is more meaningful. Finally, Theorem 15 gives convergence guarantees on $\hat{x}_t$ *during* the algorithm, whereas they only bound the error for the "last" iterate x_T.

4.4. Proof of Theorem 15

We now give a detailed outline of the proof. Its full details can be found in Appendix C.

Our proof technique begins as our ASAGA analysis. In particular, Properties 3, 4, 5 are also verified for KROMAGNON[21], and as in our ASAGA analysis, we make Assumption 6 (bounded overlaps). Consequently, the basic recursive contraction inequality (15) and Lemma 10 also hold. However, when we derive the equivalent of Lemma 13, we get a slightly different form, which prompts a difference in the rest of the proof technique.

Lemma 19 (Suboptimality bound on $\mathbb{E}\|g_t\|^2$)

$$\mathbb{E}\|g_t\|^2 \leq 4Le_t + 4L\tilde{e}_k \qquad \forall k \geq 0, km \leq t \leq (k+1)m\,, \tag{41}$$

where $\tilde{e}_k := \mathbb{E}f(\tilde{x}_k) - f(x^*)$ and $\tilde{x}_k$ is the initial iterate for epoch k.

We give the proof in Appendix C.1. To derive both terms, we use the same technique as for the first term of Lemma 13. Although this is a much simpler result than Lemma 13 in the case of ASAGA, two key differences prevent us from reusing the same Lyapunov function proof technique. First, the e_0 term in Lemma 13 is replaced by $\tilde{e}_k$ which depends on the epoch number. Second, this term is not multiplied by a geometrically decreasing quantity, which means the $-2\gamma e_0$ term is not sufficient to cancel out all of the e_0 terms coming from subsequent inequalities. To solve this issue, we go to more traditional SVRG techniques.

The rest of the proof is as follows:

1. By substituting Lemma 19 into Lemma 10, we get a master contraction inequality (42) in terms of a_{t+1}, a_t and $e_u, u \leq t$.

21. Note that similarly to ASAGA, the KROMAGNON algorithm which we analyze reads the parameters first and then samples. This is necessary in order for Property 3 to be verified at $r = t$, although not practical when it comes to actual implementation.

2. As in Johnson and Zhang (2013), we sum the master contraction inequality over a whole epoch, and then use the same randomization trick (44) to relate $(e_t)_{km \leq t \leq (k+1)m-1}$ to $\tilde{e}_k$.

3. We thus obtain a contraction inequality between $\tilde{e}_k$ and $\tilde{e}_{k-1}$, which finishes the proof for Theorem 15.

4. We then only have to derive the conditions on γ, τ and m under which we contractions and compare convergence rates to finish the proofs for Corollary 16, Corollary 17 and Corollary 18.

We list the key points below with their proof sketch, and give the detailed proof in Appendix C.

Master inequality. As in our Asaga analysis, we apply (41) to the result of Lemma 10, which gives us that for all $k \geq 0, km \leq t \leq (k+1)m-1$ (see Appendix C.2):

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + (4L\gamma^2 C_1 - 2\gamma)e_t + 4L\gamma^2 C_2 \sum_{u=\max(km, t-\tau)}^{t-1} e_u + (4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,. \tag{42}$$

Contraction inequality. As we previously mentioned, the term in $\tilde{e}_k$ is not multiplied by a geometrically decreasing factor, so using the same Lyapunov function as for Asaga cannot work. Instead, we apply the same method as in the original Svrg paper (Johnson and Zhang, 2013): we sum the master contraction inequality over a whole epoch. This gives us (see Appendix C.2):

$$a_{(k+1)m} \leq a_{km} + (4L\gamma^2 C_1 + 4L\gamma^2\tau C_2 - 2\gamma) \sum_{t=km}^{(k+1)m-1} e_t + m(4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,. \tag{43}$$

To cancel out the $\tilde{e}_k$ term, we only have one negative term on the right-hand side of (43): $-2\gamma \sum_{t=km}^{(k+1)m-1} e_t$. This means we need to relate $\sum_{t=km}^{(k+1)m-1} e_t$ to $\tilde{e}_k$. We can do it using the same randomization trick as in Johnson and Zhang (2013): instead of choosing the last iterate of the k^{th} epoch as $\tilde{x}_k$, we pick one of the iterates of the epoch uniformly at random. This means we get:

$$\tilde{e}_k = \mathbb{E} f(\tilde{x}_k) - f(x^*) = \frac{1}{m} \sum_{t=(k-1)m}^{km-1} e_t \tag{44}$$

We now have: $\sum_{t=km}^{(k+1)m-1} e_t = m\tilde{e}_{k+1}$. Combined with the fact that $a_{km} \leq \frac{2}{\mu}\tilde{e}_k$ and that we can remove the positive $a_{(k+1)m}$ term from the left-hand-side of (43), this gives us our final recursion inequality:

$$\big(2\gamma m - 4L\gamma^2 C_1 m - 4L\gamma^2\tau C_2 m\big)\tilde{e}_{k+1} \leq \big(\frac{2}{\mu} + 4L\gamma^2 C_1 m + 4L\gamma^2\tau C_2 m\big)\tilde{e}_k \tag{45}$$

Replacing C_1 and C_2 by their values (defined in 17) in (45) directly leads to Theorem 15.

Algorithm 5 Hogwild

```
Initialize shared variable x
keep doing in parallel
    x̂ = inconsistent read of x
    Sample i uniformly in {1, ..., n}
    Let S_i be f_i's support
    [δx]_{S_i} := −γ f_i'(x̂)
    for v in S_i do
        [x]_v ← [x]_v + [δx]_v        // atomic
    end for
end parallel loop
```

5. Hogwild Analysis

In order to show that our improved "after read" perturbed iterate framework can be used to revisit the analysis of other optimization routines with correct proofs that do not assume homogeneous computation, we now provide the analysis of the Hogwild algorithm (i.e. asynchronous parallel constant step size Sgd) first introduced in Niu et al. (2011).

We start by describing Hogwild in Algorithm 5, and then give our theoretical convergence and speedup results and their proofs. Note that our framework allows us to easily remove the classical bounded gradient assumption, which is used in one form or another in most of the literature (Niu et al., 2011; De Sa et al., 2015; Mania et al., 2017) – although it is inconsistent with strong convexity in the unconstrained regime. This allows for better bounds where the uniform bound on $\|f_i'(x)\|^2$ is replaced by its variance at the optimum.

5.1. Theoretical Results

We now state the theoretical results of our analysis of Hogwild with inconsistent reads and writes in the "after read framework". We give an outline of the proof in Section 5.2 and its full details in Appendix D. We start with a useful definition.

Definition 20 *Let $\sigma^2 = \mathbf{E}\|f_i'(x^*)\|^2$ be the variance of the gradient estimator at the optimum.*

For reference, we start by giving the rate of convergence of serial Sgd (see e.g. Schmidt, 2014).

Theorem 21 (Convergence guarantee and rate of Sgd) *Let $a \leq \frac{1}{2}$. Then for any step size $\gamma = \frac{a}{L}$, Sgd converges in expectation to b-accuracy at a geometric rate of at least: $\rho(a) = a/\kappa$, i.e., $\mathbb{E}\|x_t - x^*\|^2 \leq (1-\rho)^t \|x_0 - x^*\|^2 + b$, where $b = 2\frac{\gamma\sigma^2}{\mu}$.*

As Sgd only converges linearly up to a ball around the optimum, to make sure we reach ϵ-accuracy, it is necessary that $\frac{2\gamma\sigma^2}{\mu} \leq \epsilon$, i.e. $\gamma \leq \frac{\epsilon\mu}{2\sigma^2}$. All told, in order to get linear convergence to ϵ-accuracy, serial Sgd requires $\gamma \leq \min\big\{\frac{1}{2L}, \frac{\epsilon\mu}{2\sigma^2}\big\}$. The proof can be found in Appendix D.3.

Theorem 22 (Convergence guarantee and rate of HOGWILD) *Let*

$$a^*(\tau) := \frac{1}{5\left(1+2\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)} \quad \text{where } \xi(\kappa,\Delta,\tau) := \sqrt{1+\frac{1}{2\kappa}\min\{\frac{1}{\sqrt{\Delta}},\tau\}} \quad \text{(note that } \xi(\kappa,\Delta,\tau)\approx 1 \text{ unless } \kappa < 1/\sqrt{\Delta}\ (\leq\sqrt{n})\text{).} \tag{46}$$

For any step size $\gamma = \frac{a}{L}$ with $a \leq \min\left\{a^*(\tau), \frac{\kappa}{\tau}\right\}$, the inconsistent read iterates of Algorithm 5 converge in expectation to b-accuracy at a geometric rate of at least: $\rho(a) = a/\kappa$, i.e., $\mathbb{E}\|\hat{x}_t - x^*\|^2 \leq (1-\rho)^t(2\|x_0 - x^*\|^2) + b$, where $b = (\frac{8\gamma(C_1+\tau C_2)}{\mu} + 4\gamma^2 C_1\tau)\sigma^2$ and C_1 and $C_2(\gamma)$ are defined in (17).

Once again this result is quite close to the one obtained for serial SGD. Note that we recover this exact condition (up to a small constant factor) if we set $\tau = 0$, i.e. if we force our asynchronous algorithm to be serial.

The condition $a \leq \frac{\kappa}{\tau}$ is equivalent to $\gamma\mu\tau \leq 1$ and should be thought of as a condition on τ. We will see that it is always verified in the regime we are interested in, that is the linear speed-up regime (where more stringent conditions are imposed on τ).

We now investigate the conditions under which HOGWILD is linearly faster than SGD. Note that to derive these conditions we need not only compare their respective convergence rates, but also the size of the ball around the optimum to which both algorithms converge. These quantities are provided in Theorems 21 and 22.

Corollary 23 (Speedup condition) *Suppose $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}},\kappa\})$. Then for any step size $\gamma \leq \frac{a^*(\tau)}{L} = \mathcal{O}(\frac{1}{L})$ (i.e., any allowable step size for* SGD*),* HOGWILD *converges geometrically to a ball of radius $r_h = \mathcal{O}(\frac{\gamma\sigma^2}{\mu})$ with rate factor $\rho = \frac{\gamma\mu}{2}$ (similar to* SGD*), and is thus linearly faster than its sequential counterpart up to a constant factor.*

Moreover, a universal step size of $\Theta(\frac{1}{L})$ can be used for HOGWILD *to be adaptive to local strong convexity with a similar rate to* SGD *(i.e., knowledge of κ is not required).*

If $\gamma = \mathcal{O}(1/L)$, HOGWILD obtains the same convergence rate as SGD and converges to a ball of equivalent radius. Since the maximum step size guaranteeing linear convergence for SGD is also $\mathcal{O}(1/L)$, HOGWILD is linearly faster than SGD for any reasonable step size – under the condition that $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}},\kappa\})$. We also remark that since $\gamma \leq 1/L$ and $\tau \leq \kappa$, we have $\gamma\mu\tau \leq 1$, which means the condition $a \leq \frac{\kappa}{\tau}$ is superseded by $a \leq a^*(\tau)$ in Theorem 22.

We note that the condition on τ is much more restrictive if the condition number is small than for ASAGA and KROMAGNON. This can be explained by the fact that both SAGA and SVRG have a composite rate factor which is not directly proportional to the step size. As a result, in the well-conditioned setting these algorithms enjoy a range of step sizes that all give the same contraction rate. This allows their asynchronous variants to use smaller step sizes while maintaining linear speedups. SGD, on the other hand, has a rate factor that is directly proportional to its step size, hence the more restrictive condition on τ.

Function values results. Our results are derived directly on iterates, that is, we bound the distance between $\hat{x}_t$ and x^*. We can easily obtain results on function values to

bound $\mathbb{E}f(\hat{x}_t) - f(x^*)$ by adapting the classical smoothness inequality:[22] $\mathbb{E}f(x_t) - f(x^*) \leq \frac{L}{2}\mathbb{E}\|x_t - x^*\|^2$ to the asynchronous parallel setting.

Convergence to ϵ-accuracy. As noted in Mania et al. (2017), for our algorithm to converge to ϵ-accuracy for some $\epsilon > 0$, we require an additional bound on the step size to make sure that the radius of the ball to which we converge is small enough. For SGD, this means using a step size $\gamma = \mathcal{O}(\frac{\epsilon\mu}{\sigma^2})$ (see Appendix D.3). We can also prove that under the conditions that $\tau = \mathcal{O}(\frac{1}{\sqrt{\Delta}})$ and $\gamma\mu\tau \leq 1$, HOGWILD requires the same bound on the step size to converge to ϵ-accuracy (see Appendix D.4).

If ϵ is small enough, the active upper bound on the step size is $\gamma = \mathcal{O}(\frac{\epsilon\mu}{\sigma^2})$ for both algorithms. In this regime, we obtain a relaxed condition on τ for a linear speedup. The condition $\tau \leq \kappa$ which came from comparing maximum allowable step sizes is removed. Instead, we enforce $\gamma\mu\tau \leq 1$, which gives us the weaker condition $\tau = \mathcal{O}(\frac{\sigma^2}{\epsilon\mu^2})$. Our condition on the overlap is then: $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}}, \frac{\sigma^2}{\epsilon\mu^2}\})$. We see that this is similar to the condition obtained by Mania et al. (2017, Theorem 4) in their HOGWILD analysis, although we have the variance at the optimum σ^2 instead of a squared global bound on the gradient.

Comparison to related work.

- We give the first convergence analysis for HOGWILD with no assumption on a global bound on the gradient (M). This allows us to replace the usual dependence in M^2 by a term in σ^2 which is potentially significantly smaller. This means improved upper bounds on the step size and the allowed overlap.
- We obtain the same condition on the step size for linear convergence to ϵ-accuracy of HOGWILD as previous analysis for serial SGD (e.g. Needell et al., 2014) – given $\tau \leq 1/\gamma\mu$.
- In contrast to the HOGWILD analysis from Niu et al. (2011); De Sa et al. (2015), our proof technique handles inconsistent reads and a non-uniform processing speed across f_i's. Further, Corollary 23 gives a better dependence on the sparsity than in Niu et al. (2011), where $\tau \leq \mathcal{O}(\Delta^{-1/4})$, and does not require various bounds on the gradient assumptions.
- In contrast to the HOGWILD analysis from Mania et al. (2017, Thm. 3), removing their gradient bound assumption enables us to get a (potentially) significantly better upper bound condition on τ for a linear speedup. We also give our convergence guarantee on $\hat{x}_t$ *during* the algorithm, whereas they only bound the error for the "last" iterate x_T.

5.2. Proof of Theorem 22 and Corollary 23

Here again, our proof technique begins as our ASAGA analysis, with Properties 3, 4, 5 also verified for HOGWILD[23]. As in our ASAGA analysis, we make Assumption 6. Consequently, the basic recursive contraction inequality (15) and Lemma 10 also hold. As for KROMAGNON, the proof diverges when we derive the equivalent of Lemma 13.

22. See e.g. Moulines and Bach (2011).
23. Once again, in our analysis the HOGWILD algorithms reads the parameters before sampling, so that Property 3 is verified for $r = t$.

Lemma 24 (Suboptimality bound on $\mathbb{E}\|g_t\|^2$) *For all $t \geq 0$,*

$$\mathbb{E}\|g_t\|^2 \leq 4Le_t + 2\sigma^2\,. \tag{47}$$

We give the proof in Appendix D.1. To derive both terms, we use the same technique as for the first term of Lemma 13. This result is simpler than both Lemma 13 (for Asaga) and Lemma 19 (for Kromagnon). The second term in this case does not even vanish as t grows. This reflects the fact that constant step size Sgd does not converge to the optimum but rather to a ball around it. However, this simpler form allows us to simply unroll the resulting master inequality to get our convergence result.

The rest of the proof is as follows:

1. By substituting Lemma 24 into Lemma 10, we get a master contraction inequality (48) in terms of a_{t+1}, a_t, $(e_u, u \leq t)$ and σ^2.

2. We then unroll this master inequality and cleverly regroup terms to obtain a contraction inequality (49) between a_t, a_0 and σ^2.

3. By using that $\|\hat{x}_t - x^*\|^2 \leq 2a_t + 2\|\hat{x}_t - x_t\|^2$, we obtain a contraction inequality directly on the "real" iterates (as opposed to the "virtual" iterates as in Mania et al., 2017), subject to a maximum step size condition on γ. This finishes the proof for Theorem 22.

4. Finally, we only have to derive the conditions on γ and τ under which Hogwild converges with a similar convergence rate to a ball with a similar radius than serial Sgd to finish the proof for Corollary 23.

We list the key points below with their proof sketch, and give the detailed proof in Appendix D.

Master inequality. As in our Asaga analysis, we plug (47) in Lemma 10, which gives us that (see Appendix D.2):

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + (4L\gamma^2 C_1 - 2\gamma)e_t + 4L\gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} e_u + 2\gamma^2\sigma^2(C_1 + \tau C_2)\,. \tag{48}$$

Contraction inequality on x_t. As we previously mentioned, the term in σ^2 does not vanish so we cannot use either our Asaga or our Kromagnon proof technique. Instead, we unroll Equation (48) all the way to $t = 0$. This gives us (see Appendix D.2):

$$a_{t+1} \leq (1-\frac{\gamma\mu}{2})^{t+1}a_0 + (4L\gamma^2 C_1 + 8L\gamma^2\tau C_2 - 2\gamma)\sum_{u=0}^{t}(1-\frac{\gamma\mu}{2})^{t-u}e_u + \frac{4\gamma\sigma^2}{\mu}(C_1+\tau C_2)\,. \tag{49}$$

Contraction inequality on $\hat{x}_t$. We now use that $\|\hat{x}_t - x^*\|^2 \leq 2a_t + 2\|\hat{x}_t - x_t\|^2$ together with our previous bound (65). Together with (49), we get (see Appendix D.2):

$$\mathbb{E}\|\hat{x}_t - x^*\|^2 \leq (1 - \frac{\gamma\mu}{2})^{t+1}2a_0 + \Big(\frac{8\gamma(C_1+\tau C_2)}{\mu} + 4\gamma^2 C_1\tau\Big)\sigma^2 + (24L\gamma^2 C_1 + 16L\gamma^2\tau C_2 - 4\gamma)\sum_{u=0}^{t}(1-\frac{\gamma\mu}{2})^{t-u}e_u\,. \tag{50}$$

To get our final contraction inequality, we need to safely remove all the e_u terms, so we enforce $16L\gamma^2 C_1 + 16L\gamma^2\tau C_2 - 4\gamma \leq 0$. This leads directly to Theorem 22.

Convergence rate and ball-size comparison. To prove Corollary 23, we simply show that under the condition $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}}, \kappa\})$, the biggest allowable step size for Hogwild to converge linearly is $\mathcal{O}(1/L)$, as is also the case for Sgd; and that the size of the ball to which both algorithms converge is of the same order. The proof is finished by remarking that for both algorithms, the rates of convergence are directly proportional to the step size.

6. Empirical Results

We now present the results of our experiments. We first compare our new sequential algorithm, Sparse Saga, to its existing alternative, Saga with lagged updates and to the original Saga algorithm as a baseline. We then move on to our main results, the empirical comparison of Asaga, Kromagnon and Hogwild. Finally, we present additional results, including convergence and speedup figures with respect to the number of iteration (i.e. "theoretical speedups") and measures on the τ constant.

6.1. Experimental Setup

Models. Although Asaga can be applied more broadly, we focus on logistic regression, a model of particular practical importance. The associated objective function takes the following form:

$$\frac{1}{n}\sum_{i=1}^{n}\log\left(1 + \exp(-b_i a_i^\intercal x)\right) + \frac{\mu}{2}\|x\|^2, \tag{51}$$

where $a_i \in \mathbb{R}^d$ and $b_i \in \{-1, +1\}$ are the data samples.

Data sets. We consider two sparse data sets: RCV1 (Lewis et al., 2004) and URL (Ma et al., 2009); and a dense one, Covtype (Collobert et al., 2002), with statistics listed in the table below. As in Le Roux et al. (2012), Covtype is standardized, thus 100% dense. Δ is $\mathcal{O}(1)$ in all data sets, hence not very insightful when relating it to our theoretical results. Deriving a less coarse sparsity bound remains an open problem.

Hardware and software. Experiments were run on a 40-core machine with 384GB of memory. All algorithms were implemented in Scala. We chose this high-level language despite its typical 20x slowdown compared to C (when using standard libraries, see Appendix F.2) because our primary concern was that the code may easily be reused and extended for research purposes (to this end, we have made all our code available at `http://www.di.ens.fr/sierra/research/asaga/`).

Table 1: Basic data set statistics.

	n	d	density	L
RCV1	697,641	47,236	0.15%	0.25
URL	2,396,130	3,231,961	0.004%	128.4
Covtype	581,012	54	100%	48428

6.2. Implementation Details

Regularization. Following Schmidt et al. (2016), the amount of regularization used was set to $\mu = 1/n$. In each update, we project the gradient of the regularization term (we multiply it by D_i as we also do with the vector $\bar{\alpha}$) to preserve the sparsity pattern while maintaining an unbiased estimate of the gradient. For squared ℓ_2, the Sparse Saga updates becomes:

$$x^+ = x - \gamma(f_i'(x) - \alpha_i + D_i\bar{\alpha} + \mu D_i x). \tag{52}$$

Comparison with the theoretical algorithm. The algorithm we used in the experiments is fully detailed in Algorithm 2. There are two differences with Algorithm 1. First, in the implementation we choose i_t at random *before* we read the feature vector a_{i_t}. This enables us to only read the necessary data for a given iteration (i.e. $[\hat{x}_t]_{S_i}, [\hat{\alpha}_i^t], [\bar{\alpha}^t]_{S_i}$). Although this violates Property 3, it still performs well in practice.

Second, we maintain $\bar{\alpha}^t$ in memory. This saves the cost of recomputing it at every iteration (which we can no longer do since we only read a subset data). Again, in practice the implemented algorithm enjoys good performance. But this design choice raises a subtle point: the update is not guaranteed to be unbiased in this setup (see Appendix F.3 for more details).

Step sizes. For each algorithm, we picked the best step size among 10 equally spaced values in a grid, and made sure that the best step size was never at the boundary of this interval. For Covtype and RCV1, we used the interval $[\frac{1}{10L}, \frac{10}{L}]$, whereas for URL we used the interval $[\frac{1}{L}, \frac{100}{L}]$ as it admitted larger step sizes. It turns out that the best step size was fairly constant for different number of cores for both Asaga and Kromagnon, and both algorithms had similar best step sizes (0.7 for RCV1, 0.05 for URL and 5×10^{-5} for Covtype).

6.3. Comparison of Sequential Algorithms: Sparse Saga vs Lagged updates

We compare the Sparse Saga variant proposed in Section 3.1 to two other approaches: the naive (i.e., dense) update scheme and the lagged updates implementation described in Defazio et al. (2014). Note that we use different datasets from the parallel experiments, including a subset of the RCV1 data set and the Realsim data set (see description in Appendix F.1). Figure 2 reveals that sparse and lagged updates have a lower cost per iteration than their dense counterpart, resulting in faster convergence for sparse data sets. Furthermore, while the two approaches had similar convergence in terms of number of iterations, the Sparse Saga scheme is slightly faster in terms of runtime (and as previously pointed out, sparse updates are better adapted for the asynchronous setting). For the dense data set (Covtype), the three approaches exhibit similar performance.

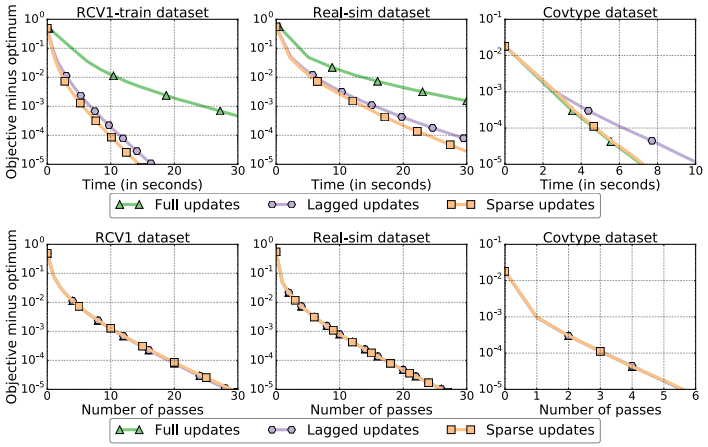
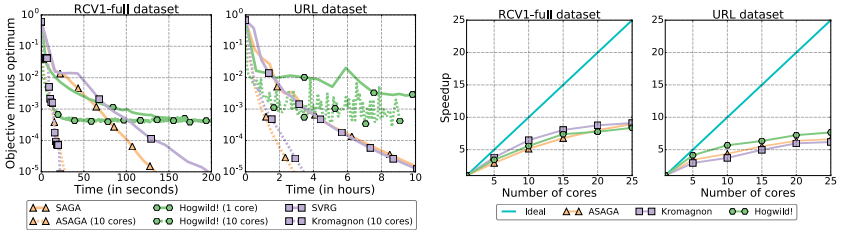


Figure 2: **Lagged vs Sparse Saga updates**. Suboptimality with respect to time for different Saga update schemes on various data sets. First row: suboptimality as a function of time. Second row: suboptimality as a the number of passes over the data set. For sparse data sets (RCV1 and Real-sim), lagged and sparse updates have a lower cost per iteration which result in faster convergence.



(a) Suboptimality as a function of time. (b) Speedup as a function of the number of cores

Figure 3: **Convergence and speedup for asynchronous stochastic gradient descent methods**. We display results for RCV1 and URL. Results for Covtype can be found in Section 6.6.

6.4. Asaga vs. Kromagnon vs. Hogwild

We compare three different asynchronous variants of stochastic gradient methods on the aforementioned data sets: Asaga, presented in this work, Kromagnon, the asynchronous sparse Svrg method described in Mania et al. (2017) and Hogwild (Niu et al., 2011). Each method had its step size chosen so as to give the fastest convergence (up to a suboptimality of 10^{-3} in the special case of Hogwild). The results can be seen in Figure 3a: for each method we consider its asynchronous version with both one (hence sequential) and ten processors. This figure reveals that the asynchronous version offers a significant speedup over its sequential counterpart.

We then examine the speedup relative to the increase in the number of cores. The speedup is measured as time to achieve a suboptimality of 10^{-5} (10^{-3} for Hogwild) with one core divided by time to achieve the same suboptimality with several cores, averaged over 3 runs. Again, we choose step size leading to fastest convergence[24] (see Appendix F.2 for information about the step sizes). Results are displayed in Figure 3b.

As predicted by our theory, we observe linear "theoretical" speedups (i.e. in terms of number of iterations, see Section 6.6). However, with respect to running time, the speedups seem to taper off after 20 cores. This phenomenon can be explained by the fact that our hardware model is by necessity a simplification of reality. As noted in Duchi et al. (2015), in a modern machine there is no such thing as *shared memory*. Each core has its own levels of cache (L1, L2, L3) in addition to RAM. These faster pools of memory are fully leveraged when using a single core. Unfortunately, as soon as several cores start writing to common locations, cache coherency protocols have to be deployed to ensure that the information is consistent across cores. These protocols come with computational overheads. As more and more cores are used, the shared information goes lower and lower in the memory stack, and the overheads get more and more costly. It may be the case that on much bigger data sets, where the cache memory is unlikely to provide benefits even for a single core (since

24. Although we performed grid search on a large interval, we observed that the best step size was fairly constant for different number of cores, and similar for Asaga and Kromagnon.

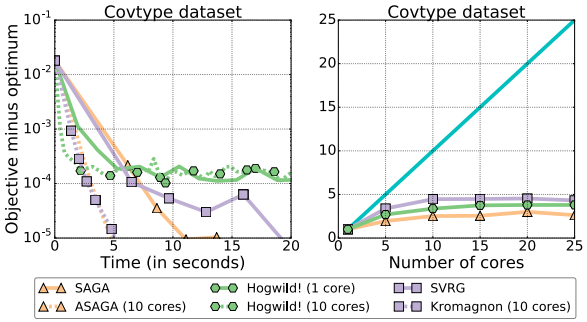


Figure 4: Comparison on the Covtype data set. Left: suboptimality. Right: speedup. The number of cores in the legend only refers to the left plot.

sampling items repeatedly becomes rare), the running time speedups actually improve. More experimentation is needed to quantify these effects and potentially increase performance.

6.5. Effect of Sparsity

Sparsity plays an important role in our theoretical results, where we find that while it is necessary in the "ill-conditioned" regime to get linear speedups, it is not in the "well-conditioned" regime. We confront this to real-life experiments by comparing the convergence and speedup performance of our three asynchronous algorithms on the Covtype data set, which is fully dense after standardization. The results appear in Figure 4.

While we still see a significant improvement in speed when increasing the number of cores, this improvement is smaller than the one we observe for sparser data sets. The speedups we observe are consequently smaller, and taper off earlier than on our other data sets. However, since the observed "theoretical" speedup is linear (see Section 6.6), we can attribute this worse performance to higher hardware overhead. This is expected because each update is fully dense and thus the shared parameters are much more heavily contended for than in our sparse datasets.

One thing we notice when computing the Δ constant for our data sets is that it often fails to capture the full sparsity distribution, being essentially a maximum: for all three data sets, we obtain $\Delta = \mathcal{O}(1)$. This means that Δ can be quite big even for very sparse data sets. Deriving a less coarse bound remains an open problem.

6.6. Theoretical Speedups

In the previous experimental sections, we have shown experimental speedup results where suboptimality was a function of the running time. This measure encompasses both theoretical algorithmic properties and hardware overheads (such as contention of shared memory) which are not taken into account in our analysis.

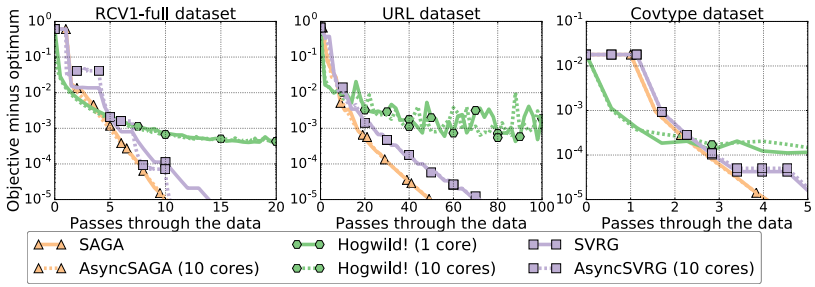


Figure 5: **Theoretical speedups**. Suboptimality with respect to number of iterations for Asaga, Kromagnon and Hogwild with 1 and 10 cores. Curves almost coincide, which means the theoretical speedup is almost the number of cores p, hence linear.

In order to isolate these two effects, we now plot our convergence experiments where suboptimality is a function of the number of iterations; thus, we abstract away any potential hardware overhead.[25] The experimental results can be seen in Figure 5.

For all three algorithms and all three data sets, the curves for 1 and 10 cores almost coincide, which means that we are indeed in the "theoretical linear speedup" regime. Indeed, when we plotted the amount of iterations required to converge to a given accuracy as a function of the number of cores, we obtained straight horizontal lines for our three algorithms.

The fact that the speedups we observe in running time are less than linear can thus be attributed to various hardware overheads, including shared variable contention – the compare-and-swap operations are more and more expensive as the number of competing requests increases – and cache effects as mentioned in Section 6.4.

6.7. A Closer Look at the τ Constant

6.7.1. Theory

In the parallel optimization literature, τ is often referred to as a proxy for the number of cores. However, intuitively as well as in practice, it appears that there are a number of other factors that can influence this quantity. We will now attempt to give a few qualitative arguments as to what these other factors might be and how they relate to τ.

Number of cores. The first of these factors is indeed the number of cores. If we have p cores, $\tau \geq p-1$. Indeed, in the best-case scenario where all cores have exactly the same execution speed for a single iteration, $\tau = p-1$.

Length of an iteration. To get more insight into what τ really encompasses, let us now try to define the worst-case scenario in the preceding example. Consider 2 cores. In the worst

25. To do so, we implement a global counter which is sparsely updated (every 100 iterations for example) in order not to modify the asynchrony of the system. This counter is used only for plotting purposes and is not needed otherwise.

case, one core runs while the other is stuck. Then the overlap is t for all t and eventually grows to $+\infty$. If we assume that one core runs twice as fast as the other, then $\tau = 2$. If both run at the same speed, $\tau = 1$.

It appears then that a relevant quantity is R, the ratio between the fastest execution time and the slowest execution time for a single iteration. We have $\tau \leq (p-1)R$, which can be arbitrarily bigger than p.

There are several factors at play in R itself. These include:

- the speed of execution of the cores themselves (i.e. clock time).
- the data matrix itself. Different support sizes for f_i means different gradient computation times. If one f_i has support of size n while all the others have support of size 1 for example, R may eventually become very big.
- the length of the computation itself. The longer our algorithm runs, the more likely it is to explore the potential corner cases of the data matrix.

The overlap is then upper bounded by the number of cores multiplied by the ratio of the maximum iteration time over the minimum iteration time (which is linked to the sparsity distribution of the data matrix). This is an upper bound, which means that in some cases it will not really be useful. For example, if one factor has support size 1 and all others have support size d, the probability of the event which corresponds to the upper bound is exponentially small in d. We conjecture that a more useful indicator could be ratio of the maximum iteration time over the expected iteration time.

To sum up this preliminary theoretical exploration, the τ term encompasses much more complexity than is usually implied in the literature. This is reflected in the experiments we ran, where the constant was orders of magnitude bigger than the number of cores.

6.7.2. Experimental Results

In order to verify our intuition about the τ variable, we ran several experiments on all three data sets, whose characteristics are reminded in Table 2. δ_l^i is the support size of f_i.

Table 2: Density measures including minimum, average and maximum support size δ_l^i of the factors.

	n	d	density	$\max(\delta_l^i)$	$\min(\delta_l^i)$	$\bar{\delta}_l$	$\max(\delta_l^i)/\bar{\delta}_l$
RCV1	697,641	47,236	0.15%	1,224	4	73.2	16.7
URL	2,396,130	3,231,961	0.003%	414	16	115.6	3.58
Covtype	581,012	54	100%	12	8	11.88	1.01

To estimate τ, we compute the average overlap over 100 iterations, i.e. the difference in labeling between the end of the hundredth iteration and the start of the first iteration on, divided by 100. This quantity is a lower bound on the actual overlap (which is a maximum, not an average). We then take its maximum observed value. The reason why we use an average is that computing the overlap requires using a global counter, which we do not want

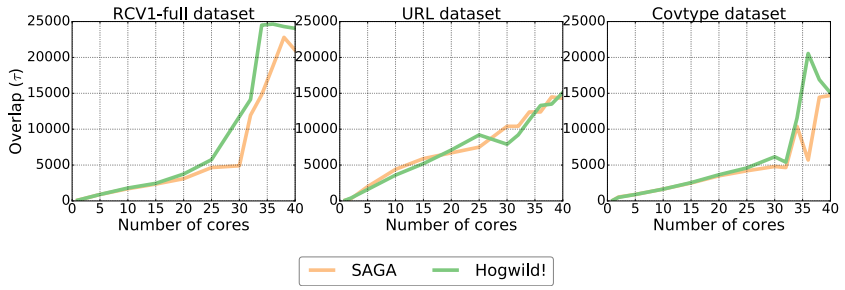


Figure 6: **Overlap**. Overlap as a function of the number of cores for both Asaga and Hogwild on all three data sets.

to update every iteration since it would make it a heavily contentious quantity susceptible of artificially changing the asynchrony pattern of our algorithm.

The results we observe are order of magnitude bigger than p, indicating that τ can indeed not be dismissed as a mere proxy for the number of cores, but has to be more carefully analyzed.

First, we plot the maximum observed τ as a function of the number of cores (see Figure 6). We observe that the relationship does indeed seem to be roughly linear with respect to the number of cores until 30 cores. After 30 cores, we observe what may be a phase transition where the slope increases significantly.

Second, we measured the maximum observed τ as a function of the number of epochs. We omit the figure since we did not observe any dependency; that is, τ does not seem to depend on the number of epochs. We know that it must depend on the number of iterations (since it cannot be bigger, and is an increasing function with respect to that number for example), but it appears that a stable value is reached quite quickly (before one full epoch is done).

If we allowed the computations to run forever, we would eventually observe an event such that τ would reach the upper bound mentioned in Section 6.7.1, so it may be that τ is actually a very slowly increasing function of the number of iterations.

7. Conclusions and Future Work

Building on the recently proposed "perturbed iterate" framework, we have proposed a novel perspective to clarify an important technical issue present in a large fraction of the recent convergence rate proofs for asynchronous parallel optimization algorithms. To resolve it, we have introduced a novel "after read" framework and demonstrated its usefulness by analyzing three asynchronous parallel incremental optimization algorithms, including Asaga, a novel sparse and fully asynchronous variant of the incremental gradient algorithm Saga. Our proof technique accommodates more realistic settings than is usually the case in the literature (such as inconsistent reads and writes and an unbounded gradient); we obtain tighter conditions

than in previous work. In particular, we show that Asaga is linearly faster than Saga under mild conditions, and that sparsity is not always necessary to get linear speedups. Our empirical benchmarks confirm speedups up to 10x.

Schmidt et al. (2016) have shown that Sag enjoys much improved performance when combined with non-uniform sampling and line-search. We have also noticed that our Δ_r constant (being essentially a maximum) sometimes fails to accurately represent the full sparsity distribution of our data sets. Finally, while our algorithm can be directly ported to a distributed master-worker architecture, its communication pattern would have to be optimized to avoid prohibitive costs. Limiting communications can be interpreted as artificially increasing the delay, yielding an interesting trade-off between delay influence and communication costs.

These constitute interesting directions for future analysis, as well as a further exploration of the τ term, which we have shown encompasses more complexity than previously thought.

Acknowledgments

We would like to thank Xinghao Pan for sharing with us their implementation of Kromagnon, as well as Alberto Chiappa for spotting a typo in the proof. This work was partially supported by a Google Research Award and the MSR-Inria Joint Center. FP acknowledges financial support from the chaire *Économie des nouvelles données* with the *data science* joint research initiative with the *fonds AXA pour la recherche*.

Appendix Outline:

- In Appendix A, we adapt the proof from Hofmann et al. (2015) to prove Theorem 2, our convergence result for serial Sparse Saga.
- In Appendix B, we give the complete details for the proof of convergence for Asaga (Theorem 8) as well as its linear speedup regimes (Corollary 9).
- In Appendix C, we give the full details for the proof of convergence for Kromagnon (Theorem 15) as well as a simpler convergence result for both Svrg (Corollary 16) and Kromagnon (Corollary 17) and finally the latter's linear speedup regimes (Corollary 18)
- In Appendix D, we give the full details for the proof of convergence for Hogwild (Theorem 22) as well as its linear speedup regimes (Corollary 23).
- In Appendix E, we explain why adapting the lagged updates implementation of Saga to the asynchronous setting is difficult.
- In Appendix F, we give additional details about the data sets and our implementation.

Appendix A. Sparse Saga – Proof of Theorem 2

Proof sketch for Hofmann et al. (2015). As we will heavily reuse the proof technique from Hofmann et al. (2015), we start by giving its sketch.

First, the authors combine classical strong convexity and Lipschitz inequalities to derive the following inequality (Hofmann et al., 2015, Lemma 1):

$$\mathbf{E}\|x^+ - x^*\|^2 \leq (1-\gamma\mu)\|x-x^*\|^2 + 2\gamma^2\mathbf{E}\|\alpha_i - f_i'(x^*)\|^2 + (4\gamma^2 L - 2\gamma)\big(f(x) - f(x^*)\big). \quad (53)$$

This gives a contraction term, as well as two additional terms; $2\gamma^2\mathbf{E}\|\alpha_i - f_i'(x^*)\|^2$ is a positive variance term, but $(4\gamma^2 L - 2\gamma)\big(f(x) - f(x^*)\big)$ is a negative suboptimality term (provided γ is small enough). The suboptimality term can then be used to cancel the variance one.

Second, the authors use a classical smoothness upper bound to control the variance term and relate it to the suboptimality. However, since the α_i are partial gradients computed at previous time steps, the upper bounds of the variance involve suboptimality at previous time steps, which are not directly relatable to the current suboptimality.

Third, to circumvent this issue, a Lyapunov function is defined to encompass both current and past terms. To finish the proof, Hofmann et al. (2015) show that the Lyapunov function is a contraction.

Proof outline. Fortunately, we can reuse most of the proof from Hofmann et al. (2015) to show that Sparse Saga converges at the same rate as regular Saga. In fact, once we establish that Hofmann et al. (2015, Lemma 1) is still verified we are done.

To prove this, we show that the gradient estimator is unbiased, and then derive close variants of equations (6) and (9) in their paper, which we remind the reader of here:

$$\mathbf{E}\|f_i'(x) - \bar{\alpha}_i\|^2 \leq 2\mathbf{E}\|f_i'(x) - f_i'(x^*)\|^2 + 2\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2 \quad \text{Hofmann et al. (2015, Eq. 6)}$$

$$\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2 \leq \mathbf{E}\|\alpha_i - f_i'(x^*)\|^2 . \quad \text{Hofmann et al. (2015, Eq. 9)}$$

Unbiased gradient estimator. We first show that the update estimator is unbiased. The estimator is unbiased if:

$$\mathbf{E}D_i\bar{\alpha} = \mathbf{E}\alpha_i = \frac{1}{n}\sum_{i=1}^n \alpha_i . \quad (54)$$

We have:

$$\mathbf{E}D_i\bar{\alpha} = \frac{1}{n}\sum_{i=1}^n D_i\bar{\alpha} = \frac{1}{n}\sum_{i=1}^n P_{S_i} D\bar{\alpha} = \frac{1}{n}\sum_{i=1}^n \sum_{v\in S_i} \frac{[\bar{\alpha}]_v e_v}{p_v} = \sum_{v=1}^d \left(\sum_{i\,|\,v\in S_i} 1\right)\frac{[\bar{\alpha}]_v e_v}{np_v} ,$$

where e_v is the vector whose only nonzero component is the v component which is equal to 1.

By definition, $\sum_{i|v\in S_i} 1 = np_v$, which gives us Equation (54).

Deriving Hofmann et al. (2015, Equation 6). We define $\bar{\alpha}_i := \alpha_i - D_i\bar{\alpha}$ (contrary to Hofmann et al. (2015) where the authors define $\bar{\alpha}_i := \alpha_i - \bar{\alpha}$ since they do not concern themselves with sparsity). Using the inequality $\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2$, we get:

$$\mathbf{E}\|f_i'(x) - \bar{\alpha}_i\|^2 \leq 2\mathbf{E}\|f_i'(x) - f_i'(x^*)\|^2 + 2\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2, \quad (55)$$

which is our equivalent to Hofmann et al. (2015, Eq. 6), where only our definition of $\bar{\alpha}_i$ differs.

Deriving Hofmann et al. (2015, Equation 9). We want to prove Hofmann et al. (2015, Eq. 9):

$$\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2 \leq \mathbf{E}\|\alpha_i - f_i'(x^*)\|^2. \quad (56)$$

We have:

$$\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2 = \mathbf{E}\|\alpha_i - f_i'(x^*)\|^2 - 2\mathbf{E}\langle \alpha_i - f_i'(x^*), D_i\bar{\alpha}\rangle + \mathbf{E}\|D_i\bar{\alpha}\|^2. \quad (57)$$

Let $D_{\neg i} := P_{S_i^c} D$; we then have the orthogonal decomposition $D\alpha = D_i\alpha + D_{\neg i}\alpha$ with $D_i\alpha \perp D_{\neg i}\alpha$, as they have disjoint support. We now use the orthogonality of $D_{\neg i}\alpha$ with any vector with support in S_i to simplify the expression (57) as follows:

$$\begin{aligned}
\mathbf{E}\langle \alpha_i - f_i'(x^*), D_i\bar{\alpha}\rangle &= \mathbf{E}\langle \alpha_i - f_i'(x^*), D_i\bar{\alpha} + D_{\neg i}\bar{\alpha}\rangle && (\alpha_i - f_i'(x^*) \perp D_{\neg i}\bar{\alpha}) \\
&= \mathbf{E}\langle \alpha_i - f_i'(x^*), D\bar{\alpha}\rangle \\
&= \langle \mathbf{E}\big(\alpha_i - f_i'(x^*)\big), D\bar{\alpha}\rangle \\
&= \langle \mathbf{E}\alpha_i, D\bar{\alpha}\rangle && (f'(x^*) = 0) \\
&= \bar{\alpha}^\intercal D\bar{\alpha} . && (58)
\end{aligned}$$

Similarly,

$$\begin{aligned}
\mathbf{E}\|D_i\bar{\alpha}\|^2 &= \mathbf{E}\langle D_i\bar{\alpha}, D_i\bar{\alpha}\rangle \\
&= \mathbf{E}\langle D_i\bar{\alpha}, D\bar{\alpha}\rangle && (D_i\bar{\alpha} \perp D_{\neg i}\bar{\alpha}) \\
&= \langle \mathbf{E}D_i\bar{\alpha}, D\bar{\alpha}\rangle \\
&= \bar{\alpha}^\intercal D\bar{\alpha} . && (59)
\end{aligned}$$

Putting it all together,

$$\mathbf{E}\|\bar{\alpha}_i - f_i'(x^*)\|^2 = \mathbf{E}\|\alpha_i - f_i'(x^*)\|^2 - \bar{\alpha}^\intercal D\bar{\alpha} \leq \mathbf{E}\|\alpha_i - f_i'(x^*)\|^2. \tag{60}$$

This is our version of Hofmann et al. (2015, Equation 9), which finishes the proof of Hofmann et al. (2015, Lemma 1). The rest of the proof from Hofmann et al. (2015) can then be reused without modification to obtain Theorem 2. ■

Appendix B. Asaga – Proof of Theorem 8 and Corollary 9

B.1. Initial Recursive Inequality Derivation

We start by proving Equation (15). Let $g_t := g(\hat{x}_t, \hat{\alpha}^t, i_t)$. From (10), we get:

$$\begin{aligned}\|x_{t+1} - x^*\|^2 = \|x_t - \gamma g_t - x^*\|^2 &= \|x_t - x^*\|^2 + \gamma^2\|g_t\|^2 - 2\gamma\langle x_t - x^*, g_t\rangle \\ &= \|x_t - x^*\|^2 + \gamma^2\|g_t\|^2 - 2\gamma\langle \hat{x}_t - x^*, g_t\rangle + 2\gamma\langle \hat{x}_t - x_t, g_t\rangle.\end{aligned}$$

In order to prove Equation (15), we need to bound the $-2\gamma\langle \hat{x}_t - x^*, g_t\rangle$ term. Thanks to Property 3, we can write:

$$\mathbb{E}\langle \hat{x}_t - x^*, g_t\rangle = \mathbb{E}\langle \hat{x}_t - x^*, \mathbf{E}g_t\rangle = \mathbb{E}\langle \hat{x}_t - x^*, f'(\hat{x}_t)\rangle\,.$$

We can now use a classical strong convexity bound as well as a squared triangle inequality to get:

$$\begin{aligned}-\langle \hat{x}_t - x^*, f'(\hat{x}_t)\rangle &\leq -\big(f(\hat{x}_t) - f(x^*)\big) - \frac{\mu}{2}\|\hat{x}_t - x^*\|^2 && \text{(Strong convexity bound)} \\ -\|\hat{x}_t - x^*\|^2 &\leq \|\hat{x}_t - x_t\|^2 - \frac{1}{2}\|x_t - x^*\|^2 && (\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2) \\ -2\gamma\mathbb{E}\langle \hat{x}_t - x^*, g_t\rangle &\leq -\frac{\gamma\mu}{2}\mathbb{E}\|x_t - x^*\|^2 + \gamma\mu\mathbb{E}\|\hat{x}_t - x_t\|^2 - 2\gamma\big(\mathbb{E}f(\hat{x}_t) - f(x^*)\big)\,. && (61)\end{aligned}$$

Putting it all together, we get the initial recursive inequality (15), rewritten here explicitly:

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + \gamma^2\mathbb{E}\|g_t\|^2 + \gamma\mu\mathbb{E}\|\hat{x}_t - x_t\|^2 + 2\gamma\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle - 2\gamma e_t\,, \tag{62}$$

where $a_t := \mathbb{E}\|x_t - x^*\|^2$ and $e_t := \mathbb{E}f(\hat{x}_t) - f(x^*)$.

B.2. Proof of Lemma 10 (inequality in terms of $g_t := g(\hat{x}_t, \hat{\alpha}^t, i_t)$)

To prove Lemma 10, we now bound both $\mathbb{E}\|\hat{x}_t - x_t\|^2$ and $\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle$ with respect to $(\mathbb{E}\|g_u\|^2)_{u\leq t}$.

Bounding $\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle$ ***in terms of*** g_u***.***

$$\begin{aligned}\frac{1}{\gamma}\mathbb{E}\langle \hat{x}_t - x_t, g_t\rangle &= \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\langle G_u^t g_u, g_t\rangle && \text{(by Equation (11))} \\ &\leq \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}|\langle g_u, g_t\rangle| && (G_u^t \text{ diagonal matrices with terms in } \{0,1\}) \\ &\leq \sum_{u=(t-\tau)_+}^{t-1} \frac{\sqrt{\Delta}}{2}(\mathbb{E}\|g_u\|^2 + \mathbb{E}\|g_t\|^2) && \text{(by Proposition 11)} \\ &\leq \frac{\sqrt{\Delta}}{2}\sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 + \frac{\sqrt{\Delta}\tau}{2}\mathbb{E}\|g_t\|^2. && (63)\end{aligned}$$

Bounding $\mathbb{E}\|\hat{x}_t - x_t\|^2$ ***with respect to*** g_u Thanks to the expansion for $\hat{x}_t - x_t$ (11), we get:

$$\|\hat{x}_t - x_t\|^2 \leq \gamma^2 \sum_{u,v=(t-\tau)_+}^{t-1} |\langle G_u^t g_u, G_v^t g_v\rangle| \leq \gamma^2 \sum_{u=(t-\tau)_+}^{t-1} \|g_u\|^2 + \gamma^2 \sum_{\substack{u,v=(t-\tau)_+ \\ u\neq v}}^{t-1} |\langle G_u^t g_u, G_v^t g_v\rangle|\,.$$

Using (18) from Proposition 11, we have that for $u \neq v$:

$$\mathbb{E}|\langle G_u^t g_u, G_v^t g_v\rangle| \leq \mathbb{E}|\langle g_u, g_v\rangle| \leq \frac{\sqrt{\Delta}}{2}(\mathbb{E}\|g_u\|^2 + \mathbb{E}\|g_v\|^2)\,. \tag{64}$$

By taking the expectation and using (64), we get:

$$\begin{aligned}\mathbb{E}\|\hat{x}_t - x_t\|^2 &\leq \gamma^2 \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 + \gamma^2\sqrt{\Delta}(\tau-1)_+ \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 \\ &= \gamma^2\big(1 + \sqrt{\Delta}(\tau-1)_+\big) \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 \\ &\leq \gamma^2\big(1 + \sqrt{\Delta}\tau\big) \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2. && (65)\end{aligned}$$

We can now rewrite (15) in terms of $\mathbb{E}\|g_t\|^2$, which finishes the proof for Lemma 10 (by introducing C_1 and C_2 as specified by 17 in Lemma 10):

$$\begin{aligned}a_{t+1} &\leq (1 - \frac{\gamma\mu}{2})a_t - 2\gamma e_t + \gamma^2\mathbb{E}\|g_t\|^2 + \gamma^3\mu(1 + \sqrt{\Delta}\tau) \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 \\ &\quad + \gamma^2\sqrt{\Delta} \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2 + \gamma^2\sqrt{\Delta}\tau\mathbb{E}\|g_t\|^2 \\ &\leq (1 - \frac{\gamma\mu}{2})a_t - 2\gamma e_t + \gamma^2 C_1\mathbb{E}\|g_t\|^2 + \gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} \mathbb{E}\|g_u\|^2. && (66)\end{aligned}$$

■

B.3. Proof of Lemma 13 (suboptimality bound on $\mathbb{E}\|g_t\|^2$)

We now derive our bound on g_t with respect to suboptimality. From Appendix A, we know that:

$$\mathbb{E}\|g_t\|^2 \leq 2\mathbb{E}\|f'_{i_t}(\hat{x}_t) - f'_{i_t}(x^*)\|^2 + 2\mathbb{E}\|\hat{\alpha}^t_{i_t} - f'_{i_t}(x^*)\|^2 \tag{67}$$

$$\mathbb{E}\|f'_{i_t}(\hat{x}_t) - f'_{i_t}(x^*)\|^2 \leq 2L\big(\mathbb{E}f(\hat{x}_t) - f(x^*)\big) = 2Le_t\,. \tag{68}$$

N. B.: In the following, i_t is a random variable picked uniformly at random in $\{1,...,n\}$, whereas i is a fixed constant.

We still have to handle the $\mathbb{E}\|\hat{\alpha}^t_{i_t} - f'_{i_t}(x^*)\|^2$ term and express it in terms of past suboptimalities. We know from our definition of t that i_t and $\hat{x}_u$ are independent $\forall u < t$. Given the "after read" global ordering, $\mathbf{E}$ – the expectation on i_t conditioned on $\hat{x}_t$ and all "past" $\hat{x}_u$ and i_u – is well defined, and we can rewrite our quantity as:

$$\begin{aligned}\mathbb{E}\|\hat{\alpha}^t_{i_t} - f'_{i_t}(x^*)\|^2 = \mathbb{E}\big(\mathbf{E}\|\hat{\alpha}^t_{i_t} - f'_{i_t}(x^*)\|^2\big) &= \mathbb{E}\frac{1}{n}\sum_{i=1}^n \|\hat{\alpha}^t_i - f'_i(x^*)\|^2 \\ &= \frac{1}{n}\sum_{i=1}^n \mathbb{E}\|\hat{\alpha}^t_i - f'_i(x^*)\|^2.\end{aligned}$$

Now, with i fixed, let $u^t_{i,l}$ be the time of the iterate last used to write the $[\hat{\alpha}^t_i]_l$ quantity, i.e. $[\hat{\alpha}^t_i]_l = [f'_i(\hat{x}_{u^t_{i,l}})]_l$. We know[26] that $0 \leq u^t_{i,l} \leq t-1$. To use this information, we first need to split $\hat{\alpha}_i$ along its dimensions to handle the possible inconsistencies among them:

$$\mathbb{E}\|\hat{\alpha}^t_i - f'_i(x^*)\|^2 = \mathbb{E}\sum_{l=1}^d \big([\hat{\alpha}^t_i]_l - [f'_i(x^*)]_l\big)^2 = \sum_{l=1}^d \mathbb{E}\Big[\big([\hat{\alpha}^t_i]_l - [f'_i(x^*)]_l\big)^2\Big].$$

This gives us:

$$\begin{aligned}\mathbb{E}\|\hat{\alpha}^t_i - f'_i(x^*)\|^2 &= \sum_{l=1}^d \mathbb{E}\Big[\big(f'_i(\hat{x}_{u^t_{i,l}})_l - f'_i(x^*)_l\big)^2\Big] \\ &= \sum_{l=1}^d \mathbb{E}\Big[\sum_{u=0}^{t-1} \mathbb{1}_{\{u^t_{i,l}=u\}}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2\Big] \\ &= \sum_{u=0}^{t-1}\sum_{l=1}^d \mathbb{E}\Big[\mathbb{1}_{\{u^t_{i,l}=u\}}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2\Big].\end{aligned} \tag{69}$$

We will now rewrite the indicator so as to obtain independent events from the rest of the equality. This will enable us to distribute the expectation. Suppose $u > 0$ ($u = 0$ is a special case which we will handle afterwards). $\{u^t_{i,l} = u\}$ requires two things:

26. In the case where $u = 0$, one would have to replace the partial gradient with α^0_i. We omit this special case here for clarity of exposition.

1. at time u, i was picked uniformly at random,
2. (roughly) i was not picked again between u and t.

We need to refine both conditions because we have to account for possible collisions due to asynchrony. We know from our definition of τ that the t^{th} iteration finishes before at $t+\tau+1$, but it may still be unfinished by time $t+\tau$. This means that we can only be sure that an update selecting i at time v has been written to memory at time t if $v \leq t-\tau-1$. Later updates may not have been written yet at time t. Similarly, updates before $v = u+\tau+1$ may be overwritten by the u^{th} update so we cannot infer that they did not select i. From this discussion, we conclude that $u^t_{i,l} = u$ implies that $i_v \neq i$ for all v between $u+\tau+1$ and $t-\tau-1$, though it can still happen that $i_v = i$ for v outside this range.

Using the fact that i_u and i_v are independent for $v \neq u$, we can thus upper bound the indicator function appearing in (69) as follows:

$$\mathbb{1}_{\{u^t_{i,l}=u\}} \leq \mathbb{1}_{\{i_u=i\}}\mathbb{1}_{\{i_v\neq i\ \forall v \text{ s.t. } u+\tau+1\leq v\leq t-\tau-1\}}. \tag{70}$$

This gives us:

$$\begin{aligned}&\mathbb{E}\Big[\mathbb{1}_{\{u^t_{i,l}=u\}}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2\Big] \\ &\quad\leq \mathbb{E}\Big[\mathbb{1}_{\{i_u=i\}}\mathbb{1}_{\{i_v\neq i\ \forall v \text{ s.t. } u+\tau+1\leq v\leq t-\tau-1\}}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2\Big] \\ &\quad\leq P\{i_u = i\}P\{i_v \neq i\ \forall v \text{ s.t. } u+\tau+1 \leq v \leq t-\tau-1\}\mathbb{E}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2 \\ &\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (i_v \perp\!\!\!\perp \hat{x}_u, \forall v \geq u) \\ &\quad\leq \frac{1}{n}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}\mathbb{E}\big(f'_i(\hat{x}_u)_l - f'_i(x^*)_l\big)^2.\end{aligned} \tag{71}$$

Note that the third line used the crucial independence assumption $i_v \perp\!\!\!\perp \hat{x}_u, \forall v \geq u$ arising from our "After Read" ordering. Summing over all dimensions l, we then get:

$$\mathbb{E}\Big[\mathbb{1}_{\{u^t_{i,l}=u\}}\|f'_i(\hat{x}_u) - f'_i(x^*)\|^2\Big] \leq \frac{1}{n}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}\mathbb{E}\|f'_i(\hat{x}_u) - f'_i(x^*)\|^2. \tag{72}$$

So now:

$$\begin{aligned}\mathbb{E}\|\hat{\alpha}^t_{i_t} - f'_{i_t}(x^*)\|^2 - \lambda\tilde{e}_0 &\leq \frac{1}{n}\sum_{i=1}^n\sum_{u=1}^{t-1}\frac{1}{n}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}\mathbb{E}\|f'_i(\hat{x}_u) - f'_i(x^*)\|^2 \\ &= \sum_{u=1}^{t-1}\frac{1}{n}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}\frac{1}{n}\sum_{i=1}^n\mathbb{E}\|f'_i(\hat{x}_u) - f'_i(x^*)\|^2 \\ &= \sum_{u=1}^{t-1}\frac{1}{n}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}\mathbb{E}\Big(\mathbf{E}\|f'_{i_u}(\hat{x}_u) - f'_{i_u}(x^*)\|^2\Big) \quad (i_u \perp\!\!\!\perp \hat{x}_u) \\ &\leq \frac{2L}{n}\sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u \qquad \text{(by Equation 68)} \\ &= \frac{2L}{n}\sum_{u=1}^{(t-2\tau-1)_+}(1-\frac{1}{n})^{t-2\tau-u-1}e_u + \frac{2L}{n}\sum_{u=\max(1,t-2\tau)}^{t-1}e_u\,.\end{aligned} \tag{73}$$

Note that we have excluded $\tilde{e}_0$ from our formula, using a generic λ multiplier. We need to treat the case $u = 0$ differently to bound $\mathbb{1}_{\{u^t_{i,l}=u\}}$. Because all our initial α_i are initialized to a fixed α^0_i, $\{u^t_i = 0\}$ just means that i has not been picked between 0 and $t-\tau-1$, i.e. $\{i_v \neq i \ \forall \ v \text{ s.t. } 0 \leq v \leq t-\tau-1\}$. This means that the $\mathbb{1}_{\{i_u=i\}}$ term in (70) disappears and thus we lose a $\frac{1}{n}$ factor compared to the case where $u > 1$.

Let us now evaluate λ. We have:

$$\begin{aligned}\mathbb{E}\Big[\mathbb{1}_{\{u^t_i=0\}}\|\alpha^0_i - f_i'(x^*)\|^2\Big] &\leq \mathbb{E}\Big[\mathbb{1}_{\{i_v\neq i \ \forall \ v \text{ s.t. } 0\leq v\leq t-\tau-1\}}\|\alpha^0_i - f_i'(x^*)\|^2\Big] \\ &\leq P\{i_v \neq i \ \forall \ v \text{ s.t. } 0 \leq v \leq t-\tau-1\}\mathbb{E}\|\alpha^0_i - f_i'(x^*)\|^2 \\ &\leq (1-\frac{1}{n})^{(t-\tau)_+}\mathbb{E}\|\alpha^0_i - f_i'(x^*)\|^2. \end{aligned} \tag{74}$$

Plugging (73) and (74) into (67), we get Lemma 13:

$$\mathbb{E}\|g_t\|^2 \leq 4Le_t + \frac{4L}{n}\sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u + 4L(1-\frac{1}{n})^{(t-\tau)_+}\tilde{e}_0\,, \tag{75}$$

where we have introduced $\tilde{e}_0 = \frac{1}{2L}\mathbb{E}\|\alpha^0_i - f_i'(x^*)\|^2$. Note that in the original Saga algorithm, a batch gradient is computed to set the $\alpha^0_i = f_i'(x_0)$. In this setting, we can write Lemma 13 using only $\tilde{e}_0 \leq e_0$ thanks to (68). In the more general setting where we initialize all α^0_i to a fixed quantity, we cannot use (68) to bound $\mathbb{E}\|\alpha^0_i - f_i'(x^*)\|^2$ which means that we have to introduce $\tilde{e}_0$.

B.4. Lemma 13 for Ahsvrg

In the simpler case of Ahsvrg as described in 4.2, we have a slight variation of (69):

$$\mathbb{E}\|f_i'(x^t_0) - f_i'(x^*)\|^2 = \sum_{u=0}^{t-1}\mathbb{E}\|\mathbb{1}_{\{u^t_i=u\}}\big(f_i'(\hat{x}_u) - f_i'(x^*)\big)\|^2. \tag{76}$$

Note that there is no sum over dimensions in this case because the full gradient computations and writes are synchronized (so the reference gradient is consistent).

As in Section B.3, we can upper bound the indicator $\mathbb{1}_{\{u^t_i=u\}}$. Now, $\{u^t_{i,l} = u\}$ requires two things: first, the next B variable sampled after the u^{th} update, $\tilde{B}_u$[27], was 1; second, B was 0 for every update between u and t (roughly). Since the batch step is fully synchronized, we do not have to worry about updates from the past overwriting the reference gradient (and the iterates x_u where we compute the gradient contains all past updates because we have waited for every core to finish its current update).

However, updating the state variable s to 1 once a $B = 1$ variable is sampled is not atomic. So it is possible to have iterations with time label bigger than u and that still use an older reference gradient for their update[28]. Fortunately, we can consider the state update as any regular update to shared parameters. As such, Assumption 6 applies to it. This means that we can be certain that the reference gradient has been updated for iterations with time label $v \geq u + \tau + 1$.

This gives us:

$$\begin{aligned}\mathbb{E}\|\mathbb{1}_{\{u^t_i=u\}}\big(f_i'(\hat{x}_u) - f_i'(x^*)\big)\|^2 &\leq \mathbb{E}\Big[\mathbb{1}_{\{\tilde{B}_u=1\}}\mathbb{1}_{\{B_v=0 \ \forall v \text{ s.t. } u+1\leq v\leq t-\tau-1\}}\|f_i'(\hat{x}_u) - f_i'(x^*)\|^2\Big] \\ &\leq \frac{1}{n}\big(1-\frac{1}{n}\big)^{(t-\tau-u-1)_+}\mathbb{E}\|f_i'(\hat{x}_u) - f_i'(x^*)\|^2\,. \end{aligned} \tag{77}$$

This proves Lemma 13 for Ahsvrg (while we actually have a slightly better exponent, $(t-\tau-u-1)_+$, we can upperbound it by the term in Lemma 13). Armed with this result, we can finish the proof of Theorem 8 for Ahsvrg in exactly the same manner as for Asaga. By remarking that the cost to get to iteration t (including computing reference batch gradients) is the same in the sequential and parallel version, we see that our analysis for Corollary 9 for Asaga also applies for Ahsvrg, so both algorithms obey the same convergence and speedup results.

B.5. Master Inequality Derivation

Now, if we combine the bound on $\mathbb{E}\|g_t\|^2$ which we just derived (i.e. Lemma 13) with Lemma 10, we get:

$$\begin{aligned} a_{t+1} \leq &(1-\frac{\gamma\mu}{2})a_t - 2\gamma e_t \\ &+ 4L\gamma^2C_1e_t + \frac{4L\gamma^2C_1}{n}\sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u + 4L\gamma^2C_1(1-\frac{1}{n})^{(t-\tau)_+}\tilde{e}_0 \\ &+ 4L\gamma^2C_2\sum_{u=(t-\tau)_+}^{t-1}e_u + 4L\gamma^2C_2\sum_{u=(t-\tau)_+}^{t-1}(1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0 \\ &+ \frac{4L\gamma^2C_2}{n}\sum_{u=(t-\tau)_+}^{t-1}\sum_{v=1}^{u-1}(1-\frac{1}{n})^{(u-2\tau-v-1)_+}e_v\,. \end{aligned} \tag{78}$$

If we define $H_t := \sum_{u=1}^{t-1}(1-\frac{1}{n})^{(t-2\tau-u-1)_+}e_u$, then we get:

$$\begin{aligned} a_{t+1} \leq &(1-\frac{\gamma\mu}{2})a_t - 2\gamma e_t \\ &+ 4L\gamma^2C_1\big(e_t + (1-\frac{1}{n})^{(t-\tau)_+}\tilde{e}_0\big) + \frac{4L\gamma^2C_1}{n}H_t \\ &+ 4L\gamma^2C_2\sum_{u=(t-\tau)_+}^{t-1}\big(e_u + (1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0\big) + \frac{4L\gamma^2C_2}{n}\sum_{u=(t-\tau)_+}^{t-1}H_u\,, \end{aligned} \tag{79}$$

which is the master inequality (27).

27. We introduce this quantity because the iterations where full gradients are computed do not receive a time label since they do not correspond to updates to the iterates.

28. Conceivably, another core could start a new iteration, draw $B = 1$ and try to update s to 1 themselves. This is not an issue since the operation of updating s to 1 is idempotent. Only one reference gradient would be computed in this case.

B.6. Lyapunov Function and Associated Recursive Inequality

We define $\mathcal{L}_t := \sum_{u=0}^{t}(1-\rho)^{t-u}a_u$ for some target contraction rate $\rho < 1$ to be defined later. We have:

$$\mathcal{L}_{t+1} = (1-\rho)^{t+1}a_0 + \sum_{u=1}^{t+1}(1-\rho)^{t+1-u}a_u = (1-\rho)^{t+1}a_0 + \sum_{u=0}^{t}(1-\rho)^{t-u}a_{u+1}\,. \tag{80}$$

We now use our new bound on a_{t+1}, (79):

$$\begin{aligned}
\mathcal{L}_{t+1} &\le (1-\rho)^{t+1}a_0 + \sum_{u=0}^{t}(1-\rho)^{t-u}\Big[(1-\frac{\gamma\mu}{2})a_u - 2\gamma e_u + 4L\gamma^2 C_1\big(e_u + (1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0\big) \\
&\qquad + \frac{4L\gamma^2 C_1}{n}H_u + \frac{4L\gamma^2 C_2}{n}\sum_{v=(u-\tau)_+}^{u-1}H_v \\
&\qquad + 4L\gamma^2 C_2\sum_{v=(u-\tau)_+}^{u-1}(e_v + (1-\frac{1}{n})^{(v-\tau)_+}\tilde{e}_0)\Big] \\
&\le (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t \\
&\qquad + \sum_{u=0}^{t}(1-\rho)^{t-u}\Big[-2\gamma e_u + 4L\gamma^2 C_1\big(e_u + (1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0\big) \\
&\qquad\qquad + \frac{4L\gamma^2 C_1}{n}H_u + \frac{4L\gamma^2 C_2}{n}\sum_{v=(u-\tau)_+}^{u-1}H_v \\
&\qquad\qquad + 4L\gamma^2 C_2\sum_{v=(u-\tau)_+}^{u-1}(e_v + (1-\frac{1}{n})^{(v-\tau)_+}\tilde{e}_0)\Big].
\end{aligned} \tag{81}$$

We can now rearrange the sums to expose a simple sum of e_u multiplied by factors r_u^t:

$$\mathcal{L}_{t+1} \le (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + \sum_{u=1}^{t}r_u^t e_u + r_0^t\tilde{e}_0\,. \tag{82}$$

B.7. Proof of Lemma 14 (sufficient condition for convergence for Asaga)

We want to make explicit what conditions on ρ and γ are necessary to ensure that r_u^t is negative for all $u \ge 1$. Since each e_u is positive, we will then be able to safely drop the sum term from the inequality. The r_0^t term is a bit trickier and is handled separately. Indeed, trying to enforce that r_0^t is negative results in a significantly worse condition on γ and eventually a convergence rate smaller by a factor of n than our final result. Instead, we handle this term directly in the Lyapunov function.

Computation of r_u^t. Let's now make the multiplying factor explicit. We assume $u \ge 1$.

We split r_u^t into five parts coming from (81):

- r_1, the part coming from the $-2\gamma e_u$ terms;

- r_2, coming from $4L\gamma^2 C_1 e_u$;
- r_3, coming from $\frac{4L\gamma^2 C_1}{n}H_u$;
- r_4, coming from $4L\gamma^2 C_2\sum_{v=(u-\tau)_+}^{u-1}e_v$;
- r_5, coming from $\frac{4L\gamma^2 C_2}{n}\sum_{v=(u-\tau)_+}^{u-1}H_v$.

r_1 is easy to derive. Each of these terms appears only in one inequality. So for u at time t, the term is:

$$r_1 = -2\gamma(1-\rho)^{t-u}. \tag{83}$$

For much the same reasons, r_2 is also easy to derive and is:

$$r_2 = 4L\gamma^2 C_1(1-\rho)^{t-u}. \tag{84}$$

r_3 is a bit trickier, because for a given $v > 0$ there are several H_u which contain e_v. The key insight is that we can rewrite our double sum in the following manner:

$$\begin{aligned}
&\sum_{u=0}^{t}(1-\rho)^{t-u}\sum_{v=1}^{u-1}(1-\frac{1}{n})^{(u-2\tau-v-1)_+}e_v \\
&\quad= \sum_{v=1}^{t-1}e_v\sum_{u=v+1}^{t}(1-\rho)^{t-u}(1-\frac{1}{n})^{(u-2\tau-v-1)_+} \\
&\quad\le \sum_{v=1}^{t-1}e_v\Big[\sum_{u=v+1}^{\min(t,v+2\tau)}(1-\rho)^{t-u} + \sum_{u=v+2\tau+1}^{t}(1-\rho)^{t-u}(1-\frac{1}{n})^{u-2\tau-v-1}\Big] \\
&\quad\le \sum_{v=1}^{t-1}e_v\Big[2\tau(1-\rho)^{t-v-2\tau} + (1-\rho)^{t-v-2\tau-1}\sum_{u=v+2\tau+1}^{t}q^{u-2\tau-v-1}\Big] \\
&\quad\le \sum_{v=1}^{t-1}(1-\rho)^{t-v}e_v(1-\rho)^{-2\tau-1}\big[2\tau + \frac{1}{1-q}\big],
\end{aligned} \tag{85}$$

where we have defined:

$$q := \frac{1-1/n}{1-\rho}, \quad \text{with the assumption } \rho < \frac{1}{n}\,. \tag{86}$$

Note that we have bounded the $\min(t, v+2\tau)$ term by $v+2\tau$ in the first sub-sum, effectively adding more positive terms.

This gives us that at time t, for u:

$$r_3 \le \frac{4L\gamma^2 C_1}{n}(1-\rho)^{t-u}(1-\rho)^{-2\tau-1}\big[2\tau + \frac{1}{1-q}\big]. \tag{87}$$

For r_4 we use the same trick:

$$\begin{aligned}
\sum_{u=0}^{t}(1-\rho)^{t-u}\sum_{v=(u-\tau)_+}^{u-1}e_v &= \sum_{v=0}^{t-1}e_v\sum_{u=v+1}^{\min(t,v+\tau)}(1-\rho)^{t-u} \\
&\le \sum_{v=0}^{t-1}e_v\sum_{u=v+1}^{v+\tau}(1-\rho)^{t-u} \le \sum_{v=0}^{t-1}e_v\tau(1-\rho)^{t-v-\tau}.
\end{aligned} \tag{88}$$

This gives us that at time t, for u:

$$r_4 \leq 4L\gamma^2 C_2(1-\rho)^{t-u}\tau(1-\rho)^{-\tau}\,. \tag{89}$$

Finally we compute r_5 which is the most complicated term. Indeed, to find the factor of e_w for a given $w>0$, one has to compute a triple sum, $\sum_{u=0}^{t}(1-\rho)^{t-u}\sum_{v=(u-\tau)_+}^{u-1} H_v$. We start by computing the factor of e_w in the inner double sum, $\sum_{v=(u-\tau)_+}^{u-1} H_v$.

$$\sum_{v=(u-\tau)_+}^{u-1}\sum_{w=1}^{v-1}(1-\frac{1}{n})^{(v-2\tau-w-1)_+}e_w = \sum_{w=1}^{u-2}e_w \sum_{v=\max(w+1,u-\tau)}^{u-1}(1-\frac{1}{n})^{(v-2\tau-w-1)_+}\,. \tag{90}$$

Now there are at most τ terms for each e_w. If $w \leq u-3\tau-1$, then the exponent is positive in every term and it is always bigger than $u-3\tau-1-w$, which means we can bound the sum by $\tau(1-\frac{1}{n})^{u-3\tau-1-w}$. Otherwise we can simply bound the sum by τ. We get:

$$\sum_{v=(u-\tau)_+}^{u-1} H_v \leq \sum_{w=1}^{u-2}\big[\mathbb{1}_{\{u-3\tau\leq w\leq u-2\}}\tau + \mathbb{1}_{\{w\leq u-3\tau-1\}}\tau(1-\frac{1}{n})^{u-3\tau-1-w}\big]e_w\,. \tag{91}$$

This means that for w at time t:

$$\begin{aligned}
r_5 &\leq \frac{4L\gamma^2C_2}{n}\sum_{u=0}^{t}(1-\rho)^{t-u}\big[\mathbb{1}_{\{u-3\tau\leq w\leq u-2\}}\tau + \mathbb{1}_{\{w\leq u-3\tau-1\}}\tau(1-\frac{1}{n})^{u-3\tau-1-w}\big]\\
&\leq \frac{4L\gamma^2C_2}{n}\Big[\sum_{u=w+2}^{\min(t,w+3\tau)}\tau(1-\rho)^{t-u} + \sum_{u=w+3\tau+1}^{t}\tau(1-\frac{1}{n})^{u-3\tau-1-w}(1-\rho)^{t-u}\Big]\\
&\leq \frac{4L\gamma^2C_2}{n}\tau\Big[(1-\rho)^{t-w}(1-\rho)^{-3\tau}3\tau\\
&\qquad + (1-\rho)^{t-w}(1-\rho)^{-1-3\tau}\sum_{u=w+3\tau+1}^{t}(1-\frac{1}{n})^{u-3\tau-1-w}(1-\rho)^{-u+3\tau+1+w}\Big]\\
&\leq \frac{4L\gamma^2C_2}{n}\tau(1-\rho)^{t-w}(1-\rho)^{-3\tau-1}\big(3\tau+\frac{1}{(1-q)}\big)\,.
\end{aligned} \tag{92}$$

By combining the five terms together (83, 84, 87, 89 and 92), we get that $\forall u$ s.t. $1\leq u\leq t$:

$$\begin{aligned}
r_u^t \leq (1-\rho)^{t-u}\Big[&-2\gamma + 4L\gamma^2C_1 + \frac{4L\gamma^2C_1}{n}(1-\rho)^{-2\tau-1}\big(2\tau+\frac{1}{1-q}\big)\\
&+4L\gamma^2C_2\tau(1-\rho)^{-\tau} + \frac{4L\gamma^2C_2}{n}\tau(1-\rho)^{-3\tau-1}\big(3\tau+\frac{1}{1-q}\big)\Big].
\end{aligned} \tag{93}$$

Computation of r_0^t. Recall that we treat the $\tilde{e}_0$ term separately in Section B.3. The initialization of Saga creates an initial synchronization, which means that the contribution of $\tilde{e}_0$ in our bound on $\mathbb{E}\|g_t\|^2$ (75) is roughly n times bigger than the contribution of any e_u for $1<u<t$.[29] In order to safely handle this term in our Lyapunov inequality, we only need to prove that it is bounded by a reasonable constant. Here again, we split r_0^t in five contributions coming from (81):

29. This is explained in details right before (74).

- r_1, the part coming from the $-2\gamma e_u$ terms;
- r_2, coming from $4L\gamma^2C_1e_u$;
- r_3, coming from $4L\gamma^2C_1(1-\frac{1}{n})^{(u-\tau)_+}\tilde{e}_0$;
- r_4, coming from $4L\gamma^2C_2\sum_{v=(u-\tau)_+}^{u-1}e_v$;
- r_5, coming from $4L\gamma^2C_2\sum_{v=(u-\tau)_+}^{u-1}(1-\frac{1}{n})^{(v-\tau)_+}\tilde{e}_0$.

Note that there is no $\tilde{e}_0$ in H_t, which is why we can safely ignore these terms here.

We have $r_1 = -2\gamma(1-\rho)^t$ and $r_2 = 4L\gamma^2C_1(1-\rho)^t$.

Let us compute r_3.

$$\begin{aligned}
&\sum_{u=0}^{t}(1-\rho)^{t-u}(1-\frac{1}{n})^{(u-\tau)_+}\\
&\qquad= \sum_{u=0}^{\min(t,\tau)}(1-\rho)^{t-u} + \sum_{u=\tau+1}^{t}(1-\rho)^{t-u}(1-\frac{1}{n})^{u-\tau}\\
&\qquad\leq (\tau+1)(1-\rho)^{t-\tau} + (1-\rho)^{t-\tau}\sum_{u=\tau+1}^{t}(1-\rho)^{\tau-u}(1-\frac{1}{n})^{u-\tau}\\
&\qquad\leq (1-\rho)^t(1-\rho)^{-\tau}\big(\tau+1+\frac{1}{1-q}\big)\,.
\end{aligned} \tag{94}$$

This gives us:

$$r_3 \leq (1-\rho)^t 4L\gamma^2C_1(1-\rho)^{-\tau}\big(\tau+1+\frac{1}{1-q}\big)\,. \tag{95}$$

We have already computed r_4 for $u>0$ and the computation is exactly the same for $u=0$. $r_4 \leq (1-\rho)^t 4L\gamma^2C_2\frac{\tau}{1-\rho}$.

Finally we compute r_5.

$$\begin{aligned}
&\sum_{u=0}^{t}(1-\rho)^{t-u}\sum_{v=(u-\tau)_+}^{u-1}(1-\frac{1}{n})^{(v-\tau)_+}\\
&\qquad= \sum_{v=1}^{t-1}\sum_{u=v+1}^{\min(t,v+\tau)}(1-\rho)^{t-u}(1-\frac{1}{n})^{(v-\tau)_+}\\
&\qquad\leq \sum_{v=1}^{\min(t-1,\tau)}\sum_{u=v+1}^{v+\tau}(1-\rho)^{t-u} + \sum_{v=\tau+1}^{t-1}\sum_{u=v+1}^{\min(t,v+\tau)}(1-\rho)^{t-u}(1-\frac{1}{n})^{v-\tau}\\
&\qquad\leq \tau^2(1-\rho)^{t-2\tau} + \sum_{v=\tau+1}^{t-1}(1-\frac{1}{n})^{v-\tau}\tau(1-\rho)^{t-v-\tau}\\
&\qquad\leq \tau^2(1-\rho)^{t-2\tau} + \tau(1-\rho)^t(1-\rho)^{-2\tau}\sum_{v=\tau+1}^{t-1}(1-\frac{1}{n})^{v-\tau}\tau(1-\rho)^{-v+\tau}\\
&\qquad\leq (1-\rho)^t(1-\rho)^{-2\tau}\big(\tau^2+\tau\frac{1}{1-q}\big)\,.
\end{aligned} \tag{96}$$

Which means:

$$r_5 \leq (1-\rho)^t 4L\gamma^2 C_2 (1-\rho)^{-2\tau}\big(\tau^2 + \tau\frac{1}{1-q}\big). \tag{97}$$

Putting it all together, we get that: $\forall t \geq 0$,

$$r_0^t \leq (1-\rho)^t\Big[\Big(-2\gamma + 4L\gamma^2 C_1 + 4L\gamma^2 C_2 \frac{\tau}{1-\rho}\Big)\frac{e_0}{\tilde{e}_0} + 4L\gamma^2 C_1 (1-\rho)^{-\tau}\big(\tau + 1 + \frac{1}{1-q}\big) + 4L\gamma^2 C_2 \tau (1-\rho)^{-2\tau}\big(\tau + \frac{1}{1-q}\big)\Big]. \tag{98}$$

Sufficient condition for convergence. We need all $r_u^t, u \geq 1$ to be negative so we can safely drop them from (82). Note that for every u, this is the same condition. We will reduce that condition to a second-order polynomial sign condition. We also remark that since $\gamma \geq 0$, we can upper bound our terms in γ and γ^2 in this upcoming polynomial, which will give us sufficient conditions for convergence.

Now, recall that $C_2(\gamma)$ (as defined in (17)) depends on γ. We thus need to expand it once more to find our conditions. We have:

$$C_1 = 1 + \sqrt{\Delta}\tau; \qquad C_2 = \sqrt{\Delta} + \gamma\mu C_1\,.$$

Dividing the bracket in (93) by γ and rearranging as a second degree polynomial, we get the condition:

$$4L\left(C_1 + \frac{C_1}{n}(1-\rho)^{-2\tau-1}\Big[2\tau + \frac{1}{1-q}\Big] + \Big[\frac{\sqrt{\Delta}\tau}{(1-\rho)^{\tau}} + \frac{\sqrt{\Delta}\tau}{n}(1-\rho)^{-3\tau-1}(3\tau + \frac{1}{1-q})\Big]\right)\gamma + 8\mu C_1 L\tau\Big[(1-\rho)^{-\tau} + \frac{1}{n}(1-\rho)^{-3\tau-1}(3\tau + \frac{1}{1-q})\Big]\gamma^2 + 2 \leq 0\,. \tag{99}$$

The discriminant of this polynomial is always positive, so γ needs to be between its two roots. The smallest is negative, so the condition is not relevant to our case (where $\gamma > 0$). By solving analytically for the positive root ϕ, we get an upper bound condition on γ that can be used for any overlap τ and guarantee convergence. Unfortunately, for large τ, the upper bound becomes exponentially small because of the presence of τ in the exponent in (99). More specifically, by using the bound $1/(1-\rho) \leq \exp(2\rho)$[30] and thus $(1-\rho)^{-\tau} \leq \exp(2\tau\rho)$ in (99), we would obtain factors of the form $\exp(\tau/n)$ in the denominator for the root ϕ (recall that $\rho < 1/n$).

Our Lemma 14 is derived instead under the assumption that $\tau \leq \mathcal{O}(n)$, with the constants chosen in order to make the condition (99) more interpretable and to relate our convergence result with the standard SAGA convergence (see Theorem 2). As explained in Section 6.7, the assumption that $\tau \leq \mathcal{O}(n)$ appears reasonable in practice. First, by using Bernoulli's inequality, we have:

$$(1-\rho)^{k\tau} \geq 1 - k\tau\rho \qquad \text{for integers} \quad k\tau \geq 0\,. \tag{100}$$

30. This bound can be derived from the inequality $(1 - x/2) \geq \exp(-x)$ which is valid for $0 \leq x \leq 1.59$.

To get manageable constants, we make the following slightly more restrictive assumptions on the target rate ρ[31] and overlap τ:[32]

$$\rho \leq \frac{1}{4n} \tag{101}$$

$$\tau \leq \frac{n}{10}\,. \tag{102}$$

We then have:

$$\frac{1}{1-q} \leq \frac{4n}{3} \tag{103}$$

$$\frac{1}{1-\rho} \leq \frac{4}{3} \tag{104}$$

$$k\tau\rho \leq \frac{3}{40} \qquad \text{for } 1 \leq k \leq 3 \tag{105}$$

$$(1-\rho)^{-k\tau} \leq \frac{1}{1-k\tau\rho} \leq \frac{40}{37} \qquad \text{for } 1 \leq k \leq 3 \text{ and by using (100).} \tag{106}$$

We can now upper bound loosely the three terms in brackets appearing in (99) as follows:

$$(1-\rho)^{-2\tau-1}\big[2\tau + \frac{1}{1-q}\big] \leq 3n \tag{107}$$

$$\sqrt{\Delta}\tau(1-\rho)^{-\tau} + \frac{\sqrt{\Delta}\tau}{n}(1-\rho)^{-3\tau-1}(3\tau + \frac{1}{1-q}) \leq 4\sqrt{\Delta}\tau \leq 4C_1 \tag{108}$$

$$(1-\rho)^{-\tau} + \frac{1}{n}(1-\rho)^{-3\tau-1}(3\tau + \frac{1}{1-q}) \leq 4\,. \tag{109}$$

By plugging (107)–(109) into (99), we get the simpler sufficient condition on γ:

$$-1 + 16LC_1\gamma + 16LC_1\mu\tau\gamma^2 \leq 0\,. \tag{110}$$

The positive root ϕ is:

$$\phi = \frac{16LC_1(\sqrt{1 + \frac{\mu\tau}{4LC_1}} - 1)}{32LC_1\mu\tau} = \frac{\sqrt{1 + \frac{\mu\tau}{4LC_1}} - 1}{2\mu\tau}\,. \tag{111}$$

We simplify it further by using the inequality:[33]

$$\sqrt{x} - 1 \geq \frac{x-1}{2\sqrt{x}} \qquad \forall x > 0\,. \tag{112}$$

Using (112) in (111), and recalling that $\kappa := L/\mu$, we get:

$$\phi \geq \frac{1}{16LC_1\sqrt{1 + \frac{\tau}{4\kappa C_1}}}\,. \tag{113}$$

31. Note that we already expected $\rho < 1/n$.

32. This bound on τ is reasonable in practice, see Appendix 6.7.

33. This inequality can be derived by using the concavity property $f(y) \leq f(x) + (y-x)f'(x)$ on the differentiable concave function $f(x) = \sqrt{x}$ with $y = 1$.

Since $\frac{\tau}{C_1} = \frac{\tau}{1+\sqrt{\Delta}\tau} \leq \min(\tau, \frac{1}{\sqrt{\Delta}})$, we get that a sufficient condition on our step size is:

$$\gamma \leq \frac{1}{16L(1+\sqrt{\Delta}\tau)\sqrt{1+\frac{1}{4\kappa}\min(\tau, \frac{1}{\sqrt{\Delta}})}}\,. \tag{114}$$

Subject to our conditions on γ, ρ and τ, we then have that: $r_u^t \leq 0$ for all u s.t. $1 \leq u \leq t$. This means we can rewrite (82) as:

$$\mathcal{L}_{t+1} \leq (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + r_0^t\tilde{e}_0\,. \tag{115}$$

Now, we could finish the proof from this inequality, but it would only give us a convergence result in terms of $a_t = \mathbb{E}\|x_t - x^*\|^2$. A better result would be in terms of the suboptimality at $\hat{x}_t$ (because $\hat{x}_t$ is a real quantity in the algorithm whereas x_t is virtual). Fortunately, to get such a result, we can easily adapt (115).

We make e_t appear on the left side of (115), by adding γ to r_t^t in (82):[34]

$$\gamma e_t + \mathcal{L}_{t+1} \leq (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + \sum_{u=1}^{t-1} r_u^t e_u + r_0^t\tilde{e}_0 + (r_t^t + \gamma)e_t. \tag{116}$$

We now require the stronger property that $\gamma + r_t^t \leq 0$, which translates to replacing -2γ with $-\gamma$ in (93):

$$\begin{aligned} 0 \geq \Big[&-\gamma + 4L\gamma^2 C_1 + \frac{4L\gamma^2 C_1}{n}(1-\rho)^{-2\tau-1}\big(2\tau + \frac{1}{1-q}\big) \\ &+ 4L\gamma^2 C_2 \tau(1-\rho)^{-\tau} + \frac{4L\gamma^2 C_2}{n}\tau(1-\rho)^{-3\tau}\big(3\tau + \frac{1}{1-q}\big)\Big]. \end{aligned} \tag{117}$$

We can easily derive a new stronger condition on γ under which we can drop all the $e_u, u > 0$ terms in (116):

$$\gamma \leq \gamma^* = \frac{1}{32L(1+\sqrt{\Delta}\tau)\sqrt{1+\frac{1}{8\kappa}\min(\tau, \frac{1}{\sqrt{\Delta}})}}, \tag{118}$$

and thus under which we get:

$$\gamma e_t + \mathcal{L}_{t+1} \leq (1-\rho)^{t+1}a_0 + (1-\frac{\gamma\mu}{2})\mathcal{L}_t + r_0^t\tilde{e}_0. \tag{119}$$

This finishes the proof of Lemma 14. ■

34. We could use any multiplier from 0 to 2γ, but choose γ for simplicity. For this reason and because our analysis of the r_t^t term was loose, we could derive a tighter bound, but it does not change the leading terms.

B.8. Proof of Theorem 8 (convergence guarantee and rate of Asaga)

End of Lyapunov convergence. We continue with the assumptions of Lemma 14 which gave us (119). Thanks to (98), we can also rewrite $r_0^t \leq (1-\rho)^{t+1}A$ where A is a constant which depends on n, Δ, γ and L but is finite and crucially does not depend on t. In fact, by reusing similar arguments as in B.7, we can show the loose bound $A \leq \gamma n$ under the assumptions of Lemma 14 (including $\gamma \leq \gamma^*$).[35] We then have:

$$\begin{aligned} \mathcal{L}_{t+1} \leq \gamma e_t + \mathcal{L}_{t+1} &\leq (1-\frac{\gamma\mu}{2})\mathcal{L}_t + (1-\rho)^{t+1}(a_0 + A\tilde{e}_0) \\ &\leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (a_0 + A\tilde{e}_0)\sum_{k=0}^{t+1}(1-\rho)^{t+1-k}(1-\frac{\gamma\mu}{2})^k. \end{aligned} \tag{120}$$

We have two linearly contracting terms. The sum contracts linearly with the minimum geometric rate factor between $\gamma\mu/2$ and ρ. If we define $m := \min(\rho, \gamma\mu/2)$, $M := \max(\rho, \gamma\mu/2)$ and $\rho^* := \nu m$ with $0 < \nu < 1$,[36] we then get:[37]

$$\begin{aligned} \gamma e_t \leq \gamma e_t + \mathcal{L}_{t+1} &\leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (a_0 + A\tilde{e}_0)\sum_{k=0}^{t+1}(1-m)^{t+1-k}(1-M)^k \\ &\leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (a_0 + A\tilde{e}_0)\sum_{k=0}^{t+1}(1-\rho^*)^{t+1-k}(1-M)^k \\ &\leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (a_0 + A\tilde{e}_0)(1-\rho^*)^{t+1}\sum_{k=0}^{t+1}(1-\rho^*)^{-k}(1-M)^k \\ &\leq (1-\frac{\gamma\mu}{2})^{t+1}\mathcal{L}_0 + (1-\rho^*)^{t+1}\frac{1}{1-\eta}(a_0 + A\tilde{e}_0) \\ &\leq (1-\rho^*)^{t+1}\big(a_0 + \frac{1}{1-\eta}(a_0 + A\tilde{e}_0)\big), \end{aligned} \tag{121}$$

where $\eta := \frac{1-M}{1-\rho^*}$. We have $\frac{1}{1-\eta} = \frac{1-\rho^*}{M-\rho^*}$.

By taking $\nu = \frac{4}{5}$ and setting $\rho = \frac{1}{4n}$ – its maximal value allowed by the assumptions of Lemma 14 – we get $M \geq \frac{1}{4n}$ and $\rho^* \leq \frac{1}{5n}$, which means $\frac{1}{1-\eta} \leq 20n$. All told, using $A \leq \gamma n$, we get:

$$e_t \leq (1-\rho^*)^{t+1}\tilde{C}_0, \tag{122}$$

where:

$$\tilde{C}_0 := \frac{21n}{\gamma}\Big(\|x_0 - x^*\|^2 + \gamma\frac{n}{2L}\mathbb{E}\|\alpha_i^0 - f_i'(x^*)\|^2\Big)\,. \tag{123}$$

Since we set $\rho = \frac{1}{4n}, \nu = \frac{4}{5}$, we have $\nu\rho = \frac{1}{5n}$. Using a step size $\gamma = \frac{a}{L}$ as in Theorem 8, we get $\nu\frac{\gamma\mu}{2} = \frac{2a}{5\kappa}$. We thus obtain a geometric rate of $\rho^* = \min\{\frac{1}{5n}, a\frac{2}{5\kappa}\}$, which we simplified

35. In particular, note that e_0 does not appear in the definition of A because it turns out that the parenthesis group multiplying e_0 in (98) is negative. Indeed, it contains less positive terms than (93) which we showed to be negative under the assumptions from Lemma 14.

36. ν is introduced to circumvent the problematic case where ρ and $\gamma\mu/2$ are too close together, which does not prevent the geometric convergence, but makes the constant $\frac{1}{1-\eta}$ potentially very big (in the case both terms are equal, the sum even becomes an annoying linear term in t).

37. Note that if $m \neq \rho$, we can perform the index change $t+1-k \to k$ to get the sum.

to $\frac{1}{5}\min\{\frac{1}{n}, a\frac{1}{\kappa}\}$ in Theorem 8, finishing the proof. We also observe that $\tilde{C}_0 \leq \frac{60n}{\gamma} C_0$, with C_0 defined in Theorem 2. ■

B.9. Proof of Corollary 9 (speedup regimes for Asaga)

Referring to Hofmann et al. (2015) and our own Theorem 2, the geometric rate factor of Saga is $\frac{1}{5}\min\{\frac{1}{n}, \frac{a}{\kappa}\}$ for a step size of $\gamma = \frac{a}{5L}$. We start by proving the first part of the corollary which considers the step size $\gamma = \frac{a}{L}$ with $a = a^*(\tau)$. We distinguish between two regimes to study the parallel speedup our algorithm obtains and to derive a condition on τ for which we have a linear speedup.

Well-conditioned regime. In this regime, $n > \kappa$ and the geometric rate factor of sequential Saga is $\frac{1}{5n}$. To get a linear speedup (up to a constant factor), we need to enforce $\rho^* = \Omega(\frac{1}{n})$. We recall that $\rho^* = \min\{\frac{1}{5n}, a\frac{1}{5\kappa}\}$.

We already have $\frac{1}{5n} = \Omega(\frac{1}{n})$. This means that we need τ to verify $\frac{a^*(\tau)}{5\kappa} = \Omega(\frac{1}{n})$, where $a^*(\tau) = \frac{1}{32\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)}$ according to Theorem 8. Recall that $\xi(\kappa,\Delta,\tau) := \sqrt{1 + \frac{1}{8\kappa}\min\{\frac{1}{\sqrt{\Delta}}, \tau\}}$. Up to a constant factor, this means we can give the following sufficient condition:

$$\frac{1}{\kappa\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)} = \Omega\Big(\frac{1}{n}\Big) \tag{124}$$

i.e.

$$\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau) = \mathcal{O}\Big(\frac{n}{\kappa}\Big)\,. \tag{125}$$

We now consider two alternatives, depending on whether κ is bigger than $\frac{1}{\sqrt{\Delta}}$ or not. If $\kappa \geq \frac{1}{\sqrt{\Delta}}$, then $\xi(\kappa,\Delta,\tau) < 2$ and we can rewrite the sufficient condition (125) as:

$$\tau = \mathcal{O}(1)\frac{n}{\kappa\sqrt{\Delta}}. \tag{126}$$

In the alternative case, $\kappa \leq \frac{1}{\sqrt{\Delta}}$. Since $a^*(\tau)$ is decreasing in τ, we can suppose $\tau \geq \frac{1}{\sqrt{\Delta}}$ without loss of generality and thus $\xi(\kappa,\Delta,\tau) = \sqrt{1+\frac{1}{8\kappa\sqrt{\Delta}}}$. We can then rewrite the sufficient condition (125) as:

$$\begin{aligned}\frac{\tau\sqrt{\Delta}}{\sqrt{\kappa}\sqrt[4]{\Delta}} &= \mathcal{O}(\frac{n}{\kappa})\,;\\ \tau &= \mathcal{O}(1)\frac{n}{\sqrt{\kappa}\sqrt[4]{\Delta}}\,.\end{aligned} \tag{127}$$

We observe that since we have supposed that $\kappa \leq \frac{1}{\sqrt{\Delta}}$, we have $\sqrt{\kappa\sqrt{\Delta}} \leq \kappa\sqrt{\Delta} \leq 1$, which means that our initial assumption that $\tau < \frac{n}{10}$ is stronger than condition (127).

We can now combine both cases to get the following sufficient condition for the geometric rate factor of Asaga to be the same order as sequential Saga when $n > \kappa$:

$$\tau = \mathcal{O}(1)\frac{n}{\kappa\sqrt{\Delta}}; \quad \tau = \mathcal{O}(n)\,. \tag{128}$$

Ill-conditioned regime. In this regime, $\kappa > n$ and the geometric rate factor of sequential Saga is $a\frac{1}{\kappa}$. Here, to obtain a linear speedup, we need $\rho^* = \mathcal{O}(\frac{1}{\kappa})$. Since $\frac{1}{n} > \frac{1}{\kappa}$, all we require is that $\frac{a^*(\tau)}{\kappa} = \Omega(\frac{1}{\kappa})$ where $a^*(\tau) = \frac{1}{32\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)}$, which reduces to $a^*(\tau) = \Omega(1)$.

We can give the following sufficient condition:

$$\frac{1}{\left(1+\tau\sqrt{\Delta}\right)\xi(\kappa,\Delta,\tau)} = \Omega(1) \tag{129}$$

Using that $\frac{1}{n} \leq \Delta \leq 1$ and that $\kappa > n$, we get that $\xi(\kappa,\Delta,\tau) \leq 2$, which means our sufficient condition becomes:

$$\begin{aligned}\tau\sqrt{\Delta} &= \mathcal{O}(1)\\ \tau &= \frac{\mathcal{O}(1)}{\sqrt{\Delta}}.\end{aligned} \tag{130}$$

This finishes the proof for the first part of Corollary 9.

Universal step size. If $\tau = \mathcal{O}(\frac{1}{\sqrt{\Delta}})$, then $\xi(\kappa,\Delta,\tau) = \mathcal{O}(1)$ and $(1+\tau\sqrt{\Delta}) = \mathcal{O}(1)$, and thus $a^*(\tau) = \Omega(1)$ (for any n and κ). This means that the universal step size $\gamma = \Theta(1/L)$ satisfies $\gamma \leq a^*(\tau)$ for any κ, giving the same rate factor $\Omega(\min\{\frac{1}{n}, \frac{1}{\kappa}\})$ that sequential Saga has, completing the proof for the second part of Corollary 9. ■

Appendix C. Kromagnon – Proof of Theorem 15 and Corollary 18

C.1. Proof of Lemma 19 (suboptimality bound on $\mathbb{E}\|g_t\|^2$)

Mania et al. (2017, Lemma 9), tells us that for serial sparse Svrg we have for all $km \leq t \leq (k+1)m-1$:

$$\mathbb{E}\|g_t\|^2 \leq 2\mathbb{E}\|f'_{i_t}(\hat{x}_t) - f'_{i_t}(x^*)\|^2 + 2\mathbb{E}\|f'_{i_t}(\hat{x}_k) - f'_{i_t}(x^*)\|^2. \tag{131}$$

This remains true in the case of Kromagnon. We can use Hofmann et al. (2015, Equations 7 and 8) to bound both terms in the following manner:

$$\mathbb{E}\|g_t\|^2 \leq 4L(\mathbb{E}f(\hat{x}_t) - f(x^*)) + 4L(\mathbb{E}f(\tilde{x}_k) - f(x^*)) \leq 4Le_t + 4L\tilde{e}_k\,. \tag{132}$$

■

C.2. Proof of Theorem 15 (convergence guarantee and rate of Kromagnon)

Master inequality derivation. As in our Asaga analysis, we plug Lemma 19 in Lemma 10, which gives us that for all $k \geq 0, km \leq t \leq (k+1)m-1$:

$$a_{t+1} \leq (1-\frac{\gamma\mu}{2})a_t + \gamma^2 C_1(4Le_t + 4L\tilde{e}_k) + \gamma^2 C_2 \sum_{u=\max(km,t-\tau)}^{t-1} (4Le_t + 4L\tilde{e}_k) - 2\gamma e_t\,. \tag{133}$$

By grouping the $\tilde{e}_k$ and the e_t terms we get our master inequality (42):

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + (4L\gamma^2 C_1 - 2\gamma)e_t + 4L\gamma^2 C_2 \sum_{u=\max(km,t-\tau)}^{t-1} e_u + (4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,.$$

Contraction inequality derivation. We now adopt the same method as in the original Svrg paper (Johnson and Zhang, 2013); we sum the master inequality over a whole epoch and then we use the randomization trick:

$$\tilde{e}_k = \mathbb{E}f(\tilde{x}_k) - f(x^*) = \frac{1}{m}\sum_{t=(k-1)m}^{km-1} e_t \tag{134}$$

This gives us:

$$\begin{aligned}\sum_{t=km+1}^{(k+1)m} a_t \leq &(1 - \frac{\gamma\mu}{2})\sum_{t=km}^{(k+1)m-1} a_t + (4L\gamma^2 C_1 - 2\gamma)\sum_{t=km}^{(k+1)m-1} e_t \\ &+ 4L\gamma^2 C_2 \sum_{t=km}^{(k+1)m-1}\sum_{u=\max(km,t-\tau)}^{t-1} e_u + m(4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,.\end{aligned} \tag{135}$$

Since $1 - \frac{\gamma\mu}{2} < 1$, we can upper bound it by 1, and then remove all the telescoping terms from (135). We also have:

$$\begin{aligned}\sum_{t=km}^{(k+1)m-1}\sum_{u=\max(km,t-\tau)}^{t-1} e_u = \sum_{u=km}^{(k+1)m-2}\sum_{t=u+1}^{\min((k+1)m-1,u+\tau)} e_u &\leq \tau \sum_{u=km}^{(k+1)m-2} e_u \\ &\leq \tau \sum_{u=km}^{(k+1)m-1} e_u \,.\end{aligned} \tag{136}$$

All told:

$$a_{(k+1)m} \leq a_{km} + (4L\gamma^2 C_1 + 4L\gamma^2\tau C_2 - 2\gamma)\sum_{t=km}^{(k+1)m-1} e_t + m(4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,. \tag{137}$$

Now we use the randomization trick (134):

$$a_{(k+1)m} \leq a_{km} + (4L\gamma^2 C_1 + 4L\gamma^2\tau C_2 - 2\gamma)m\tilde{e}_{k+1} + m(4L\gamma^2 C_1 + 4L\gamma^2\tau C_2)\tilde{e}_k \,. \tag{138}$$

Finally, in order to get a recursive inequality in $\tilde{e}_k$, we can remove the positive $a_{(k+1)m}$ term from the left-hand side of (138) and bound the a_{km} term on the right-hand side by $2e_{km}/\mu$ using a standard strong convexity inequality. We get our final contraction inequality (45):

$$(2\gamma m - 4mL\gamma^2 C_1 - 4mL\gamma^2\tau C_2)\tilde{e}_{k+1} \leq (\frac{2}{\mu} + 4mL\gamma^2 C_1 + 4mL\gamma^2\tau C_2)\tilde{e}_k \,.$$

■

C.3. Proof of Corollary 16, Corollary 17 and Corollary 18 (speedup regimes)

***A simpler result for* Svrg.** The standard convergence rate for serial Svrg is given by:

$$\theta := \frac{\frac{1}{\mu\gamma m} + 2L\gamma}{1 - 2L\gamma} \,. \tag{139}$$

If we define a such that $\gamma = a/4L$, we obtain:

$$\theta = \frac{\frac{4\kappa}{am} + \frac{a}{2}}{1 - \frac{a}{2}} \,. \tag{140}$$

Now, since we need $\theta \leq 1$, we see that we require $a \leq 1$. The optimal value of the denominator is then 1 (when $a = 0$), whereas the worst case value is $1/2$ ($a = 1$). We can thus upper bound θ by replacing the denominator with $1/2$, while satisfied that we do not lose more than a factor of 2. This gives us:

$$\theta \leq \frac{8\kappa}{am} + a \,. \tag{141}$$

Enforcing $\theta \leq 1/2$ can be done easily by choosing $a \leq 1/4$ and $m = 32\kappa/a$. Now, to be able to compare algorithms easily, we want to frame our result in terms of rate factor per gradient computation ρ, such that (38) is verified:

$$\mathbb{E}f(\tilde{x}_k) - f(x^*) \leq (1 - \rho)^{k(2m+n)}\left(\mathbb{E}f(x_0) - f(x^*)\right) \qquad \forall k \geq 0 \,.$$

We define $\rho_b := 1 - \theta$. We want to estimate ρ such that $(1 - \rho)^{2m+n} = 1 - \rho_b$. We get that $\rho = 1 - (1 - \rho_b)^{\frac{1}{2m+n}}$. Using Bernoulli's inequality, we get:

$$\rho \geq \frac{\rho_b}{2m+n} \geq \frac{1}{2(2m+n)} \geq \frac{1}{4}\min\left\{\frac{1}{2m}, \frac{1}{n}\right\} \geq \frac{1}{4}\min\left\{\frac{a}{64\kappa}, \frac{1}{n}\right\} . \tag{142}$$

This finishes the proof for Corollary 16. ■

***A simpler result for* Kromagnon.** We also define a such that $\gamma = a/4L$. Theorem 15 tells us that:

$$\theta = \frac{\frac{4\kappa}{am} + \frac{a}{2}C_3(1 + \frac{\tau}{16\kappa})}{1 - \frac{a}{2}C_3(1 + \frac{\tau}{16\kappa})} \,. \tag{143}$$

We can once again upper bound θ by removing its denominator at a reasonable worst-case cost of a factor of 2:

$$\theta \leq \frac{8\kappa}{am} + aC_3(1 + \frac{\tau}{16\kappa}) \,. \tag{144}$$

Now, to enforce $\theta \leq 1/2$, we can choose $a \leq \frac{1}{4C_3(1+\frac{\tau}{16\kappa})}$ and $m = \frac{32\kappa}{a}$. We also obtain a rate factor per gradient computation of: $\rho \geq \frac{1}{4}\min\{\frac{a}{64\kappa}, \frac{1}{n}\}$. This finishes the proof of Corollary 17. ■

Speedup conditions. All we have to do now is to compare the rate factors of Svrg and Kromagnon in different regimes. Note that while our convergence result hold for any $a \leq 1/4$ Svrg (or the slightly more complex expression in the case of Kromagnon), the best step size (in terms of number of gradient computations) ensuring $\theta \leq \frac{1}{2}$ is the biggest allowable one – thus this is the one we use for our comparison.

Suppose we are in the "well-conditioned" regime where $n \geq \kappa$. The rate factor of Svrg is $\Omega(1/n)$. To make sure we have a linear speedup, we need the rate factor of Kromagnon to also be $\Omega(1/n)$, which means that:

$$\frac{1}{256\kappa C_3 + 16\tau C_3} = \Omega(\frac{1}{n}) \tag{145}$$

Recalling that $C_3 = 1 + 2\tau\sqrt{\Delta}$, we can rewrite (145) as:

$$\kappa = \mathcal{O}(n)\,; \quad \kappa\tau\sqrt{\Delta} = \mathcal{O}(n)\,; \quad \tau = \mathcal{O}(n)\,; \quad \tau^2\sqrt{\Delta} = \mathcal{O}(n)\,. \tag{146}$$

We can condense these conditions down to:

$$\tau = \mathcal{O}(\frac{n}{\kappa\sqrt{\Delta}})\,; \qquad \tau = \mathcal{O}(\sqrt{n\Delta^{-1/2}})\,. \tag{147}$$

Suppose now we are in the "ill-conditioned" regime, where $\kappa \geq n$. The rate factor of Svrg is now $\Omega(1/\kappa)$. To make sure we have a linear speedup, we need the rate factor of Kromagnon to also be $\Omega(1/\kappa)$, which means that:

$$\frac{1}{256\kappa C_3 + 16\tau C_3} = \Omega(\frac{1}{\kappa}) \tag{148}$$

We can derive the following sufficient conditions:

$$\tau = \mathcal{O}(\frac{1}{\sqrt{\Delta}})\,; \qquad \tau = \mathcal{O}(\kappa)\,. \tag{149}$$

Since $\kappa \geq n$, we obtain the conditions of Corollary 18 and thus finish its proof. ■

Appendix D. Hogwild – Proof of Theorem 22 and Corollary 23

D.1. Proof of Lemma 24 (suboptimality bound on $\mathbb{E}\|g_t\|^2$)

As was the case for proving Lemma 13 and Lemma 19, we simply introduce $f_i'(x^*)$ in g_t to derive our bound.

$$\begin{aligned}
\mathbb{E}\|g_t\|^2 = \mathbb{E}\|f_i'(\hat{x}_t)\|^2 &= \mathbb{E}\|f_i'(\hat{x}_t) - f_i'(x^*) + f_i'(x^*)\|^2 \\
&\leq 2\mathbb{E}\|f_i'(\hat{x}_t) - f_i'(x^*)\|^2 + 2\mathbb{E}\|f_i'(x^*)\|^2 \quad (\|a+b\|^2 \leq 2\|a\|^2 + 2\|b\|^2) \\
&\leq 4Le_t + 2\sigma^2\,. \qquad \text{(Hofmann et al. (2015), Eq (7) \& (8))}
\end{aligned}$$

■

D.2. Proof of Theorem 22 (convergence guarantee and rate of Hogwild)

Master inequality derivation. Once again, we plug Lemma 24 into Lemma 10 which gives us:

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + \gamma^2 C_1(4Le_t + 2\sigma^2) + \gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1}(4Le_u + 2\sigma^2) - 2\gamma e_t\,. \tag{150}$$

By grouping the e_t and the σ^2 terms we get our master inequality (48):

$$a_{t+1} \leq (1 - \frac{\gamma\mu}{2})a_t + (4L\gamma^2 C_1 - 2\gamma)e_t + 4L\gamma^2 C_2 \sum_{u=(t-\tau)_+}^{t-1} e_u + 2\gamma^2\sigma^2(C_1 + \tau C_2)\,.$$

Contraction inequality derivation (x_t). We now unroll (48) all the way to $t = 0$ to get:

$$\begin{aligned}
a_{t+1} \leq (1 - \frac{\gamma\mu}{2})^{t+1}a_0 &+ \sum_{u=0}^{t}(1 - \frac{\gamma\mu}{2})^{t-u}(4L\gamma^2 C_1 - 2\gamma)e_u \\
&+ \sum_{u=0}^{t}(1 - \frac{\gamma\mu}{2})^{t-u}4L\gamma^2 C_2 \sum_{v=(u-\tau)_+}^{u-1} e_v \\
&+ \sum_{u=0}^{t}(1 - \frac{\gamma\mu}{2})^{t-u}2\gamma^2\sigma^2(C_1 + \tau C_2)\,.
\end{aligned} \tag{151}$$

Now we can simplify these terms as follows:

$$\begin{aligned}
\sum_{u=0}^{t}(1 - \frac{\gamma\mu}{2})^{t-u} \sum_{v=(u-\tau)_+}^{u-1} e_v &= \sum_{v=0}^{t-1}\sum_{u=v+1}^{\min(t,v+\tau)}(1 - \frac{\gamma\mu}{2})^{t-u}e_v \\
&= \sum_{v=0}^{t-1}(1 - \frac{\gamma\mu}{2})^{t-v}e_v \sum_{u=v+1}^{\min(t,v+\tau)}(1 - \frac{\gamma\mu}{2})^{v-u} \\
&\leq \sum_{v=0}^{t-1}(1 - \frac{\gamma\mu}{2})^{t-v}e_v\tau(1 - \frac{\gamma\mu}{2})^{-\tau} \\
&\leq \tau(1 - \frac{\gamma\mu}{2})^{-\tau}\sum_{v=0}^{t}(1 - \frac{\gamma\mu}{2})^{t-v}e_v\,.
\end{aligned} \tag{152}$$

This $(1 - \frac{\gamma\mu}{2})^{-\tau}$ term is easily bounded, as we did in (107) for Asaga. Using Bernoulli's inequality (100), we get that if we assume $\tau \leq \frac{1}{\gamma\mu}$,[38]

$$(1 - \frac{\gamma\mu}{2})^{-\tau} \leq 2\,. \tag{153}$$

38. While this assumption on τ may appear restrictive, it is in fact weaker than the condition for a linear speed-up obtained by our analysis in Corollary 23.

We note that the last term in (151) is a geometric sum:

$$\sum_{u=0}^{t}(1-\frac{\gamma\mu}{2})^{t-u}\sigma^2 = \frac{2}{\gamma\mu}\sigma^2\,. \tag{154}$$

We plug (152)–(154) in (151) to obtain (49):

$$a_{t+1} \leq (1-\frac{\gamma\mu}{2})^{t+1}a_0 + (4L\gamma^2C_1 + 8L\gamma^2\tau C_2 - 2\gamma)\sum_{u=0}^{t}(1-\frac{\gamma\mu}{2})^{t-u}e_u + \frac{4\gamma\sigma^2}{\mu}(C_1+\tau C_2)\,.$$

Contraction inequality derivation ($\hat{x}_t$). We now have an contraction inequality for the convergence of x_t to x^*. However, since this quantity does not exist (except if we fix the number of iterations prior to running the algorithm and then wait for all iterations to be finished – an unwieldy solution), we rather want to prove that $\hat{x}_t$ converges to x^*. In order to do this, we use the simple following inequality:

$$\|\hat{x}_t - x^*\|^2 \leq 2a_t + 2\|\hat{x}_t - x_t\|^2\,. \tag{155}$$

We already have a contraction bound on the first term (49). For the second term, we combine (65) with Lemma 24 to get:

$$\mathbb{E}\|\hat{x}_t - x_t\|^2 \leq 4L\gamma^2C_1\sum_{u=(t-\tau)_+}^{t-1} e_u + 2\gamma^2\tau\sigma^2\,. \tag{156}$$

To make it easier to combine with (49), we rewrite (156) as:

$$\begin{aligned}\mathbb{E}\|\hat{x}_t - x_t\|^2 &\leq 4L\gamma^2C_1(1-\frac{\gamma\mu}{2})^{-\tau}\sum_{u=(t-\tau)_+}^{t-1}(1-\frac{\gamma\mu}{2})^{t-1-u}e_u + 2\gamma^2\tau\sigma^2\\ &\leq 8L\gamma^2C_1\sum_{u=(t-\tau)_+}^{t-1}(1-\frac{\gamma\mu}{2})^{t-1-u}e_u + 2\gamma^2\tau\sigma^2\\ &\leq 8L\gamma^2C_1\sum_{u=0}^{t-1}(1-\frac{\gamma\mu}{2})^{t-1-u}e_u + 2\gamma^2\tau\sigma^2\,.\end{aligned} \tag{157}$$

Combining (49) and (157) gives us (50):

$$\begin{aligned}\mathbb{E}\|\hat{x}_t - x^*\|^2 \leq (1-\frac{\gamma\mu}{2})^{t+1}2a_0 + (\frac{8\gamma(C_1+\tau C_2)}{\mu} + 4\gamma^2C_1\tau)\sigma^2\\ + (24L\gamma^2C_1 + 16L\gamma^2\tau C_2 - 4\gamma)\sum_{u=0}^{t}(1-\frac{\gamma\mu}{2})^{t-u}e_u\,.\end{aligned}$$

Maximum step size condition on γ. To prove Theorem 22, we need an inequality of the following type: $\mathbb{E}\|\hat{x}_t - x_t\|^2 \leq (1-\rho)^t a + b$. To give this form to Equation (50), we need

to remove all the $(e_u, u < t)$ terms from its right-hand side. To safely do so, we need to enforce that all these terms are negative, hence that:

$$24L\gamma^2C_1 + 16L\gamma^2\tau C_2 - 4\gamma \leq 0\,. \tag{158}$$

Plugging the values of C_1 and C_2 we get:

$$4L\mu\tau(1+\sqrt{\Delta}\tau)\gamma^2 + 6L(1+2\sqrt{\Delta}\tau)\gamma - 1 \leq 0\,. \tag{159}$$

As in our Asaga analysis, this reduces to a second-order polynomial sign condition. We remark again that since $\gamma \geq 0$, we can upper bound our terms in γ and γ^2 in this polynomial, which will still give us sufficient conditions for convergence. This means if we define $C_3 := 1 + 2\sqrt{\Delta}\tau$, a sufficient condition is:

$$4L\mu\tau C_3\gamma^2 + 6LC_3\gamma - 1 \leq 0\,. \tag{160}$$

The discriminant of this polynomial is always positive, so γ needs to be between its two roots. The smallest is negative, so the condition is not relevant to our case (where $\gamma > 0$). By solving analytically for the positive root ϕ, we get an upper bound condition on γ that can be used for any overlap τ and guarantee convergence. This positive root is:

$$\phi = \frac{3}{4}\frac{\sqrt{1+\frac{\mu\tau}{2LC_3}}-1)}{LC_3}\,. \tag{161}$$

We simplify it further by using (112):

$$\phi \geq \frac{3}{16LC_3\sqrt{1+\frac{\tau}{2\kappa C_3}}}\,. \tag{162}$$

This finishes the proof for Theorem 22. ■

D.3. Proof of Theorem 21 (convergence result for serial Sgd)

In order to analyze Corollary 23, we need to derive the maximum allowable step size for serial Sgd. Note that Sgd verifies a simpler contraction inequality than Lemma 10. For all $t \geq 0$:

$$a_{t+1} \leq (1-\gamma\mu)a_t + \gamma^2\mathbb{E}\|g_t\|^2 - 2\gamma e_t\,, \tag{163}$$

Here, the contraction factor is $(1-\gamma\mu)$ instead of $(1-\frac{\gamma\mu}{2})$ because $\hat{x}_t = x_t$ so there is no need for a triangle inequality to get back $\|x_t - x^*\|^2$ from $\|\hat{x}_t - x^*\|^2$ after we apply our strong convexity bound in our initial recursive inequality (see Section B.1). Lemma 24 also holds for serial Sgd. By plugging it into (163), we get:

$$a_{t+1} \leq (1-\gamma\mu)a_t + (4L\gamma^2 - 2\gamma)e_t + 2\gamma^2\sigma^2\,. \tag{164}$$

We then unroll (164) until $t = 0$ to get:

$$a_{t+1} \leq (1-\gamma\mu)^{t+1}a_0 + (4L\gamma^2 - 2\gamma)\sum_{u=0}^{t}(1-\gamma\mu)^{t-u}e_u + 2\frac{\gamma\sigma^2}{\mu}\,. \tag{165}$$

To get linear convergence up to a ball around the optimum, we need to remove the $(e_u, 0 \leq u \leq t)$ terms from the right-hand side of the equation. To safely do this, we need these terms to be negative, i.e. $4L\gamma^2 - 2\gamma \leq 0$. We can then trivially derive the condition on γ to achieve linear convergence: $\gamma \leq \frac{1}{2L}$.

We see that if $\gamma = a/L$ with $a \leq 1/2$, SGD converges at a geometric rate of at least: $\rho(a) = a/\kappa$, up to a ball of radius $2\frac{\gamma\sigma^2}{\mu}$ around the optimum. Now, to make sure we reach ϵ-accuracy, we need $\frac{2\gamma\sigma^2}{\mu} \leq \epsilon$, i.e. $\gamma \leq \frac{\epsilon\mu}{2\sigma^2}$. All told, in order to get linear convergence to ϵ-accuracy, serial SGD requires $\gamma \leq \min\left\{\frac{1}{2L}, \frac{\epsilon\mu}{2\sigma^2}\right\}$.

D.4. Proof of Corollary 23 (speedup regimes for HOGWILD)

The convergence rate of both SGD and HOGWILD is directly proportional to the step size. Thus, in order to make sure HOGWILD is linearly faster than SGD for any reasonable step size, we need to show that the maximum allowable step size ensuring linear convergence for HOGWILD – given in Theorem 22 – is of the same order as the one for SGD, $\mathcal{O}(1/L)$. Recalling that $\gamma = \frac{a}{L}$, we get the following sufficient condition: $a^*(\tau) = \mathcal{O}(1)$.

Given (46), the definition of $a^*(\tau)$, we require both:

$$\tau\sqrt{\Delta} = \mathcal{O}(1)\,; \qquad \sqrt{1 + \frac{1}{2\kappa}\min\{\frac{1}{\sqrt{\Delta}}, \tau\}} = \mathcal{O}(1)\,. \tag{166}$$

This gives us the final condition on τ for a linear speedup: $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}}, \kappa\})$.

To finish the proof of Corollary 23, we only have to show that under this condition, the size of the ball is of the same order regardless of the algorithm used.

Using $\gamma\mu\tau \leq 1$ and $\tau \leq \frac{1}{\sqrt{\Delta}}$, we get that $(\frac{8\gamma(C_1+\tau C_2)}{\mu} + 4\gamma^2 C_1\tau)\sigma^2 = \mathcal{O}(\frac{\gamma\sigma^2}{\mu})$, which finishes the proof of Corollary 23. Note that these two conditions are weaker than $\tau = \mathcal{O}(\min\{\frac{1}{\sqrt{\Delta}}, \kappa\})$, which allows us to get better bounds in case we want to reach ϵ-accuracy with $\frac{\epsilon\mu}{\sigma^2} \ll \frac{1}{2L}$. ■

Appendix E. On the Difficulty of Parallel Lagged Updates

In the implementation presented in Schmidt et al. (2016), the dense part ($\bar{\alpha}$) of the updates is deferred. Instead of writing dense updates, counters c_d are kept for each coordinate of the parameter vector – which represent the last time these variables were updated – as well as the average gradient $\bar{\alpha}$ for each coordinate. Then, whenever a component $[\hat{x}]_d$ is needed (in order to compute a new gradient), we subtract $\gamma(t - c_d)[\bar{\alpha}]_d$ from it and c_d is set to t. It is possible to do this without modifying the algorithm because $[\bar{\alpha}]_d$ only changes when $[\hat{x}]_d$ also does.

In the sequential setting, this results in the same iterations as performing the updates in a dense fashion, since the coordinates are only stale when they are not used. Note that at the end of an execution all counters have to be subtracted at once to get the true final parameter vector (and to bring every c_d counter to the final t).

In the parallel setting, several issues arise:

- two cores might be attempting to correct the lag at the same time. In which case since updates are done as additions and not replacements (which is necessary to ensure that there are no overwrites), the lag might be corrected multiple times, i.e. overly corrected.

- we would have to read and write atomically to each $[\hat{x}_d], c_d, [\bar{\alpha}]_d$ triplet, which is highly impractical.

- we would need to have an explicit global counter, which we do not in ASAGA (our global counter t being used solely for the proof).

- in the dense setting, updates happen coordinate by coordinate. So at time t the number of $\bar{\alpha}$ updates a coordinate has received from a fixed past time c_d is a random variable, which may differs from coordinate to coordinate. Whereas in the lagged implementation, the multiplier is always $(t - c_d)$ which is a constant (conditional to c_d), which means a potentially different $\hat{x}_t$.

- the trick used in Reddi et al. (2015) for asynchronous parallel SVRG does not apply here because it relies on the fact that the "reference" gradient term in SVRG is constant throughout a whole epoch, which is not the case for SAGA.

All these points mean both that the implementation of such a scheme in the parallel setting would be impractical, and that it would actually yields a different algorithm than the dense version, which would be even harder to analyze. This is confirmed by Pan et al. (2016), where the authors tried to implement a parallel version of the lagged updates scheme and had to alter the algorithm to succeed, obtaining an algorithm with suboptimal performance as a result.

Appendix F. Additional Empirical Details

F.1. Detailed Description of Data Sets

We run our experiments on four data sets. In every case, we run logistic regression for the purpose of binary classification.

RCV1 *($n = 697,641$, $d = 47,236$).* The first is the Reuters Corpus Volume I (RCV1) data set (Lewis et al., 2004), an archive of over 800,000 manually categorized newswire stories made available by Reuters, Ltd. for research purposes. The associated task is a binary text categorization.

URL *($n = 2,396,130$, $d = 3,231,961$).* Our second data set was first introduced in Ma et al. (2009). Its associated task is a binary malicious URL detection. This data set contains more than 2 million URLs obtained at random from Yahoo's directory listing (for the "benign" URLs) and from a large Web mail provider (for the "malicious" URLs). The benign to malicious ratio is 2. Features include lexical information as well as metadata. This data set was obtained from the libsvmtools project.[39]

39. http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary.html

Covertype *($n = 581,012$, $d = 54$).* On our third data set, the associated task is a binary classification problem (down from 7 classes originally, following the pre-treatment of Collobert et al., 2002). The features are cartographic variables. Contrarily to the first two, this is a dense data set.

Realsim *($n = 73,218$, $d = 20,958$).* We only use our fourth data set for non-parallel experiments and a specific compare-and-swap test. It constitutes of UseNet articles taken from four discussion groups (simulated auto racing, simulated aviation, real autos, real aviation).

F.2. Implementation Details

Hardware. All experiments were run on a Dell PowerEdge 920 machine with 4 Intel Xeon E7-4830v2 processors with 10 2.2GHz cores each and 384GB 1600 MHz RAM.

Software. All algorithms were implemented in the Scala language and the software stack consisted of a Linux operating system running Scala 2.11.7 and Java 1.6.

We chose this expressive, high level language for our experimentation despite its typical 20x slower performance compared to C because our primary concern was that the code may easily be reused and extended for research purposes (which is harder to achieve with low level, heavily optimized C code; especially for error prone parallel computing).

As a result our timed experiments exhibit sub-optimal running times, e.g. compared to Konečný and Richtárik (2013). This is as we expected. The observed slowdown is both consistent across data sets (roughly 20x) and with other papers that use Scala code (e.g. Mania et al. 2017, Ma et al. 2015, Fig. 2).

Despite this slowdown, our experiments show state-of-the-art results in convergence per number of iterations. Furthermore, the speed-up patterns that we observe for our implementation of Hogwild and Kromagnon are similar to the ones given in Mania et al. (2017); Niu et al. (2011); Reddi et al. (2015) (in various languages).

The code we used to run all the experiments is available at `http://www.di.ens.fr/sierra/research/asaga/`.

Necessity of compare-and-swap operations. Interestingly, we have found necessary to use compare-and-swap instructions in the implementation of Asaga. In Figure 7, we display suboptimality plots using non-thread safe operations and compare-and-swap (CAS) operations. The non-thread safe version starts faster but then fails to converge beyond a specific level of suboptimality, while the compare-and-swap version does converges linearly up to machine precision.

For *compare-and-swap* instructions we used the `AtomicDoubleArray` class from the Google library `Guava`. This class uses an `AtomicLongArray` under the hood (from package `java.util.concurrent.atomic` in the standard Java library), which does indeed benefit from lower-level CPU-optimized instructions.

Efficient storage of the α_i. Storing n gradient may seem like an expensive proposition, but for linear predictor models, one can actually store a single scalar per gradient (as proposed in Schmidt et al., 2016), which is what we do in our implementation of Asaga.

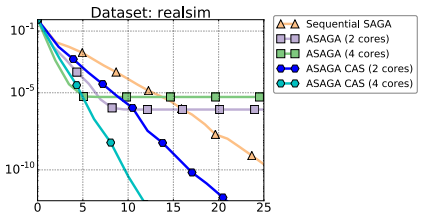


Figure 7: **Compare and swap in the implementation of Asaga.** Suboptimality as a function of time for Asaga, both using compare-and-swap (CAS) operations and using standard operations. The graph reveals that CAS is indeed needed in a practical implementation to ensure convergence to a high precision.

F.3. Biased Update in the Implementation

In the implementation detailed in Algorithm 2, $\bar{\alpha}$ is maintained in memory instead of being recomputed for every iteration. This saves both the cost of reading every data point for each iteration and of computing $\bar{\alpha}$ for each iteration.

However, this removes the unbiasedness guarantee. The problem here is the definition of the expectation of $\hat{\alpha}_i$. Since we are sampling uniformly at random, the average of the $\hat{\alpha}_i$ is taken at the precise moment when we read the α_i^t components. Without synchronization, between two reads to a single coordinate in α_i and in $\bar{\alpha}$, new updates might arrive in $\bar{\alpha}$ that are not yet taken into account in α_i. Conversely, writes to a component of α_i might precede the corresponding write in $\bar{\alpha}$ and induce another source of bias.

In order to alleviate this issue, we can use coordinate-level locks on α_i and $\bar{\alpha}$ to make sure they are always synchronized. Such low-level locks are quite inexpensive when d is large, especially when compared to vector-wide locks.

However, as previously noted, experimental results indicate that this fix is not necessary.

References

Dimitri P. Bertsekas and John N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Prentice Hall, 1989.

R. Collobert, S. Bengio, and Y. Bengio. A parallel mixture of SVMs for very large scale problems. *Neural Computation*, 2002.

Christopher De Sa, Ce Zhang, Kunle Olukotun, and Christopher Ré. Taming the wild: A unified analysis of Hogwild!-style algorithms. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

Aaron Defazio, Francis Bach, and Simon Lacoste-Julien. SAGA: A fast incremental gradient method with support for non-strongly convex composite objectives. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

John C. Duchi, Sorathan Chaturapruek, and Christopher Ré. Asynchronous stochastic convex optimization. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

Thomas Hofmann, Aurelien Lucchi, Simon Lacoste-Julien, and Brian McWilliams. Variance reduced stochastic gradient descent with neighbors. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

Cho-Jui Hsieh, Hsiang-Fu Yu, and Inderjit Dhillon. PASSCoDe: Parallel asynchronous stochastic dual co-ordinate descent. In *Proceedings of the* 32^{nd} *International Conference on Machine Learning (ICML)*, 2015.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. In *Advances in Neural Information Processing Systems 26 (NIPS)*, 2013.

Jakub Konečný and Peter Richtárik. Semi-stochastic gradient descent methods. *arXiv:1312.1666*, 2013.

Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Advances in Neural Information Processing Systems 25 (NIPS)*, 2012.

Rémi Leblond, Fabian Pedregosa, and Simon Lacoste-Julien. ASAGA: Asynchronous parallel SAGA. In *Proceedings of the* 20^{th} *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. RCV1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 2004.

Xiangru Lian, Yijun Huang, Yuncheng Li, and Ji Liu. Asynchronous parallel stochastic gradient for nonconvex optimization. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

Ji Liu, Stephen J. Wright, Christopher Ré, Victor Bittorf, and Srikrishna Sridhar. An asynchronous parallel stochastic coordinate descent algorithm. *Journal of Machine Learning Research*, 2015.

Chenxin Ma, Virginia Smith, Martin Jaggi, Michael I. Jordan, Peter Richtarik, and Martin Takac. Adding vs. averaging in distributed primal-dual optimization. In *Proceedings of the* 32^{nd} *International Conference on Machine Learning (ICML)*, 2015.

Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Identifying suspicious URLs: an application of large-scale online learning. In *Proceedings of the* 26^{th} *International Conference on Machine Learning (ICML)*, 2009.

Horia Mania, Xinghao Pan, Dimitris Papailiopoulos, Benjamin Recht, Kannan Ramchandran, and Michael I. Jordan. Perturbed iterate analysis for asynchronous stochastic optimization. *SIAM Journal on Optimization*, 2017.

Eric Moulines and Francis R. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011.

Deanna Needell, Rachel Ward, and Nati Srebro. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. In *Advances in Neural Information Processing Systems 27 (NIPS)*, 2014.

Feng Niu, Benjamin Recht, Christopher Re, and Stephen Wright. Hogwild: A lock-free approach to parallelizing stochastic gradient descent. In *Advances in Neural Information Processing Systems 24 (NIPS)*, 2011.

Xinghao Pan, Maximilian Lam, Stephen Tu, Dimitris Papailiopoulos, Ce Zhang, Michael I. Jordan, Kannan Ramchandran, Chris Re, and Benjamin Recht. Cyclades: Conflict-free asynchronous machine learning. In *Advances in Neural Information Processing Systems 29 (NIPS)*, 2016.

Fabian Pedregosa, Rémi Leblond, and Simon Lacoste-Julien. Breaking the nonsmooth barrier: A scalable parallel method for composite optimization. In *Advances in Neural Information Processing Systems 30 (NIPS)*, 2017.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczos, and Alex Smola. On variance reduction in stochastic gradient descent and its asynchronous variants. In *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015.

Sashank J. Reddi, Ahmed Hefny, Suvrit Sra, Barnabás Póczós, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *Proceedings of the* 33^{rd} *International Conference on Machine Learning (ICML)*, 2016.

M. Schmidt, N. Le Roux, and F. Bach. Minimizing finite sums with the stochastic average gradient. *Mathematical Programming*, 2016.

Mark Schmidt. Convergence rate of stochastic gradient with constant step size. *UBC Technical Report*, 2014.

Shai Shalev-Shwartz and Tong Zhang. Stochastic dual coordinate ascent methods for regularized loss. *Journal of Machine Learning Research*, 2013.

Shen-Yi Zhao and Wu-Jun Li. Fast asynchronous parallel stochastic gradient descent. In *Proceedings of the* 30^{th} *AAAI Conference on Artificial Intelligence*, 2016.

Journal of Machine Learning Research 19 (2018) 1-30 Submitted 2/18; Revised 9/18; Published 11/18

Clustering is semidefinitely not that hard: Nonnegative SDP for manifold disentangling

Mariano Tepper MTEPPER@FLATIRONINSTITUTE.ORG
Flatiron Institute, Simons Foundation, NY, USA

Anirvan M. Sengupta ANIRVANS@PHYSICS.RUTGERS.EDU
Dept. of Physics and Astronomy, Rutgers University, NJ, USA
Flatiron Institute, Simons Foundation, NY, USA

Dmitri Chklovskii DCHKLOVSKII@FLATIRONINSTITUTE.ORG
Flatiron Institute, Simons Foundation, NY, USA
NYU Langone Medical Center, New York, NY

Editor: Daniel Lee

Abstract

In solving hard computational problems, semidefinite program (SDP) relaxations often play an important role because they come with a guarantee of optimality. Here, we focus on a popular semidefinite relaxation of K-means clustering which yields the same solution as the non-convex original formulation for well segregated datasets. We report an unexpected finding: when data contains (greater than zero-dimensional) manifolds, the SDP solution captures such geometrical structures. Unlike traditional manifold embedding techniques, our approach does not rely on manually defining a kernel but rather enforces locality via a nonnegativity constraint. We thus call our approach NOnnegative MAnifold Disentangling, or NOMAD. To build an intuitive understanding of its manifold learning capabilities, we develop a theoretical analysis of NOMAD on idealized datasets. While NOMAD is convex and the globally optimal solution can be found by generic SDP solvers with polynomial time complexity, they are too slow for modern datasets. To address this problem, we analyze a non-convex heuristic and present a new, convex and yet efficient, algorithm, based on the conditional gradient method. Our results render NOMAD a versatile, understandable, and powerful tool for manifold learning.

Keywords: K-means, semidefinite programming, manifolds, conditional gradient method

1. Introduction

In the quest for an algorithmic theory of biological neural networks, some of the authors have recently proposed a soft K-means clustering network that may model insect olfactory processing and other computations (Pehlevan et al., 2017). This network was derived by performing online optimization on the non-convex K-means objective function. Whereas the network dynamics and learning rules are biologically plausible, the nonconvexity of the objective makes it difficult to analyze the solutions and algorithm convergence.

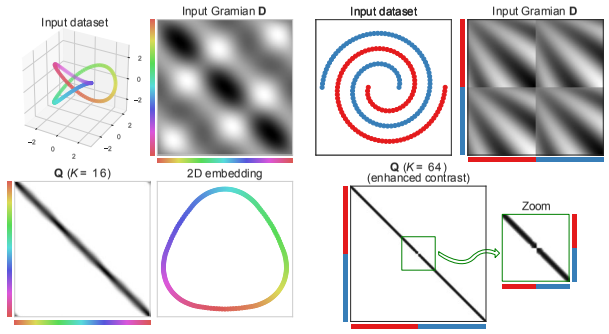
Here, to understand the solutions computed by the clustering neural network, we consider a convex SDP relaxation of K-means (Kulis et al., 2007; Peng and Wei, 2007; Awasthi et al., 2015). Given data points $\{\mathbf{x}_i\}_{i=1}^n$ we define the Gramian matrix, $\mathbf{D}$, such that $(\mathbf{D})_{ij} = {\mathbf{x}_i}^\top \mathbf{x}_j$. Then, we search for a cluster co-association matrix $\mathbf{Q}$, such that $(\mathbf{Q})_{ij} = 1$ if points i and j belong to the same cluster and $(\mathbf{Q})_{ij} = 0$ if they do not. The optimum $\mathbf{Q}_*$ can be found by solving the following optimization problem (the acronym will be explained below):

$$\mathbf{Q}_* = \underset{\mathbf{Q}\in\mathbb{R}^{n\times n}}{\operatorname{argmax}} \operatorname{Tr}(\mathbf{DQ}) \quad \text{s.t.} \quad \mathbf{Q1} = \mathbf{1},\ \operatorname{Tr}(\mathbf{Q}) = K,\ \mathbf{Q} \succeq 0,\ \mathbf{Q} \geq \mathbf{0}. \tag{NOMAD}$$

Its link with the original K-means clustering formulation is explained in Appendix A.[1]

First, we focus on the question: what does NOMAD compute? Until now, theoretical efforts have concentrated on showing that NOMAD is a good surrogate for K-means. Awasthi et al. (2015) study its solutions on datasets consisting of linearly separable clusters and demonstrate that they reproduce hard-clustering assignments of K-means. Moreover, the solution to NOMAD achieves hard clustering even for some datasets on which Lloyd's algorithm (Lloyd, 1982) fails (i.e., Iguchi et al. (2015); Mixon et al. (2016)). Related problems have been studied by Amini and Levina (2014); Javanmard et al. (2015); Yu et al. (2012).

In this work, we analyze NOMAD in a different regime than previous studies. Instead of focusing on cases and parameter settings where it approximates the original K-means formulation, we concentrate on alternative settings and discover that NOMAD is not merely a convex K-means imitator. NOMAD finds the manifold structure in the data, even discriminating different manifolds. Fig. 1 shows two examples of this unexpected behavior where NOMAD dissects the geometry of



(a) (b)

Figure 1: NOMAD, originally introduced as a convex relaxation of K-means clustering, surprisingly learns manifold structures in the data. (a) Learning the manifold of a trefoil knot, which cannot be "untied" in 3D without cutting it. NOMAD understands that this is a closed manifold, yielding a circulant matrix $\mathbf{Q}$, which can be "unfolded" in 2D. (b) Learning multiple manifolds with NOMAD. Although they are linearly non-separable, NOMAD correctly finds two submatrices, one for each manifold (for visual clarity, we enhance the contrast of $\mathbf{Q}$).

1. **Notation.** $(\mathbf{X})_{ij}$, $(\mathbf{X})_{:j}$, $(\mathbf{X})_{i:}$ denote the (i,j)-th entry of matrix $\mathbf{X}$, the j-th column of $\mathbf{X}$, and the i-th row of $\mathbf{X}$, respectively. For vectors, we employ lowercase and we use a similar notation but with a single index. We write $\mathbf{X} \geq \mathbf{0}$ if a matrix $\mathbf{X}$ is entry-wise nonnegative and $\mathbf{X} \succeq \mathbf{0}$ if it is positive semidefinite.

the data. Because of this and of the central role played by the nonnegativity constraint in the SDP we call it a NOnnegative MAnifold Disentangling (NOMAD).

The next question is: how can we compute these solutions? Despite the theoretical advantages of convex optimization, in practice, the use of SDPs for clustering has remained limited. This is mainly due to the lack of efficient algorithms to solve the convex optimization problem. We address this issue by presenting an efficient convex solver for NOMAD, based on the conditional gradient method. The new algorithm can handle large datasets, extending the applicability of NOMAD to more interesting and challenging scenarios.

Organization. We first study the behavior of NOMAD theoretically by analyzing its solution for a simple synthetic example of a regular manifold with symmetry (Sec. 2). In this context, we demonstrate how NOMAD departs from standard K-means. Building on this analysis, we suggest that NOMAD has non-trivial manifold learning capabilities (Sec. 3) and demonstrate numerically NOMAD's good performance in non-trivial examples, including synthetic and real datasets. Then, motivated by the relatively slow performance of standard SDP solvers, we focus on scaling NOMAD to large modern datasets. In Sec. 4, we study both theoretically and experimentally an heuristic non-convex Burer-Monteiro-style algorithm (Kulis et al., 2007). Finally, we present a new convex and yet efficient algorithm for NOMAD. This algorithm allows us, for the first time, to study provable solutions of NOMAD on large datasets. Our software is publicly available at `https://github.com/simonsfoundation/sdp_kmeans`.

2. Theoretical analysis of manifold learning capabilities of NOMAD

Starting with the appearance of Isomap (Tenenbaum et al., 2000) and locally-linear embedding (LLE) (Roweis and Saul, 2000), there has been outstanding progress in the area of manifold learning (e.g., Belkin and Niyogi, 2003; Hadsell et al., 2006; Weinberger and Saul, 2006; Weiss et al., 2008). For a data matrix $\mathbf{X} = [\mathbf{x}_i]_{i=1}^n$ of column-vectors/points $\mathbf{x}_i \in \mathbb{R}^d$, the majority of these modern methods have three steps:

1. Determine the neighbors of each point. This can be done in two ways: (1) keep all point within some fixed radius ρ or (2) compute κ nearest neighbors.
2. Construct a weighting matrix $\mathbf{W}$, where $(\mathbf{W})_{ij} = 0$ if points i and j are not neighbors, and $(\mathbf{W})_{ij}$ is inversely proportional to the distance between points i and j otherwise.
3. Compute an embedding from $\mathbf{W}$ that is locally isometric to $\mathbf{X}$.

For the third step, many different and powerful approaches have been proposed, from computing shortest paths on a graph (Tenenbaum et al., 2000), to using graph spectral methods (Belkin and Niyogi, 2003), to using neural networks (Hadsell et al., 2006).

However, the success of these techniques depends critically on the ability to capture the data structure in the first two steps. Correctly setting either ρ or κ is a non-trivial task that is left to the user of these techniques. Furthermore, a kernel (most commonly an RBF) is often involved in the second step, adding an additional parameter (the kernel width/scale) to the user to determine. Expectedly, the optimal selection of these parameters plays a critical role in the overall success of the manifold learning process.

NOMAD departs drastically from this setup as no kernel selection nor nearest neighbor search are involved. Yet, the solution $\mathbf{Q}_*$ is effectively a kernel which is automatically learned from the data. Because $\mathbf{Q}_*$ is positive semidefinite it can be factorized as $\mathbf{Q}_* = \mathbf{Y}^\top\mathbf{Y}$, defining a feature map from $\mathbf{X}$ to $\mathbf{Y}$.

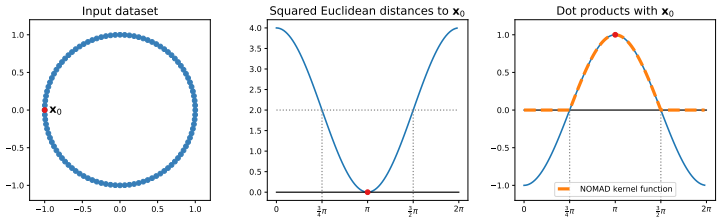


Figure 2: Correspondence between using a kernel/threshold in distance-space and nonnegativity of Gramian-based representations. In this toy example, the constraint $\mathbf{Q} \geq 0$ in NOMAD is equivalent to setting to zero distances that are greater than $\sqrt{2}$ (squared distances greater than 2). We use $\mathbf{x}_0$ as a reference but rotational symmetry makes this argument valid for all points in the dataset.

To illustrate intuitively the differences and similarities with prior work on manifold learning we use LLE (Roweis and Saul, 2000) as an example. LLE optimizes the cost function

$$\Phi(\mathbf{Y}) = \operatorname{Tr}\left((\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})\left(\mathbf{Y}^\top\mathbf{Y}\right)\right), \tag{1}$$

where $\mathbf{W}$ is the adjacency matrix of a weighted nearest-neighbors graph. The key to finding a matrix $\mathbf{Y}$ that is locally isometric to $\mathbf{X}$, while unwrapping the data manifold, is to remove from $\mathbf{W}$ the connections between distant points $(\mathbf{X})_{:i}$ and $(\mathbf{X})_{:j}$. This is done with some technique to find nearest neighbors.

NOMAD also tries to align the output Gramian, $\mathbf{Q}$, to the input Gramian, $\mathbf{D}$, but discards distant data points differently. As negative entries in $\mathbf{D}$ cannot be matched because $\mathbf{Q}$ is nonnegative, the best option would be to set the corresponding element of $\mathbf{Q}$ to zero. This effectively discards pairs of input data points whose inner product is negative thus enforcing locality in the angular space (Cho and Saul, 2009), see Fig. 2. In fact, this argument can be taken further by noting that the constraint $\mathbf{Q1} = \mathbf{1}$ allows us to replace the Gramian $\mathbf{D}$ with the negative squared distance matrix,

$$-\tfrac{1}{2}\sum_{ij}\|\mathbf{x}_i-\mathbf{x}_j\|_2^2\,(\mathbf{Q})_{ij} = -\sum_i(\mathbf{D})_{ii}\sum_j(\mathbf{Q})_{ij} + \operatorname{Tr}(\mathbf{DQ}) = -\operatorname{Tr}(\mathbf{D}) + \operatorname{Tr}(\mathbf{DQ}). \tag{2}$$

Finally, the constraint $\operatorname{Tr}(\mathbf{Q}) = K$ allows further control of the neighborhood size of NOMAD (modulating the actual width of its kernel function, see Fig. 2). Next, we develop further intuition about the manifold-learning capabilities of NOMAD by analyzing theoretically the dataset in Fig. 2.

As we mentioned before, the SDP formulation of Peng and Wei (2007) was developed as a clustering algorithm. Whether this method actually delivers a clustered solution depends on the geometry of the dataset. When the dataset consists of well-segregated clusters, the resulting $\mathbf{Q}_*$ has block diagonal structure. We empirically observe that, when the dataset is sampled from a regular manifold, the solution $\mathbf{Q}_*$ does not break down the dataset into artificial clusters and actually preserves the manifold structure (see Sec. 3). In a simple example, where the manifold exhibits a high degree of symmetry, we demonstrate analytically that this behavior occurs. The following sections are devoted to this task.

2.1. Analysis of NOMAD on a 2D ring dataset

We analyze the case in which the input data to NOMAD possess rotational symmetry, i.e., data are arranged uniformly on a ring, see Fig. 2. In this case, we can write the SDP as a linear program (LP) in the circular Fourier basis. This new representation allows to visualize that NOMAD lifts the data into a high-dimensional space, with K controlling its dimensionality.

In the example in Fig. 2, the entries of $\mathbf{D}$ can be described by $(\mathbf{D})_{ij} = \mathbf{x}_i{}^\top \mathbf{x}_j = \cos(\alpha_i - \alpha_j)$, where α_i, α_j are the angles of points $\mathbf{x}_i, \mathbf{x}_j$, respectively (Fig. 2). Since the points are uniformly distributed over the ring, $\mathbf{D}$ is a circulant matrix, i.e., $\cos(\alpha_i - \alpha_j) = \cos(\alpha_{i+k} - \alpha_{k+k})$. The solution $\mathbf{Q}_*$ to NOMAD is circulant too (Bachoc et al., 2012). Being circulant matrices, $\mathbf{D}$ and $\mathbf{Q}_*$ are diagonalized by the discrete Fourier transform (DFT), i.e.,

$$\mathbf{D} = \mathbf{F}\,\mathrm{diag}(\mathbf{d})\mathbf{F}^{\mathrm{H}}, \qquad \mathbf{Q}_* = \mathbf{F}\,\mathrm{diag}(\mathbf{q})\mathbf{F}^{\mathrm{H}}, \tag{3}$$

where $\mathbf{q}, \mathbf{d} \geq \mathbf{0}$ respectively are vectors containing the eigenvalues of $\mathbf{D}$ and $\mathbf{Q}_*$, $\mathbf{F}^{\mathrm{H}}$ is a Hermitian conjugate of $\mathbf{F}$, and $\mathbf{F} \in \mathbb{C}^{n\times n}$ is the unitary DFT matrix, with entries $(p, k = 0, \ldots, n-1)$

$$(\mathbf{F})_{pk} = \tfrac{1}{\sqrt{n}} \exp\left(-i2\pi p \tfrac{k}{n}\right). \tag{4}$$

Hence, and in accord with the constraint $\mathbf{Q}_*\mathbf{1} = \mathbf{1}$, we have that $(\mathbf{F})_{0:} = \frac{1}{\sqrt{n}}\mathbf{1}$ and $(\mathbf{q})_0 = 1$.

2.2. A linear program on the data manifold

We express the objective function and the constraints of NOMAD in terms of $\mathbf{d}$ and $\mathbf{q}$, i.e.,

$$\mathrm{Tr}\,(\mathbf{D}\mathbf{Q}_*) = \mathbf{d}^\top \mathbf{q}, \tag{5}$$

$$\mathrm{Tr}\,(\mathbf{Q}) = \mathbf{1}^\top \mathbf{q} = K, \tag{6}$$

$$(\mathbf{Q})_{kk'} = (\mathbf{F})_{k:}\,\mathrm{diag}(\mathbf{q})(\mathbf{F}^{\mathrm{H}})_{:k'} = \sum_{p=0}^{n-1} \frac{(\mathbf{q})_p}{n} \cos\left(2\pi p \frac{k'-k}{n}\right) \geq 0. \tag{7}$$

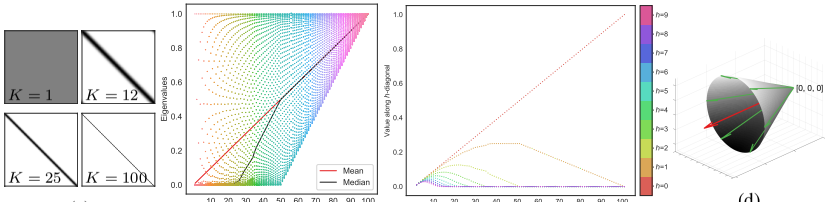
This reformulation allows us to rewrite NOMAD as a linear program

$$\max_{\mathbf{q}} \mathbf{d}^\top \mathbf{q} \quad \text{s.t.} \quad (\forall \tau)\ \mathbf{c}_\tau{}^\top \mathbf{q} \geq 0, \quad \mathbf{1}^\top \mathbf{q} = K, \quad \mathbf{q} \geq \mathbf{0}, \quad (\mathbf{q})_0 = 1, \tag{8}$$

where $(\mathbf{c}_\tau)_p = \frac{1}{n}\cos\left(2\pi p \frac{\tau}{n}\right)$.

Problem (8) sheds light on the inner workings of NOMAD. First, the constraint $\mathbf{1}^\top\mathbf{q} = K$ ensures that $\mathbf{q}$ does not grow to infinity and acts as a budget constraint. Let us assume for a moment that we remove the constraint $\mathbf{c}_\tau{}^\top\mathbf{q} \geq 0$ (the equivalent of $\mathbf{Q} \geq \mathbf{0}$). Then, the program will try to set to K the entry of $\mathbf{q}$ corresponding to the largest eigenvalue of $\mathbf{d}$; this $\mathbf{q}$ will violate as K gets bigger the removed constraint (since $(\mathbf{c}_\tau)_p$ is a sinusoid). Then the effect of this constraint is to spread the allocated budget among several eigenvalues (instead of just the largest). The experiment in Fig. 3 confirms this: the number of active eigenvalues of $\mathbf{Q}_*$ grows with K. We can interpret this as increasing the intrinsic dimensionality of the problem in such a way that only local interactions are considered.

Interpretation of K. The circulant property of $\mathbf{Q}_*$ for the 2D ring sheds further light on the meaning of K. In Fig. 3(c), we observe that the number of significant elements in each of $\mathbf{Q}_*$ is $\lceil n/K \rceil$.



(a) (b) (c) (d)

Figure 3: Evolution of the NOMAD solution for the 2D ring dataset (with 100 points, see Fig. 2) with increasing parameter K. (a) As K increases, the solution, $\mathbf{Q}_*$, concentrates more and more towards the diagonal. (b) As K increases, the number of active eigenvalues in the solution, $\mathbf{Q}_*$, grows resulting in the more uniform distribution of eigenvalues and greater mean/median (notice that the mean being linear comes from the trace constraint). (c) We define the h-diagonal of $\mathbf{Q}_*$ as the entries (i, j) for which $i - j = h$. As $\mathbf{Q}_*$ is a circulant matrix, each h-diagonal contains a single repeated value. We plot these values, assigning a different color to each h. The effect of the scaling constraint $\mathrm{Tr}\,(\mathbf{Q}) = K$ becomes evident: when one h-diagonal becomes inactive, all remaining h'-diagonals need to be upscaled. (d) The eigenvectors of $\mathbf{Q}_*$ form a high-dimensional cone (cartoon representation, cone axis in red and eigenvectors in green).

Thus, we can interpret K as a parameter that effectively sets the size of the local neighborhood on the manifold. In standard manifold learning methods this size is set by a combination of the number of nearest neighbors and the shape and scale of the kernel function. In NOMAD, all these variables are incorporated into a single parameter and balanced with the help of the remaining problem constraints.

In general, for non-symmetric and irregularly sampled manifolds, K is chosen to capture the manifold underlying the dataset: the neighborhood size needs to be small enough to capture the desired manifold features, but big enough to avoid capturing unwanted structure (e.g., noise). If sampling density differs in different areas, the size will adjust locally as needed.

2.3. Lifting the ring to a high-dimensional cone

Here, we show that NOMAD effectively embeds the data manifold into a space where its structure, i.e., rotational symmetry, is preserved. We now make use of the half-wave symmetry in $\mathbf{Q}_*$, noting that they can be fully represented with only one half of the Fourier basis. We can then decompose it with the real Fourier basis

$$\mathbf{Q}_* = \tilde{\mathbf{F}}\,\mathrm{diag}(\tilde{\mathbf{q}})\tilde{\mathbf{F}}^\top, \tag{9}$$

where $\tilde{\mathbf{q}} = [(\mathbf{q})_0, (\mathbf{q})_1, (\mathbf{q})_1, \ldots, (\mathbf{q})_{n-1}, (\mathbf{q})_{n-1}]^\top$ and $\tilde{\mathbf{F}} \in \mathbb{R}^{n\times n}$ has entries $(p, k = 0, \ldots, n-1)$

$$(\tilde{\mathbf{F}})_{pk} = \begin{cases} \frac{2}{\sqrt{n}} \cos\left(2\pi p \frac{k}{n}\right) & \text{if } k \text{ is even,} \\ \frac{2}{\sqrt{n}} \sin\left(2\pi p \frac{k-1}{n}\right) & \text{if } k \text{ is odd.} \end{cases} \tag{10}$$

Let $\tilde{\mathbf{Y}} = \mathrm{diag}(\mathbf{q})^{1/2}\tilde{\mathbf{F}}^\top$. Notice that $\langle \tilde{\mathbf{Y}}_{:i} / \left\|\tilde{\mathbf{Y}}_{:i}\right\|_F, \tilde{\mathbf{F}}_{:0}\rangle = \langle \tilde{\mathbf{F}}_{:i}, \tilde{\mathbf{F}}_{:0}\rangle = \frac{4}{n}$, meaning that the vectors $\tilde{\mathbf{Y}}_{:i}$ are the extreme rays of a right circular cone with the eigenvector $\tilde{\mathbf{F}}_{:0} = \frac{2}{\sqrt{n}}[1, 0, \ldots, 0]^\top$ as its symmetry axis, see Fig. 3(d). Thus, we can interpret the solution to NOMAD as lifting the 2D ring structure into a cone. As mentioned before, this cone is high-dimensional, with as many directions as needed to preserve the nonnegativity of $\mathbf{Q}$.

We identify the rank of the solution $\mathbf{Q}$ with the number of active eigenvalues. The bigger the K, the higher the rank. The constraint $\mathbf{Q1} = \mathbf{1}$ in NOMAD leads to a fanning-out effect in the data representation. Intuitively, this fan-out effect is key to the disentanglement of datasets with complex topologies. Spin-model-inspired SDPs for community detection (Javanmard et al., 2015) achieve a similar fanning-out by dropping the constraint $\mathbf{Q1} = \mathbf{1}$ and adding the related term $-\gamma\mathbf{1}^\top\mathbf{Q1}$ to the objective function.

With the LP framework and the geometric picture in place, we can begin to understand how the solution evolves as the parameter K increases from 1 to n. At $K = 1$, only the eigenvalue $(\mathbf{q})_0$ is active and every vector $(\tilde{\mathbf{Y}})_{:i}$ is the same with each entry equal to $1/n$. When K slightly above 1, the eigenvalue $(\mathbf{q})_1$ becomes active (nonzero), introducing the first nontrivial Fourier component. Geometrically, the vectors $\{(\tilde{\mathbf{Y}})_{:i}\}$ now open up into a narrow cone. As K increases, the cone widens and, at some point, the angle between two of the vectors reaches $\pi/2$ (this activates the nonnegativity constraint in Eq. (7)). Further increase of K necessitates use of a larger number of Fourier modes. Finally, at $K = n$ all modes are active and all vectors $\{(\tilde{\mathbf{Y}})_{:i}\}$ become orthogonal to each other. Fig. 3(b) depicts the progression with K of the number of active modes.

Summary. Previous studies (Kulis et al., 2007; Peng and Wei, 2007; Awasthi et al., 2015), focus solely on cases where NOMAD exhibits K-means-like solutions (i.e., hard-clustering). Sec. 2 provides a characterization of the NOMAD solutions on a simple example with a high degree of symmetry, showing that they are drastically different from K-means. These solutions connect neighboring points, with the neighborhood size determined by K. These neighborhoods overlap, as they would in soft-clustering, in a way that preserves global features of the manifold, including its symmetry. This is a feature sought after by manifold learning methods and help place NOMAD among reliable manifold analysis techniques.

3. Analyzing data manifolds with NOMAD: Experimental results

In the previous section, we showed that NOMAD recovers the data manifold in an idealized 2D ring dataset. Here, we extend this observation numerically to more complex datasets for which analytical form of the transformation that diagonalizes $\mathbf{Q}_*$ (nor $\mathbf{D}$) is not known, see figs. 1, 5 and 7. We visualize the solution $\mathbf{Q}_*$ by embedding it in a low-dimensional space. While our goal is not dimensionality reduction, we learn the data manifold with NOMAD, and use standard spectral dimensionality reduction to visualize the results.

Recovering multiple manifolds. K-means cannot effectively recover multiple distinct manifolds (although in some very particular cases, with well separated and linearly separable manifolds, it may group the points correctly). Interestingly, NOMAD does not inherit this limitation. Of course, if we set the NOMAD parameter K to the number of manifolds that we want to recover, there is no hope in the general case to obtain a result substantially better than the one obtained with Lloyd's algorithm (Lloyd, 1982). However, setting the NOMAD parameter K to be higher than the number of manifolds leads to a correct identification and characterization of their structures. Note that

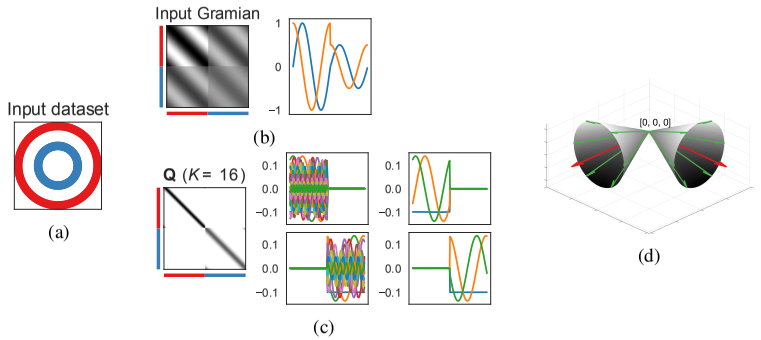


Figure 4: Solution of NOMAD on the dataset consisting of two 2D rings. (a) Two-ring dataset. (b) Input Gramian, $\mathbf{D}$, and its two eigenvectors ($\mathbf{D}$ has rank 2). Note that the eigenvectors of $\mathbf{D}$ do not segregate the rings. (c) The solution, $\mathbf{Q}$, of NOMAD contains two sets of eigenvectors with disjoint support: one set describing the points in each ring (we show all eigenvectors and a detail on the first 3 within each set). (d) The eigenvectors of $\mathbf{Q}$ form two orthogonal high-dimensional cones: one cone for each ring (cartoon representation, cone axis in red and eigenvectors in green). Notice how these cones become linearly-separable.

setting a similarly large K would not help K-means, as it is designed to partition the data, thus breaking each manifold into several pieces.

An example with two rings is presented in Fig. 4. We can expect that, as the single ring in Sec. 2 is described by Fourier modes, NOMAD describes two rings with two sets of Fourier modes with disjoint support; the solution is now arranged as two orthogonal high-dimensional cones, see Fig. 4(d). In a sense, the manifold learning problem is already solved, as there are two circulant submatrices, one for each manifold, with no interactions between them. If the user desires a hard assignment of points to manifolds, we can simply consider $\mathbf{Q}_*$ as the adjacency matrix of a weighted graph and compute its connected components.

Discussion of the experimental results. To demonstrate the manifold-learning capabilities of the NOMAD, we present several examples, both synthetic and real-world. The trefoil knot in Fig. 1(a) is a 1D manifold in 3D; it is the simplest example of a nontrivial knot, meaning that it is not possible to "untie" it in three dimensions without cutting it. However, the manifold learning procedure in Sec. 3 learns a closed 1D manifold. We also present examples using real-world high-dimensional datasets, recovering in every case structures of interest, see Fig. 6. In figs. 6(a) to 6(c), NOMAD respectively uncovers the camera rotation, the orientation of the lighting source, and specific handwriting features.

To demonstrate the multi-manifold learning and manifold-disentangling capabilities of NOMAD, we use several standard synthetic datasets, see figs. 1(b), 4 and 5. In all of these examples, NOMAD is able to disentangle clusters that are not linearly separable. We also present results for a real-world dataset (Fig. 7) which is similar to the one in Fig. 6(a) but with two objects. NOMAD

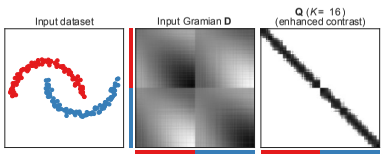
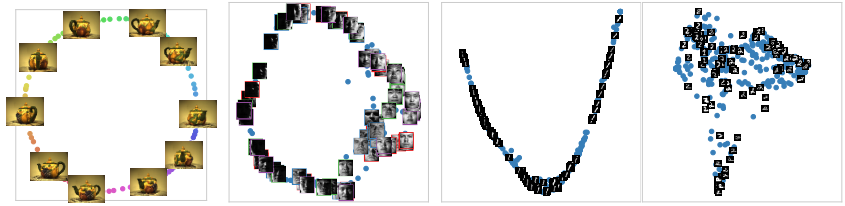


Figure 5: Learning multiple manifolds with NOMAD. The points are arranged in two semicircular manifolds and contaminated with Gaussian noise. Although the manifolds are linearly non-separable, NOMAD correctly finds two submatrices, one for each manifold (for visual clarity, we enhance the contrast of $\mathbf{Q}$).



(a) Teapots ($K = 20$) (b) Yale Faces ($K = 16$) (c) MNIST digits ($K = 16$)

Figure 6: Finding two-dimensional embeddings with NOMAD. (a) 100 images obtained by viewing a teapot from different angles in a plane. The input vectors size is 23028 (76×101 pixels, 3 color channels). The manifold uncovers the change in orientation. (b) 256 images from 4 different subjects (each subject is marked with a different color in the figure), obtained by changing the position of the illumination source. The input vectors size is 32256 (192×168 pixels). The manifold uncovers the change in illumination (from frontal, to half-illuminated, to dark faces, and back). (c) 500 images handwritten instances of the same digit. The input vectors size is 784 (28×28 pixels). On the left and on the right, images of the digits 1 and 2, respectively. The manifold of 1s uncovers their orientation, while the manifold of 2s parameterizes features like size, slant, and line thickness. Details are better perceived by zooming on the plots.

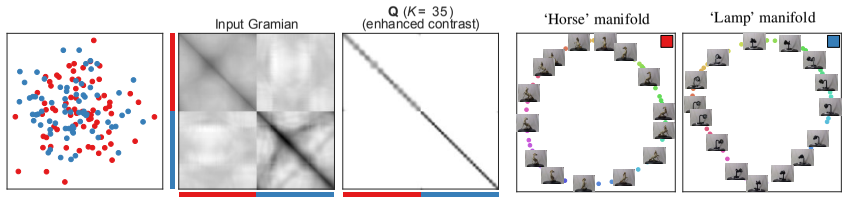


Figure 7: 144 images obtained by viewing a lamp and a horse figurine from different angles in a plane. The input vectors size is 589824 (384×512 pixels, 3 color channels). We plot the input data using a 2D spectral embedding (the points corresponding to each object are colored differently). NOMAD correctly finds two submatrices, one for each manifold (for visual clarity, we enhance the contrast of $\mathbf{Q}$); furthermore, NOMAD recovers closed manifolds.

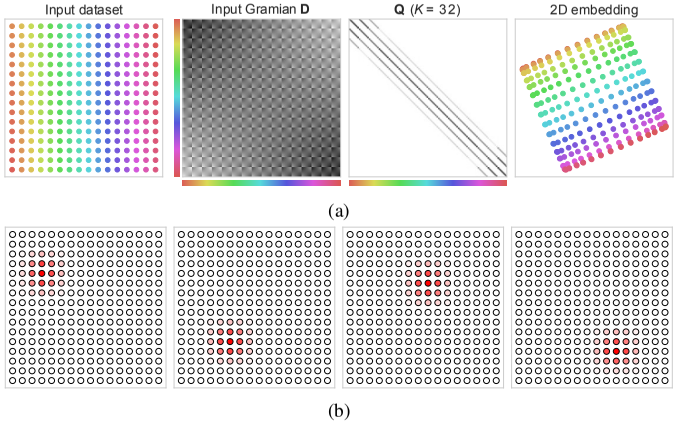


Figure 8: Learning a 2D manifold embedded in a 10-dimensional ambient space; the first two dimensions are regular samples on a 2D grid (shown on the left) and the remaining ones are Gaussian noise. (a) NOMAD recovers the right 2D structure (of course, some distortion is expected along the edges). (b) We show a few columns of $\mathbf{Q}$ on top of the data itself: the red level indicates the value of the corresponding entry in $\mathbf{Q}$ (red maps to high values, white maps to a zero). NOMAD effectively tiles the dataset with a collections of overlapping local neighborhoods centered at each data point. These patches contain all the information necessary to reconstruct the intrinsic manifold geometry.

recovers two closed manifolds, each of which containing the viewpoints of one object. The structure of the solution is similar to the one in Fig. 4(d).

Finally, we include an example in which NOMAD captures the structure of a 2D manifold living in a 10-dimensional space, see Fig. 8. NOMAD assigns a local patch to each data point (non-zero values for neighboring points, zeros elsewhere). These local patches tile the manifold with overlap (as in soft-clustering), allowing to recover its grid structure. Such tiling takes place in all of the examples included in the paper.

3.1. Manifold disentangling with multi-layer NOMAD

The recursive application of NOMAD, with successively decreasing values of K, enhances its manifold-disentangling capabilities. The pseudocode is as follows:

$\mathbf{D}_1 \leftarrow \mathbf{X}^\top \mathbf{X}$;
for $l = 1, 2, \ldots$ **do**
  Choose K_l (for all $l > 1$ we require $K_l \leq K_{l-1}$);
  Find the solution $\mathbf{Q}_l$ of NOMAD with input matrix $\mathbf{D}_l$ and parameter K_l;
  $\mathbf{D}_{l+1} \leftarrow \mathbf{Q}_l$;
return $\{\mathbf{Q}_l\}_{l=1,2,\ldots}$

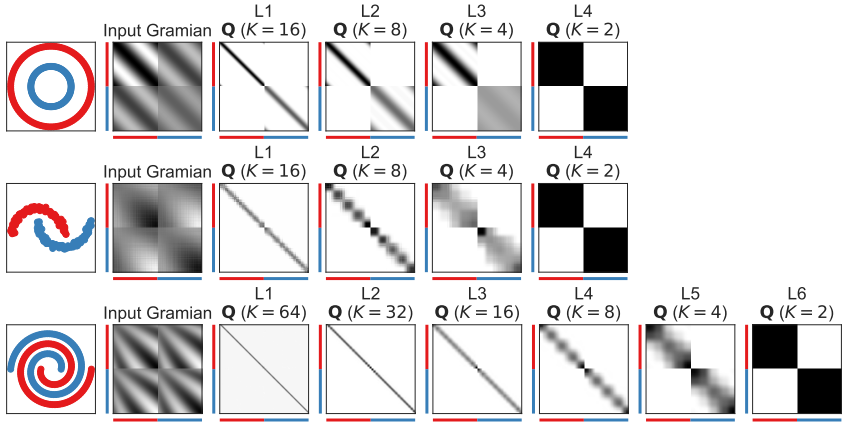


Figure 9: Results of recursive NOMAD application (multi-layer NOMAD). For each example, we show matrices $\mathbf{Q}_*$ computed by the successive application of the algorithm. Multi-layer NOMAD untangles these linearly non-separable manifolds and, in the final layer, assigns each manifold to one cluster.

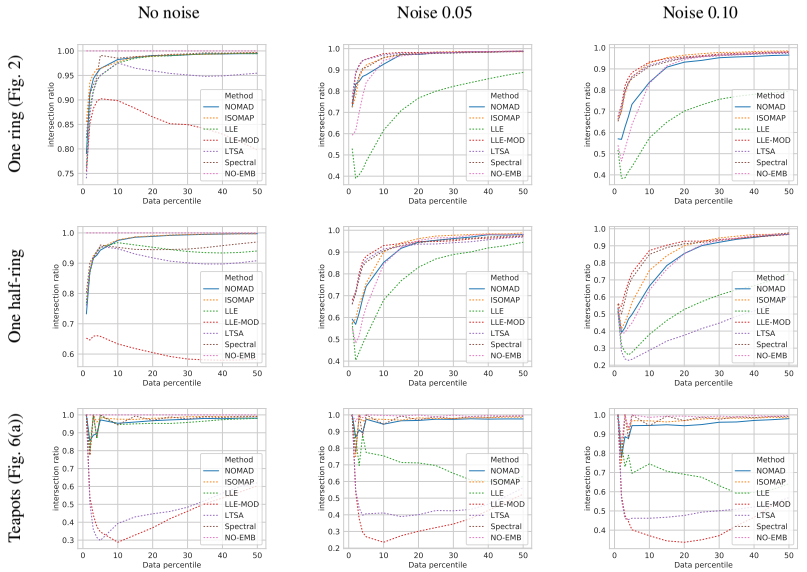
In Fig. 9, we present the evolution of successive matrices $\mathbf{Q}_l$. In all of these examples, multi-layer NOMAD is able to correctly identify clusters that are not linearly separable, something unattainable with single-layer NOMAD or with K-means clustering. Interestingly, we find that the manifolds are already segregated after one application of NOMAD in the direction of the leading eigenvectors of $\mathbf{Q}$ (see Fig. 4). The rest of the NOMAD layers little-by-little sieve out the (unwanted) smaller eigenvalues in an unsupervised fashion.

To turn this algorithm into a general data-analysis tool, we need an automated selection of the values $\{K_l\}$ which is a non-trivial task in general. Additional results, using different sequences $\{K_l\}$, can be found in Appendix C. Further research is needed to develop such algorithm and fully understand multi-layer NOMAD's interesting behavior.

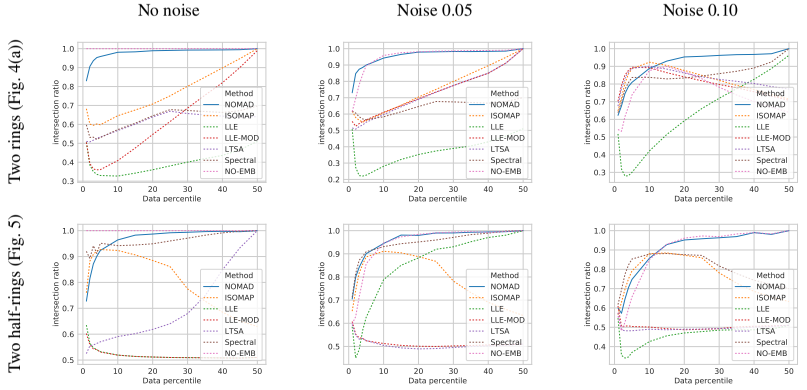
3.2. Geodesic-distance preservation: NOMAD versus existing manifold learning techniques

In order to compare different methods for manifold discovery, we need to agree upon appropriate metrics, which itself is an active area of research (e.g., Zhang et al., 2012). In particular, we want a metric that allows fair comparison among outputs of methods with very different objectives. Since our method is not explicitly geared towards dimension reduction, or towards variance maximization, we prefer metrics that emphasize the preservation of intrinsic structure of the manifold. In particular, we hope to preserve the ordering of intrinsic distances along the manifold, something that guarantees that the neighborhood structure remain similar.

Concretely, (1) we compute N nearest neighbors for each dataset point and build a weighted graph with these distances as edges; (2) we use this graph to compute geodesic distances using



(a) One manifold: NOMAD is among the best-in-class methods.



(b) Two manifolds: NOMAD outperforms all considered methods.

Figure 10: Comparison of the robustness of geodesic distances to the addition of noise for different manifold learning methods. The description of the experimental protocol is in Sec. 3.2. The method termed NO-EMB directly computes distances on the noisy data.

Dijkstra's algorithm; (3) we finally sort the distances in increasing order. We consider this ordering our ground truth.

We then add noise to the each point in the dataset. Noise was added by creating $5d$ additional dimensions that contain Gaussian noise with standard deviation 0.05 and 0.10. Using the noisy dataset, we run the geodesic-distance-sorting process on the embeddings produced by different manifold learning algorithms using N nearest neighbors. For NOMAD, instead of resorting to nearest neighbors, we set $K = n/N$ and use the non-zero entries in $\mathbf{Q}$ to determine the graph connectivity. As NOMAD yields a similarity matrix $\mathbf{Q}$, we derive distances from it with the formula $(\mathbf{Q})_{ii} + (\mathbf{Q})_{jj} - 2(\mathbf{Q})_{ij}$. Using this weighted graph, we compute and sort the geodesic distances.

For our distance-preservation measure, we use a bullseye score: for each method, we count the fraction of points in the top p percentile of distances that are also present in the top p percentile of ground truth distances.

As seen in Fig. 10, when the data is sampled from a single manifold, NOMAD performs very well, on par with the best algorithms included in our comparison. However, in the two-manifolds case, NOMAD clearly outperforms all other methods, nearly matching the performance of direct distance computations on the noisy data.

4. Heuristic non-convex solvers for large-scale NOMAD

Standard SDPs involve $O(n^2)$ variables and their resulting time complexity is often $O(n^3)$. Consequently, standard solvers (O'Donoghue et al., 2016a) will struggle with large datasets. NOMAD lends itself to a fast and big-data-friendly implementation (Kulis et al., 2007). This is done by posing a related problem

$$\max_{\mathbf{Y}\in\mathbb{R}^{r\times n}} \operatorname{Tr}\left(\mathbf{D}\mathbf{Y}^\top\mathbf{Y}\right) \quad \text{s.t.} \quad \operatorname{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right) = K,\ \mathbf{Y}^\top\mathbf{Y}\mathbf{1} = \mathbf{1},\ \mathbf{Y} \geq \mathbf{0}. \tag{11}$$

In this new problem, we have forgone convexity in exchange of reducing the number of unknowns from $O(n^2)$ to rn. For example, Kulis et al. (2007) set $r = K$. The problematic constraint $\mathbf{Y}^\top\mathbf{Y} \geq \mathbf{0}$, involving $O(n^2)$ terms, has been replaced by the much stronger but easier to enforce $\mathbf{Y} \geq \mathbf{0}$. The speed gain is shown in Fig. 15. See Appendix E for a description of the algorithm.

However, strictly speaking, the new constraint is equivalent to the old one only if $\mathbf{Q}$ is completely positive. An $n \times n$ matrix $\mathbf{A}$ is called completely positive (CP) if there exists $\mathbf{B} \geq \mathbf{0}$ such that $\mathbf{A} = \mathbf{B}^\top\mathbf{B}$. The least possible number of rows of $\mathbf{B}$ is called the cp-rank of $\mathbf{A}$. Whereas matrix $\mathbf{A}$ is doubly nonnegative (DN), i.e. $\mathbf{A} \geq \mathbf{0}$ and $\mathbf{A} \succeq 0$, not every DN matrix (with $n > 4$) is CP (Maxfield and Minc, 1962).

We are thus interested in two questions. First, is the solution $\mathbf{Q}_*$ to NOMAD completely positive? Answering this question in the affirmative would allow for theoretically sound and fast implementations of NOMAD. Whereas the set of CP matrices forms a convex cone, the problems of determining whether a matrix is inside the set and of projecting a matrix into the set are NP-hard leading us to the second question: What is the cp-rank of $\mathbf{Q}_*$? This issue is critical because it determines the number of unknowns. For example, if $\text{cp-rank}(\mathbf{Q}_*) \leq K$, (11) would be easier to solve. These questions are difficult only when NOMAD produces a soft-clustering $\mathbf{Q}_*$, as in all of the examples in this paper. Indeed, it is not hard to prove that, whenever NOMAD produces a hard-clustering $\mathbf{Q}_*$, $\mathbf{Q}_*$ is CP (see Awasthi et al., 2015, for such conditions).

Let us now go back to the example in Sec. 2 (points arranged regularly on a ring). For this example, we can establish a simple sufficient condition on K, for $\mathbf{Q}_*$ to be CP. Recall that if $\mathbf{D}$ is

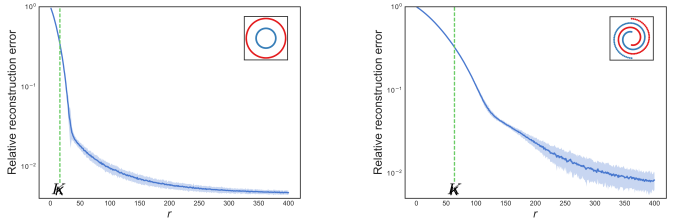


Figure 11: We empirically study the cp-rank of $\mathbf{Q}_*$. As a proxy of the exact nonnegative decomposition, we compute the rank-r symmetric NMF $\mathbf{Q}_* \approx {\mathbf{Y}_+}^\top\mathbf{Y}_+$ for different values of r. We show the mean plus/minus two standard deviations of the relative error $\left\|\mathbf{Q}_* - {\mathbf{Y}_+}^\top\mathbf{Y}_+\right\|_F / \left\|\mathbf{Q}_*\right\|_F$ computed from 50 different SNMFs for each r (their differences stem from the random initialization). Both datasets have 200 points. Clearly, setting $r = K$ is not enough to properly reconstruct $\mathbf{Q}_*$.

circulant, $\mathbf{Q}_*$ is circulant (Bachoc et al., 2012). In Proposition 2 of Appendix B, we prove that if the solution $\mathbf{Q}_*$ to NOMAD is a circulant matrix, then it is CP for every $K \leq 3/2$ or $K \geq \frac{n}{2}$. Naturally, more theory is needed to shed light onto this problem in general scenarios as it is unclear whether similar results exist.

Complementarily, we have studied the questions raised in this section from an experimental viewpoint. We use the symmetric nonnegative matrix factorization (SNMF) of $\mathbf{Q}_*$, see Appendix D, as a proxy for checking whether $\mathbf{Q}_*$ is CP. The rationale is that if the approximation with SNMF is very tight, it is highly likely that $\mathbf{Q}_*$ is CP. These experiments are presented in Fig. 11. We found that, with a properly chosen rank r, SNMF can indeed accurately approximate $\mathbf{Q}_*$. However, setting $r = K$ is in general not enough and leads to a poor reconstruction. These two facts support the idea that $\mathbf{Q}_*$ is CP, but has a cp-rank much higher than K.

Our experiments with the non-convex algorithm in Appendix E lead to similar conclusions as those with SNMF, see Fig. 12. Setting $r = K$, leads to a poor approximation of $\mathbf{Q}_*$ and, as observed by Kulis et al. (2007), to hard-clustering. Setting $r \gg K$ leads to much improved reconstructions, at the expense of speed.

5. A fast and convex algorithm for NOMAD

The Burer-Monteiro solver forgoes convexity in favor of speed. However, as discussed in the previous section, this conversion carries theoretical and practical difficulties that are not easily overcome. In this section, we propose an algorithm for NOMAD that is fast and yet convex.

5.1. Augmented Lagrangian formulation

First, we redefine the variables in NOMAD by setting $\mathbf{P} = \mathbf{Q} - \mathbf{E}_n$, where $\mathbf{E}_n = \frac{1}{n}\mathbf{1}\mathbf{1}^\top$. Then,

$$\max_{\mathbf{P}} \operatorname{Tr}(\mathbf{D}\mathbf{P}) \quad \text{s.t.} \quad \mathbf{P}\mathbf{1} = \mathbf{0}, \quad \operatorname{Tr}(\mathbf{P}) = K - 1, \quad \mathbf{P} \succeq \mathbf{0}, \quad \mathbf{P} + \mathbf{E}_n \geq \mathbf{0}. \tag{12}$$

As usual in the optimization literature, we handle this constraint with an augmented Lagrangian method. The augmented Lagrangian of Problem (12) with respect to the constraint $\mathbf{P} + \mathbf{E}_n \geq \mathbf{0}$ is

$$g(\mathbf{P}, \mathbf{\Gamma}) = -\operatorname{Tr}(\mathbf{D}\mathbf{P}) + \operatorname{Tr}(\mathbf{\Gamma}(\mathbf{P} + \mathbf{E}_n)) + \tfrac{\gamma}{2}\left\|[\mathbf{P} + \mathbf{E}_n]_-\right\|_F^2, \tag{13}$$

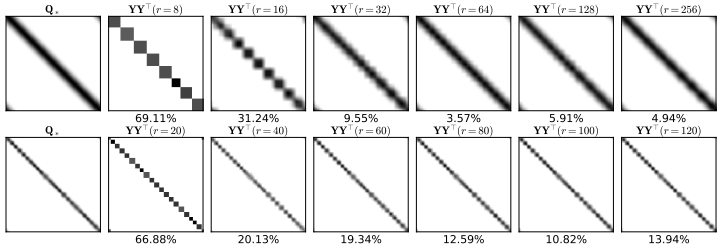


Figure 12: Comparison of the results obtained with a standard SDP solver (first column) and with a low-rank non-convex approach (remaining columns, r denotes the rank of the obtained solution). **(Top)** Dataset in Fig. 4; we set $K = 8$ in all cases. **(Bottom)** Dataset in Fig. 6(a); we set $K = 20$ in all cases. In each case, we also display the relative error between the matrix $\mathbf{Y}^\top\mathbf{Y}$ and $\mathbf{Q}_*$. Interestingly, setting $r = K$ produces hard clustering solutions (see the block diagonal structure of the matrices on the second column), while increasing r produces "softer" solutions. This suggests that the cp-rank of $\mathbf{Q}_*$ is (much) greater than K.

where $\mathbf{\Gamma} \geq \mathbf{0}$ is the associated Lagrange multiplier, and $[\cdot]_- = \min(\cdot, 0)$ is the projection operator onto the negative orthant. We can then pose Problem (12) as

$$\min_{\mathbf{P}} \max_{\mathbf{\Gamma}\geq 0} g(\mathbf{P}, \mathbf{\Gamma}) \quad \text{s.t.} \quad \mathbf{P1} = \mathbf{0}, \quad \operatorname{Tr}(\mathbf{P}) = K - 1, \quad \mathbf{P} \succeq \mathbf{0}. \tag{14}$$

We solve it using the method of multipliers, i.e.,

$$\mathbf{P}_{t+1} = \operatorname*{argmin}_{\mathbf{P}} g(\mathbf{P}, \mathbf{\Gamma}_t) \quad \text{s.t.} \quad \mathbf{P1} = \mathbf{0}, \quad \operatorname{Tr}(\mathbf{P}) = K - 1, \quad \mathbf{P} \succeq \mathbf{0}, \tag{15a}$$

$$\mathbf{\Gamma}_{t+1} = [\mathbf{\Gamma}_t + \tau(\mathbf{P}_{t+1} + \mathbf{E}_n)]_- . \tag{15b}$$

5.2. A conditional gradient method for SDPs with an orthogonality constraint

In this section, we introduce a very efficient algorithm to solve

$$\max_{\mathbf{Z}} f(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{Z} \succeq 0, \ \operatorname{Tr}(\mathbf{Z}) = s, \ \mathbf{Zb} = \mathbf{0}. \tag{16}$$

of which Problem (15a) is an instance.

To this end we modify an algorithm to efficiently solve the SDP (Hazan, 2008)

$$\max_{\mathbf{Z}} f(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{Z} \succeq 0, \ \operatorname{Tr}(\mathbf{Z}) = s, \tag{17}$$

where function f is differentiable and concave. The iterative algorithm consists, at each iteration $t = 0 \ldots$, of the following steps:

1. Let $\mathbf{v}_t$ be the largest algebraic eigenvector of $\nabla f(\mathbf{Z}_t)$.
2. $\mathbf{Z}_{t+1} = (1-\alpha)\mathbf{Z}_t + \alpha s \mathbf{v}_t\mathbf{v}_t^\top$ with $\alpha = 2/(t+2)$.

Algorithm 1: Conditional gradient algorithm for SDPs with an orthogonality constraint

input : function f to minimize, scale parameter s.
output : solution $\mathbf{Z}_{t+1} \in \mathcal{P}_s$ to Problem (17).

Initialize $\mathbf{Z}_0 = \mathbf{0}$;
for $t = 0, \ldots, \infty$ **do**
  Let $\mathbf{v}$ be the largest algebraic eigenvector of $\nabla f(\mathbf{Z})$ such that $\mathbf{v}^\top\mathbf{b} = 0$;
  $\alpha \leftarrow 2/(t+2)$;
  $\mathbf{Z}_{t+1} \leftarrow (1-\alpha)\mathbf{Z}_t + \alpha s \mathbf{v}\mathbf{v}^\top$;
  if *converged* **then** break ;

This algorithm is an instance of the Frank-Wolfe/conditional-gradient algorithm (Frank and Wolfe, 1956). As such it provides a solution without performing any projections. First, $\mathbf{Z}_{t+1}$ is a non-negative linear combination of two positive semidefinite matrices, and is thus positive semidefinite itself. Second, the iterations maintain the invariant $\operatorname{Tr}(\mathbf{Z}_t) = s$ as $\operatorname{Tr}(\mathbf{Z}_{t+1}) = (1-\alpha)\operatorname{Tr}(\mathbf{Z}_t) + \alpha s \operatorname{Tr}\left(\mathbf{v}_t\mathbf{v}_t^\top\right) = (1-\alpha)\operatorname{Tr}(\mathbf{Z}_t) + \alpha s$.

We now show how to extend this algorithm to handle an orthogonality constraint. Let $\mathcal{P}_s$ be the convex cone of positive semidefinite matrices with trace s that are orthogonal to a given vector $\mathbf{b}$, i.e.,

$$\mathcal{P}_s = \{\mathbf{Z} \succeq 0, \ \operatorname{Tr}(\mathbf{Z}) = s, \ \mathbf{Zb} = \mathbf{0}\}. \tag{18}$$

Notice that setting $\mathbf{b} = \mathbf{1}$ yields the constraints of Problem (15a). We seek to solve

$$\max_{\mathbf{Z}} f(\mathbf{Z}) \quad \text{s.t.} \quad \mathbf{Z} \in \mathcal{P}_s. \tag{19}$$

Fortunately, we can push the constraint $\mathbf{Zb} = \mathbf{0}$ into the eigenvector computation. We begin by noticing that the final solution is a weighted sum of the matrices $\mathbf{v}_t\mathbf{v}_t{}^\top$. It then suffices to require that, for every t, $\mathbf{v}_t\mathbf{v}_t{}^\top\mathbf{b} = \mathbf{0}$, which reduces to $\mathbf{v}_t{}^\top\mathbf{b} = \mathbf{0}$. This naturally yields a new iterative method, summarized in Alg. 1. This algorithm has the same performance guarantee as Hazan's (2008), given by the following proposition, which we prove in Appendix F.

Proposition 1 *Let $\mathbf{X}, \mathbf{Z} \in \mathcal{P}_s$ and $\mathbf{Y} = \mathbf{X} + \alpha(\mathbf{Z} - \mathbf{X})$ and $\alpha \in \mathbb{R}$. The curvature constant of f is*

$$C_f := \sup_{\mathbf{X},\mathbf{Z},\alpha} \tfrac{1}{\alpha^2}[f(\mathbf{X}) - f(\mathbf{Y}) + (\mathbf{Y} - \mathbf{X}) \bullet \nabla f(\mathbf{X})]. \tag{20}$$

Let $\mathbf{Z}^\star$ be the solution to Problem (19). The iterates $\mathbf{Z}_t$ of Alg. 1 satisfy for all $t > 1$

$$f(\mathbf{Z}^\star) - f(\mathbf{Z}_t) \leq \tfrac{8C_f}{t+2}. \tag{21}$$

5.3. A conditional gradient algorithm for NOMAD

Alg. 2 summarizes the proposed method of multipliers, see iterations (15a) and (15b), to solve Problem (12). The inner problem (15a) is solved using Alg. 1. A few remarks are in order:

- When using the method of multipliers, it is often not necessary (nor desirable) to solve the inner problem to a high precision (Goldstein and Osher, 2009). In our implementation we set $N_{\text{inner}} = 10$.

Algorithm 2: Conditional gradient algorithm for NOMAD

input : matrix $\mathbf{D}$, scale parameter k.
output : solution $\mathbf{Q}$ to NOMAD.

Initialize $\mathbf{P}_0 = \mathbf{0}$; $\mathbf{\Gamma} \leftarrow \mathbf{0}$; $\gamma = 1$;
for $t = 1, \ldots, \infty$ **do**
  for $t_{inner} = 1, \ldots, N_{inner}$ **do**
    Let $\nabla g(\mathbf{P}, \mathbf{\Gamma}) = -\mathbf{D} + \mathbf{\Gamma} + \gamma\,[\mathbf{P} + \mathbf{E}_n]_-$;
    Let $\mathbf{A} = (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\nabla g(\mathbf{P}, \mathbf{\Gamma})(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)$;
    Let $\mathbf{v}$ be the smallest algebraic eigenvector of $\mathbf{A}$, such that $\mathbf{v}^\top\mathbf{1} = 0$;
    $\mathbf{H} \leftarrow (K-1)\mathbf{v}\mathbf{v}^\top$;
    $\alpha \leftarrow 2/(t + t_{\text{inner}} + 2)$;
    $\mathbf{P} \leftarrow (1-\alpha)\mathbf{P} + \alpha\mathbf{H}$;
  $\mathbf{\Gamma} \leftarrow [\mathbf{\Gamma} + \tau\,(\mathbf{P} + \mathbf{E}_n)]_-$;
  if *converged* **then** break ;
$\mathbf{Q} \leftarrow \mathbf{P} + \mathbf{E}_n$

- There is no need to need for a highly accurate eigenvector computation (Hazan, 2008). We use the Lanczos algorithm and set its accuracy to $(t+1)^{-1}$.
- Alg. 1 solves a maximization problem and requires the eigenvector with the largest algebraic eigenvalue. To solve the minimization problem (15a), we simply compute the eigenvector with the smallest algebraic eigenvalue (Jaggi, 2013).
- As $\mathbf{b} = \mathbf{1}$, we can enforce the orthogonality constraint $\mathbf{v}_t{}^\top\mathbf{1} = 0$ by computing the maximum eigenvalue of $\mathbf{A} = (\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)\nabla g(\mathbf{P}, \mathbf{\Gamma})(\mathbf{I} - \frac{1}{n}\mathbf{1}\mathbf{1}^\top)$. This operation can be carried out very efficiently.

Complexity. The complexity of Alg. 1 is similar to that of Hazan's (2008), plus an additional factor to compute $\mathbf{A}$. From Proposition 1, Alg. 1 yields a solution with accuracy ε, i.e., $f(\mathbf{Z}_t) \geq f(\mathbf{Z}^\star) - \varepsilon$, in $\frac{4C_f}{\varepsilon} - 1$ iterations. Computing $\nabla g(\mathbf{P}, \mathbf{\Gamma})$, $\mathbf{A}$, and $\mathbf{H}_t$ require n^2 operations. Let T_{EIG} be the number of iterations of the eigensolver, each iteration taking $O(n^2)$ operations. Additional operations require $O(n)$ time. Then, the overall complexity of Alg. 1 is

$$O\left(\frac{C_f}{\varepsilon}\left[n + n^2 + n^2 T_{\text{EIG}}\right]\right). \tag{22}$$

For the Lanczos algorithm, and our accuracy setting of $(t+1)^{-1}$, we have $T_{\text{EIG}} = O((t+1)\log n)$. In this case, the complexity per iteration is $O(n^2 \log n)$. As a comparison, standard SDP solvers have a complexity of $O(n^3)$ per iteration. These solvers also involve significant memory usage, while our algorithm has an optimal space complexity of $O(n^2)$.

5.4. Experimental analysis

Throughout the iterations of Alg. 2, $\mathbf{P} \in \mathcal{P}_{k-1}$, see Eq. (18). Thus, we only need to keep track of the constraint $\mathbf{Q} = \mathbf{P} + \mathbf{E}_n \geq \mathbf{0}$ and of the value of the objective $\text{Tr}\,(\mathbf{DP})$.

We illustrate with two typical examples the empirical convergence of these values in Fig. 13. the convergence the objective value is clearly superlinear, while we observe a linear convergence for the nonnegativity constraint. Accelerating the latter rate is an interesting line of future research.

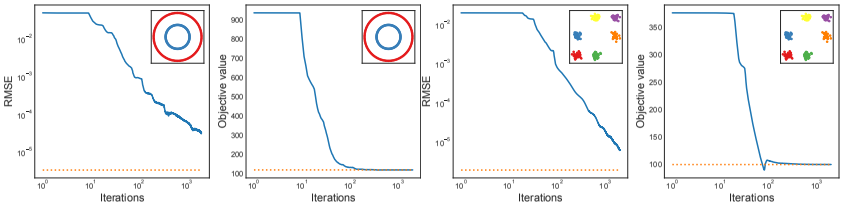


Figure 13: Prototypical examples of the behavior of the proposed conditional gradient NOMAD solver as its iterations progress. On the left plots, we show the RMSE of $[\mathbf{Q}]_- = [\mathbf{P} + \mathbf{E}_n]_-$, with the average computed over its non-zero entries. After about 10 iterations, the RMSE drops linearly, as usual for the method of multipliers. On the right plots, we display the objective value $\text{Tr}\,(\mathbf{DP})$, which usually converges in a few hundred iterations. In each case, as a reference, we show in orange the values returned by the standard SDP solver. The proposed algorithm enforces the nonnegativity constraint in NOMAD less accurately (although accurate enough for practical purposes), while exactly enforcing all the other constraints.

We can see in Fig. 13 that standard solvers enforce the nonnegativity constraint more accurately. However, they do not exactly enforce $\mathbf{P} \in \mathcal{P}_{k-1}$. There is a trade-off between what can be enforced up to which precision, making the solutions sometimes not exactly comparable.

We show the suitability of the proposed NOMAD solver in Fig. 14. In the vast majority of cases the solutions are the same. While the proposed method enforces the nonnegativity constraint less accurately than the standard solver, it enforces all the other constraints exactly. This is why in the teapot example, bottom left of Fig. 14, the solution of the proposed method looks less jagged than the one of the standard solver: the constraint $\mathbf{Q1} = \mathbf{1}$ is more accurately enforced, resulting in a more "circulant" representation.

In Fig. 15, we present the speed comparison of computing NOMAD with three different methods: two state-of-the-art SDP solvers, SCS (O'Donoghue et al., 2016b) and SDPNAL+ (Yang et al., 2015), the low-rank Burer-Monteiro solver (discussed in Sec. 4), and the proposed conditional gradient method. The Burer-Monteiro method is the fastest. Keep in mind that the latter does not guarantee convergence to the global optimal solution; this is particularly true specially in its fastest setting, i.e., by keeping r relatively small, see Sec. 4. Among solvers that solve a convex problem, for very small problems (up to 250 points), standard SDP solvers are the fastest. For larger problems the proposed solver is significantly faster. It is important to point out that, in theory, the speed difference grows significantly larger. This is hard to show in practice as standard solvers either run out of memory very quickly (SCS) or are implemented to time out for big instances (SDPNAL+); the proposed solver has a much more efficient use of memory.

We highlight the extended computational capabilities of the proposed conditional gradient method with an example that cannot be handled by standard SDP solvers. We use as input the 9603×9603 Gramian formed by all (vectorized) images of the digit zero in MNIST. The proposed algorithm is able to compute a solution to NOMAD with ease for a problem size about 100 times larger than the upper size limit for standard solvers. In the 2D embedding of the solution (see Sec. 3 for details about its computation), shown in Fig. 16, we can clearly see that the images are organized by their intrinsic characteristics (elongation and rotation).

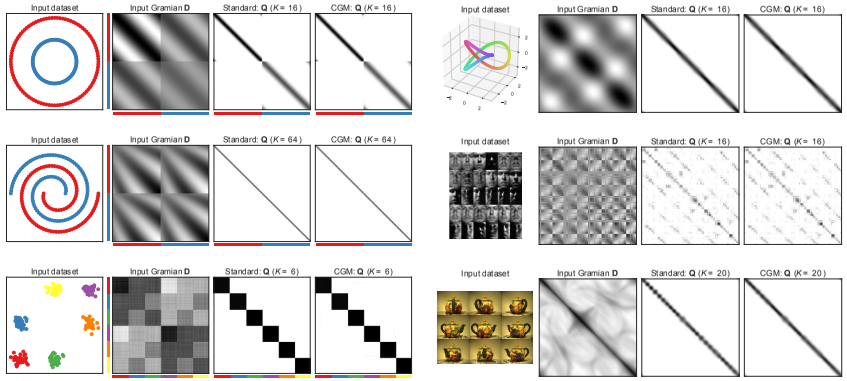


Figure 14: Comparison of the standard SDP solver with the proposed conditional gradient solver (CGM) for NOMAD on different datasets. In most cases, the results are practically indistinguishable while being delivered much faster.

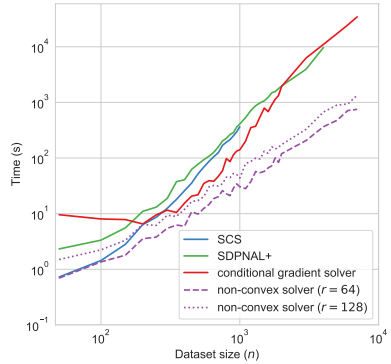


Figure 15: Running time comparison (smaller is better) of different NOMAD solvers for $K = 16$ (SCS (O'Donoghue et al., 2016b) and SDPNAL+ (Yang et al., 2015) are written in a highly optimized C/C++ code, while we use our non-optimized Python code for the others). The non-convex solver is much faster than the convex ones. Unfortunately, it may yield different results, see Fig. 12, and may not converge to the global maximum. The conditional gradient algorithm proposed in this paper is much faster than SCS and SDPNAL+ (about three times faster for $n = 10^3$) but guarantees converging to the global optimum. Additionally, the proposed algorithm handles large problems seamlessly: in our desktop with 128GB of RAM, SCS (running under CVXPY) runs out of memory with instances larger than $n = 1200$) while SDPNAL+ times out before converging for instances larger than $n = 4000$.

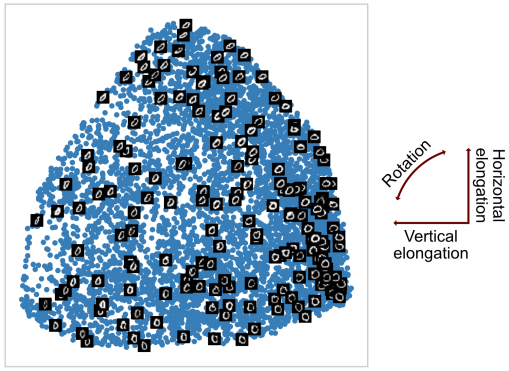


Figure 16: We show the 2D embedding of the digit 0 in MNIST, computed in the same fashion as in Fig. 6. In this case, we use all 9603 images of the digit and obtain a 9603×9603 matrix. We compute the solution of NOMAD with $K = 128$ using the proposed conditional gradient method (Alg. 2). In contrast, traditional SDP-solvers can only handle dense matrices approximately 100 times smaller. As in Fig. 6, the data gets organized according to different visual characteristics of the hand-written digit (e.g., orientation and elongation).

6. Conclusions

In this work, we showed that NOMAD can learn multiple low-dimensional data manifolds in high-dimensional spaces. An SDP instance, it is convex and can be solved in polynomial-time. Unlike most manifold learning algorithms, the user does not need to select/use a kernel and no nearest-neighbors searches are involved.

We also studied the computational performance of NOMAD. We first focused on a non-convex Burer-Monteiro-style algorithm and performed both theoretical and empirical analysis. Finally, we presented a new algorithm for NOMAD based on the conditional gradient method. The proposed algorithm is convex and yet efficient. This algorithm allows us, for the first time, to analyze the behavior of NOMAD on large datasets.

Related and future work. It has not escaped our attention that NOMAD can be considered as an instance of kernel alignment (Cristianini et al., 2002). In supervised setting, kernel alignment has been previously formulated as an SDP (e.g., Lanckriet et al., 2004; Cortes et al., 2012). Even beyond the distinction between the supervised and unsupervised scenarios, this body of work differs significantly from NOMAD. Its goal is to optimally combine pre-computed kernel matrices, whereas NOMAD learns such a matrix from scratch. Nonetheless, we find this connection with kernel learning very promising and plan to investigate it further in the future.

Acknowledgments

We thank Afonso Bandeira, Alexander Genkin, Victor Minden, and Cengiz Pehlevan for helpful discussions.

References

A. Amini and E. Levina. On semidefinite relaxations for the block model. Technical report, arXiv:1406.5647, 2014.

P. Awasthi, A. Bandeira, M. Charikar, R. Krishnaswamy, S. Villar, and R. Ward. Relax, no need to round: Integrality of clustering formulations. In *ITCS*, 2015.

C. Bachoc, D. C. Gijswijt, A. Schrijver, and F. Vallentin. Invariant semidefinite programs. In *Handbook on semidefinite, conic and polynomial optimization*, pages 219–269. Springer, 2012.

M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 15(6):1373–1396, 2003.

N. Boumal, V. Voroninski, and A. Bandeira. The non-convex Burer-Monteiro approach works on smooth semidefinite programs. In *NIPS*, 2016.

R. Byrd, P. Lu, J. Nocedal, and C. Zhu. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5):1190–1208, 1995.

Y. Cho and L. Saul. Kernel methods for deep learning. *NIPS*, pages 342–350, 2009.

K. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. *ACM Transactions on Algorithms*, 6(4):1–30, 2010.

C. Cortes, M. Mohri, and A. Rostamizadeh. Algorithms for learning kernels based on centered alignment. *Journal of Machine Learning Research*, 13(Mar):795–828, 2012.

N. Cristianini, J. Kandola, A. Elisseeff, and J. Shawe-Taylor. On kernel-target alignment. *NIPS*, 2002.

C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the β-divergence. *Neural Computation*, 23(9):2421–2456, 2011.

M. Frank and P. Wolfe. An algorithm for quadratic programming. *Naval Research Logistics Quarterly*, 3(1-2):95–110, 1956.

T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

E. Hazan. Sparse approximate solutions to semidefinite programs. In *LATIN*, 2008.

Y. Hou, J. Whang, D. Gleich, and I. Dhillon. Non-exhaustive, overlapping clustering via low-rank semidefinite programming categories and subject descriptors. In *KDD*, 2015.

T. Iguchi, D. Mixon, J. Peterson, and S. Villar. On the tightness of an SDP relaxation of k-means. Technical report, arXiv:1505.04778, 2015.

M. Jaggi. Revisiting Frank-Wolfe: Projection-free sparse convex optimization. *ICML*, 2013.

A. Javanmard, A. Montanari, and F. Ricci-Tersenghi. Phase transitions in semidefinite relaxations. Technical report, arXiv:1511.08769, 2015.

M. Kaykobad. On nonnegative factorization of matrices. *Linear Algebra and its Applications*, 96: 27–33, 1987.

B. Kulis, A. Surendran, and J. Platt. Fast low-rank semidefinite programming for embedding and clustering. In *AISTATS*, 2007.

G. Lanckriet, N. Cristianini, P. Bartlett, L. Ghaoui, and M. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5(Jan):27–72, 2004.

S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2): 129–137, 1982.

J. Maxfield and H. Minc. On the matrix equation X' X= A. *Proceedings of the Edinburgh Mathematical Society (Series 2)*, 13(02):125–129, 1962.

D. Mixon, S. Villar, and R. Ward. Clustering subgaussian mixtures by semidefinite programming. Technical report, arXiv:1602.06612, 2016.

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3): 1042–1068, 2016a.

B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3): 1042–1068, 2016b.

C. Pehlevan, A. Genkin, and D. Chklovskii. A clustering neural network model of insect olfaction. In *Asilomar*, 2017.

J. Peng and Y. Wei. Approximating K-means-type clustering via semidefinite programming. *SIAM Journal on Optimization*, 18(1):186–205, jan 2007.

S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500), 2000.

W. So and C. Xu. When does CP-rank equal rank? Technical report, arXiv:1308.3193, 2013.

J. Tenenbaum, V. Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500), 2000.

M. Tepper and G. Sapiro. A bi-clustering framework for consensus problems. *SIAM Journal on Imaging Sciences*, 7(4):2488–2525, 2014.

M. Tepper and G. Sapiro. Compressed nonnegative matrix factorization is fast and accurate. *IEEE Transactions on Signal Processing*, 64(9):2269–2283, 2016.

K. Weinberger and L. Saul. An introduction to nonlinear dimensionality reduction by maximum variance unfolding. *AAAI*, 2006.

Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. *NIPS*, 2008.

Y. Xu, W. Yin, Z. Wen, and Y. Zhang. An alternating direction algorithm for matrix completion with nonnegative factors. *Front. Math. China*, 7(2):365–384, 2012.

L. Yang, D. Sun, and K. C. Toh. SDPNAL+: A majorized semismooth Newton-CG augmented Lagrangian method for semidefinite programming with nonnegative constraints. *Mathematical Programming Computation*, 7(3):331–366, 2015.

Y. Yu, J. Neufeld, R. Kiros, X. Zhang, and D. Schuurmans. Regularizers versus losses for nonlinear dimensionality reduction. In *ICML*, 2012.

P. Zhang, Y. Ren, and B. Zhang. A new embedding quality assessment method for manifold learning. *Neurocomputing*, 97:251–266, 2012.

Appendix A. Relationship with K-means

K-means seeks to cluster a dataset $\mathcal{X} = \{\mathbf{x}_i\}_{i=1}^n$ by computing

$$\min_{\boldsymbol{\mathcal{C}}_K} \sum_{k=1}^K \sum_{\mathbf{x}_i \in \mathcal{C}_k} \left\| \mathbf{x}_i - \tfrac{1}{|\mathcal{C}_k|} \sum_{\mathbf{x}_j \in \mathcal{C}_k} \mathbf{x}_j \right\|_2^2, \tag{K-means}$$

where $\boldsymbol{\mathcal{C}}_K = \{\mathcal{C}_k\}_{k=1}^K$ is a partition of $\mathcal{X}$, i.e., $\mathcal{C}_k \cap \mathcal{C}_{k'} = \emptyset$ and $\bigcup_k \mathcal{C}_k = \mathcal{X}$. Albeit its popularity, it is known to be NP-Hard and, in practice, users employ an heuristic (Lloyd, 1982, originally developed in 1957) to find a solution. The objective function of K-means, henceforth denoted J_K, can be rewritten (dropping the terms that are constant with respect to $\mathcal{C}_k$) as

$$J_K = -\sum_{k=1}^K \tfrac{1}{|\mathcal{C}_k|} \sum_{\mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}_k} \mathbf{x}_i^\top \mathbf{x}_j. \tag{23}$$

Let $\mathbf{X} \in \mathbb{R}^{d \times n}$ be the matrix formed by horizontally concatenating the vectors in $\mathcal{X}$. Let $\mathbf{z}_k \in \{0,1\}^n$ be the indicator vector of set $\mathcal{C}_k$. Let $\mathbf{Y}$ be the $k \times n$ matrix with rows $\mathbf{Y}_{k:} = |\mathcal{C}_k|^{-1/2} \mathbf{z}_k$. We have

$$\begin{aligned} J_K &= -\sum_{k=1}^K \tfrac{1}{|\mathcal{C}_k|} \sum_{i,j} \mathbf{x}_i^\top \mathbf{x}_j \cdot (\mathbf{z}_k^\top \mathbf{z}_k)_{ij} && \text{(24a)} \\ &= -\sum_{i,j} (\mathbf{X}^\top \mathbf{X})_{ij} (\mathbf{Y}^\top \mathbf{Y})_{ij} && \text{(24b)} \\ &= -\operatorname{Tr}\left(\mathbf{X}^\top \mathbf{X} \mathbf{Y}^\top \mathbf{Y}\right). && \text{(24c)} \end{aligned}$$

By construction, the matrix $\mathbf{Q} = \mathbf{Y}^\top \mathbf{Y}$ exhibits the following properties

$$\mathbf{Q}\mathbf{1} = \mathbf{1}, \tag{25}$$
$$\operatorname{Tr}(\mathbf{Q}) = K. \tag{26}$$

Let $\mathbf{D}$ be the Gramian matrix, i.e., $\mathbf{D} = \mathbf{X}^\top \mathbf{X}$. We can then re-cast K-means as the optimization problem

$$\max_{\mathbf{Y} \in \mathcal{V}_\mathbf{Y}^{K \times n}} \operatorname{Tr}(\mathbf{D}\mathbf{Q}) \quad \text{s.t.} \quad \begin{array}{c} \mathbf{Q}\mathbf{1} = \mathbf{1}, \\ \operatorname{Tr}(\mathbf{Q}) = K, \\ \mathbf{Q} = \mathbf{Y}^\top \mathbf{Y}. \end{array} \tag{27}$$

where $\mathcal{V}_\mathbf{Y} = \{0\} \cup \left\{|\mathcal{C}_k|^{-1/2}\right\}_{k=1}^K$. Seeking to apply the desirable properties of SDP to K-means, we can pose (Kulis et al., 2007; Peng and Wei, 2007)

$$\max_{\mathbf{Q} \in \mathbb{R}^{n \times n}} \operatorname{Tr}(\mathbf{D}\mathbf{Q}) \quad \text{s.t.} \quad \begin{array}{c} \mathbf{Q}\mathbf{1} = \mathbf{1}, \\ \operatorname{Tr}(\mathbf{Q}) = K, \\ \operatorname{rank}(\mathbf{Q}) = K, \\ \mathbf{Q} \succeq 0,\ \mathbf{Q} \geq \mathbf{0}. \end{array} \tag{28}$$

where mixed-integer program is relaxed into the real-valued nonnegative program, directly optimizing over $\mathbf{Q}$. NOMAD is as a relaxation of this problem, simply obtained by removing the rank constraint.

Appendix B. On the complete positivity of NOMAD solutions on circulant matrices

Proposition 2 *If the solution $\mathbf{Q}_*$ to NOMAD is a circulant matrix, then it is CP for every $K \leq 3/2$ or $K \geq \frac{n}{2}$.*

Proof For $K \leq 3/2$, plugging the constraint $\mathbf{Q}_*\mathbf{1} = \mathbf{1}$ into Corollary 2.6 in (So and Xu, 2013, p. 7) gives the desired result.

Let us address $K \geq \frac{n}{2}$. Kaykobad (1987) proved that every diagonally dominant matrix $\mathbf{A}$, i.e., $|(\mathbf{A})_{ii}| \geq \sum_{j \neq i} |(\mathbf{A})_{ij}|$ for all i, is a CP matrix. We have to prove then that $\mathbf{Q}_* \geq \mathbf{0}$ is diagonally dominant. We have $\operatorname{Tr}(\mathbf{Q}_*) = K$ and, since $\mathbf{Q}_*$ is circulant, all $(\mathbf{Q}_*)_{ii}$ have the same value. Then, $(\mathbf{Q}_*)_{ii} = K/n$. From $\mathbf{Q}_*\mathbf{1} = \mathbf{1}$, $\sum_{j \neq i} (\mathbf{Q}_*)_{ij} = 1 - (\mathbf{Q}_*)_{ii} = 1 - K/n$. Hence, $\mathbf{Q}_*$ is diagonally dominant for $K \geq \frac{n}{2}$. ■

Appendix C. Additional results

We include additional results of the multi-layer NOMAD algorithm using different values of k in each layer.

Appendix D. Symmetric NMF

In this section, we present the algorithm used to compute the symmetric NMF of a matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$, defined as

$$\min_{\mathbf{Y} \in \mathbb{R}^{n \times r}} \left\| \mathbf{A} - \mathbf{Y}\mathbf{Y}^\top \right\|_F^2 \quad \text{s.t.} \quad \mathbf{Y} \geq \mathbf{0}. \tag{SNMF}$$

We use the alternating direction method of multipliers (ADMM) to solve it. In short, ADMM solves convex optimization problems by breaking them into smaller subproblems, which are individually

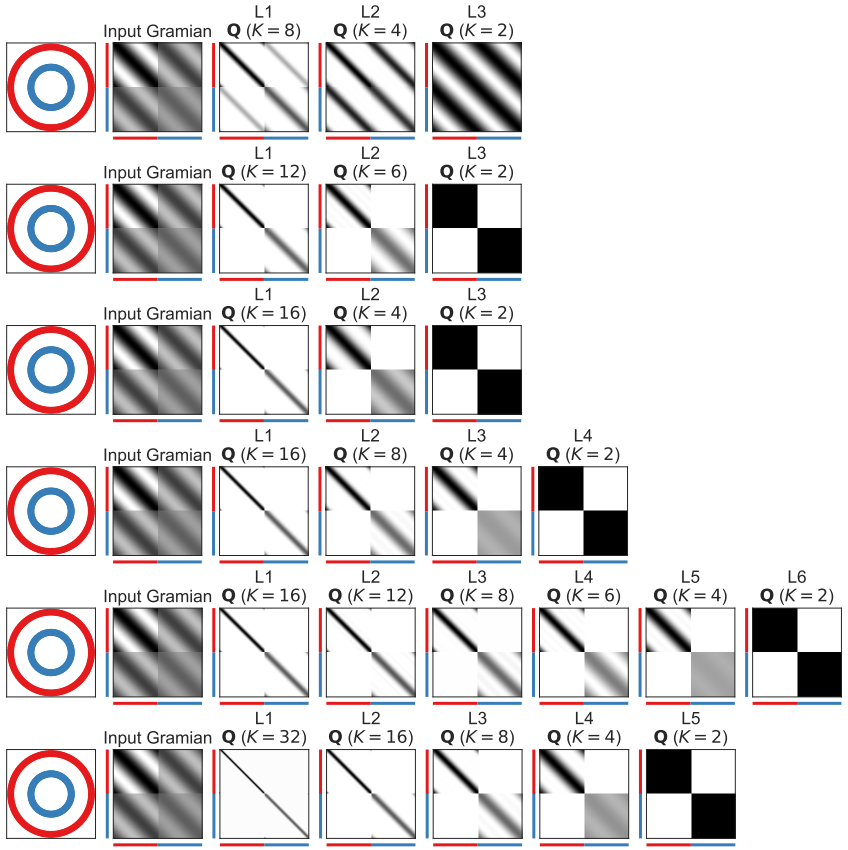


Figure 17: Additional results of multi-layer NOMAD.

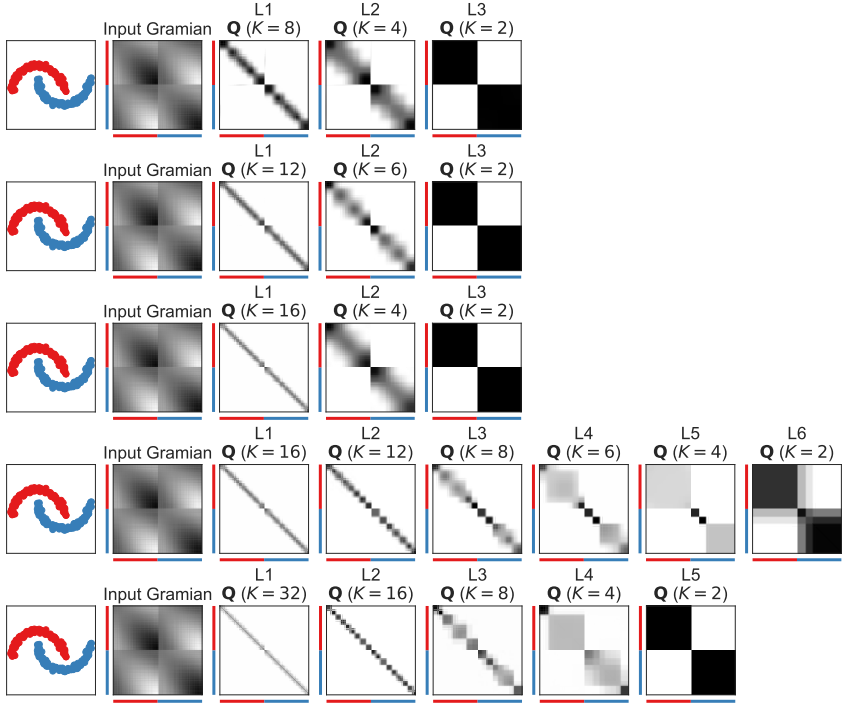


Figure 18: Additional results of multi-layer NOMAD.

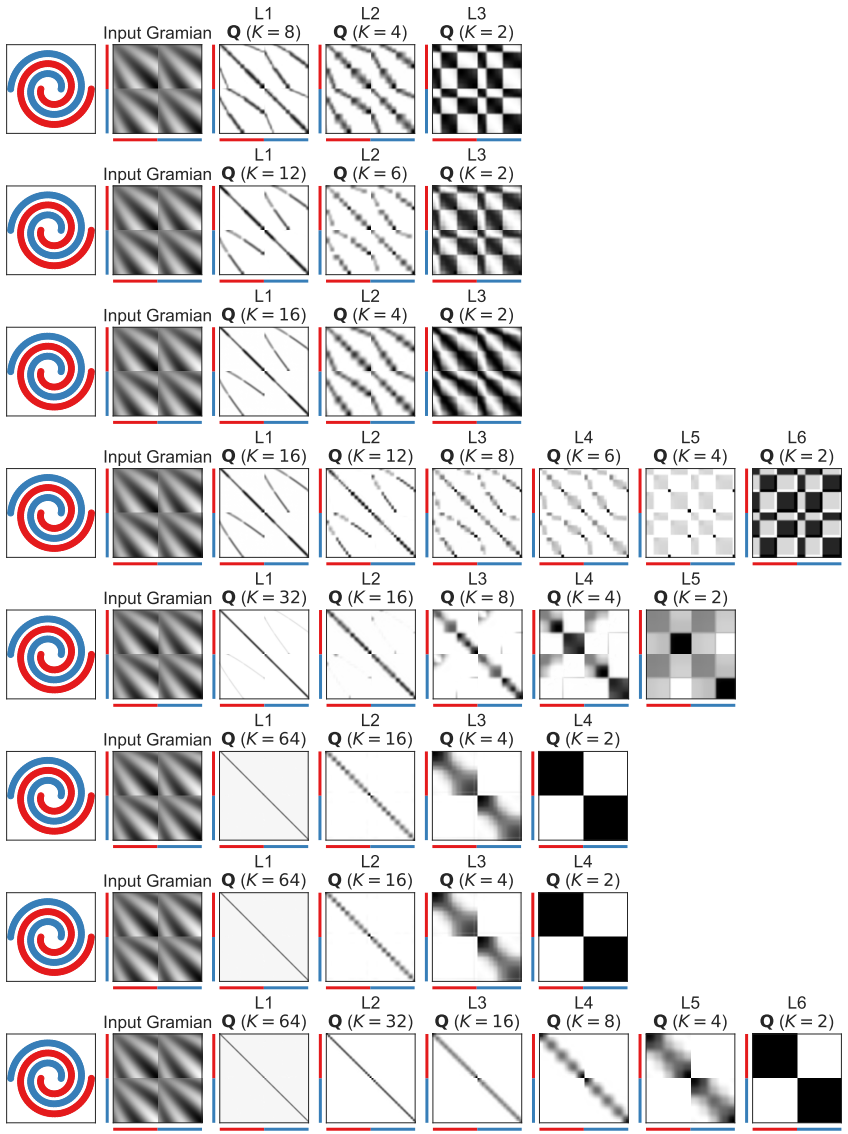


Figure 19: Additional results of multi-layer NOMAD.

easier to handle. It has also been extended to handle non-convex problems, e.g., to solve several flavors of NMF (Févotte and Idier, 2011; Xu et al., 2012; Tepper and Sapiro, 2014, 2016).

Problem (SNMF) can be equivalently re-formulated as

$$\min_{\mathbf{Y}\in\mathbb{R}^{n\times r}} \left\|\mathbf{A}-\mathbf{Y}\mathbf{X}^\top\right\|_F^2 \quad \text{s.t.} \quad \mathbf{Y}=\mathbf{X},\ \mathbf{Y}\geq\mathbf{0},\ \mathbf{X}\geq\mathbf{0}, \tag{29}$$

and we consider its augmented Lagrangian,

$$\mathscr{L}\left(\mathbf{X},\mathbf{Y},\mathbf{\Gamma}\right)=\tfrac{1}{2}\left\|\mathbf{A}-\mathbf{Y}\mathbf{X}^\top\right\|_F^2+\tfrac{\sigma}{2}\left\|\mathbf{Y}-\mathbf{X}\right\|_F^2-\operatorname{Tr}\left(\mathbf{\Gamma}^\top\left(\mathbf{Y}-\mathbf{X}\right)\right) \tag{30}$$

where $\mathbf{\Gamma}$ is a Lagrange multiplier, σ is a penalty parameter.

The ADMM algorithm works in a coordinate descent fashion, successively minimizing $\mathscr{L}$ with respect to $\mathbf{X},\mathbf{Y}$, one at a time while fixing the other at its most recent value and then updating the multiplier $\mathbf{\Gamma}$. For the problem at hand, these steps are

$$\mathbf{X}^{(t+1)}=\underset{\mathbf{X}\geq\mathbf{0}}{\operatorname{argmin}}\,\mathcal{L}\left(\mathbf{X},\mathbf{X}^{(t)},\mathbf{\Gamma}^{(t)}\right), \tag{31a}$$

$$\mathbf{Y}^{(t+1)}=\underset{\mathbf{Y}\geq\mathbf{0}}{\operatorname{argmin}}\,\mathcal{L}\left(\mathbf{X}^{(t+1)},\mathbf{Y},\mathbf{\Gamma}^{(t)}\right), \tag{31b}$$

$$\mathbf{\Gamma}^{(t+1)}=\mathbf{\Gamma}^{(t)}-\eta\sigma\left(\mathbf{X}^{(t+1)}-\mathbf{Y}^{(t+1)}\right). \tag{31c}$$

In our experiments, we fix η and σ to 1. We initialize the algorithm with a random matrix.

Appendix E. Non-convex SDP solver

We follow the algorithm proposed in Kulis et al. (2007); Hou et al. (2015) to solve Problem (11). Our approach has a small but fundamental difference: instead of setting $r=K$, we allow for $r\geq K$. We define the augmented Lagrangian of Problem (11) as

$$\begin{aligned}\mathcal{L}\left(\mathbf{Y},\mu,\lambda\right)=&-\operatorname{Tr}\left(\mathbf{D}\mathbf{Y}^\top\mathbf{Y}\right)+\tfrac{\sigma}{2}\left\|\mathbf{Y}^\top\mathbf{Y}\mathbf{1}-\mathbf{1}\right\|_2^2-\mu^\top(\mathbf{Y}^\top\mathbf{Y}\mathbf{1}-\mathbf{1})\\&+\tfrac{\varphi}{2}(\operatorname{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right)-K)^2-\lambda(\operatorname{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right)-K),\end{aligned} \tag{32}$$

where μ,λ are Lagrange multipliers, σ,φ are penalty parameters. We obtain $\mathbf{Y}$ by running the steps

$$\mathbf{Y}^{(t+1)}=\underset{\mathbf{Y}\geq\mathbf{0}}{\operatorname{argmin}}\,\mathcal{L}\left(\mathbf{Y},\mu^{(t)},\lambda^{(t)}\right), \tag{33a}$$

$$\mu^{(t+1)}=\mu^{(t)}-\eta\sigma\left(\mathbf{Y}^\top\mathbf{Y}\mathbf{1}-\mathbf{1}\right), \tag{33b}$$

$$\lambda^{(t+1)}=\lambda^{(t)}-\eta\varphi\left(\operatorname{Tr}\left(\mathbf{Y}^\top\mathbf{Y}\right)-K\right). \tag{33c}$$

This is a non-standard approach since the minimization over $\mathbf{Y}$ (the gradient $\partial\mathcal{L}/\partial\mathbf{Y}$ is given in Hou et al. (2015)) is a non-convex problem. Although there are no guarantees about the convergence of the procedure, theoretical assurances for related problems have been presented in Boumal et al. (2016). To perform the minimization with respect to $\mathbf{Y}$, we use the L-BFGS-B algorithm (Byrd et al., 1995) with bound constraints ($(\mathbf{Y})_{ij}\in[0,1]$). Finally, the initialization to the overall iterative algorithm is done with symmetric nonnegative matrix factorization, see Appendix D. In our experiments, we fix η, φ, and σ to 1 and prenormalize $\mathbf{D}$ (dividing by its Frobenius norm).

Appendix F. Proofs for the conditional gradient algorithm

This section closely follows the works of Hazan (2008) and Clarkson (2010).

Lemma 3 *The dual objective of*

$$\max_{\mathbf{Z}} f(\mathbf{Z}) \quad s.t. \quad \mathbf{Z}\mathbf{b} = \mathbf{0}, \quad \operatorname{Tr}(\mathbf{Z}) = k-1, \quad \mathbf{Z} \succeq \mathbf{0} \tag{34}$$

is

$$\min_{\mathbf{Z}} w(\mathbf{Z}), \tag{35}$$

where

$$w(\mathbf{Z}) = s\,\phi(\mathbf{Z}) + f(\mathbf{Z}) - \operatorname{Tr}\left(\mathbf{Z}\nabla f(\mathbf{Z})\right), \tag{36}$$

$$\phi(\mathbf{Z}) = \max_{\substack{\|\mathbf{v}\|_2=1 \\ \mathbf{v}^\top\mathbf{b}=0}} \mathbf{v}^\top \nabla f(\mathbf{Z})\mathbf{v}. \tag{37}$$

Proof The Lagrangian relaxation of Problem (19) is

$$\begin{aligned}\ell = -\max_{\boldsymbol{\Psi},\psi,\mathbf{y}} \min_{\mathbf{Z}} \; & -f(\mathbf{Z}) - \operatorname{Tr}(\boldsymbol{\Psi}\mathbf{Z}) \\ & + \psi(\operatorname{Tr}(\mathbf{Z}) - s) + \mathbf{y}^\top\mathbf{Z}\mathbf{b},\end{aligned} \tag{38}$$

where $\boldsymbol{\Psi} \succeq \mathbf{0}$, $\psi \in \mathbb{R}$, and $\mathbf{y} \in \mathbb{R}^n$ are Lagrange multipliers. Differentiating and equating to zero we get $\boldsymbol{\Psi} = -\nabla f(\mathbf{Z}) + \psi\mathbf{I} + \mathbf{y}\mathbf{b}^\top$. Note that $\boldsymbol{\Psi} \succeq \mathbf{0}$ implies that $\mathbf{y} = c\mathbf{b}$ for some $c \in \mathbb{R}$, yielding

$$\boldsymbol{\Psi} = -\nabla f(\mathbf{Z}) + \psi\mathbf{I} + c\mathbf{b}\mathbf{b}^\top \succeq \mathbf{0}. \tag{39}$$

Plugging Eq. (39) and $\mathbf{y} = c\mathbf{b}$ in Eq. (38) we get

$$\begin{aligned}\ell &= \max_{\psi} \min_{\mathbf{Z}} f(\mathbf{Z}) - \operatorname{Tr}\left(\left(\nabla f(\mathbf{Z}) + \psi\mathbf{I} + c\mathbf{b}\mathbf{b}^\top\right)\mathbf{Z}\right) \\ &\qquad + \psi(\operatorname{Tr}(\mathbf{Z}) - s) + c\mathbf{b}^\top\mathbf{Z}\mathbf{b} \qquad (40) \\ &= \max_{\psi\in\mathbb{R}} \min_{\mathbf{Z}} f(\mathbf{Z}) - \operatorname{Tr}\left(\nabla f(\mathbf{Z})\,\mathbf{Z}\right) + \psi s \qquad (41) \\ &= \max_{\substack{\mathbf{Z} \\ \psi\in\mathbb{R}}} f(\mathbf{Z}) - \operatorname{Tr}\left(\nabla f(\mathbf{Z})\,\mathbf{Z}\right) - \psi s. \qquad (42)\end{aligned}$$

From Eq. (39), $\psi\mathbf{I} \succeq \nabla f(\mathbf{Z}) - c\mathbf{b}\mathbf{b}^\top$, implying

$$\begin{aligned}\psi &\geq \lambda_{\max}\left\{\nabla f(\mathbf{Z}) - c\mathbf{b}\mathbf{b}^\top\right\} \qquad (43a) \\ &\geq \lambda_{\max}\left\{\nabla f(\mathbf{Z}) - d\mathbf{b}\mathbf{b}^\top\right\} \quad \forall d > c \qquad (43b) \\ &\geq \phi(\mathbf{Z}). \qquad (43c)\end{aligned}$$

The last inequality comes from taking $d \to +\infty$, thus shifting the eigenvalue associated with $\mathbf{b}$ (should there be one) away from the maximum. Without loss of generality, we set $\psi = \phi(\mathbf{Z})$, finally obtaining $\ell = \min_{\mathbf{Z}} w(\mathbf{Z})$.

Proposition 4 *Let $\mathbf{X}, \mathbf{Z} \in \mathcal{P}_s$ and $\mathbf{Y} = \mathbf{X} + \alpha(\mathbf{Z} - \mathbf{X})$ and $\alpha \in \mathbb{R}$. The curvature constant of f is*

$$C_f := \sup_{\mathbf{X},\mathbf{Z},\alpha} \tfrac{1}{\alpha^2}[f(\mathbf{X}) - f(\mathbf{Y}) + (\mathbf{Y} - \mathbf{X}) \bullet \nabla f(\mathbf{X})]. \tag{44}$$

Let $\mathbf{Z}^\star$ be the solution to

$$\max_{\mathbf{Z}} f(\mathbf{Z}) \quad s.t. \quad \mathbf{Z}\mathbf{b} = \mathbf{0}, \quad \operatorname{Tr}(\mathbf{Z}) = k-1, \quad \mathbf{Z} \succeq \mathbf{0}. \tag{45}$$

The iterates $\mathbf{Z}_t$ of Alg. 1 satisfy for all $t > 1$

$$f(\mathbf{Z}^\star) - f(\mathbf{Z}_t) \leq \tfrac{8C_f}{t+2}. \tag{46}$$

Proof Let

$$h(\mathbf{Z}) = \tfrac{1}{4C_f}[f(\mathbf{Z}^\star) - f(\mathbf{Z})]. \tag{47}$$

Proving that $h(\mathbf{Z}_t) \leq \frac{2}{t+2}$ yields the desired result. From Lemma 3, we have

$$w(\mathbf{Z}) \geq w(\mathbf{Z}^\star) \geq f(\mathbf{Z}^\star) \geq f(\mathbf{Z}). \tag{48}$$

By eqs. (36) and (37), we have

$$\begin{aligned}\mathbf{v}_t^\top \nabla f(\mathbf{Z}_t)\mathbf{v}_t &= \phi(\mathbf{Z}) \qquad (49) \\ &= w(\mathbf{Z}_t) - f(\mathbf{Z}_t) + \operatorname{Tr}\left(\mathbf{Z}\,\nabla f(\mathbf{Z}_t)\right). \qquad (50)\end{aligned}$$

Therefore,

$$\operatorname{Tr}\left((\mathbf{H}_t - \mathbf{Z}_t)\,\nabla f(\mathbf{Z}_t)\right) = w(\mathbf{Z}_t) - f(\mathbf{Z}_t). \tag{51}$$

Now, using Eq. (20)

$$\begin{aligned}f(\mathbf{Z}_{t+1}) &= f(\mathbf{Z}_t + \alpha_t(\mathbf{H}_t - \mathbf{Z}_t)) \qquad (52a) \\ &\geq f(\mathbf{Z}_t) + \alpha_t \operatorname{Tr}\left((\mathbf{H}_t - \mathbf{Z}_t)\,\nabla f(\mathbf{Z}_t)\right) + \alpha^2 C_f. \qquad (52b)\end{aligned}$$

Putting Eq. (52b) together with eqs. (48) and (51),

$$\begin{aligned}f(\mathbf{Z}_{t+1}) &\geq f(\mathbf{Z}_t) + \alpha_t(w(\mathbf{Z}_t) - f(\mathbf{Z}_t)) - \alpha_t^2 C_f \qquad (53a) \\ &\geq f(\mathbf{Z}_t) + \alpha_t(f(\mathbf{Z}^\star) - f(\mathbf{Z}_t)) - \alpha_t^2 C_f \qquad (53b) \\ &= f(\mathbf{Z}_t) + 4\alpha_t C_f h(\mathbf{Z}_t) - \alpha_t^2 C_f. \qquad (53c)\end{aligned}$$

From the definition of $h(\mathbf{Z}_t)$, this implies

$$h(\mathbf{Z}_{t+1}) \leq h(\mathbf{Z}_t) - \alpha_t h(\mathbf{Z}_t) + \tfrac{\alpha_t^2}{4}. \tag{54}$$

Finally, we prove inductively that $h(\mathbf{Z}_t) \leq \frac{2}{t+2}$. In the first iteration, $\alpha_1 = 1$ and $h(\mathbf{Z}_2) \leq \frac{1}{4}$. By taking $\alpha = \frac{2}{t+2}$, we have

$$\begin{aligned}h(\mathbf{Z}_{t+1}) &\leq h(\mathbf{Z}_t) - \alpha_t h(\mathbf{Z}_t) + \tfrac{\alpha_t^2}{4} \qquad (55a) \\ &\leq (1 - \alpha_t)h(\mathbf{Z}_t) + \tfrac{\alpha_t^2}{2} \qquad (55b) \\ &= (1 - \tfrac{2}{t+2})\tfrac{2}{t+2} + \tfrac{2}{(t+2)^2} \leq \tfrac{2}{t+3}. \qquad (55c)\end{aligned}$$

Journal of Machine Learning Research 19 (2018) 1-7 Submitted 3/18; Revised 11/18; Published 11/18

Seglearn: A Python Package for Learning Sequences and Time Series

David M. Burns d.burns@utoronto.ca
Sunnybrook Research Institute

Cari M. Whyne cari.whyne@sunnybrook.ca
Sunnybrook Research Institute
2075 Bayview Ave. Room S620.
Toronto, ON, Canada. M4N 3M5.

Editor: Alexandre Gramfort

Abstract

`seglearn` is an open-source Python package for performing machine learning on time series or sequences. The implementation provides a flexible pipeline for tackling classification, regression, and forecasting problems with multivariate sequence and contextual data. Sequences and series may be learned directly with deep learning models or via feature representation with classical machine learning estimators. This package is compatible with `scikit-learn` and is listed under `scikit-learn` "Related Projects". The package depends on `numpy`, `scipy`, and `scikit-learn`. `seglearn` is distributed under the BSD 3-Clause License. Documentation includes a detailed API description, user guide, and examples. Unit tests provide a high degree of code coverage. Source code and documentation can be downloaded from `https://github.com/dmbee/seglearn`.

Keywords: Machine-Learning, Time-Series, Sequences, Python

1. Introduction

Many real-world machine learning problems e.g. voice recognition, human activity recognition, power systems fault detection, stock price and temperature prediction, involve data that is captured as sequences over a period of time (Aha, 2018). Sequential data sets do not fit the standard supervised learning framework, where each sample $(\mathbf{x}, y)$ within the data set is assumed to be independently and identically distributed (iid) from a joint distribution $P(\mathbf{x}, y)$ (Bishop, 2011). Instead, the data consist *sequences* of $(\mathbf{x}, y)$ pairs, and nearby values of $(\mathbf{x}, y)$ within a *sequence* are likely to be correlated to each other. Sequence learning exploits the sequential relationships in the data to improve algorithm performance.

2. Supported Problem Classes

Sequence data sets have a general formulation (Dietterich, 2002) as sequence pairs $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{N}$, where each $\mathbf{X}_i$ is a multivariate sequence with T_i samples $\langle \mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,T_i} \rangle$ and each $\mathbf{y}_i$ target is a univariate sequence with T_i samples $\langle y_{i,1}, y_{i,2}, ..., y_{i,T_i} \rangle$. The targets $\mathbf{y}_i$ can either be sequences of categorical class labels (for classification problems), or sequences of continuous data (for regression problems). The number of samples T_i varies between the sequence pairs in the data set. Time series with a regular sampling period may be treated equivalently to sequences. Irregularly sampled time series are formulated with an additional sequence variable $\mathbf{t}_i$ that increases monotonically and indicates the timing of samples in the data set $\{(\mathbf{t}_i, \mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^{N}$.

Important sub-classes of the general sequence learning problem are sequence classification and sequence prediction. In sequence classification problems (eg song genre classification), the target for each sequence is a fixed class label y_i and the data takes the form $\{(\mathbf{X}_i, y_i)\}_{i=1}^{N}$. Sequence prediction involves predicting a future value of the target $(y_{i,t+f})$ or future values $\langle y_{i,t+1}, y_{i,t+2}, ..., y_{i,t+f} \rangle$, given $\langle \mathbf{x}_{i,1}, \mathbf{x}_{i,2}, ..., \mathbf{x}_{i,t} \rangle$, $\langle y_{i,1}, y_{i,2}, ..., y_{i,t} \rangle$, and sometimes also $\langle \mathbf{x}_{i,t+1}, \mathbf{x}_{i,t+2}, ..., \mathbf{x}_{i,t+f} \rangle$.

A final important generalization is the case where contextual data associated with each sequence, but not varying within the sequence, exists to support the machine learning algorithm performance. Perhaps the algorithm for reading electrocardiograms will be given access to laboratory data, the patient's age, or known medical diagnoses to assist with classifying the sequential data recovered from the leads.

`seglearn` provides a flexible, user-friendly framework for learning time series and sequences in all of the above contexts. Transforms for sequence padding, truncation, and sliding window segmentation are implemented to fix sample number across all sequences in the data set. This permits utilization of many classical and modern machine learning algorithms that require fixed length inputs. Sliding window segmentation transforms the sequence data into a piecewise representation (segments), which is particularly effective for learning periodized sequences (Bulling et al., 2014). An interpolation transform is implemented for resampling irregularly sampled time series. The sequence or time series data can be learned directly with various neural network architectures (Lipton et al., 2015), or via a feature representation which greatly enhances performance of classical algorithms (Bulling et al., 2014).

3. Installation

The `seglearn` source code is available at: `https://github.com/dmbee/seglearn`. It is operating system agnostic, and implemented purely in Python. The dependencies are `numpy`, `scipy`, and `scikit-learn`. The package can be installed using pip:

```
$ pip install seglearn
```

Alternatively, seglearn can be installed from the sources:

```
$ git clone https://github.com/dmbee/seglearn
$ cd seglearn
$ pip install .
```

Unit tests can be run from the root directory using `pytest`.

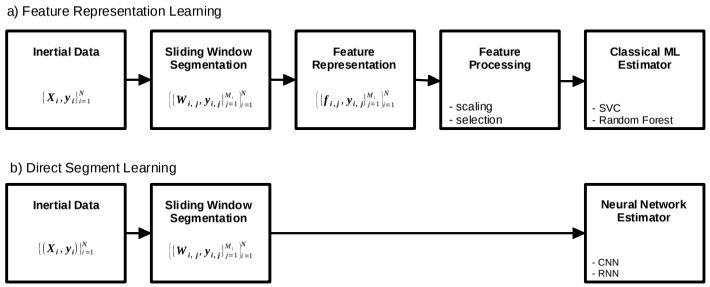


Figure 1: Example `seglearn` pipelines for a) learning segment feature representations, b) learning segments directly. $\boldsymbol{X}_i$: time series, $\boldsymbol{y}_i$: time series target, N: number of time series in the data set, $\boldsymbol{W}_{i,j}$: segment (derived from $\boldsymbol{X}_i$), $\boldsymbol{f}_{i,j}$: feature vector (calculated from $\boldsymbol{W}_{i,j}$), $\boldsymbol{y}_{i,j}$: segment target, M_i: number of segments derived from time series $\boldsymbol{X}_i$, SVC: Support Vector Classifier, CNN: Convolution Neural Network, RNN: Recurrent Neural Network.

4. Implementation

The `seglearn` API was implemented for compatibility with `scikit-learn` and its existing framework for model evaluation and selection. The `seglearn` package provides means for handling sequence data, segmenting it, computing feature representations, calculating train-test splits and cross-validation folds along the temporal axis.[1] An iterable, indexable data structure is implemented to represent sequence data with supporting contextual data.

The `seglearn` functionality is provided within a `scikit-learn` pipeline allowing the user to leverage `scikit-learn` transformer and estimator classes, which are particularly helpful in the feature representation approach to segment learning. Direct segment learning with neural networks is implemented in pipeline using the `keras` package, and its `scikit-learn` API. Examples of both approaches are provided in the documentation and example gallery. The integrated learning pipeline, from raw data to final estimator, can be optimized within the `scikit-learn model_selection` framework. This is important because segmentation parameters (eg window size, segment overlap) can have a significant impact on sequence learning performance (Burns et al., 2018; Bulling et al., 2014).

Sliding window segmentation transforms sequence data into a piecewise representation (segments), such that predictions are made and scored for all segments in the data set. Sliding window segmentation can be performed for data sets with a single target value per sequence, in which case that target value is mapped to all segments generated from the parent sequence. If the target for each is sequence is also a sequence, the target is segmented

1. Note splitting time series data along the temporal axis violates the assumption of independence between train and test samples. However, this is useful in some cases, such as the analysis of a single series.

as well and various methods may be used to select a single target value from the target segment (e.g. mean value, middle value, last value, etc.) or the target segment sequence can be predicted directly if an estimator implementing sequence to sequence prediction is utilized.

A human activity recognition data set (Burns et al., 2018) consisting of inertial sensor data recorded by a smartwatch worn during shoulder rehabilitation exercises is provided with the source code to demonstrate the features and usage of the `seglearn` package.

5. Basic Example

This example demonstrates the use of `seglearn` for performing sequence classification with our smartwatch human activity recognition data set.

```
>>> import seglearn as sgl
>>> from sklearn.model_selection import train_test_split
>>> from sklearn.ensemble import RandomForestClassifier
>>> from sklearn.preprocessing import StandardScaler
>>>
>>> data = sgl.load_watch()
>>> X_train, X_test, y_train, y_test = train_test_split(data["X"], data["y"])
>>>
>>> clf = sgl.Pype([("seg", sgl.SegmentX(width=100, overlap=0.5)),
...                 ("features", sgl.FeatureRep()),
...                 ("scaler", StandardScaler()),
...                 ("rf", RandomForestClassifier())])
>>>
>>> clf.fit(X_train, y_train)
>>> score = clf.score(X_test, y_test)
>>> print("accuracy score:", score)

accuracy score:  0.7805084745762711
```

6. Comparison to other Software

Three other Python packages for performing machine learning on time series and sequences were identified: `tslearn` (Tavenard, 2017), `cesium-ml` (Naul et al., 2016), and `tsfresh` (Christ et al., 2018). These were compared to `seglearn` based on time series learning capabilities (Table 1), and performance (Table 2).

`cesium-ml` (v0.9.6) and `tsfresh` (v0.11.1) support feature representation learning of multi-variate time series, and currently implement more features than does `seglearn`. However, the feature representation transformers are implemented as a pre-processing step, independent to the otherwise sklearn compatible pipeline. This design choice precludes end-to-end model selection. There are no examples or apparent support for problems where the target is a sequence/time series or integration with deep learning models.

	tslearn	cesium-ml	ts-fresh	seglearn
Active development (2018)	✓	✓	✓	✓
Documentation	✓	✓	✓	✓
Unit Tests	✓	✓	✓	✓
Multivariate time series	✓	✓	✓	✓
Context data	X	✓	X	✓
Time series target	X	X	X	✓
Sliding window segmentation	X	X	X	✓
Temporal folds	X	X	X	✓
sklearn compatible model selection	X	X	X	✓
Feature representation learning	X	✓	✓	✓
Number of implemented features	N/A	58	64	20
Deep learning	X	X	X	✓
Classification	✓	✓	✓	✓
Clustering	✓	✓	✓	✓
Regression	✓	✓	✓	✓
Forecasting	X	✓	✓	✓

Table 1: Comparison of time series learning package features for `tslearn` v0.1.18.4, `cesium-ml` v0.9.6, `tsfresh` v0.11.1 and `seglearn` v1.0.2.

`tslearn` (v0.1.18.4) implements time-series specific classical algorithms for clustering, classification, and barycenter computation for time series with varying lengths. There is no support for feature representation learning, learning context data, or deep learning.

The performance comparison was conducted using our human activity recognition data set with 140 multivariate time series with 6 channels sampled uniformly at 50 Hz and 7 activity classes. The series' were all truncated to 4 seconds (200 samples). Classification accuracy was measured on 35 series' held out for testing, and 105 used for training. `seglearn`, `cesium-ml`, and `tsfresh` were tested using the sklearn implementation of the SVM classifier with a radial basis function (RBF) kernel on 5 features (median, minimum, maximum, standard deviation, and skewness) calculated on each channel (total 30 features). `tslearn` was evaluated with its own SVM classifier implementing a global alignment kernel (Cuturi et al., 2007). The testing was performed using an Intel Core i7-4770 testbed with 16 GB of installed memory, on Linux Mint 18.3 with Python 2.7.12.

Classification accuracy was identical between `cesium-ml`, `tsfresh`, and `seglearn` (as they used the same features and classifier in the evaluation) though `seglearn` significantly outperformed the other packages in terms of computation time. Classification performance of the global alignment kernel SVM (GAK-SVM) implemented in `tslearn` was poor on our data set, even following hyper-parameter optimization of gamma by grid search over the log space $[10^{-4}, 10^{4}]$. GAK-SVM is not typically applied to raw inertial data for human activity recognition in the literature, and better performance has been achieved with this algorithm in other time series applications (Cuturi and Doucet, 2011; Lorincz et al., 2013).

	tslearn	cesium-ml	ts-fresh	seglearn
Classification accuracy	0.057	**0.714**	**0.714**	**0.714**
Computation time (seconds)	0.79	62.9	0.40	**0.088**

Table 2: Comparison of time series learning package performance on our human activity recognition dataset.

References

David Aha. UCI Machine Learning Repository, March 2018. URL `https://archive.ics.uci.edu/ml/index.php`.

Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2nd edition, April 2011. ISBN 978-0-387-31073-2.

Andreas Bulling, Ulf Blanke, and Bernt Schiele. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Computing Surveys*, 46(3):1–33, January 2014. ISSN 03600300. doi: 10.1145/2499621.

David Burns, Nathan Leung, Michael Hardisty, Cari Whyne, Patrick Henry, and Stewart McLachlin. Shoulder Physiotherapy Exercise Recognition: Machine Learning the Inertial Signals from a Smartwatch. *arXiv:1802.01489 [cs]*, February 2018. arXiv: 1802.01489.

Maximilian Christ, Nils Braun, Julius Neuffer, and Andreas W. Kempa-Liehr. Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package). *Neurocomputing*, 307:72–77, September 2018. ISSN 0925-2312. doi: 10.1016/j.neucom.2018.03.067.

Marco Cuturi and Arnaud Doucet. Autoregressive Kernels For Time Series. *arXiv:1101.0673 [stat]*, January 2011. arXiv: 1101.0673.

Marco Cuturi, Jean-Philippe Vert, Oystein Birkenes, and Tomoko Matsui. A Kernel for Time Series Based on Global Alignments. In *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, pages II–413–II–416, Honolulu, HI, April 2007. IEEE. ISBN 978-1-4244-0727-9. doi: 10.1109/ICASSP.2007.366260.

Thomas G. Dietterich. Machine Learning for Sequential Data: A Review. In *Structural, Syntactic, and Statistical Pattern Recognition*. Springer, Berlin, Heidelberg, 2002. ISBN 978-3-540-44011-6 978-3-540-70659-5. doi: 10.1007/3-540-70659-3_2.

Zachary C. Lipton, John Berkowitz, and Charles Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

Andras Lorincz, Laszlo Jeni, Zoltan Szabo, Jeffrey Cohn, and Takeo Kanade. Emotional Expression Classification using Time-Series Kernels. *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 889–895, June 2013. doi: 10.1109/CVPRW.2013.131. arXiv: 1306.1913.

Brett Naul, Stfan van der Walt, Arien Crellin-Quick, Joshua S. Bloom, and Fernando Prez. cesium: Open-Source Platform for Time-Series Inference. *arXiv:1609.04504 [cs]*, September 2016. arXiv: 1609.04504.

Romain Tavenard. tslearn: A machine learning toolkit dedicated to time-series data, 2017. URL `https://github.com/rtavenar/tslearn`.

Journal of Machine Learning Research 19 (2018) 1-5 Submitted 6/18; Revised 10/18; Published 11/18

DALEX: Explainers for Complex Predictive Models in R

Przemysław Biecek przemyslaw.biecek@gmail.com
Faculty of Mathematics and Information Science, Warsaw University of Technology
75 Koszykowa Street, Warsaw, Poland
Samsung Research Poland

Editor: Alexandre Gramfort

Abstract

Predictive modeling is invaded by elastic, yet complex methods such as neural networks or ensembles (model stacking, boosting or bagging). Such methods are usually described by a large number of parameters or hyper parameters - a price that one needs to pay for elasticity. The very number of parameters makes models hard to understand.

This paper describes a consistent collection of explainers for predictive models, a.k.a. black boxes. Each explainer is a technique for exploration of a black box model. Presented approaches are model-agnostic, what means that they extract useful information from any predictive method irrespective of its internal structure. Each explainer is linked with a specific aspect of a model. Some are useful in decomposing predictions, some serve better in understanding performance, while others are useful in understanding importance and conditional responses of a particular variable.

Every explainer presented here works for a single model or for a collection of models. In the latter case, models can be compared against each other. Such comparison helps to find strengths and weaknesses of different models and gives additional tools for model validation. Presented explainers are implemented in the `DALEX` package for R. They are based on a uniform standardized grammar of model exploration which may be easily extended.

Keywords: interpretable machine learning, explainable artificial intelligence, predictive modelling, model visualization

1. Introduction

Predictive modeling has a large number of applications in almost every area of human activity, starting from medicine, marketing, logistic, banking and many others. Due to the increasing amount of collected data, models become more sophisticated and complex.

It is believed that there is a trade-off between the interpretability and accuracy of a model (see Johansson et al., 2011). It comes from the observation that the most elastic models usually have higher accuracy but in turn they are also more complex. Complexity here means a large number of model parameters that affect the final prediction. That number is big enough to make the model ununderstandable for an ordinary human being.

In many areas we cannot sacrifice interpretability, either because of legal requirements (see *right to explanation* in GDPR), or because it leads to unfair decisions (see O'Neil, 2016) or because it is important for users (see Lundberg and Lee, 2017). Interpretability brings multiple benefits such as: a) helps to extract interpretable patterns from trained models; b) helps to identify reasons behind poor predictions; c) increases trust in model predictions (see Ribeiro et al., 2016); d) reduces the hidden debt in machine learning models (see Sculley et al., 2015); e) helps to detect bias in machine learning models; f) creates additional safety catch that may protect from overfitted models.

In this paper we present a consistent general framework for exploration of black-box models. This framework covers the most known approaches to interpretability and structure exploration, such as Partial Dependence Plots (Greenwell, 2017), Accumulated Local Effects Plots (Apley, 2017), Merging Path Plots (Sitko and Biecek, 2017), Break Down Plots (Staniak and Biecek, 2018), Permutational Variable Importance Plots (Fisher et al., 2018) or Cateris Paribus Plots. An unique feature of `DALEX` explainers is that they can be natively used to compare two or more models. Model comparison helps to understand differences in model responses, gives new insights that may be used to construct new, better features.

Presented framework is available as an open source package `DALEX` for R. The R language (R Core Team, 2017) is one of the most popular languages for statistical and machine learning modeling. `DALEX` works with any predictive model. The extended user documentation[1] contains examples for the most popular frameworks, such as `caret` (Kuhn, 2008), `mlr` (Bischl et al., 2016), Random Forest and Gradient Boosting Machines. The `DALEX` package is available on at CRAN and GitHub[2] along with technical documentation[3].

Example explainers presented in this paper were recorded with the `archivist` package (Biecek and Kosinski, 2017). To save space, we present only graphical explainers. Numerical explainers can be downloaded with R commands listed in footnotes.

2. Architecture

Figure 1 presents the general architecture of the `DALEX` package. This methodology is model-agnostic and works for predictive models, such as classification or regression models.

Methods for understanding of global structure of a model (a.k.a. model explainers) and for understanding of a local structure of a model (a.k.a. prediction explainers) are implemented in separate functions. We call these functions *explainers* since they are designed to explain a single feature of a model. Every explainer returns numerical summaries in a tabular format. These tables may be visualized with generic `plot` function. The `plot` function works also for multiple models and overlays model explainers in a single chart. See examples in Figure 1 panels F, I and J.

2.1. Prediction Understanding: Explainers for Variable Attribution

The most known approaches to explanations of a single prediction are *LIME* method (Ribeiro et al., 2016), for local variable importance, and *Shapley values* (Lundberg and Lee, 2017), for local variable attribution. *Break Down Plots* are fast approximations of *Shapley values*. Comparison of these methods is presented in Staniak and Biecek (2018). An example for these explainers[4] is presented in Figure 1 panels C and D.

Note, that for non additive models, the local model behaviour may be very different from global model behaviour. Consider $f(x_1, x_2) = x_1 * x_2$ around point $(0, 0)$.

1. User documentation is available at `https://pbiecek.github.io/DALEX_docs`.
2. Development version is available at `https://github.com/pbiecek/DALEX`.
3. Technical documentation is available at `https://pbiecek.github.io/DALEX`.
4. Access this explainer with `archivist::aread('pbiecek/DALEX_arepo/72b47')`.

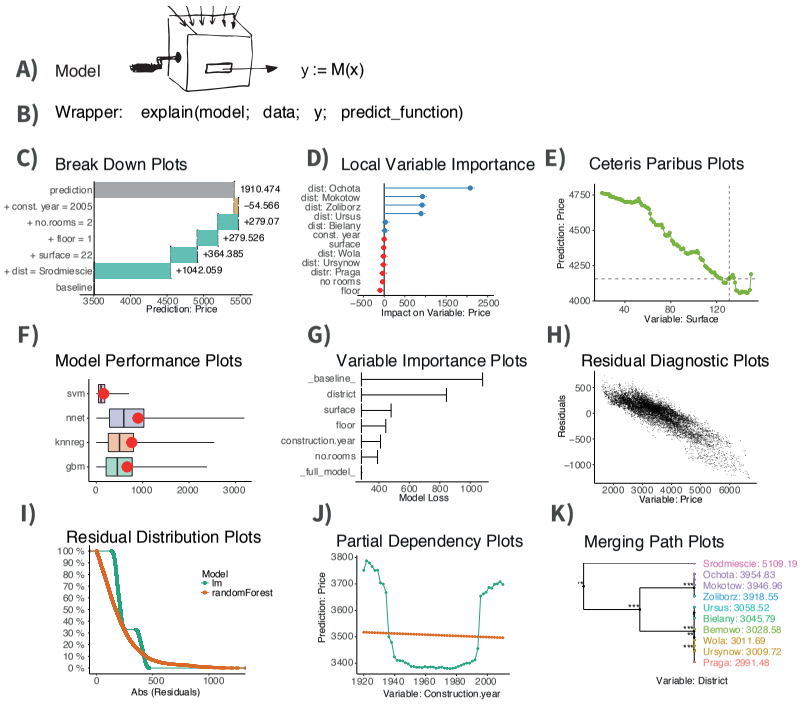


Figure 1: Architecture of the `DALEX` package is based on simple unified grammar. A) Any predictive model $M : \mathcal{R}^p \to \mathcal{R}$ may be used. B) Models are first enriched with additional metadata with the function `explain()`. Each explainer returns numerical summaries that can be plotted with generic `plot()` function. C, D, E) Explainers for a single prediction. F) Comparison of four models. G, H) Explainers for a single model, I) Comparison of residuals for two models. J, K) Explainers for a single variable, respectively continuous and categorical.

2.2. Prediction Understanding: Explainers for What-If Scenarios

Ceteris Paribus Profiles show how the model response changes as a function of a single variable. These plots recollect similarities to more known *Partial Dependency Plots*. The difference between them is that *Ceteris Paribus Profiles* are focused on a single observation.

An example for this explainer[5], is presented in Figure 1 panel E. One can read how model response will change for an altered value of a single variable.

5. Access this explainer with `archivist::aread('pbiecek/DALEX_arepo/c8989')`.

2.3. Model Understanding: Explainers for Model Performance

Model performance is often summarized with a single number such as *F1* or *accuracy*. This makes it easier to construct a ranking of models and choose the best one. However, more descriptive statistics are better when it comes to understanding of a model. The descriptive statistics most often used for classification is *ROC (Receiver Operating Characteristic)* with various extensions for regression as in Hernndez-Orallo (2013).

The `DALEX` package offers a selection of tools for exploration of model performance, see Figure 1 panels F and I[6], and model diagnostic, see Figure 1 panel H. The latter is available through the `auditor` package (Gosiewska and Biecek, 2018), closely integrated with `DALEX`.

2.4. Model Understanding: Explainers for Effect of a Single Variable

The `DALEX` package offers a selection of tools for better understanding of a conditional model's response based on a single variable. For continues variables it supports *Partial Dependence Plot* (Greenwell, 2017) as implemented in the `pdp` package and *Accumulated Local Effects Plot* (Apley, 2017) as implemented in `ALEPlot` package, see Figure 1 panel J[7]. For categorical variables it supports *Merging Path Plot* (Sitko and Biecek, 2017) as implemented in the `factorMerger` package. See Figure 1 panel K.

2.5. Model Understanding: Explainers for Variable Importance

The `DALEX` package offers a model-agnostic procedure to calculate variable importance. The model-agnostic approach is based on permutational approach introduced initially for *Random Forest* (Breiman, 2001) and then extended for other models by Fisher et al. (2018).

An example for these explainers[8] is presented in Figure 1 panel G. It's common in variable importance charts to hitch bars in 0. Charts in the `DALEX` package present not only drop in model performance but also the initial model performance. In that way one can compare variables between models with different initial performance.

3. Summary

In this article we have introduced consistent methodology and tools for model-agnostic explanations. Global explainers (for model understanding) and local explainers (for prediction understanding) are based on uniform grammar. Every explainer creates a numerical summary, visual summary and allows for comparison of multiple models. The `DALEX` package is tested with CI tools and is easy to extend[9]. Here we presented `DALEX` 0.2.5 with `R` 3.5.1.

Acknowledgments

The work was financially supported by RENOIR Project under the Marie Skłodowska-Curie grant agreement No 691152 and NCN Opus grant 2016/21/B/ST6/02176.

6. Access this explainer with `archivist::aread('pbiecek/DALEX_arepo/b4eb1')`.
7. Access this explainer with `archivist::aread('pbiecek/DALEX_arepo/3b150')`.
8. Access this explainer with `archivist::aread('pbiecek/DALEX_arepo/9378c')`.
9. Extended version of this paper is available at `https://arxiv.org/pdf/1806.08915v1.pdf`.

References

Dan Apley. *ALEPlot: Accumulated Local Effects (ALE) Plots and Partial Dependence (PD) Plots*, 2017. R package version 1.0.

Przemyslaw Biecek and Marcin Kosinski. archivist: An R Package for Managing, Recording and Restoring Data Analysis Results. *Journal of Statistical Software*, 82(11):1–28, 2017. doi: 10.18637/jss.v082.i11.

Bernd Bischl, Michel Lang, Lars Kotthoff, Julia Schiffner, Jakob Richter, Erich Studerus, Giuseppe Casalicchio, and Zachary M. Jones. mlr: Machine Learning in R. *Journal of Machine Learning Research*, 17(170):1–5, 2016.

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

Aaron Fisher, Cynthia Rudin, and Francesca Dominici. *Model Class Reliance: Variable Importance Measures for any Machine Learning Model Class, from the "Rashomon" Perspective*, 2018. URL `http://arxiv.org/abs/1801.01489`.

Alicja Gosiewska and Przemyslaw Biecek. *auditor: an R Package for Model-Agnostic Visual Validation and Diagnostic*, 2018. URL `https://arxiv.org/abs/1809.07763`.

Brandon Greenwell. pdp: An R package for Constructing Partial Dependence Plots. *The R Journal*, 9(1):421–436, 2017.

Jos Hernndez-Orallo. ROC curves for regression. *Pattern Recognition*, 46(12):33953411, 2013. doi: 10.1016/j.patcog.2013.06.014.

Ulf Johansson, Cecilia Snstrd, Ulf Norinder, and Henrik Bostrm. Trade-off between accuracy and interpretability for predictive in silico modeling. *Future Medicinal Chemistry*, 3(6): 647–663, 2011. doi: 10.4155/fmc.11.23.

Max Kuhn. Building Predictive Models in R Using the caret Package. *Journal of Statistical Software*, 28(5), 2008. doi: 10.18637/jss.v028.i05.

Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems 30*, pages 4765–4774, 2017.

Cathy O'Neil. *Weapons of Math Destruction: How Big Data Increases Inequality and Threatens Democracy*. Crown Publishing Group, New York, NY, USA, 2016.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?"*. ACM Press, 2016. doi: 10.1145/2939672.2939778. URL `https://arxiv.org/abs/1602.04938`.

David Sculley, Gary Holt, Daniel Golovin, Eugene Davydov, Todd Phillips, Dietmar Ebner, Vinay Chaudhary, Michael Young, Jean-Francois Crespo, and Dan Dennison. Hidden technical debt in machine learning systems. In *Proceedings of the 28th International Conference on Neural Information Processing Systems*. MIT Press, 2015.

Agnieszka Sitko and Przemyslaw Biecek. *The Merging Path Plot: adaptive fusing of k-groups with likelihood-based model selection*, 2017. URL `https://arxiv.org/abs/1709.04412`.

Mateusz Staniak and Przemyslaw Biecek. *Explanations of Model Predictions with live and breakDown Packages*, 2018. URL `https://arxiv.org/abs/1804.01955`.